UNIVERSITAT POLITÈCNICA DE CATALUNYA
Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial

Identifiability and calibration of water network models

Tesi presentada per Ramon Pérez Magrané

Terrassa , Juny 2003

Identifiability and calibration of water network models

Thesis presented at the Department of *Enginyeria de Sistemes automàtica i Informàtica Industrial* of the *Universitat Politècnica de Catalunya* in order to get the European PhD degree by Ramon Pérez.

Directors: Dr. Joseba Quevedo Casin
Dr. Bogumil Ulanicki

Terrassa , June 2003

Al meu pare Angel

Agraïments

Gràcies als meus pares que m'han ensenyat a caminar. Tota la meva família, en especial la Joana, Aloma, Laia i Arnau, m'heu donat molta alegria. Les amistats de Reus, l'origen de les quals ja s'ha oblidat, m'heu donat constància. Navegant m'heu ensenyat a aprofitar el vent encara que vingui de cara. A Calaf hi he trobat cada any tranquilitat. Gràcies per la benvinguda dels amics egarencs,de naixement o adopció. Joseba, el teu entusiasme ha estat necessari pel dia a dia. Bardzo Ci dziekuje Bogumil za twoja dokladnosc. I had my best tea and my best pints in Leicester. Drew, you have improved my spelling and writing. La xocolata per sant Ramon, les maleses dels diabòlics, els professors de Terrassa i Manresa, els que no doneu classe per que ja us donem prou feina (tu també Joan) m'heu donat una envejable atmòsfera. Espero que mi redacción tenga influencias de tí, Mercedes. Luís, gracias por tu pragmatismo y enseñarme a apreciar lo bueno. Tinc la paraula gràcies al llavis sovint perquè és més fàcil dir-la en el moment que intentar recordar tots els que la mereixeu.

Index

Glossary

Area	area of a reservoir
C*	cotree of a graph
G*	augmented graph
H(K)	matrix of mass balance equation with flows
HG(k)	Index set of signomial terms substituted for k time steps
IG(k)	Index set of posinomial terms
R	resistance of a pipe
R0	true resistance of a pipe
T*	tree of a graph
J	Jacobian
U	subset of a graph Set of output singular values
V	Set of input singular values
Q^t	t subproblem generated in branch and bound algorithm
S	feasible region in optimisation problems
S^t	feasible region of the t subproblem
Y	admittance of a pipe
b	number of pumps
d	demand pattern
d0	true demand pattern
f(x)	cost function
h	head of junctions
h0	head of junctions measurement
h(x), g(x)	equality and inequality constraints
hr	level of reservoirs
hr0	level of reservoir measurements

l	number of pipes
n	number of junctions
q	flow through a pipe
q0	flow through a pipe measurement
r	number of reservoirs
u	input of a system
v	number of valves
w	demand factor
w0	true demand factor
x	variable
x_r	variable corresponding to parameter
x_t	variable corresponding to state variable
x*	minimiser
z	measurement
Σ	matrix of singular values
Λ	incidence matrix
ρ	weigh factor in cost function descriptor in classification process posinomial functions
σ	singular value
ξ	measurement error

Abstract

Process control and supervision is based mainly in the use of models, which have to be as accurate as possible. Complex processes, like water distribution networks, fall into such a situation too. Water is a necessary element and its shortage in good conditions is a major problem. Therefore, a good management of water distribution is vital.

This thesis has been carried out with the collaboration of two research groups. One is more oriented to the application -water networks- and the other one is more oriented to the technology –control and supervision-. Experience of both groups has generated the necessity of calibration of water network models in order to be able to do good simulations, optimisations, supervisions, leak detections, etc. For this reason, the main objective of this thesis is to develop a closed methodology for the calibration process of water distribution network models.

The originality of this work comes both from the magnitude of the problem and the techniques used. On the one hand, a water network includes different elements (nodes, reservoirs, pipes, valves, and pumps), and its calibration requires the study of those elements in detail. On the other hand it has to be applied to huge systems. Special attention has been paid to the three main parts of calibration: identifiability study, macrocalibration and microcalibration. Each of those steps needs specific techniques. Some of the techniques used in this thesis are little known or unknown at all in the water industry.

The identifiability study has been developed for different case studies, ranging from simple, illustrating case to real huge networks. The simplest experiments were performed with linear and static networks. In general, networks are non-linear and the use of more than one time-step in the measurements provides better identifiability conditions. The methodology proposed allows the determination of the extended-period identifiability for general networks (non-linear). The obtained tool helps in the design of identification problems using topological information of the network.

When the model is generated, large errors are introduced. These errors are detected in a first calibration effort, macrocalibration. This process is done manually and the objective in this thesis is to give support to such work. The methodology followed by the experts has been analysed. Specific algorithms have been used in this thesis for each kind of error. In order to detect errors in huge amount of elements classification algorithms have been used. Those algorithms allow the generation of knowledge from simulation experiments and optimisation of likelihood functions.

The parameter tuning, microcalibration, is treated as an optimisation problem. The non-convexity of the problem is detected by a detailed characterisation. This non-convexity shows to be a problem for the local optimisers. The capabilities of some global optimisation algorithms have been explored. The computing cost for the global optimisers, especially when huge networks are identified, represents a major limitation. The Extended Kalman Filter has been used with promising results.

Results of this thesis have been presented in three Conferences [Per-01a], [Per-01b], [Per-03] and a paper is will be finished before the end of the year.

El control i supervisió de processos es basa generalment en la utilització de models. Models que han de ser tan acurats com sigui possible. Processos complexes com les xarxes de distribució d'aigua no escapen d'aquesta situació. Una bona gestió d'aquest element tan necessari i cada cop més escàs en les condicions adequades és una necessitat vital.

Aquesta tesi ha estat realitzada amb la col·laboració de dos grups de recerca, un més orientat a l'aplicació –xarxes d'aigua- i l'altre ala metodologia –control i supervisió -. L'experiència d'ambdós grups va generar la necessitat de la calibració de models de xarxes per tal de poder realitzar bones simulacions, optimitzacions, supervisió, detecció de fuites, etc. L'objectiu principal d'aquesta tesi és desenvolupar una metodologia per aquesta calibració.

L'originalitat d'aquest treball rau tan en l'abast del problema com les tècniques emprades. Una xarxa inclou elements diversos i la calibració requereix l'estudi detallat de cada element així com l'aplicació a sistemes immensos. Els diversos passos per assolir una bona calibració requereixen tècniques específiques. En aquesta tesi s'ha aplicat tècniques poc o gens conegudes en l'àmbit de les xarxes d'aigua.

Tres grans parts es poden destacar. L'estudi d'identificabilitat que s'ha realitzat per diferents casuístiques partint dels casos més senzills i il·lustratius fins a sistemes reals. L'eina obtinguda permet dissenyar els problemes d'identificació a partir de

característiques tipològiques de la xarxa. Per als errors més grans i que normalment s'introdueixen en el moment de generar el model s'ha investigat algoritmes de classificació. Aquesta tècnica permet generar el coneixement a partir d'experiments de simulació i minimització de funcions de versemblança. S'han explorat les possibilitats de l'optimització global per superar els reptes que representa la no convexitat dels problemes de sintonia de paràmetres.

Els resultats d'aquesta tesi s'han presentat en tres congressos [Per-01a], [Per-01b], [Per-03] i s'està redactant un article per a revista abans de final d'any.

1. Introduction

The first civilisations appeared by the big rivers. The flowing waters of the Euphrates, the Nile, the Indus and the Yangtze were silent witnesses of the human settlements along their banks and the flourishing of their culture [Rub-02]. They switched from subsistence agriculture to organised farming in a short time. Water and civilisation, two terms historically associated that now appear unsettled and in some cases contradictory. The reason? The shortage of this odourless, tasteless and in small quantities colourless liquid resource. The world-wide consumption of the last century has increased seven fold. In Spain (data from the hydrologic plans that are gathered in the White Book of the Water in Spain) the actual demand is 35,000 hm^3/year, 68% corresponds to agricultural irrigation, 18% usage by population and industry and 14% in the cooling of energy stations.

In the last half of the XX century the answer to this increase in the demand was basically the construction of more and larger hydraulic infrastructures, especially reservoirs and canals for deviating rivers. More than 85% of the 40,000 reservoirs built in the world have been constructed in the last 35 years. In Spain (Greenpeace [Gre-03]) there are 1,200 reservoirs, the country with the largest number of such infrastructures for surface area and inhabitants. Such engineering works have guaranteed the supply of water to great urban and rural areas but have, in the opinion of ecologists, led to the degradation of the fluvial deltas and increased the risk of extinction of some species in wetland areas.

These developments, that have given an extraordinary hydraulic patrimony and supplied more than 40 million inhabitants, irrigated more than three million hectares and produced hydroelectric energy needed for the development of the country, are now in crisis. Nobody can ignore the fact that it is not possible to satisfy a demand without limit with a permanent increase of the offer of a commodity that has ecological, physical and economic limitations. As the most accessible sources of water are exhausted new resources are obtained in an increasingly complicated and hence more

expensive way. This leads to a worse quality of the resource and the necessity of a new culture of water usage based on a more rational and sustainable use of this valuable resource.

The challenge is to spend less and more efficiently. To guarantee a sufficient supply of water does not in itself suffice to solve the distribution problems. It is necessary to continue conserving water. This is a broad concept that includes all those techniques orientated to help in the saving and better management of this liquid. These Techniques appear in the White Book of Water [Min-00].

Such techniques include the modernisation and rehabilitation of the networks to minimise leakage. This is a problem that affects not only the urban centres where 30% of the water that enters the network does not get to its destination at the consumption points but also the watering infrastructures. There are 100,000 kilometres of canals where 30% are 100 years old and most of the remainder more than 40 years old. Installation of low consumption devices nowadays allows savings of 50% without losing quality in the service. There are models of taps, showers and toilets with such improved efficiency. Wastewater can be reused after a good process of depuration. Education campaigns are crucial for saving water and the introduction of new tariffs can stimulate such savings. The contamination of both surface (rivers become increasingly polluted as they pass by cities and industrial areas) and ground water (polluted by nitrates, heavy metals and organic components and affected by *salinisation*) should be reduced.

Drinking water quality is not an objective any more but an obligation. For centuries, Europe had to cope with plagues and epidemics unaware of their origin. In 1854 in London, during a cholera epidemic, Dr John Snow discovered that the means of dissemination of this disease was water. He treated the water with chlorine and it eradicated the disease. After this experience the temporary treatment of water with chlorine began, first in the United Kingdom and afterwards in the United States becoming generalised at the beginning of XX century. It brought about the reduction of diseases such as Hepatitis A, Cholera and Typhus. The purification of water with chlorine has been the sanitary measure that has saved most lives in the last century.

In Spain the modern treatment stations (ETAP) use slow filtration systems that reproduce the conditions of the riverbeds. In these installations during 1999 five millions cubic meters of water were treated, this was the volume of water available for purification. In Spain the European legislation defines three types of water depending on which treatment it undergoes. This ranges from the simplest physical treatment and disinfecting to the most sophisticated physical and chemical treatment and disinfecting. So now in the quick filtration plants water undergoes treatment that includes different phases: caption, mixing with coagulating and reactive substances, decantation and sand

separation, flocculation, sedimentation, filtration and disinfecting. This procedure may last five hours and some new tools have been added. These technologies allow the fulfilment of the quality standards. Active carbon (produced by the combustion in special and controlled conditions of organic substances) that presents a big exposed area can trap by absorption suspended particles and dissolved substances. The disinfecting process using Ozone is very efficient but is more expensive than the use of chlorine. Inverse osmosis uses membranes to separate dissolved salts. In general solar technologies applied to the purification of water are only useful for water with low contamination.

Desalination of water is the other choice to increase the availability of water instead of reusing residual water. It is a perfectly viable solution from a technological point of view and widely used in Spain, where there are more than 700 plants working. The problem is the high cost of the technology. Nowadays the water obtained by this process is the most expensive in Spain. Nevertheless the latest advances in this technology and the introduction of inverse osmosis have decreased the energetic cost to 4Kwh for each cubic metre. The cost seems to be decreasing and the supporters of this technology believe that soon this water will be cheaper on the East Coast of Spain than the cost in the cession of water from the river Ebro included in the *Plan Hidrologico Español.* On the other hand the energy needed is a great drawback for obtaining a sustainable supply of water by this method. Anyway this technology that used to be restricted to the Canary Islands and Ceuta is extended now to the Baleares, Murcia, and the Costa del Sol. An important plant is working in Almeria where there is the biggest desalination plant in Europe with a production estimated at 120,000 cubic meters a day, that can be duplicated, uses the inverse osmosis and will solve the supply problems of the nearby urban centres.

Although the chemical purification methodologies guarantee that the water that comes from the taps is all right from the sanitary point of view in the last few years the consumption of bottled water has increased spectacularly. This happens despite consumers having to pay between 500 and 1000 times more for this water than that in the public network, and that in more than 50% of cases this bottled water has the same quality as the tap water except for some salts and aggregated minerals. Consumption of bottled water increases in the world by 7% each year. The reason has to be found in the fact that consumers don't like tap water, sometimes because it has a disagreeable taste or odour or sometimes because it can appear white colour.

Aware of this problem the distribution companies have started plans to determine and increase the quality of the water offered to their customers. For example the *Societat General d'Aigües de Barcelona* (Agbar), in collaboration with the French society *Lyonaise des Aux* and the North American *Water Works Association Research*, have developed a system to quantify the colour, taste and odour of the drinking water. If there

is any case in which one of these characteristics becomes unpleasant the causes are discovered and corrected. For this purpose there are some professionals that can detect 30 different tastes and smells in the water like the ones who work with wine. Typical tastes of chlorine, humid soil, metallic, cooked vegetable or the classical acid, sweet, salt and bitter can define the water. After the taste process the chemical analysis allows the detection of substances that produce such problems and the water can be treated adequately.

All these initiatives support the idea that is frequently held amongst the specialists; that bottled water does not solve the problem of water healthiness. First of all, as the association WWF/Adena says, because it should be able to take for granted that the water that comes from the taps and wells is pure and healthy for human consumption. On this purpose the association points towards the more environmental friendly agricultural techniques. Because the purity of the public networks' water would increase, instead of the dependence on bottled water, with an additional benefit to the environment. This reasoning is justified also in the fact that 25% bottled water is consumed some distance away from the producing plant which means high environmental costs related to the manufacturing and transport of this water.

Actually what is apparent after all these facts is that the water that is consumed is cheap but the solutions to guarantee its availability are not. In Spanish cities, the price per cubic metre in the year 2000 oscillated between 0.1 EUR in Melilla or 0.23 EUR in Huesca and 1.25 EUR or 1.3 EUR paid by a Balear or Barcelona citizen. Though the notable differences reflect the difficulties that each have to obtain potable water (especially dramatic on the Mediterranean coast) the reality is that water is a limited resource and should be treated not just as a social commodity, but an economical one too. Citizens should use water in the most efficient way and pay for the real cost of this precious resource. The Worldwatch Institute estimates that only 15% of the real cost is paid, a fact which does not encourage water conservation [Wor-01].

Careful usage as recommended by experts, should be supported by different initiatives. Specific legislation and recommendations for the population about this necessity would help avoid wastage. The adoption of financial incentives would stimulate the substitution (for uses that do not require high quality water) of the potable water with that coming from the regeneration of residual waters. The objective of these programmes would be the adjustment of the resource to the demands in order to liberate high quality flows to the most appropriate uses. It is not sensible to use potable water to water the plants, wash the car, or clean the streets.

1.1. Use of models

The problem outlined in the previous section affects all of society. Some of the solutions suggested so far depend on the customers, others depend on the governments, some depend on production companies and some on the distribution companies. Distribution of water also presents some challenges.

Distribution has improved in the last century although it existed in ancient civilisations. In particular there are many *Ponts del Diable* (aqueducts) in Europe that are an important part of the civil engineering legacy of the Roman Empire. The name in Catalan comes from the belief, in the high Middle Age, that such big structures had been built by the Devil. Of course the health conditions and standards have changed since those first distribution networks. The quality control of the water that flows through the network is an obligation nowadays. The physical conditions of the water offered to the customers, (if not so vital as its sanitation) is an objective of all the distribution companies. It is necessary to assure a pressure of water and a reliability of the service so that people, industry, hostelry, hospitals and all of society can trust in a basic service. Another important aspect already mentioned is the huge quantity of water that disappears in the network before it reaches its target.

Supervision of processes is part of the CIM (Computer Integrated Manufacturing) pyramid and it takes an important place in it. The knowledge provided by supervision allows an understanding of what happens in an industry or plant. This knowledge is indispensable for any quality certification, as the quality concept has more to do with the reliability of the data and assertions than to improve an unknown level. This knowledge is useful to correct any failure in the process or to improve its functioning. In a distribution network such supervision uses the data-loggers distributed in the network (the communications are often by radio due to the remote location of these measures), informatic applications that show the working state of the network and, depending on the company, some decision tools that help the human operators. In the figure 1.1 an integrated system as proposed in the European project WATERNET, Knowledge Capture for Advanced Supervision of Water Distribution Networks, ESPIRIT IV nº 22,186 (June 1996- June 1998). [Cem-97] is shown. In it the Distributed Information Management System (DIMS) gets data from Remote Units. These data are used for all other packages.

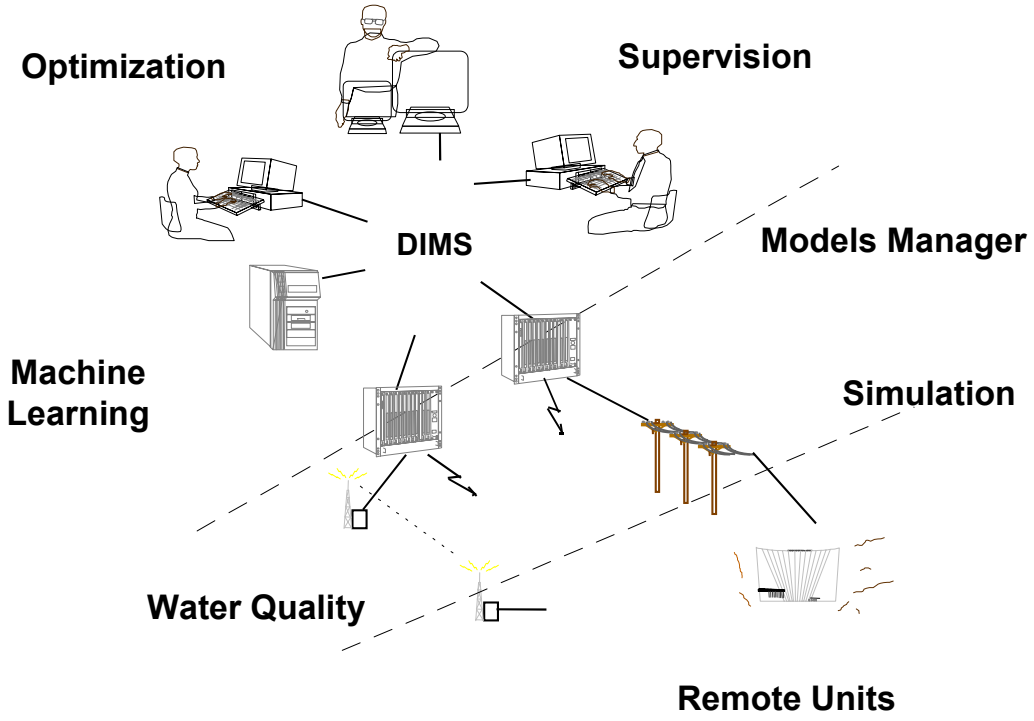


Fig. 1. 1

In this project a complete tool for water distribution network management was developed. In the figure there appear with the supervisor other tools that in a modern system are increasingly important. The optimisation tool tries to save money by taking advantage of the different electric and water tariffs. The machine learning is used to generate demand forecast and to make decision in case of fault detection in the systems behaviour. The water quality tool uses models of the network to forecast the chlorine concentration and age of the water in the network in order to assure the standards. Remote units send the information to the supervisor. Simulation uses models to forecast the hydraulic behaviour of the network so that the supervisor and optimiser can compare the reality with the expected behaviour. The model manager is a very important piece as all the tools use a kind of model. In figure 1.2 the different models used in this project are presented.

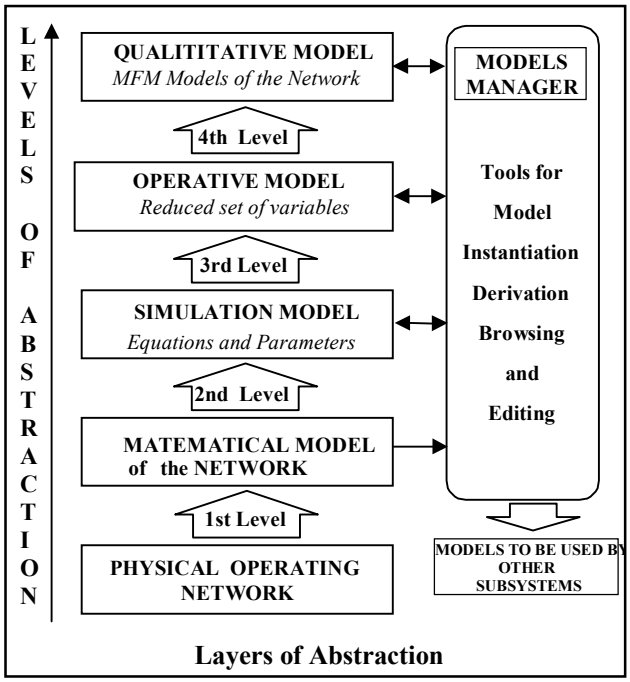


Fig. 1. 2

From the physical-operating model, where the real devices are, an initial theoretical model is developed. The second level of abstraction gives a model suitable for simulation with all the nodes, pipes, pumps, valves and reservoirs. This model gives the forecasted hydraulic behaviour of the network. For optimisation and even for the quality simulation such a huge model has to be simplified which is done in the next level by deleting small pipes, reservoirs, nodes and thus obtaining a small model that represents the behaviour. Finally, the machine-learning tool uses a qualitative model.

The main problem that appeared in this project was the reliability of the model. It should have been calibrated by one of the partners. They developed a heuristic way for the calibration and simplification but did not get good results in the timing required by the rest of the tasks. Finally our group used a manual way for the adjustment of parameters in order to produce a model for the optimisation. As can be seen in figure 1.2 the lack of a good simulation model invalidates the entire model management tool and most of the tools are therefore useless for that network.

1.2. Calibration Process

The ability to model larger water distribution systems has improved considerably over the past decades. The cost to set up and execute a mathematical representation or network simulation model has fallen with the advent of widespread usage of microcomputers. However the cost for calibration of the simulation model and for data collection required for estimating parameters has not decreased. In many cases, due to

the high expense and perceived lack of benefits, a thorough calibration effort is not performed. Without good parameter estimates, a simulation model may not be modelling reality and therefore the planning or operating decisions made, based on simulation analysis, may be in serious error.

When calibrating a water distribution system, the desire is to determine the actual values of the system parameters such as pipe roughness. With those values, the system can be simulated for a range of conditions with confidence in the model's results. However, all the parameters are typically not measured directly due to the expense of data collection. Four types of data (state variables) may be available from the field tests or the network telemetry system: nodal pressure heads, nodal demands, pipe flow, and tank levels. All four state variables supply information which can be used to determine the proper values of the system's parameters. Pump operations and pressure reducing valve settings may also be collected during these tests.

The surrogate objective of the calibration exercise is to match the observed values of the state variables with those generated by the network simulation model. In practice, given a set or sets of measured state variables, engineers apply trial and error techniques with their judgement to vary the parameters and accomplish this task. Trial and error techniques are extremely tedious and have no guarantee of reasonable results. The number of parameters to be estimated complicates the task. For example, the transition capacity of a pipe diminishes at an unknown rate with the time, which varies from system to system and even within a given system. In addition, losses that occur through regulating or pressure reducing valves may increase with time or they may be closed when believed to be open. Many nodal demands are usually not known precisely, and must also be estimated at any instant or over a period of time. The process is further complicated by the amount of data required to give confidence in the parameter estimation. As more pipes are added to the network model, more data is necessary to accurately determine the true field values or to determine an equivalent model of the true system hydraulics in the case of a skeletal system model. If data is collected for a single load, compensating errors may occur in situations where more than one set of parameter values gives good modelling results. This may also occur when more than one demand pattern is analysed, but the range of different values should be reduced. Therefore, a calibration procedure should be capable of analysing multiple demand patterns while considering all of the unknown parameters.

Several researchers have proposed algorithms for system calibration. Walski [Wals-83] suggested lumping groups of pipes to equivalent pipes and determining the lumped coefficients using analytic equations for single demand patterns. When the number of unknown parameters equals the number of nodes or pipes for single load patterns, it is possible to rework the equations of continuity and energy to consider the parameters as unknown, which can be solved in an iterative manner ([Don-74], [Rah-80], [Gof-81],

and [Orm-86b]). Shamir [Sha-74] presented a method to calibrate the network using an optimisation algorithm which could analyse single demand patterns. Later, Coulbeck [Cou-85] linearised the network equations and used an optimisation procedure. Finally, Ormsbee [Orm-88] combined a simulation model and a modified Box Complex Direct Search method to consider sets of independent loads or extended time period simulations.

Ormsbee and Lingireddy [Orm-97] describe the calibration task. The water distribution system must be represented by a node-link database (links represent individual pipe sections, and nodes represent points in the system where two or more pipes join or where water is being input or withdrawn. Data associated with pipes include length, diameter and roughness. Data associated with nodes include elevation and demand associated. Physical data for reservoir (geometry, initial water level), pumps (pump flow-head characteristic curve) and valves must be obtained. In general the most uncertain model parameters include pipe roughness and nodal demands. Because of the difficulty of obtaining economic and reliable measurements, final model values are normally determined by model calibration. In general, a network model calibration can be divided into seven main steps:

Identify the intended use of the model: Before calibrating a model it is important to identify its intended use (pipe sizing, operational studies, design studies, water quality studies) and the associated type of hydraulic analysis (steady state versus extended-period). Water quality and operational studies require an extended-period analysis, whereas some design studies may be performed using a steady-state analysis. Walski [Wals-83] justifies the guidance about the collected field data required given by the intended use and the type of the analysis.

Determine model parameter estimates: In most models some degree of uncertainty is associated with several parameter-pipe roughness, demands assigned to each node-. Initial estimates of pipe roughness values may be obtained using average values from literature but this information's specific applicability decreases significantly as the pipes age increases. To obtain initial estimates of roughness it is best to divide the water distribution system into composite zones that contain pipes of similar material and age. Walski [Wals-84] describes parallel-pipe method and two-hydrant method for estimation of the roughness. Initial average estimates of nodal demands can be obtained by identifying a region of influence associated with each node, identifying the types of demand units in the service area, and multiplying the number of each type by an associated demand factor [Wals-95].

Collect Calibration Data: The most common types of data are those for flow rate, tank water level and pressure. Depending on the level of instrumentation and telemetry, much of the data may already be collected as part of normal operations.

Evaluate model results: Using telemetry data, the model simulates operating conditions (different simulators are available, EPANET, FINESSE, PICCOLO, SIAR, WATERNET, etc.) for the day the field data were collected. The predicted pressures, flows and tank water levels are then compared with the observed values in attempt to assess model accuracy. Deviations between results may be caused by several factors (erroneous parameters, erroneous data, incorrect network geometry).

Perform macrolevel calibration: If one (or more) of the measured state variable values is different from the modelled values by an amount deemed excessive (i.e. greater than 30%), the cause for the difference probably extends beyond errors in the estimates for either pipe roughness or demands. The only way to adequately address such macrolevel errors is to systematically review the data associated with the model and compare them with the field data.

Perform sensitivity analysis: Before attempting a microlevel calibration, it is helpful to perform a sensitivity analysis of the model to help identify the most likely source of model error.

Perform microlevel calibration: The parameters to be adjusted in this final phase of calibration are pipe roughness and nodal demands. Historically, most attempts at model calibration have employed an empirical or trial-and-error approach which can prove to be extremely time-consuming and frustrating. Calibration of multitank systems can sometimes be facilitated by collecting multiple data sets with all but one the tanks closed [Crui-92]. Several researchers have proposed different algorithms for use in automatically calibrating hydraulic network models. Most of these techniques have been restricted to steady-state calibration. These techniques have been based on the use of analytical equations [Wals-83], simulation models [Rah-80][Gof-81][Bou-91] and optimisation methods [Mer-83][Cou-85][Orm-89]. . The Water Software Systems group in Leicester proposes a microcalibration procedure [Ula-00].

1.3. Thesis objectives

The main objective of this thesis is the development of a tool for water distribution systems calibration. Such calibration is based in the last instance on optimisation that is used for solving inverse mathematical problems such as calibration. The search for a global minimum of an error function (difference between measurements and model) may lead to global optimisation algorithms. The analysis of the need of such algorithms and the study of the contribution they can make to the calibration are specific objectives.

A methodology for this development must be established, such a methodology should allow the application of this module to a wide variety of water distribution systems. This module will generate a calibrated model from the initial one and some real measurements that will allow the parameter estimation.

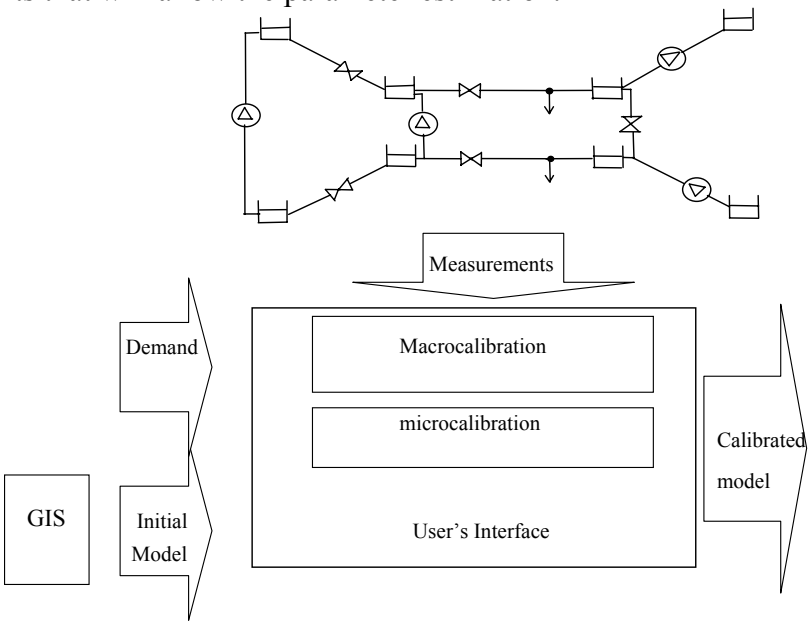


Fig. 1. 3

In figure 1.3 appear the tasks the calibration includes. From the GIS (Geographic Information System) database an initial model is extracted, and some measurements must be done. Using this information the topological and the parametrical calibration produce the definitive model. The calibration is divided in the macro, based on trial and error methodology dealing with crude errors, and the micro, based on optimisation for model parameter tuning. For each task some more concrete objectives have been defined.

The intended use of the models is simulation. In figure 1.2 these models appear as the previous stage of more abstracted models. These kinds of models are the key ones for the management objectives that have been suggested for a better usage of water. Simulation models work in an extended-period scenario. Parameters to be estimated are those that are less likely to be measured and change within time. Special emphasis will be put on pipe roughness and node demand factors but these could be generalised to pump curves or valves coefficients. These networks are often divided in regions containing similar material and age that generally are isolated by reservoirs and valves. These are the networks that will be calibrated in this thesis.

The theoretical identifiability of the model has to be studied. What is a minimum number of measurements that have to be taken in order to observe the state of the system? These measurements are taken in different loads, and the state of the system can include the parameters to be calibrated. A rigorous study will be carried out, but the result has to be a practical way of determining the identifiability of a network.

Connected with the identifiability the meter placement in order to get the correct data for the calibration will be studied, and a methodology developed to do this task. Once more a tool has to be the result of the work, so that companies that have to calibrate networks can design the experiments with an important saving in meters. This will provide some knowledge about the identifiability of the network depending on the topology.

Macrocalibration, the first step of the calibration, is carried out mainly in the industries by experts in a tiresome process of trial and error. An automatic system could carry out the macrocalibration that concerns the topological and crude errors in the model, or at least could support the task of the experts. Different approaches for such a challenge will be explored. The knowledge of experts will be organised and the methodology of the decision making defined.

The last step in the calibration, the parameter tuning or microcalibration, is based on optimisation. First of all a serious study of the requirements of the algorithms that will be used is carried out. A classification of the problems and algorithms is done in order to choose a proper algorithm. The optimisation module aims at finding a global minimum of the overall differences between the measurements and corresponding simulation resulting in a parameter tuning. There are different algorithms that solve this kind of problem. The research group of ESAII-Terrassa [Pui-99] is working on control applications of interval arithmetic, thus algorithms based on this arithmetic will be tried for this application. The equation formulation for water distribution network calibration using these algorithms and the development of the methodology that enables the application to other networks is an important objective. The numerical libraries that exist to solve such problems should be proven for this specific application and eventually improved.

A prototype for the calibration of networks will be constructed. This application will include the three tasks that are developed in this thesis. This prototype will be programmed basically in Matlab [Mat-03]. This choice is justified as the group is used to such an environment and the programming task is not a key point of this work. The prototype will be developed in such a way to allow remote work using the internet, so that more companies could test it and decide to apply it to their own problem.

This calibration methodology will be described in the thesis in order to allow industrial applications. The application to other networks is another objective in order to state the generality of the methodology. Small examples help one to understand the different phases of the process. Finally a huge network comparable with parts of real networks is used in each section to test the methodology.

1.4. Thesis organisation

This thesis is organised in an introduction, three main sections and a last section for conclusions. Results are presented in each section first for small networks that help in the understanding of the methodology, and a last case study. In the conclusions section the personal achievements, the objectives fulfilment and the future work proposed are included.

Section 2 presents the model that centres the attention of the work, equations and parameters. The identifiability study of this problem from different points of view - static, dynamic, extended-period, linear and non-linear- is done, and a methodology is chosen. The experimental design for the calibration based on this identifiability study is presented in this section.

The macrocalibration process is treated in section 3. First the expert work manually done is analysed. The knowledge organisation is chosen. The methodology is described and adapted to the macrocalibration purpose. The developed tool is tested with different networks and mismatches in them.

Finally in section 4 the microcalibration problem is studied. First of all the problem characterisation based on the classification of optimisation problems and algorithms, fixes the kind of algorithms that will be studied. Different algorithms are presented and applied to small examples to understand their way of working. These algorithms are presented in an order of improving results. Finally a real case is treated and the results obtained from the calibration using real data are presented.

2. Extended period Identifiability in Water Networks

The concept of state occupies a central position in modern control theory. However, it appears in many other technical and non-technical contexts as well. In thermodynamics the equations of *state* are prominently used. Binary sequential networks are normally analysed in terms of their *states*. In everyday life, monthly financial *state*ments are commonplace. The State of the nation debate is another familiar example. In all these examples the concept is essentially the same. It is a complete summary of the status of the system at a particular point in time. Knowledge of the state at some initial time t_0, plus the knowledge of the system inputs after t_0, allows the determination of the state at later time t_1. The systems of interest for this thesis are dynamic systems. The word dynamic refers to something active or changing with time. Figure 2.1 shows the space state representation for a discrete-time linear system where the state is defined only for discrete times.

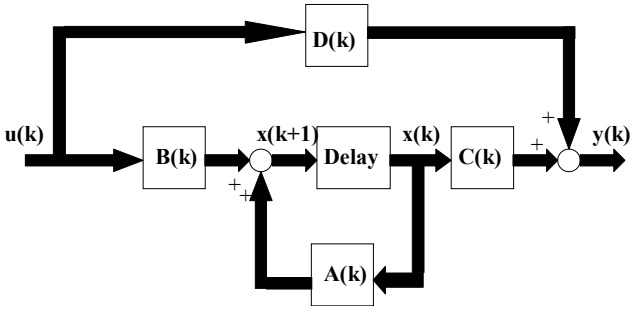


Fig. 2. 1: space state representation for a discrete-time linear system

One possible representation for the water distribution network is to use levels in reservoirs as state variables, inputs are demands and external apportions, outputs would be any measurement and the relation between two time steps (matrix A(k)) comes from the change of level in reservoirs. The physical water network behaviour is monitored through a telemetry system. However only a limited number of flows and heads are measured directly and they do not constitute a complete picture of the System State. It is therefore necessary to estimate unknown variables based on available measurements and mathematical models of the network.

Observability is a property of the coupling between the state and the output and thus involves the matrices A and C. A linear system is observable at t_0 if $\mathbf{x}(t_0)$ can be determined from the output function $\mathbf{y}(t_0,t_1)$ (or output sequence) for $t_0 \leq t_1$, where t_1 is some finite time. If this is true for all t_0 and $\mathbf{x}(t_0)$ it is called completely observable. Clearly Observability of a system will be a major requirement in filtering and state estimation or reconstruction problems.

In the water distribution systems, state estimation is not used in the same sense as in control. Most tasks of supervision are done using simulation. The model itself possesses parameters, which are not known exactly. The model needs to be calibrated to update the parameter values. Here the problem of finding these parameter will be treated as a generalisation of state estimation. The measurements are not ideal, the estimation-calibration problem is then formulated as a problem of minimising a suitably chosen measure of the inconsistencies.

Before the optimisation problem is stated and the algorithms to solve it are chosen it has to be proven that the system has a unique solution. This chapter treats the Identifiability (a concept close to observability) conditions of a water network that has to be calibrated. This problem includes the state estimation problem. In the first section the special case of water network calibration is presented with particularities of such a system.

It is always possible to tune the parameters (**p***) of a model so as to make its input-output behaviour (**M(p*)**) identical to that of the process (**M(p)**) for any time and input, which will be denoted by **M(p*)=M(p)**. If a system is identifiable it means that this identical input-output behaviour implies that the parameters **p*** of the model equals those of the process (**p**) [Wal-97].

The problem is presented in section 2.2. The parallelism of estimation and equation solution is treated in section 2.2. In section 2.3 the Identifiability concept in general and for these particular systems is presented. The classification based on the repetition of measurement follows in sections 2.4, 2.5 and 2.6. The identification problem design that is closely related with the identifiability is treated in sections 2.7 and 2.8. Small networks have been used in these sections as examples. In section 2.9 a bigger network is treated as a case study.

2.1. Water Network state and parameter estimation problem

The choice of state variables is not unique. Typically in a water network, nodal heads (including reservoir heads), that is **h** and $\mathbf{h_r}$, and reservoir inflows $\mathbf{q_{lr}}$ [Brd-94] are chosen as the state variables.

$$x = [h_r, h, q_{lr}] \tag{2.1}$$

The network model is assumed to be represented by flow continuity equations and by branch flow-head characteristics modelled by Hazen-Williams relations. Thus, the demands are expressed in terms of flows. The branch characteristics are used to express flows in terms of **x**.

$$\begin{aligned} &\Lambda\, q + q_{lr} - w * d = 0 \\ &\Delta h_j = R_j\, q_j \,|\, q_j \,|^{0.852} \qquad j = 1,2,...,l \end{aligned} \tag{2.2}$$

Where $\mathbf{\Lambda}$ is the incidence matrix of the network, **q** is the vector of flows [m^3/h] in the **l** pipes. The distribution of demand is assumed to be constant and represented by vector **w**, total demand, **d**, is the variable obtained by demand prediction and measurements in boundary flows. $\mathbf{\Delta h_j}$ [m] represents the head loss in pipe **j** which has $\mathbf{R_j}$ resistance and flow $\mathbf{q_j}$. Other elements such as valves can be understood as generalisation of pipes with minor changes in equations and for simplicity are not included in this study. The specificity of such elements is taken into account in the first step of calibration, *macrocalibration*. The knowledge used in this process depends on the behaviour and type of each element and even for different valves and pumps is different. In the *microcalibration* the work is restricted to pipes while a generalisation could be done without changing qualitatively the algorithm and results.

Some of these heads and flows are measured. All the heads and reservoir inflows should be determined in order to estimate the state of the network. Some parameters appear in the equations, such as resistances and demand factors. The state estimation problem assumes that these parameters are known so that a model of the network is available and the estimation will depend only on the observability of the system based on the measured variables. If all of these parameters are not known they will have to be estimated as well and the problem becomes a generalised one, a **state and parameter estimation**. The calibration process of a network includes the measurement of physical parameters (reservoir areas, length of pipes, etc.) but some parameters have to be estimated. These are generally resistances (or in alternative formulation, roughness) of pipes and demand distribution factors, this is the state and parameter estimation problem as it has been stated.

The main difference between variables and parameters is that the latter are constant in time, assuming a determined time horizon. For state estimation the system has been treated as a static one. Dynamics of the water systems would lead to transient equations the use of which is not generally extended in simulation, optimisation and, generally, in water management procedures. The quasi-static approach used in simulation comes from the reservoir dynamics. For each simulation time step the system is assumed static and a mass balance in reservoirs relates to each time step.

$$A(h_{ri}(t_1) - h_{ri}(t_2)) + \Lambda\, q = 0 \quad i = 1,2,...,r \tag{2.3}$$

Where **A** [m^2] is the area of the reservoir, a known parameter.

State estimation
The quasi-static problem does not improve state estimation qualitatively. It improves numerically as the measurements are not exact and, as it has been said, the problem is treated as an optimisation one.

Parameter estimation
A qualitative difference could appear as these parameters do not change in time so the number of variables do not increase exactly as the number of equations if the number of time steps increase. So this quasi-static approach is taken into account when the identifiability study is carried on. When parameters are estimated identifiability substitutes observability concept.

Example 2.1: In order to illustrate the difference between both problems when more than one time-step is used. Imagine a reservoir, a pipe and a junction as appears in figure 2.2.

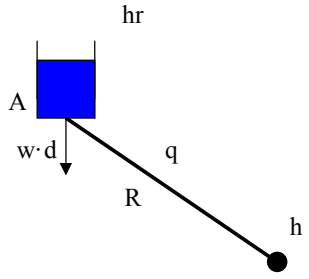


Fig. 2. 2

These are the equations of the reservoir:

$$(hr(k) - hr(k+1))\cdot A = q + w\cdot d$$

In the first example it is assumed that all parameters – Area (**A)** and demand coefficient **(w)-** are known. Level of the reservoir is measured but flow and total demand are not measured. With one time step there is one equation, the state is not observable because there are two variables unknown. If another time step is taken there are two equations and four variables, as both demand and flow change each time step. The equations can be rewritten as follows:

$$(hr(1)-hr(2))\cdot A = q(1) + w\cdot d(1)$$
$$(hr(2)-hr(3))\cdot A = q(2) + w\cdot d(2)$$
$$\begin{bmatrix} 1 & w & 0 & 0 \\ 0 & 0 & 1 & w \end{bmatrix} \begin{bmatrix} q(1) \\ d(1) \\ q(2) \\ d(2) \end{bmatrix} = \begin{bmatrix} (hr(1)-hr(2))*A \\ (hr(2)-hr(3))*A \end{bmatrix}$$

Numerically it is clear that the system of equations is non-determined as the matrix is 2x4:

$$hr = \begin{bmatrix} 120 \\ 119 \\ 118.5 \end{bmatrix}; A = 50; w = 0.3$$

$$\begin{bmatrix} 1 & w & 0 & 0 \\ 0 & 0 & 1 & w \end{bmatrix} \begin{bmatrix} q(1) \\ d(1) \\ q(2) \\ d(1) \end{bmatrix} = \begin{bmatrix} (hr(1)-hr(2))*A \\ (hr(2)-hr(3))*A \end{bmatrix}$$

On the other hand if flow and total demand are known and what is searched for are the parameters A and w equations are rewritten for two time steps (with only one time step the problem is not soluble as there are two variables and one equation):

$$(hr(1)-hr(2))A = q(1) + w*d(1)$$
$$(hr(2)-hr(3))A = q(2) + w*d(2)$$
$$\begin{bmatrix} (hr(1)-hr(2) & -d(1) \\ (hr(2)-hr(3) & -d(2) \end{bmatrix} \begin{bmatrix} A \\ w \end{bmatrix} = \begin{bmatrix} q(1) \\ q(2) \end{bmatrix}$$

Now numerically it is seen that the problem is determined and the solution is unique.

$$hr = \begin{bmatrix} 120 \\ 119 \\ 118.5 \end{bmatrix}; d = \begin{bmatrix} 100 \\ 40 \end{bmatrix}; q = \begin{bmatrix} 20 \\ 13 \end{bmatrix}$$
$$\begin{bmatrix} 1 & -100 \\ 0.5 & -40 \end{bmatrix} \begin{bmatrix} A \\ w \end{bmatrix} = \begin{bmatrix} 20 \\ 13 \end{bmatrix} \Rightarrow \begin{bmatrix} A \\ w \end{bmatrix} = \begin{bmatrix} 50 \\ 0.3 \end{bmatrix}$$

So the increase of the time steps is qualitatively different in the parameter estimation problem and in the state estimation problem.

2.2. Estimation problem and equation solution

The parameter and state estimation problem in static approach appears to be a set of equations that have to be solved (equation 2.4). The variables are the unmeasured heads, flows and the unknown parameters (x). The state estimation problem would be a particular case with no parameters in variable vector. As the network equations are non-linear it is a non-linear equations set that has to be solved. The inaccuracy (ξ) in measurements (z) leads us to treat the problem as a minimising one, resolution techniques are not discussed here. The equivalence with the equation set solution is used for the identifiability study discussed in further sections.

$$z = g(x) + \xi \tag{2.4}$$

A network has **r** reservoirs, **n** junctions (nodes will be used in general for both reservoirs and junctions), and **l** pipes. Two vectors could be defined, the non-measured vectors of heads ($\mathbf{h}_{r-}$, $\mathbf{h}_{-}$), flows ($\mathbf{q}_{-}$), resistance ($\mathbf{R}_{-}$) and demand factors ($\mathbf{w}_{-}$) and the measured or known vectors ($\mathbf{h}_{r+}$, $\mathbf{h}_{+}$, $\mathbf{q}_{+}$, $\mathbf{R}_{+}$, $\mathbf{w}_{+}$). The expression of equations is too general if has to be applied to any network.

$$\begin{aligned} & g_{1-}(q_-, w_-, d) = g_{1+}(q_-, w_-, d, q_{lr}) \\ & g_{2-}(q_-, h_-, R_-, q_+, R_+) = g_{2+}(h_+, q_+, R_+) \qquad j = 1,2,...,l \\ & \sum w_- = 1 - \sum w_+ \end{aligned} \tag{2.5}$$

The last equation comes from the normalisation of the demand factors.

Example 2.2: The simple network showed in figure 2.3 will be used to illustrate the equations formulation of the system. The measurements are levels in r1 and r2, head in junction n1, flows pipes p1 and p4 and both demand factors w1 and w2.

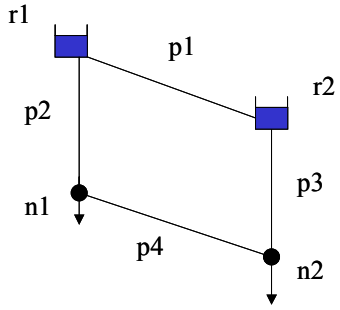


Fig. 2. 3

$$\Lambda=\begin{bmatrix}-1 & -1 & 0 & 0\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & 1\end{bmatrix};\quad \begin{aligned}-q_2 &= -A_1(h_{r1}(2)-h_{r1}(1))+q_1\\ -q_3 &= -A_2(h_{r2}(2)-h_{r2}(1))-q_1\\ q_2 &= q_4+w_1*d\\ q_3 &= -q_4+w_2*d\end{aligned};\quad \begin{aligned}R_1\ q_1\,|\,q_1\,|^{0.852} &= -h_{r1}+h_{r2}\\ R_2\ q_2\,|\,q_2\,|^{0.852} &= -h_{r1}+h_1\\ -h_2+R_3\ q_3\,|\,q_3\,|^{0.852} &= -h_{r2}\\ -h_2+R_4\ q_4\,|\,q_4\,|^{0.852} &= -h_1\end{aligned}$$

These equations have to be changed if some new measurement is introduced. For the general development and easier modification of the problem, adding sensors or changing the network, all the variables and parameters are included in vector **x=[h_r h q R w]** and treated as unknown variables. As a variable is measured or a parameter known a new equation is introduced. For this particular case:

$$x=\begin{bmatrix}h_{r1} & h_{r2} & h_1 & h_2 & q_1 & q_2 & q_3 & q_4 & h_{r1} & h_{r2} & R_1 & R_2 & R_3 & R_4 & w_1 & w_2\end{bmatrix}$$

$$\begin{aligned}&A_1(x_9-x_1)-x_5-x_6=0\\ &A_2(x_{10}-x_2)+x_5-x_7=0\\ &x_6-x_8-x_{15}*d=0\\ &x_7+x_8-x_{16}*d=0\end{aligned};\quad \begin{aligned}x_1-x_2-x_{11}*x_5*|x_5|^{0.852}&=0\\ x_1-x_3-x_{12}*x_6*|x_6|^{0.852}&=0\\ x_2-x_4-x_{13}*x_7*|x_7|^{0.852}&=0\\ x_3-x_4-x_{14}*x_8*|x_8|^{0.852}&=0\\ x_{15}+x_{16}&=1\end{aligned};$$

Where

$$\begin{aligned}x_1&=h_{r1}(1)\\ x_2&=h_{r2}(1)\\ x_3&=h_1\\ x_5&=q_1\\ x_8&=q_4\end{aligned};\quad \begin{aligned}x_9&=h_{r1}(2)\\ x_{10}&=h_{r2}(2)\\ x_{15}&=w_1\\ x_{16}&=w_2\end{aligned}$$

The last nine equations are those that introduce measurements or known parameters. This notation makes it easy to introduce or remove sensors. Reservoirs introduce the value of one variable in two different moments as the quasi-static equations described. This is the way to know the reservoir inflow. The order used is not arbitrary as will be seen when a new time step is introduced. Levels of reservoirs are already present for this new time step, only heads and flows are introduced after these levels and the levels for the next time step before resistances and demand factors. So this time generalisation is the easiest.

Example 2.3: Here are shown the equations for the same network for two time steps.

$$x = [h_{r1}(1)\ \ h_{r2}(1)\ \ h_1(1)\ \ h_2(1)\ \ q_1(1)\ \ q_2(1)\ \ q_3(1)\ \ q_4(1)\ \ h_{r1}(2)\ \ h_{r2}(2)\ \ h_1(2)\ \ h_2(2)\ \ q_1(2)\ \ q_2(2)\ \ q_3(2)\ \ q_4(2)\ \ h_{r1}(3)\ \ h_{r1}(3)\ \ R_1\ \ R_2\ \ R_3\ \ R_4\ \ w_1\ \ w_2]$$

$$A_1(x_9 - x_1) - x_5 - x_6 = 0 \qquad A_1(x_{17} - x_9) - x_{13} - x_{14} = 0$$

$$A_2(x_{10} - x_2) + x_5 - x_7 = 0 \qquad A_2(x_{18} - x_{10}) + x_{13} - x_{15} = 0$$

$$x_6 - x_8 - x_{23} * d = 0 \qquad x_{14} - x_{16} - x_{23} * d = 0$$

$$x_7 + x_8 - x_{24} * d = 0 \qquad x_{15} + x_{16} - x_{24} * d = 0$$

$$x_1 - x_2 - x_{19} * x_5 * |x_5|^{0.852} = 0 \ ; \quad x_9 - x_{10} - x_{19} * x_{13} * |x_{13}|^{0.852} = 0 \ ; \qquad x_1 + x_{16} = 1$$

$$x_1 - x_3 - x_{20} * x_6 * |x_6|^{0.852} = 0 \quad x_9 - x_{11} - x_{20} * x_{14} * |x_{14}|^{0.852} = 0$$

$$x_2 - x_4 - x_{21} * x_7 * |x_7|^{0.852} = 0 \quad x_{10} - x_{12} - x_{21} * x_{15} * |x_{15}|^{0.852} = 0$$

$$x_3 - x_4 - x_{22} * x_8 * |x_8|^{0.852} = 0 \quad x_{11} - x_{12} - x_{22} * x_{16} * |x_{16}|^{0.852} = 0$$

$$x_1 = h_{r1}(1) \qquad x_9 = h_{r1}(2) \qquad x_{17} = h_{r1}(3)$$

$$x_2 = h_{r2}(1) \qquad x_{10} = h_{r2}(2) \qquad x_{18} = h_{r2}(3)$$

$$x_3 = h_1(1) \ ; \qquad x_{11} = h_1(2) \ ; \qquad x_{23} = w_1$$

$$x_5 = q_1(1) \qquad x_{13} = q_1(2)$$

$$x_8 = q_4(1) \qquad x_{16} = q_4(2) \qquad x_{24} = w_2$$

2.3. Identifiablity

When one system is being identified, input has to be selected in order to get a response of the system that will be measured during a period of time. When a water network has to be identified the measurements that will be taken appear as the outputs of the system. The conditions in which these measurements are taken if not forced, can be selected as the inputs. Finally the number of measurements taken by each meter is the extension in time of the identification process.

Looking at a water network at one time instant; the flows, heads, boundary flows, demands, demand factors and resistances are related by equations 2.2 and 2.3. The identification problem tries to determine parameters such as resistances and demand coefficients, which are constant (they change in a large period of time) and would provide the model of this network. Total demands can be measured or known a posteriori by invoicing. Boundary flows can be determined as well. These boundary flows come from a reservoir where head is easy to measure or are bought outside and the flow is measured. The measurement of heads in the nodes and flows in the pipes depend on the availability of meters or the opportunity to add them. As more measurements are available, namely **outputs**, the calibration will be possible or easier and more robust. The variables not measured; flows, heads, boundary flows, demands, demand factors and resistances have to be treated as parameters to be determined and a large number of such unmeasured parameters reduce the identifiability of the system.

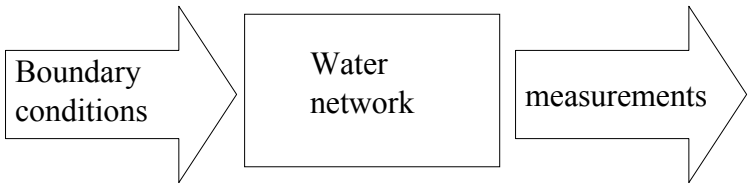
The identifiability study tries to determine the necessary meters that have to be installed in order to have a good calibration but not to increase the cost unnecessarily.

As the parameters of the network (resistances and demand factors) are constant in time they relate measurements taken in different time steps. That means that if the same measurements are taken more than once the number of unknown variables increases but not as the number of equations does. So a system that is not identifiable with just one time step can become identifiable if some time steps are included. As the **Time period** of the input signal and output measurement is determined in the identification of a system, so must in this case the number of time steps that make the system identifiable. If the minimum of time steps is determined and more than these are taken into account the optimisation will take advantage of this redundancy.

If many measurements are taken in the same conditions they will not add any information. Many measurements have in this case the aim of filtering the noise in the measurements. The selection of a good **input** is a vital step in the identification problem in order to get the information from that output generated by the system. In the identification of water network some variables can be altered or even fixed as heads in some nodes related to a reservoir or boundary flows. If this actuation is not possible the measurements can be taken in different conditions selecting the instants when total demands, level of reservoirs, etc. are different. A good selection of the input set is necessary to get a good set of outputs with the information for calibrating the model.

In the previous section the estimation problem has been presented as a set of equations. The identifiability of the system would be equivalent for the equation to have a unique solution. Such a condition implies that the number of equations is the same as the number of variables. All these variables have to appear in at least one equation as the unique unknown variable or with variables that can be determined by other equations. This study is done for the static problem [Oza-86],[Car-91] using graph analysis. For the linear case this condition reduces to the invertibility of the matrix of the equations set [Sor-80]. Water networks' models are in general non-linear, only in special cases where all heads or flows are measured it will become linear. The approximation by Taylor series drives to the use of jacobian to study the identifiability of the network.

The interest of using more than one time step in the measurement makes necessary the generalisation of the identifiability conditions. The identifiability study should determine; which are the variables that have to be measured, how many time steps must be taken into account, and which are the conditions in each time step that make all these measurements useful for the calibration (figure 2.4).



$$t(i) \in \{t(1)\ldots t(k)\}$$

Fig. 2. 4:Identification experiment

Conditions of identifiability for non-linear dynamic systems can be found in the literature [Wal-96]. If they have to be applied to water networks without using transient equations the system becomes a discrete one based on the quasi-static equations. State variables are the levels of reservoirs that are the variables that evolve in time, this leads to a difficult algebraic manipulation of differential equations that will make such an approach non operative. The topological approach that appears to be so useful in the static problem (for it allows the localisation of the missing sensors) has appeared to be difficult to generalise. Besides it would only assure the existence of equations from where the variable could be inferred but with no idea whether these equations are numerically well posed as no numerical value enters in its process. The generalisation of the Jacobian for the identifiability study appears to be the best approach. This equation approach converts the quasi-static problem to an **extended period static** problem as all the equations are treated as belonging to the same set with no time meaning.

If the formulation of the system is taken from an equation point of view, then all this discussion would be about the invertibility of a function in general and a matrix in the linear case. This approach has been used and has been crucial in the most general cases. Nevertheless the system approach, and with the concept of identifiability as a property, has been kept. Such an approach brings us to the more physical world. It is important not to forget the origin of the issues that we deal with when immense arrays are used and linear and non-linear functions are analysed, modified and inverted.

2.4. Static Identifiability

To study the static case will be insufficient for the purpose of this thesis but it is a good starting point. The Identifiability condition can be presented as both a necessary and sufficient condition.

1. The number of equations has to be greater than or equal to the number of variables.
2. Each variable has to appear in an equation where all the rest of the variables are measured or can be inferred from another equation.

The second condition is not obvious especially in non-linear systems. In this section two different approaches will be presented. They are applied to the same simple example.

2.4.1. Graph interpretation

Much literature exists on the study of observability for electric networks [Fet-76], [Bret-96]. A good idea is to exploit the analogy between these systems and water networks that are much less studied from a theoretical point of view. Carpentier [Car-91] uses the algorithm proposed by Ozawa [Oza-86] to determine the identifiability of a water network with some measurements provided. The idea in the background is that some operations in graphs (contracting, deleting) are equivalent to operations on equations (eliminating known variables, using equations to know variables). In studying water network problems, it is often convenient to treat the network as if it was a closed one. That is why the datum node is introduced so from the graph G the augmented graph G^* is generated. The head loss is used instead of heads and levels. The head loss equations (2.6) are added to those in equations 2.2 and 2.3.

$$\Gamma\ \Delta h = 0 \tag{2.6}$$

Example 2.4: Now the subsequent steps for the identifiability determination are explained using the network of figure 2.3 with the measurements of example 2.2, note that reservoirs in this case are treated as junctions and demand is equivalent to inflow which in this case is known. In figure 2.5 the augmented graph is presented.

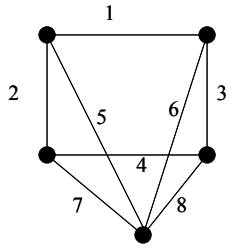


Fig. 2. 5

1. *First of all the pipe subsets are represented explicitly*
   $U_b^* = \{1,5,6,7\}$ *elements with two or more parameters known (demand and head loss)*
   $U_\pi^* = \{2\}$ *elements of which only head difference is known*
   $U_f^* = \{4,8\}$ *elements of which only flow is known (demand)*
   $U_r^* = \{\varnothing\}$ *elements of which only resistance is known (flow is unknown)*
   $U_u^* = \{3\}$ *elements for which nothing is known*
2. *The flow graph is defined as:* $G_{fr}^* = G^* \cdot (U_u^* \cup U_\pi^* \cup U_r^*) \times U_r^*$, *it contains no arc. Where* $G^* \cdot U_r^*$ *means the graph obtained for* G^* *by deleting all the arcs not in* U_r^*. $G^* \times U_r^*$ *means the graph obtained from* G^* *by contracting all the arcs not in* U_r^*. *Tree and cotree of the flow graph are* $T_{fr}^* = \{\varnothing\}; C_{fr}^* = \{\varnothing\}$ *that means that there are no unknown flow and no flow equation left after the operation. A tree of a graph G, T, is a maximal partial graph of G which does not contain any cycle (loop). The cotree,*

C_T, associated with T is the partial graph of G incorporating all the arcs of G-T and has no cocicle (cut).

3. *The head graph is defined as: $G^*_{\pi r} = G^* \times (U^*_u \cup U^*_f \cup U^*_r) \cdot U^*_r$, is empty as well. Tree and cotree of the head graph are $T^*_{\pi r} = \{\varnothing\}; C^*_{\pi r} = \{\varnothing\}$ with the same meaning about variables and equations.*
4. *The pipes in which only the resistance is known can be divided in three subsets:*
   - *U^*_{r+} minimal subset of U^*_r that $U^*_{r+} \supseteq T^*_{fr} \cap C^*_{\pi r}$, $U^*_{r+} \cap T^*_{fr}$ is tree of $G^*_{fr} \times U^*_{r+}$ and $U^*_{r+} \cap C^*_{\pi r}$ is cotree of $G^*_{\pi r} \cdot U^*_{r+}$. In this case $U^*_{r+} = \{\varnothing\}$.*
   - *U^*_{r-} minimal subset of U^*_r that $U^*_{r-} \supseteq C^*_{fr} \cap T^*_{\pi r}$, $U^*_{r-} \cap C^*_{fr}$ is tree of $G^*_{fr} \cdot U^*_{r-}$ and $U^*_{r-} \cap T^*_{fr}$ is cotree of $G^*_{\pi r} \times U^*_{r-}$. In this case $U^*_{r-} = \{\varnothing\}$.*
   - $U^*_{r0} = U^*_r - U^*_{r+} - U^*_{r-} = \{\varnothing\}$
5. *The tree of G^* that allows us to write flow equations must be determined from its definition $T^*_f = T^*_{fbf} \cup T^*_{fr} \cup T^*_{f\pi u} = \{1,2,3,5\}$ where:*

   $T^*_{fbf} = \{1,5\}$ is a tree of $G^* \times (U^*_b \cup U^*_f)$

   $T^*_{f\pi u} = \{2,3\}$ is a tree of $G^* \cdot (U^*_\pi \cup U^*_u)$

   $T^*_{fr} = T^*_{fr+} \cup T^*_{fr0} \cup T^*_{fr-} = \{\varnothing\}$
6. *The tree for the head equations $T^*_\pi = T^*_{\pi b\pi} \cup T^*_{\pi r+} \cup T^*_{\pi r0} \cup T^*_{\pi r-} \cup T^*_{\pi fu} = \{2,3,4\}$ where:*

   $T^*_{\pi b\pi} = \{1,3,5,8\}$ is a tree of $G^* \cdot (U^*_b \cup U^*_\pi)$

   $T^*_{\pi fu} = \{3\}$ is a tree of $G^* \times (U^*_f \cup U^*_u)$

   $T^*_{\pi r+} = T^*_{\pi r} \cap U^*_{r+} = \{\varnothing\}$ is a tree of $G^*_{\pi r} \cdot U^*_{r+}$

   $T^*_{\pi r-} = T^*_{\pi r} \cap U^*_{r-} = \{\varnothing\}$ is a tree of $G^*_{\pi r} \times U^*_{r-}$

   $T^*_{\pi r0} = T^*_{\pi r} \cap U^*_{r0} = \{\varnothing\}$ is a tree of $G^*_{\pi r} \cdot (U^*_{r+} \cup U^*_{r-}) \times U^*_{r0}$
7. *Redundancy in flow equations comes from the equations generated by arcs in $T^*_{fbf} \cup T^*_{fr+} = \{1,5\}$, which involve only variables of $U^*_b \cup U^*_{r+} \cup U^*_f$. Only two of the four flow equation (linear independent) are used to determine q_2 and q_3, the other two are redundant*
8. *Redundancy in head equations comes from the equations generated by arcs in $C^*_{\pi b\pi} \cup C^*_{\pi r+} = \{6,8\}$, which involve only variables of $U^*_b \cup U^*_{r+} \cup U^*_\pi$. Equations $\Delta h_1 + \Delta h_6 + \Delta h_5 = 0$ and $\Delta h_2 + \Delta h_7 + \Delta h_5 = 0$ are redundant.*
9. *$C^*_{fr0} = \{\varnothing\}$ and $T^*_{fr0} = \{\varnothing\}$ generate the equations which involve calculable variables. In this case this set is empty.*
10. *The flow equations generated by the arcs in $T^*_{fr-} = \{\varnothing\}$ and the head difference equations generated by the arcs in $C^*_{fr-} = \{\varnothing\}$ cannot be solved without adding new measurements. This part of the global problem remains unidentifiable. So in this case, flow in 2 and 3 are unidentifiable.*

11. *The flow of an arc in* $U_\pi^* \cup U_u^* = \{3,2\}$ *can be obtained if a cocycle, generated by the arc such that all the other arcs in the cocycle belong to* $U_b^* \cup U_{r+}^* \cup U_f^* \cup U_{r0}^* = \{1,4,5,6,7,8\}$*, exists. In this case the cocycles, generated by 2 and 3,* $\{1,2,4,5\}$ $\{1,3,4,6\}$ *fulfil the condition and so flow is identifiable.*
12. *The head difference of an arc in* $U_f^* \cup U_u^* = \{3,4,8\}$ *can be obtained if a cycle, generated by this arc such that all the other arcs in the cycle belong to* $U_b^* \cup U_{r+}^* \cup U_\pi^* \cup U_{r0}^* = \{1,2,5,6,7\}$*, exists. In this case the cycles generated don't fulfil the condition so head differences are unidentifiable.*

A resistance variable of an arc in $U_\pi^* \cup U_f^* \cup U_u^* = \{2,3,4,8\}$ can finally be obtained if the flow and the head difference of this arc were previously obtained. In this case only 2 can be obtained. Estimation in these conditions has given the wrong results shown in figure 2.7. The unidentifiability of the system doesn't come from condition 4 -that has to do with pipes where roughness is known. As the roughness is constant some more measurements taken in other time steps can improve this identifiability.

2.4.2. Equation interpretation

In some cases, when all heads are known, equations of the model become linear. Equation 2.4 can be written in matrix form, equation 2.7. The identifiability condition can be expressed in terms of the rank of matrix G as it is equivalent to the solvability of the equations set. Linear algebra imposes that a condition (the number of variables and rank of G are equal) has to be necessary and sufficient.

$$z = G^* x + \xi \tag{2.7}$$

Kalman filter allows the estimation of parameter in a system from the inputs and outputs measured. Such a filter is robust to the measurement noise. Kalman filter gives a estimation of these parameters (ψ) generated as the arguments of a minimisation (equation 2.8).

$$\psi_i = \arg\min_{r \in \Re^n} \sum_{j=1}^{i} \left\| z_j - G_j x_r \right\|^2 \qquad \text{i} = 1,....,\text{t} \tag{2.8}$$

Where x_r are the parameters that are searched. Assuming that matrix G_1'$\cdot$ G_1 is positive defined (this is a identifiability condition is equivalent to the analysis of rank of G_1 that has already stated), the least squares estimate can be generated by a recursive algorithm.

$$\begin{aligned}
&\psi_i = \psi_{i-1} + H_i^{-1} G_i'(y_i - G_i \psi_{i-1}) \qquad \text{i} = 1,....,\text{t} \\
&\text{H}_\text{i} = \lambda H_{i-1} + G_i' G_i \qquad \text{i} = 1,....,\text{t} \\
&\text{H}_0 = 0 \\
&0 < \lambda < 1
\end{aligned} \tag{2.9}$$

where λ is a exponential forgetting factor. When λ=1 the usual least squares problem is obtained. As the system is non-linear a extended Kalman filter (EKF) has to be introduced [Ver-97]. Here extended has nothing to do with the extended period that will be used later in this thesis but with the generalisation for non-linear systems. This method may be viewed as an incremental version of the Gauss-Newton-like iteration. The method starts with some x_{r0}, then updates x_r via a Gauss-Newton –Like iteration aimed at minimising

$$\left\| f(x_r) \right\|^2 = \left\| z - g(x_r) \right\|^2 \tag{2.10}$$

then updates x_r via a Gauss-Newton-like iteration aimed at minimising and similarly continues, with the *i*th step consisting of a Gauss-Newton-like iteration aimed at minimising the partial sum

$$\sum_{j=1}^{i} \left\| f_j(x_r) \right\|^2$$

Thus, a cycle through the data set of EKF sequentially generates the vectors

$$\psi_i = \arg \min_{r \in \Re^n} \sum_{j=1}^{i} \left\| \tilde{f}_j(x_r, \psi_{i-1}) \right\|^2$$

Where $\tilde{f}_j(r, \psi_{i-1})$ are the linearised functions

$$\tilde{f}_j(x_r, \psi_{i-1}) = f_j(\psi_{i-1}) + \nabla f_j(\psi_{i-1})'(x_r - \psi_{i-1})$$

and $\psi_0 = x_{r0}$ is the initial estimate of x_r. Using the formulas of Kalman Filter (2.9) with the identifications

$$\begin{aligned}
&z_i = f_j(\psi_{i-1}) - \nabla f_j(\psi_{i-1})' \psi_{i-1} \\
&G_i = -\nabla f_j(\psi_{i-1})'
\end{aligned} \tag{2.11}$$

The first identification is equivalent to work with increments what is normal when the linearisation is used. The algorithm can be rewritten as

$$\begin{aligned}
&\psi_i = \psi_{i-1} + H_i^{-1}\nabla f_i' f_i \qquad i = 1,....,t \\
&H_i = \lambda H_{i-1} + \nabla f_i \nabla f_i' \qquad i = 1,....,t \\
&H_0 = 0
\end{aligned} \tag{2.12}$$

The convergence condition is following Bertsekas reasoning [Ver-97] that $\nabla f_1(r)\ \nabla f_1(r)'$ has rank equal to the number of parameters that are estimated. In the water system calibration problem $f_1(y,x,r)=z-g_1(x,r)$ is the vectorial equation system of the network. The Jacobian that is searched would give an idea of the relation of outputs (measurements, z) and parameters (r) independently of the state variables that are not measured but are not intended to be estimated. Differentiating $f_1(z,x,r)$

$$\begin{aligned}
&\frac{\partial(f_1(z,x_t,x_r))}{\partial z}dz + \frac{\partial(f_1(z,x_t,x_r))}{\partial x_t}dx_t + \frac{\partial(f_1(z,x_t,x_r))}{\partial x_r}dx_r = 0 \\
&I\nabla_{x_r} z + J_{x_t}\nabla_{x_r} x_t + J_{x_r} = 0 \\
&\begin{bmatrix} I & J_{x_t} \end{bmatrix}\begin{bmatrix} \nabla_{x_r} z \\ \nabla_{x_r} x_t \end{bmatrix} = -J_{x_r} \\
&\begin{bmatrix} \nabla_{x_r} z \\ \nabla_{x_r} x_t \end{bmatrix} = -\left[\begin{bmatrix} I & J_{x_t} \end{bmatrix}^T \begin{bmatrix} I & J_{x_t} \end{bmatrix}\right]^{-1}\begin{bmatrix} I & J_{x_t} \end{bmatrix}^T J_{x_r}
\end{aligned} \tag{2.13}$$

This development is based on the implicit function theorem:

IMPLICIT FUNCTION THEOREM Let the function $G:\mathbb{R}^{n+m}\rightarrow\mathbb{R}^m$ be C^1 in the neighbourhood Ω of the vector $(\mathbf{x}_0, \mathbf{z}_0)$ in $\mathbb{R}^{n+m}$ with $G(\mathbf{x}_0, \mathbf{z}_0)=0$. If $|J_G(\mathbf{z}_0)|\neq 0$, then there exists a neighbourhood Ω_1 of $\mathbf{x}_0$ in $\mathbb{R}^n$, a neighbourhood of $(\mathbf{x}_0,\mathbf{z}_0)$ in $\mathbb{R}^{n+m}$, and a function $\mathbf{g}:\mathbb{R}^n\rightarrow\mathbb{R}^m$ in $C^1(\Omega_1)$ such that $\mathbf{z}=\mathbf{g}(\mathbf{x})$ satisfies $\mathbf{G}(\mathbf{x},\mathbf{g}(\mathbf{x}))=0$ in Ω_2. Furthermore, $J_g(\mathbf{x})=-[J_G(\mathbf{y})]^{-1}J_G(\mathbf{x})$

The idea is to solve the first part of equations in last system of equation 2.13 in order to obtain the relation between measurements and parameters ($\nabla_{xr}z$). To do it analytically is an extremely difficult problem because the intermediate variables x_t, that are not measured have to be eliminated from equations that are non-linear. It is done numerically for a working point. Using the first notation presented in example 2.1 (without ordering the variables in a convenient way) equations can not allow an explicit expression without losing generality. It would make necessary to redo all the calculations depending on the measurements available. The second form of the equations presented in example 2.2 for general purposes is useful when the Jacobian has to be calculated. It means to assume that all variables have to be estimated and include new equations for measurements such as 2.14 for those that are measured or known

$$x_i = z_i \tag{2.14}$$

These new equations will generate rows in the Jacobian with one number 1 each in the position of the measurement (reservoirs, nodes and pipes in this order). They are represented in equation 2.15 by **B**. Vector *x* includes all the variables, measured and estimated: reservoirs and junctions' head, pipes, valves and pumps' flow, pipes' roughness and junctions' demand factors in this order. Equation 2.15 presents the form of a generic Jacobian where variables (x) are both state(x_t) and parameters(x_r).

$$J = \begin{bmatrix} A*I(r,r) & zeros(r,n) & & -A*I(r,r) & zeros(r+n,l) & -d\begin{bmatrix} zeros(r,n) \\ I(n,n) \end{bmatrix} \\ zeros(n,r) & zeros(n,n) & \Lambda & zeros(n,r) & & \\ -\Lambda^T & & -1.852R_n \begin{bmatrix} q_1^{0.852} & 0 & 0 \\ ... & ... & ... \\ 0 & 0 & q_l^{0.852} \end{bmatrix} & zeros(l,r) & -\begin{bmatrix} q_1^{1.852} & 0 & 0 \\ ... & ... & ... \\ 0 & 0 & q_l^{1.852} \end{bmatrix} & zeros(l,n) \\ zeros(1,r+n) & & zeros(1,l) & zeros(1,r) & zeros(1,l) & ones(1,n) \\ B(z,r+n) & & B(z,l) & B(z,r) & B(z,l) & B(z,n) \end{bmatrix} \tag{2.15}$$

Where z is the number of measurements. B is a sub-matrix with z rows and r+n+l all zeros except single 1 in each row in the position of a measurement. The identifiability condition is that the rank of J is equal to the number of variables, 2*r+2*n+2*l (heads, flows, roughness and demand factors). This Jacobian will be used later for the identification problem design because the relation between measurements and state variables is important for the meter placement. Following equation 2.13 and in order to have a compact matrix to determine the identifiablility of the network a new matrix will be generated numerically:

$$\begin{bmatrix} I & J_{x_t} \end{bmatrix} = \begin{bmatrix} & A*I(r,r) & zeros(r,n) & & -A*I(r,r) \\ & zeros(n,r) & zeros(n,n) & \Lambda & zeros(n,r) \\ zeros(r+n+l+1) & -\Lambda^T & & -1.852R_n \begin{bmatrix} q_1^{0.852} & 0 & 0 \\ ... & ... & ... \\ 0 & 0 & q_l^{0.852} \end{bmatrix} & zeros(l,r) \\ I(l+n,l+n) & zeros(1,l) & & zeros(1,l) & zeros(1,r) \\ & B(z,r+n) & & B(z,l) & B(z,r) \end{bmatrix} \tag{2.16}$$

This is a (r+2n+2l+1×2r+2n+2l) matrix, but it is used to obtain a square matrix with the dimensions of the number of parameters (l+n×l+n) $\nabla_{xr}z$ $\nabla_{xr}z'$ solving 2.13 partially. This has to have rank l+n in order to assure the observability as is stated previously [Bert-96].

Example 2.5: at the same system of example 2.4. Variables are 16 –two levels for each reservoir (4); head and demand factor for each junction (4); flow and roughness for each

pipe (8)-. The system is not identifiable as had been stated by the previous method. The first two rows of Jacobian correspond to the mass balance in the reservoir, the next two rows to the mass balance of the junctions. The next four rows correspond to the hydraulic equations by Hazen-Williams in each pipe. The next eight rows correspond to the measurements -four levels and four flows-. The last row corresponds to the relation between demand factors.

$$J = \begin{bmatrix}
19.6 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -19.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 19.6 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & -19.6 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
1 & -1 & 0 & 0 & -24.6 & 0 & 0 & 0 & 0 & 0 & -1.7 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & -12.2 & 0 & 0 & 0 & 0 & 0 & -1.7 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 3.8 & 0 & 0 & 0 & 0 & 0 & -1.4 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & -14.8 & 0 & 0 & 0 & 0 & 0 & -1.7 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

The dimension of the Jacobian is 18x16 but its rank is of 15. So the system is not identifiable, as the number of variables is not equal to the rank of the Jacobian. The calculation of $\nabla_{xr}z\ \nabla_{xr}z'$ is done numerically. The result is a 6×6 matrix:

$$\nabla_{x_r} z' \nabla_{x_r} z = \begin{bmatrix}
-0.001 & -0.001 & -0.001 & -0.001 & -0.007 & -0.017 \\
-0.001 & 0.087 & 0.041 & 0.056 & 0.478 & 1.203 \\
-0.001 & 0.041 & 0.026 & 0.034 & 0.290 & 0.738 \\
-0.001 & 0.056 & 0.034 & 0.058 & 0.394 & 1.01 \\
-0.007 & 0.478 & 0.290 & 0.394 & 4.409 & 8.542 \\
-0.017 & 1.203 & 0.738 & 1.014 & 8.542 & 22.864
\end{bmatrix}$$

The rank of this matrix is 5 so it is smaller than the number of parameters that are intended to be estimated. In figures 2.6 and 2.7 appears the estimation of the parameters. Of course the measurements are well estimated and so are the flows (it was seen in previous methodology that the problem was in head estimation). The flow sub-problem may be identifiable but the heads at the junctions and the resistaces are not well estimated. The (·) represents the estimated values and (o) the actual ones.

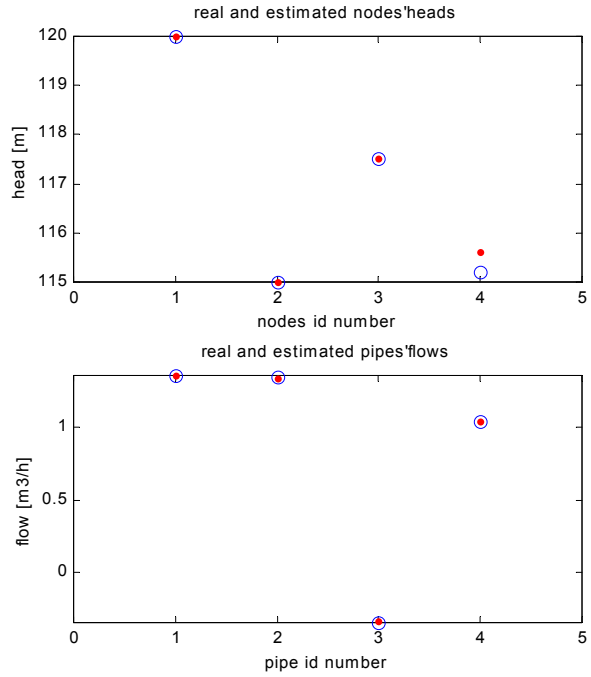

real and estimated nodes'heads
head [m]
nodes id number
real and estimated pipes'flows
flow [m3/h]
pipe id number

Fig. 2. 6

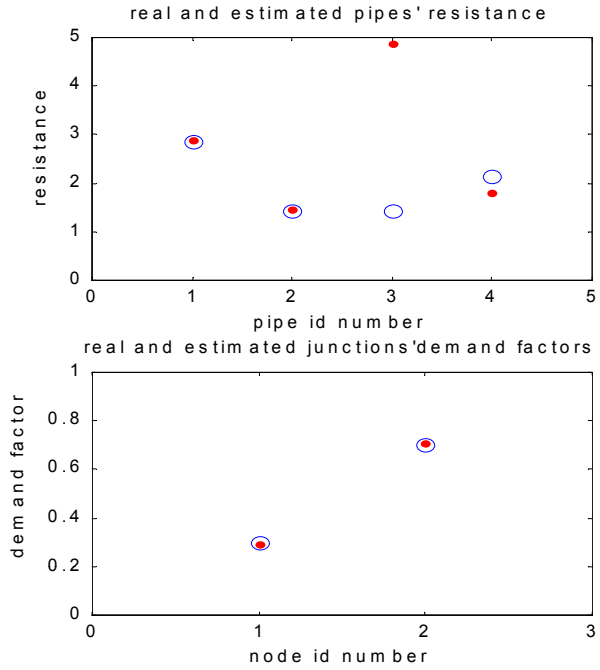

real and estimated pipes' resistance
resistance
pipe id number
real and estimated junctions'demand factors
demand factor
node id number

Fig. 2. 7

2.5. Dynamic Identifiability

When the identifiability is studied for a calibration problem some constant parameters are estimated. It means that the demand factors or the resistances of the pipes do not change in each sample of measurements. So if more than one time step is taken into account some more information is provided. Perhaps a system that is not identifiable just with one time step becomes identifiable with more time steps, as the number of equations introduced may be greater than the new variables to be estimated.

Walter [Wal-96] uses the state space formulation in order to study the identifiability of a system using the dynamic information. Consider the model

$$M(p):\begin{cases}\dfrac{d}{dt}x(t) = f(x(t),u(t),t,p), & x(0) = x_0(p)\\ y(t,p) = h(x(t),p).\end{cases} \tag{2.17}$$

If

$$a_k(p) = \lim_{t\to 0+}\frac{d}{dt}y(t,p) \tag{2.18}$$

then a suffient condition for M(·) to be uniquely identifiable is

$$a_k(\hat{p}) = a_k(p^*), \qquad k = 0,1,\ldots k_{\max} \Rightarrow \hat{p} = p^* \tag{2.19}$$

There k_{max} is a positive integer, small enough for the computations to remain tractable.

The model that is being calibrated is quasi-static. It means that in a time step heads are considered constant and flows in steady state. Dynamics are introduced by the reservoirs that increase or decrease their level and a new steady state is calculated. The methodology proposed by Walter is adapted for the study of extended-period identifiability of a network.

Example 2.6: A simple case, figure 2.8, that is not observable using only one time-step measurement is posed in state space and the Taylor approach is used to prove it is identifiable as a dynamic system. Using optimisation its calibration has been successful.

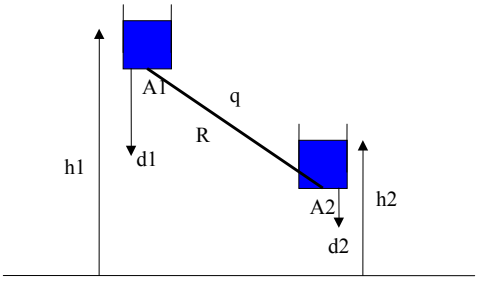


Fig. 2. 8

Equations of this dynamic not linear system are

$$A1\frac{dh1}{dt} = -q - d1$$

$$A2\frac{dh2}{dt} = q - d2$$

$$h1 - h2 = Rq^{1.852}$$

$$y = h2$$

The state variables are the level h1 and h2, measurement is h2 and the unknown parameter is R. Demands are the inputs of the system. Output is head of the second reservoir. Using Ozawa's algorithm, it has been proved that knowing demands, h2 and ignoring flow q, resistance and h1 the model is not observable in one time step measurement.

The flow q appears in the static equation so cannot be treated as a state variable, it is substituted in the dynamic equation and the problem appears as follows.

$$A1\frac{dx1}{dt} = -Y^{0.54}(x1 - x2)^{0.54} - u1$$

$$A2\frac{dx2}{dt} = Y^{0.54}(x1 - x2)^{0.54} - u2$$

$$y = x2$$

Here Y is the admittance and u represents the inputs, demands. An initial condition is necessary in order to examine the identifiability following the approach presented. The head at the initial moment is x(0)=[110 100]. The equations with k_{max}=2 show that the dynamic system is identifiable

$$a_1 = 100$$

$$a_2 = \lim_{t \to 0} \frac{dx2}{dt} = \frac{1}{A2} Y^{0.54} (x1(0) - x2(0))^{0.54}$$

from the second one it can be seen that only one solution for Y is possible so parameter R=1/Y can be found. Using the *constr.m* function of Matlab the optimisation problem converged and the calibration was succesful. A problem appears in the case of nodes which cannot be treated as dynamic elements and have to be substituted, as was done with flows.

Example 2.7: Network in figure 2.9 includes a node that presents problems for the formulation as has already been told. Parameters to be identified are R1 and R2. A1 and A2 are known, demands d1 and d2 are treated as inputs and the measurements are done in h1 and h2.

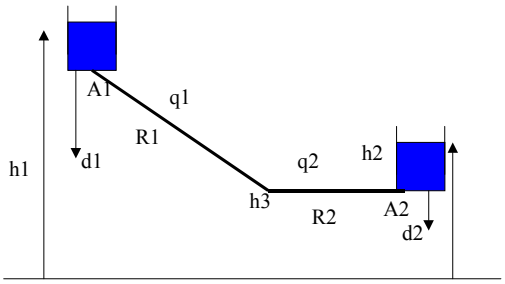


Fig. 2. 9

$$A1 \frac{dh_1}{dt} = -q_1 - d1$$

$$A2 \frac{dh_2}{dt} = q_2 - d2$$

$$h1 - h3 = R_1 q_1^{1.852}$$

$$h3 - h2 = R_2 q_2^{1.852}$$

$$q_1 = q_2$$

$$y = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

These equations can be treated in order to obtain a new formulation useful for the identifiability study.

$$A1\frac{dx_1}{dt} = -\left(\frac{Y_1}{1+Y_1R_2}\right)^{0.54}(x1-x2)^{0.54} - u1$$

$$A2\frac{dx_2}{dt} = \left(\frac{Y_1}{1+Y_1R_2}\right)^{0.54}(x1-x2)^{0.54} - u2$$

$$y = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where $x_1=h_1$, $x_2=h_2$, $Y_1=1/R_1$, $u_1=d_1$, $u_2=d_2$.

Using Ozawa's approach the network is not identifiable with just one measurement in h_1 and h_2. In order to study the identifiability, the approach proposed by Walter is used. The initial conditions needed are:

$$x(0) = \begin{bmatrix} 110 \\ 100 \end{bmatrix}$$

The equations with $k_{max}=1$ show that the dynamic system is not identifiable

$$a_0 = \begin{bmatrix} 110 \\ 100 \end{bmatrix}$$

$$a_1 = \begin{bmatrix} \lim_{t\to 0}\frac{dx_1}{dt} \\ \lim_{t\to 0}\frac{dx_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{A1}\left(\frac{Y_1}{1+Y_1R_2}\right)^{0.54}(x1(0)-x2(0))^{0.54} \\ \frac{1}{A2}\left(\frac{Y_1}{1+Y_1R_2}\right)^{0.54}(x1(0)-x2(0))^{0.54} \end{bmatrix} = \begin{bmatrix} -\left(\frac{Y_1}{1+Y_1R_2}\right)^{0.54}0.0347 \\ \left(\frac{Y_1}{1+Y_1R_2}\right)^{0.54}0.0347 \end{bmatrix}$$

Both parameters always appear together in the same expression so only this combination of the parameters could be identified taking more time steps. However this has been tried using optimisation and the identification failed as was expected. The network was easily observed introducing the measurement of h_3. The identifiability has not been studied using this approach, as it is not obvious to introduce the dynamic of a node without reservoir.

Example 2.8: A modification to the previous system is done and the study though difficult to work out seems to indicate that identifiability improves. The identification of both resistances, when only a node without demand and no head measure separates the pipes, it showed to be impossible even increasing the number of measurements. This identification could be even useless, as an equivalent resistance could be enough for both pipes together. A demand at the node makes the identification necessary but it may improve the identifiability as well. Figure 2.10 shows the new network.

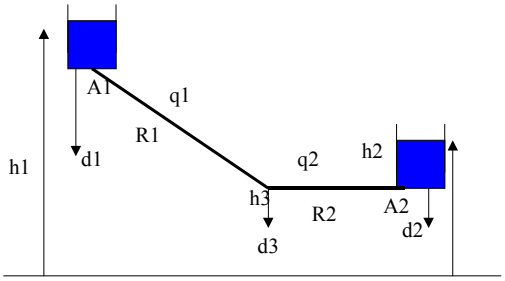


Fig. 2. 10

$$A1\frac{dh_1}{dt} = -q_1 - d1$$

$$A2\frac{dh_2}{dt} = q_2 - d2$$

$$h1 - h3 = R_1 q_1^{1.852}$$

$$h3 - h2 = R_2 q_2^{1.852}$$

$$q_1 = q_2 + d_3$$

$$y = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

These equations can be treated in order to obtain a new formulation useful for the identifiability study. The isolation of q1 gives a very complicated result due to the exponent 1.852. It has been solved using Maple [Map-03] but in order to simplify the equations this exponent is approximed to 2.

$$A1\frac{dx_1}{dt} = -\frac{2R_2u3 \pm \sqrt{(2R_2u3)^2 - 4(R_1+R_2)(u3^2 + x_2 - x_1)}}{2(R_1+R_2)} - u1$$

$$A2\frac{dx_2}{dt} = \frac{2R_2u_3 \pm \sqrt{(2R_2u_3)^2 - 4(R_1+R_2)(u_3^{\,2} + x_2 - x_1)}}{2(R_1+R_2)} - u2 - u3$$

$$y = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where $x_1=h_1$, $x_2=h_2$, $Y_1=1/R_1$, $u_1=d_1$, $u_2=d_2$, $u_3=d_3$.

Using Ozawa's approach the network is not identifiable with just one measurement in h_1 and h_2. In order to study the identifiability the approach proposed by Walter should be used. The inclusion of the third demand in a non-linear relationship between the resistances intuitively means that in order to obtain a_0, a_1,...,a_{kmax} only a set of parameters work, if it works with exponent 2 it can be assumed that with 1.852 it would.

This has been tried using optimisation and the identification worked while the same failed if demand 3 was zero. The expressions of a_0, a_1,...,a_{kmax} contain derivatives of the output in respect to the time. Expressions are too complicated to do it without software (Mapple, Mathematica). It has been done but even for such simple example the solution is a multiple and very complicates expression.

The complexity of these studies even for such small networks makes difficult their use for real networks. The study of the extended-period identifiability should use easier expressions that can be generalised easily. The linear algebra theories seem to fulfil this condition. Even for non-linear models the approach is similar to the methodology proposed by Walter as both use Taylor series approximation.

2.6. Extended Period Identifiability

In state estimation problems the identifiability condition doesn't change when new time steps are added, unless the system is treated as a dynamic one. Transient equations appear in such cases and they are much more complicated, especially for huge networks. Using the quasi-static approach allows us to use simpler equations related from one time step to the next one by reservoir equations. Each time step is equivalent to the former and introduces the same number of variables so the identifiability can be decided from one time step in state estimation. If parameters are estimated the number of variables increases differently from the number of equations as these parameters are constants. This is the main reason to introduce the **extended period identifiability** concept, not called dynamic as it is based on quasi-static equations.

Here once more it should be recalled that the meaning of extended is different to that in Extended Kalman Filter (EKF) where it concerns be with non-linearity. Here extended is a time concept and at the end of this sections both extended meanings will meet when extended-period non-linear systems are treated.

The introduction of new equations (repeating the measurements in different times) can be useful but the conditions cannot be the same. In figure 2.4 the three important elements of identification experiments appear. So far the output (measurements) has been treated. Adding new time steps the time horizon is introduced in this problem. The inputs have to change during this horizon otherwise outputs wouldn't carry more information about the system. Inputs must be as orthogonal as possible in order to contain as much information as possible for the identification process. These **orthogonal inputs** have to be obtained in water networks working in changing conditions of demands or reservoir levels.

The problem in extended period identifiability is no longer just counting equations but to assure they are not linearly dependent. The generalisation of topological approach

would not help in this new challenge. So the generalisation of Jacobian methodology is presented after some results for the linear particular case.

2.6.1. Linear case

If all heads are measured this problem becomes linear, so the l non-linear equations become linear. Using a nodal model, equations 2.2 can be written as:

$$\Lambda \cdot Q(\Delta h) + w \cdot d = 0 \tag{2.20}$$

where Q(Δh) is a vector of pipe equations

$$Q(\Delta h) = \begin{bmatrix} y_1 \mid \Delta h_1 \mid^{0.54} sign(\Delta h_1) \\ ... \\ y_l \mid \Delta h_l \mid^{0.54} sign(\Delta h_l) \end{bmatrix} \tag{2.21}$$

where y_i is the admittance of pipe i. ΔH can be defined as

$$\Delta H = \begin{bmatrix} \mid \Delta h_1 \mid^{0.54} sign(\Delta h_1) & 0 & ... & 0 \\ 0 & \mid \Delta h_2 \mid^{0.54} sign(\Delta h_2) & ... & ... \\ ... & 0 & ... & 0 \\ 0 & ... & 0 & \mid \Delta h_l \mid^{0.54} sign(\Delta h_l) \end{bmatrix} \tag{2.22}$$

So that equation 2.20 can be written as

$$\begin{bmatrix} \Lambda \ \Delta H & d \end{bmatrix} \begin{bmatrix} zeros(r,n) \\ I(n) \end{bmatrix} \begin{bmatrix} y \\ w \end{bmatrix} = 0 \tag{2.23}$$

where **I** is a unit matrix and y and w are the parameters to be estimated. Equations are aggregated in equation 2.23 for each time step and it becomes like equation 2.24.

$$\begin{bmatrix} \Lambda \ \Delta H(1), & d(1) & zeros(r,n) \\ & & I(n) \\ \Lambda \ \Delta H(2), & d(2) & zeros(r,n) \\ & & I(n) \\ & ... & \\ \Lambda \ \Delta H(K), & d(K) & zeros(r,n) \\ & & I(n) \end{bmatrix} \begin{bmatrix} y \\ w \end{bmatrix} = 0 \tag{2.24}$$

The matrix

$$H(K)=\begin{bmatrix} \Lambda\,\Delta H(1), & d(1) & \begin{matrix} zeros(r,n) \\ I(n) \end{matrix} \\ \Lambda\,\Delta H(2), & d(2) & \begin{matrix} zeros(r,n) \\ I(n) \end{matrix} \\ & \cdots & \\ \Lambda\,\Delta H(K), & d(K) & \begin{matrix} zeros(r,n) \\ I(n) \end{matrix} \end{bmatrix} \tag{2.25}$$

has *n*K+1* rows, when relation between all the demand factors is added, and *(l+n)* columns where *l* denotes the number of branches in the network. The model is identifiable if matrix H(K) has a full rank of (*l+n*).
There is a theorem, which may make the problem of extended-period identifiability easier:

Theorem 1 [Sor-80]
The matrix H(K) has a full rank if and only if the matrix $H(K)^T$ H(K) is positive definite. Applying this theorem to our matrix H(K), given by 2.25, the following is obtained

$$(H(K))^T H(K)=\begin{bmatrix} \sum_{k=1}^{K}\Delta H(k)\,\Lambda^T\Lambda\,\Delta H(k) & \sum_{k=1}^{K} d(k)\Delta H(k)\,\Lambda^T \\ \sum_{k=1}^{K} d(k)\,\Lambda\,\Delta H(k) & \sum_{k=1}^{K} d^2(k)\,\mathrm{I} \end{bmatrix} \tag{2.26}$$

This is a (*l+n*)×(*l+n*) matrix and its size does not depend on the number of time steps K. It is easier to investigate matrix 2.26 than matrix 2.25. This matrix may not be positive-defined for K=1 but if we take more time steps K>1 then the matrix may become positive-defined. In order to get information from each time step the measurements must be independent.

Example 2.9: To exemplify the identifiability studies and in order to get some conclusions the same square network is used. Now all heads (reservoir levels included) are known and so are the demands. No flow is measured and resistances have to be estimated as well. That means that from reservoir equations the inflow can be obtained. The values of variables are the following:

$$hr = \begin{bmatrix} 120 & 115 \\ 119.9 & 115.1 \\ 119.7 & 115.2 \end{bmatrix}; h = \begin{bmatrix} 117.5 & 115.2 \\ 117.3 & 115.1 \end{bmatrix}; q = \begin{bmatrix} 1.35 & 1.34 & -0.34 & 1.04 \\ 1.32 & 1.37 & -0.17 & 1.01 \end{bmatrix}; d = \begin{bmatrix} 1 \\ 1.2 \end{bmatrix}$$

$$A1 = \begin{bmatrix} 2.4 & 1.6 & 0 & 0 & 0 & 0 \\ -2.4 & 0 & -0.4 & 0 & 0 & 0 \\ 0 & -1.6 & 0 & 1.6 & 1 & 0 \\ 0 & 0 & 0.4 & -1.6 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad \lambda 1 = \begin{bmatrix} 0 \\ 0 \\ 1.5 \\ 3.6 \\ 7.4 \\ 13.6 \end{bmatrix}$$

$$A2 = \begin{bmatrix} 2.4 & 1.6 & 0 & 0 & 0 & 0 \\ -2.4 & 0 & -0.4 & 0 & 0 & 0 \\ 0 & -1.6 & 0 & 1.6 & 1 & 0 \\ 0 & 0 & 0.4 & -1.6 & 0 & 1 \\ 2.3 & 1.7 & 0 & 0 & 0 & 0 \\ -2.3 & 0 & -0.2 & 0 & 0 & 0 \\ 0 & -1.7 & 0 & 1.5 & 1.2 & 0 \\ 0 & 0 & 0.2 & -1.5 & 0 & 1.2 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \lambda 2 = \begin{bmatrix} 0.05 \\ 0.01 \\ 2.06 \\ 6.2 \\ 14.8 \\ 26.8 \end{bmatrix}$$

The eigenvalues of A^TA are λ. A real symmetric matrix is positive defined iff all its eigenvalues are positive. A^TA is easily proven to be real and symmetric. Minimum eigenvalue for matrix $A1^TA1$ is $\lambda 1=0$ when only one time step is used so the system is not identifiable and the results of trying to estimate the parameters are shown in figure 2.11. Here only the constant parameters are presented. Demand factors can be estimated easily and the numerical problems affect the resistances estimation.

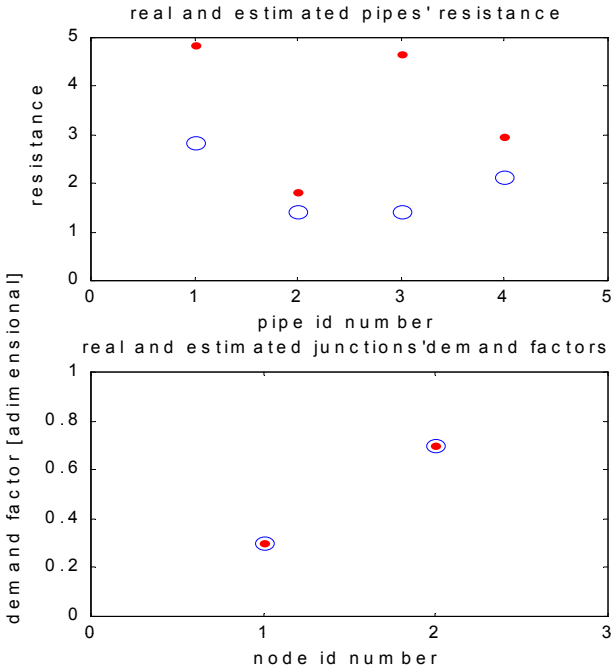


Fig. 2. 11

When two measurements are taken the matrix is positive definite as the minimum eigenvalue is λ=0.01, it is very small and it could mean that the problem is not well conditioned. The repetition of measurements improves the results. In figure 2.12 the evolution of the minim eigenvalue (log) is shown as the number of measurements increases. With 2 measurements it is already identifiable, with more measurements the numerical identifiability improves as the minim eigenvalue increases. This is the reason why the estimation is not that good and why with more measurements it improves, the results with 10 measurements appear in figure 2.13. Both resistances and demand factors appear well estimated and this estimation improves as the number of time steps increases.

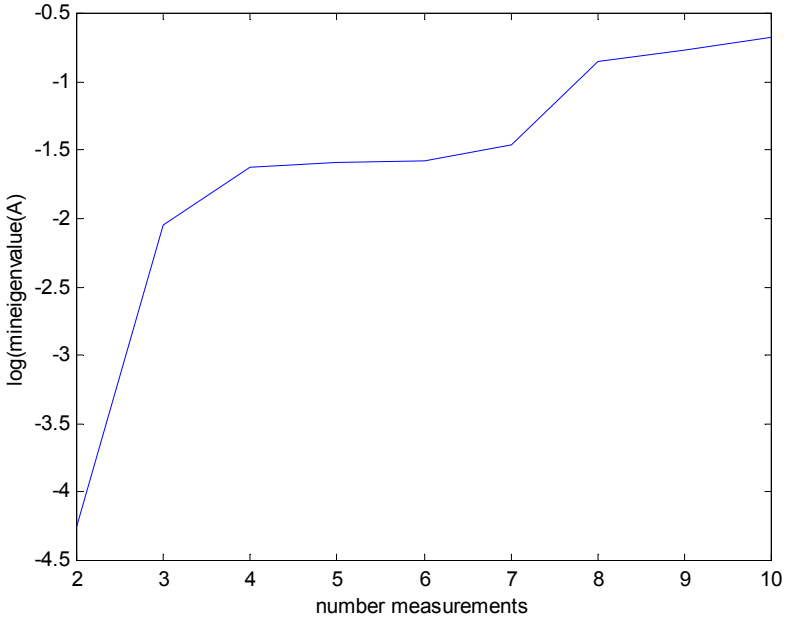


Fig. 2. 12

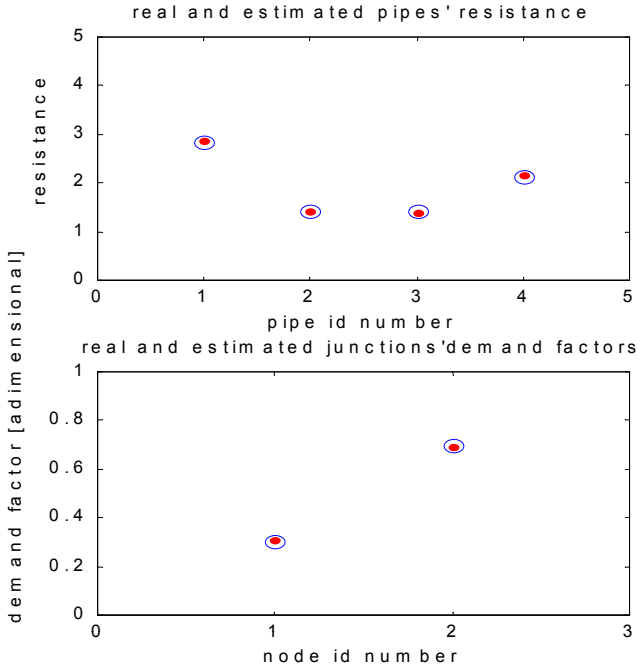


Fig. 2. 13

2.6.2. Known demand factors

To simplify the problem even further it can be assumed that w is known. Then equation 2.23 becomes

$$\left[\Lambda\ \Delta H\right] y = q_N \tag{2.27}$$

where q_N represents all nodal flows including the demands. If measurements are repeated many times and a collection of equations 2.24 can be arranged into a single equation

$$\begin{bmatrix} \Lambda\ \Delta H(1) \\ \Lambda\ \Delta H(2) \\ \dots \\ \Lambda\ \Delta H(K) \end{bmatrix} y = \begin{bmatrix} q_N(1) \\ q_N(2) \\ \dots \\ q_N(K) \end{bmatrix} \tag{2.28}$$

or alternatively

$$H(K)\ y = Q_N(K) \tag{2.29}$$

where

$$H(K) = \begin{bmatrix} \Lambda\ \Delta H(1) \\ \Lambda\ \Delta H(2) \\ \dots \\ \Lambda\ \Delta H(K) \end{bmatrix} \text{ and } Q_N(K) = \begin{bmatrix} q_N(1) \\ q_N(2) \\ \dots \\ q_N(K) \end{bmatrix}$$

The diagonal elements of each matrix $\Delta H(k)$ can be arranged into a column vector $\Delta \mathbf{h}^*(\boldsymbol{k})$

$$\Delta \mathbf{h}^*(k) \overset{\Delta}{=} \begin{bmatrix} |\Delta h_1(k)|^{0.54}\ sign(\Delta h_1) \\ \dots \\ |\Delta h_m(k)|^{0.54}\ sign(\Delta h_m) \end{bmatrix} \tag{2.30}$$

THEOREM

The number of linearly independent vectors $\Delta \mathbf{h}^*(k)$, k=1,2,…,K is denoted by s. Then

$$rank\ H(K) = s \tag{2.31}$$

This means that if

$$s = l \tag{2.32}$$

the water network is identifiable.

PROOF
The condition presented is a sufficient but not necessary. Then it must be demonstrated just in one way. By linear algebra it is known that in a linear system as 2.29 the condition to have a compatible and determined system is that the rank of matrix H(K) is m (number of unknown variables) assumed that the measurements are consistent and so the system can not be incompatible.

As the hypothesis of the theorem is that s=*l*, it (the theorem) will be demonstrated when 2.31 is demonstrated:

$$H(K)=\begin{bmatrix}\Lambda\ \Delta H(1)\\ \Lambda\ \Delta H(2)\\ \dots\\ \Lambda\ \Delta H(K)\end{bmatrix}=\begin{bmatrix}\Lambda & 0 & \dots & 0\\ 0 & \Lambda & \dots & 0\\ \dots & \dots & \dots & \dots\\ 0 & 0 & \dots & \Lambda\end{bmatrix}\begin{bmatrix}\Delta H(1)\\ \Delta H(2)\\ \dots\\ \Delta H(K)\end{bmatrix}$$

$$\Lambda=\begin{bmatrix}\Lambda_1 & \dots & \Lambda_m\end{bmatrix} \tag{2.33}$$

$$\Delta H(i)=\begin{bmatrix}\Delta h_1^*(i) & 0 & 0\\ 0 & \dots & 0\\ 0 & 0 & \Delta h_l^*(i)\end{bmatrix}$$

H(K) has m columns, if it is demonstrated that there are m linear independent rows then the rank of this matrix will be m. H(l) can be written:

$$H(K)=\begin{bmatrix}\Delta h_1^*(1)\Lambda_1 & \Delta h_2^*(1)\Lambda_2 & \dots & \Delta h_l^*(1)\Lambda_l\\ \Delta h_1^*(2)\Lambda_1 & \Delta h_2^*(2)\Lambda_2 & \dots & \Delta h_l^*(2)\Lambda_l\\ \dots & \dots & \dots & \dots\\ \Delta h_1^*(l)\Lambda_1 & \Delta h_2^*(l)\Lambda_2 & \dots & \Delta h_l^*(l)\Lambda_l\end{bmatrix} \tag{2.34}$$

where Λ_j are columns of the incidence matrix and the s=*l* linear independent vectors $\Delta\mathbf{h}^*(\boldsymbol{k})$ have been used. So we have m submatrixes (*n-1*×*l*). In order to have non-zero element rows we can use a linear combination of all the elements of a submatrix and substitute the first row. By construction (as in an incidence matrix) each column has two non-zero elements 1 and –1, these new rows will not have zero elements as they are the products of the sum of all the rows multiplied by its order:

$$H'(K)=\begin{bmatrix} \Delta h_1^*(1)x_1 & \Delta h_2^*(1)x_2 & \dots & \Delta h_l^*(1)x_l \\ \dots & \dots & \dots & \dots \\ \Delta h_1^*(1)\Lambda_{n-11} & \Delta h_2^*(1)\Lambda_{n-12} & & \Delta h_l^*(1)\Lambda_{n-1l} \\ \Delta h_1(2)x_1 & \Delta h_2^*(2)x_2 & & \Delta h_l^*(2)x_l \\ \dots & \dots & \dots & \dots \\ \Delta h_1^*(2)\Lambda_{n-11} & \Delta h_2^*(2)\Lambda_{n-12} & & \Delta h_l^*(2)\Lambda_{n-1l} \\ \dots & \dots & \dots & \dots \\ \Delta h_1^*(l)x_1 & \Delta h_2^*(l)x_2 & & \Delta h_l^*(l)x_l \\ \dots & \dots & \dots & \dots \\ \Delta h_1^*(l)\Lambda_{n-11} & \Delta h_2^*(l)\Lambda_{n-12} & & \Delta h_l^*(l)\Lambda_{n-1l} \end{bmatrix} \tag{2.35}$$

where

$$\Delta h_i^*(k)x_i=\sum_{j=1}^{n-1} j\Delta h_i^*(k)\Lambda_{ji}=\Delta h_i^*(k)\sum_{j=1}^{n-1} j\Lambda_{ji} \tag{2.36}$$

As these rows are linear combination of the others, the rank of the matrix cannot increase. So if rank(H'(K)) is m so will be rank(H(K)). By taking only these new rows we get an m×m matrix. If all the columns are linear independent so will be the rows. To have a column, which is a linear combination of the columns, equation 2.37 should have a different solution apart from the trivial ($\lambda_1=\lambda_2...=\lambda_l=0$):

$$\lambda_1\begin{bmatrix}\Delta h_1^*(1)x_1\\ .\\ \Delta h_1^*(l)x_1\end{bmatrix}+\lambda_2\begin{bmatrix}\Delta h_2\ (1)x_2\\ .\\ \Delta h_2^*(l)x_2\end{bmatrix}+\dots+\lambda_l\begin{bmatrix}\Delta h_l^*(1)x_l\\ .\\ \Delta h_l^*(l)x_l\end{bmatrix}=\begin{bmatrix}0\\ \dots\\ 0\end{bmatrix} \tag{2.37}$$

It can be written as a linear system:

$$\begin{bmatrix} \Delta h_1^*(1) & \Delta h_2^*(1) & \dots & \Delta h_l^*(1) \\ \Delta h_1^*(2) & \Delta h_2^*(2) & \dots & \Delta h_l^*(2) \\ \dots & \dots & \dots & \dots \\ \Delta h_1^*(l) & \Delta h_2^*(l) & \dots & \Delta h_l^*(l) \end{bmatrix}\begin{bmatrix}\lambda_1 x_1\\ \lambda_2 x_2\\ \dots\\ \lambda_l x_l\end{bmatrix}=\begin{bmatrix}0\\ 0\\ \dots\\ 0\end{bmatrix} \tag{2.38}$$

In order to have a non trivial solution the *l×l* matrix of the former system should be singular and it is not possible as the rows are the linearly independent $\Delta\mathbf{h}^*(k)$, so it has been demonstrated that H'(K) has rank m and so has H(K).

Example 2.10: The same system, figure 2.3, used before has been tested. The results are the same as those obtained when the demands were not known. So the system is

identifiable with two or more time steps. The condition $s=l=4$ implies that there are three linear independent vectors $\Delta\mathbf{h}^*(k)$ so at least four time steps should be taken. It is a sufficient but not necessary condition, so it is not inconsistent that with two time steps it works. In table 2.1 appear the singular values for different numbers of time steps. Here it can be seen that the problem is not numerically well conditioned.

Table 2.1

1 time step	2 time steps	3 time steps	4 time steps
3.31	4.6	5.6	4.91
-	0.16	0.46	0.59
-	-	0.01	0.01
-	-		0.002

2.6.3. Non-linear case

The linear methodology has been generalised for more time steps. When not all heads are known the methodology based on Jacobian, presented in section 2.4.2 for the static case, should be used. The notation used in section 2.2 is justified now because of the easy generalisation. Some indices will be defined, they will help in the construction of the $\nabla f_1(r)\cdot\nabla f_1(r)'$ matrix, a generalisation of equation 2.16. There are still *r* reservoirs, *n* junctions (with demand factor associated), *l* pipes (with resistances associated). The indices for the levels of reservoirs (hr), heads of junctions (h) and flows of pipes (q) that are measured are *zhr, zh, zq* respectively. The indices for the resistances of pipes (R) and demand factors of junctions (w) that are known are *zr* and *zw* respectively. The indices for the levels of reservoirs (hr), heads of junctions (h) and flows of pipes (q) that are not measured are *xhr, xh* and *xq* respectively. The indices for the resistances of pipes (R) and demand factors of junctions (w) that are known are *xr* and *xw* respectively. The last ones are the parameters that are tuned and the non-measured variables have to be estimated as they are part of the same problem. An important index is *t* for the number of time-steps. Total demand d(k) is a curve for *k=0...t*, there are *t* vectors of levels, heads and flows (hr(k), h(k) and q(k)).

The vector in which variables have to be estimated includes all the variables and parameters, so that if the measurements change or the number of time steps increases the new system is automatically generated- this was seen in the static study. Equation 2.39 shows the construction of the vector x:

$$x = \begin{bmatrix} x(1) \\ . \\ . \\ x(t) \\ hr(t+1) \\ R \\ w \end{bmatrix} \text{ where } x(k) = \begin{bmatrix} hr(k) \\ h(k) \\ q(k) \end{bmatrix} \tag{2.39}$$

The total number of variables is (l+n+r)*(t+1). Available equations are:

$$\begin{aligned} & A_i(hr_i(k) - hr_i(k+1)) + \Lambda(i)\, q(k) = 0 \quad i = 1,2,...,r \\ & \Lambda(i)q - w_i d(k) = 0 \quad i = r+1,2,...,r+n \\ & \Delta h_j(k) - R_j\ q_j(k)\,|\,q_j(k)\,|^{0.852} = 0 \quad j = 1,2,...,m \\ & k = 0,1,...,t \\ & \sum_{i=1}^{n} w_i = 1 \end{aligned} \tag{2.40}$$

And those generated by measurements:

$$\begin{aligned} & x((k-1)*(r+n+l)+zhr) = hr(k,zhr) \\ & x((k-1)*(r+n+l)+r+zh) = h(k,zh) \\ & x((k-1)*(r+n+l)+r+n+zf) = q(k,zf) \\ & k = 0,1,...,t \\ & x(t*(r+n+l)+zhr) = hr(t+1,zhr) \\ & x(t*(r+n+l)+r+zr) = R(zr) \\ & x(t*(r+n+l)+r+l+zw) = w(zw) \end{aligned} \tag{2.41}$$

Jacobian will be composed by one corresponding to the variables in which the number increases with time steps (Jx_t) and one corresponding to the constant parameters (Jxr).

$$J = \begin{bmatrix} J_{x_t} & J_{x_r} \\ zeros((r+n+l)*t+r+l) & ones(n) \end{bmatrix} \tag{2.42}$$

The last row is the relation between demand factors. The first one is the composition of those corresponding to each time-step.

$$J_{x_t} = \begin{bmatrix} J_{x_t}(1) & D(2) & 0 & 0 & . & 0 \\ 0 & J_{x_t}(2) & D(3) & 0 & . & . \\ . & 0 & 0 & . & . & \\ . & . & . & . & . & \\ 0 & 0 & 0 & J_{x_t}(t) & D(t+1) & 0 \\ & & zeros(z_0,(r+n+l)(t+1)) & & & \end{bmatrix} \tag{2.43}$$

$$J_{x_t}(k) = \begin{bmatrix} A*I(r,r) & & \Lambda & & zeros(r+n,l) & \\ zeros(n,n) & & & & & \\ \Lambda^T & -1.852R_n & \begin{bmatrix} q_1^{0.852}(k) & 0 & 0 \\ ... & ... & ... \\ 0 & 0 & q_l^{0.852}(k) \end{bmatrix} & \begin{bmatrix} q_1^{1.852}(k) & 0 & 0 \\ ... & ... & ... \\ 0 & 0 & q_l^{1.852}(k) \end{bmatrix} & & \\ & & B_t & & & \end{bmatrix}$$

$$D(k) = \begin{bmatrix} -A*I(r,r) \\ zeros(n+l+z_t,r) \end{bmatrix} \tag{2.44}$$

B_t is a submatrix with z_t rows and r+n+l all zeros except a 1 in each row in the position of a measurement (*zhr,zh,zf*).

$$J_{x_r}(t) = \begin{bmatrix} J_{x_r}(1) \\ . \\ . \\ J_{x_r}(t) \\ B_0 \end{bmatrix} \tag{2.45}$$

$$J_{x_r}(k) = \begin{bmatrix} -I(r,r) & -d(k)\begin{bmatrix} zeros(r,n) \\ I(n,n) \end{bmatrix} \\ zeros(n,r) & \\ zeros(l,r) & zeros(l,n) \\ zeros(z_t,l+n) & \end{bmatrix} \tag{2.46}$$

B_0 is a submatrix with z_0 rows and n+l all zeros except a 1 in each row in the position of a measurement (*zr,zw*).

The identifiability condition is that the rank of J is equal to the number of variables, this is the generalisation of condition used in section 2.4.2. In order to get a more compact condition equation 2.16 can be applied with new J_{xt}. And matrix $\nabla_{xr}z$ plays a equivalent

role to H(K) in the linear case. $\nabla_{xr}z\ \nabla_{xr}z'$ has always the same dimension, l+n×l+n. The identifiability condition in this case is this matrix not to be singular. Each Jacobian is a very large matrix and no example will be shown explicitly.

Example 2.11: This methodology can be applied to the linear case so that identifiablility of example 2.9 with more than 1 time step has been stated as Jacobian becomes of full-rank. For the non-linear case, the dynamic identifiability was used in example 2.6. It showed that the dynamic system was identifiable, the same conclusion would be obtained applying the extended-period identifiability. For the example 2.4 and 2.5 this is the methodology that allows the generalisation of the identifiability as with unknown heads the problem is non-linear. The Jacobian when there are two time steps used is 30x24 and its rank is 24 so it is full rank and the system is identifiable the compact matrix $\nabla_{xr}z\ \nabla_{xr}z'$ is a 6×6 matrix with rank 6. Figure 2.14 shows the estimation of the parameters that has been successful. Here the increase of time-steps has saved the addition of a sensor.

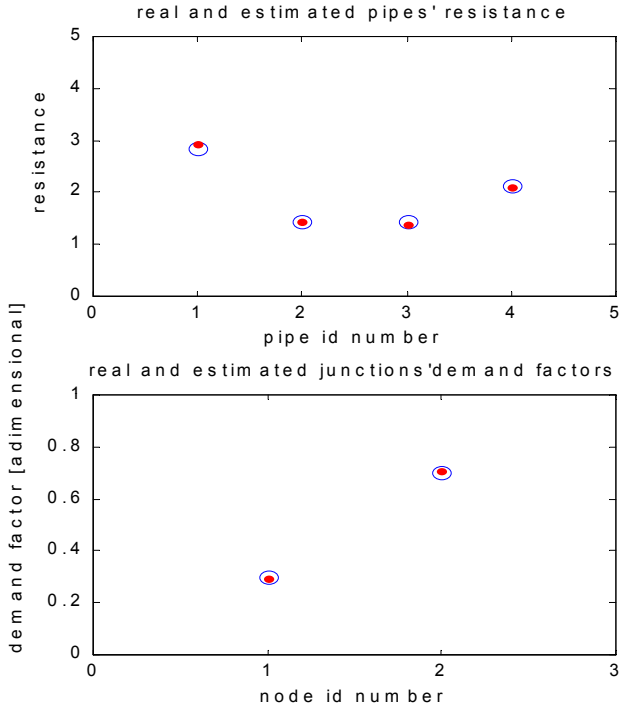


Fig. 2. 14

This is the methodology used from this point. It is the most general that can be applied to non-linear systems with more than one time-step. The extended period identifiability is then studied using the jacobian of the function.

2.7. Identification problem design

In section 2.3 the three main decisions for the identification experiments are presented. Sensor placement, understood as output specification, is a common problem for static and extended period problems. Introducing new time steps, the time horizon and orthogonality of inputs come across. The minimum time horizon to assure identifiability, given a set of sensors, is determined by methodologies presented in previous section. More time steps are always taken in order to have redundancy so it is not critical as long as the network is identifiable. These repeated measurements provide new information if they are not taken in the same conditions. The choice of conditions – inputs- is a really critical point in identification process. So a methodology should help to make these two decisions easy when trying to identify a big network.

In the linear case a matrix that doesn't increase with time steps $(H(K))^T H(K)$ has been used. Such a matrix does not exist in non-linear case because flows cannot be substituted in general, as they can be in the linear case where all heads are known. The equivalent matrix is the Jacobian of the measurements with respect to the parameters. When this Jacobian is multiplied with its transpose a square matrix is found with dimensions equal to the number of parameters. If all variables are taken as parameters that have to be estimated this product (jacobian by its transpose) loses the compactness but it can be used in the identifiability study as it has been shown in previous section and the presence of all variables will allow the meter placement decisions. To assure identifiability this product should be positive definite -that means (as it is real and symmetric) its eigenvalues are positive. These eigenvalues and eigenvectors give the singular value decomposition of the Jacobian. This decomposition will be very useful when making decisions in the process of calibrating a network.

Interpretation of this Jacobian and its singular values is fundamental in order to get some conclusions. In multivariable control [Sko-96] singular value decomposition is well known and gives a useful tool to treat both spaces, input and output for a system. Singular values relate some special directions in input space to some directions in output space.

Definition 2.1: Any complex l×m matrix A may be factorised into singular value decomposition

$$A = U\Sigma V^T \tag{2.47}$$

where the l×l matrix U and the m×m matrix V are unitary, and the l×m matrix Σ contains a diagonal matrix Σ_1 of real, non-negative singular values, σ_i, arranged in a descending order as in

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix};\ l \geq m$$
$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix};\ l \leq m \qquad (2.48)$$
$$\Sigma_1 = diag\{\sigma_1, \sigma_2, ..., \sigma_k\}\ \ k = min\{l, m\}$$
$$\overline{\sigma} \equiv \sigma_1 \geq \sigma_1 \geq ... \geq \sigma_k \equiv \underline{\sigma}$$

The unitary matrices U and V form orthonormal basis for the column (output) space and the row (input) space of A. The column vectors of V, denoted v_i, are called right or input singular vectors and the column vectors of U, denoted u_i, are called left or output singular vectors. Singular values are calculated following equation 2.49.

$$\sigma_i(A) = \sqrt{\lambda_i(A^H A)} \qquad (2.49)$$

Where $\lambda_i(A^H A)$ is the eigenvalue *i* of the matrix product $A^H A$.

Example 2.12: In the example 2.4 and 2.5 the same problem is analysed. The jacobian used in 2.5 is not a square matrix as it is not in general. This matrix has the physical meaning of a linear approximation of the function that relates the variables that are estimated and the measurements. The singular value decomposition gives the following results.

$$\Sigma = \begin{bmatrix}
28.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 27.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 24.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 14.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 12.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.99 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.97 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.12 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.003 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00 & 0
\end{bmatrix}$$

It was already known that the system was not identifiable. The last singular value is 0 that means that the jacobian is not full rank which confirms that the system is not identifiable. Now using the singular vectors some further knowledge can be extracted.

In figure 2.15 the magnitude order of the singular value is presented and it can be seen that the last one is zero.

In the second plot the rounds (o) represent the values of the components of the input singular vector corresponding to the smallest singular value. This singular vector is the direction in the input space. In this particular case it can be seen that the components that are most relevant correspond to those variables that are not measured , i.e. head of node 2 and roughness in pipes 3 and 4. Symbol sum (+) is used for the same value if the variable is measured. It can be seen that neither head in second junction or roughness are measured. This suggests that introducing a sensor in this node would improve the identifiability.

The third plot shows only the value of the sum of the absolute values of the components of the last singular vector (V) related with levels, heads and flows that are the ones that can be measured. In this case again the most relevant is the head of node 2 that is not measured.

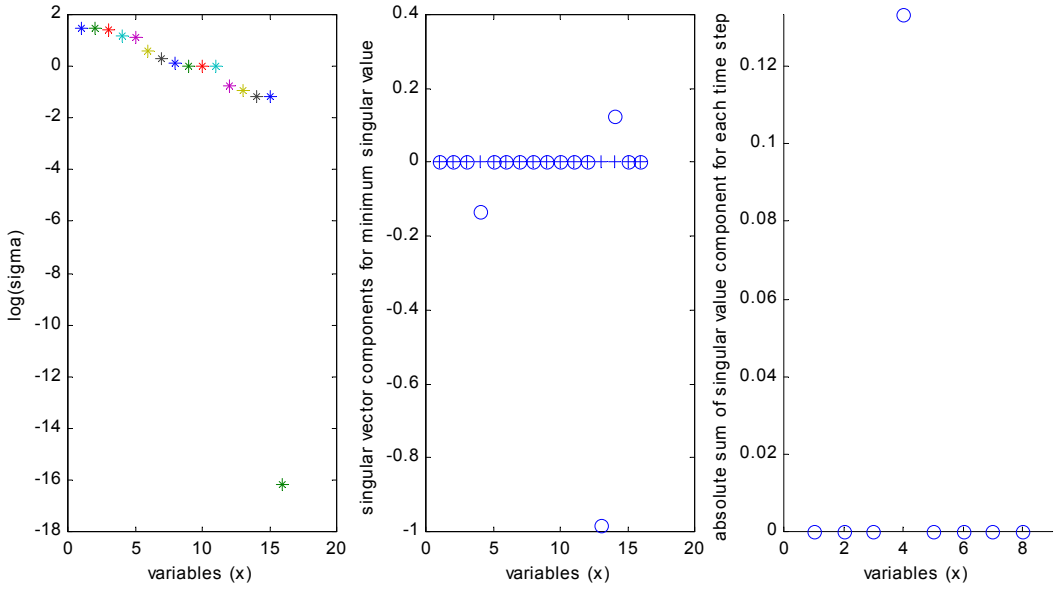


Fig. 2. 15

Watching the identification as a set of non-linear equations approximated in a linear one by Jacobian, this matrix relates an input space of variables with an output space of measures. The non-identifiability of the problem is detected by a zero singular value that is related with a specific direction in input space. Variables included in this direction would improve identifiability in case they were measured. Observation of the input singular vectors gives a guide in the difficult task of sensor placement.

For the orthogonality of inputs (not variables but conditions) a sufficient condition has been given for the linear case where demands are known. These are rather restrictive situation and conditions. For non-linear cases the examination of minimum singular value gives an idea of the numerical conditioning of the problem. For the coefficient of maximum and minimum singular value -condition number- is used in control. As the original model has the same structure as the real one and conditions that can be changed are known, the results of the simulations with the original model will give a good estimation of what will happen in experimentation.

2.8. Topology characterisation

Using the same techniques some knowledge about topologies in networks has been investigated. Basically the two main topologies used in water distribution networks are the tree ones with no redundancy and those that contain loops introduced to assure a reliable service. Two studies have been carried out, since in order to compare these two topologies a criterion of size has been defined. First two networks with the same number of nodes are used that means that the number of pipes is greater when some loops exist. A part of anytown network shown in figure 2.19 is used. The identifiability of a tree network with 1 reservoir, 7 junctions and 7 pipes (figure 2.16) is compared with the identifiability of a network with 1 reservoir and 7 junctions and 12 pipes (figure 2.17). There are 5 loops (elements of cotree). 24 time steps, corresponding to day simulation with 1 hour sampling, are used. In table 2.2 the difference in the sensor requirement by both topologies is shown. The sensors have been chosen using the jacobian approach to decide the identifiability and the singular value decomposition to decide the next sensor to be added. The network with loops has more elements to identify but needs fewer sensors because of the topology.

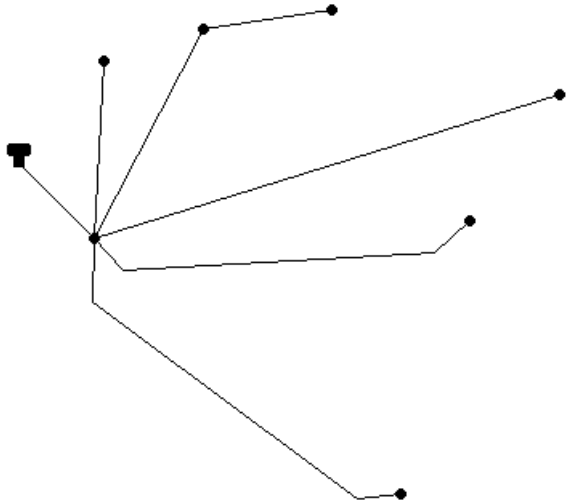


Fig. 2. 16

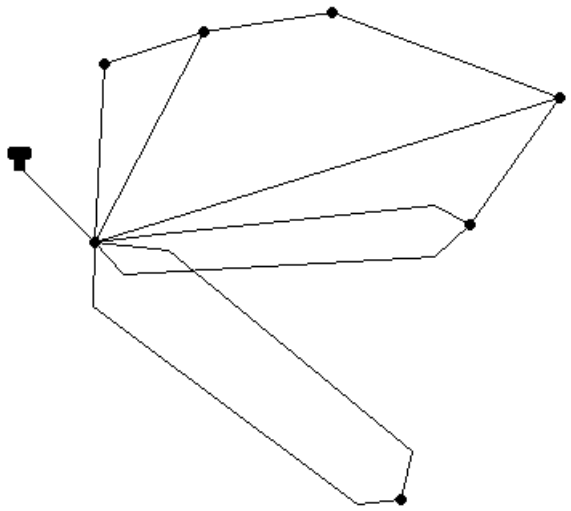


Fig. 2. 17

Table 2.2

Network	Sensors
Tree network (figure 2.16), 7 junctions and 7 pipes	1 level, 7 head and 6 flow
Loop network (figure 2.17), 7 junctions and 12 pipes	1 level, 6 head and 4 flow
Loop network (figure 2.18), 5 junctions and 7 pipes	1 level, 5 head

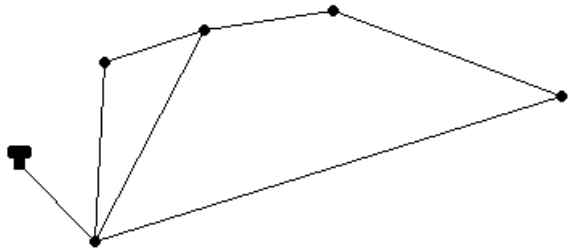


Fig. 2. 18

Using the number of pipes as a measure of the network an equivalent network with 7 pipes but with loops, instead of the tree structure, is used (figure 2.18). Its identifiability has been compared with that of the tree network. Again fewer sensors are required as can be seen in table 2.2.

2.9. Case Study

Recent developments, including the restructuring of water networks into smaller subnetworks (called Pressure Management Areas (PMA)) and the introduction of more pressure and flow monitoring points have increased the opportunity of getting reliable models. A network called **Anytown,** a benchmark provided by WSS [WSS-03], is used to show the methodology of the experiment design for identification based on the identifiability study presented in this thesis. It could be part of a real network, a PMA controlled by two PRV and with 24 nodes, 1 tank, 2 reservoirs and 53 pipes.

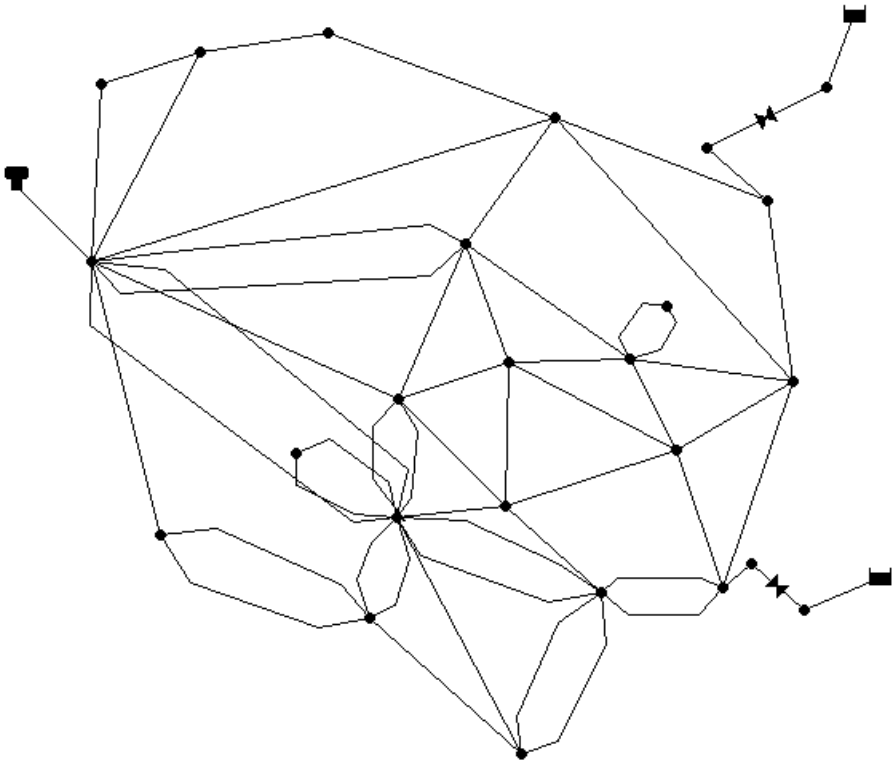


Fig. 2. 19

It is assumed that the reservoirs and tanks have level sensors and their in/out flows are measured too. With one time step,3 level sensors, 24 head sensors and 54 flow sensors are required. Figure 2.20 shows the singular values and vectors before adding the last sensor in junction 5 and become identifiable. Only two components appear; two heads (resistances and demand factors are not plotted).

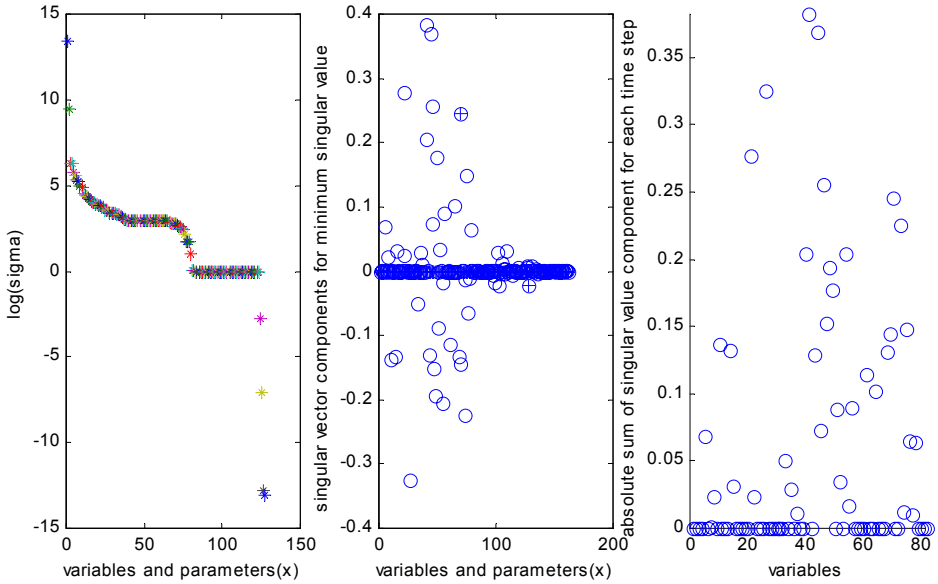


Fig. 2. 20

In figure 2.21 the minimum singular value after adding the last sensor has increased to ~10^{-10}. It can be seen in the first plot where the logarithm of singular values are presented.

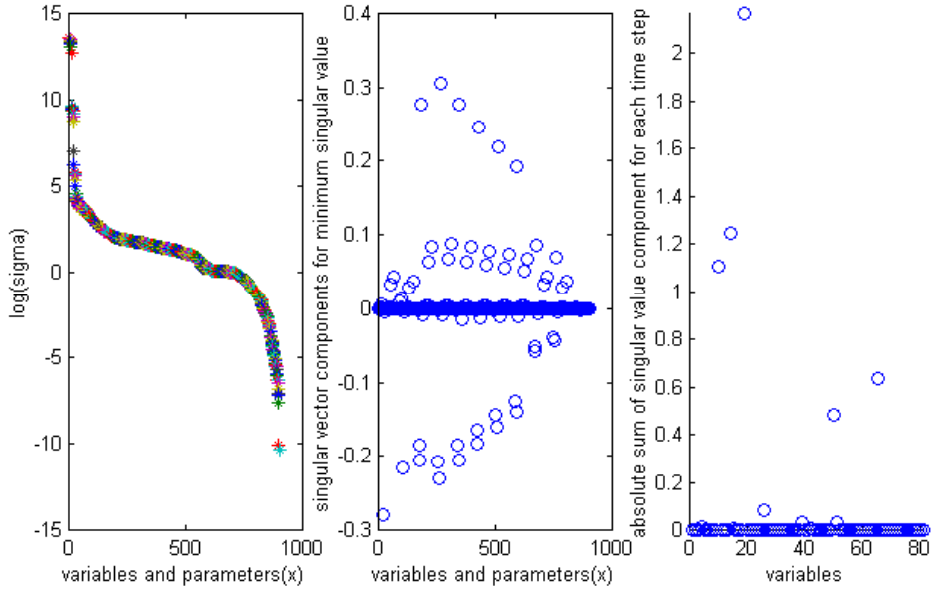


Fig. 2. 21

Using 10 time steps the number of sensors is reduced to 3 level sensors, 12 head sensors and 5 flow sensors. With 24 time steps the number of sensors is the same. With this

configuration the minimum singular value is similar to the previous ~10^{-10} (figure 2.22) and the system becomes identifiable. It is not very well numerically conditioned but the decrease of number of sensors makes clear the importance of taking into account the advantages of more than one time step in identification.

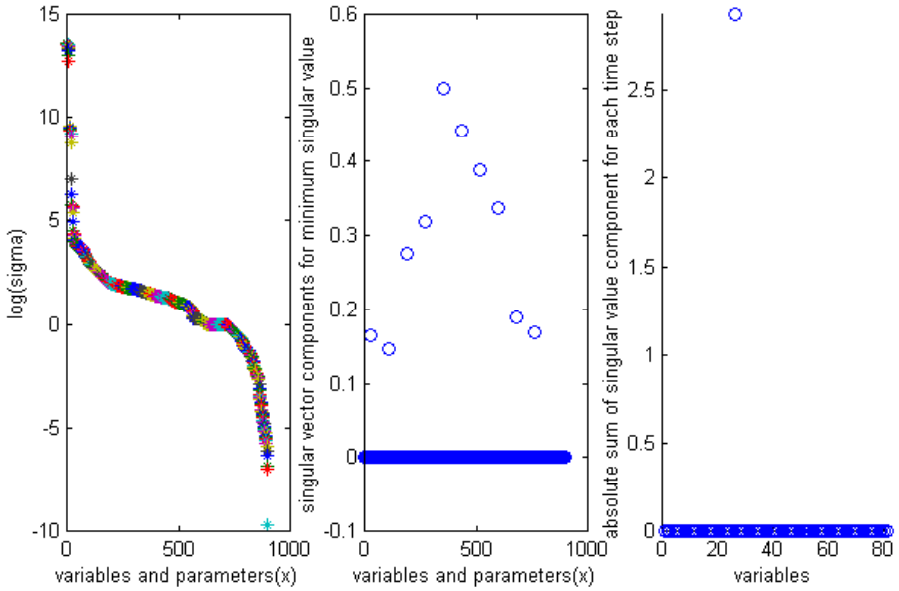


Fig. 2. 22

The results of the identification are not satisfying. These bad results are due to the algorithm used. In section 4.3 it is demonstrated that the optimisation needed for such a problem is global optimisation. In this chapter such algorithms have not been used. Only the Matlab local algorithms based on gradient have been used for their availability in the moment of the study of identifiablility. These algorithm have been useful for small examples when the identifiablity methodology was defined. Next step in the calibration process is to start to improve the model from the initial one. This is done in chapters 3 and 4.

3. Macrocalibration of Water Networks

The first step of calibration should attempt to correct gross errors introduced during model building. The characteristic of these errors is that the measurements disagree by a large amount. The typical causes are:

Inaccurate plant models such as pumps and reservoirs
Incorrect network topology
Incorrect boundary conditions

This stage of calibration is normally done manually using some common sense rules [WSS-98]. But it can be time-consuming and require skill and judgement. In order to perform macrocalibration automatically, the knowledge has to be organised. An expert system allows the diagnosis of errors and finding their causes. Another approach would be the classification of the errors for correction.

In the research group in Terrassa [UPC-03], some work has been done on diagnosis using classification [Pie-91]. Aguilar [Agui-01] treats the supervision of process. It includes data collection, data processing and finally interpretation as meaningful messages for an eventual operator. This last functionality needs a dialogue with process experts so that the functional states of the process are correctly defined and interpreted, making reference to physical, chemical or biological concepts. Most essential is to be able to determine if the process is in normal or abnormal operation, so that the supervisory system helps the operator to choose the possible actions. The proposed methodology can be described in three steps: first to make an independent clustering from past running data, second to establish a dialogue with the expert so that as the classes resulting from the clustering process can be associated with meaningful physiological states (most work has been done in chemical process), third and finally to construct a finite state machine that represents the possible transitions between physiological states and enables the real time monitoring of the process [Sarr-02].

The tool is expected to diagnose and correct errors. When there is more than one error present the strategy of MYCIN can be adopted [Dur-94]. When some possible causes of errors have been detected and the changes in the model have been proposed, the tool

should test these changes and check the improvement in the results according to the scheme in figure 3.1.

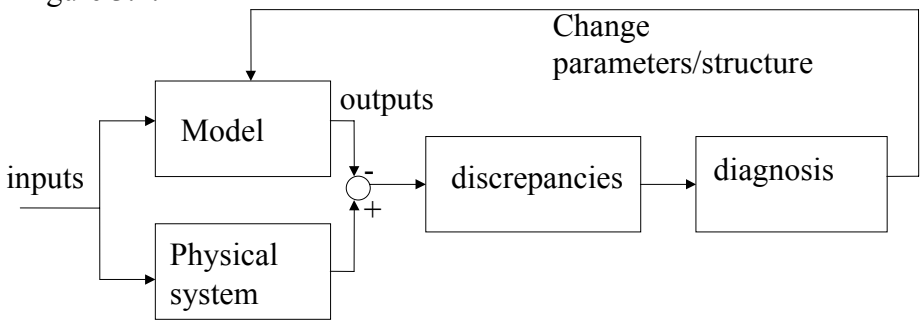


Fig. 3. 1

Of course, as the errors searched for in this phase of calibration are mixed with the ones intended to be solved in further steps, the former are more difficult to find out. This is a weak point of the calibration process that can only be solved by means of iteration of the whole process. The idea is to help the manual process of finding big errors. In figure 3.2 this idea of iteration is presented. Macroclibration is repeated after the microcalibration with the new parameters so the model will improve with each step.

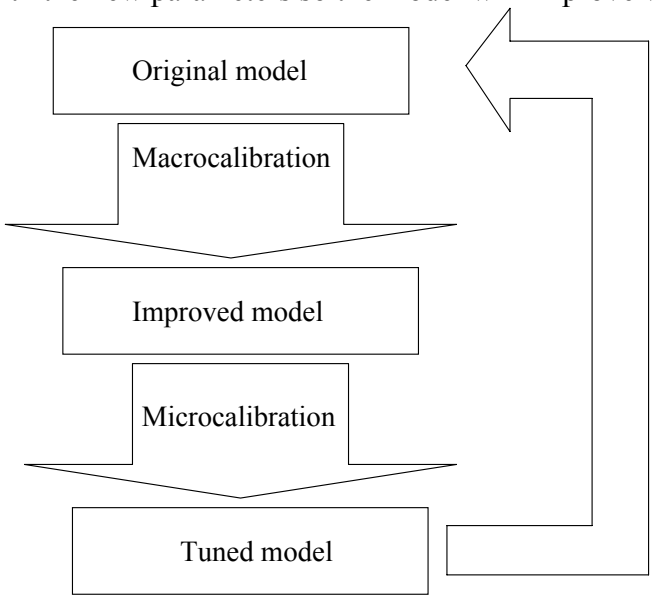


Fig. 3. 2

The main idea is that this tool allows the detection of these errors. When one error is detected it can be checked in the information sources. Each improvement in the model may allow the detection of other errors thus the process is iterative.

In a real network normally the measurements available are not more than 20% of the variables. Such a situation is a handicap for this tool but not worse than for the expert who is nowadays carrying out this process. The methodology proposed in this thesis has

been selected to make easier the generation of the knowledge used for the error detection. This knowledge will be different depending on the distribution of sensors in the network.

In section 3.1 the knowledge organisation is discussed. In 3.2 the errors that are treated in this process are presented following expert's indications. Classification method used is described in section 3.3. The tool for macrocalibration is presented in section 3.4, there the concrete aspects of the implementation are described and illustrated by examples.

3.1. Knowledge Organisation

The knowledge provided by calibration experts needs to be organised to allow the identification of errors in the model. This organisation is not a trivial task and different approaches can be used. The choice will depend on the origin of this information. Two approaches are presented and the choice is justified.

3.1.1. Expert system

This system would examine the errors between the results of the initial model and the measurements, diagnose the causes of large errors and prescribe some action in order to reduce them. Knowledge derived from the experts that do this task could be captured in rules (knowledge base) in a intuitive way [Dur-94]. However if the knowledge comes from experimentation, the generation of rules is not direct and requires much heuristics and definition of thresholds etc. The "working memory" consists of the model results and measurements. The "inference engine" combines both informations and automatically the conclusions that the expert would have deduced (figure 3.3).

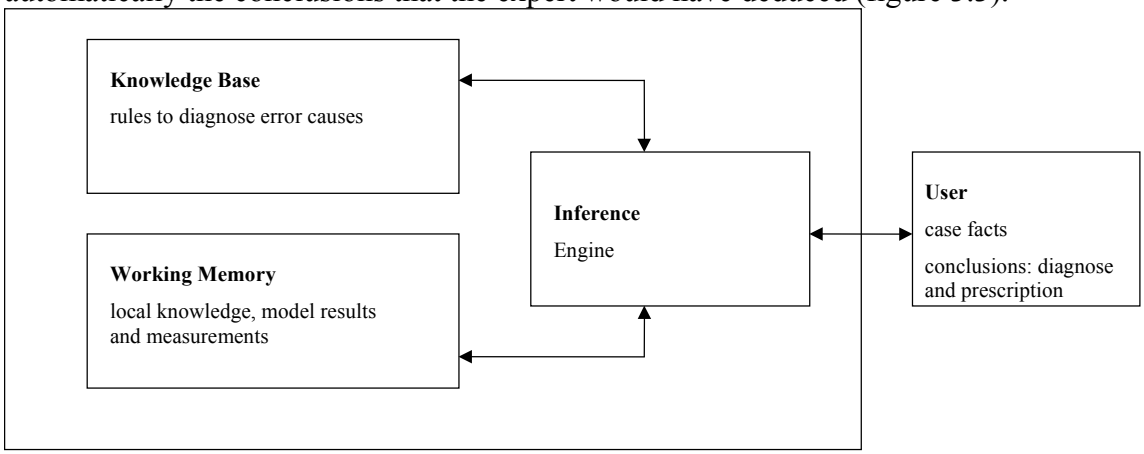


Fig. 3. 3

There are different approaches to knowledge representation. A rule-based Expert System is a computer program that processes problem-specific information contained in the working memory with a set of rules contained in a knowledge base, using an inference engine to infer new information. This is the most popular choice. For many problems, humans naturally express their problem solving knowledge in IF…THEN type statements. A rule is an independent chunk of knowledge, its independence permits it to be reviewed and verified easily. New rules can be added anywhere in the knowledge base, so long as the necessary logical relationships are assured. Heuristic rules that work in a common-sense fashion can be written and the system can establish a level of belief in the rules' premises and conclusions.

3.1.2. Classifying algorithms

The goal of classification is the identification of structures in data similar to known structures. Methods developed in the field of soft computing such as fuzzy logic are becoming increasingly popular. Such methods offer an attractive alternative to statistical approaches, as they do not require prior assumptions of statistical models.

The objects to be classified in the calibration problem are the nodes and elements. Each object may be associated with a class (error), which is characterised by the ρ-coefficient. Descriptors are derived for each class. Descriptors will be the same for nodes and reservoirs, other descriptors will be defined for elements (pipes, valves and pumps).

The first step is to define the descriptors of the classes. This is the most heuristic part of the method. The number of descriptors defines the dimension of the space of objects to be classified. The quantification of each descriptor defines the object. The classification process uses this information to recognise whether an object belongs to a class that is defined by a certain quantification of the descriptors.

In figure 3.4 the process of classification is represented. The objects will be all the nodes, and elements with the descriptors of their possible modelling errors. Classes correspond to error in the elevation, demand, roughness and so on. The macrocalibration finishes when all objects are classified as non-erroneous ones.

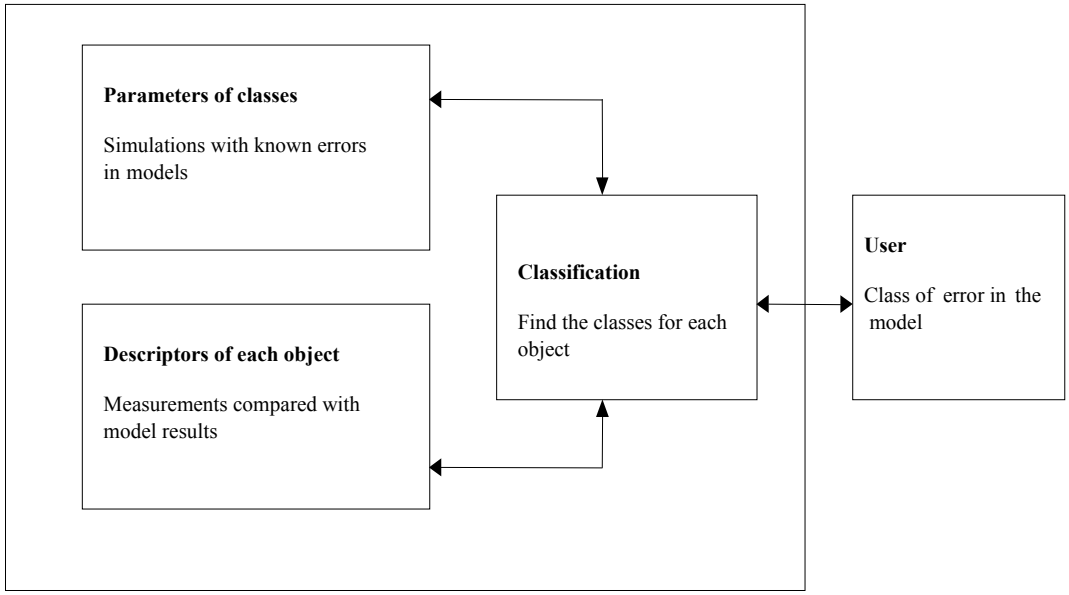


Fig. 3. 4

A major drawback for the use of an expert system is the generation of the rules, i.e. the knowledge that should be provided by an expert. The experts are not always available and even then the work is not straightforward. The use of logical connectives based classifying algorithms allows generation of knowledge automatically using optimisation. The definition of the descriptors is based on heuristics. Rules in expert systems, when working with quantities, need thresholds in order to decide whether the antecedents are fulfilled. In the classification process no threshold must be defined. The choice of LAMDA (learning algorithm for multivariate data analysis) [Agu-99] classification method is based on these considerations. Rules of the expert system are presented as example in section 3.4, this natural language is useful to understand the definition of the errors. From these rules it is easy to generate the descriptors.

3.2. Errors in Macrocalibration Stage

After some meetings with experts who manually carry out this stage of the calibration process the steps followed have been organised. This will allow performing it more automatically. In some cases information will be translated to the fuzzy method of conceptual clustering proposed for this process, some steps will be programmed as a classical algorithm. This could help the expert to detect and even correct the gross errors in the model. The steps are presented in the order that they normally appear in the process.

3.2.1. Disconnection

First of all the existence of separate networks is at least good information for the calibration to begin with, most of the time this is due to errors made while introducing

data. Special interest should be taken for separate strings (networks without supplies). Software such as Strumap (GIS by Geodesys Limited) [Geo-03] includes such tracing that detects these disconnections.

Example 3.1: The network in figure 3.5 is an obvious case of disconnected network. Obvious because if it is compared with the one in figure 3.6 the difference is clear between them. With a huge amount of pipes and nodes it is not so easy to detect if a pipe is missing. Thus experts examine whether there is an unsuspected disconnected subnetwork.

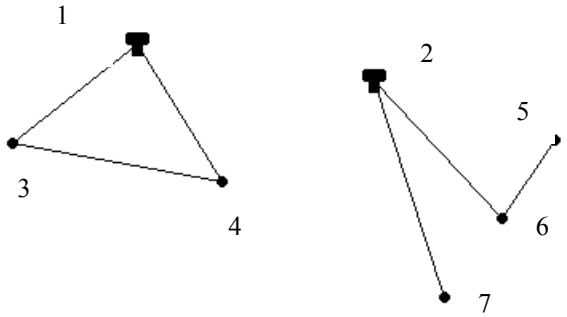


Fig. 3. 5

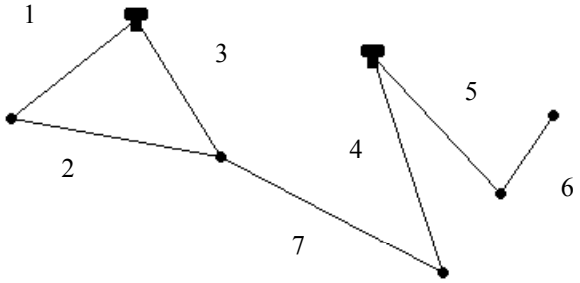


Fig. 3. 6

In this case as there are reservoirs in both subnetworks the simulation finishes successfully. The model could be accepted but the comparison (figure 3.7) of the heads in one time step (t=10h) –correct (connected model ‘o’) vs. erroneous (disconnected model ‘*’)- shows the difference.

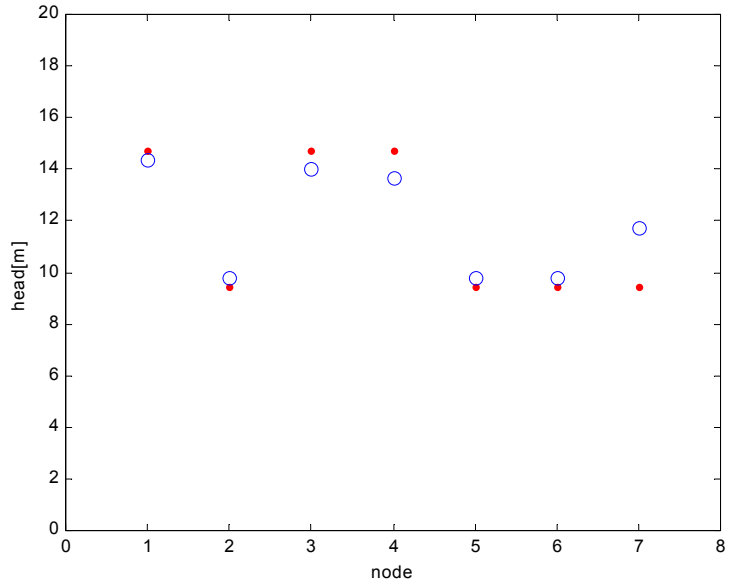


Fig. 3. 7

Figure 3.8 shows the evolution of level in both reservoirs. In the disconnected model (dashed line) reservoir 1, with higher elevation, does not supply as much water as in the real system, this water has to be supplied by the second reservoir.

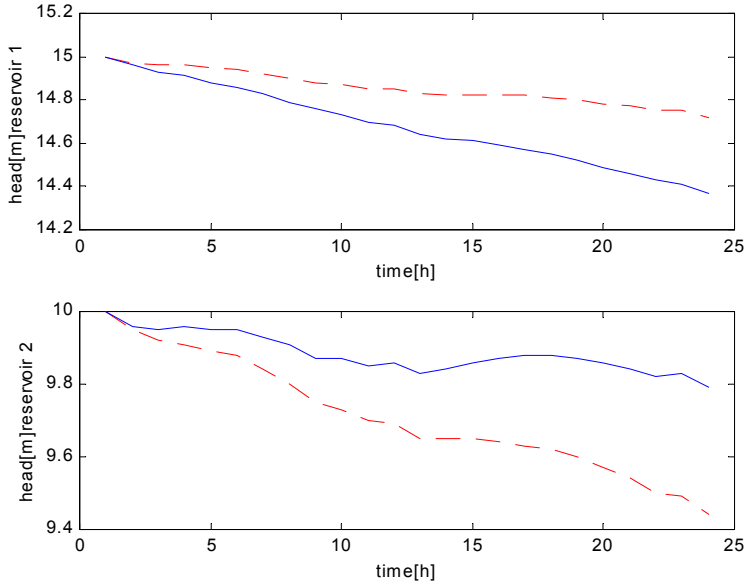


Fig. 3. 8

If the flows were measured the disconnection could have been detected when the missing pipe's flow were compared but in general there are no flow meters dispersed

throughout the network. Tuning parameters cannot solve such a discrepancy. The disconnection is studied before any further step in the calibration, and when detected, test if it is coherent with the real network.

3.2.2. Total demand

In simulators the domestic demand is modelled as a pattern of total demand that then is allocated by fixed demand normalised coefficients associated to the nodes. Before any study of the model is done the total demand (profile) must be compared with the SCADA data. A discrepancy may come from the use of old profiles.

Example 3.2: The same network of figure 3.6 has been simulated with two demand patterns. In figure 3.9 appear both the patterns –the correct and the wrong one (dashed)-.

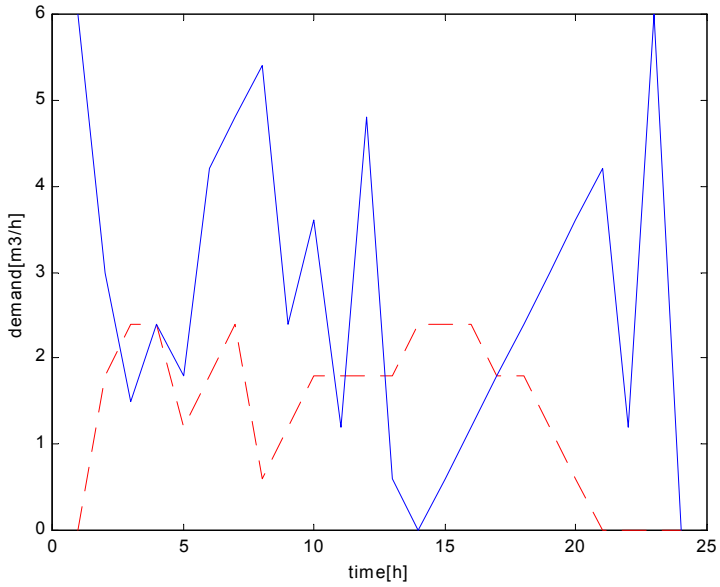


Fig. 3. 9

Looking at heads at nodes and reservoirs in figure 3.10 and flows in figure 3.11 (time t=10h) it can be seen that this kind of error is widespread and unless it is detected such discrepancy invalidates parameter tuning.

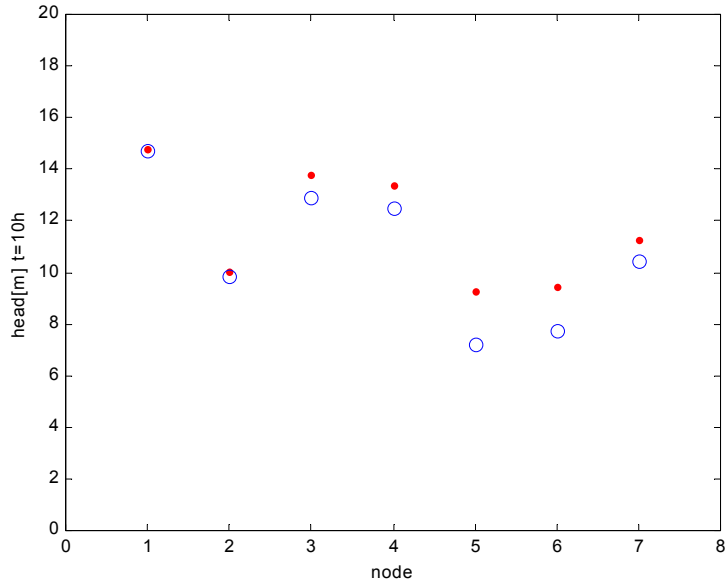


Fig. 3. 10

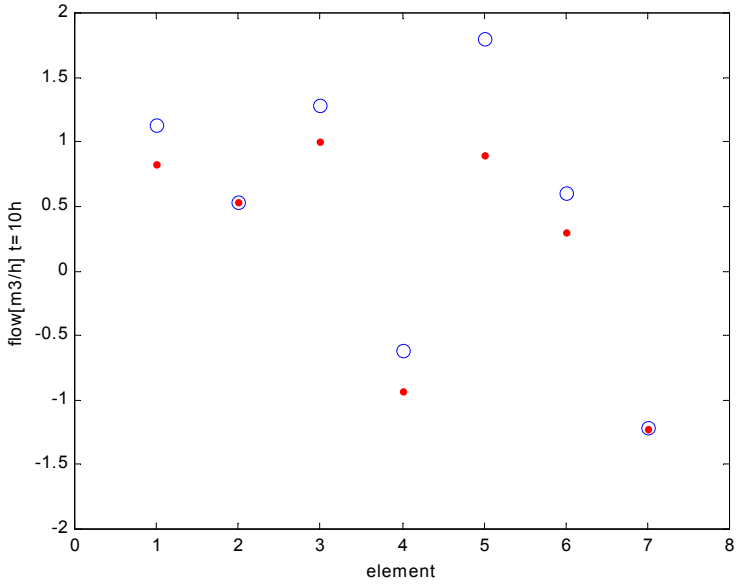


Fig. 3. 11

3.2.3. Supplies

Boosters are the pumps that introduce the water to the network. A difference in the flow compared with that of the SCADA could be due to the model of supply used. These

supplies may be modelled as a reservoir with constant level from which flow is coming. The reality is often that water comes from another network with no constant pressure. A solution is to substitute the reservoir with a tank with curve varying level equal to that at the end of the booster that has always pressure logger. If the inflow is not yet equal to that in the SCADA the problem must be in the network or may be caused by other supplies that can have a similar modelling error and affect this one.

Example 3.3: The network of figure 3.6 has been modified. In figure 3.12 it has a pump that supplies water from another network. This pump links two independent subnetworks. The modelling of the first subnetwork, the one of figure 3.6, could be intended independently as the other subnetwork could belong to another company.

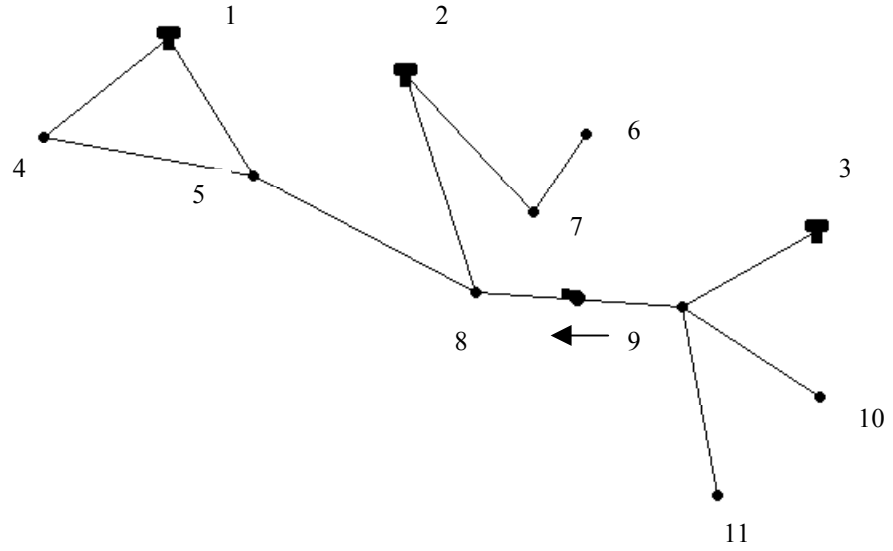


Fig. 3. 12

It could be modelled as figure 3.13 where only one subnetwork is taken into account. The supply is modelled as a reservoir with a constant level.

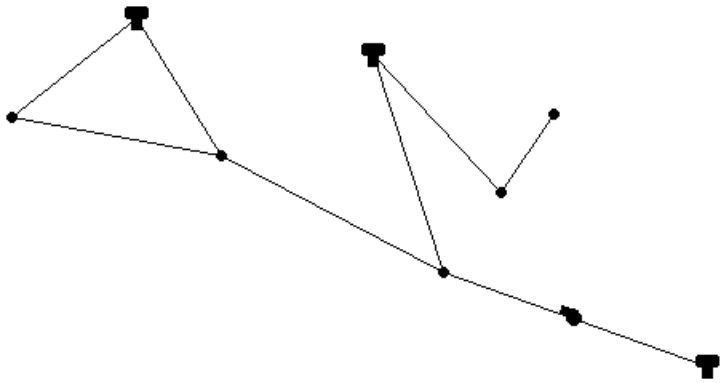


Fig. 3. 13

In figure 3.14 the evolution of the flow through the pump is plotted. In the real network the flow changes throughout time depending on the head of the junction. It decreases as the head of the junction decreases. In the simplified model (dashed line) it remains almost constant, though the pattern is very similar it has no tendency, because the reservoir has a constant level.

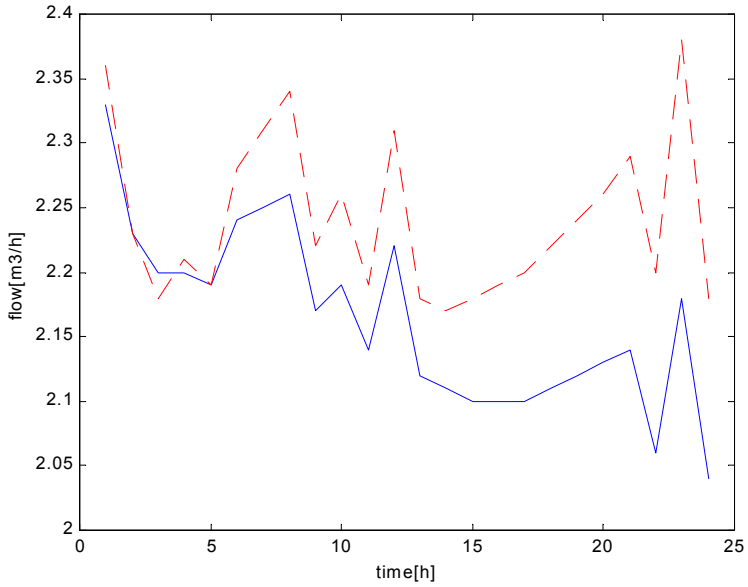


Fig. 3. 14

3.2.4. Overflow in reservoirs

If a reservoir presents an overflow it may be due to an error in typing of size (area). Characteristics of A-valves associated to this reservoir can be the cause as well. These two possible causes should be checked but could be anything in the network. The same should be done for empty reservoirs.

Example 3.4: In case such an error is present in the model it wouldn't allow a correct simulation. Warnings would signal that the problem might be in the area of reservoirs. On figure 3.15 the real evolution of the level in the first reservoir of model 3.6 differs from the evolution given by a model (dashed line) when the diameter of this reservoir is half the original.

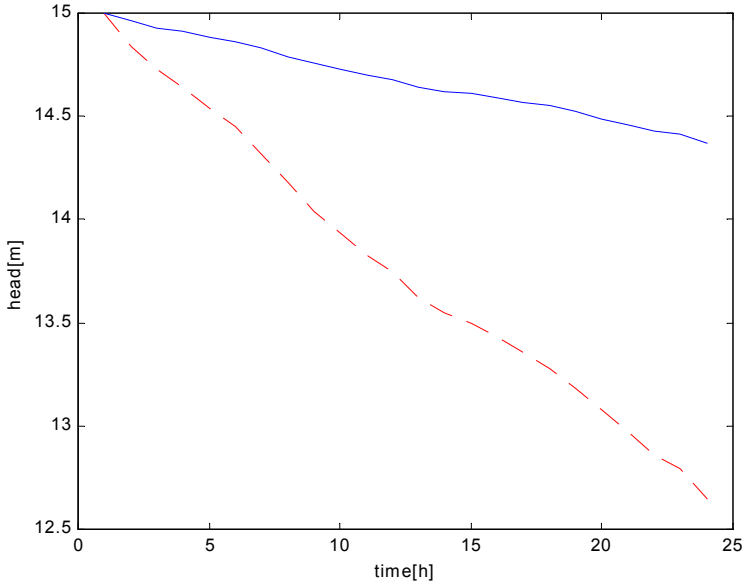


Fig. 3. 15

3.2.5. Pump curves

A fixed-head-pump must accomplish a relation between the head increase and flow that appear in the model as a curve characterised by three parameters (A,B,C). If the SCADA data of a pump does not correspond to this curve what is most likely to happen is that the manufacture's parameters have not been updated. Updating these parameters would consist of adjusting real data to a new curve.

Variable-speed-pumps (vsp) and head-control-pumps (hcp) do not always follow the same curve. Depending on the velocity the relation may vary but some limits would never be violated: maximum flow, negative head loss. Causes are difficult to find out. For hcp, wrong setpoint could be the cause.

Example 3.5: The network of figure 3.12 has a pump with a characteristic described by the curve on figure 3.16 (continuous line). This curve is taken as the real one due to the ageing. A slightly different characteristic could correspond to the original pump, it could be provided by the constructor.

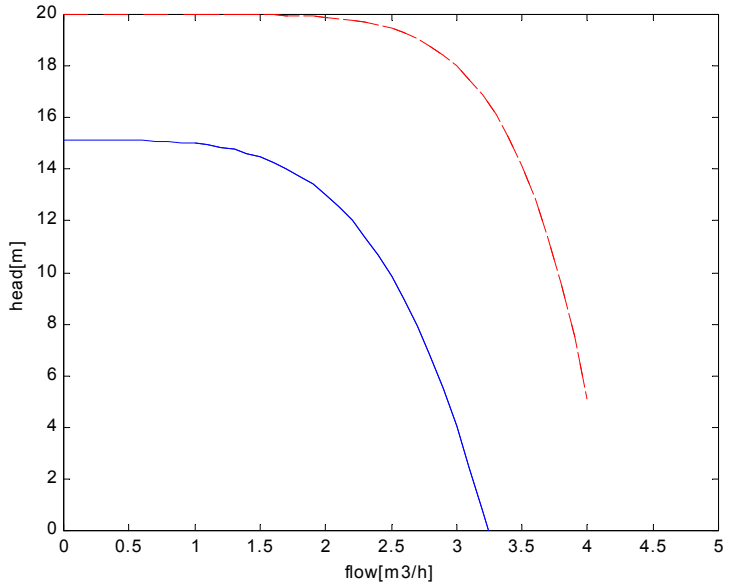


Fig. 3. 16

Figure 3.17 shows the evolution of the flows of the pump, with the real (continuous line) and theoretical (dashed line) pump curves. Figure 3.18 shows the evolution of the head at the end node of the pump. These are the effects of a bad pump curves and thus the pump curves have to be calibrated as time passes and ageing occurs or even poor original information could invalidate the model.

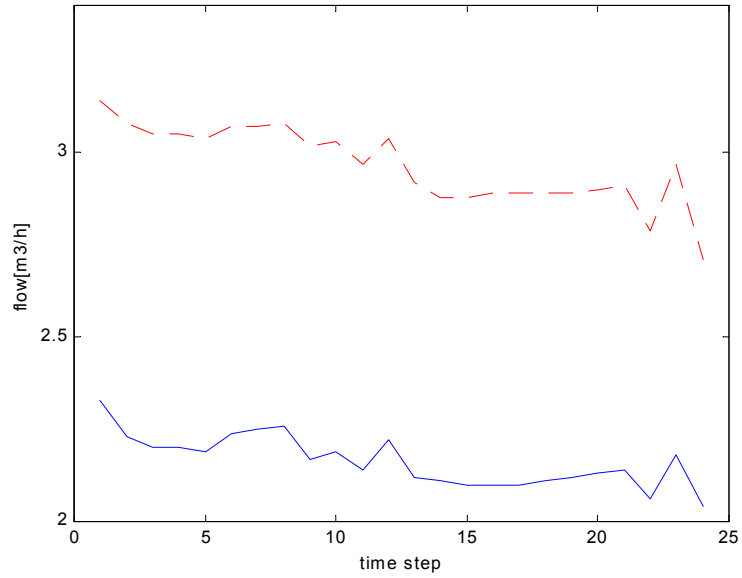


Fig. 3. 17

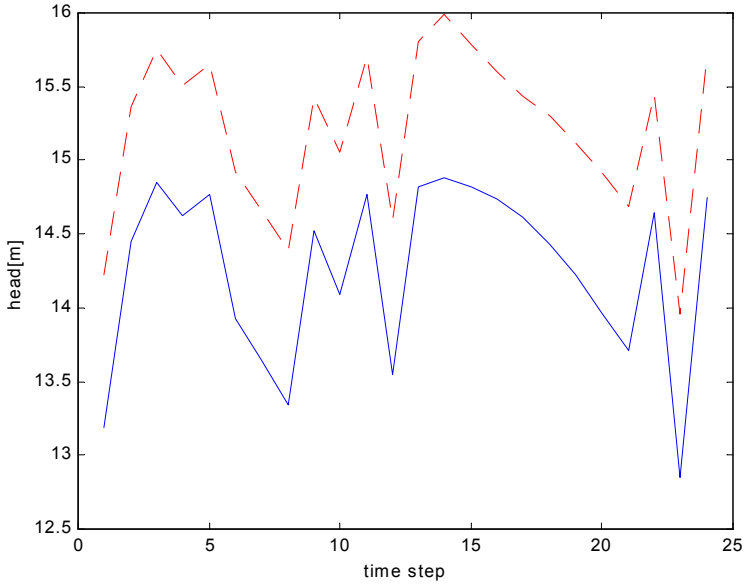


Fig. 3. 18

Even if the pump curve is correct but the speed of this device is variable, the problem can be the working speed. In figure 3.19 the flow of the pump (continuous line) is compared with the flow of the same pump working with 20% less speed. On figure 3.20 the heads of nodes are represented for a time step, here the effect on the origin and end node of the pump can be observed.

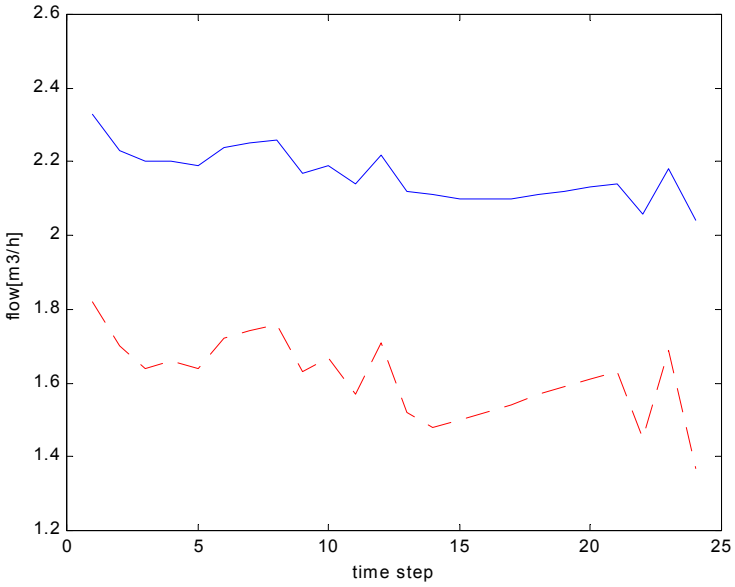


Fig. 3. 19

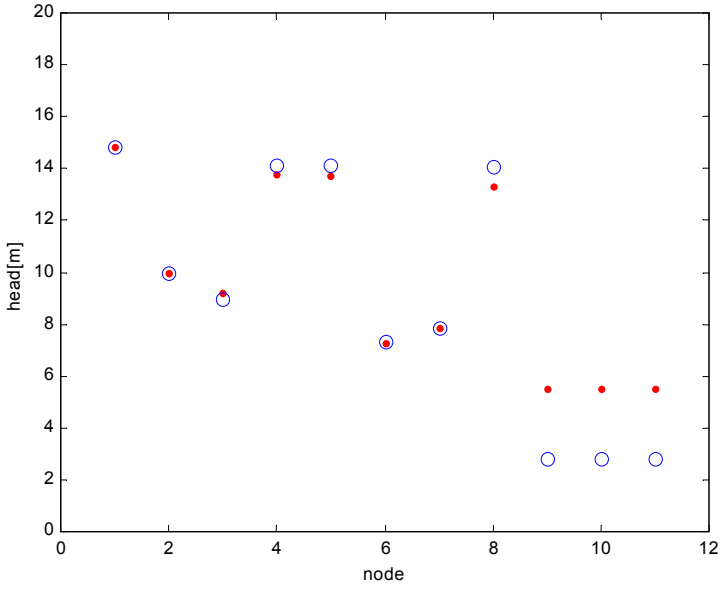


Fig. 3. 20

3.2.6. Pressures

The low and even negative pressures at a node or group of nodes signal a model error. Different causes can be explored.

1. Error in elevation (negative or positive).

2. PRV (Pressure reducing valves) are very often used nowadays in order to control pressure in PMA (pressure management areas). A wrong set point of this valve can cause bad pressures in an area.
3. A more difficult formulation is necessary to take into account the comprehension of the global network behaviour. Where does water come from and where does it go? This is essential to detect disconnections that are not suspected. A disconnection close to a main source should be suspicious.

Example 3.6: Network on figure 3.21 has a PRV. On figure 3.22 the heads of all nodes are plotted for four models. The correct model (o) and one with a wrong setting on the valve (*) and one with a wrong topological value at node 5, the ending node of the valve, (+) and a wrong topological value in node 1(·).

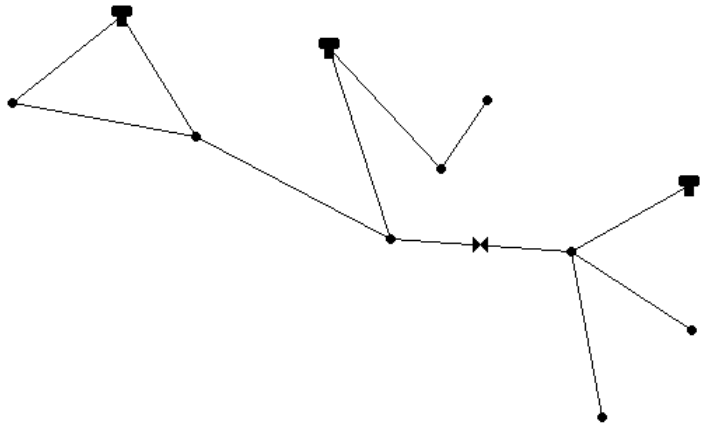


Fig. 3. 21

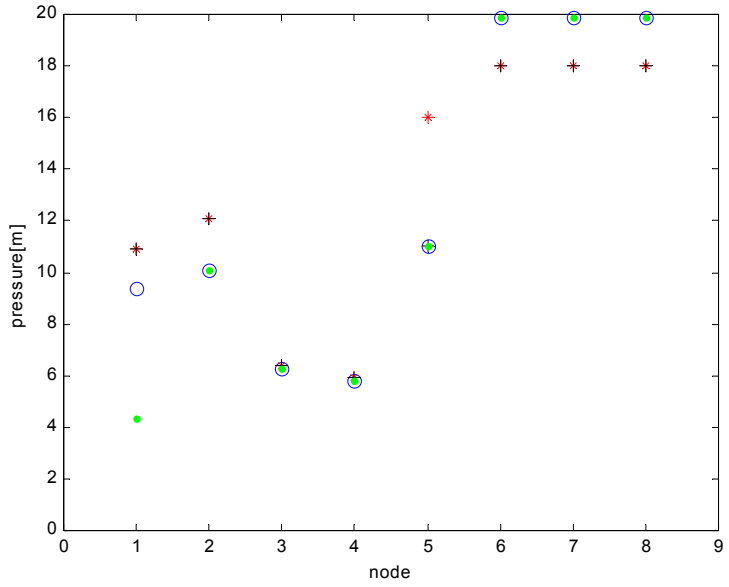


Fig. 3. 22

The PRV assures that node 5 has 11m of pressure and it is wrong only for the model with a wrong setting. If the ending node of the PRV has a wrong topological value it affects the pressure of all nodes in the region (node 3 and 4 are the ones related through a reservoir and thus they are less affected) in a similar way to that due to a wrong setting, but for the node itself. If the topological error is in any node but 5 it affects only this node and all the rest have the correct pressure.

3.2.7. Pressure reducing valves

These elements must be watched in particular. If any of these elements have a zero of flow or head loss the model may have a problem as these elements are introduced with a specific function.

1) It may be reversed. A particular case and one which is more easily found out is that in which the valve goes from a network part to a main supply pipe.
2) The network upstream may be disconnected and have no demand, it should be tested in the original model.
3) A non-existing connection introduced by error in the model between two PMA can be detected as the cause of this. Simplification of the model can help to detect this wrong connections and disconnections.

Example 3.7: Pressures of model in figure 3.21 (o) are compared with the same network with the reversed valve. The results for a time step (t=10) are shown in figure 3.23. It is difficult to find any special characteristic for this error in the model if this figure is compared with 3.22.

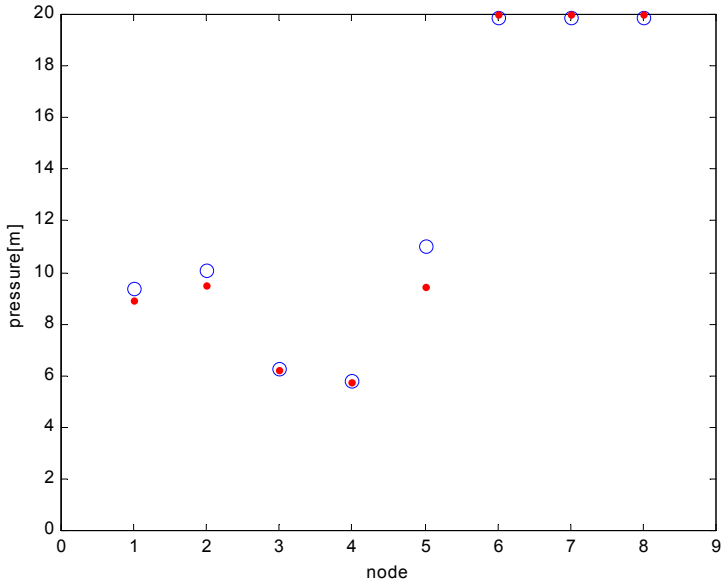


Fig. 3. 23

Taking both nodes (start and end) of the valve, the evolution in time of the head can shed more light on the cause of the mismatch. This is presented in figure 3.24.

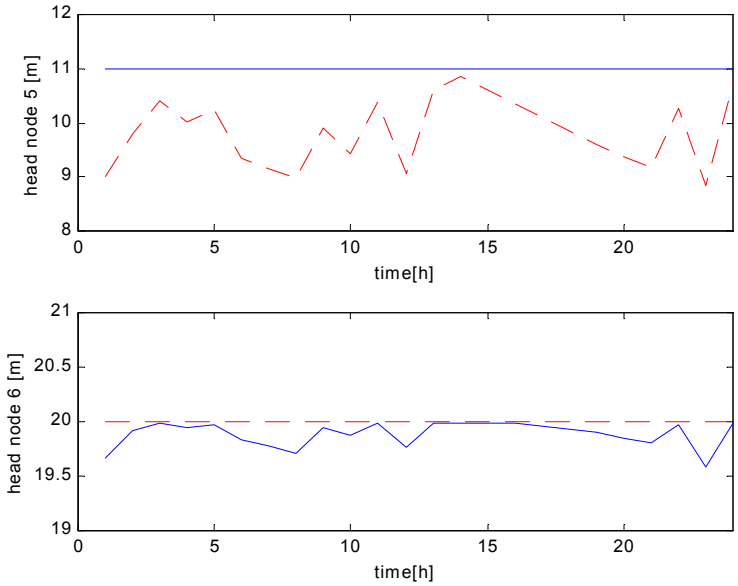


Fig. 3. 24

This work, done in collaboration with experts that develop the macrocalibration manually and presented using simple examples, shows the need of a tool that solves

such tasks as automatically as possible. Even if the decisions are not taken automatically some software has to be developed to help the expert.

3.3. Lamda Classification Method

The goal of classification is the identification of structures in data similar to known structures. Methods developed in the field of soft computing such as fuzzy logic are becoming increasingly popular. Such methods offer an attractive alternative to statistical approaches as they do not require prior assumptions of the statistical properties of the data.

LAMDA (learning algorithm for multivariate data analysis) [Pie-89] is a fuzzy method of conceptual clustering and classification. The LAMDA algorithm computes the degree of adequacy of an object to a class with the partial or marginal information available.

The difference between LAMDA and the classical clustering and classification approaches is that LAMDA models the total indistinguishability or homogeneity inside the feature space from which the information is extracted. This is done by means of a special class, so called non-informative class, NIC, which accepts all objects in the same status. Therefore, the degree of adequacy of the objects of this class acts as a minimum threshold to assign an element to a significant class. The minimum threshold is therefore not fixed arbitrarily, but is automatically determined by the proper context.

Consider a set of objects or situations X (nodes and elements with possible model errors) and a set of attributes of finite cardinal n (discrepancies between measurements and model results, surrounding discrepancies, evolution in time of these discrepancies, etc.). An object is represented by n-component vector $\mathbf{x}$ where $x_j \in A_j$ is the value taken by j^{th} descriptor of that object. The attributes can be of qualitative or quantitative type (we are using only quantitative).

All the concepts presented now are illustrated in example 3.8. To make possible a direct correspondence between objects and classes, the latter must be described with reference to the same attribute used for observations. Given an object $\mathbf{x}$ and a class $C_i \in C$, LAMDA computes for every attribute the so-called marginal adequacy degree MAD_{ij}: $Aj \times C_i \rightarrow [0,1]$ between the value that the attribute j takes over $\mathbf{x}$ and the value that the attribute takes over C_i. Thus a MAD vector can be associated with object $\mathbf{x}$. This vector has a number of components equal to the number of attributes. MAD is a membership function derived from a fuzzy generalisation of a binomial probability law assuming that $x_j \in [0,1]$:

$$MAD_{ji} = \rho_{ji}^{x_j}(1-\rho_{ji})^{1-x_j} \tag{3.1}$$

where $\rho_{ji} \in (0,1)$ is the possibility of the object **x** to belong to class C_i concerning only that attribute. Note that when $\rho_{ji}=0.5$, for every distance function ν, MAD_{ij} is equal to 0.5.Thus, the NIC class is characterised by a MAD vector with $\rho_{jNIC}=0.5$ for every j.

All previously calculated marginal information given by the n MAD_{ji} is used to compute the global adequacy degree (GAD_i) of each class. GAD_i is obtained by summarising that MAD_{ij} through a logical aggregation operator L. Two properties required for such operators are commutativity and monotonicity. By commutativity we mean that the order in which we index the MAD_{ji} functions does not affect the classification. By monotonicity we mean that as the value of an individual MAD_{ij} increases, so does the GAD_i. To connect all MAD_{ji}, LAMDA uses a type of aggregation operators called Mixed Connectives of linear compensation [Wais-98]. These operations are located within the framework of fuzzy sets theory. In LAMDA the mixed connectives used are derived from the following linear convex combination:

$$L_\alpha = \alpha T(\cdot) + (1-\alpha)N(\cdot) \tag{3.2}$$

where α is a parameter belonging to the unit interval, T(·) is a continuous n-norm and N(·) its dual n-conorm. An example of these two norms is the pair used for this application, the product and the sum minus the product:

$$\begin{aligned} T(a,b) &= a \cdot b \\ N(a,b) &= a + b - a \cdot b \end{aligned} \tag{3.3}$$

In figure 3.25 the classification process is represented in three steps. In the first step for each object the marginal adequacy degrees ($m_{11},..,m_{1n};..;m_{k1},..,m_{kn)}$ are computed from the object respect to all the classes (k) and for the descriptors (n) following equation 3.1. In the second step, the global adequacy degree (g_1, g_2,.., g_k) is computed from the MAD of each class using the mixed connective of the equation 3.2. Finally, in the step 3 the maximum GAD is determined classifying the object to the corresponding class.

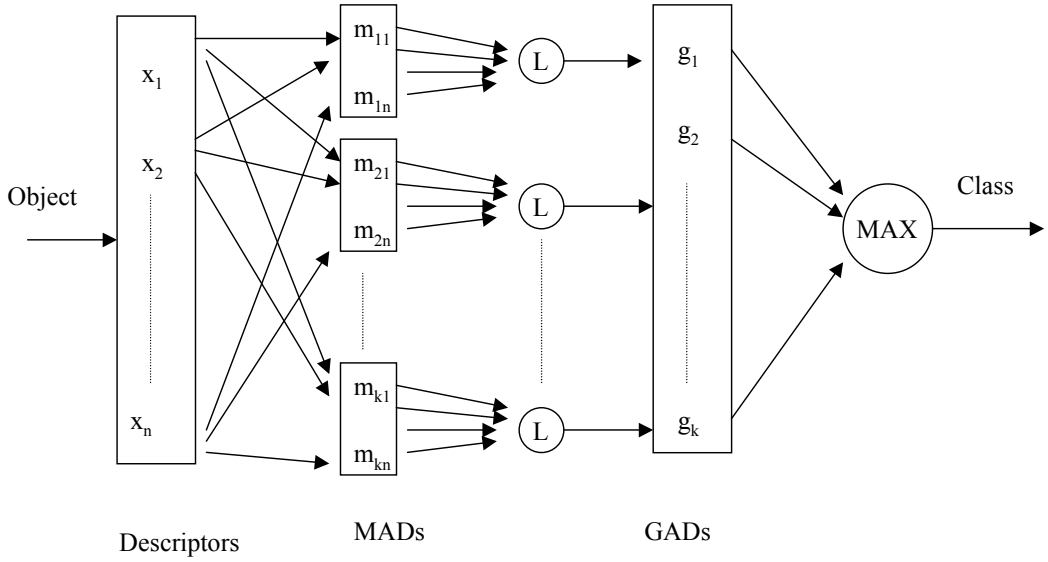


Fig. 3. 25

Data for each class and each attribute are used separately to obtain the corresponding estimate by maximising a likelihood criterion. Consider a subset of X of finite cardinal t where all objects belong to class C_i. Being x^k_j the value that the k^{th} object **x** of the subset takes in the j^{th} attribute. The estimated parameters ρ_{ji} maximise likelihood criteria which represents the belonging to a class of all the cases which are known to belong to it:

$$V(x_j, \rho_i) = \prod_j \rho_i^{x_j} (1-\rho_i)^{1-x_j} \tag{3.4}$$

Where x_j are the descriptors of objects that are known to belong to class defined by ρ_i-coefficients. When an object is associated to one of these classes the actuation is to change the erroneous parameter and recalculate the descriptors in order to reclassify. The representation of a class is expected to vary after a new individual has been assigned to it. For the quantitative descriptors the probability is modified following equation 3.5.

$$\rho'_{kj} = \rho_{kj} + \frac{x_j - \rho_{kj}}{N+1} \tag{3.5}$$

Where N is the number of objects previously assigned to this class.

Two principal properties make LAMDA a convenient classification technique. Firstly, with the NIC class, it is possible to identify in the feature space where there was no sufficient learning data leading the classificator to not being able to determine a class. Secondly, due to the simple learning algorithm and the sequential expression of learning equations, it is possible to adapt the classification parameters on-line.

Example 3.8: In the model of figure 3.6 two errors in the first junction (the one on the left end) are introduced. First one of elevation (topographic head) and the dashed line represents in figure 3.26 the evolution of the pressure in this junction and flows in the pipes connected to it (these have no difference with the correct model, continuous line). The dotted line is the evolution of the same model but with an error in demand, instead of elevation, in the same junction.

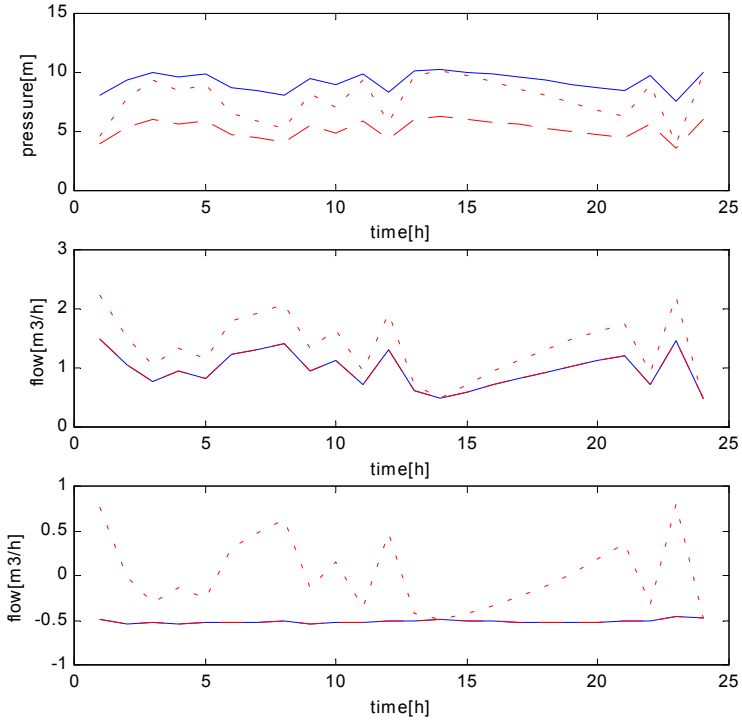


Fig. 3. 26

The head is different for the three cases. So to find one of these two errors just looking at the integral of this error it would be impossible to distinguish between both causes of error. Using the error in flows of pipes associated with this node more information can help because for the topographic error these flows are like the correct model. Let's define as descriptors of errors the integral of the error and whether the flows are correct or not. These descriptors are values between 0 and 1, open set. The integral of the error is normalised using the maximum for all the junctions.

$$x_j = \frac{\sum_{k=0}^{k=t} 0.9^{t-k} \left| h_{0j} - h_j \right|}{\max_i \left(\sum_{k=0}^{k=t} 0.9^{t-k} \left| h_{0i} - h_i \right| \right)} \tag{3.6}$$

The validity of the flows is evaluated calculating the integral of the absolute value of the error in flow for each pipe that is related with each node and adding all these, afterwards it is normalised. Both descriptors use the evolution of the variable, the integral has a forgetting factor in order to give more relevance to the last values. In table 3.1 the descriptors for the first node in each case are presented:

Table 3.1

Class\ρ	Integral of error	Error in flows
No error	0.0001	0.0001
Topographic error	0.9999	0.0001
Demand error	0.9999	0.87

In a straightforward way these values for the erroneous cases could be selected as the descriptors of each class '*topographic error*' and '*demand error*' in figure 3.27 the point on the bidimensional space of both descriptors appears.

Let's take the same network that can have an error (actually there is an error in demand in junction 2). The descriptors for node 1 are evaluated; marginal adequacy degree (MAD) is calculated for each attribute and the general adequacy degree (GAD) comes from these MAD. In figure 3.27 the three points (o) represent the centre of each class localised by their ρ (table 3.1). For each junction (object) the descriptors have been calculated (table 3.2) and represented (* for those of NIC and + for that with error in demand).

Table 3.2

Junction\descriptor	Integral of error	Error in flows
1	0.6696	0.6510
2	1	0.8510
3	0.0508	0
4	0.0517	0
5	0.3976	1

With these descriptors and applying equation 3.1 and 3.2 the marginal and the global adequacy degree is calculated (α=0.9). In table 3.3 the marginal degrees are presented for junction 2. In table 3.4 the general adequacy degree for the 5 junctions are presented

Table 3.3

class \marginal adequacy degree	MAD_{21}	MAD_{22}
NIC	0.5	0.5
Topographic error	0.9999	0.0004

Demand error	0.9999	0.6554

Table 3.4

junction \global adequacy degree	GAD_1	GAD_2	GAD_3
1	0.3	0.0051	0.0667
2	0.3	0.1003	0.6898
3	0.3	0.1001	0.0140
4	0.3	0.1001	0.0140
5	0.3	0.0004	0.0881

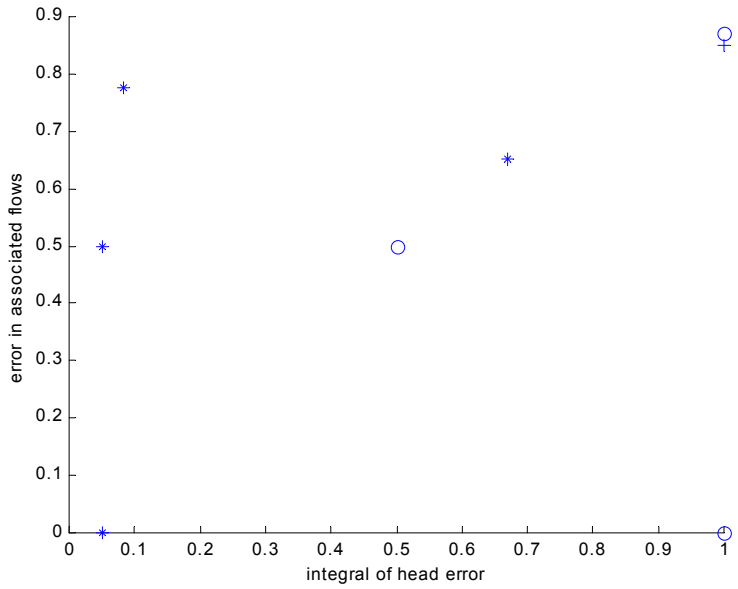


Fig. 3. 27

Junction 2 is the one closest to the class 3, a demand error. Using equation 3.5 the value of ρ is recalculated. This value has changed for the third class after classifying one object in it: ρ_3=(1.0000 0.8605).

There a grid of points has been classified in order to have a graphical idea of the classification areas. On figure 3.28 the biggest area corresponds to NIC (points represented by o), around the two errors some points are classified (* for the topographic and + for the demand). The use of two descriptors allows the representation on two dimensions. When the number of descriptors increases so does the dimension of the space.

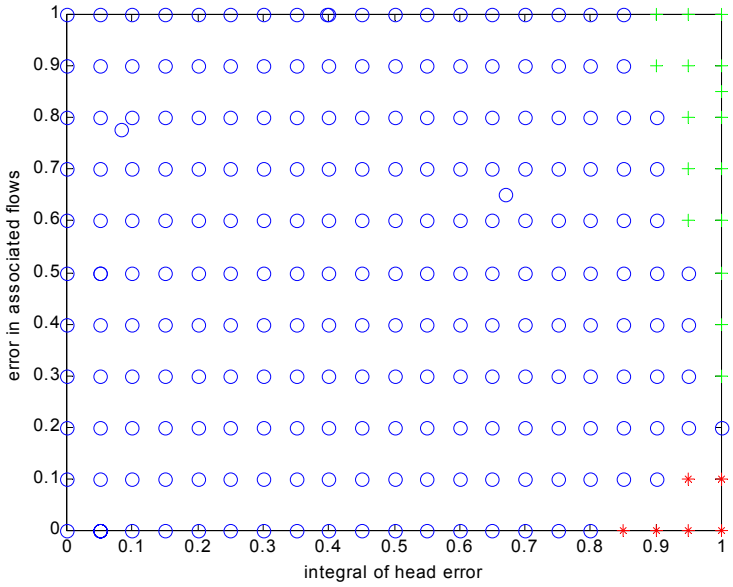


Fig. 3. 28

Parameter α changes this classification. If its value decreases the classification areas increases in expense of NIC. On figure 3.29 the same classification is done with α=0.5

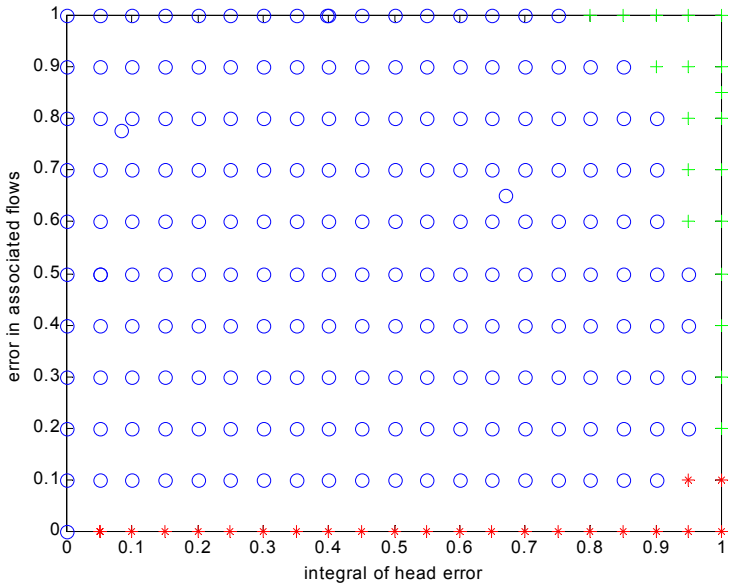


Fig. 3. 29

Generation of parameters ρ for third class is done minimising the likelihood function of equation 3.4 where the objects x are the nodes of the used network where a demand

error has been forced. This has been done with 5 cases and the result is a different parameter than those used so far that came from one case.
ρ=(0.9999 0.7727)

3.4. Macrocalibration Process

The errors presented in section 3.2 are the ones that should be detected in order to improve the model before the calibration process goes on. The macrocalibration procedure proposed here would take into account all these situations. Classification methods, and particularly LAMDA algorithms, have shown to be useful to generating and organising knowledge. This algorithm is used in some of the steps in the macrocalibration process. For some specific steps a classical search algorithm is enough and more suitable.

The classification methods are applied to each element in the network. So it has been used to detect errors related to a specific element. A missing pipe is not detected by this tool, unless it causes a disconnection, this was the expert methodology presented in section 3.2. The knowledge is based in the integral of errors (head, flows,...) in the element and its neighbours, derivatives (sum of the variation in time of the error), parity (whether the signs of two errors are the same, e.g. head in a junction and flows in pipes related to it) , etc. A forgetting factor is used in their calculation in order to weigh the last values (equation 3.6). This knowledge will be different depending on the measurements available. If an element has no meter the descriptors used will not include its own variables but from nearby elements some information can be induced. The generation of this knowledge based on optimisation allows the customisation. Of course the lack of sensors makes the detection poorer.

In figure 3.30 the 7 steps of the proposed procedure are presented in the order suggested by experts. They are developed later in this section. The methodology described by figure 3.1 is the one that is used in each step. When one of the error detectors gives a positive response (demand pattern differences between model and real system, for example) the user is asked to analyse that specific part of the model, improve it and start the macrocalibration again. Nevertheless the process can be executed until the end before any improvement is introduced in order to decide which of all the errors detected is more dramatic. Numerical data and graphic representation is given in each step in order to allow decision-making.

The prototype developed is based on Matlab in order to use the optimisation toolbox in the knowledge generation process. For this generation and for the test two simulation models have been used. One is supposed to be correct (in further work data will come from SCADA system) and the other one has some errors that should be detected in this process. Simulation has been carried out using EPANET.

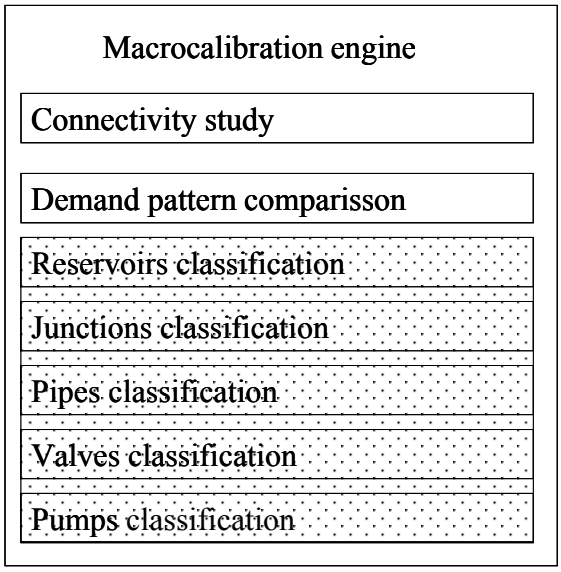


Fig. 3. 30

3.4.1. Connectivity Study

Using output and input files for the simulation model, an incidence matrix is generated. With this matrix and knowing the number of reservoirs, junctions, pipes, valves and pumps the network is easily analysed. This functionality gives the number of disconnected networks along with the nodes and elements that each one contains.

Example 3.9: Applying the macrocalibration tool would warn that there are two disconnected subnetworks. It provides information about each subnetwork, three matrices with the reservoirs, junctions and pipes of each and an object with all the incidence matrices.

These matrices show which element is in each subnetwork. Each row represents a subnetwork and includes the number of the element (reservoir in *ress*, junciton in *nods* and pipe in *pips*) so the number of columns is equal to number of each kind of element. Finally each *lambdas* is the incidence matrix for each subnetwork.

$$ress = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad nods = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 & 5 \end{bmatrix} \quad pips = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 & 6 \end{bmatrix}$$

$$lambdas\{1\} = \begin{bmatrix} -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad lambdas\{1\} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

3.4.2. Demand Pattern Comparison

From the same files used before, the demand pattern can be measured if the areas with common patterns are known. This pattern is compared with the real one that can be obtained from the inflows of reservoirs and boundary flows that are normally known.

3.4.3. Classification

In the first attempt all nodes (reservoirs and junctions) and all elements (pipes, valves and pumps) were treated together in two classification steps. It has shown to be easier and more reliable to do it in separate processes so that descriptors can be defined more specifically without increasing the complexity in excess.

Table 3.5

Element	Error
Junction	Topographic Demand factor
Reservoir	Topographic Wrong area
Pipe	Roughness
Valve	Setting Reversed direction
Pump	Curve Speed Variable head at origin

First of all the knowledge has been generated. Some known errors have been introduced in different networks. The discrepancies between the correct model and the erroneous one, together with the knowledge of the error location and using a likelihood function that has to be maximised, allowed the generation of these descriptors. The heuristic part that remains is to define them but the numerical values are generated automatically

following a supervised learning. As example the generation of descriptors for junctions is presented.

Example 3.10: The two possible errors in models treated in this example are a wrong elevation of a junction, and demand associated with a junction. The first step is to define the descriptors associated with junctions. The evolution of the discrepancies in the variables associated with a junction with a known error in demand and the elements surrounding has been analysed. Therefore the conditions that are accomplished by these variables when the known error appears can be decided heuristically. The expert analysis of these evolutions of the discrepancies in different networks and in different conditions has led to the laws below. This language is used here because it is more understandable however it will be translated into descriptors.

If *there is an important error in reservoir level*
And *this error is not constant*
And *this error has a definite (positive or negative) derivative*
And *there is error in the flows of adjacent elements*
And *the parity of these errors and the error in level is opposite*
And *there is error in the head of nodes related by these elements*
And *the parity of these errors and the error in level is the same*
Then *may be an error in the demand*

The same is done for an error in a topographic head. The discrepancies that are taken into account are the same, so the laws, and therefore the descriptors will be the same.

If *there is an important error in node head*
And *this error is almost constant*
And *this error has a no definite (positive nor negative) derivative*
And *there is no error in the flows of adjacent elements*
And *the parity of these errors and the error in level is not opposite and not the same*
And *there is no error in the head of nodes related by these elements*
And *the parity of these errors and the error in level is not opposite and not the same*
Then *may be an error in topographic head*

There are 7 descriptors that represent this that take values between 0 and 1 depending on the accomplishment of each statement. The quantification of the ρ-parameters of a class takes the place of the thresholds in the expert system. The maximisation of likelihood function (equation 3.4) gives these parameters automatically.

Table 3.6 shows the ρ-coefficients for each descriptor and class including the NIC. This is the translation of the laws presented into classes. These have been obtained by experiments comparing simulations of correct and incorrect models where the errors were known. These wrong models were generated with quantities in the errors, topology, elevations and parameters values.

Table 3.6

Descriptor	Topographic error	Demand error	NIC
Integral absolute error (low, indifferent, great)	0.9999	0.8750	0.50
Error is constant (no, indifferent, yes)	0.9999	0.5568	0.50
Definite parity in derivative (no, indifferent, yes)	0.50	0.5625	0.50
Error in flows (no, indifferent, yes)	0.0001	0.2573	0.50
Parity of errors with flows (opposite, indifferent, same)	0.5000	0.0748	0.50
Error in heads (no, indifferent, yes)	0.2639	0.7018	0.50
Parity of errors with heads/levels (opposite, indifferent, same)	0.7500	0.9361	0.50

Using these descriptors for the classification of errors in network of figure 3.6 the result is shown in figure 3.31. Here the five junctions are presented with the integral of error in their heads without normalising it (the main variable related to these elements) and using different colours for each class. Nodes 1, 3 and 5 have been classified in the NIC (o), second node in demand error class (+) and the fourth in topological error class (*). It has worked properly even with two errors in the same network. Sometimes one error can hide other smaller errors that can be detected after solving the main ones following figure 3.2.

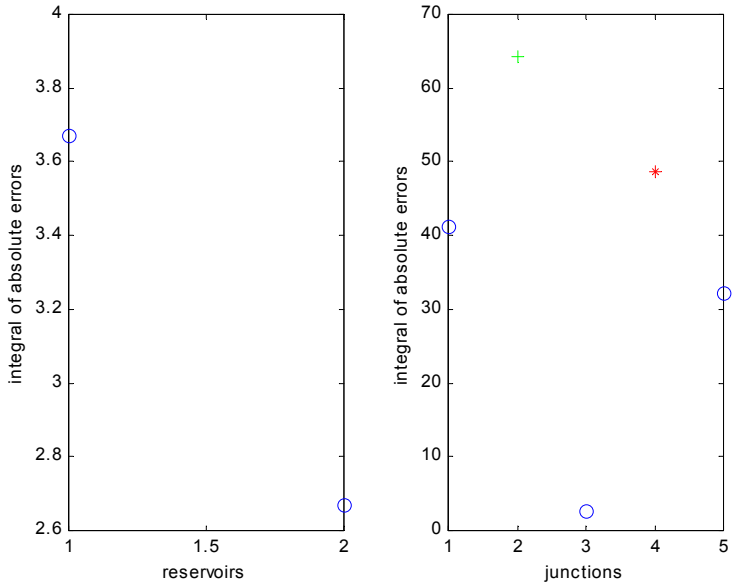


Fig. 3. 31

This process is repeated for reservoirs (topographic and area error), pipes (roughness error), valves (setting error and reversed position) and pumps (curve error, variable head at the origin). Network characteristics are sometimes critical for the descriptors and specific generation of descriptors might be necessary. During the classification process the descriptors are modified so that the knowledge is improved with their use.

Example 3.11: In the previous section some examples have been used to illustrate the errors that we intend to solve in this calibration step. Some of these errors will be treated using LAMDA classification algorithms. Here these examples will illustrate the functionality of this algorithm.

In example 3.4 the erroneous area of the first reservoir implied a wrong simulation of its level but it affected not only this component of the network. In the classification process (figure 3.32) the behaviour of the first junction is associated to a demand error (+) together with the classification of the reservoir in the class of erroneous area (+). This makes clear the need of a feedback after any change in the network. Of course if the reservoir area is checked and the model improved both simulations will agree in the results. But if the demand is changed in junction 1 the simulation of the new model will not improve and the process shown in figure 3.1 avoids introducing new errors instead of correcting the ones that exist.

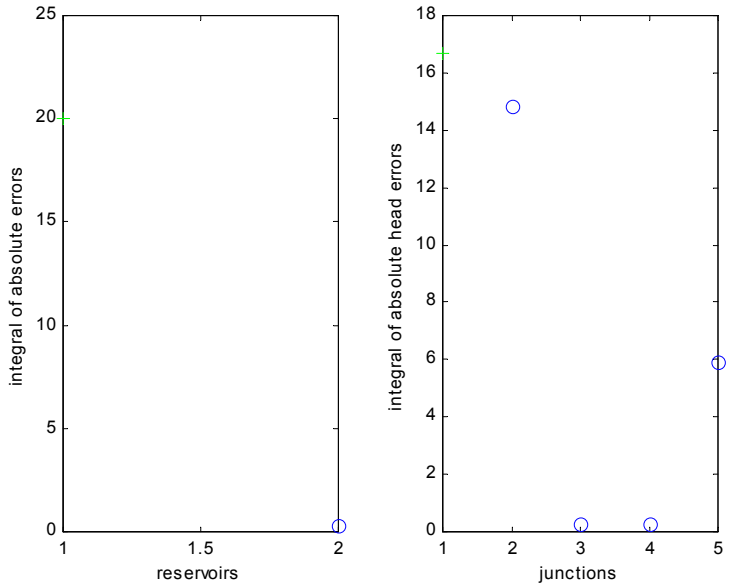


Fig. 3. 32

In example 3.5 the pump had an erroneous curve and it can be detected using the classification (figure 3.33). All pipes are classified in the NIC class even those that have an important error in their flows. The only pump appears classified with an error in its curve (*).

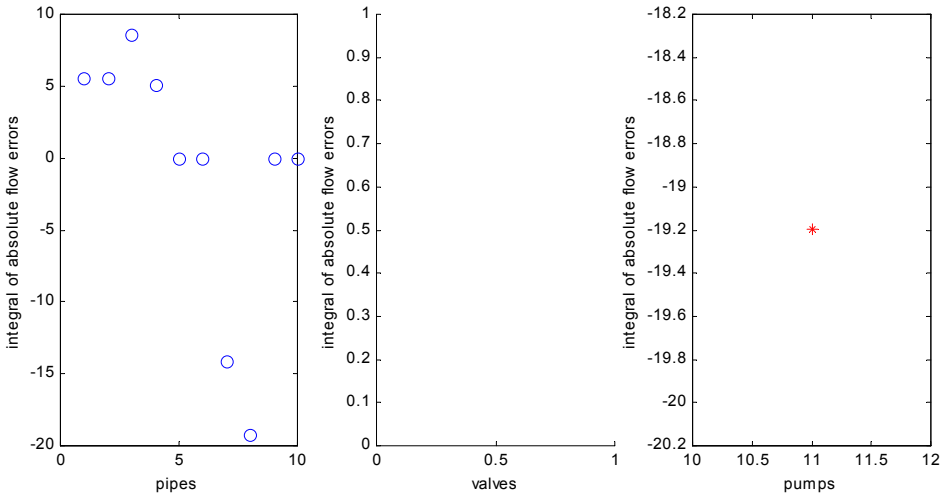


Fig. 3. 33

The behaviour of the model with an error in speed in the pump is very similar to the behaviour when the curve is erroneous. No different classes are generated for both errors and both errors are classified in the same class. When one of these errors is found the cause can be checked, as the speed is information that is available and if it is correct the curve has to be recalculated. On figure 3.34 the pump with erroneous (example 3.6) speed is classified (*).

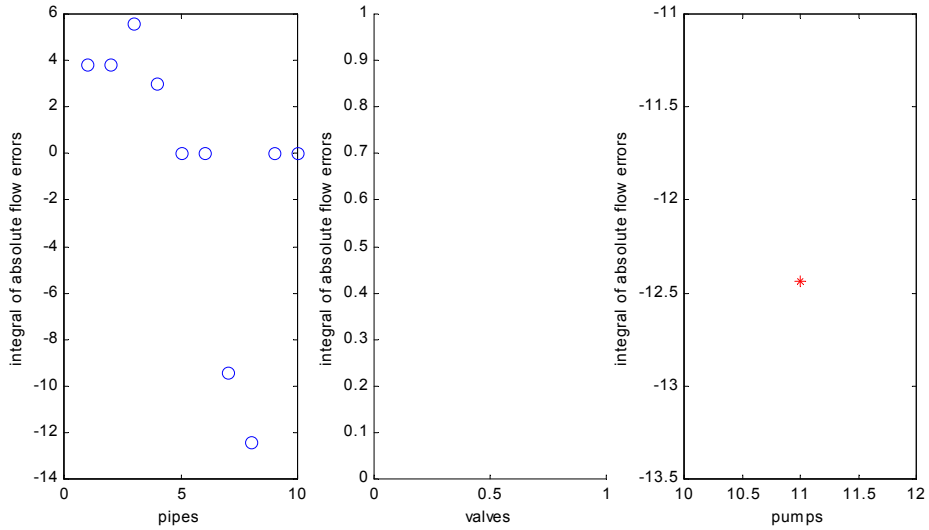


Fig. 3. 34

In example 3.7 a reversed valve, is shown to affect heads of nodes, which are perfectly modelled. With classification this node appears in NIC class. In figure 3.35 the valve is classified in the class of reversed valves (+). In figure 3.36 the same system is classified

but in this case the valve has a wrong setting instead of being reversed and it is well classified (*) in the corresponding class.

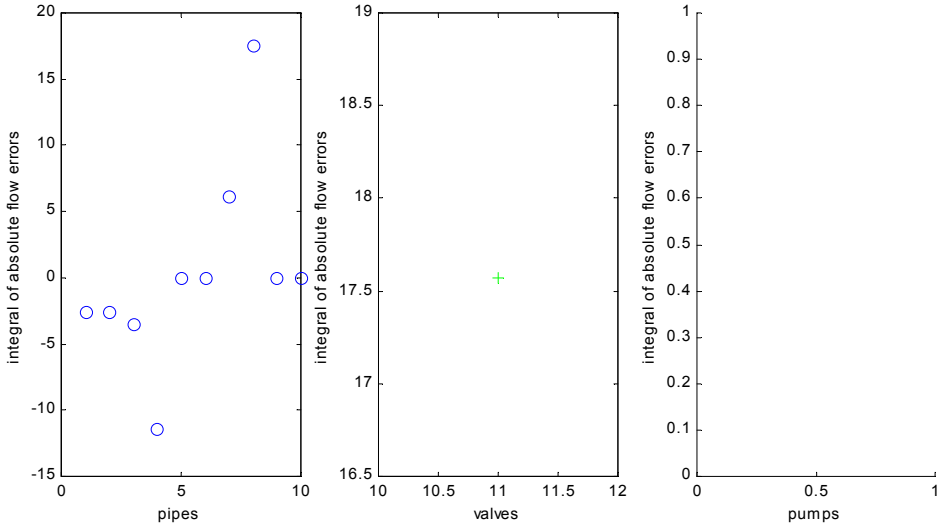


Fig. 3. 35

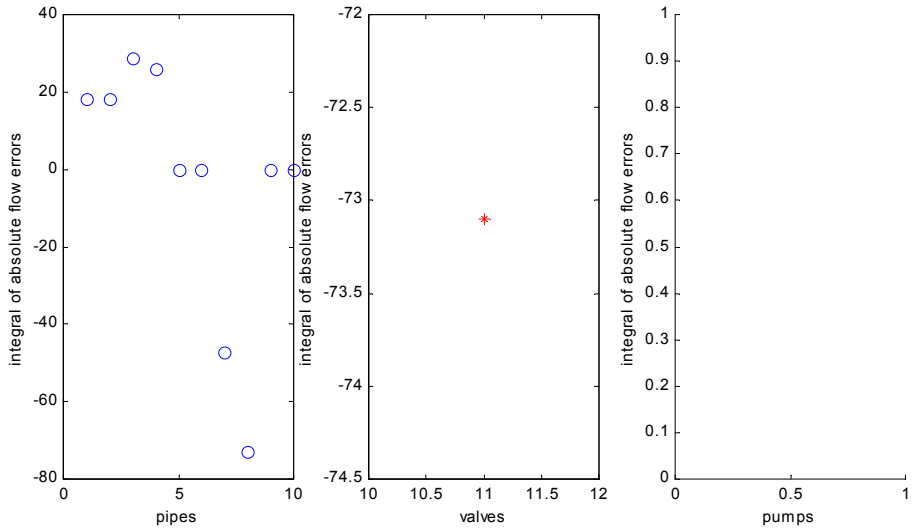


Fig. 3. 36

3.5. Case Study

The network as shown in section 2.9 with some modifications (introduction of valves and pumps, figure 3.37) for illustrating the identifiability study and the identification problem design is used now for illustrating the macrocalibration process. Two simulation models are used. The first one is the original. The one used in section 2.9 is

used as the real system and the results of simulation are compared with those of the second model in order to improve it. A second model is generated from the original introducing some errors. Such errors can easily appear in the typing process or even due to the use of an old model. The macrocalibration process described so far will detect some of these errors. Some are small mismatches of roughness and demand factors that are left to the last step of the calibration process.

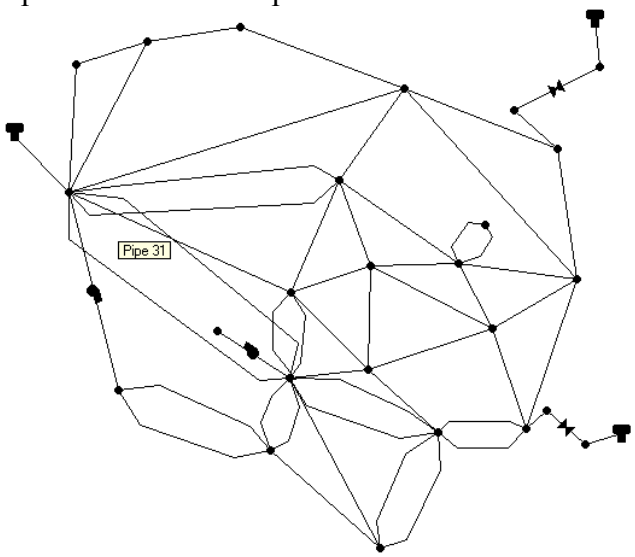


Fig. 3. 37

The erroneous model is constructed introducing mismatches in the original model. All demand factors are rounded to an integer. The roughness of pipes has an error within 5% of its value. Both types of errors can be produced by the use of an old model where the change of demand behaviour and the ageing of pipes have not been taken into account. All these mismatches will not be detected by the macrocalibration, or at least it is not the aim of this step.

An erroneous demand pattern substitutes the correct one. Such a pattern can have changed after the model has been used for some time. Other great errors are introduced: great roughness error, great demand factor error, erroneous valve setting, reversed valve, erroneous pump curve, erroneous pump speed, erroneous topographic head, erroneous area. All these errors are more likely to be a consequence of mistyping in the process of generating the model. Their nature makes them susceptible of being detected by the macrocalibration process.

The macrocalibration process as presented in figure 3.30 has been developed in a Matlab environment. Castells [Cas-01] programmed it for his project directed by the author. This environment allows the use of such techniques through the network using the Matlab server toolbox. This work is now improved in order to integrate all the steps

of the calibration presented in this work so that the use of such tool or at least the demonstration can be done in the easiest way.

A first analysis of the model compared with the measurements (a correct model in this case) detects differences in heads and flows (figures 3.38 and 3.39). In figure 3.40 can be seen that the total demand pattern does not fit with the real one. Adjusting this pattern is the first improvement to be made and the errors are reduced.

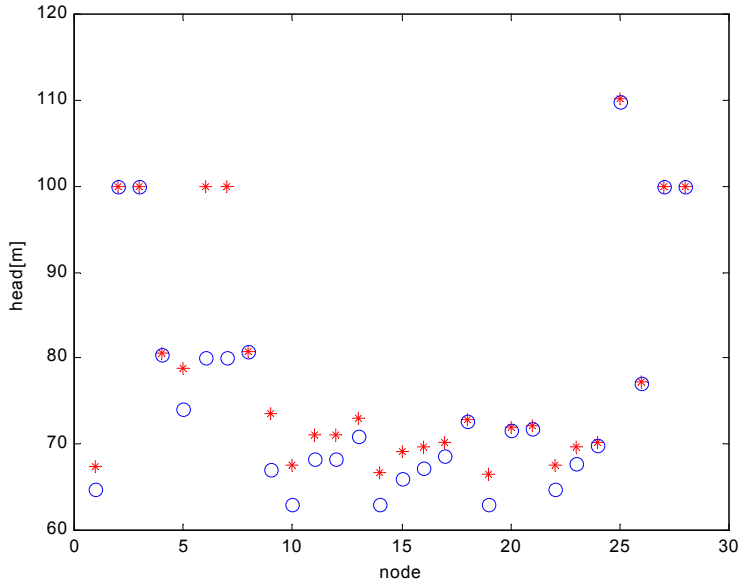


Fig. 3. 38

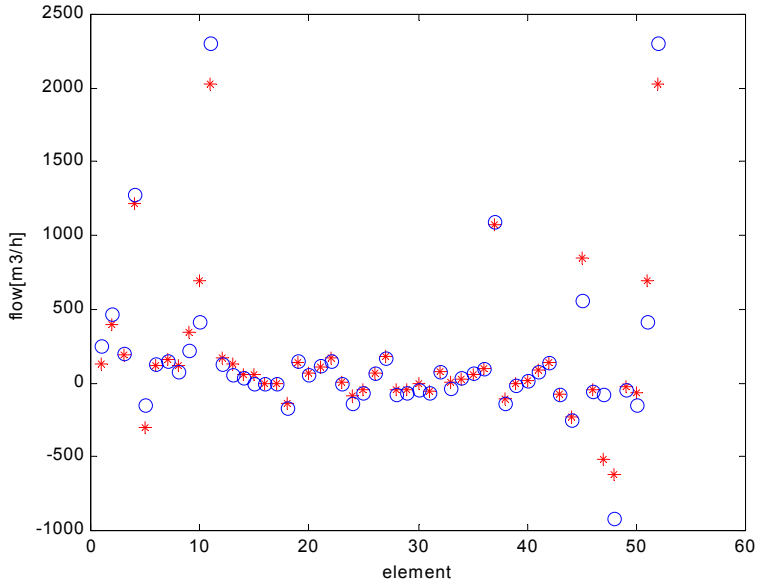


Fig. 3. 39

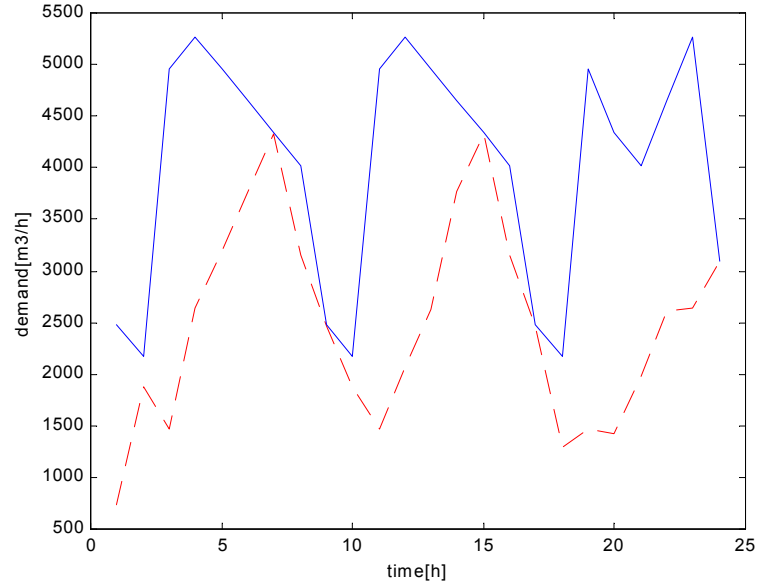


Fig. 3. 40

The classification done after this first adjust signals some errors. In figure 3.41 area error is signalled in reservoir 1. It had a diameter 10 times smaller. Such an error could be a typing error and can be solved in the study after this classification. Other errors of demand are checked and are not due to one big error so they should be solved in the

next step, microcalibration. On figure 3.42 two other errors are detected; a roughness of pipe 51 is signalled and the setting of valve 1.

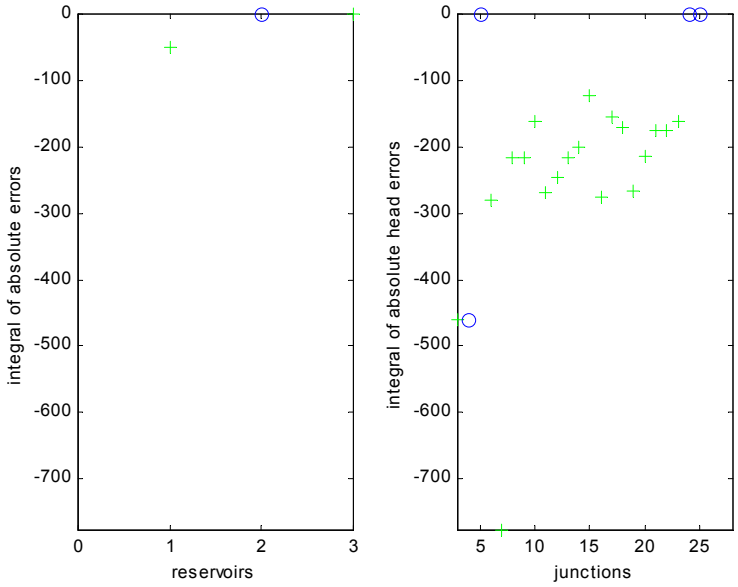


Fig. 3. 41

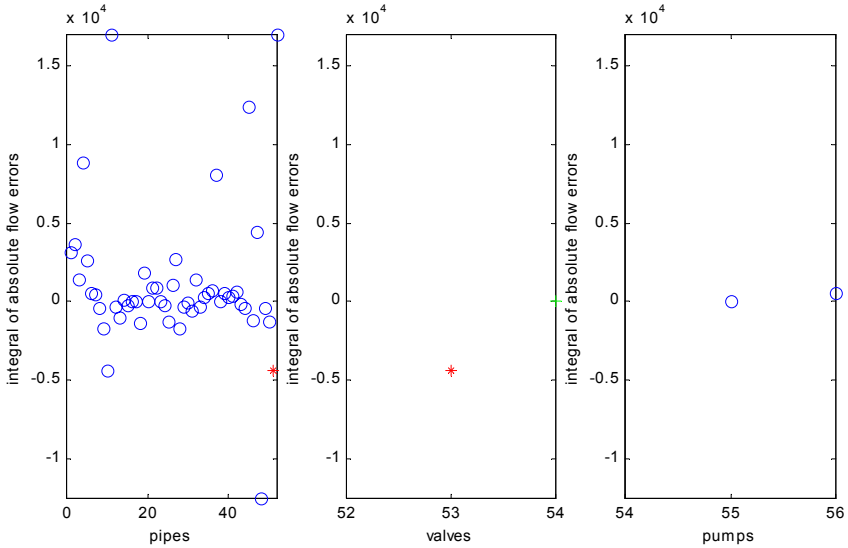


Fig. 3. 42

Once these three errors have been corrected a new classification is carried out. Here two new errors are detected a topographic error in node 7 and a curve error in the second pump. Once these errors are corrected no more big errors are detected. The real and simulated heads and flows appear on figure 3.45 and 3.46. They have improved and the most important fact is that this model is ready for the microcalibration process.

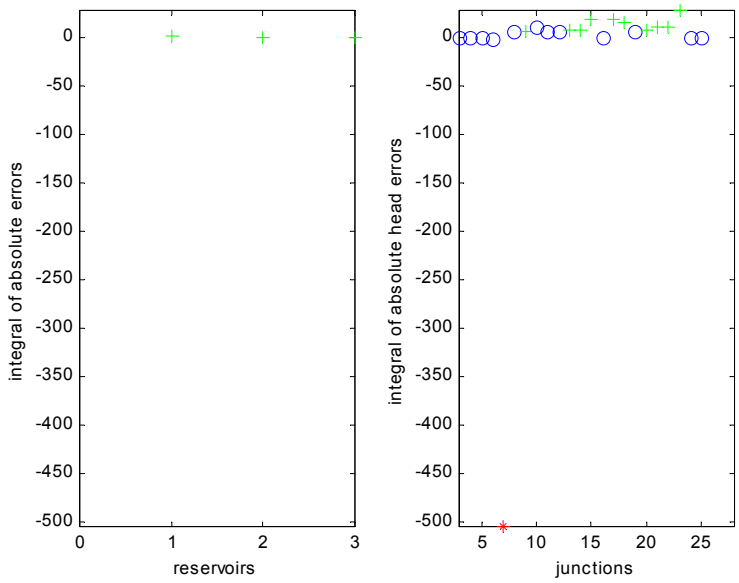


Fig. 3. 43

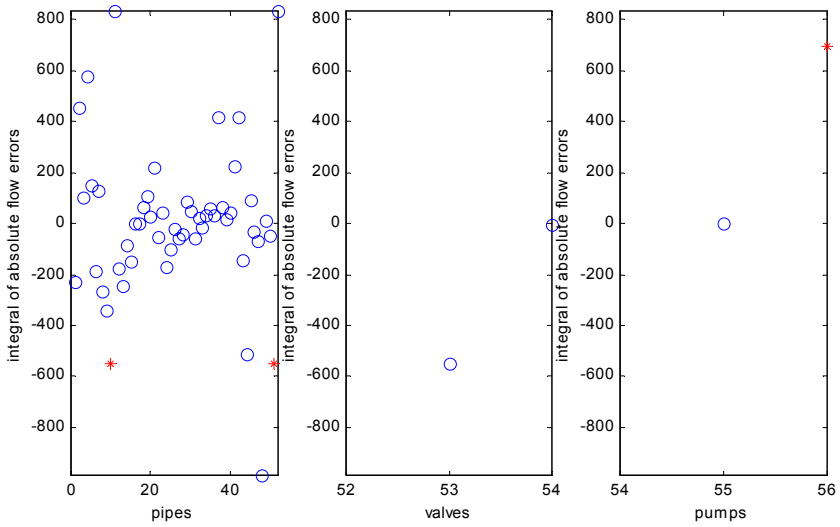


Fig. 3. 44

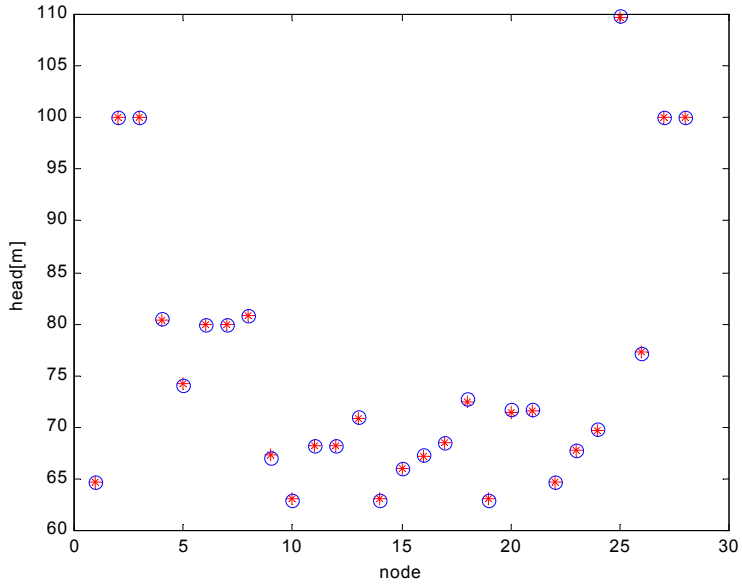


Fig. 3. 45

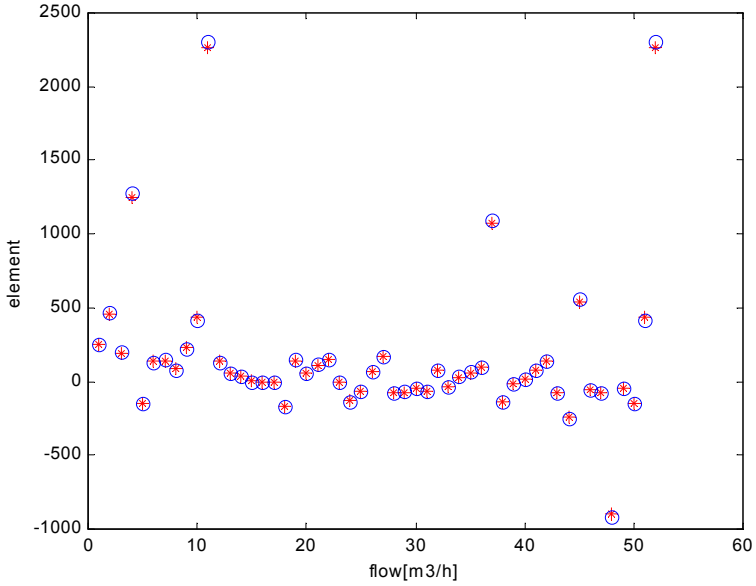


Fig. 3. 46

4. Microcalibration of water networks

The Macrocalibration process can detect some errors in the model as have been shown in the previous chapter. These are big errors, the ones that are not solved by tuning some parameters as their existence is unexpected. But there are other errors in the model. Generally they are less dramatic but more widespread and abundant. These little mismatches with the reality are treated in identification problems as the variables of an optimisation problem because of the over-estimation in the equations. The problem tries to find the set of parameters that give a simulation closer to reality. The reality provided by the measurements of levels, heads and flows together with some parameters that are assumed to be correct as the area of reservoirs, demand curves, length and diameter of the pipes are the inputs of the problem. The formulation comes from the known topology of the network and the laws of physics. The outputs of the optimisation are variables such as pipe roughness and demand factors. This idea is shown by figure 4.1.

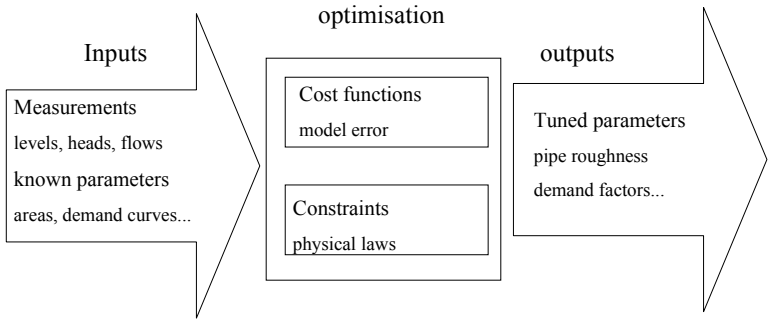


Fig. 4. 1

The first step is to formulate this problem. Formulation is not unique as will be seen in the next section. A better formulation can have better numerical behaviour but the characteristics of the problem do not change. These characteristics are studied in section 4.3 after a brief overview of the optimisation problem classification presented in section 4.1. In further sections some algorithms that have been tested in this application are described and numerical results are used to decide the best option. A real network with real data is used to see try the application of the methodology of this thesis.

4.1. Optimisation problems classification

A general **Optimisation Problem** for minimisation is defined as follows.

Given a Set D and a function $f : D \to P$, *find at least a point* $\mathbf{x}^* \in D$ *that satisfies* $f(\mathbf{x}^*) \le f(\mathbf{x}) \forall x \in D$, *or demonstrate that such a point does not exist.*

A possible mathematical formulation of the minimisation is:

$$\begin{aligned} &min \quad f(\mathbf{x}) \\ &subject\ to \quad \mathbf{x} \in D \end{aligned} \tag{4.1}$$

In this formulation $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ is a n-dimensional vector of unknown variables. Function *f* is the **objective function** and D is the **feasibility domain** of *x* as specified by the **constraints.**

A vector $\boldsymbol{x}^* \le D$ that satisfies $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$ for all $\boldsymbol{x}^* \in D$ is called a **global minimiser** of *f* over *D* and the value of $f(\boldsymbol{x}^*)$ is called the **global minimum.** A vector $\boldsymbol{x}^* \in D$ is called a **local minimiser** of *f* over *D* if $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$ for all $\boldsymbol{x}^* \in D$ close to **x***. The value of *f* at this point is called **local minimum**. Maximisation problems can be formulated as minimisation ones just using $max\ f(D) = -min(-f(D))$.

The optimisation problems can be classified depending on the mathematical characteristics of the objective function and constraints [Pui-99]. Such a classification is presented in figure 4.2 with special attention to the problem that is examined in this thesis.

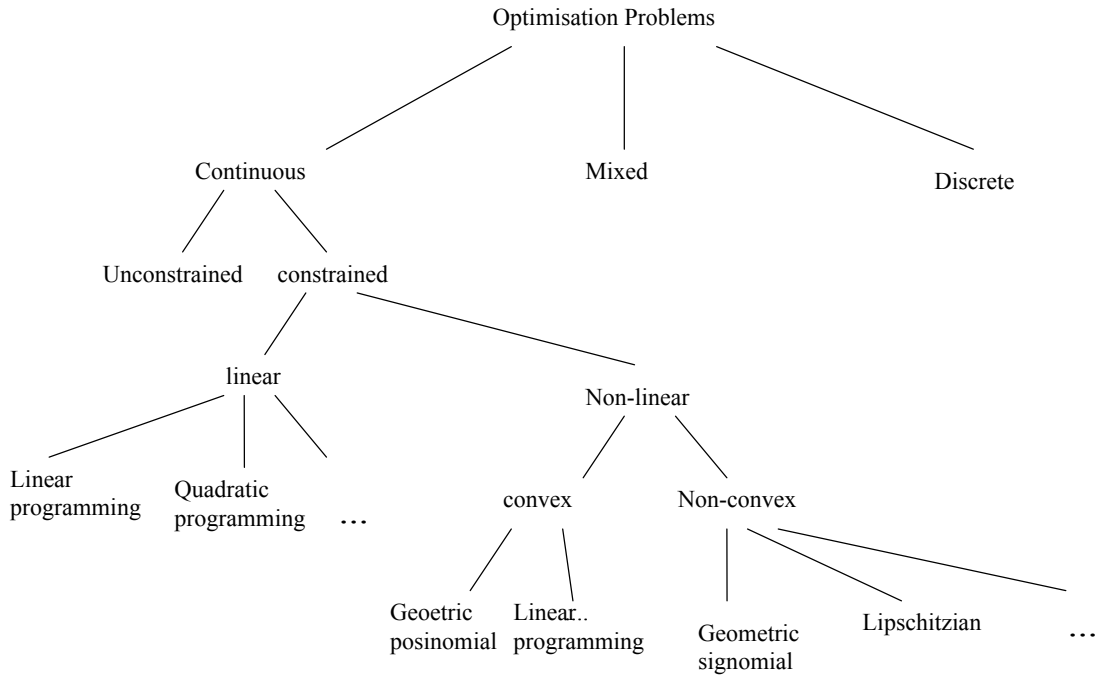


Fig. 4. 2

Optimisation problems are classified as ***continuous*** if the unknown variables take real and continuous values, that means that *D* is the domain of the real numbers. A problem is ***discrete*** if its variables take discrete values, normally integers and that is why they are known sometimes by ***integer problems***. There are some problems with both types of variables that are called ***mixed***. This is the first classification that appears in the tree. The problem examined in this thesis belongs to the first group. All variables (heads, flows, demand factors, roughness...) take real continuous values.

Constraints are another concept used for the classification. Problems can be ***constrained*** if they have constraints otherwise they are ***unconstrained***. The last ones can be formulated:

$$\begin{aligned} &min \quad f(\mathbf{x}) \\ &subject\ to \quad \mathbf{x} \in R^n \end{aligned} \tag{4.2}$$

The problem treated here has constraints that are the physics of the network. Depending on the form of these constraints it can be classified in ***linear*** or ***non-linear***. If the constraints that define the domain *D* are linear the problem is relatively easy to solve. Included in this type of problems there are two classes that have been exhaustively studied and well solved: ***lineal program*** and ***quadratic program***. If the constraints and the objective function are lineal the problem can be formulated:

$$\begin{aligned} &min \quad \sum_{i=1}^{n} c_i x_i \\ &subject\ to \quad \sum_{i=1}^{n} a_{ij} x_i \geq b_j \quad for \quad j = 1,2,\cdots,m \\ &\qquad \mathbf{x} \geq 0 \\ &\qquad \mathbf{x} \in R^n \end{aligned} \tag{4.3}$$

Where c_i, a_{ij} and b_j are constant real coefficients. A linear programming problem has a unique local minimum that is global.

If the objective function *f* is a quadratic function and the constraints are linear it can be formulated:

$$\begin{aligned} &min \quad \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j \\ &subject\ to \quad \sum_{i=1}^{n} a_{ij} x_i \geq b_j \quad for \quad j = 1,2,\cdots,m \\ &\qquad \mathbf{x} \geq 0 \\ &\qquad \mathbf{x} \in R^n \end{aligned} \tag{4.4}$$

Where c_i, a_{ij} and b_j are constant real coefficients.

The problem treated here has a quadratic objective function but constraints are not linear. The ***non-linear programming*** can be formulated as:

$$\begin{aligned} &min \quad f(\mathbf{x}) \\ &subjecto\ to \quad h(\mathbf{x}) = 0 \\ &\qquad g(\mathbf{x}) \leq 0 \\ &\qquad \mathbf{x} \in R^n \end{aligned} \tag{4.5}$$

Where *h(**x**)* represents a set of equality constraints and *g(**x**)* a set of inequality constraints. Problems with non-linear constraints are difficult to solve as they determine feasibility zones difficult to find and a non-linear objective function can have local minima. A problem is ***convex*** if the function *f* and domain *D* are convex, these problems have only a global minimum. All problems that don't satisfy this condition are ***non-convex***. In this last group are included most of the real problems. It is demonstrated later that the case treated here is one.

When the problem is non-convex a new classification is needed from the algorithm point of view. This is showed in figure 4.3.

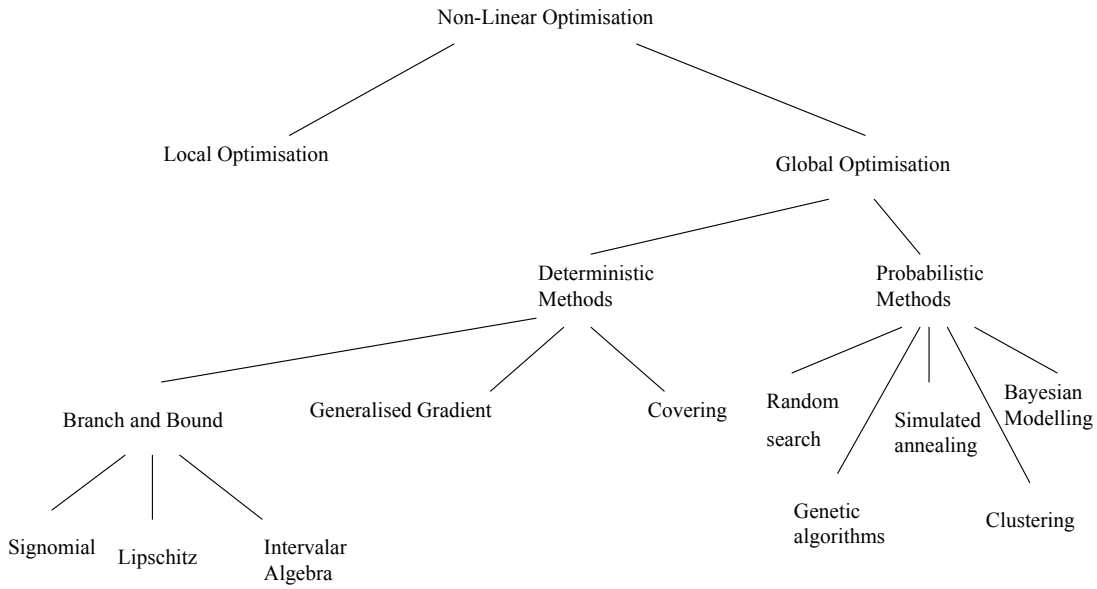


Fig. 4. 3

Now the problem is classified depending on the aim of the optimisation. Here the global minimiser is searched as this is the one that has the parameters of the network. The local optimisation done in this thesis has been used to illustrate the need of a ***global optimiser***.

Deterministic optimisation methods do an exhaustive search of the feasibility domain *D* of the optimisation problem. In order to do this they use different strategies. ***Branch and Bound*** methods find and keep areas that may contain minima while eliminating the ones that cannot contain any. ***Covering*** based methods approach the solution using tighter bounds each time. ***Generalised Gradient*** methods avoid being trapped in a local optimum changing the shape of the function.

4.2. Formulation of the optimisation problem

The aim of this problem is to find a model that simulates the system as well as possible. What has to be minimised is the error between the variables of the model and the measurements. The most frequently used objective function is the quadratic error of the simulation results. It is calculated by comparing the measurements of head and flow with that predicted by the model (equation 4.6), weighted so that the addition has sense. With an ideal model this error should be zero, so the minimisation of this error will lead to an approximate model. The physics of the network are the constraints of the problem; the mass conservation both in nodes and reservoirs (equations 4.7 and 4.8) and head-flow relationship for pipes (equation 4.9).

$$J = \sum_{l=1}^{t}\left[\sum_{i=1}^{r+n}\left(h_{0il} - h_{il}\right)^2 + \rho\sum_{i=1}^{m}\left(q_{0il} - q_{il}\right)^2\right] \tag{4.6}$$

where
h_{il} is head at the node (junction or reservoir) *i* in time step *l* (variable)
h_{0il} is head at the node (junction or reservoir) *i* in time step *l* (measurement)
q_{il} is flow in element *i* in time step *l* (variable)
q_{0il} is flow in element *i* in time step *l* (measurement)
ρ is the weight factor
r is number of reservoirs
n is number of junctions
m is number of elements
t is number of time steps

$$A_i h_{il+1} - A_i h_{il} - \sum_{j=1}^{p} q_{jl} = 0 \; for \; \forall \; \mathrm{i} \le \mathrm{r} \text{ and } \mathrm{l} \le \mathrm{t} \tag{4.7}$$

h_{il} is the level of water in reservoir *i* at time step *l*

q_{jl} are the flows that arrive to reservoir i in time step *l*
p is number of flows that arrive to the reservoir *1* in time step *l*
A_i is the area of reservoir i

$$\sum_{i=1}^{m} \Lambda_{ji} q_{il} - d_{jl} = 0 \;\; for \; \forall \; \mathrm{j} \le \mathrm{n} \text{ and } \mathrm{l} \le \mathrm{t} \tag{4.8}$$

d_{jl} are the demands at junction *j* in time step *l*

$$h_{1il} - h_{2il} - R_i q_{il}^{\,1,852} = 0 \; for \; \forall \; \mathrm{i} \le \mathrm{m} \text{ and } \mathrm{l} \le \mathrm{t} \tag{4.9}$$

h_{1il} and h_{2il} are heads of nodes in pipe *i* instant *l*
R_i is roughness of pipe i
This is not the only way to formulate such a problem. It is the most straightforward but even in this work some modifications have been used in order to show certain properties or to solve it more easily. Using the residual formulation allows one to have an unconstrained problem and is especially useful in the convexity study. In equation 4.10 constraints presented in equations 4.7 to 4.9 have become a part of the cost function. In this equation measured variables are substituted by their value.

$$J = \rho_1 \sum_{\substack{i=1 \\ l=1}}^{\substack{i=r \\ l=t}} \left(A_i h_{il+1} - A_i h_{il} - \sum_{j=1}^{p} q_{jl} \right)^2 + \rho_2 \sum_{\substack{i=1 \\ l=1}}^{\substack{i=n \\ l=t}} \left(\sum_{i=1}^{p} q_{il} - d_{il} \right)^2 + \rho_3 \sum_{\substack{i=1 \\ l=1}}^{\substack{i=m \\ l=t}} (h_{1il} - h_{2il} - R_i q_{il}^{1,852})^2 \tag{4.10}$$

4.3. Problem characterisation

Before the techniques and the solver are selected the problem must be characterised. Both the objective function and the constraints have to be analysed in order to be sure that the solution exists and to use the best technique for this particular problem. The best-behaved optimisation problems are those that are convex, see classification in section 4.1.

Definition 4.1: A function is convex in a convex set S if and only if $\forall$ x' and x'' $\in$S

$$\begin{aligned} &f(cx'+(1-c)x'') \le cf(x') + (1-c)f(x'') \\ &0 \le c \le 1 \end{aligned} \tag{4.11}$$

The convexity of the objective functions and the constraints would be a useful property because of the following theorem.

Theorem 4.1:
Consider a NLP (non-linear programming) that is a minimisation problem. Suppose the feasible region S for NLP is a convex set. If f(x) is convex on S, then any local minimum for NLP is an optimal solution to this NLP.

The optimisation problem that has to be solved is a NLP. The study of each part of the problem has been done.

4.3.1. Cost function

The cost function presented in equation 4.6 is convex (as any quadratic function is) and the sum of convex functions is convex. It can be demonstrated easily from the definition:

$$f(h) = (h_0 - h)^2 \Rightarrow f(ch'+(1-c)h'') = (h_0 - ch' - (1-c)h'')^2 = (c(h_0 - h') - (1-c)(h_0 - h''))^2 \le$$
$$\le (c(h_0 - h'))^2 + ((1-c)(h_0 - h''))^2 \le c(h_0 - h')^2 + (1-c)(h_0 - h'')^2 = cf(h') + (1-c)f(h'')$$

and

$$if\ f(h) \text{ and } g(h) \text{ are convex} \Rightarrow (f+g)(ch'+(1-c)h'') = f(ch'+(1-c)h'') + g(ch'+(1-c)h'') \leq$$
$$\leq cf(h') + (1-c)f(h'') + g(h') + (1-c)g(h'') = c(f+g)(h') + (1-c)(f+g)(h'')$$

4.3.2. Constraints

Not only the cost function has to be convex to assure the global minimum of a problem. The set has to be convex and this set is defined by the constraints. To avoid the study of the convexity of this set, a second formulation can be used, equation 4.10. It is assumed that the set is a hyper-rectangle defined by bounds in the variables and the constraints are included in the cost function that has to be convex. Each term of the sum has to be convex and the convexity of the sum of them is straightforward. For the first term constraint definition can be used:

$$f([h,\vec{q}]) = \left(A_i h_{il} - A_i h_{il+1} - \sum_{j=1}^{p} q_{jl} \right)^2 \Rightarrow f(c[h_i',\vec{q}'] + (1-c)[h_i'',\vec{q}'']) =$$
$$= \left(A_i(ch_{il}' + (1-c)h_{il}'') - A_i(ch_{il+1}' + (1-c)h_{il+1}'') - \sum_{j=1}^{p}(cq_{jl}' + (1-c)q_{jl}'') \right)^2 =$$
$$= \left(A_i ch_{il}' - A_i ch_{il+1}' - \sum_{j=1}^{p} cq_{jl}' + A_i(1-c)h_{il}'' - A_i(1-c)h_{il+1}'' - \sum_{j=1}^{p}(1-c)q_{jl}'' \right)^2 \leq$$
$$\leq \left(A_i ch_{il}' - A_i ch_{il+1}' - \sum_{j=1}^{p} cq_{jl}' \right)^2 + \left(A_i(1-c)h_{il}'' - A_i(1-c)h_{il+1}'' - \sum_{j=1}^{p}(1-c)q_{jl}'' \right)^2 \leq$$
$$\leq c\left(A_i h_{il}' - A_i h_{il+1}' - \sum_{j=1}^{p} q_{jl}' \right)^2 + (1-c)\left(A_i h_{il}'' - A_i h_{il+1}'' - \sum_{j=1}^{p} q_{jl}'' \right)^2 = cf([h',\vec{q}']) + (1-c)f([h'',\vec{q}'']) \tag{4.12}$$

The demonstration for constraint presented in equation 4.8 is equivalent. The problem arises when the third constraint (equation 4.9) is treated. If the definition is applied to this part of the cost function the condition of convexity cannot be assured because of the product of two variables and the exponent of one of them.

$$f([h_1,h_2,R_i,q_i]) = (h_1 - h_2 - R_i q_i^{1,85})^2 \Rightarrow f(c[h_1',h_2',R_i',q_i'] + (1-c)[h_1'',h_2'',R_i'',q_i'']) =$$
$$= (ch_1' + (1-c)h_1 - ch_2 + (1-c)h_2 - (cR_i + (1-c)R_i)(cq_i + (1-c)q_i)^{1,85})^2$$

somehow this problem had already been detected in the identifiability study. In general non-linear functions are not necessarily non-convex but the linear ones are all convex. This constraint was the one that made the problem non-linear. The particular case of a

linear problem appeared when all heads were known and hence both variables could be isolated, the same would happen if all flows were known. Now it is intended to determine not only convexity of functions but also whether the problem is convex or not. A property of convex functions is used.

Theorem 4.2
Suppose $f(x_1,x_2,...x_n)$ has continuous second-order partial derivatives for each point $x=(x_1, x_2,...x_n)$. It is a convex function on S if and only if for each $x \in S$, H is positive semidefinite, i.e. all principal minors of H and the diagonal elements are nonnegative.

The Hessian of third constraint introduced in cost function of equation 4.10 is presented in equation 4.13. It can have negative diagonal elements or principal minors. In next section an example of non-convexity proved by this property is shown.

$$H(h_{1i},h_{2i},q_i,R_i)=\begin{bmatrix} 2 & -2 & -3.704q_i^{0.852} & -2q_i^{1.852} \\ -2 & 0 & 3.704q_i^{0.852} & 2q_i^{1.852} \\ -3.704q_i^{0.852} & 3.704q_i^{0.852} & -3.704(0.852h_1q_i^{-0.15}-0.852h_2q_i^{-0.15}+2.704R_iq_i^{1.704}) & -3.704((h_1-h_2)q_i^{0.852}-2R_iq_i^{1.704}) \\ -2q_i^{1.852} & 2q_i^{1.852} & -3.704((h_1-h_2)q_i^{0.852}-2R_iq_i^{1.704}) & 2q_i^{3.704} \end{bmatrix} \quad (4.13)$$

The problem cannot be assured to be convex and as it becomes more complex it can be assured that it is not. The problem is linear except for the last constraint where two variables are multiplying and one has a coefficient. The feasible region of an Optimisation problem involving non-linear equality constraints, is not a convex set, regardless of whether the equality constraint is convex or not [Flou-01]. The set generated by an equality constraint on a bi-dimensional space would be a line, in order to have a convex set this line should be a straight one generated only by linear equality constraints.

Example 4.1: A simple example of the optimisation that is supposed to be done will shed light on the problem of non-convexity. A very simple case is used again. Figure 4.4 shows a network with two reservoirs, two pipes, a demand associated with the junction. The measurements of flow in pipe one and head in junction will be used for the cost function. The parameters to be determined are the resistance in both pipes, and the variables to be determined are the flows. Demand and levels in reservoirs are known, they are not introduced in the cost function in order to reduce the dimensions to two and make the example illustrative.

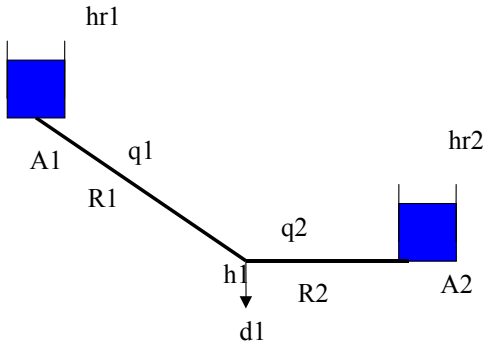


Fig. 4. 4

$$J = \left(\frac{h_{10} - h_1}{h_{10}} \right)^2 + \left(\frac{q_{10} - q_1}{q_{10}} \right)^2$$

$$h_{r10} - h_1 = R_1 q_1^{1.852}$$

$$h_{r20} - h_1 = R_2 q_2^{1.852}$$

$$q_1 + q_2 - d_1 = 0$$

Taking h_{10} and h_{20} as a known value and using as domain roughness R_1 and R_2 and getting the possible values of q_1 and h_1 from the constraints the cost function is represented in figure 4.5 and 4.6 for a set of possible roughness. It can be seen that the function has a minimum and does not show to have problems for the convergence of the minimisation.

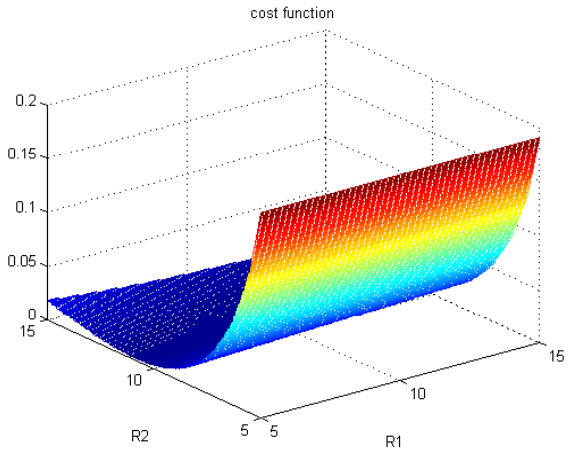


Fig. 4. 5

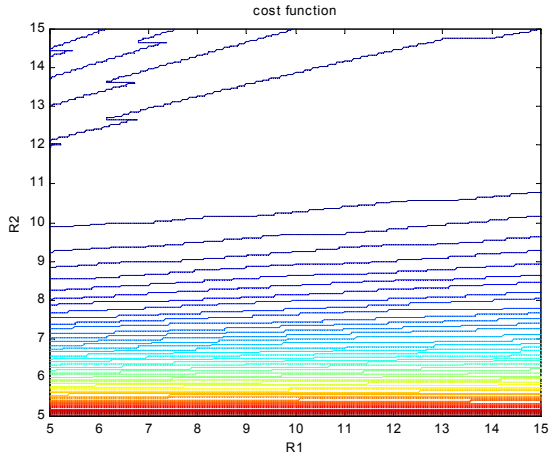


Fig. 4. 6

The problem does appear when the set is studied. It is not convex. In figure 4.7 the possible q_1 and h_1 are represented, they have been obtained from the set of possible roughness and using the constraints.

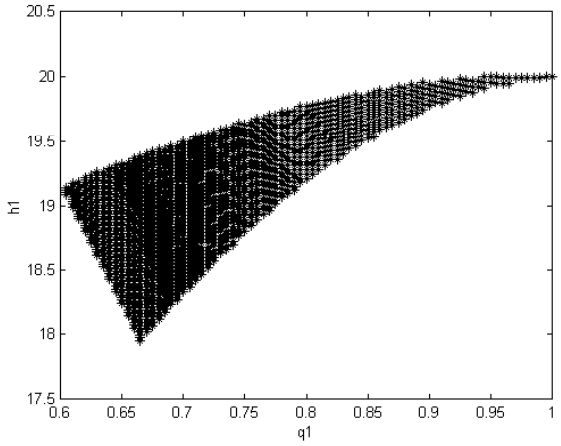


Fig. 4. 7

This is what was expected from a signomial programming. The no-convexity doesn't guarantee the global optimum using local optimisation.

Looking at figure 4.2, classification of the problem treated in this thesis can be summarised in a non-convex (non-linear) constrained continuous problem. Following this classification, and assuming that what is searched is the global minima, figure 4.3 provides some enlightenment regarding which kind of algorithm should be used. Before deciding which algorithms should be used some characteristics of the problem are studied. Through the investigation of cost minimisation techniques for engineering design problems, Zener [Zen-61],[Zen-62] stimulated the initial work in the area that is now known as *geometric programming*. The term geometric programming was adopted

because of the crucial role that the arithmetic –geometric mean inequality played in this initial development.

A posynomial is a function *g* of positive vector variable $x \in R^m$ having the form

$$g(x) = \sum_{i=1}^{N} u_i(x)$$

with (4.14)

$$u_i(x) = c_1 x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_m^{a_{im}}, i = 1,2,...,N$$

Where a_{ij} are arbitrary real numbers and the coefficients c_i are positive.
A *posynomial program* is a minimisation problem of the form

$$min\ g_0(x) \qquad \text{subject to } g_k(x) \le 1,\ \ k = 1,2,...,p \text{ and } x > 0 \tag{4.15}$$

where each g_k, k=1,2,...p, is a posynomial.
The problem presented in equations 4.6 to 4.9 can be formulated as:

$$min\ J = \sum_{l=1}^{t}\left[\sum_{i=1}^{n} {h_{il}}^2 - 2h_{il}h_{0il} + {h_{0il}}^2 + \rho\sum_{i=1}^{m} {q_{il}}^2 - 2q_{il}q_{0il} + {q_{0il}}^2\right] \tag{4.16}$$

There is no product of variables and the coefficients are integers.

$$\begin{aligned} &h_{il}{h_{il+1}}^{-1} + {A_i}^{-1}{h_{il+1}}^{-1}\sum_{j=1}^{p} q_{jl} \le 1 \\ &{h_{il}}^{-1}h_{il+1} - {A_i}^{-1}{h_{il}}^{-1}\sum_{j=1}^{p} q_{jl} \le 1 \text{ for } \forall\ i \le r \text{ and } l \le t \end{aligned} \tag{4.17}$$

$$\begin{aligned} &d_{jl}^{-1}\sum_{i=1}^{m} \Lambda_{ji} q_{il} \le 1 \\ &{q_{1l}}^{-1} d_{jl} - {q_{1l}}^{-1}\sum_{i=2}^{m} \Lambda_{ji} q_{il} \le 1 \text{ for } \forall\ i \le n \text{ and } l \le t \end{aligned} \tag{4.18}$$

$$\begin{aligned} &{h_{1il}}^{-1}h_{2il} + {h_{1il}}^{-1} R_i {q_{il}}^{1,852} \le 1 \\ &h_{1il}{h_{2il}}^{-1} - {h_{2il}}^{-1} R_i {q_{il}}^{1,852} \le 1 \text{ for } \forall\ i \le m \text{ and } l \le t \end{aligned} \tag{4.19}$$

This problem is slightly different to the posynomial as not all the coefficients are positive. Such programs are called *signomial programs* and were first studied by Passy and Wilde [Pas-67] and Blau and Wilde [Bla-69]. While posynomial programs have an equivalent reformulation as convex programs, no such formulation exists for a

signomial case. Duffin and Peterson [Duf-67] show that any signomial program can be transformed into an equivalent program of the form

$$\begin{aligned} &min\ g_0(x) \\ &\text{subject to} \\ &g_k(x) \le 1 \\ &g_k(x) \ge 1,\quad k = 1,2,...,p \end{aligned} \tag{4.20}$$

Where $g_k(x)$ is a posynomial for k=1,2...q. This program is called the *reversed posynomial program*.

The *reversed constraints* g(x)≥1 reformulated in equation 4.17 to 1.19 give rise to a non-convex feasible region. In general, local minima for signomial problems are not global minima.

4.4. Local Optimisation

The problem that is faced here is a non-linear and non-convex optimisation one. In order to solve such a problem classically, a methodology based on the search of an optimum from a seed, and using a search guided by the derivatives of the objective function, and with the local minimum conditions was used. These set of techniques are called **local optimisation** or **non-linear programming,** in contrast to those that guarantee the finding of the global optimum, **global optimisation.** The non-linear programming techniques work correctly, however they cannot assure the global optimum, but rather a local one. Some times this local optimum is a global one, when there is only one optimum; both objective function and feasible region are convex.

A **non-linear programming** problem with equality constraints like that presented in equations 4.6 to 4.9 can be expressed using Lagrange **multipliers**:

$$L(x,\lambda) = \sum_{l=1}^{r}\left[\sum_{i=1}^{n}\left(h_{0il} - h_{il}\right)^2 + \rho\sum_{i=1}^{m}\left(q_{0il} - q_{il}\right)^2\right] + \sum_{l=1}^{t}\sum_{j=1}^{r}\lambda_{1lj}\left(A_j h_{jl+1} - A_j h_{jl} - \sum_{i=1}^{p} q_{il}\right)$$
$$+ \sum_{l=1}^{t}\sum_{j=1}^{n}\lambda_{2lj}\left(\sum_{i=1}^{m}\Lambda_{ji} q_{il} - d_{jl}\right) + \sum_{l=1}^{t}\sum_{j=1}^{m}\lambda_{3lj}\left(h_{1il} - h_{2il} - R_i q_{il}^{1,852}\right) \tag{4.21}$$

Once the Lagrangian is constructed what is searched is the point (x_1*, x_2*,....,λ_1*, λ_2*....) that minimises L(**x***,**λ***). Then it accomplishes:

$$\frac{\partial L}{\partial \lambda_i} = 0 \tag{4.22}$$

This demonstrates that **x*** satisfies the constraints of the optimisation problem. So now the problem becomes

$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial x_2} = \ldots \frac{\partial L}{\partial \lambda_{i1}} = \frac{\partial L}{\partial \lambda_{i2}} = \ldots = 0 \tag{4.23}$$

When the minimisation problem has a differentiable function with continuity most of the algorithms applied are based on the following idea. From an initial point x_0 vectors are generated successively x_1, x_2,... so that the cost function decreases in each iteration.

$$J(x_{k+1}) < J(x_k) \tag{4.24}$$

Thus successively the estimation is improving till the minimum is reached or the algorithm cannot advance anymore. The next step is obtained using the following expression of equation 4.25.

$$x_{k+1} = x_k + \alpha_k d_k \tag{4.25}$$

Where if the gradient $\nabla f(x_k) \neq 0$ the direction is chosen so that:

$$\nabla f(x_k)' d_k < 0 \tag{4.26}$$

Normally direction is chosen as:

$$d_k = -D_k \nabla f(x_k) \tag{4.27}$$

The step α_k is fixed and positive. If D_k is chosen I (identity) then the method is called a **gradient descendent method**. If the gradient $\nabla f(x_k)=0$ then the algorithm stops. This is the basic algorithm based on gradient. It is the simplest and the one with the slowest convergence. The most sophisticated algorithm based on this idea is the **Newton method**. It has the fastest convergence. The difference in respect to the previous algorithm is the descent direction used.

$$d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k) \tag{4.28}$$

Conjugate gradient method and **quasi-Newton method**, are intermediate algorithms from the complexity point of view. When equality non-linear constraints are present New algorithms are developed as an evolution of these.

Reduced Gradients based methods are extensions of the algorithms used for the linear constraints. These methods have as a basic principle to remain in a feasible region defined by the non-linear constraints, while the cost function is minimised.

Sometimes in the cost function some constraints are introduced. This is called the penalisation or barrier function. Associated with this methodology exist a parameter *c* that determines the severity of the penalisation. These are the **augmented Lagrangian methods.**

When a point **x*** fulfils the sufficient conditions of a local minimum in a problem of optimisation with non-linear constraints, such a point is a minimum of the Lagrange function in a sub-space of vectors which are orthogonals to the gradient of the constraints. Such a property suggests that **x*** can be defined as the solution of a sub-problem of optimisation with linear constraints selected in a way that the minimisation is done inside the desired subspace. Restricting the desired subspace the Lagrange has not to be augmented in order to convert the point **x*** from a stationary point to a minimum. **Projected Lagrangian Methods** include those that contain a sequence of sub-problems of optimisation with linear constraints based in a Lagrange function.

4.4.1. Matlab

Matlab has a toolbox of optimisation with different optimisation functions that allow solving problems of non-linear optimisation finding local optima. In this case what is needed is a function that solves problems with non-linear constraints.

Function **constr** of MATLAB optimises functions with constraints. The algorithm used by this function is called **Sequential Quadratic Programming.** This algorithm belongs to those based on projected Lagrangian. It uses the algorithm of quasi-Newton to solve the Kuhn-Tucker conditions of the original problem. The problem associated is a minimisation of a quadratic approximation of the Lagrangian function over the feasible region defined by the linear approximation of the constraints.

The problem presented in equation 4.5 generates in an iteration (x_k,u_k,v_k), where u_k and v_k are respectively the estimation of Lagrange multipliers for the equality and inequality constraints, the following quadratic optimisation sub-problem:

$$\begin{aligned} QP(x_k, v_k, u_k) \quad \min \quad & f(x_k) + \nabla f(x_k)^t d + \tfrac{1}{2} d^t \nabla^2 L(x_k) d \\ \text{subject to} \quad & g_i(x_k) + \nabla g_i(x_k)^t d \le 0 \quad \text{para } i = 1...m \\ & h_i(x_k) + \nabla h_i(x_k)^t d = 0 \quad \text{para } i = 1...n \end{aligned} \tag{4.29}$$

where

$$\nabla^2 L(x_k) = \nabla^2 f(x_k) + \sum_{i=1}^{m} u_{ki} \nabla^2 g_i(x_k) + \sum_{i=1}^{n} v_{ki} \nabla^2 h_i(x_k)$$

If d_k solves problem QP(x_k,u_k,v_k) with the Lagrange multipliers u_{k+1}, v_{k+1} and if d_k=0 then x_k together with (u_{k+1},v_{k+1}) gives a solution to the problem P. Otherwise a new

point is generated $x_k=x_k+d_k$, k is increased in 1 and the process is repeated. One drawback of this method is the need of calculating second derivatives of the objective function and the constraints, besides $\nabla^2 L(x_k)$ can be non-positive defined. In order to avoid this last problem the quasi-Newton positive-defined approximations for $\nabla^2 L(x_k)$ can be used.

One of the main drawbacks of this method is that it guarantees convergence only when the seed is close enough to the solution, which is actually difficult. In order to avoid this problem and improve the convergence, the merit function is introduced. Merit function is minimised with the objective function but it is only useful as a descending function, it guides the algorithm and indicates the progress. This has to be easy to evaluate in order not to deteriorate the convergence speed of the algorithm. An example of such a function is presented in equation 4.30:

$$F_E(x) = f(x) + \mu\left[\sum_{i=1}^{m} max\{0, g_i(x)\} + \sum_{i=1}^{n} |h_i(x)|\right] \tag{4.30}$$

Algorithm 4.1: Sequential Quadratic Programming

(1) Initialise k=1 and a seed is selected in the feasible region x_1. An approximation B_k positive defined is selected for $\nabla^2 L(x_k)$ defined in respect to the lagrangian multipliers $u_k \geq 0$ and v_k associated with the constraints. The optimisation problem P is solved. B_k could be arbitrary with no relation with $\nabla^2 L(x_k)$.

(2) The quadratic problem is solved where $\nabla^2 L(x_k)$ has been substituted by B_k. d_k and (u_{k+1}, v_{k+1}) are obtained.

(a) If $d_k=0$ the algorithm stops.

(b) Otherwise $x_{k+1}=x_k+\lambda d_k$, where λ minimises $F_E(x_k+\lambda d_k)$ for $\lambda \in R^1$ and $\lambda \geq 0$. B_k is updated to a positive matrix B_{k+1} using the quasi-Newton method. k is increased in 1 and step 2 is repeated.

(c)

This is the optimisation tool used in the examples 2.5, 2.9 and 2.11. It has been useful for small problems where the seed could be selected close enough to the result but in the case study of section 2.9, when a more realistic networks was calibrated, the algorithm showed to be insufficient. The non-convexity of the problem demonstrated in previous sections fits with this observation.

Example 4.2: Figure 4.8 shows a simplification of Barcelona's network. This is used to show how this algorithm fails as the network grows and to have an example to test other algorithms that should work better.

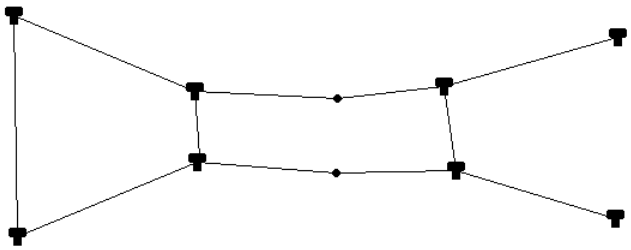


Fig. 4. 8

Having all levels and three flows measured, variables are 8 flows, two heads, 11 roughness and two demand factors. The system is identifiable with 3 time-steps but the algorithm fails in the estimation even with 8 time steps. Figure 4.9 shows the result of the estimation for levels, heads and flows. Figure 4.10 shows the result of the estimation for roughness and demand factors.

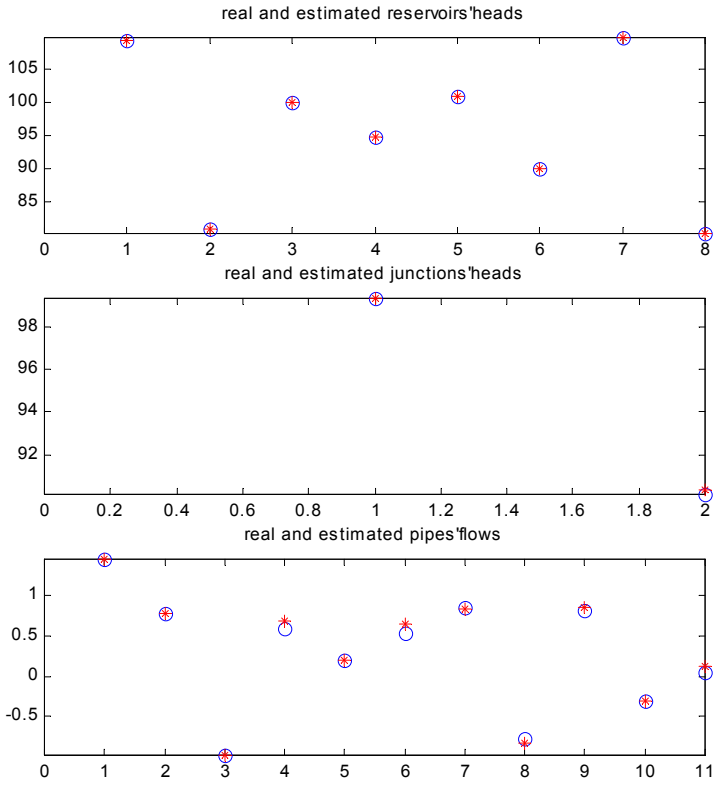


Fig. 4. 9

Though most of the flows are not known, both flows and demands are well estimated because of the linear relations. The non-linear relations, that have been shown to be non-convex relations, make impossible a good estimation of the roughness.

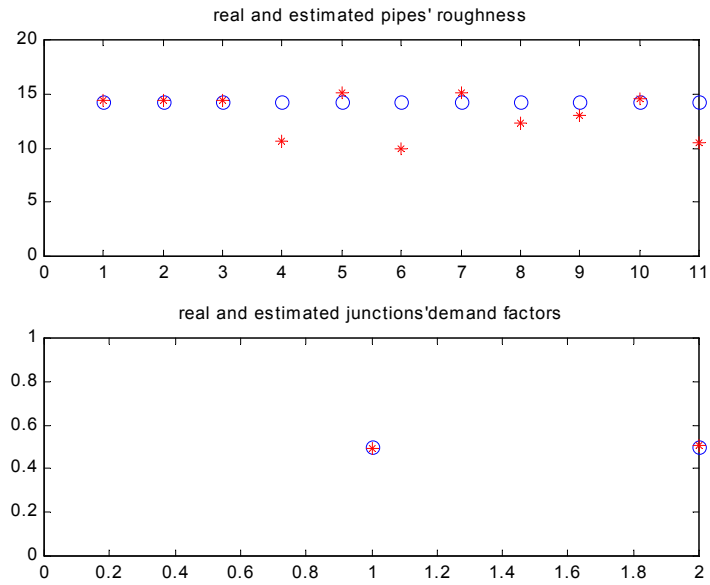


Fig. 4. 10

4.4.2. GAMS-CONOPT

Another solver available in the group is GAMS. GAMS (General Algebraic Modelling System) [Gam-03] was developed by Meeraus and Brooke of the GAMS Development Corporation. For linear, and non-linear programming, GAMS is a high-level modelling language that is formally similar to Fortran. It allows the user to describe models in concise algebraic statements that are readily comprehensible to other readers. Features such as sensitivity analysis and report generation are provided. GAMS provides an interface to the MINOS system developed by Murtagh [Mur-86]. Limitations on problem size include a limit of 32,767 rows in constraints matrix for the MINOS module. Additional modules (which link GAMS to other numerical optimisation codes) are available as options. Some of these are available for certain environments. In the group some work had been done with CONOPT [Con-03] for non-linear mixed–integer programming. It had been used in the pump-scheduling of the network in WATERNET project.

The algorithm in CONOPT is based on the generalised reduced-gradient (GRG) algorithm. All matrix operations are implemented by using sparsematrix techniques to allow very large models. Without compromising the reliability of the GRG approach, the overhead of the GRG algorithm is minimised by, for example, using dynamic

feasibility tolerances, reusing Jacobians whenever possible, and using an efficient reinversion routine. The algorithm uses many dynamically set tolerances and therefore runs, in most cases, with default parameters.

Reduced-gradient algorithms avoid the use of penalty parameters by searching along curves that stay near the feasible set. Essentially, these methods take the formulation 4.5 and use the equality constraints to eliminate a subset of the variables, thereby reducing the original problem to a bound–constrained problem in the space of the remaining variables. If x_B is the vector of eliminated or basic variables and x_N is the vector of non-basic variables, then $x_B=h(x_N)$.

Results obtained using this solver were not better than those obtained with Matlab. The local optimisations algorithms show to be insufficient for this problem.

4.5. Global Optimisation

The problem presented in equation 4.5 corresponds to a general optimisation problem with equality and inequality constraints. The box $X=(x_1,...x_n)$ defines an artificial limitation on the search region, while $\underline{x}_i \le x_i \le \bar{x}_i$ represent actual bound constraints for a subset of the coordinates. In fact, if actual bound constraints exist, it can be advantageous to treat bound constraints separately from the inequality and equality constraints.

Deterministic algorithms used to find the global minimum of a function, in a region, do an exhaustive search of all such regions. In this thesis three different algorithms have been used. All of them have in common that they use the branch and bound algorithm for the search. The basic branch and bound is presented:

Algorithm 4.2: Abstract Branch-and-Bound Pattern for Optimisation)
INPUT:an initial box X_0
OUTPUT:a list of boxes that have been proven to contain critical points and a list U of boxes with small objective function values, but which could not otherwise be resolved.

1. Initialise a list of boxes L by placing the initial search region $\mathbf{X_0}$ in L.
2. DO WHILE L≠∅.
   a) *Remove the first box **X** from L.*
   b) *(Process **X**) Do one of the following:*
      - *reject **X***
      - *reduce the size of **X**;*
      - *determine that **X** contains a unique critical point, then find the critical point to high accuracy;*
      - *subdivide **X** to make it more likely to succeed at rejecting, reducing, or verifying uniqueness.*

*c) Insert one or more boxes derived from **X** onto L, U or C, depending on the size of the resulting box(es) from Step 2b and whether the (possible) computational existence test in that step determined an unique critical point.*

END DO

Algorithm 4.2 represents a general description, and many details such as stopping criteria and tolerances, are absent. Such details differ in particular actual algorithms. The three algorithms used in this thesis used different methods for both branching (subdivide the boxes or regions) and bounding (finding lower and upper bounds for the function in each region in order to reject boxes). Reasons for selecting these algorithms were different in each one.

Signomial programming: In the characterisation of the problem, it was formulated as a signomial problem. It is a non-convex problem but taking advantage of the form and using a special way in the branching process the bounding can be done based on convex optimisation. This algorithm has been used to utilise the geometry of the problem.

Lipschitzian optimisation: some branch-and-bound algorithms use the lipschitz constant of the function for the bounding process. The research group works mainly with Matlab software and there is a collaboration agreement with the distribution company. In Matlab environment there is a commercial software for global optimisation based in these algorithms and has been tested for the calibration problem.

Interval arithmetic: the research group has worked since 1998 in interval arithmetic with different purposes such as fault detection, robust control, identification....The knowledge on this subject included the optimisation algorithms that use interval arithmetic for the bounding of the function and interval boxes for the branching. A tool based on these algorithms has been tested presenting the best results.

4.5.1. Signomial programming

A *signomial programming* problem such as the one presented with equations 4.16 to 4.19 can generally be expressed in the following form as two non-convex separable programming problems:

$$\begin{aligned} &min\ f_0(x) = g_0(x) - h_0(x) \\ &\text{subject to } f_k(x) = g_k(x) - h_k(x) \le 1,\ \ k = 1,2,...,q \text{ and } x \ge 0 \end{aligned} \tag{4.31}$$

where each $g_k(t)$ and $h_k(t)$ are posynomial

$$g_k(x) = \sum_{i \in IG(k)} c_i \prod_{j=1}^{p} x_j^{a_{ij}}$$
$$h_k(x) = \sum_{i \in IH(k)} c_i \prod_{j=1}^{p} x_j^{a_{ij}}, \quad k = 0,1...,q \tag{4.32}$$

Where the number of posynomial terms is n, and q is the number of the signomial constraints. The exponents a_{ij} are arbitrary real numbers and the coefficients c_i are all positive. Define the exponent matrix **A** as

$$\mathbf{A} = \{a_{ij}\}_{nxm}$$

The index sets *IG(k)*and *IH(k)* (k=0,1,...,q) are mutually disjointed subsets of the set

$$\{1,2,...,n\} = \bigcup_{k=0}^{q} (IG(k) \cup HG(k))$$

and are ordered in such a way that

$$i_0 < i_0' < i_1 < i_1' < ... < i_q < i_q'$$

Where $i_k \in IG(k)$ and $i_k' \in IH(k)$. Some of the sets *IG(k)* and *IH(k)* may be empty although the sets $IG(k) \cup IH(k)$ must not be empty for any k=0,1,...,q.
Two new sets are defined

$$IG = \bigcup_{k=0}^{q} IG(k)$$
$$IH = \bigcup_{k=0}^{q} IH(k)$$

Notice, when the set *IH* is empty, i.e., IH=0, the program becomes posynomial. Each posynomial program is equivalent to a *convex program* (minimises a convex function over a convex domain). This equivalence is established by introducing the one-to one transformation by letting $\mathbf{z}=(z_1,... z_m)^T$ and letting

$$x_j = e^{z_j}, \quad j = 1,2,...,m \tag{4.33}$$

The problem is equivalent to the convex problem

$$\begin{aligned} &min \sum_{i\in IG(0)} c_i e^{(\mathbf{a}_i,\mathbf{z})} \\ &\text{subject to } \sum_{i\in IG(k)} c_i e^{(\mathbf{a}_i,\mathbf{z})} \quad k=1,2,...,p \end{aligned} \tag{4.34}$$

Where $\mathbf{a_i}=(a_{i1},...,a_{im})$ is the *i*th row of the exponent matrix **A** and c_i is the positive coefficient associated with the *i*th posynomial term. $(\mathbf{a_i},\mathbf{z})$ denotes the inner product of the vectors $\mathbf{a_i}$ with the vector **z**. One can easily check that $e^{(\mathbf{a_i},\mathbf{z})}$ defines a convex function of **z**. The significance of this equivalence is that every local solution of convex problem presented in equation 4.34, $\mathbf{z}^*$, is also a global optimal solution to it. Because the transformation 4.33 is a one-to-one transformation, the corresponding $\mathbf{x}^*=(x^*_1,..., x^*_m)^T$ is also a **global** solution of the original problem. Similarly, the one-to-one transformation can be applied to signomial program *P* that becomes problem *Q*

$$\begin{aligned} &minF_0(\mathbf{z}) = G_0(\mathbf{z}) - H_0(\mathbf{z}) \\ &\text{subject to } F_k(\mathbf{z}) = G_k(\mathbf{z}) - H_k(\mathbf{z}) \le 1, \quad k=1,2,...,q \\ &\text{where} \\ &G_k(\mathbf{z}) = \sum_{i\in IG(k)} c_i e^{(\mathbf{a}_i,\mathbf{z})} \\ &H_k(\mathbf{z}) = \sum_{i\in IH(k)} c_i e^{(\mathbf{a}_i,\mathbf{z})}, \quad k=1,2,...,q \end{aligned} \tag{4.35}$$

Obviously functions $G_k(\mathbf{z})$ and $H_k(\mathbf{z})$ are convex since they are sums of convex functions of linear functions.

There have been some investigations of methods for finding global minima for signomial problems. Falk [Fal-73] proposes a branch-and bound algorithm that converges to global minima for (nonconvex) signomial programs. Using the initial a priori lower and upper bounds on variable **z**, **l** and **L**, can be defined sets

$$R = \{\mathbf{z} : \mathbf{l} \le \mathbf{z} \le \mathbf{L}\} \tag{4.36}$$

Given set *R* above a related set S^l *is described*

$$S^1 = \{\mathbf{z} : m_i^1 \le (\mathbf{a},\mathbf{z}) \le M_i^1; i \in IH\} \tag{4.37}$$

where $m^l{}_i=\min\{(\mathbf{a_i},\mathbf{z}):\mathbf{z}\in R\}$ and $M^l{}_i=\max\{(\mathbf{a_i},\mathbf{z}):\mathbf{z}\in R\}$; $i\in IH$. Each inequality definition of S^l defines a region in $\Re^n$ between two parallel hyperplanes. Notice that the elements of the row vector $\mathbf{a_i}$ decide the slope of the hyperplanes. The numbers $m^l{}_i$ and $M^l{}_i$ can be easily computed to be

$$m_i^1 = \sum_{j=1}^{n} min\{a_{ij}l_j, a_{ij}L_j\}$$
$$M_i^1 = \sum_{j=1}^{n} max\{a_{ij}l_j, a_{ij}L_j\}; \quad i \in IH \tag{4.38}$$

Clearly, $R \subseteq S^1$ and hence S^1 contains global solution of problem Q. The algorithm generates a sequence of problems Q^t's defined on sets S^t, t=1,2,3,... The set S^t is defined as

$$S^t = \left\{\mathbf{z} : m_i^t \le (\mathbf{a}, \mathbf{z}) \le M_i^t; i \in IH\right\} \tag{4.39}$$

where numbers $m^1{}_i$ and $M^1{}_i$ are updated during the process of branching-and –bounding. Notice that $S^t \subseteq S^1$ and $S^1 = \cup_{t \in I(s)} S^t$ where $\boldsymbol{I}$(s) is the set containing all the intermediate nodes at step *s*. In figure 4.11 the darkest region corresponds to the rectangular bounds of the original problem *R*, the lighter shaded region corresponds to the overestimation *S*.

Each of these problems Q^t's is referred to as a node in the process of branching-and-bounding. New problems are created by selecting an existing node Q^T and by setting up new problems that serve better estimates of Problem Q than Q^T at each stage s, only two new nodes are created. This process of generating new problems is referred to as branching. Hence, stage 1 consist of Problem Q^1, stage 2 of problems Q^1, Q^2 and Q^3, and stage s of Problems $Q^1,...,Q^{2s-1}$. Therefore, there are (2s-1) sub-problems at the stage s. If all the problems Q^t's were drawn in one chart with the branchings shown as a line (branch) connecting two related nodes, a branch-and-bound tree would be obtained. Problem Q^t is defined as

$$minF_0(\mathbf{z}) = G_0(\mathbf{z}) - L^t{}_0(\mathbf{z})$$
$$\text{subject to } F_k(\mathbf{z}) = G_k(\mathbf{z}) - L^t{}_k(\mathbf{z}) \le 1, \quad k = 1,2,...,q$$
where
$$\mathrm{L}^{\mathrm{t}}_{\mathrm{k}}(z) = \sum_{i \in IH(k)} l_i^t(\mathbf{z}) \tag{4.40}$$
and
$$l_{\mathrm{i}}^{\mathrm{t}}(\mathbf{z}) = \left(\frac{C_i}{M_i^t - m_i^t}\right)\left(\left(M_i^t e^{m_i^t} - m_i^t e^{M_i^t}\right) + \left(e^{M_i^t} - e^{m_i^t}\right)(\mathbf{a_i}, \mathbf{z})\right)$$

Notice that $l^t{}_i$ is a linear overestimate of $c_i e^{(\mathbf{a}_i, \mathbf{z})}$ when $m^t{}_i \le (\mathbf{a}_i, \mathbf{z}) \le M^t{}_i$, i.e.

$$l_{\mathrm{i}}^{\mathrm{t}}(\mathbf{z}) \ge c_i e^{(\mathbf{a_i}, \mathbf{z})} \tag{4.41}$$

so that $F^t{}_k(\mathbf{z})$ underestimates $F_k(\mathbf{z})$ when $\mathbf{z} \in S^t$:

$$L_{\mathrm{i}}^{\mathrm{t}}(\mathbf{z}) \geq H_k(\mathbf{z}) \tag{4.42}$$

and

$$F_{\mathrm{k}}^{\mathrm{t}}(\mathbf{z}) \geq F_k(\mathbf{z}), \mathbf{z} \in \mathbf{S^t} \tag{4.43}$$

this overestimate is shown in figure 4.12. Following with the same example. Problem Q^t is a convex program so it can be solved by any convex programming algorithm and achieve its global solution. Solution for each problem Q^t is defined $\mathbf{z^t}$ if it exists and the lower and upper bounds on F_0 over $F(Q) \cap S^t$.

$$\begin{aligned} v^t &= \begin{cases} F_0^t(\mathbf{z}^t) \text{ if } \mathbf{z}^t \text{ exists} \\ +\infty \text{ otherwise} \end{cases} \\ V^t &= \begin{cases} F_0^t(\mathbf{z}^t) \text{ if } \mathbf{z}^t \text{ exists and feasible in Q} \\ +\infty \text{ otherwise} \end{cases} \end{aligned} \tag{4.44}$$

$\mathbf{z^*}$ is the global solution of problem Q and v^* is the value of objective function $F(\mathbf{z^*})$. The numbers $v_b(s)$ and $V_b(s)$ defined as

$$\begin{aligned} v_b(s) &= \min_{t \in I(s)} \{v^t\} \\ V_b(s) &= \min_{t \in I(s)} \{V^t\} \end{aligned} \tag{4.45}$$

Constitute the best available lower and upper bounds on v^*.A node Q^t is called *intermediate* at the stage if no new problem has been created from it. Set $I(s)=\{t: Q^t$ is an intermediate problem at stage $s\}$. The branching is controlled by the Weak Refining Rule. Only two new nodes are created in each stage. A node Q^T such that satisfies $T \in I(s)$ and $v^T=V^s$ is chosen. Index K that maximises $L^T{}_K(z^T)-H_K(z^T)$ is chosen and then select any $I \in IH(K)$ so that the difference $l^T{}_K(z^T)-c_i e^{(\mathbf{a}_i \mathbf{z^T})}$ is maximal. Next the corresponding interval (m^T_I, M^T_I) is divided into $(m^T_I, (a_I, \mathbf{z^T}))$ and $((a_I, \mathbf{z^T}), M^T_I,$ upon which new problems Q^{2s} and Q^{2s+1} are created. An algorithm based on signomial programming has been implemented using MATLAB. It will be useful to follow what has been said so far with the steps required when using the example of only one pipe. These are the steps:

Algorithm 4.3

Step 1: Initialisation.

Set up and solve Problem Q^1 to obtain $\mathbf{z^1}$, v^1, V^1, U^1

Set s=1, t=1, $I(1)$

Step 2: Check for solution

If $V_b(s)=v_b(s)$ then a global solution of Problem Q is $\mathbf{z}^*=\mathbf{z}^T$ where $v^T=V^s$. Otherwise go to step 3.

Step 3: Choose a Branching Node T
Select $T\in I(s)$ such that $v^T=v_b(s)$. Go to Step 4.
Step 4: Select a Branching Term ($\mathbf{a_i}$,$\mathbf{z}$)
Step 4.1 Select a $K\in\{0,1,...q\}$ such that $L^T{}_k(\mathbf{z}^T)-H_k(\mathbf{z}^T)$ is maximal.
Step 4.2 Select an $I\in IH(K)$ so that $l^T{}_i(\mathbf{z}^T)-e^{(\mathbf{ai},\mathbf{zT})}$ is maximal.
Step 4.3 Go to Step 5
Step 5: Generate Problem Q^{2s}(see Fig. 4.13)
Set $m^{2s}{}_i=m^T{}_i$; $M^{2s}{}_i=M^T{}_i$; $\forall i\in IH;\ i\neq I$
$M^{2s}{}_I=(a_I,\mathbf{z}^T)$; $m^{2s}{}_I=m^T{}_I$
Step 6: Generate Problem Q^{2s+1}
Set $m^{2s+1}{}_i=m^T{}_i$; $M^{2s+1}{}_i=M^T{}_i$; $\forall i\in IH;\ i\neq I$
$m^{2s+1}{}_I=(a_I,\mathbf{z}^T)$; $\mathrm{M}^{2s+1}{}_I=m^T{}_I$
Step 7: Solve Problems Q^{2s} and Q^{2s+1}
Step 7.1 Solve (or check feasibility) Problems Q^{2s} and Q^{2s+1} to obtain:
$\mathbf{z}^{2s}$, $\mathbf{z}^{2s+1}$, $\mathbf{v}^{2s}$, $\mathbf{v}^{2s+1}$, u^{2s}, u^{2s+1}.
Step 7.2 Compute V^{s+1} and U^{s+1}.
Step 7.3 Update the set $I(s+1)$ from $I(s)$:
delete the node T, add two nodes 2s and 2s+1. Set s:=s+1.

Step 7.4 Go to Step 3.

Example 4.3: Figure 2.8 shows the simplest system used to illustrate this algorithm. Heads are known, so variables are flow and roughness. Constraint 4.9 is expressed in residual form 4.10. The problem appears this way. There are no constraints and only two variables in order to make it understandable.

$$min\ f_0(x) = (hr1 - h1 - R_1 q_1^{1.852})^2 = (hr1 - h1)^2 + R_1{}^2 q_1^{3.7} - 2(hr1 - h1) R_1 q_1^{1.852} = g_0(x) - h_0(x)$$

$$(hr1(2) - hr1(1)) * Area - q_1 = 0 \Rightarrow \begin{matrix} q_1 / ((hr1(2) - hr1(1)) * Area) \le 1 \\ q_1^{-1} * (hr1(2) - hr1(1)) * Area \le 1 \end{matrix}$$

$$0.5 \le q_1 \le 1$$

$$12 \le R_1 \le 12.5$$

$$g_0(x) = (hr1 - h1)^2 + R_1{}^2 q_1^{3.7}$$

$$h_0(x) = 2(hr1 - h1) R_1 q_1^{1.852}$$

$$g_1(x) = ((hr1(1) - hr1(2)) * Area)^{-1} q_1$$

$$g_2(x) = ((hr1(1) - hr1(1)) * Area) q_1^{-1}$$

$$IG_0 = \{1,2,4,5\}$$

$$IH_0 = \{3\}$$

$$c = \left\{(hr1 - h1)^2, 1, 2(hr1 - h1), ((hr1(2) - hr1(1)) * Area)^{-1}, ((hr1(2) - hr1(1)) * Area)\right\}$$

$$A = \begin{bmatrix} 0 & 0 \\ 3.7 & 2 \\ 1.85 & 1 \\ 1 & 0 \\ -1 & 0 \end{bmatrix}$$

$$q_1 = e^{z_1}; R_1 = e^{z_2}$$

In figure 4.11 both regions R (original boundary region) and S^1 for the first problem is shown. In order to get this region all these values have been calculated:

$l_1 = \ln(0.5) = -0.69 \le x_1 \le 0 = \ln(1) = L_1$

$l_2 = \ln(12) = 2.49 \le x_2 \le 2.53 = \ln(12.5) = L_2$

$m^1 = [0 \quad 0 \quad 1.2]$

$M^1 = [0 \quad 0 \quad 2.5]$

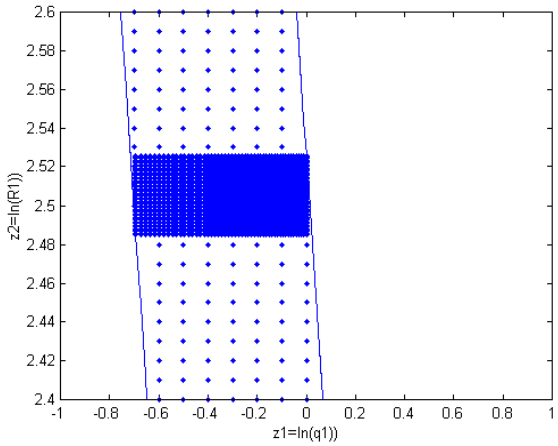


Fig. 4. 11

The overestimation of the only *H* term appears in figure 4.12. These are the values for this special case:

$l_3^1(z) = -63.6 + 88.1 * (1.852 * z_1 + z_2)$

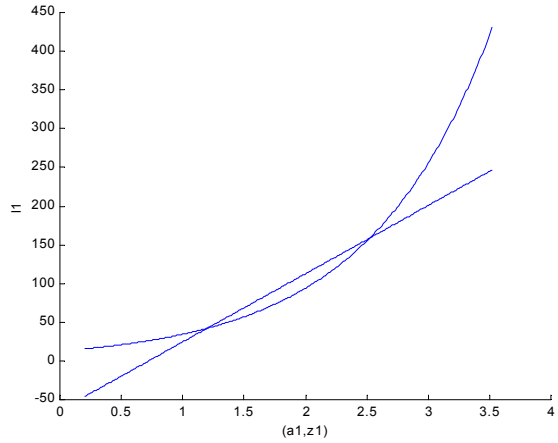


Fig. 4. 12

Now follows the algorithm for this problem.

Step 1: The problem is initialised

$s = 1; t = 1; I = \{1\}$

$z^1 = \begin{bmatrix} -0.3198 & 2.4849 \end{bmatrix}$

$v^1 = -18.65$

$V^1 = 0.0659$

$v_b(1) = -18.65$

$V_b(1) = 0.0659$

Step 2

Obviously $V_b(s) \neq v_b(s)$ so go to step 3

Step 3

The node chosen the first time is the only one and it is true that $v^1 = v_b(1)$

Step 4

The signomial term selected is the only one that has H_k term. It is term H_0 corresponding to cost function. And as it has only one term this is chosen, $IH=3$.

Step 5

$m^2 = \begin{bmatrix} 0 & 0 & 1.2 \end{bmatrix}$

$M^2 = \begin{bmatrix} 0 & 0 & 1.8 \end{bmatrix}$

$1_3^2(z) = -29.7 + 59.9 * (1.852 * z_1 + z_2)$

Step 6

$m^3 = \begin{bmatrix} 0 & 0 & 1.8 \end{bmatrix}$

$M^3 = \begin{bmatrix} 0 & 0 & 2.5 \end{bmatrix}$

$1_3^3(z) = -134.7 + 116.3 * (1.852 * z_1 + z_2)$

In figure 4.13 appears the new overestimation for the two new problems Q^2 and Q^3.

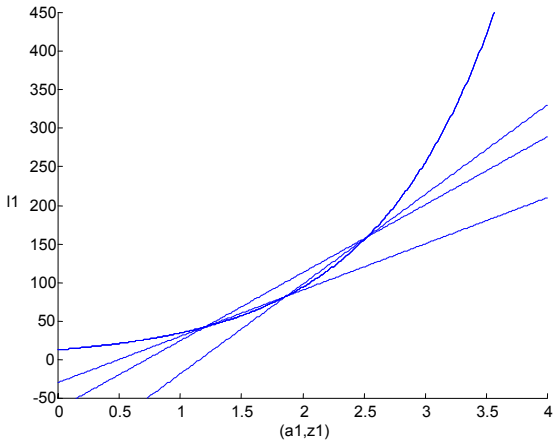


Fig. 4. 13

Step 7

$z^2 = \begin{bmatrix} -0.3581 & 2.4849 \end{bmatrix}$

$v^2 = -0.7630$

$V^2 = 0.0863$

$z^2 = \begin{bmatrix} -0.3579 & 2.5259 \end{bmatrix}$

$v^3 = \infty$

$V^3 = \infty$

The first node is removed and the third one is removed too as $v^3 > V_b(1)$.

$v_b(2) = -0.763$

$V_b(2) = 0.0659$

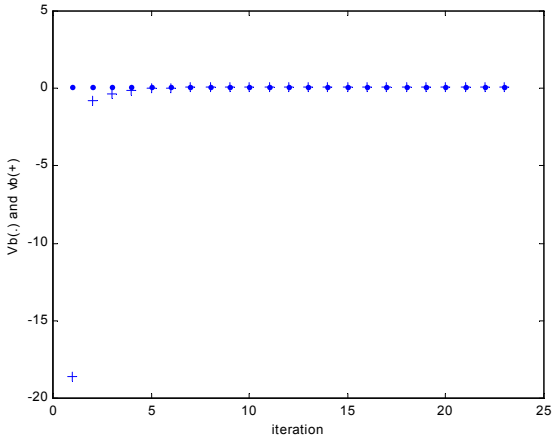


Fig. 4. 14

This algorithm assures the global optimal. In figure 4.14 the boundaries of the cost function appear and it can be seen that it converges and solution is found. It generates a huge number of optimisation problems that explodes when the number of variables increases as occurs with problems in the real network.

4.5.2. Lipschitzian Optimisation

DIRECT (dividing rectangles) is an algorithm for finding the minimum of a Lipschitz continuous function. DIRECT is designed to solve problems subject to box constraints. An exact mathematical definition of the problem is given below after the definition of Lipschitz continuity.

Definition 4.2: *Let $f:R^N \rightarrow R$. f is called Lipschitz continuous with Lipschitz-constant γ if and only if*

$$\|f(x)-f(x')\| \le \gamma \|x-x'\| \quad \forall \mathrm{x}, \mathrm{x}' \tag{4.46}$$

Problem
Let $f:R^N \rightarrow R$. f is called Lipschitz continuous with Lipschitz-constant γ solve

$$\begin{aligned} &\min_{x\in\Omega} f(x) \\ &\Omega = \left\{x \in R^N \mid l_i \le (x)_i \le u_i\right\} \\ &-\infty < l_i \le u_i < \infty \end{aligned} \tag{4.47}$$

It is important to note that the function is not needed to be continuous and the derivatives do not need to exist.

The **classical 1-D Lipschitzian Optimisation** will help to understand the base of these algorithms. Let N=1 and let f be Lipschitz continuous with constant γ on [a,b]. Therefore by setting x'=a and x'=b in equation 4.46 the following inequalities are produced:

$$\begin{aligned} f(x) &\ge f(a)-\gamma(x-a) \\ f(x) &\ge f(a)+\gamma(x-a) \end{aligned} \tag{4.48}$$

With these inequalities a piecewise linear function $\hat{f}$, which lies below f.

$$\begin{aligned} &\hat{f}(x) = \begin{cases} f(a)-\gamma(x-a) & \text{if } x \le x(a,b) \\ f(a)+\gamma(x-a) & \text{otherwise} \end{cases} \\ &x(a,b) = \left[f(a)-f(b)\right]/(2\gamma)+\left[a+b\right]/2 \end{aligned} \tag{4.49}$$

Note that also that $\hat{f}$ has a minimum value of

$$B(a,b) = \left[f(a)+f(b)\right]/2 - \gamma(b-a)/2$$

Example 4.4: Figure 4.15 shows this using the same cost function as in example 4.3. This is the constraint that makes the problem non-convex. Here it appears one-dimensional in order to show the way this algorithm works as simply as possible.

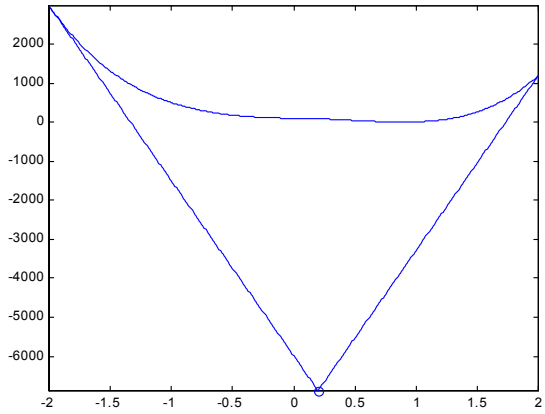


Fig. 4. 15

The idea is now to divide the search area into two intervals $I_1=[a,x(a,b)]$ and $I_2=[x(a,b),b]$, calculate the new values of x and B for each of these two intervals and choose a new interval to divide. Obviously the interval with the lowest value of B will be chosen. Following this ideas is Piyavskii's [Piy-72] algorithm, that obtains a piecewise linear function, this produces a better approximation of the real function in every iteration.

Algorithm 4.4

1. *n=1, sample=1,* $l_{sample}=a, u_{sample}=b$
2. *Calculate* B_1, x_1
3. *Do while n<numit*
   a. *n=n+1*
   b. $l_n=x_{sample}, u_n=u_{sample}, u_{sample}=x_{sample}$
   c. *Calculate* B_n, x_n, B_{sample}, x_{sample}
   d. *Choose new interval to sample*
4. *substitute* x_{sample} ,l_{sample} , u_{sample} *and* B_{sample} *by* x_n, x_{sample} , l_n, l_{sample} , u_n, u_{sample}, B_n, B_{sample}

In this algorithm the first two steps are the initialisation, where *n* is a counter-variable, *sample* is the interval in which the algorithm is sampling and l_{sample} and u_{sample} are the lower and upper bounds of the sampling area In the inner loop the counter-variable *n* is increased, after which the active interval is divided into two new intervals. In the next step the algorithm recalculates the values of *x* and *B* and then chooses a new interval to sample. This new interval will be the interval with the lowest value of B. This is done until a given number of function evaluation is done. In figure 4.16 an example of the first 3 steps iterations is shown and it can be seen that the piecewise linear function $\hat{f}$ becomes a better approximation of the real function in every iteration. Because of the appearance, $\hat{f}$ is often called saw-tooth cover.

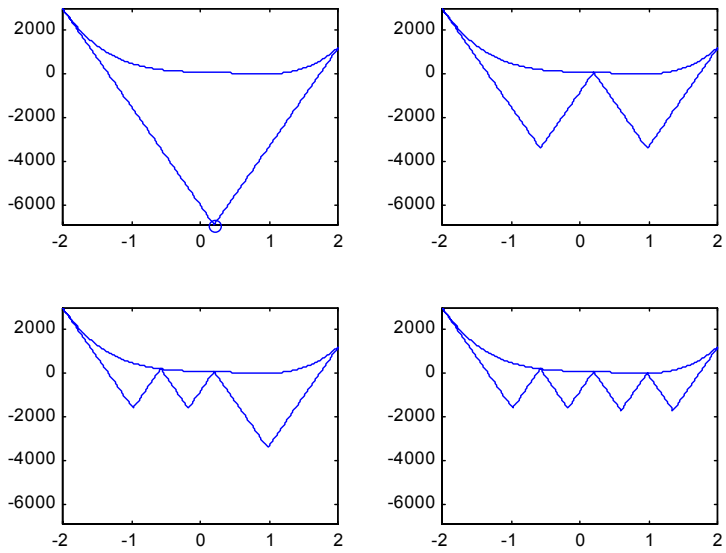


Fig. 4. 16

However, there are two problems to deal with:

- In higher dimensions this algorithm has the need to store 2^N corners for each Hyperrectangle, where N is the dimension of the problem and more restrictive, has to evaluate the functions at all these points.
- In applications the Lipschitz-constant is normally not known if the function itself is a complicated problem or a simulation.

These problems are addressed by the DIRECT algorithm, where DIRECT stands for *dividing rectangles*. This algorithm was developed by D.R. Jones, Perttunen and B.E. Stuckman [Jon-93].

Inequality of equation 4.48 for the one-dimensional formulation will be used to illustrate the procedure. Let $c=(a+b)/2$ and set $x'=c$. Then for $\forall x \in [a,b]$

$$\begin{aligned} x \geq c &: f(c) - \gamma(x-c) \leq f(x) \leq f(c) + \gamma(x-c) \\ x < c &: f(c) + \gamma(x-c) \leq f(x) \leq f(c) - \gamma(x-c) \end{aligned} \tag{4.50}$$

These two inequalities define an area in which the function has to lie. Furthermore a lower bound $D(a,b)$ is found for f in $[a,b]$ by setting $x=a$ or $x=b$ in these inequalities:

$$D(a,b) = f(c) - \gamma(b-a)/2 \tag{4.51}$$

In figure 4.17 this area (shaded) is shown for the cost function used so far.

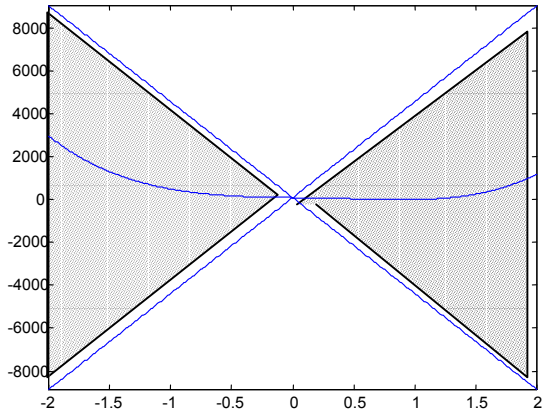


Fig. 4. 17

One major issue for the DIRECT algorithm is how to divide the interval into subintervals. It divides the interval into three subintervals. Three are needed enabling those which have been evaluated already to be saved; the point is a middle point of these intervals. This means that in every step the function has to be evaluated at two new points. Figure 4.18 shows the dividing strategy for the example.

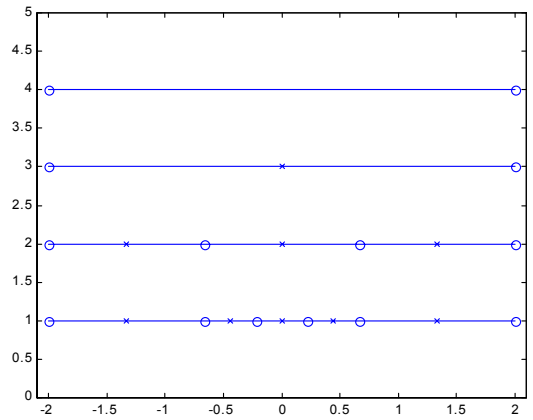


Fig. 4. 18

The next question is how to decide which interval to choose and if this can be done without knowing the Lipschitz-constant γ. Therefore suppose the minimisation area has already been partitioned in M intervals $[a_i,b_i]$ with centre in c_i. A graph is created with the length $(b_i-a_i)/2$ of each of these intervals on the x-axis and $f(c_i)$ on the y-axis, as shown in figure 4.19. If a line with slope γ is laid through each of these points, the intersection of this line and the y-axis is the point $(0,D(a_i,b_i))$, where $D(a_i,b_i)$ is the lower bound for the function in this interval. Therefore it is clear that the interval with the lowest value of $D(a_i,b_i)$ will be chosen as the one in which the algorithm will sample

in the next iteration. This is the methodology used to decide that the interval of the middle had to be divided in third iteration as shown in figure 4.18.

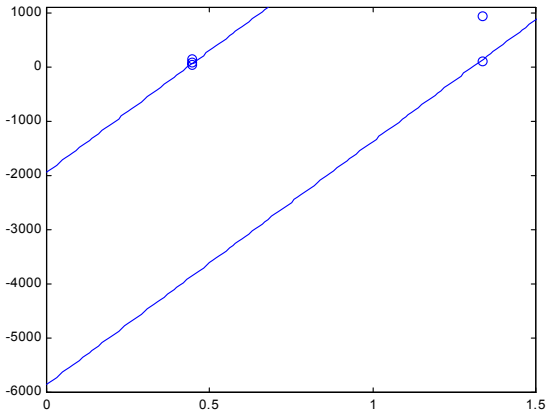


Fig. 4. 19

As stated before, in many applications the Lipschitz constant γ is not known. Therefore an estimation of such a constant has to be found based on the known data. This is done by looking at the convex hull of data-points. For all data-points at the lower side of the convex hull there exists some constant $\tilde{K}_i$ such that the corresponding interval would be the choice if $\gamma=\tilde{K}_i$. It can be easily seen that from several intervals with the same length only the one with the lowest function-value at the middle point can be chosen. A formal definition of this is given in definition 4.3. First inequality expresses this property; second inequality forces the possibility of having sufficient decrease in the interval.

definition 4.3:

Let $\varepsilon>0$be a positive constant and f_{min} be the current best function value. Interval j is said to be potentially optimal if there exists some rate of change constant $\tilde{K}>0$ such that

$$f(c_j)-\tilde{K}\left[(b_j-a_j)/2\right]\le f(c_i)-\tilde{K}\left[(b_j-a_j)/2\right],\forall i$$

$$f(c_j)-\tilde{K}\left[(b_j-a_j)/2\right]\le f_{min}-\varepsilon\left|f_{min}\right|$$

Taking all this together, the one-dimensional DIRECT algorithm can be described as follows:

Algorithm 4.5

1. *m=1,c_1=(a+b)/2*
2. *evaluate f(c_1), f_{min}=f(c_1), t=0*
3. *Do while t<numit and m<numfunc*
   a. *Identify the set S of potentially optimal intervals*
   b. *DO while S≠0*
      i. *Take j∈S*

ii. *Sample new points (c_{m+1},c_{m+2}), update borders*
iii. *Evaluate $f(c_{m+1})$, $f(c_{m+2})$, update f_{min}*
iv. *Set M=m+2, S=S\{j}*

c. *t=t+1*

Once again in the algorithm the first two steps are the initialisation. The variable *m* is a counter for the number of function evaluations done and *t* is a counter for the number of iterations. There are no better termination criteria than to stop after *numit* iterations or after *numfunc* function evaluations. The identification of the potentially optimal intervals is done using definition 4.3.

Figure 4.20 shows the surface of the cost function used so far in this section but with two variables, flow and roughness. In figure 4.21 the contour shows that this surface is not convex and has very flat zones.

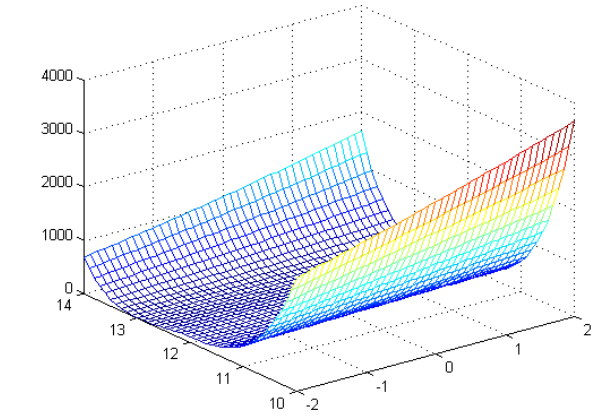


Fig. 4. 20

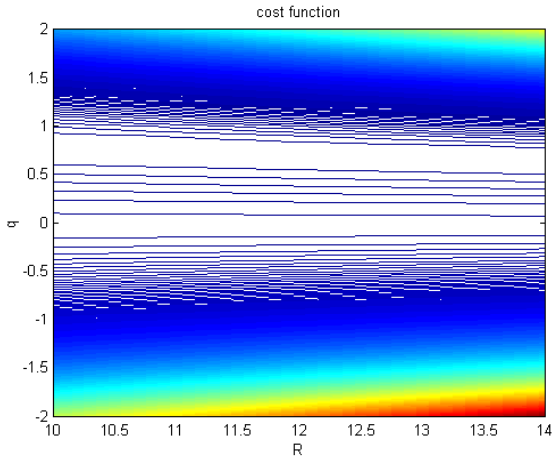


Fig. 4. 21

The main difference in higher dimension compared to the 1-D case is the problem of the division of the search space. For better understanding firstly the hypercube is described and then the more general case of a hyper-rectangle.

1. Let *c (13,0)* be the centre point of the hypercube, in this example a square *{(12,13),(-1,1)}*. Sample the points $c \pm \delta e_i$, where δ equals 1/3 of the side-length of the cube and e_i is the *i*-th Euclidean base-vector. Define w_i by

$$w_i = min\{f(c+\delta e_i\}, \{f(c-\delta e_i\}$$

Divide then in the order given by w_i, starting by the lowest w_i. This means the hypercube is first divided along the vertical direction as $w_2=0.05$. The remaining area is divided in the direction of the second lowest until it has been divided along all directions. This is showed in figure 4.22.

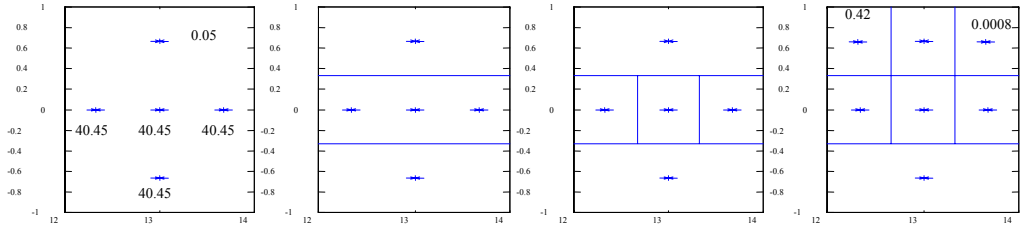


Fig. 4. 22

2. A hyper-rectangle is divided along the longest side, which assures that the maximal side-length decreases. The last step divides the hyper-rectangle with lowest value of the function in the longest side.

4.5.3. The solver: Tomlab

The Tomlab optimisation environment for MATLAB is a flexible and reliable environment for the solution of most types of applied optimisation problems. Tomlab /SOL was written by Professor Kenneth Holmström, in cooperation with the Systems Optimisation Laboratory (SOL) at Stanford and UC San Diego. It runs on a matlab 5 or later version.

The toolbox includes two solvers for global optimisation, which are possible to mix together in the solution process. The routine *glcSolve* implements an extended version of DIRECT that handles problems with both non-linear and integer constraints. Since no Lipschitz is used there is no natural way of defining convergence (except when the optimal function value is known). Therefore *glcSolve* is run for a predefined number of function evaluations and considers the best function value found as the optimal one. It is possible for the user to restart *glcSolve* with the final status of all parameters from the

previous run. DIRECT does not exactly handle equality constraints but the inequality constraint can be used equivalently.

The example of two-reservoirs has been identified using 39 iterations and 6.65 CPU time. Barcelona network from figure 4.8 has been identified As the results seemed to be promising another more complex network has tried to be identified. No better results than those obtained by the local optimisers were found.

4.5.4. Interval Newton Methods

Interval arithmetic was introduced in its modern form by R. E Moore (Moore 1966). If $\boldsymbol{x}$ and $\boldsymbol{y}$ are elements of I($\mathbb{R}$) then the four elementary operations for such idealised interval arithmetic are defined:

$$
\begin{aligned}
&\mathbf{x} = \left[\underline{x}, \overline{x}\right] \\
&\mathbf{y} = \left[\underline{y}, \overline{y}\right] \\
&\mathbf{x} + \mathbf{y} = \left[\underline{x} + \underline{y}, \overline{x} + \overline{y}\right] \\
&\mathbf{x} + \mathbf{y} = \left[\underline{x} - \overline{y}, \overline{x} - \underline{y}\right] \\
&\mathbf{x} \times \mathbf{y} = \left[min\left\{\underline{x}\overline{y}, \overline{x}\underline{y}, \underline{x}\underline{y}, \overline{x}\overline{y}\right\}, max\left\{\underline{x}\overline{y}, \overline{x}\underline{y}, \underline{x}\underline{y}, \overline{x}\overline{y}\right\}\right] \\
&\frac{1}{\mathbf{x}} = \left[1/\overline{x}, 1/\underline{x}\right] \text{if } \underline{x} > 0 \text{ or } \overline{x} < 0 \\
&\mathbf{x} \div \mathbf{y} = \mathbf{x} \times 1/\mathbf{y}
\end{aligned}
\tag{4.52}
$$

Interval arithmetic has some main difficulties that have to be taken into account; ordinary interval division is not always defined, over-estimation when there is multi-incidence of a intervalar variable, Interval arithmetic is subdistributive.

If such arithmetic is implemented on a computer using, equations 4.52, then the default rounding may result in non-rigor. Nonetheless, with *directed rounding*, such bounds can be computed rigorously. In particular if instead of rounding to the nearest, the lower bound of the interval is rounded down to the largest machine number less than the exact range, and the upper bound is rounded up to the smallest machine number greater than the actual result, then the computed interval necessarily contains the exact range. *Simulated Directed Rounding* is a device with which rigorous enclosures to floating point results may be obtained in FORTRAN-77 and other languages that do not have access to true directed rounding, or in machines without hardware-supported directed rounding. This is one example of the difficulties that interval arithmetic implies and has to be taken into account when the software and hardware are chosen.

Example 4.4: Imagine a cost function like the one presented in equation 4.7 with only one variable, the level of a reservoir and its measurement. The value of this variable has a range so the range of this cost function can be evaluated using interval arithmetic presented in equation 4.52.

$$J = (120 - h_{r1})^2 = (120 - h_{r1})(120 - h_{r1})$$
$$h_{r1} = [100,130]$$
$$J = [-200,400]$$

In this example appears clearly the effect of multincidence. The product $h_{r1}*h_{r1}$ could never give a negative value unless it had been evaluated using intervalar arithmetic. The term *standard function* means a set of functions, such as the specified in FORTRAN-77 standard, that are bundled with a particular compiler or software package.

Definition 4.4: *A function* $\mathbf{f}:\mathbf{I}(\mathbb{R})\rightarrow\mathbf{I}(\mathbb{R})$ *is said to be an interval extension of* $f:\mathbb{R}\rightarrow\mathbb{R}$ *provided*

$$\{f(x) \perp x \in \mathbf{x}\} \subseteq \mathbf{f}(\mathbf{x})$$

for all intervals $\mathbf{x}\subset\mathbf{I}(\mathbb{R})$ *within the domain of* $\mathbf{f}$*. (Interval extensions are defined similarly for* $\mathbb{R}^n\rightarrow\mathbb{R}^n$*)*

An early criticism of intervalar arithmetic was that the bounds on the ranges so obtained were too wide or pessimistic to be of value. This is frequently the case when intervalar arithmetic is applied naively, but a good understanding of its properties leads to appropriate and effective algorithms.

Theorem 4.1: *Suppose* $\phi:\Re^n\rightarrow\Re$*, is formally written as a sequence of computations of standard functions* $\{w_j\}^q_{j=1}$ *interspersed with the four basic arithmetic operations. Suppose that in this expression, each of the n variables* $\{x_i\}^n_{i=1}$ *occurs formally only once. Then, if* $\phi(\mathbf{X})$ *is evaluated with exact interval arithmetic and computation of the exact ranges of each* w_j*, the resulting interval enclosure will be the exact range* $\phi(\mathbf{X})$*.*

If in example 4.4 the square were not interpreted as the product of two variables so that the number of incidences could be kept as one, the result would have been [0,400]. So in formulating the optimisation problem and in order to maximise the sharpness of the interval extension it is suggested to minimise the number of incidences of each variable.

Definition 4.5: *If* $F:\Re^n\rightarrow\Re^m$ *is a function computable as an expression, algorithm or computer program involving the four elementary arithmetic operations interspersed with evaluations of standard functions* $\{w_j\}^q_{j=1}$*, then a natural interval extension of F,*

whose value over an interval vector ***X*** *is denoted* ***F(X)****, is obtained by replacing each occurrence of each component* x_i *of X by the corresponding interval component* $\mathbf{x_i}$ *of* ***X****, by executing all operations according to formulas 4.52 and by computing the exact ranges of the standard functions.*

Natural interval extensions are first order. Second order are somewhat desirable, and indeed, almost indispensable in certain contexts. Second-order extensions may be obtained by series expansions and bounding the range of derivatives.

Definition 4.6: *Suppose* $\phi: D \subseteq \Re^n \to \Re$ *has continuous derivatives,* $X \subseteq D$ *and* $X_m \in X$*. Then the mean value extension for* ϕ *over X and centered at* X_m *is defined by*

$$\phi_2(\mathbf{X}, X_m) := \phi(X_m) + \nabla\phi(\mathbf{X})(\mathbf{X} - X_m)$$

where $\nabla\phi(\mathbf{X})$ *is a componentwise interval enclosure for the range of* $\nabla\phi$ *over* ***X.***

It follows from the mean value theorem and the properties of interval arithmetics that $\phi_2(\mathbf{X},X_m)$ is an enclosure for the range of ϕ over **X** and an 2-order interval extension. For the example 4.4.

$$J_2(\mathbf{h_{r1}}, h_{r1}) = (120-115)^2 + 2(120-[100,130])([100,130]-115) = [-575,625]$$

A primary use of interval derivable information is in bounding ranges or variations in functions over regions. This has been seen deriving the second order interval extension. Such information is also crucial in *intervalar Newton Methods* to speed and memory-efficient verified global optimisation algorithms. However, it is not always necessary to use interval extensions of the derivative. In fact, there are techniques that actually lead to tighter bounds. To understand these techniques, it is useful to introduce the concept of a *slope matrix* for a function.

Definition 4.7: *Let* $F: \Re^n \to \Re^m$*. The matrix* ***A*** *is said to be an interval slope matrix (or, more generally, a slope set) for F over , X and centered on the interval vector* $\mathbf{X_m}$ *if, for every* $X \in \mathbf{X}$ *and* $X_m \in \mathbf{X_m}$*,*

$$F(X) - F(X_m) = A(X - X_m) \text{ for some } A \in \mathbf{A}$$

Any smallest such set of matrices satisfying this condition will be denoted by $S^{\#}(F,\mathbf{X},\mathbf{X_m})$*. An interval vector that contains* $S^{\#}(F,\mathbf{X},\mathbf{X_m})$ *(generally a good computed outer estimate) will be denoted by* $\mathbf{S}(F,\mathbf{X},\mathbf{X_m})$*.*

Description of a global optimisation problem requires specification of an objective function and constraints [Kear-96]. However, algorithms for verified global optimisation may require both floating point and interval values of the objective function, gradient and Hessian matrices, as well as, possibly, interval values of the

intermediate quantities obtained in computing a natural interval extension. With floating point and interval values of the constraints and their gradients, this makes at least ten different routines that would be required for each problem, an unacceptable burden if many problems are to be solved. The early approaches used numerical differentiation, others preprocessed using symbolic packages such as Maple. Each of these approaches has failings. Numerical differentiation can be overwhelmed by both truncation and roundoff error. Symbolic differentiation requires substantial separate machinery, and can also result in expressions for the derivatives that are many times larger than the optimal. An alternate technique is *automatic differentiation*.

An object for automatic differentiation is an- ordered pair of the form $<u,u'>$ where the elements u and u' are real numbers or intervals. The rules for *differentiation arithmetic* are based on the elementary rules of differentiation. For example the operations presented before have the following rules

$$\begin{aligned}
&\langle u,u'\rangle + \langle v,v'\rangle = \langle u+v, u'+v'\rangle \\
&\langle u,u'\rangle - \langle v,v'\rangle = \langle u-v, u'-v'\rangle \\
&\langle u,u'\rangle \cdot \langle v,v'\rangle = \langle u\cdot v, uv'+u'v\rangle \\
&\langle u,u'\rangle / \langle v,v'\rangle = \langle u/v, uv'+u'v\rangle
\end{aligned} \tag{4.53}$$

A combination of several techniques is used in state-of-the–art interval global optimisation codes for step 2b of algorithm 4.6. Some of these techniques are outlined as the following algorithm.

Algorithm 4.6 (Range check and Critical Point Verification)

INPUT*: a box* $\mathbf{X}$ *and a current rigorous upper bound* $\overline{\phi}$ *on the global minimum.*

OUTPUT*: one or more boxes derived from* $\mathbf{X}$*, or the information that* $\mathbf{X}$ *cannot contain a global minimum, or information that* $\mathbf{X}$ *contains a critical point.*

1. *(Feasibility check; for constrained problems only)*
   a. *(Exist if infeasibility is proven)* DO *for i=1 to m:*
      i. *Compute an enclosure* $c_i(\mathbf{X})$ *for the range of* c_i *over* $\mathbf{X}$*.*
      ii. IF $0 \notin c_i(\mathbf{X})$ THEN *discard* $\mathbf{X}$ *and* EXIT
   b. *Verify, if possible, that there exists at least one feasible point in* $\mathbf{X}$*.*
2. *(Range check or "midpoint test")*
   a. *Compute a lower bound* $\underline{\phi}(\mathbf{X})$ *on the range of* ϕ *over* $\mathbf{X}$*.*
   b. *IF* $\underline{\phi}(\mathbf{X}) > \overline{\phi}$ THEN *discard* $\mathbf{X}$ *and* EXIT
3. *(Update the upper bound on the minimum.) If the problem is unconstrained or feasibility was proven is Step 1b,* THEN
   a. *Use interval arithmetic to compute an upper bound* $\overline{\phi}(\mathbf{X})$ *of the objective function* ϕ *over* $\mathbf{X}$*.*

b. $\bar{\phi} \leftarrow min\left\{\bar{\phi}, \bar{\phi}(\mathbf{X})\right\}$.

4. *("monotonicity test")*

a. Compute ab enclosure $\nabla\phi(\mathbf{X})$ *of the range of* $\nabla\phi$ *over* ***X****.. (Note: if* ***X*** *is "thin", i.e. some bound constraints are active over* ***X****, then a reduced gradient can be used)*

b. IF $0 \notin \nabla\phi(\mathbf{X})$ THEN *discard* ***X*** *and* EXIT.

5. *("concavity test") If the Hessian matrix*[2] $\nabla^2\phi$ *cannot be positive definite any-where in* ***X*** THEN *discard* ***X*** *and* EXIT.
6. *Quadratic convergence and computational existence/uniqueness) Use an interval Newton method (with Fritz John equations in the constrained case) to possibly do one or more of the following:*

reduce the size of ***X****;*
discard ***X****;*
verify that a unique critical point exists in ***X****.*
(Bisection or geometric tessellation) If Step 6 did not result in a sufficient change in ***X****, then bisect* ***X*** *along a coordinate direction)(or otherwise tessellate* ***X****), returning all resulting boxes for subsequent processing.*

End Algorithm 4.6

4.5.5. The solver GlobSol

The Global Solution initiative of sun Microsystems is intended to develop R. Baker Kearfott's INOPT_90 software for solving non-linear systems and global optimisation problems into a commercial quality package. It has been used for the networks treated so far. The results didn't seem to guarantee the solution of any network. The cause of this problem may be more in the software than in the algorithm. The development of this software is on-going and it is difficult to say whether the improvement of such tools will overcome these difficulties.

4.6. The recursive least square method

The problems presented by a global optimisation solution for the estimation of parameters of the network lead to the investigation of other possibilities. Andersen [And-00] proposes a least square methodology for the state estimation. The non-linear system is linearised and the Jacobian is the base for the least square estimation problem. Bertsekas [Ber-96] presents extended Kalman filter. It is called extended because it treats with non-linear systems. These problems are extended in time in order to identify systems that would be unidentifiable with one time step. Though it is normal to think in Kalman filter for dynamic systems it can be used for random variables in order to find a mean value. Using some notes by professor Astrom [Ast-95] this point of view seems possible.

Consider two jointly distributed random vectors x and y. x is viewed as a measurement that provides information about x. What is searched is a function $f(\cdot)$, called estimator,

where *f(x)* is an estimate of *y* given *x*. A commonly used estimator is the *least squares estimator*, which is obtained by minimisation of the following expected squared error over all *f(·)*:

$$E\left[\left\|y - f(x)\right\|^2\right] = E\left[E\left[\left\|y - f(x)\right\|^2 \mid x\right]\right] \tag{4.54}$$

it is clear that *f*(·)* is a least squares estimator if *f*(x)* minimises the conditional expectation in the right-hand side above for every *x*. Suppose that the least squares estimator *E[y|x]* is very complicated or otherwise difficult to calculate. It would then make sense to consider instead minimisation over *r* of

$$E\left[\left\|y - h(x,r)\right\|^2\right]$$

over a class of estimators from a given architecture *h(·,r)* that depends on a parameter vector *r*. Assume further that there are sample pairs (x_1,y_1),..., (x_m,y_m). The preceding expected value can be approximated by

$$\frac{1}{m}\sum \left\|y_i - h(x_i,r)\right\|^2 \tag{4.55}$$

Then if *r** minimises the above sum of squares, *h(x,r*)* is an approximation to the least squares estimator *E[y,x]*.

Optimality conditions impose the convexity of the function. Such convexity can be formulated using the Hessian matrix and imposing its positive semidefinition. Such condition cannot be assured in the problems that are treated in this thesis. The approach used is to linearise the problem and treat it as a Linear Least squares problem. Consider the least squares problem for the case that the function *g(r)* are linear of the form

$$g_i(r) = y_i - X_i r \tag{4.56}$$

where $y_i \in \Re^m$ are given vectors and X_i are given $m \times n$ matrices. In other words, what is done is to fit a linear model to the set of input-output pairs (x_1,X_1),..., (x_m,X_m). Each such pair is a *data block*. The cost function has the form

$$f(r) = \frac{1}{2}\sum_{i=1}^{m} \left\|y_i - X_i r\right\|^2 \tag{4.57}$$

it is quadratic and convex. If the matrix of equation 4.47 is invertible the problem has a unique solution *r**, obtained by setting the gradient to zero. This linear square estimate can also be computed recursively using the *Kalman Filter*. This algorithm has many important applications in control and communications systems, and has been studied extensively. An important characteristic of the algorithm is that it processes the least squares terms $||y_i - X_i r||$ incrementally, one at a time. Consider equation 4.45. The *Kalman*

Filter sequentially generates the vectors that solve the partial least squares subproblems involving only the first *i* squared terms, where *i*=1,...m:

$$\Psi_i = \arg\min_{r\in\Re^n} \sum_{j=1}^{i} \left\| y_j - X_j r \right\|^2 \tag{4.58}$$

Thus the algorithm first finds ψ_1 by minimising the first term $\|y_1-X_1r\|^2$, then finds ψ_2 By minimising the sum of the two terms $\|y_1-X_1r\|^2+\|y_2-X_2r\|^2$, etc. The solution of the original least squares problem is obtained at the last step as $r^*=\psi_m$. The method can be conveniently implemented so that the solution of each new sub-problem is easy given the solution of the earlier sub-problems, as will be shown in equation 4.48.

$$minimise \sum_{j=1}^{m} \lambda^{m-j} \left\| y_j - X_j r \right\|^2 \tag{4.59}$$

subject to $r \in \Re^n$ and $0 \le \lambda \le 1$

When λ=1 the original least squares is obtained, a λ<1 de-emphasises the effect of the initial terms and is sometimes useful in the situations where the mechanism that generates the data changes slowly with time.

Algorithm 4.7 shows the implementation of this technique assuming that $X_i'X_i$ is positive defined. Such assumption will be analysed when the special case of water networks is treated.

Algorithm 4.7

1. *Set $H_0=0$, $\psi_0=r_0$ and i=1*
2. *DO for i=1 to m*
   a) $H_i=\lambda H_{i-1}+X_i'X_i$
   b) $\psi_i=\psi_{i-1}+ H_i^{-1} X_i'(y_i- X_i\psi_{i-1})$
   c) *i=i+1*

Example 4.5: Imagine that the relation between two vectors in this linear example is a constant matrix. Vector y is measured, it has a small noise (≈1%) and the two parameters r one is constant and the other one increase linearly with time.

$$y_i = Xr_i = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} r_{1i} \\ r_2 \end{bmatrix}$$

The use of the algorithm 4.7 with 100 iterations allows the proper estimation of the constant parameter. It can be seen in figure 4.23 where the continuous line is the real and the dashed line corresponds to estimation.

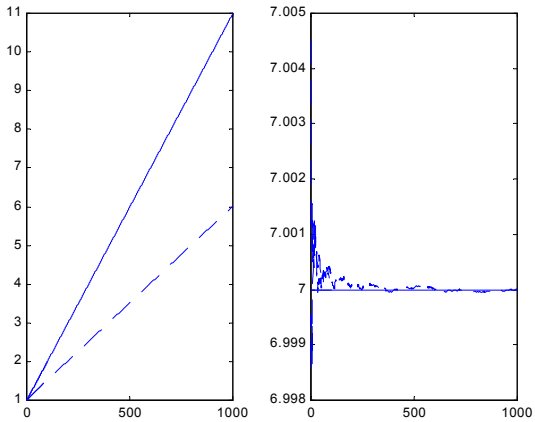


Fig. 4. 23

4.6.1. Recursive Linear Least squares applied to the water networks

In order to apply this technique to the problem that is treated here, estimation of parameters in water network has to be linearised. Function g of equation 2.4 represents equations 4.7 to 4.9, physical constraints in a water network, using the general notation used in example 2.2. This equation has been linearised becoming 2.7. Equation 4.60 shows the incremental problem that arises when a system of equations is linearised.

$$z_0 + \Delta z = g(x_0) + J_{x_0}\Delta x + \xi \Rightarrow \Delta z = J_{x_0}\Delta x + \xi \tag{4.60}$$

Looking at equation 4.55, vector *y* corresponds to the difference between the initial model simulation results and the measurements taken. Jacobian is calculated using the parameters and the unknown variables that are obtained by simulation of the original model. The result of the estimation Δx is the increment of the vector x (parameters and variables) done to the original model in order to obtain a better model.

One of the main advantages of this method is that the identifiability study developed in chapter 2 has an obvious interpretation in this formulation. If the Jacobian is full rank and its rank is higher than the number of variables and parameters that have to be estimated the pseudo inverse exists and the problem has a solution. This condition is the same condition imposed by the positive definition of $X_i'X_i$.

The linear least square method has no convergence problems because of the convexity. The reliability of the result will depend on the reliability of the linear approximation. The recursivity of the method helps in improving this reliability. This recursivity is understood in three different ways shown by figures 4.24, 4.26 and 4.28. These three approaches are illustrated with a simple example and the improvement of recalculating the working point in each iteration is shown.

Example 4.6: The network of figure 2.3 was identifiable in one time-step when all heads are known (reservoirs and junctions) and two flows are measured (q1 and q4). Using the recursive least square state and parameter estimation presented in algorithm 4.7, where X_i is now the Jacobian, the estimation of the roughness and demand factors (all unknown) is carried out. In figure 4.25 these parameters are presented, the correct ones (o), the estimated (·) and the original model (+).

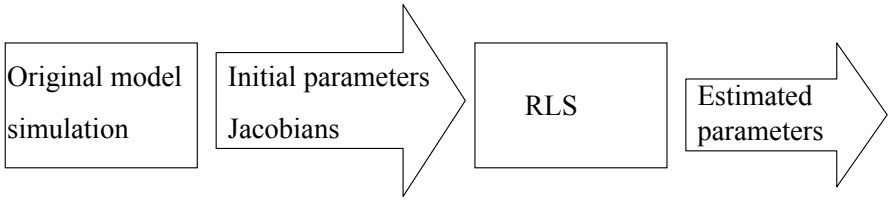


Fig. 4. 24

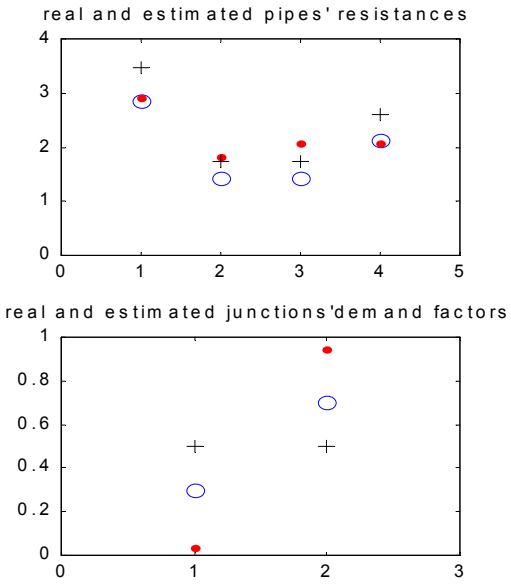


Fig. 4. 25

The parameters improve but it could not be taken as a good result. Increasing the number of time steps does not necessarily improve the estimation because as the time passes the jacobian, which uses variables in its calculation, becomes less and less reliable. The use of two consecutive days may introduce little improvement as the working point is the same and the problem becomes numerically repetitive. The rls has been repeated changing the original model that provides the jacobian as figure 4.26 shows. What has been done is the introduction of the estimated parameters as original ones and then the calculation of the Jacobian is iteratively improved. In figure 4.27 a second estimation is showed.

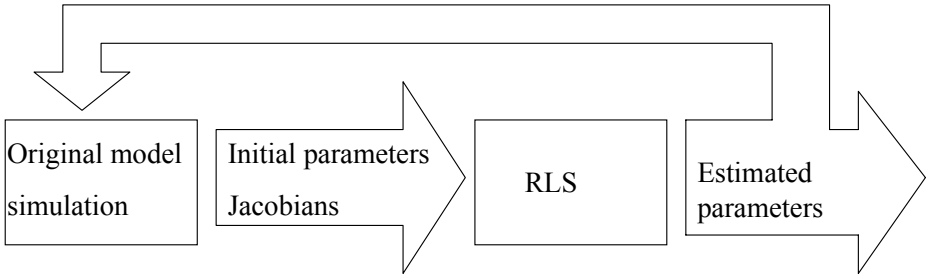


Fig. 4. 26

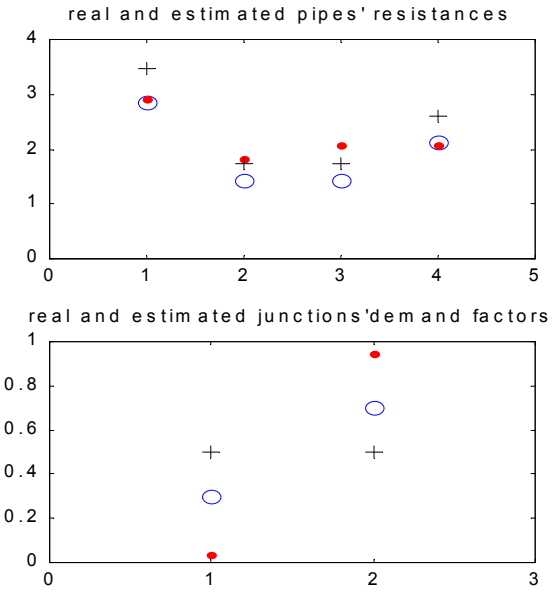


Fig. 4. 27

Little improvement is seen. The third approach is to recalculate the jacobian in each iteration. This occurs properly in the rls. The simulator requires updating in each iteration. Such updating can be done using Epanet-Toolkit that works as an interface between the simulator and other applications.

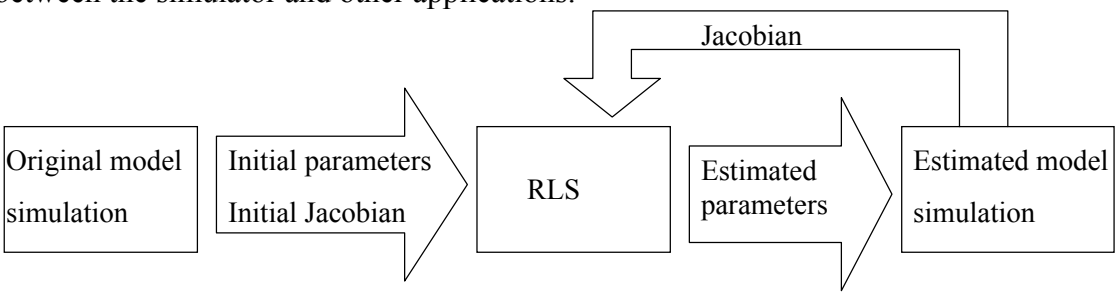


Fig. 4. 28

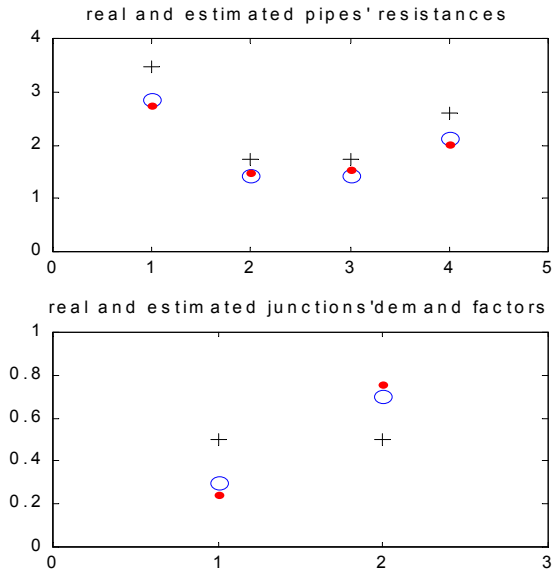


Fig. 4. 29

When the system is not identifiable with one time-step matrix H is not invertible any more. On the other-hand it has been seen that the use of more than one time step can increase the identifiability and thus this method should be generalised. The generalisation of the jacobian presented in section 2.6.3 was easily calculated due to the problem formulation used. Taking into account as many time steps as identifiability requires and using such jacobian as the linearisation of a generalised non-linear problem, Extended Kalman Filter can be applied.

The Extended Kalman Filter understood as the application of the linear Kalman Filter to the linerised system, following scheme presented in figure 4.28, is identical to the formulas 2.12 presented in section 2.4.2. In that section the convergence conditions were used to determine the identifiability of the network.

Example 4.7: The network of figure 4.8 where local optimiser failed has been used to test this method. In figure 4.30 the evolution of the parameter is shown. There are four resistances that present a wrong evolution, demand factors converge better. This has been obtained with 10 time steps. Figure 4.31 shows the real value (o), initial value (+) and estimation (*).

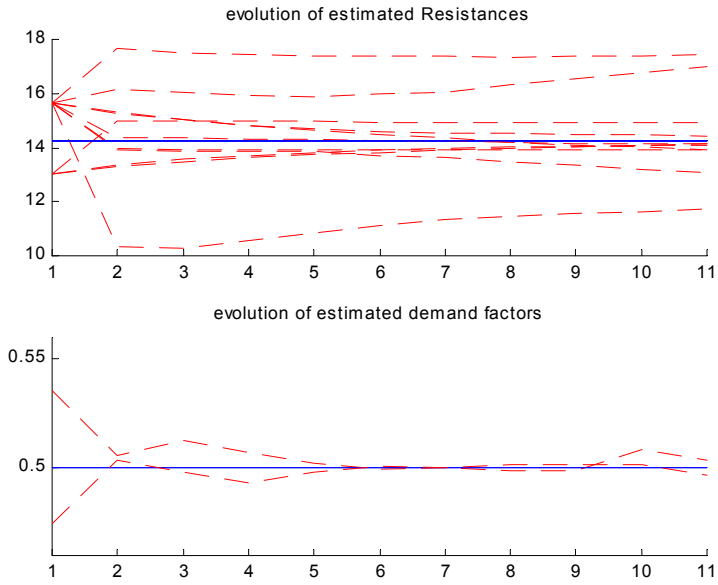


Fig. 4. 30

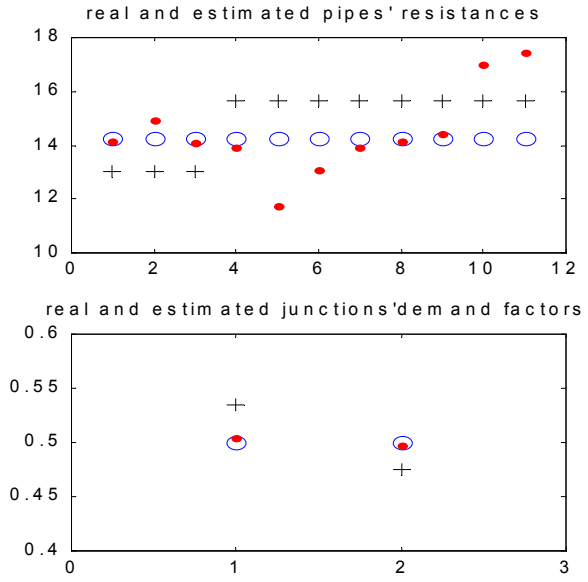


Fig. 4. 31

In order to improve the estimation based on local optimisation a search of new initial conditions has carried out. This has improved the results and a combination of these two techniques is proposed for such problems. In figure 4.32 the new results are shown.

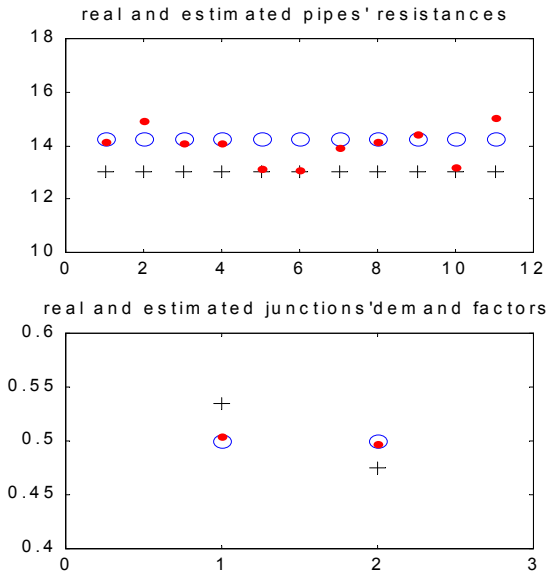


Fig. 4. 32

4.7. Case Study

The experience of WSS with real networks and their contacts had provided some real data and a real network model. They have allowed the application of the techniques proposed in this thesis. Figure 4.33 shows a real network. It has 1 reservoir, 236 junctions and 243 pipes.

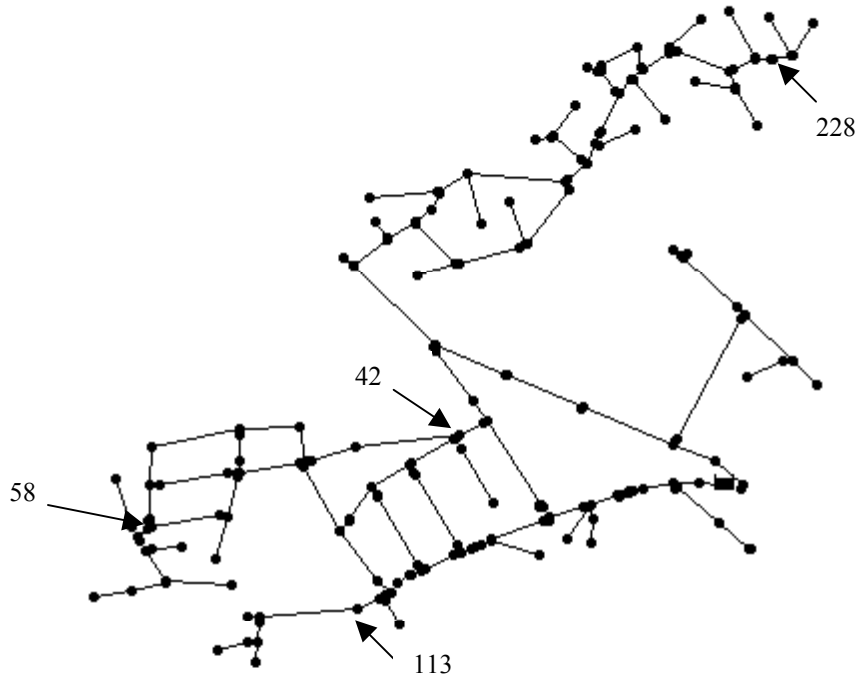


Fig. 4. 33

Data are recorded at 20 second intervals. The inlet is measured. Head in reservoir is known and thus it is represented as a curved head reservoir. In figure 4.34 both measurements (continuous) and simulations (dashed) appear as the same curve because it is introduced as an input that is known. There are four more heads that are measured; Average zonal node pressure (AZNP that corresponds to junction 42) presented in figure 4.35; target point pressure (TP, junction 58) presented in figure 4.36; critical point pressure (CP, junction 228) represented in figure 4.37; pressure measurement point 1 (PM1, junction 113). In this thesis the steps are 1 hour.

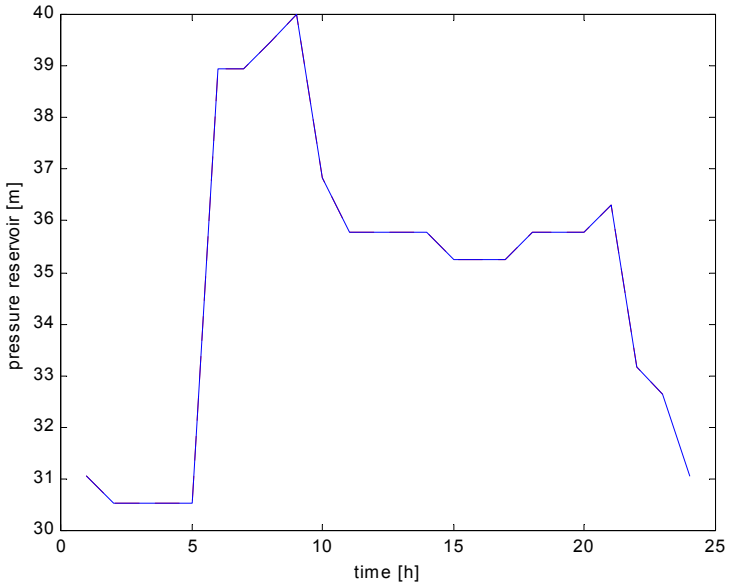


Fig. 4. 34

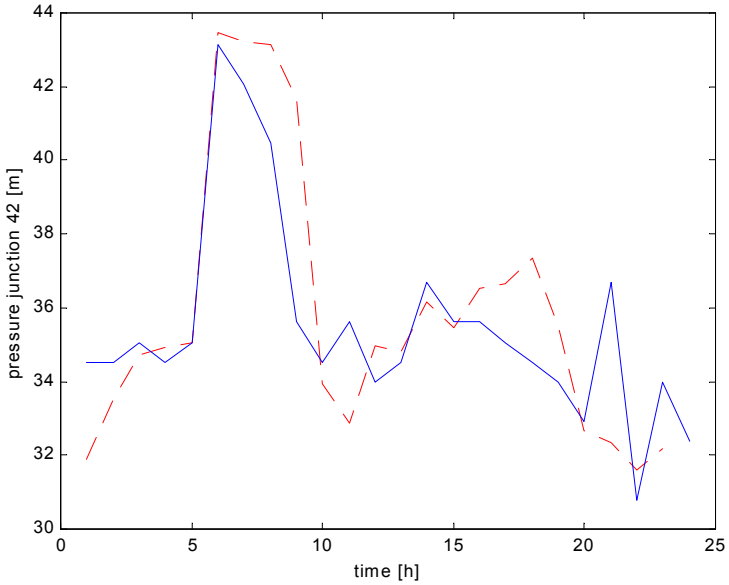


Fig. 4. 35

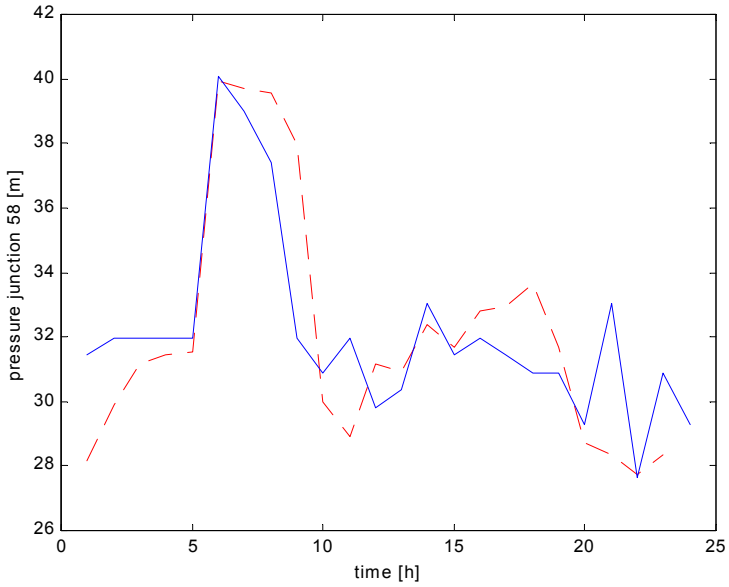


Fig. 4. 36

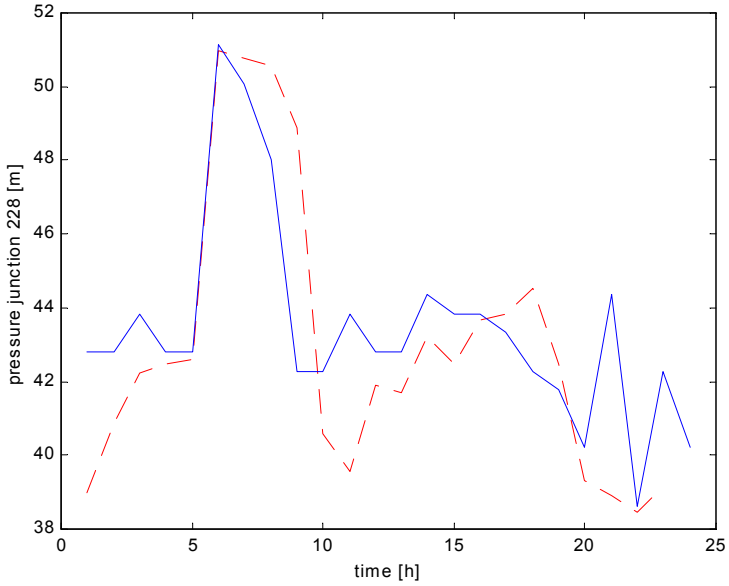


Fig. 4. 37

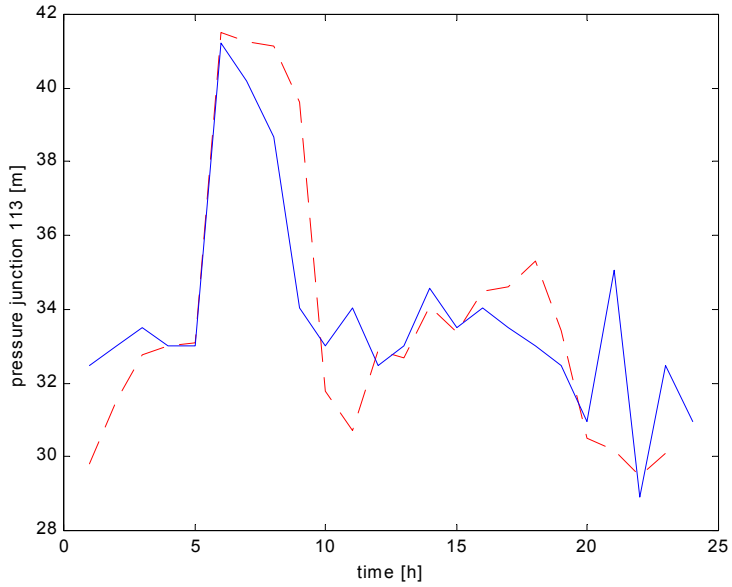


Fig. 4. 38

The first attempt to improve the model was using microcalibration. The extended Kalman Filter approach has been applied to this case study. The few measurements provided and the enormous size of the network constrained the resistances and demand factors that could be tuned. The identifiability study as presented in chapter 2 allowed to know these limits. The attempts to apply the microcalibration left almost unchanged the parameters. This is not a bad result if the head curves are observed carefully. The error between simulation and measurements is very similar in each junction in spite of their location (they are in each edge of the network). A correction in roughness or demand factors would change differently these curves for each junction.

On the other side some experiments have been carried out. First the roughness of pipe 75 (C-factor in the simulator) has been multiplied by 10. This pipe is attached to junction 42. Figure 4.38 shows the new pressure comparison for junction 42. It is almost identical to that in figure 4.34. If the demand factors of junctions 42 (0.04 in this network) and 58 (0) are exchanged the effect is not much stronger. Figures 4.30 and 4.40 show the pressures comparisons for junctions 42 and 58 in this new situation.

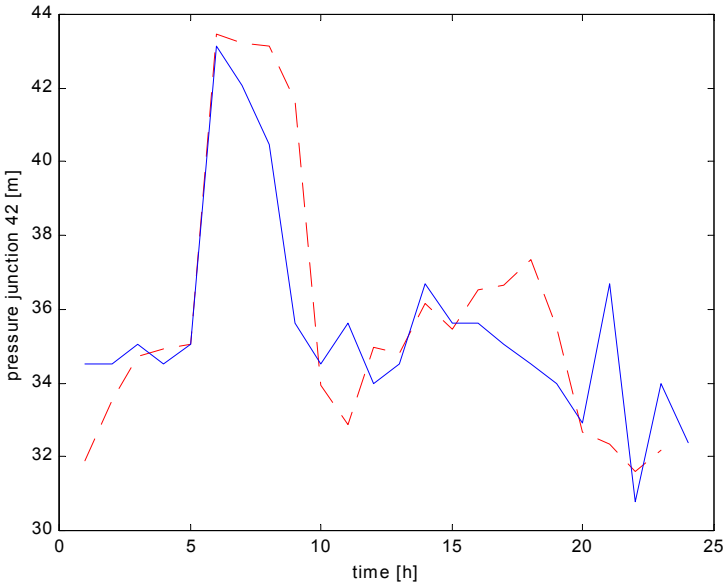


Fig. 4. 39

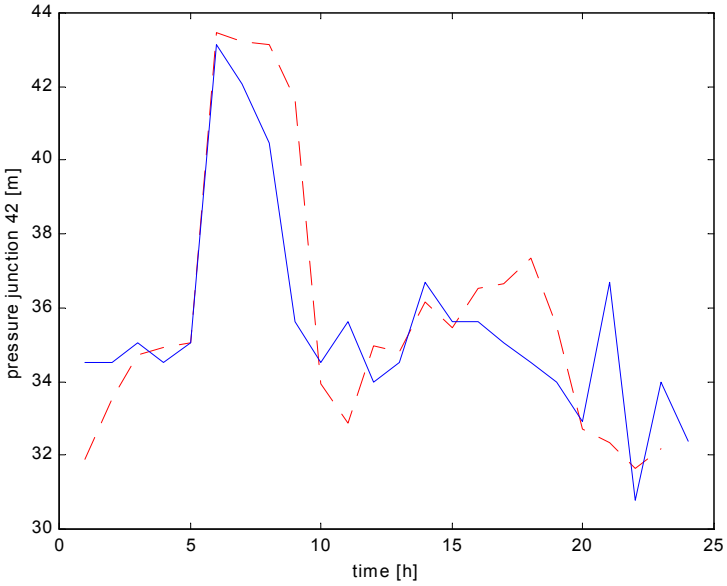


Fig. 4. 40

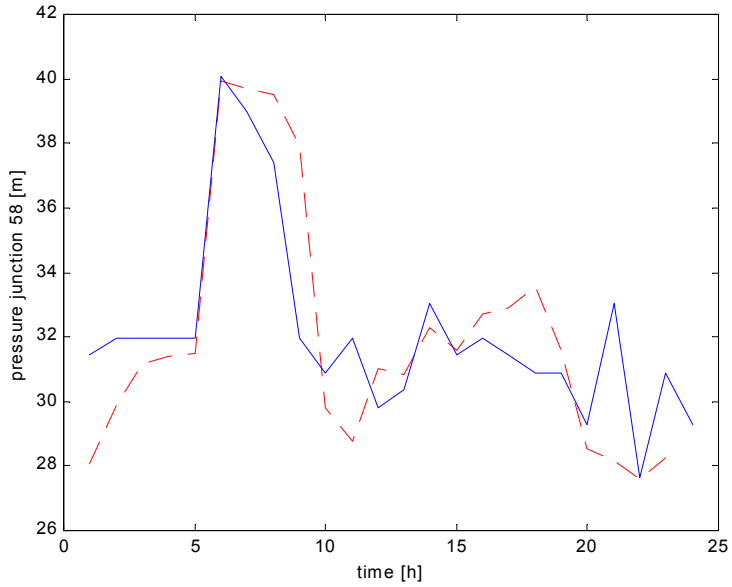


Fig. 4. 41

It is clear that the cause of the discrepancies is a global error in the model. There are two inputs in the model. The head of the reservoir is a measurement and thus it is as reliable as the measurements in the junctions. The other input is the demand curve that affects all junctions with demand associated. It is not a measurement but a prediction. Such an error in modelling would have been detected comparing the SCADA data for consumptions and this demand curve as it is explained in chapter 3. The real data are not available and a readjustment is done in this curve and the result was an important improvement in the pressure simulation. Figures 4.42 to 4.45 show the new comparisons for pressures in the junctions where it is measured. The conclusion of this real case is that a good macrocalibration is necessary in order to apply microcalibration techniques. Any improvement of demand factors and roughness would require more data. The small influence of tuning is unique resistance or demand factor is due to the small number of measurements.

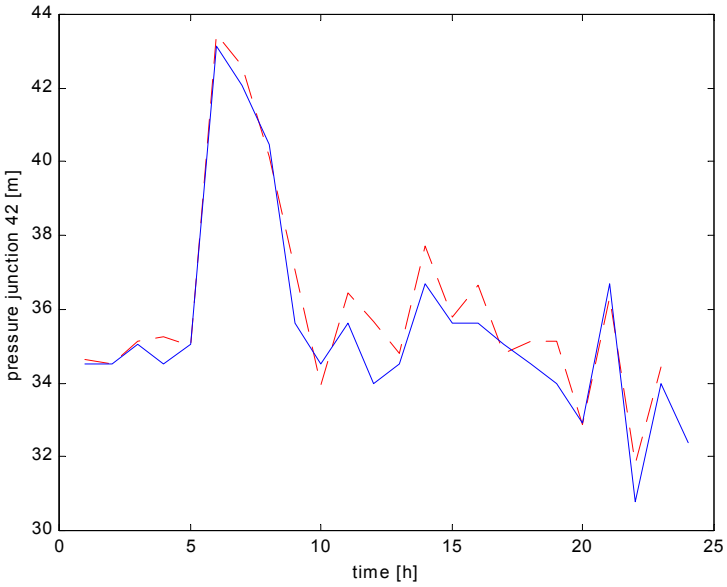


Fig. 4. 42

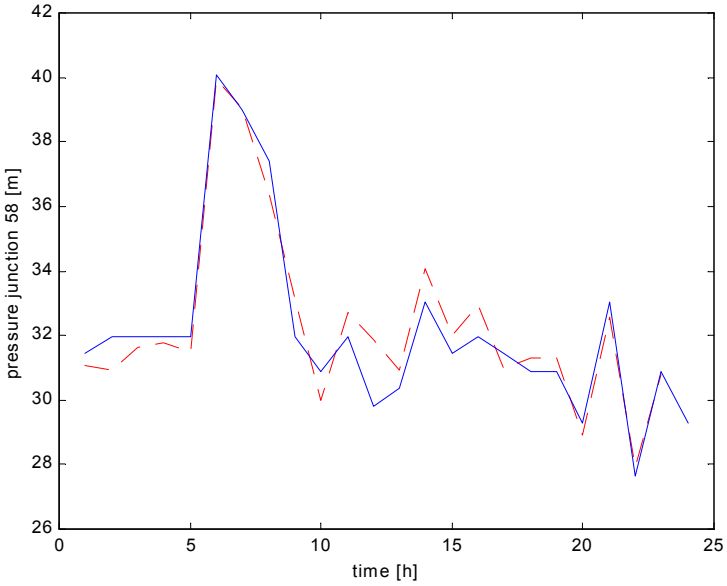


Fig. 4. 43

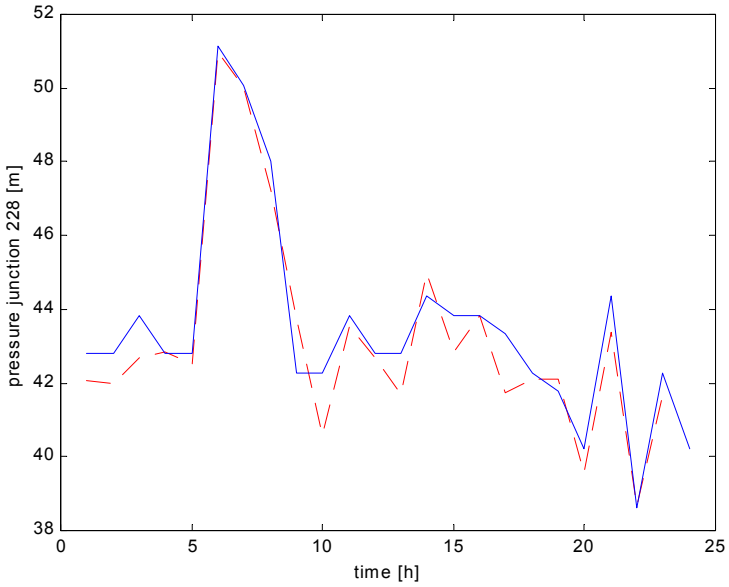


Fig. 4. 44

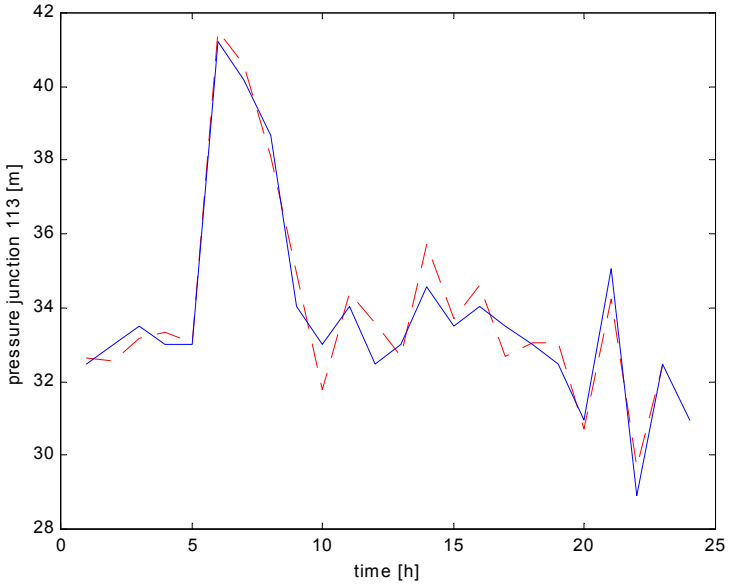


Fig. 4. 45

5. Conclusions

The general objective of this thesis was to face a major problem for the water distribution systems' management. The importance and criticality of calibration in such systems was justified in the introduction. In order to meet such a challenge, the first idea was to study the application of global optimisation techniques. Experience in optimisation of such systems and the non-convexity of the problem leads to these algorithms. The theoretical study of the identifiability and the problem design needed and the elimination of major errors in the network using some artificial intelligence techniques are two other main parts of this work.

The calibration problem is presented as the identification of a system. **Input** has to be introduced within a **time-period** in order to obtain the model from the **output**. These three factors are crucial for the **observability** of a system. The identifiability can be treated as a generalisation of observability. Definition and understanding of identifiability is provided. The importance of this analysis is justified.

State-of-the-art in calibration of networks and systems has been studied and applied to water networks. Although different approaches are found in the literature, the improvement introduced by repeating the measurements (quasi-static simulation with more than one time-step) is not well covered. The linear case is just a linear algebra problem that appears in many basic books. In general the problem is non-linear and the jacobian is used in order to linearise. The topologic identifiability uses graph theory and an enlightening example is presented.

The equations for the linear and non-linear case are developed and the identifiability conditions are generalised for extended period case. For the linear case some conditions are imposed on the input. This means that not only whether the quantity of sensors is enough and well distributed, and how many time steps must be taken into account is studied, but also the measurement conditions. The condition is a sufficient one but not necessary.

Although a topological identifiability study would be useful in order to decide the location of new sensors, this has appeared to be not possible to generalise without introducing the reservoir and their dynamics. The extended-period jacobian is a good method for the identifiability study. In order to get topological information about the non-identifiable variables the singular value decomposition is useful. Results using this methodology are coherent with the ones obtained using topological identifiability for a static case. This methodology is easily generalised using the formulation proposed in this thesis.

The design of the identification problem goes beyond the mere identifiability study. One aim of the thesis was to give a useful methodology for the sensor placement, a major challenge. The use of singular value decomposition in their interpretation as input and output directions and gains, allows the signalling of those variables that are most critical in the identification.

A big network has been used as a case study. The introduction of extended period measurements has much improved the identifiability allowing the use of fewer sensors. The results are promising. Three case studies have been used to show some characteristics of the identification problem depending on the topology. The number of sensors required when no loop is present is greater than the number required in networks with loops, comparing when both networks have the same number of nodes or same number of pipes. This could have been found out using a topological approach for one time step but not in the extended-period case.

The modelling of the water network using optimisation techniques for the parameter estimation may have convergence problems unless **macrocalibration** is performed first to remove crude errors. The first task has been to organise the errors that the experts aim to detect in this phase. Some of these errors need an easy algorithm to be studied (connectivity, total demand, etc.). Other errors are more subtle and some kind of knowledge organisation has been used in order to detect them.

Two approaches have been considered, expert system and classification. They appear in literature when macrocalibration is intended. The inference engine of such an expert systems does not seem to be especially critical as there are not so many rules enchained, the first prototypes have been implemented using Matlab.

This simplicity is common to the classification algorithms so it hasn't been a major reason for the usage. Classification methods have presented once more applied to examples of simple water networks.

The problems of generating the knowledge for an expert system do not appear using LAMDA (a classification algorithm). This algorithm has worked well when all the

information required was present and for some special cases of errors. All the process for cluster generation, object classification and cluster modification has been presented.

The results obtained applying these techniques are promising. They have been obtained with knowledge generated from a simple network and applied to a much more complicated one. Results would be more reliable if the user generated from his own network or a similar network from the topological and hydraulic point of view. The idea of this tool is not to detect, classify and solve all errors. It is a support tool for the user who has to make there own decisions. In big networks an automatic tool that signals suspicious elements may be very useful.

The last step, though of course not the least in this thesis, has been the study of different techniques for the **microcalibration** process. This process is not understood as an error detection method but one of tuning of parameters and is formulated as an optimisation problem. This formulation and its theoretical study confirm the need of global optimisers due to the non-convexity of the problem.

The comparative study of the methods has been done in a progressive way. First the local optimisers have been tested and have shown themselves insufficient as the problems become slightly complicated. The number of variables explodes as the number of time steps increase. This difficulty is presented by the global solvers that have been tested. Signomial algorithms appeared overwhelmed by the simplest network. Lipschitz optimisation gives results that cannot be guaranteed as global minima with a similar limitation to that of the local optimisers. Interval arithmetics optimiser seemed to give promising results but the software available failed and however the improved versions of it could open a field of application.

As an alternative, a classical algorithm in identification is adapted to the concrete problem. The recursive least squares is used to determine the parameters of the network using one or more time steps (extended period). The linearisation of the network has to be done for each iteration in order to obtain the best results.

The development of the prototype has all been carried out in simulation. The difficulty to obtain real data from companies, when no concrete project is in operation made it impossible to link this work with database and SCADA. Now some projects in the group with companies in Spain and South America, may allow the application of some of these techniques to real cases. On the other-hand the work done together with the Water Software Systems group in Leicester has opened a way of co-operation with it may lead to the development of new projects with industrial orientation.

My personal achievements in this thesis have been to supply a solution for each of the three main steps of the calibration process that have been studied deeply.

A tool for the identifiability is proposed. Such a tool is able to determine whether a network is identifiable with the available measurements and where should be added new sensors. This has a straightforward industrial application and is ready to be used.

A methodology and a tool to support the expert who carries out the macrocalibration process. It is not an automatic process but all the steps suggested by the expert are treated systematically and an algorithm specific has been developed. The use of LAMDA allows the generation of knowledge in a easier way compared with other methods as expert systems.

The parameter tuning has the major drawback that it drives to a non-convex optimisation problem. The original idea of applying global optimisation algorithms has been reduced to the statement of the limitations of these tools. An alternative is suggested based in Extended Kalman Filter. This recursive estimation together with the use of different initial conditions is the suggested method.

The tool for the calibration has been developed for each functionality, identifiability study, macrocalibration and microcalibration. The Web-server toolbox in Matlab allowed the remote macrocalibration and now a integration project is been carried out.

This thesis has been written in order to gather all the tested technologies and justify those that have been chosen. Those that are not so common in water industry have been illustrated with small networks.

As a future work this tool should be applied to a real network. This means an integration effort of SCADA, data base, simulator and the tools developed in this thesis. A customisation of the macrocalibration tool would be convenient. This tool should be adapted to the topology and the sensors available in the network. Such customisation is possible with the guides given in this thesis. The new software that appears in global optimisation could improve the reliability of some of the algorithms specially those based on intervalar algebra. The application of Extended Kalman Filter to a real network needs once again the integration of software and the availability of real data.

6. References

[Agua-98] Aguado J.C. *A Mixed Qualitative-Quantitative Self-Learning Classification Technique Applied to Situation Assessment in Industrial Process.* Thesis presented in Universitat Politècnica de Catalunya 1998.

[Agui-01] Aguilar J. Waissman J. Sarrate R Nejjari F. Dahhou B. *On line expert situation assessment of the biological modes in a waste water plant by means of fuzzy classification.* CIMCA 2001 proceedings. July 2001 Las Vegas. Pp. 339-345.

[And-00] Andersen J.H. Powell P.S. *A Loop Based Simulator and Implicit State-Estimator.* Water Industry Systems Research Studies Press England 2000.

[Ast-95] K.J. Åström, G.C. Goodwin, and P.R. Kumar (editors), *Adaptive Control, Filtering, and Signal Processing,* vol 74, The IMA Volumes in Mathematics and its Applications, Springer-Verlag 1995.

[Ber-97] Bertsekas D. P. Tsitsiklis J. N. *Neuro-dynamic programming* Athenea Scientific Belmont 1997.

[Bla-69] Blau G.E., Wilde D.J. *Generalized polynomial programming*, Canad.J.Chemical Engineering, 47 (1969), pp.317-326.

[Bou-91] Boulos P., Ormsbee L. *Explicit Network Calibration for multiple Loading Conditions.* Civil Engeg. Systems 8:153, 1991.

[Brd-94] Brdys M.A., Ulanicki B. *Operational Control of water Systems: Structures, algorithms and applications.* Prentice Hall 1994.

[Bret-96] Bretas N.G. London J.B.A. *Network Observability: the Critical Measurement Identification using the Symbolic Jacobian Matrix.* IEE

Proceedings, Generation, Transmission and Distribution, Vol. 143 Nº 1, pp. 123-128, January 1996.

[Car-91] Carpentier P., Cohen G. *State Estimation and Leak Detection in Water Distribution Networks*. March 1991.

[Cas-99] Castells S. *Disseny i implementació d'un sistema expert per detectar errors en els models de xarxa de distribució d'aigua.* Ptojecte final de carrera Enginyeria en automàtica i Electronica Industrial. UPC June 1999.

[Cem-97] Cembrano G. Quevedo J. Perez R. *DELIVERABLE DR5.1.- Identification of Optimisation's tasks and criteria* European project WATERNET, Knowledge Capture for Advanced Supervision of Water Distribution Networks, ESPIRIT IV nº 22.186. February 1997.

[Cem-98] Cembrano, G; Quevedo, J; Wells, G; Pérez, R; Argelaguet, R. Advanced Control of Water Distribution Networks: The Waternet Project. WESIC' 98 WORKSHOP ON EUROPEAN SCIENTIFIC AND INDUSTRIAL COLLABORATION ON PROMOTING ADVANCED TECHNOLOGIES IN MANUFACTURING. Centre cívic la Mercè. Girona - ESPAÑA (10.06.1998 - 12.06.1998).

[con-03] http://www.conopt.com/

[Cou-85] Coulbeck B. *An application of hierarchical optimisation in calibration of large-scale water networks.* Optimal Control applications and Methods. June 1985. 6, 31-42.

[Crui-92] Cruickshank J.R. & Long S.J. (1992) *Calibrating computer model of distribution systems.* Proc. AWWA Computer Conf., Nashville, Tenn.

[Don-74] Donachie R.P. *Digital program for water network analysis*. Journal of the Hydraulics Division, ASCE 1974. 100(3), 393-403.

[Duf-67] Duffin J., Peterson E.L., Zener C. *Geometric Programming.* John Wiley & Sons, New York 1967.

[Dur-94] Durkin J. *Expert Systems Design and Development* (1994) Macmillan Publishing Company New York.

[Fal-73] Falk E. (1973) *Global Solutions of Signomial Programs, Technical Report T-274*. George Washington University, Washington DC, 1973.

[Fet-75] Fetzer E.E. Anderson P.M. *Observability in the State Estimation of Power Systems.* IEEE Transactions on Power Apparatus and systems, Vol. PAS-94, no 6. November/December 1975.

[Flou-01] Floudas C.A., Pardalos P.M. *Encyclopaedia of Optimization*. Kluwer Academic Publishers 2001 Netherlands.

[Gab-98] Gablonsky J. *An Implementation of DIRECT algorithm.* North Carolina State University Department of Mathematics. September 1998.

[Geo-03] Geodesys http://www.edie.net/

[Gam-03] http://www.gams.com/

[Gof-81] Gofman E. Rodeh M.(1981) *Loop equations with Unknown pipe characteristics.* Jour. ASCE.- Hydraulics Div, ASCE September 1981. 107:9:1047.

[Gre-03] Greenpeace http://www.greenpeace.es/AGUAS/campagna.asp

[Hol-97] Holmström K. (1998) *TOMLAB- A General Purpose, Open Matlab Environment for Research and teaching Optimization*. Research Report in Mathematics. Technical report Ima-TOM-1997-3

[Jon-95] Jones D.R., Perttunen C.D., Pardalos M., Nguyen V. Thoai. *Introduction to Global Optimization.* Nonconvex Optimization and its applications. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[Kear-96] Kearfott R. Baker (1996) *Rigorous global search continuous problems by R. Baker Kearfott*. Dordrecht [etc.] Klumer Academic Publishers cop. 1996

[Map-03] http://www.maplesoft.com/

[Mat-03] http://www.mathworks.com/

[Mer-83] Meredith D.D.,(1983) *Use of Optimisation in Calibrating Water Distribution Models.* Proc. ASCE Spring Convention. Philadelfia.

[Min-00] Ministerio de Medio Ambiente *Libro Blanco del Agua* Secretaria General Técnica Ministerio de Medio Ambiente de España 2000.

[Moo-66] Moore R.E. *Interval Analysis,* Prentice Hall Inc. Eglewood Cliffs, N. J., 1966.

[Mor-93] More J. J. Wright S. J. *Optimization Software Guide* Society for Industrial and Applied Mathematics. Philadelphia 1993.

[Mur-86] Murtagh B. A. *Advanced Linear Programming: Computation and Practice.* John Wiley & Sons, Inc., New York, 1986.

[Orm-86a] Ormsbee L.E., Lingireddy S. *Calibrating hydraulic network models.* Journal of American Water Association. February 1997. 42-50.

[Orm-86b] Ormsbee L.E., Wood D.J. *Explicit pipe network calibration.* Journal of Water Resources Planning and Management. ASCE April 1986. 112(2) 166-182.

[Orm-89] Ormsbee L.E. (1989) *Implicit network calibration.* Journal of Water resources Planning and Management. ASCE March 1989. 115(2), 243-257.

[Oza-86] Ozawa T. The principal Partition of a Pair of Graphs and its Application. Discrete Applied Math., 1986.

[Pas-67] Passy U., Wilde D.J. Generalized polynomial optimizations, SIAM J. Appl. Math., 15 (1967) pp.1344-1356.

[Per-96] Pérez R. Quevedo J. Cembrano G. Argelaguet R. Wells G. DR5.4-1- Test and validation of the forecasting and optimization functions. European project WATERNET, Knowledge Capture for Advanced Supervision of Water Distribution Networks, ESPIRIT IV nº 22.186. June 1996.

[Per-01a] Pérez R. Quevedo J. *Parameter error detection in water distribution system models* III Simposio de control automático, Cuba 19-23 Març 2001

[Per-01b] Pérez R., Quevedo J. *Macrocalibration of Water Distribution System Models Using Classification* Water Software Systems: theory and applications, Research Studies Press 2001.

[Per-03] Pérez R., Quevedo J. Optimisation methods for the Calibration of Water Network Systems. Accepted for International Conference on Advances in

Water Supply Management. Imperial College September 2003 (accepted contribution)

[Pie-91] Piera N. Aguilar J., *Controlling Selectivity in Nonstandard Pattern Recognition Algorithms*, Trans. In Sys. Man and Cybernetics, vol 21, 1991, pp. 71-82

[Pie-89] Piera N., Descorches P., Aguilar J. *LAMDA: An Incremental Conceptual Clustering System.* Report 89420, LAAS/CNRS, 1989.

[Piya-72] Piyavskii S. A. *An algorithm for finding the absolute extremum of a function.* USSR Computational Mathematics and Mathematical Physics, 12:57-67, 1972.

[Pui-99] Puig V. *Aportación a la generación de Umbrales adaptativos.* Tesis doctoral Universida Politecnica de Catalunya. February 1999.

[Que-97] Quevedo J., Cembrano G., Argelaguet R., Pérez R. *Supervisión Avanzada de Redes de Distribución de Agua Potable* Escola Universitària d´Enginyers Tècnics Industrials de Terrassa-UPC. Terrassa - ESPAÑA (25.02.1997 - 26.02.1997).

[Que-99] Quevedo J., Cembrano G., Wells G., Pérez R., Argelaguet R. *Criteria for Applying Optimisation in Water Distribution Networks*. The International Conference on Computing and Control for the Water Industry-CCWI'99. Centre for Water Systems. Exeter - GRAN BRETANYA (13.09.1999 - 15.09.1999).

[Que-99] Quevedo J., Pérez R., Wells G., Cembrano G., Argelaguet R. *New Design of an Optimization Tool for Water Distribution Networks.* 14th Triennial World Congress. The Beijing International Conference Centre. Beijing 05.07.1999 - 09.07.1999

[Rah-80] Rahal C.M., Sterling M.J.H. Coulbeck B.(1980) *Parameter Tuning for Simulation Models of Water Distribution Networks.* Proc. Institution of Civil Engrs., London September 1980. 69, 751-762.

[Rub-02] Rubio Manuel C., *Potabilización.* Técnica Industrial Fundación Técnica Industrial Junio 2002.

[Sarr-02] Sarrate R. *Supervisió Intel·ligent de Processos Dinàmics Basada en Esdeveniments.* Thesis presented in Universitat Politècnica de Catalunya march 2002.

[Sha-74] Shamir U. *Optimal design and operations of water distribution systems.* Water Resources res. February 1974. 10(1), 27-36.

[Sko-96] Skogestad S. Postlethwaite I. *Multivariable Feedback Control, analysis and design.* John wiley and sons New York 1996.

[Sor-80] Sorenson H.W. *Parameter estimation, principles and problems.* Marcel Decker. New York 1980.

[Ula-00] Ulanicki, B., Bounds, P.L.M., Rance, J.P., Reynolds, L. *Open Loop and Closed Loop Pressure Control for Leakage Reduction*, (full version), Urban Water Journal, Elsevier Science, September 2000, pp. 105-114.

[UPC-03] http://webesaii.upc.es/

[Wais-98] Waissman J., Aguilar J., Dahlou B., Roux G., *Géneralisation du degré d'adequation marginale dans la méthode de classification LAMDA*, 6èmes Rencontres de la Société Francophone de Classification, 1998.

[Wais-00] Waissman J., Sarrate R., Escobet T., Aguilar J., Dahhou B. *Wastewater treatement process supervision by means of fuzzy automation model*. Proceedings of 15th IEEE International Symposium on Intelligent Control (ISIC 2000) GREECE July 2000.

[Wals-83] Walski T.M. (1983) *Technique for calibrating network models*. Journal of Water resources Planning and Management, ASCE, October 1983, 109(4), 360-372.

[Wals-84] Walski, T.M.(1984) *Analysis of water Distribution Systems.* Van Nostrand Reinold, New York

[Wals-95] Walski T.M. (1995) *Standards for Model Calibration.* Proc. AWWA Computer Conf. Norfolk Va.

[Wal-96] Walter E. Pronzato L. *On the identifiability and distinguishability of nonlinear parametric models.* Mathematic and Computer in Simulation 42 (1996) 125-134

[Wal-97] Walter E. Pronzato L. *Identification of parametric models.* Springer New York 1997.

[Wor-01] Worldwatch Institute *Estado del Mundo año 2001* Ed. Icaria, Madrid 2001

[WSS-98] Water Software Systems (WSS 1998). *A guide to water supply and distribution network model calibration.* August 1998 De Montfort University

[WSS-03] http://www.eng.dmu.ac.uk/wssys/index.html

[Zen-61] Zener C. *A mathematical aid in optimizing engineering design,* Proc. Nat. Acad. Sci. USA, 47 (1961), pp. 537-539.

[Zen-62] Zener C. *A further mathematical aid in optimizing engineering design, Ibid.,* 48 (1962), pp. 518-522.