Universidade de Aveiro Departamento de Engenharia Mecânica
2014

Daniel Fortuna Correia

Management and Control of Water Supply Systems

Universidade de Aveiro Departamento de Engenharia Mecânica
2014

Daniel Fortuna Correia

Management and Control of Water Supply Systems

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia Mecânica, realizada sob orientação científica de José Paulo Oliveira Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro e de António Gil D'Orey de Andrade Campos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri / The jury

Presidente / President	**Doutor Fernando José Neto da Silva** Professor Auxiliar da Universidade de Aveiro
Vogais / Committee	**Doutora Ana Maria Pinto de Moura** Professora Auxiliar da Universidade de Aveiro
	Doutor José Paulo Oliveira Santos Professor Auxiliar da Universidade de Aveiro (orientador)

Acknowledgements

I could not have written this thesis and accomplished this results without the help of many. Not wanting to minimize the contribution of anyone, some names needed to be highlighted due to their essential role in this process.
Professors José Paulo Santos e Gil Andrade, whose vast expertise in the field guided me through difficult situations, always directing me to the right direction.
To the entire project team, Bernardete Coelho, Bruno Pereira and José Vagos, by their help and attentive suggestions during the work development as well as the dissertation writing. I would also like to add a special acknowledgement to Eng. Carlos Teles by promoting and sponsoring this study, as well as the Águas do Douro e Paiva (ADDP) by their participation.
Finally, I wish to extend a special thank you to my parents, José António Correia and Albina Correia, my girlfriend Ana Maia and closest friends whose support and encouragement given along the course, and in particular during the dissertation, guided me throughout the toughest of challenges.

Palavras-chave

Sistemas de abastecimento de água; Métodos de previsão; Séries Temporais; Optimização; Simulação hidráulica; Sistemas SCADA

Resumo

O rápido aumento dos preços da electricidade tem provocado um aumento na preocupação com a tarefa extremamente dispendiosa de transporte de água. Através da criação de modelos hidráulicos de redes de Sistemas de Fornecimento de Água, e da previsão do seu comportamento, é possível tirar vantagem das diferentes tarifas horárias de consumo de energia, reduzindo desta forma os custos totais do bombeamento de água.

Esta tese foi desenvolvida em associação com o projecto de transferência de tecnologia denominado E-Pumping. Foca-se na procura de uma estratégia de supervisão e controlo flexível e adaptável a qualquer Sistema de Fornecimento de Água existente, bem como na previsão do consumo de água durante um período escolhido pelo utilizador final, o objectivo final é permitir o planeamento de um horário óptimo que minimize o custo do consumo de energia eléctrica.

O Protocolo OPC, associado a uma Base de Dados MySQL, foi usado para o desenvolvimento da ferramenta de supervisão e controlo flexível, constituindo no seu conjunto um módulo do Projecto E-Pumping. A escolha baseou-se em ambas as tecnologias serem adaptáveis a equipamentos de diferentes fabricantes. Esta tese produziu ainda uma ferramenta de previsão do consumo de água, adaptável a qualquer estação, constituindo um segundo módulo do projecto. Esta ferramenta foi obtida através do estudo e testes de *performance* a variados modelos baseados em séries temporais, especificamente aplicadas a este problema. Os parâmetros do modelo de base desta ferramenta são automaticamente actualizados a cada execução do programa. Ambos os módulos referidos foram integrados com uma Interface Gráfica (GUI) e implementados numa aplicação piloto instalada na rede de abastecimento de água da empresa Águas do Douro e Paiva (ADDP).

A implementação deste software em Sistemas de Abastecimento de Água por todo o país reduziria os seus custos de funcionamento, melhorando a sua capacidade de competição no mercado e, em última instância, diminuindo os preços da água para o consumidor final.

Keywords Water Supply Systems (WSSs); Forecast methods; Time series; Optimisation; Hydraulic simulation; SCADA systems

Abstract

The fast increase in the energy's price has brought a growing concern about the highly expensive task of transporting water. By creating an hydraulic model of the Water Supply System's (WSS) network and predicting its behaviour, it is possible to take advantage of the energy's tariffs, reducing the total cost on pumping activities.
This thesis was developed, in association with a technology transfer project called the E-Pumping. It focuses on finding a flexible supervision and control strategy, adaptable to any existent Water Supply System (WSS), as well as forecasting the water demand on a time period chosen by the end user, so that the pumping actions could be planned to an optimum schedule, that minimizes the total operational cost.
The OPC protocol, associated to a MySQL database were used to develop a flexible tool of supervision and control, due to their adaptability to function with equipments from various manufacturers, being another integrated modular part of the E-Pumping project.
Furthermore, in this thesis, through the study and performance tests of several statistical models based on time series, specifically applied to this problem, a forecasting tool adaptable to any station, and whose model parameters are automatically refreshed at runtime, was developed and added to the project as another module. Both the aforementioned modules were later integrated with an Graphical User Interface (GUI) and installed in a pilot application at the ADDP's network.
The implementation of this software on WSSs across the country will reduce the water supply companies' running costs, improving their market competition and, ultimately, lowering the water price to the end costumer.

Contents

List of Tables

List of Figures

Symbols and Acronyms

ACF	Auto Correlation Function
ADDP	Águas do Douro e Paiva
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
AR	Auto Regressive
ARIMA	Autoregressive Integrated Moving Average
DCOM	Distributed Component Object Model
diff	differentiation
EPA	United States Environment Protection Agency
EPAL	Empresa Portuguesa das Águas Livres
ETS	Exponential smoothing state space model
FTP	File Transfer Protocol
GIS	Geographic Information System
GPL	GNU General Public License
GSM	Global System for Mobile Communication
GUI	Graphical User Interface
GUI	Graphical User Interface
HMI	Human Machine Interface
ICS	Industrial Control System
IIT Kanpur	Indian Institute of Technology, Kanpur
MA	Moving Average
MAPE	Mean Absolute Percentage Error
MSE	Mean Square Error

ODBC	Open Database Connection
ODBC	Open Database Connectivity
OPC AE	OPC Alarms and Events
OPC DA	OPC Data Access
OPC HDA	OPC Historical Data Access
OPC UA	OPC Unified Architecture
OPC	OLE for Process Control
PACF	Partial Auto Correlation Function
PLC	Programmable Logic Controller
SCADA	Supervisory Control and Data Acquisition
SOA	Service-oriented Architecture
STL	Seasonal and Trend decomposition using Loess
TCP	Transmission Control Protocol
VSD	Variable Speed Drive
WSS	Water Supply System

Chapter 1

Introduction

1.1 Motivation

Water is one of the stems of life. The access to this element is instinctive to any living being. A series of economic and survival related activities have their roots on water and, historically, the growth of a population would be directly related to the availability of this resource. Therefore, rises a demand for a distribution system.

On the first millennia B.C, the technology *qanat*[1], developed by the Persians, allowed a distribution of water throughout the fields for agricultural irrigation purposes. However, the first public water distribution system registered appeared in Greece, in the mids V century B.C.[1]. Those systems have evolved through two and a half millennia reaching the systems we have nowadays, where failure of any kind is not tolerated.

This evolution occurred at the expense of large amounts of investment in technology and a growing amount of energy consumption. Although it provides a better and more stable service, either to a domestic or industrial client, this evolution led to a large annual total energy consumption by any mid size water plant. As publicized by Empresa Portuguesa das Águas Livres (EPAL) on the annual sustainability report [2], their energy costs represented 25% of the total of external services requested growing 9.3% against the values of 2009 and totalling 9.807.767 million Euro.

Lowering the energy cost spent on the transport of water for public supply is therefore a pressing matter. The networks currently installed, in particular, storage tanks for population supply or network pressure elevation tanks, generally use elevation stations to store and pressurize pipelines.

A special case is when the storage reservoirs are on a higher altitude than the original resource, requiring high power pumps, which will regularly consume energy to maintain the level of those reservoirs within the predefined limits. Once more, lowering the energy usage or, alternatively, reducing this energy cost becomes urgent. Knowing that pumps will handle energy on a large scale, even a slight earning will turn into a significant absolute value, along it's lifetime.

Nowadays this objective can be achieved following two different approaches:

- To upgrade the hardware, investing on more efficient pumps and piping, renewing the industrial layout, all which will require a significant investment, or even may be an impossible approach for the installed hardware may already be of high efficiency.

- Alternatively, to repair the hardware installed, which would solve the degradation resultant of years of usage, subsequently having a efficiency improvement up to 20% has shown by Reynolds [3].

A second strategy would be an intelligent management of the water deposit tank levels, associated with the energy resource pricing schedule. This second approach is more interesting, given that the necessary equipment is already installed. As the new concept proposed takes advantage of the energy cost variance, ultimately new hardware may not be needed. This strategy thus presents the possibility of having a very rewarding return of investment.

Energy companies, in order to reduce their operational costs, try to uniform the demand by their clients throughout the day. An example of the incentives offered are the lower tariffs during the night time, which are given in order for the clients to shift their demand to that time period. This results on lower energy consumption pike during the day, thus allowing a reduction of the installed available power, which has a significant cost by the simple fact of being operational.

These incentives to overnight energy consumption enable the possibility of applying a scheduling based strategy in water pumping systems, which is then taken to advantage by the water companies to reduce their energetic bill. It is relatively common for water stations to have installed Supervisory Control and Data Acquisition (SCADA) systems that enable this type of control. However, these systems are often not used to their full potential.

1.2 Objectives

The main objective of this thesis is to give support to the implementation of a new solution for optimization algorithms on Water Supply Systems (WSSs), in cooperation with the project E-Pumping [4]. This will include the creation of custom tools for water demand forecast and for supervision and control. To achieve this goal the first step was to employ statistical models, particularly time series, which based on the past water demand generate a short-term forecast, then used as a basis for water systems optimization. In a latter phase a SCADA solution of supervision and control will be designed and implemented.

The work presented in this thesis is part of the aforementioned project, financed by the company Telesensor, which specializes in water station instrumentation and efficiency improvement. The project consists in developing a new product for the company's catalogue, designed by a team of experts from the Division of Mechanical Engineering Optimization within the research group GRIDS, at the University of Aveiro.

Through this project, it was intended to promote a change in the way water reservoirs were controlled in most stations. Several of the systems installed were already capable of being commanded from a remote control center. However, the systems control did not take into account the energy cost variation over the day, filling the reservoirs only by having as indicator the water level of each reservoir. A less expensive practice would be to program these actions to a lower energy price schedule, if possible. Although some companies already apply this type of optimization using years of empirical evidence, the proposed solution guarantees that operating costs are kept to a minimum or optimal level.

1.3 Main Contributions

Four types of time series forecast were studied during the development of this thesis. Using data samples, each technique was compared to one of the others, resorting to common error indicators as Mean Absolute Percentage Error (MAPE) and the maximum deviation from the reference, thus selecting the most appropriated for the application. A forecast module was developed in C++ which implemented the classical multiplicative decomposition algorithm.

Several possibilities to employ a redundant communication scheme were studied. The final solution adopted was composed of an Ethernet module with a backup GSM communication system, applied to the data acquisition system. However, due to time constrains, only the Ethernet module was applied on the final project.

A data acquisition and control module was developed in C#, to connect to an OPC system, resorting to the OPC foundation library [5] for communication with several OPC servers. This module would be the end point of the project diagram presented in figure 2.1.

1.4 Outline of the thesis

In chapter 2, *State-of-the-art Review*, the state-of-the-art in the main areas that this thesis explores, industrial control and SCADA applications, Water Supply Systems (WSSs) optimization and statistical modelling were reviewed, also possible paths to follow in continuing these developments are explored.

The specific methods employed were further detailed in chapter 3, *Methods and Development*, where their theoretical components are exhaustively examined.

The implementation of these methods and consequent results were debated in chapter 4, *Results and Application.* This chapter also includes the presentation of the final solution.

Finally in chapter 5, *Final Remarks*, the review of the main achievements of both this thesis and project is made, and some suggestions for future developments and possible improvements are set.

Chapter 2

State-of-the-art Review

2.1 Introduction

The water supply industry has evolved towards the automation of processes, similarly to other industries. There is a tendency for every process to be monitored and remotely controlled. Relevant information is gathered on databases for further analysis, as are alarms and events set up. This as led to a more reliable and efficient water supply. To make evident the complexity of an WSS the figure 2.1 is present below.

Moving upwards in figure 2.1 starting from the water demand up to the optimization algorithms and hydraulic simulation: an array of sensors will measure vital parameters from the network, which are then read by the Programmable Logic Controllers (PLCs) and/or microcontrollers. This stage is highly important given that it is the basis of the entire network, as nearly every calculation will be based in this data. The storage of this data on a database, and its accessibility to the technicians via a SCADA system, has improved security and efficiency issues, as well as allowed researchers to develop several tools for WSS improvement.

The relevance of the WSS in our society has led this research field to be the target of several studies. Moreover, the improvement of these systems efficiency, by hydraulic simulation and optimization, is not a recent event. For instance, Walski [6] refers to the first pipe digital models, between the sixties and the seventies. As computational power increased over the last few decades, so has the complexity of the networks and models studied as stated by Martins et al., 2006 [7].

Factors as the significant estimated 30% water loss in the world, and consequent energy lost, have motivated the development of new efficiency improving tools and techniques. Between the most relevant developments figure the installation of new pump systems and configurations, or hydraulic simulation and pumping scheduling. All these improvements provide the cost optimization desired, but there is still room for improvement.

Designing optimization implies minimizing the total system cost. Without compromising the reliability and normal operation of the plant. SCADA systems have built the bridge between the two concepts, enabling real-time communication between parties. Further discussion on this theme will be presented below, on section 2.2, as part of the study for the development of a software that interacts with both data gathering and control operation.

Kiselychnyk et al 2009 [8] states that accurate estimation of water demand is a requisite for the success of the pump scheduling optimization, as these forecasts provide

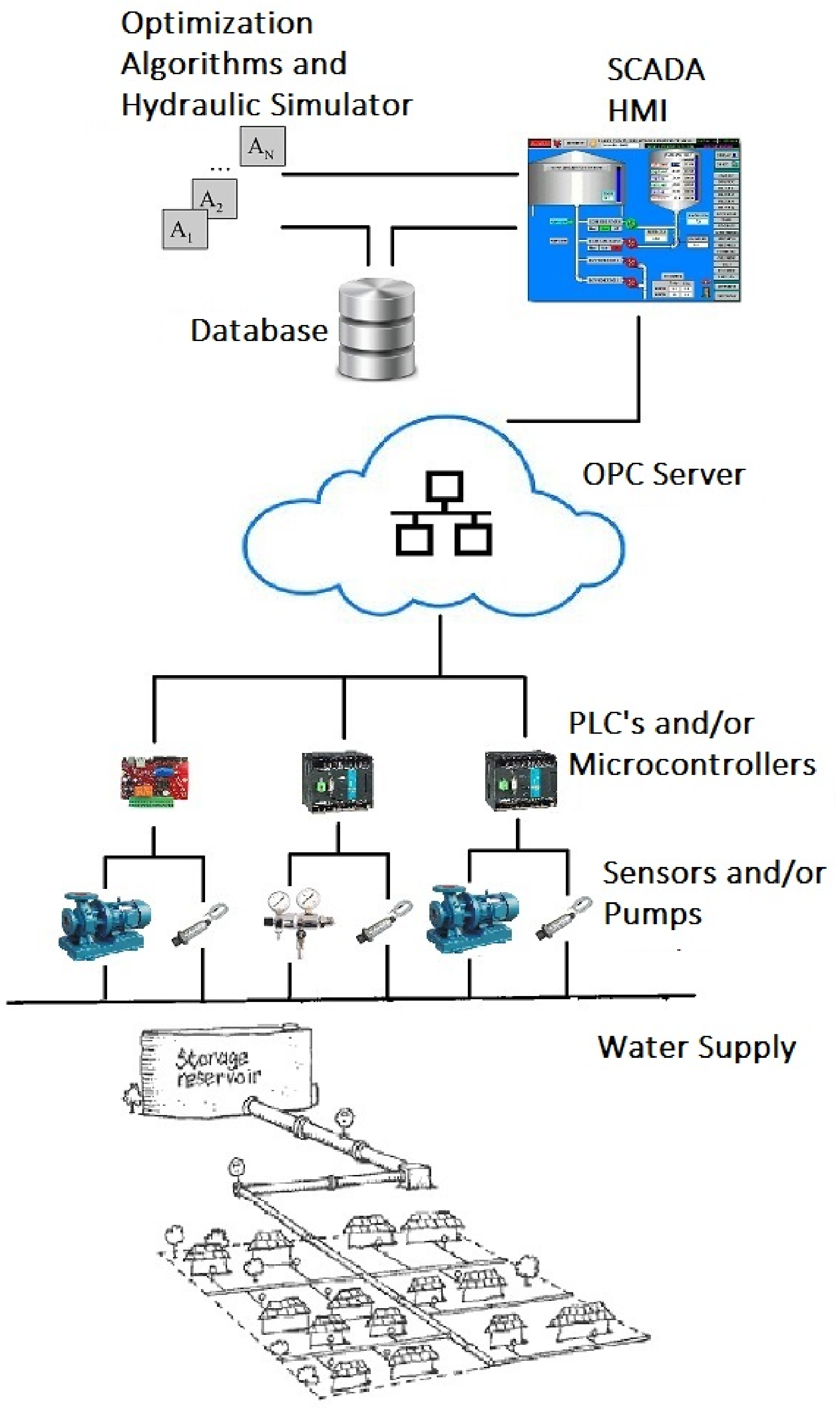


Figure 2.1: Water supply system network

a better starting ground for the approximations between water supply rate and water consumption rate. On this thesis several models of water demand forecast were studied, focusing on time series models that, despite of less precision than the state-of-the-art ANNs, have better performance when dealing with the lack of long data historic.

A review on the state-of-the-art of these components, instrumentation, SCADA systems, water demand forecast and WSS optimization, as well as current commercial solutions on the market, will be detailed on this chapter.

2.2 Industrial equipments and SCADA systems

Since their introduction in the industrial world on the late 1960's, PLCs have had a determinant role on the evolution of the quality, safety and cost control of several industrial process. The awareness for the need of supervision, at industrial level, led to the use of computers in any industry. Examples of such would be the oil industry, food, avionics, cars, or even public services such as the Water Supply Systems (WSSs).

The PLC is a kind of dedicated computer, able to handle real-time operations, that acquires both actual digital or analogue signals. It also has internal programs, timers, counters and data communication interfaces. The PLCs have been replacing relays, timers and other components hardwired by virtual ones (inside the PLC). Although inicially their main advantage was, and still is, the easiness of program alteration, the PLCs dramatically reduce the cost of ground control updating, as well as the amount of time needed to adjust processes when required. It also facilitates the troubleshooting over damaged hardware and outdated blueprints, as detailed by Segovia [9].

An array of sensors can connect to a PLC, directly and individually to its digital or analogical inputs, or through a bus communication over a pair of wires, such as I2C, SPI, or Modbus over RS485 or Ethernet. This can be considered the lower layer of communication. However, and as this thesis aims to connect directly to the control center, that usually has all the important information concentrated on a central server, the focus is on higher levels of communication.

In what concerns the central server, several options can be taken, from proprietary systems to open standards. The OPC standard allows an easy and quick integration of several proprietary equipments and software packages used in the central control. The OPC standard insures that every OPC manufacturer will be compatible and accessible through its applications, namely by OPC clients.

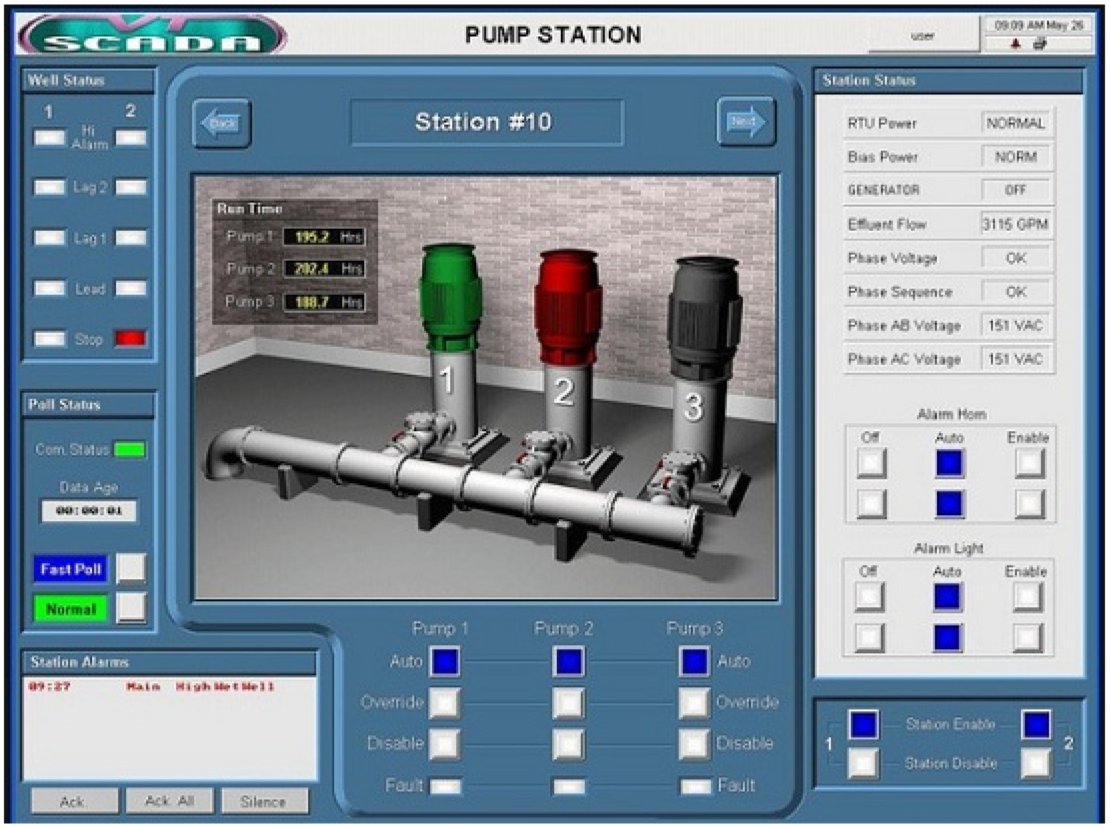
OPC provides a method for different software packages, namely Human Machine Interfaces (HMIs), to access information as well as control client devices. These devices are usually PLCs or other industrial appliances that control a process.

The OPC Data Access (OPC DA) is considered to be a classic OPC specification, based on Microsoft's Windows technology COM/DCOM (Distributed Component Object Model). It provides real-time data from processes to SCADA applications, widely used on industrial set-ups. Also as part of the classic OPC, the OPC Alarms and Events (OPC AE) and OPC Historical Data Access (OPC HDA) are often substituted by SCADA applications that handle events and an Open Database Connectivity (ODBC) that will feed a designed database.

Additionally, the OPC Foundation released a new standard, the OPC Unified Architecture (OPC UA). This standard is no longer depended on Microsoft's Distributed

Component Object Model (DCOM), instead two different protocols establish connections, an Service-oriented Architecture (SOA), used on web services over HTTP, and an TCP protocol, optimized for high performance. These allow for platform independent developments, such as low memory embedded devices as show by Fraunhofer-Application Center IOSB-INA [10], by scaling an OPC UA server down to 15 kB RAM and 10 kB ROM.

The end purpose of the mentioned server is to provide support for SCADA application for the plant's management. An example of one of these software packages available in the market can be seen in figure 2.2



Pump 1	195.2 Hrs
Pump 2	202.4 Hrs
Pump 3	188.7 Hrs

RTU Power	NORMAL
Bias Power	NORM
GENERATOR	OFF
Effluent Flow	3115 GPM
Phase Voltage	OK
Phase Sequence	OK
Phase AB Voltage	151 VAC
Phase AC Voltage	151 VAC

09:27 Main High Wet Well

Figure 2.2: SCADA demonstration [11]

2.3 Water demand forecast models

As presented above, accurate predictions of the water demand in WSS are of major importance. Walski et al. 2001 [6] defines tree basic water demand types: the costumer demand, the water losses of the system and fire emergencies demand that are stated as obligatory.

Aspects such as rainfall, air temperature, demography, water pricing and regulation influence the demand for water. Several models that account for these factors have been suggested. Zhou et al. 2000 [12], and Jain et al. 2001 [13] , who first introduced univariate and multivariate time series, have more recently developed models based on ANNs.

Regardless of the algorithm in question, discerning between short-term and long-term forecasting is the first step. This text will focus on short-term forecasting, as it is a more interesting approach for the development intended. However, long-term forecasting has a very important role when designing and dimensioning a new, or expanding an existing, water distribution network, estimating the size of reservoirs, pumping stations and pipe capacities. Both these activities require between 5 to 25 years prediction of the population growth, and consequent water demand.

On the other hand short-term forecasting is useful for efficient management and operation design and it shall be the target of further investigation throughout this thesis.

Forecast water demand on an urban environment is a difficult task, as urban water demand is a combination of domestic, public, commercial, industrial and even irrigation during certain months as stated Snoxell [14]. In addition, the variations associated with weather uncertainty, stated above, should be accounted for, inducing more complex computations.

Time series analysis focus on this variations and their evolution through time. Time series can reveal patterns of the variable in question, such as, random, trends, level shifts, cycles, unusual observations or a combination of patterns as Douglas explained [15]. Plots are often used to easily observe this phenomena, as on figure 2.3.

Apart from forecasting demand, time series plots can also be used to detect malfunctions, such as malfunctioning sensors, blown pipes or even communications problems, as it was likewise explained by Douglas [15].

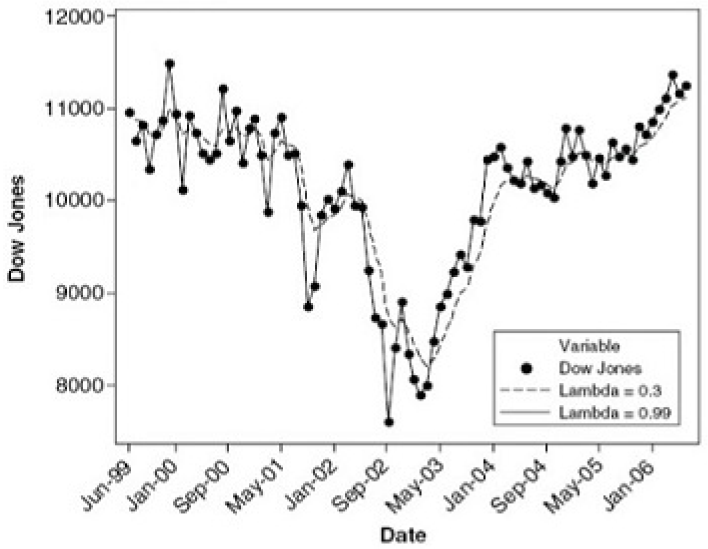


Figure 2.3: Example of time series by Douglas [15]

Artificial Neural Networks (ANNs), are the state-of-the-art development in the area of forecast and optimization. The water demand and climatological data gathered are put through a learning process as an input. Then, iterations of combinations between the data produce the output function that best fits these variables. Jonh Boudadis [16] applied time series analysis and ANNs to the forecast of the weekly water peak demand, comparing the two as to proximity to the actual observed results, and concluding that ANNs outperformed time series analysis. Much the same is concluded in [13] by Jain, when applying the two types of model to short-term water demand forecast in Indian Institute of Technology, Kanpur (IIT Kanpur). Despite having good overall performance, the use of ANNs requires the availability of a long term past data, which is not always the case. A general schema of a ANN configuration is presented on figure 2.4.

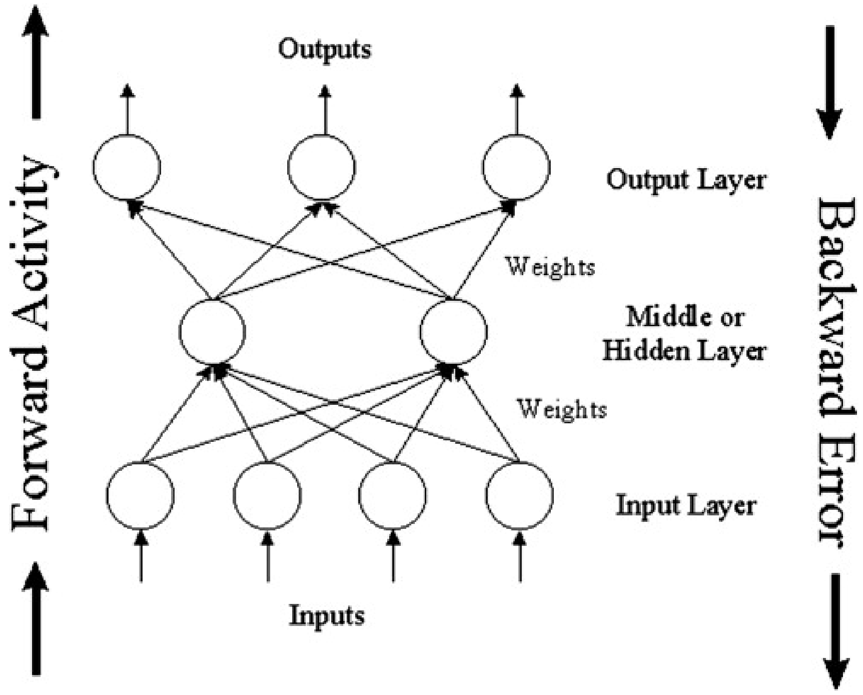


Figure 2.4: Artificial Neural Network (ANN) typical schema [17]

The same problem remains with multivariate time series, which often produce better results than univariate time series, but require an extensive database. Moreover, the excess of variables and their conditions leeway may lead to failure of these models, as suggested by Caiado [18] and Silva [19]. Quick climatic changes may also induce error.

Uncertainty about the data origin, or even the non existence of data in the field of resources forecast has lead researchers to focus on univariate time series. These are currently used for water demand forecast Caiado [18], energy demand forecast, Abdel [20], forecast of wind speed, Torres [21], or even solar irradiance, Dazhi [22].

Following this line of thought several univariate time series models come up. These conventional time series, such as, ARMA, ARIMA or even exponential smoothing, are still used and may outperform more recent solutions, in some cases.

2.4 Optimization in water supply systems

As the focus of this thesis is not to study optimization but rather the technologies that support it, only a brief introduction to the theme of optimization will be given.

To introduce the theme of optimization in water supply systems it's first necessary to present an important tool that enables a better understanding of the problem: hydraulic simulators.

Hydraulic simulators are numerical programs in which, given the network's information, it is possible to implement complex models of WSS. This technique enables the user to model a complete WSS and try out several optimization algorithms with ease. Although there is a wide range of hydraulic simulator software available, EPANET [23], developed by the United States Environment Protection Agency (EPA) as open source software, has had a substantial growth in popularity among researchers and companies in this field. An example of its application is presented in figure 2.5.

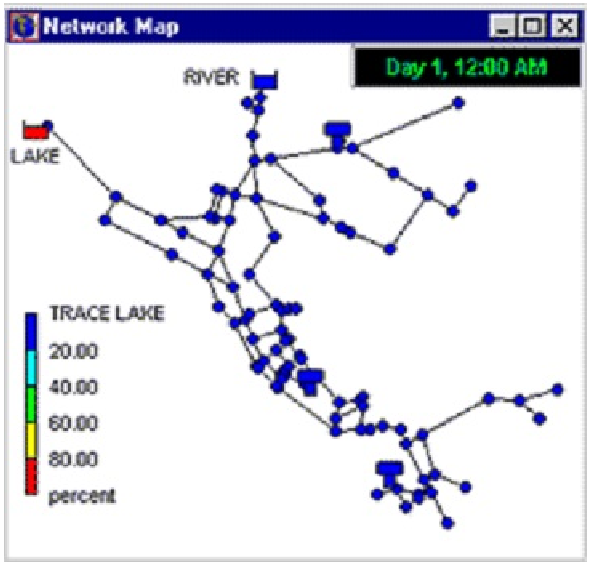


Figure 2.5: EPANET network model example [23]

Other software brands make use of EPANET as a basis for the development of different modules, as integrated SCADA and Geographic Information Systems (GISs), SCADAs systems enable data gathering and display, whereas GISs systems enable capturing, analysing and displaying geographically referenced information. An example of these software is AQUIS from 7-technologies Schneider Electric [24], that integrates both these features, along with several others that complete the integrated application.

Applications non depended of EPANET are also available, such as Aquadpt [25], which offers a complete range of features related to WSS, similar to those presented on AQUIS.

Despite not being the focus of this work, there are several types of optimization techniques that can take place on WSS. An example are piping lines improvements, such as new coatings and bottlenecks modifications that by reducing head losses improve overall efficiency, saving up to 20% of pumping energy as achieved by Gellings [26]. Replacement of inefficient equipment by high-efficiency systems also leads to improvements of up to 30%, also studied by Gellings [26]. Likewise, the use of Direct Torque Control by Variable Speed Drive (VSD) enables flow control via the variation of the pump speed, which in certain conditions may achieve 10 to 20% savings on pumping energy as stated by Kiselychnyk [8].

Some of the mentioned options might not suit specific applications, derived from the demand for high investment, which sometimes is not economically viable. Considering this, one option rises: the scheduling of operations. This option concerns specifically pumping operations and, instead of reducing energy consumption, their focus is on the energy cost. This type of optimization makes use of the hydraulic simulator in order to obtain the system equations and then employ optimization tools.

There are several optimization methods. The conventional one, *trial and error*, can be effective for simple systems, but on a larger scale, the complexity of the problems may pose significant obstacles, such as: multiple pumps, valves, head losses, etc. Consequently, it evolved to more effective algorithms.

Coelho, 2011 [27] applied two algorithms of energy resources optimization in WSS, the Levenberg-Marquardt (LM), a classic gradient-based method, and an evolutionary algorithm (EA), based on heuristic search. These methods tested on standardized networks, achieved reductions of 71% in costs associated with pumping.

To have a better understanding of pump scheduling optimization, one must first think on a smaller scale. In a best case scenario, all the hours of use of a pump would be shifted towards the less expensive energy schedule only operating at the lowest cost possible. However, the complexity of most WSS mean that these are non-linear problems that lead to more complex resolutions. For instance Reynolds [3] states that it is virtually impossible to explicitly solve the scheduling problem using common mathematical techniques.

The objective of an optimization tool is to find, with the given information, the minimum or the maximum working point of the cost function, as explained by Atkinson [28]. For the case being, it is expected to find the optimal schedule, *i.e.* the less expensive schedule, for the system to perform fully operational as a whole (the minimum of the cost function).

Reynolds [3] presented the results of the tool Aquadapt [25], a complete software that features a full comprehensive analysis of the network, including a SCADA client, a 24h forecast tool and optimization tool. As a result of the application of this software on four cities along the U.S., an average efficiency gain from 6.0% to 8.6% was achieved. In 1998, in the Spanish city of Seville, Carlos León [29] developed a similar system with the purpose of satisfying water demand and reducing energy related cost on pumping operations, whilst maintaining the quality of the treated water. The end result should resemble figure 2.6, in which is plotted the optimal pumping schedule in relation to the given electricity price.

The solution proposed on chapter 3 and which is one of this thesis objectives, follows the same philosophy.

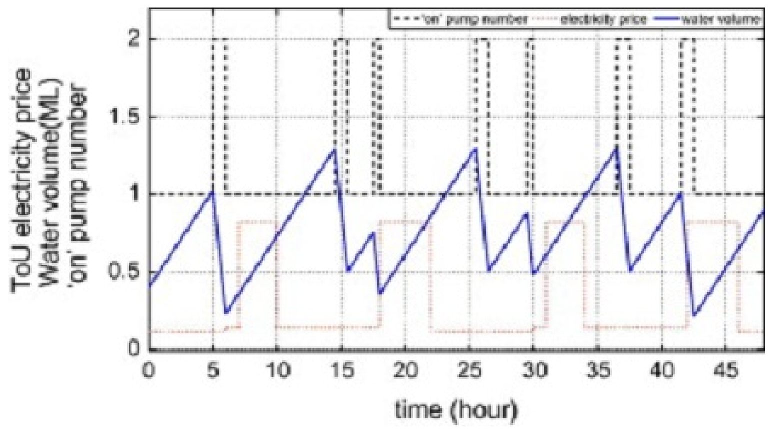


Figure 2.6: Pumping station scheduling [30]

2.5 Current commercial solutions

Water Supply Systems (WSSs) optimization is not a recent event. Several commercial applications have grown in the market in the last the decades, from multinational corporations, such as Schneider Electric, Bentley, to smaller companies, such as Decerto Aquadapt, KYPIPE or tynemarch systems.

Schneider Electric presents the AQUIS software [31], a complete plant modelling and management system. By modelling the network, having direct access to it's information and being fully integrated with the plant's SCADA system, Schneider insures a 1% accuracy model, thus capable of sensible leakage detection. A flexible water demand forecasting tool, integrated with the weather forecast, provides the basis for the optimization module that optimizes both pumps and reservoirs according to demand, energy's price and storage capacities.

Bentley, among other solutions for water, wastewater and stormwater systems, offers two software packages for pipeline modelling and optimization, the WaterCAD [32] and the WaterGEMS [33]. WaterGEMS uses a genetic algorithm in the Darwin Scheduler Module [34] to minimize the energy cost based on the energy's tariffs. Other relevant modules are available, such as the Pipe Renewal Planner Module [35], a decision support tool on the rehabilitation of water mains that determines which pipelines are in need of maintenance, or the Darwin Calibrator Module [36], a pipeline modelling tool based on genetic algorithms to create automatically an accurate model of the system, enabling system stress simulations and leakage detection. The SCADAConnect [37] module is an add-on to integrate the WaterGEMS software with the current SCADA solution installed, enabling the development of real-time response strategies.

On the other hand, Aquadapt commercialized by Decerto [25] offers system's cost reduction by moving the energy consumption to cheaper tariff bands, reducing peak energy demand, selecting the shortest water route and improving pump efficiency. To achieve these goals, Aquadapt retrieves live data from the SCADA system, filters and processes outliers, and then forecasting the plant's operations for the next 48 hours, which is updated in real-time if significant changes in the system occur.

Chapter 3

Methods and Development

3.1 Introduction

One of the main objectives of this thesis is to give support to the implementation of a new solution for optimization algorithms. The first step, in order to achieve this goal was to forecast water demand employing time series.

In a second phase, a SCADA solution, that suits the particular requirement of this type of application, mainly supervision and control applied to work with the optimization software, is designed and implemented.

For better visualization of the interaction between all the modules involved in the project, a flow chart is presented on figure 3.1. In this scheme the squares represent modules, as the arrows are a synonym of interactions or communications.

The complete project with work by running several modules in sync. The water demand forecast module, the first module, will send information into the EPANET simulation so that it working together with the optimization module can output and optimum operations schedule. Moreover, the supervision and control module will execute the operations it receives, as well as record the station's relevant data, and feed it to the project's database.

As stated by Z.Chen [38], the water consumption by a population follows a trend. Through the study of the water demand recorded through a long period of time by a WSS, it is possible to deduce certain tendencies. On this solution, the relevant data is selected and saved into a database specifically designed for the project, using the software MySQL [39].

In order to analyse the data from water demand, saved in the database, time series analysis was the tool employed. Although the state-of-the-art suggests that more modern technologies, such as ANN achieve superior results, when dealing with a short period of past data ANN revealed themselves as not appropriated for the task in consideration. As presented in the previous chapter, studies as of Jonh Boudadis [16] or even Jain [13] have shown that time series often obtain better results than ANN in these specific conditions, such as the lack of solid information.

Time series are algorithms which, in this particular case, project a short-term forecast, based on the past water demand, which is then used as a basis for the water system optimization. But they can also be used for long term forecasting.

Four types of time series were employed on a development environment, varying their parameters, and totalling eight tests, with the intention to study their performance as a

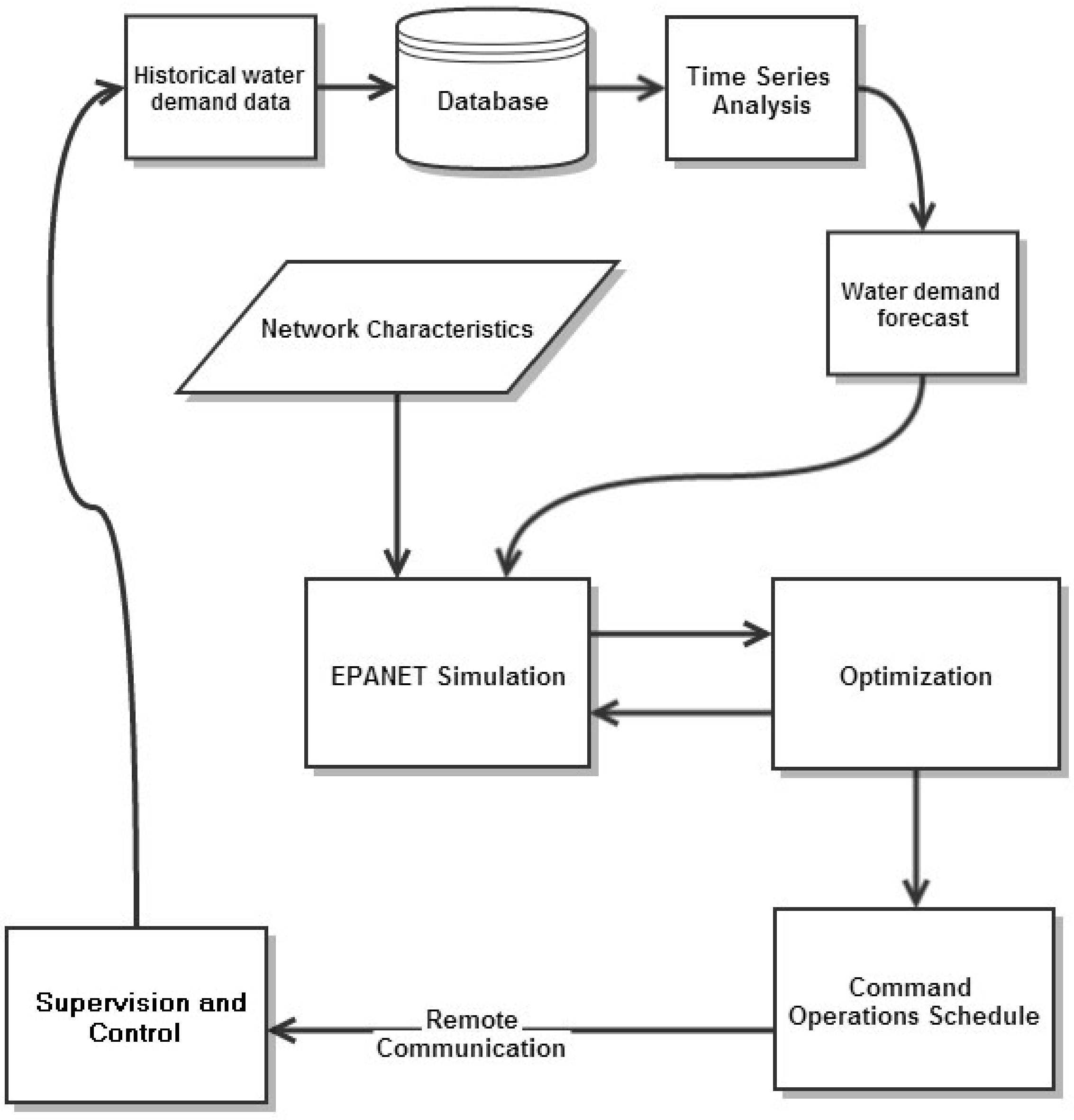


Figure 3.1: Schematic display of all the processes involved in the proposed solution.

forecasting tool. These types were five variants of exponential smoothing methods, of those three additive seasonal models and two multiplicative seasonal models, as well as a particular variation of an ARIMA model with a seasonal component, and two classical decomposition techniques with forecasting.

Having obtained a water demand forecast from the process of time series forecasting, the input of the hydraulic simulator is then established, meaning the accepted forecast will be the base for the hydraulic simulator to define the necessities of each station.

The following step is to insert the water demand forecast, as well as the system's characterization into the EPANET hydraulic simulator [40], so that the optimization algorithm chosen and the simulator can work in synchronism. Doing so will test each possible scenario giving a global view of the system behaviour as a whole. Then, iterations of combinations between the two programs, the hydraulic simulator and the optimization algorithm, will produce the output function that best fits these variables. The process is designed in such way that, after several iterations, the optimal solution is achieved.

The result of this simulation is a complete comprehensive analysis of the whole system, outputting an optimal working schedule of control for each element. This schedule is then interpreted and organized into a list of actions readable by the station's operator and fed into a designed database solution for easier interpretation, either by a machine or, ultimately, by the end user, usually resorting to plotting for this last case.

On another subject, the remote communication mentioned in the schematic from figure 3.1 is done using internet TCP/IP protocol implemented on site. This protocol uses a designed program, which establishes connection both with the SCADA system database and the OPC server, which gathers the information of the system and, if intended, has the option of automatically control the water system based on the decoded schedule - optimal solution found.

However, this project is still on an early stage. An intermediate solution was developed to allow the user to decide in using or not the optimal schedule suggested by the program, rather than implementing it automatically. The interface in this step will merely suggest the user to analyse and manually control the system based on the given information.

It is worth remarking that all the system supervision information is constantly being fed back to the database, so that the historic of the water demand increases continuously. This action will have the benefit of improving the forecast performance by enabling the forecast tool to have a better understanding of the behaviour of the WSS.

On the next sections each module will be described in detail.

3.2 Supervision and control

The objective in this section is to find a solution to the control and supervision challenges of a Water Supply System (WSS). It is proposed to achieve a flexible, yet reliable solution, with the ability to connect to different types of installations and different brands of equipments, having in mind the automation of the whole project, from the data gathering and processing to the optimization and finally control.

To reach this goal two types of Industrial Control System (ICS) intercommunications were studied, both open protocols, OPC [5] and SOA [41]. However, considering that OPC is the most widely used protocol in this type of application [42], it would be wise to first explore the characteristics and assumptions of this protocol.

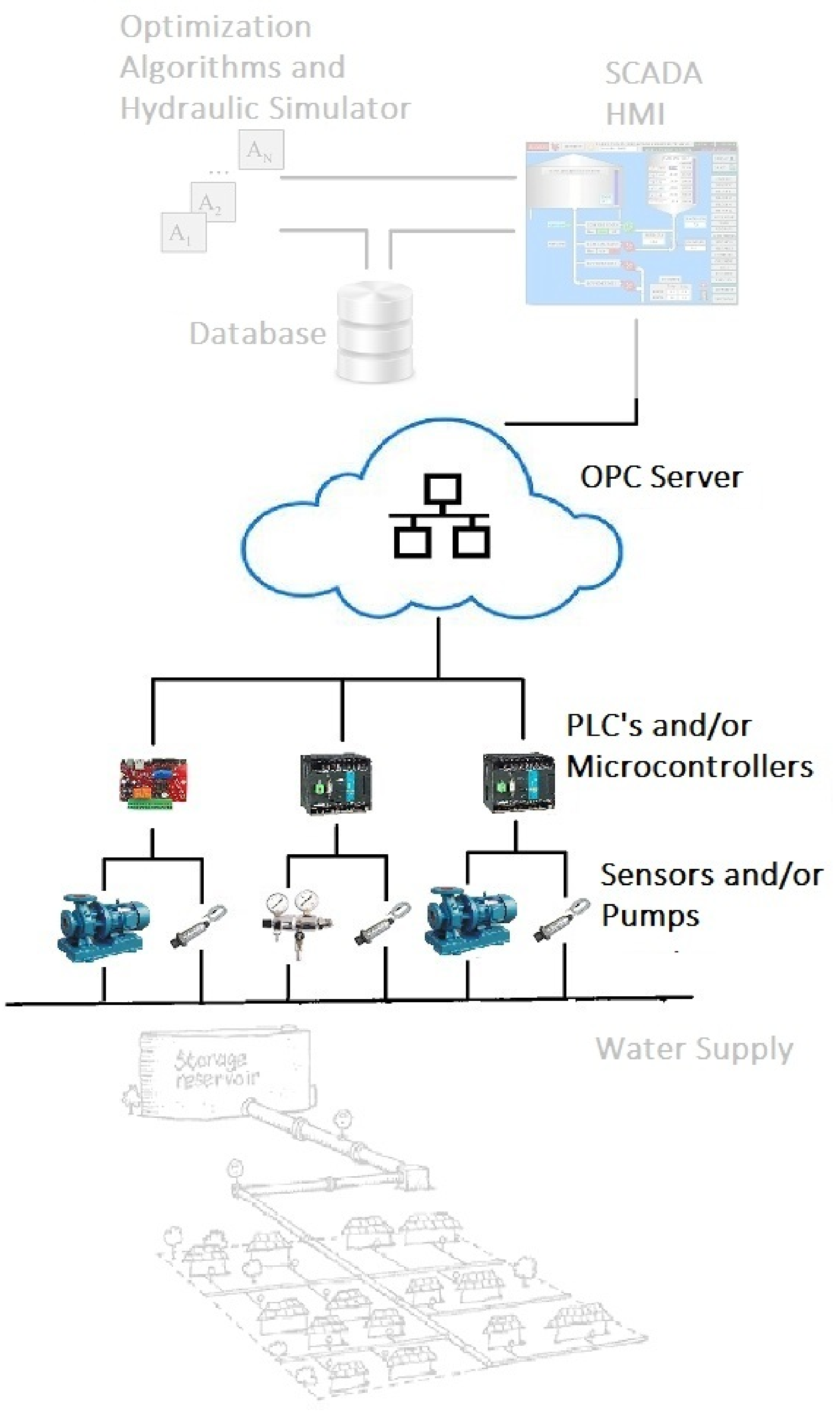


Figure 3.2: Water supply system network - detail

OPC is based on the principle of open architecture, meaning that it should be able to interoperate with several different equipments from various brands. This goal is achieved by adding an abstraction layer between the data source, in general a PLC, and the data receiver, for example a HMI. Popular within most PLC manufacturers, this protocol is constantly evolving to embrace more features. Derived from it's flexibility to work with different brands of manufactures and models, its implementation would render the program to be universal, facilitating its use on several different plants and, thus, working on virtually any recent installation available.

As an object linking protocol supported by the OPC foundation, the idea behind the classic OPC DA is to have a server that will receive all of the PLCs' informations, such as sensors information and pump or valves states. These entries are then identified as tags. Having accomplished this, the server will make the information available to all sorts of HMI, such as SCADA programs like Movicon [43], or accessible by the OPC Fundation library [5]. This type of linkage is crucial on a project that must adapt to several different plants, being able to gather data, make decisions and then act upon them.

The Águas do Douro e Paiva (ADDP) company, has agreed to be a pilot installation for the development of this project. This allowed a more focused development and testing, ideal for the project, due to the fact that ADDP's industrial equipment is compatible with the standard characteristics required. By looking at figure 3.2, one can visualize and understand the following equipment in perspective. The sensors communicate with the PLCs via physical layer of RS 485 standard, as pumps/valves are actuated by either discrete operation (on/off), *e.g.* relays, or by varying the pump's speed, *e.g.* Variable Speed Drive (VSD).

The communication between the peripheral stations and the central station, is established by a Transmission Control Protocol (TCP) over a Fiber-optic physical layer. This means that the information will be available at the central station with very little lag and with security.

Each PLC is connected to the central computer, which is using a Rockwell OPC server [44]. In turn and cyclically, the server downloads each tag, *i.e.* each equipment's information, from each device and presents it to the user for selection. Historical data is, in this case, built by a OPC DA client for Open Database Connectivity (ODBC), that establishes the connection between the OPC server and the database. This way, the information is secured on an external database for further consult and analysis. A laboratory scenery was set to simulate this layout, further described on chapter 4.1.

As this is a project that will automate some parts of the plant's operation floor, a certain trust level must be obtained with the plant's managers before implementing a complete autonomous system. This meant finding a way to build an offline version of the project, that would consecutively present the user with the optimal schedule that he should apply to the pumping stations, without actually imposing those choices.

The ADDP agreed on sending a *.DBF file, *i.e.* a database file, to an FTP location every hour with relevant tags of their OPC server, incrementing a data daily file and starting a new one at 00:00 hours. A C# program, based on the System OleDb library [45] and the System Odbc library [46], that read the file, filtered and processed the data and, then, inserted it into a local database that would be the data source for the rest of the project to run, was written specifically for the project.

The time series analysis method that would be applied to this data, required it to have an exact and consistent time interval between data and not having outliers that will

Table 3.1: Example of the linear interpolation used

Original Date	Original Value	Approx. Date	Approx. Value
2013-07-31 00:00:57	901,040.56	2013-07-31 00:00:00	901,040.38
2013-07-31 00:10:57	901,042.63	2013-07-31 00:10:00	901,042.43
2013-07-31 00:20:57	901,044.44	2013-07-31 00:20:00	901,044.27
2013-07-31 00:30:57	901,046.13	2013-07-31 00:30:00	901,045.97
2013-07-31 00:40:57	901,047.38	2013-07-31 00:40:00	901,047.26
2013-07-31 00:50:57	901,048.81	2013-07-31 00:50:00	901,048.67
2013-07-31 01:00:57	901,050.06	2013-07-31 01:00:00	901,049.94

throw the forecast off, as explained in the next chapter 3.3.

The timing of the data values was, however, not as accurate as it was expected, neither were the values blindly reliable. Some outliers occurred across the data, which needed to be removed, as well as the non existence of an exact time interval required some pre processing. It also might happen that there were missing values. In such cases, the algorithm would detect any missing dates and insert them using linear interpolation.

To solve this problem, the C# tool developed runs a threshold based filter that will eliminate any item considered not fitted given the previous and following value. This threshold was found based on the empirical knowledge that the normal variation of the water consumption should never be above 10000m^3/h. The data is then rounded to the nearest hour, or other chosen interval, and a linear interpolation between the two closest dates was implemented to acquire the discrete interval's approximated value, which in this case, had a 10 minute frequency. To exemplify the results for this program, the table 3.1 is presented.

To better explain the logic behind the solution, it was inserted the pseudocode 1. The linkage to the *.DBF file was established resorting to the System Data OleDb library available at Microsoft [45]. The data is then inserted and queried on the MySQL database using the System Data Odbc library [46], closing this way the communication loop.

The program based on the pseudo code 1, is heavily dependent on the connection to the database, hence it's low performance. Using a computer equipped with a intel core i5 processor, 4GB of RAM and a local network connection of 54 Mpbs, it took on average 15 minutes to filter and interpolate a daily file, composed of 144 entries (spaced 10 minutes between them) from each of 43 consumption points.

Algorithm 1 Data filtering tool pseudocode

```
1:  procedure Load and save raw data to database
2:      listlen ← length of data list
3:      for listlen do
4:          i ← present index of the for loop
5:          if (tag[i+1] > tag[i] + threshold) or (tag[i+1] > tag[i] - threshold) then
6:              Ignore tag
7:          else
8:              Save tag to database
9:              Check by querying database if entrie already exists
10:             if check database is true then
11:                 Use update query
12:                 Catch exception
13:             else
14:                 Use insert query
15:                 Catch exception
16: procedure Linear Interpolation
17:     for number of entries per day do
18:         Jump to next dicrete time interval
19:         Select from raw data the following discrete interval's values
20:         Select from raw data the previous discrete interval's values
21:         Calculate a linear interpolation using these variables
22:         Save calculated value to database
23:         Check by querying database if entrie already exists
24:         if check database is true then
25:             Use update query
26:             Catch exception
27:         else
28:             Use insert query
29:             Catch exception
```

3.3 Water demand forecast

Four types of time series analysis are tested in this section, as to its suitability to this particular application. By changing parameters such as the seasonal frequency, the nature of the model additive or multiplicative or even the parameter selection optimization algorithm, some variants were generated. This led to a total of eight tests, two which were based on the Holt-Winters exponential smoothing, one being an additive seasonal model and other a multiplicative seasonal model, other three were based on a similar time series analysis method to the Holt-Winters exponential smoothing, developed by Rob J. Hyndman and entitled Exponential smoothing state space model (ETS) [47] in which the author changes the parameter selection algorithm for one with higher performance. The sixth test consisted on an Autoregressive Integrated Moving Average (ARIMA) model with seasonal component and the last two were the classical decomposition techniques, additive and multiplicative decomposition.

The last two methods, additive and multiplicative classical decomposition methods, were studied in order to obtain an understanding of the workings of time series. These methods are popular among forecasting practitioners, by virtue of their accessibility to non experts in this field of studies, and are often used as teaching examples. Its ability to present seasonal and trend components information allows the user to integrate judgemental knowledge into the model, which makes it particularly useful in practice, as explained by H.C.Harrison [48]. The NCSS software manual [49], claims that, for seasonal data, these methods are often as accurate as more complex algorithms, such as the referred ARIMA or exponential smoothing, among others.

Classical decomposition is a straight forward procedure in itself. It can be, however, classified into two methods, additive decomposition and multiplicative decomposition. The first is more suitable to series in which the seasonal component amplitude remains constant, while the second is more appropriate to series with a changing seasonal component, as illustrated in the figure 3.3 below.

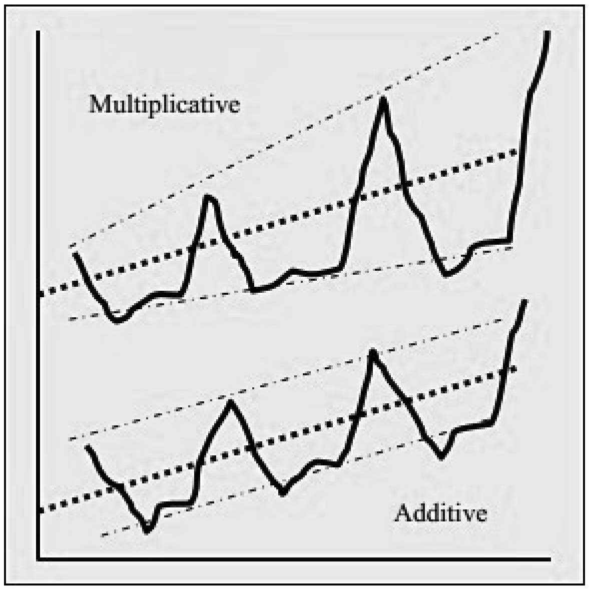


Figure 3.3: Seasonal amplitude in classical additive and multiplicative decomposition methods [50]

To begin decomposing a time series Y_t, it is first needed to identify its components accordingly to the method used. In classical decomposition, the series is divided into three different components, trend, seasonal and irregular.

This trend can be obtained by using smoothing techniques, by applying a moving average forecast with respect to the number n of seasonal periods with m, given as

$$\hat{T}_t = \frac{1}{nm} \sum_{k=1}^{nm} y_{t-k}, k = [m, 2m, ..., nm]. \tag{3.1}$$

On the particular case of the multiplicative method the mean should be removed by dividing each value by the series mean U, so that a new series is created having the reference as 1, normalized series. The series mean is mathematically written as

$$Y'_t = Y_t/U. \tag{3.2}$$

The seasonal component $\hat{S}_t$ is extracted from the raw data through a process of seasonal adjustment. Essentially the fluctuation provoked by the seasonal effect is removed, as explained in detail below. This will remove the trend component, $\hat{T}_t$, from the series, revealing the seasonal component. This is applied in the additive method by equation

$$K_t = Y_t - \hat{T}_t \tag{3.3}$$

and in the multiplicative method as equation

$$K_t = Y'_t/\hat{T}_t. \tag{3.4}$$

Stringing together all the seasonal indexes, K_t, respectively to its position on each seasonal cycle, that is, the sum of each K_t position according to it's seasonality frequency m, for n seasons, one can obtain the seasonal component, $\hat{S}_t$. In example, if the series has a weekly seasonal pattern, this would perform the mean of all mondays, tuesdays, etc, as done in the following equation:

$$\hat{S}_t = \frac{1}{n} \sum_{i=1}^{n} K_{t-(im)}. \tag{3.5}$$

Additionally, when using the multiplicative method, $\hat{S}_t$ should be normalized by dividing each component of $\hat{S}_t$ by it's mean, so that reference is centered at 1. This step is not required in the additive method, because when reconstructing the series the operator sum is used, instead of multiplying.

The remainder, otherwise called, irregular component is isolated by subtracting from the series Y_t the estimated trend $\hat{T}_t$ and seasonal component $\hat{S}_t$ adding up to equation

$$\hat{E}_t = X_t - \hat{T}_t - \hat{S}_t, \tag{3.6}$$

on the case of the additive model. Otherwise, in the multiplicative model, the components should be divided as equation

$$\hat{E}_t = Y_t/(\hat{T}_t\hat{S}_t). \tag{3.7}$$

This process only serves as an analysis of the data. In order to produce a forecast, the linear regression of the trend is extended and the seasonal effect is fed back into the series, so that the series may be reconstructed with all their components forecasted.

Although not recommended, classical decomposition is still widely used. Rob J. Hyndman presents several problems with these methods, such as the assumption that the seasonal component retains its periodicity throughout the series and that it is not robust to occasional unusual values, meaning that a filter may be applied before entering the data. These methods are also not tolerant to missing values, justifying the data filtering tool which was developed in this thesis specifically for this method, in addition to the time series forecasting tool, as it has described in chapter 3.2.

Despite the fact that the method of manually analysing the time series is an important learning tool, in order to comply with the project timetable the process of selection of the most appropriated time series analysis method, the statistics analysis program R, available on the official R project site [51], was used.

R is an open source language and environment for statistical computing. It provides a wide variety of statistical techniques, such as both linear and non linear modelling, time series analysis and clustering, among others. Several authors have contributed on the development of this software, with the addition of new packages and libraries, which improve the effectiveness of the application. On this subject, the author Rob J. Hyndman has contributed significantly to the time series analysis topic, with the addition of the time series packages, also available at [52].

For the being case of study, water demand forecast, the interest lies on the time series packages, as they enable a more effective and quick analysis of several algorithms, adding crucial tools such as seasonal decomposition and series plotting.

The decomposition of a time series is a tool devised to deconstruct the series into conceptional components, for the most part trend component, cyclical component, seasonal component and irregular component. This mechanism is particularly relevant on the case of exponential smoothing, as it will show the more adequate route to take, in searching which technique in this branch will provide a forecast with lower error.

STL decomposition is an acronym for "Seasonal and Trend decomposition using Loess" developed by Cleveland et al. 1990 [53] and added to the R library package stl [54] by Rob J. Hyndman. Loess decomposition is, to a great extent, a more advanced version of the Local Regression Smoothing, which is a fitting of a regression line recurrently over a time t of a time series Y_t. Loess computes the local regression smoother, with a local regression on which adjusted weights are assigned. Then iterations are repeated until the estimates converge, essentially relying on the fact that the series is time dependent. The weights are optimized in order to reduce the Mean Square Error (MSE). This can be executed on the R software by inserting the following lines of code 3.1, which will produce a plot of the four components of the time series referred above, as demonstrated on figure 3.4.

Having in mind that the water demand has a highly seasonal pattern, as is another example the electricity demand, it follows that this pattern can not be effectively modelled using standard polynomial models, such as presented on equation

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{3.8}$$

by Douglas [15]. A seasonal adjustment is added to the linear trend model, based on the Holt-Winters method, introduced by Holt[1957] and Winters[1960].

Seasonal adjustment is a method similar to time series decomposition. In this case the objective is to remove seasonal effects from the time series, by which, it will enable a better understanding of the underlining non-seasonal-features. By doing so, effects

Listing 3.1: STL decomposisiton in R

```
#read file
tmp<-read.table("data_8weeks.txt",header = F,skip=0)

#Trasnsform to TimeSeries
tmpTS<-ts((tmp),frequency=24*7)  #weekly seasonallity
 fiting <- stl(tmpTS[,], s.window="period") #Loess decomposition
 plot(fitting)
```

from basic weather progression throughout the year, as well as fixed holidays effects are removed.

Several techniques ramify from the initial exponential smoothing, such as exponential moving average, double exponential smoothing or even triple exponential smoothing, all similarly assigning decreasing weights to the irregular trend or, the more commonly known, random or noise effect.

However, these algorithms do not handle seasonal data, meaning that working with a time series that has a clear seasonal component, a particular variation should be used, entitled additive and multiplicative exponential smoothing data, so that the seasonal component of the series is also identified.

The two methods, additive and multiplicative exponential smoothing, added to the Holt's linear trend model given by the equation 3.9a, where ℓ_t is the level component 3.9b, and b_t the trend 3.9c. By adding or multiplying a seasonal component given by 3.9d, respectively to the additive or to the multiplicative model, will provide the capacity to adapt to the verified conditions and capture seasonality. This is done resorting to the difference between what is the general trend of the series and its seasonal fluctuation.

To provide a better understanding of the two models, additive and multiplicative exponential smoothing, a description of the mathematics of this models will be given below.

$$\hat{y}_t = \ell_t + b_t \tag{3.9a}$$

$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{3.9b}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{3.9c}$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \tag{3.9d}$$

The two methods differentiate themselves from each other, either by multiplying their components for the multiplicative method as presented in equation

$$y_t = (\ell_t + b_t) * s_t, \tag{3.10}$$

or by adding the components to the additive model as explained below

$$y_t = \ell_t + b_t + s_t. \tag{3.11}$$

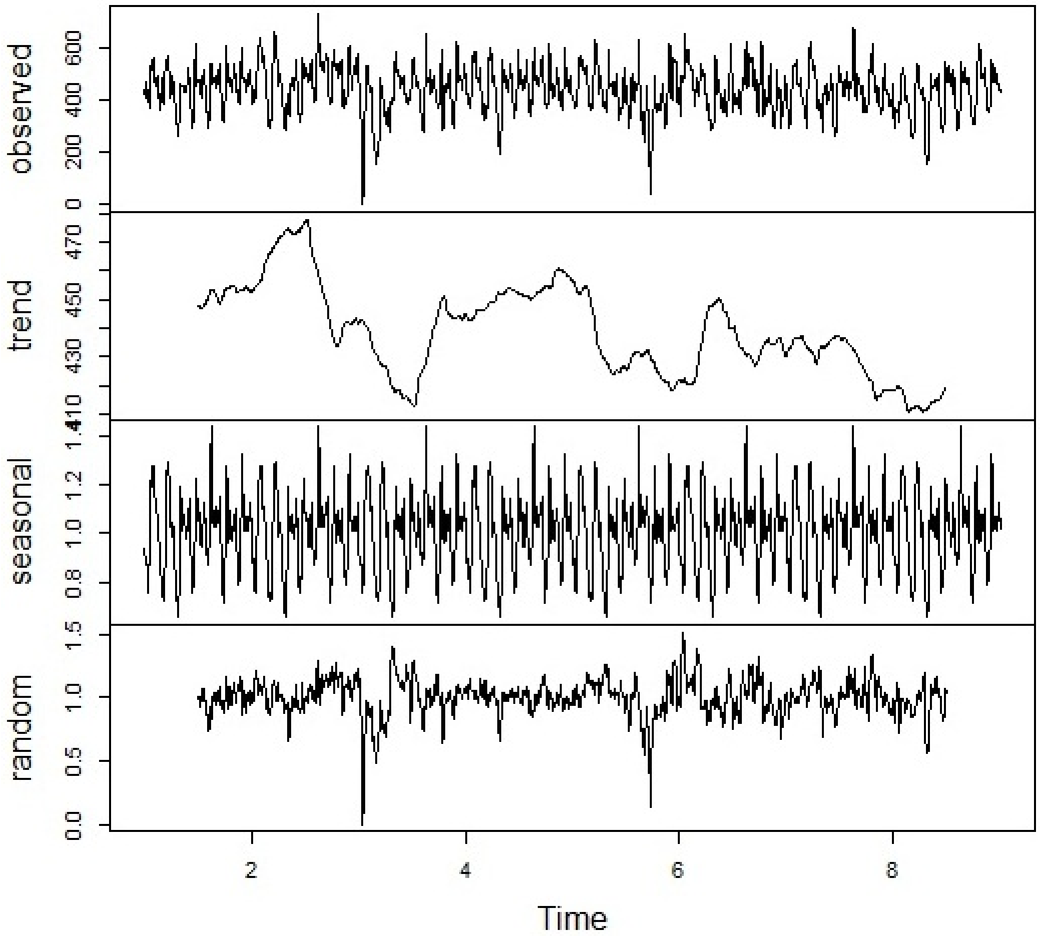


Figure 3.4: Typical Water Supply System (WSS) water demand time series decomposition

In order to make up a forecast from the seasonal additive exponential smoothing method, three equations are applied, one for the level ℓ_t in equation

$$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}), \tag{3.12}$$

one for the trend b_t given by equation

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{3.13}$$

and s_t given by equation

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \tag{3.14}$$

for the seasonal component. Three smoothing parameters are given α, β and γ. The m value expresses the period of the seasonality and h step ahead forecast. All of these components will add up following the equation of the series given by

$$y_{t+h|t} = \ell_t + hb_t + s_{t-m+h_m^+}. \tag{3.15}$$

Listing 3.2: Additive seasonal exponential smoothing forecast in R

```
fcastets4<-hw(tmpTS[,],24*7) # additive model
accuracy(fcastets4)
AIC(fit2)
plot(fcastets4)
summary(fcastets4)
```

Listing 3.3: Multiplicative seasonal exponential smoothing forecast in R

```
fcastets3<-hw(tmpTS[,],24*7) # multiplicative model
accuracy(fcastets3)
plot(fcastets3)
write.table(fcastets3,file="fcastETSMAN.csv", row.names=FALSE,
sep=";")
summary(fcastets3)
```

By solving equation 3.15 in order to $y_{t+h|t}$, a forecast, that varies accordingly to the smoothing parameters α, β and γ selected for a span of h steps ahead has given. Alternatively the same results can be obtained by running the additive Holt-Winters function of the R forecast package as exemplified in R code 3.2.

On the other hand, the seasonal multiplicative exponential smoothing method uses similar equations, however handled differently, as follows in the detailed equations:

$$y_{t+h|t} = (\ell_t + hb_t) * s_{t-m+h_m^+}, \tag{3.16}$$

where

$$\ell_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}); \tag{3.17}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}; \tag{3.18}$$

and

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}. \tag{3.19}$$

This can also be done resorting to the R program through the code reproduced on 3.3.

Furthermore, similarly to the classical decomposition, the difference between the equation 3.15 for the additive model and the equation 3.16 for the multiplicative model will translate into different proprieties on each model. The Holt-Winters seasonal additive method is favourable when dealing with seasonal variations, if these are practically constant throughout the series, as it is patent when analysing the time series plot 3.5. On other hand, Holt-Winters seasonal multiplicative method is more apt when the seasonal variations change proportionally to the level of the series, as seen on the time series plot of figure 3.6 from the same source.

Note that on figure 3.6, the amplitude of the seasonal pattern increases similarly to the growth on the series level, while on the data present in figure 3.5 the amplitude is clearly constant. Nevertheless, analysing the water demand data provided from a typical Water

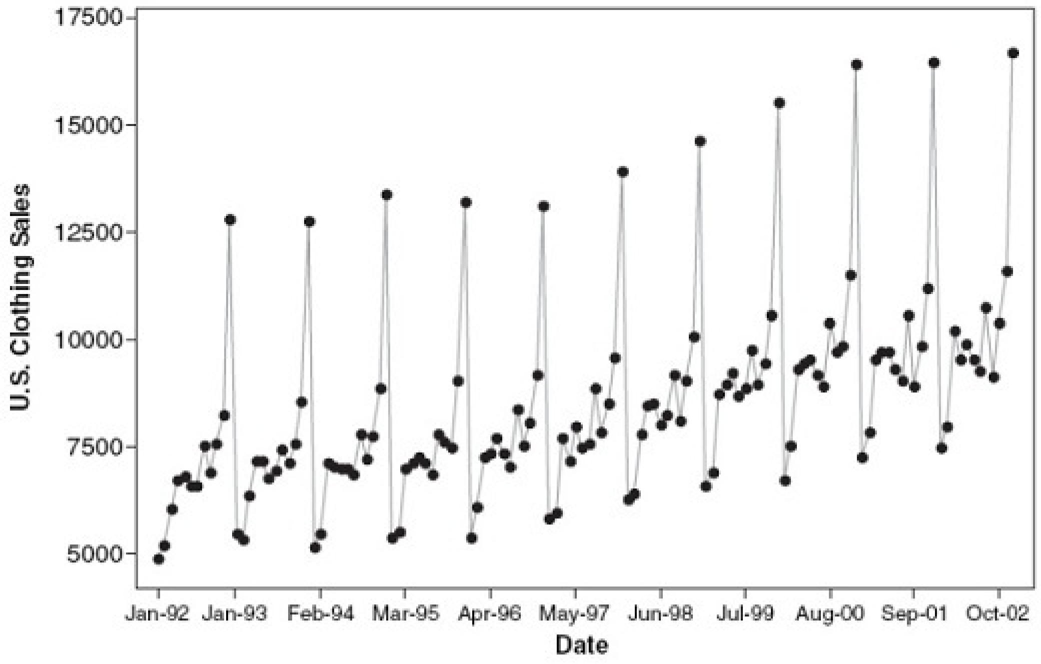


Figure 3.5: Time series plot of U.S. clothing sales from January 1992 to 2003 [15], representative of the Holt-Winters seasonal additive method model

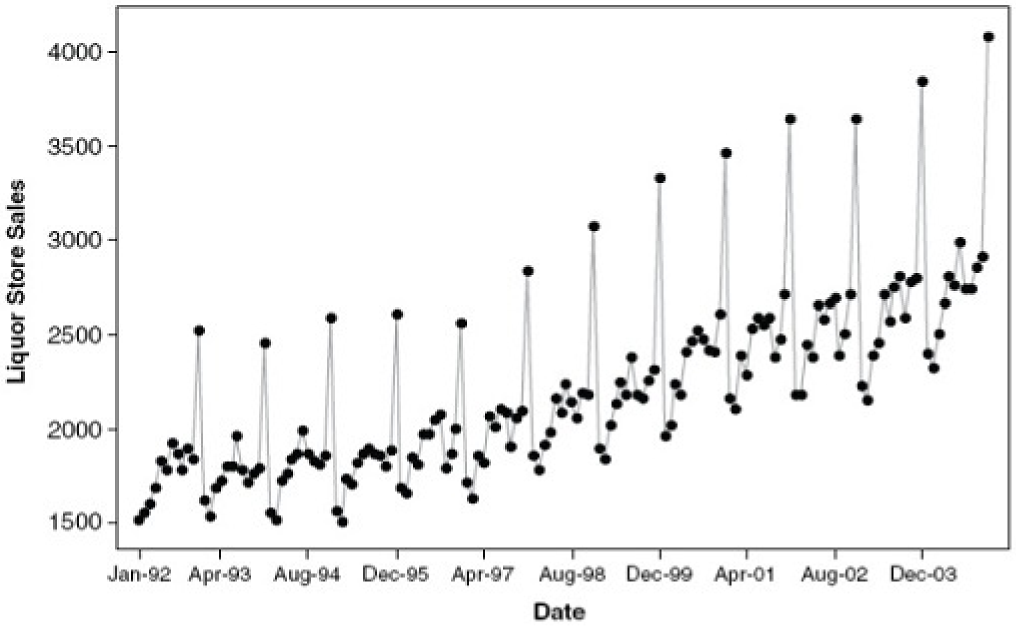


Figure 3.6: Time series plot of liquor store sales data from January 1992 to December 2004 [15], representative of the Holt-Winter seasonal multiplicative method model

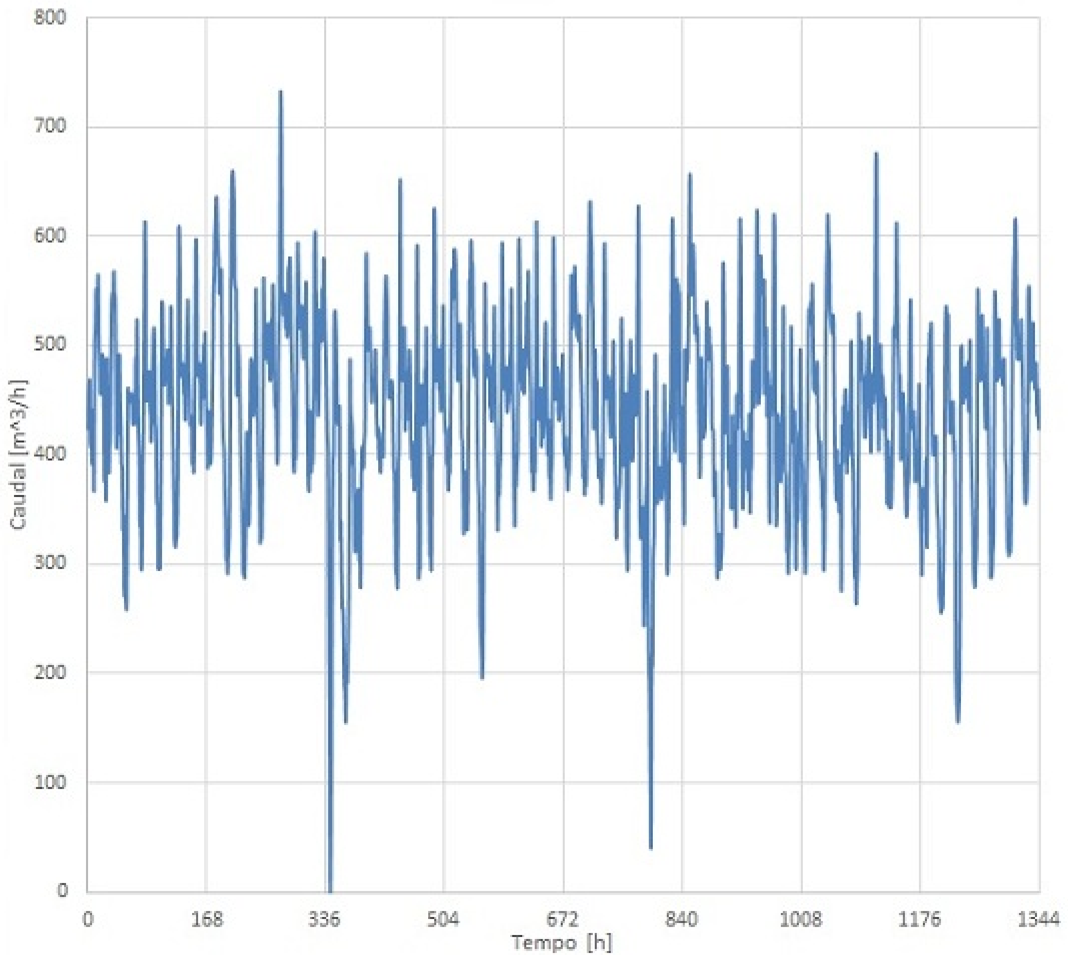


Figure 3.7: Time series plot of a water demand example

Listing 3.4: Multiplicative decomposisiton in R

```
#Multiplicative decomposition
mM<-decompose(tmpTS, type=c("multiplicative"))
plot(mW)
```

Supply System (WSS), in light of these series, it is not clear which of these categories is more suitable, as presented on figure 3.7, for it seams a middle point between the additive and the multiplicative model.

To comply with these circumstances, it was used the time series decomposition tool from the R function *decompose*, exemplified below in coding example 3.4. It allowed to clarify the read, not obvious so far, when observing the raw data. As it can be verified in figure 3.4, under the *Trend* category, the amplitude of the variation of the series has the tendency to fluctuate with a close relation to its level, although not absolutely proportional.

After implementing the two exponential methods, it is clear that the growth in the amplitude of the seasonal component is not absolutely proportional to the trend level. This means that, in the case of these applications experiment, the MAPE obtained from a 24 hour forecast was of 12.85% for the multiplicative exponential method against 6% of the additive exponential smoothing method, as present in table 4.2. In light of these results, the additive method will be significatively favoured.

The popular ARIMA model was also tested as a basis for comparison. Contrary to the previous methods suggested, which assume that the series can be represented by a sum of deterministic and stochastic components, the Autoregressive Integrated Moving Average (ARIMA) model finds repeating patterns obscured by noise in the data set. This algorithm works by searching correlations between the randomness of the three components of a time series, outputting a model structure that represents the given inputs. This method is not usually used for modelling seasonal data, nor was it developed with that purpose. However, if the data is deseasonalized and made stationary, ARIMA can be used with success.

One of the assumptions to use a ARIMA model is that the series is stationary. The stationarity of a time series is related to its dependence of time, as Rob J. Hyndman states by affirming "a stationary time series is one whose properties do not depend on the time at which the series is observed" [55]. That said, a rough way of analysing if a time series is stationary, is by taking supposed snapshots at different times/places of a series and comparing if they look much the same. An example of a stationary time series may be the daily change, in Dow Jones index on 292 consecutive days, patent in figure 3.8. It may resemble some sort of white noise and that is the objective when searching for a stationary series. Also the Auto Correlation Function (ACF) [56], whose plot is represented in figure 3.10, should drop quickly and show significant spikes related to the expected seasonality throughout the series, in this case, on a frequency of 168 as seen on the Partial Auto Correlation Function (PACF) [56] plotted also in figure 3.10.

If a series is non-stationary, as is the data in question, in order for this algorithm to work, the data of water demand patent in figure 3.7 will have to be made stationary by virtue of seasonal differentiation. Seasonal differentiation is much the same as differentiation, with the twist that the index subtracted from the series is the one observed in the previous

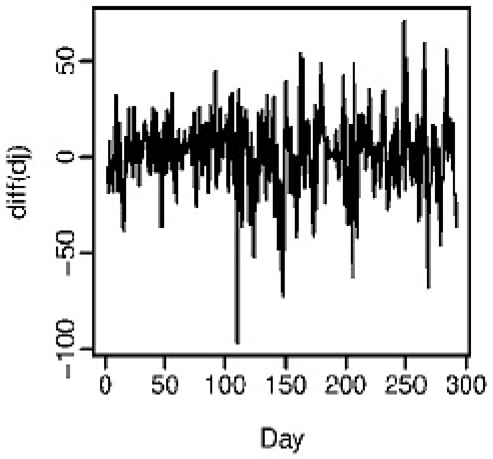


Figure 3.8: Daily change in Dow Jones index on 292 consecutive days [55]

Listing 3.5: Differenciate series

```
#Multiplicative decomposition
differenciated_series<-diff(diff(tmpTS,168))
tsdisplay(differenciated_series)
```

season.

It may happen that after seasonal differentiation there may still reside and underlining trend in the series, as patent on the data analysed on figure 3.9. In this case a general differentiation should be applied, as it is expressed like the first difference of the seasonal difference illustrated on figure 3.10, obtained by running the code 3.5 in R.

These tools enable the correct choosing of the particular model that best suits the data set. Nevertheless, other neighbour models should be tested as to its fitting, best described by the Akaike Information Criterion (AIC) indicator. This indicator measures the quality of the statistical model, for a given data, by testing the error of fitting, and computing the log-likelihood function against the number of parameters fitted.

Regardless of doing all this process, the R forecast package supplies an automated ARIMA function. This is particularly practical because it computes all the ARIMA models choosing the most appropriated method by finding the global minimum of the indicator AIC, as proven to be adequate on the M3 competition [57], exemplified in the R code 3.6.

In the next chapter, the effectiveness of these methods will be discussed as to their

Listing 3.6: Differenciate series

```
fit2<-auto.arima(tmpTS, stepwise=FALSE, approximation=FALSE)
plot(fit2)
print(fit2)
fcastfit2<-forecast(fit2,h=24) #forecasting for 24h
```

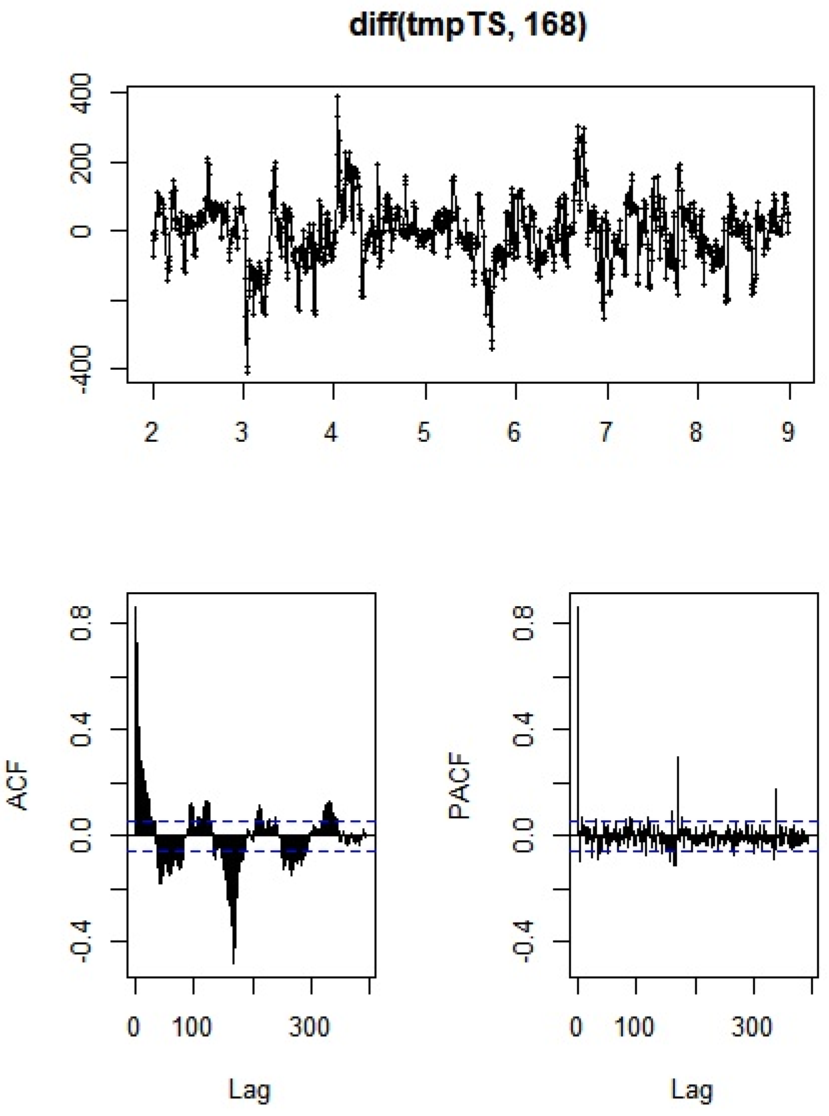


Figure 3.9: Seasonal differentiation

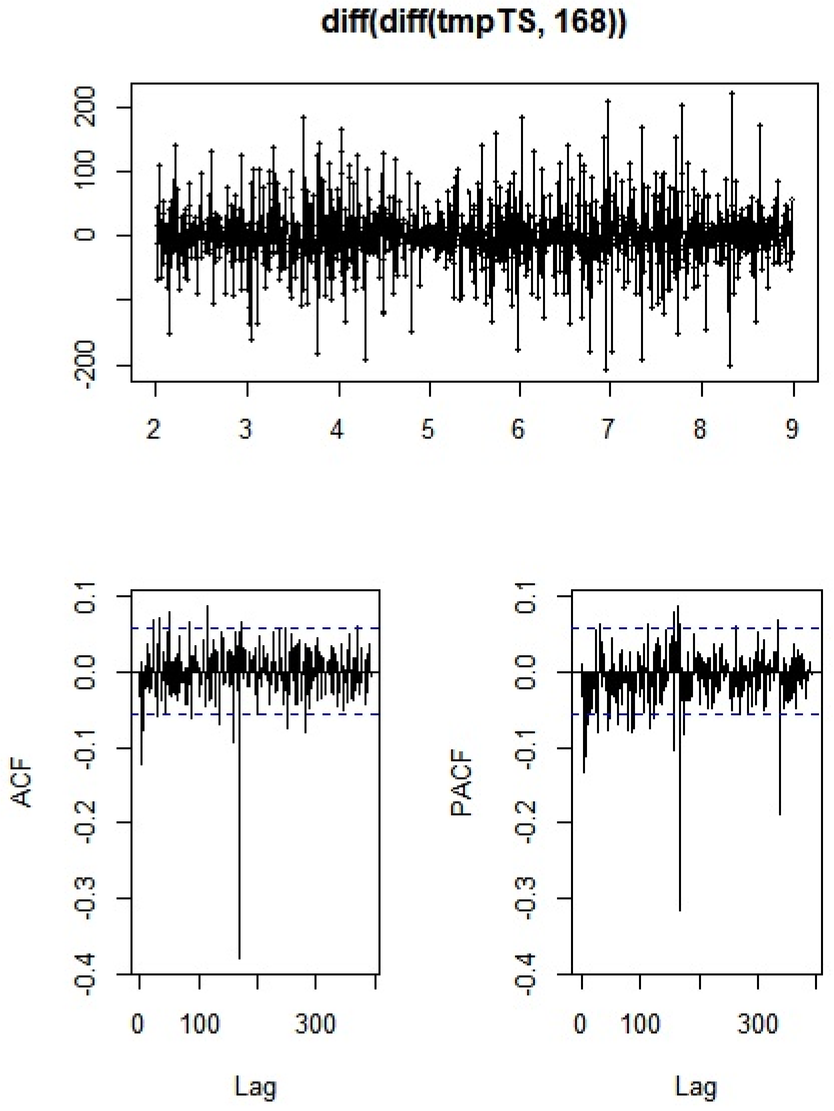


Figure 3.10: First difference of the seasonal difference

MAPE and maximum deviation from the reference values. These were the two criteria chosen to select the most satisfactory method, due to their characterisation by the general fitting of the model, as well as point deviations excessively far from the reference.

3.4 Comparison with ANN

As stated in the previous chapter 2.3, ANN are the state-of-the-art regarding data analysis and forecast. As a part of the project, in which this thesis was inserted on, José Vagos, team member of the project, developed another application for the forecast of water demand, implementing a system of ANN, based on the same set of data that has been used in this thesis and for the same purpose.

Both forecasting by either ANN or time series, use a data set, preferably long, that will be run as the training data, meaning that this data will be use to tune the model.

In the case of the ANN, a neuron-like switching network, weighted interconnected among the units, will be calibrated accordingly to the error. This is done automatically resorting to several iterations, analysing the relation between input and output, the outcome being the weights of the best model. In the case of the time series, for example the Holt-Winters model, a similar process is used, resorting to training data to select the best smoothing parameters, as explained in chapter 3.3. On the other hand, the testing data is used as a evaluation of the model, for quality measuring purposes, not for the tuning of the model. These types of data (training and testing) should not overlap.

The two different approaches to the water demand forecast were compared after being employed with similar sets of data in terms of time span.

The fact that the data set was not the same is connected to its internal characteristics, which imply that the training of the ANN uses a more complete set of data, consisting on: date and time, air temperature, humidity, amount of rain fall and registered water demand. All of these values were registered hourly for a period of two months.

The ANN algorithm will then relate every single set of data with all the others, and tune the weights of each connection.

By comparison, the time series algorithm uses a simpler set of data, consisting plainly in date time and registered water demand.

In the following set of graphs, figures 3.11 and 3.12, one can visualize each of the algorithms response to the given training data. The graphs present the forecasted water demand, plotted in red , versus the latter registered water demand on the same period, plotted in blue.

Calculating the Mean Absolute Percentage Error (MAPE) of each algorithm from the date plotted in figure 3.11 and figure 3.12, it is possible to infer the superiority of the time series forecast for a one day forecast, with an error of 6.18% versus 7.68% by the ANN algorithm. However, analysing a week long forecast, the ANN algorithm has in fact a superior performance, with an error of 6.45% versus 8.34% by the time series forecast.

Given that a one day forecast is, on this case more relevant, the time series approach would be more suitable for this application.

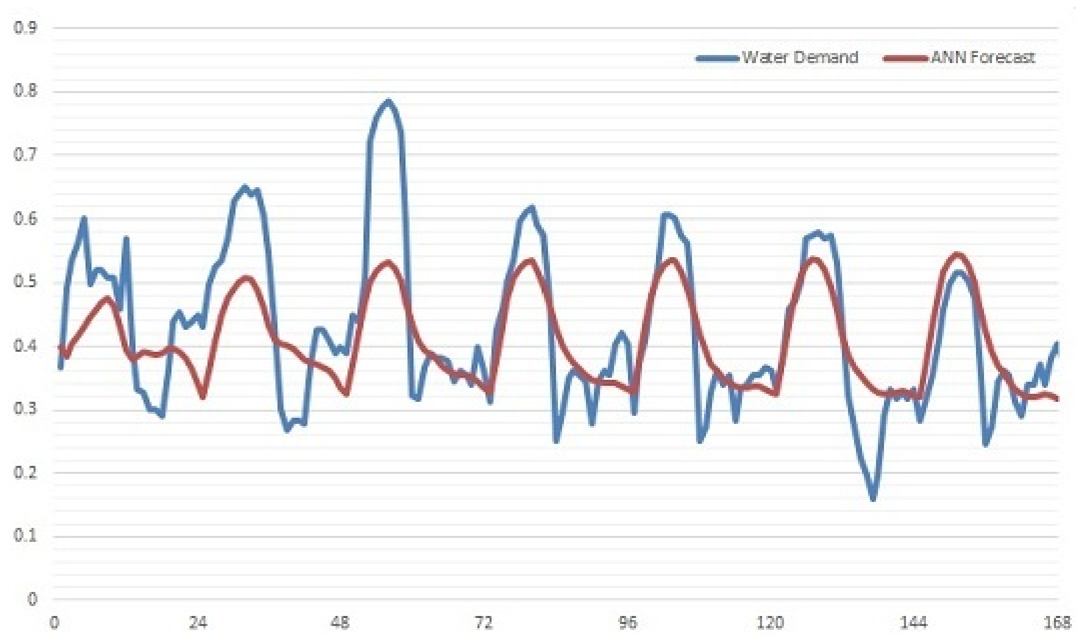


Figure 3.11: ANN model plot

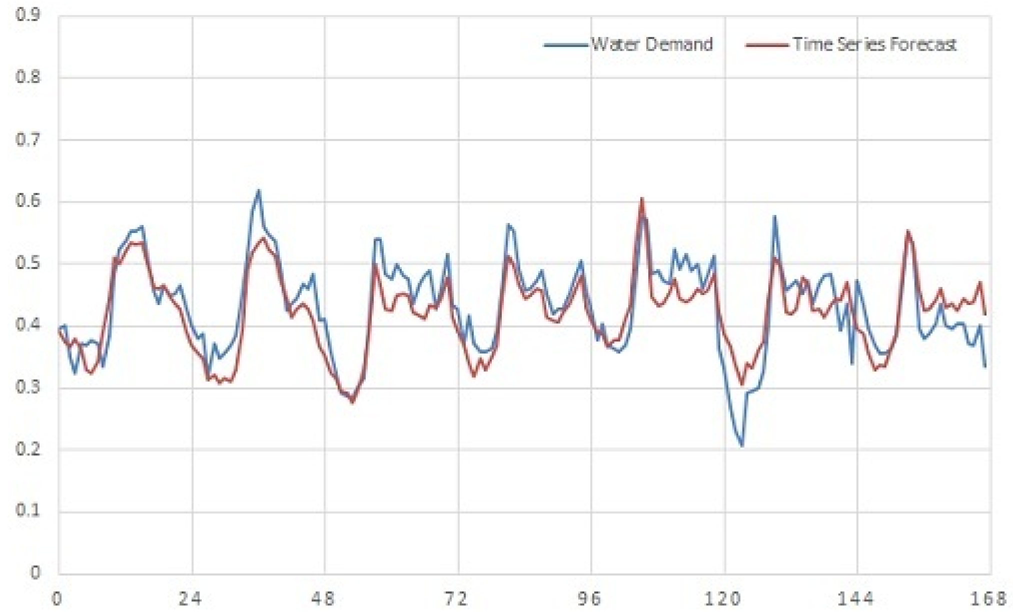


Figure 3.12: Time series model plot

Chapter 4

Results and Application

4.1 Project's automation

In this chapter the results from the forecast's accuracy relative to the reference values will be debated. Also the developed software, written in light of those results, will be presented, as well as the solution for supervision and control of the application in question. In chapter 3.2, a specific solution for gathering the Águas do Douro e Paiva (ADDP) data, which consisted in placing a *.DBF file in an File Transfer Protocol (FTP) location, is detailed. This solution enabled access to their data, and allowed the development of the project on an early stage. However, this would not be acceptable as an universal definitive solution, by virtue of not having the ability of being generalized. Having this in mind, in this section the OPC protocol is explored by using two programs designed to automate the supervision and control routines.

As also explained in chapter 3.2, a way to solve the problem of being able to interoperate with any plant, is to connect directly to their OPC server. Situated, virtually, between the PLCs (data source) and the HMI (data sink), the OPC has the ability to both read from and write to devices. To operate the OPC server in such way, the OPC Foundation library for C++/C# [5] was used to link the program developed to the server. This library makes it seamless to operate devices, such as PLCs, through third party application.

The developed software was designed to operate in the background, meaning that it would not be visualized by the end user. Its objective was to, accordingly to the time interval specified, get the information from the defined tags and save the information to the project's database.

Due to the high importance of this data, it is mandatory that none of it is lost. Having this in mind, the data gathering program developed uses a temporary local file to save the data, given the case that the database connection is offline. The program also checks for the quality of the information and writing it to the actuators. For instance if a data source is disconnected, the OPC server will flag it as an unreliable tag, which is then inserted flagged as "-1" into the database. A diagram of the work flow used is presented on figure 4.1.

Another separate program was written to execute the task of writing to the tags. In other words, the goal of this second program was to operate the pumps through communication with the PLCs. This program works in a similar manner to the previous, using the same OPC library and similar commands. The difference resides on it working by checking a database filled by the optimization program. This database contains the

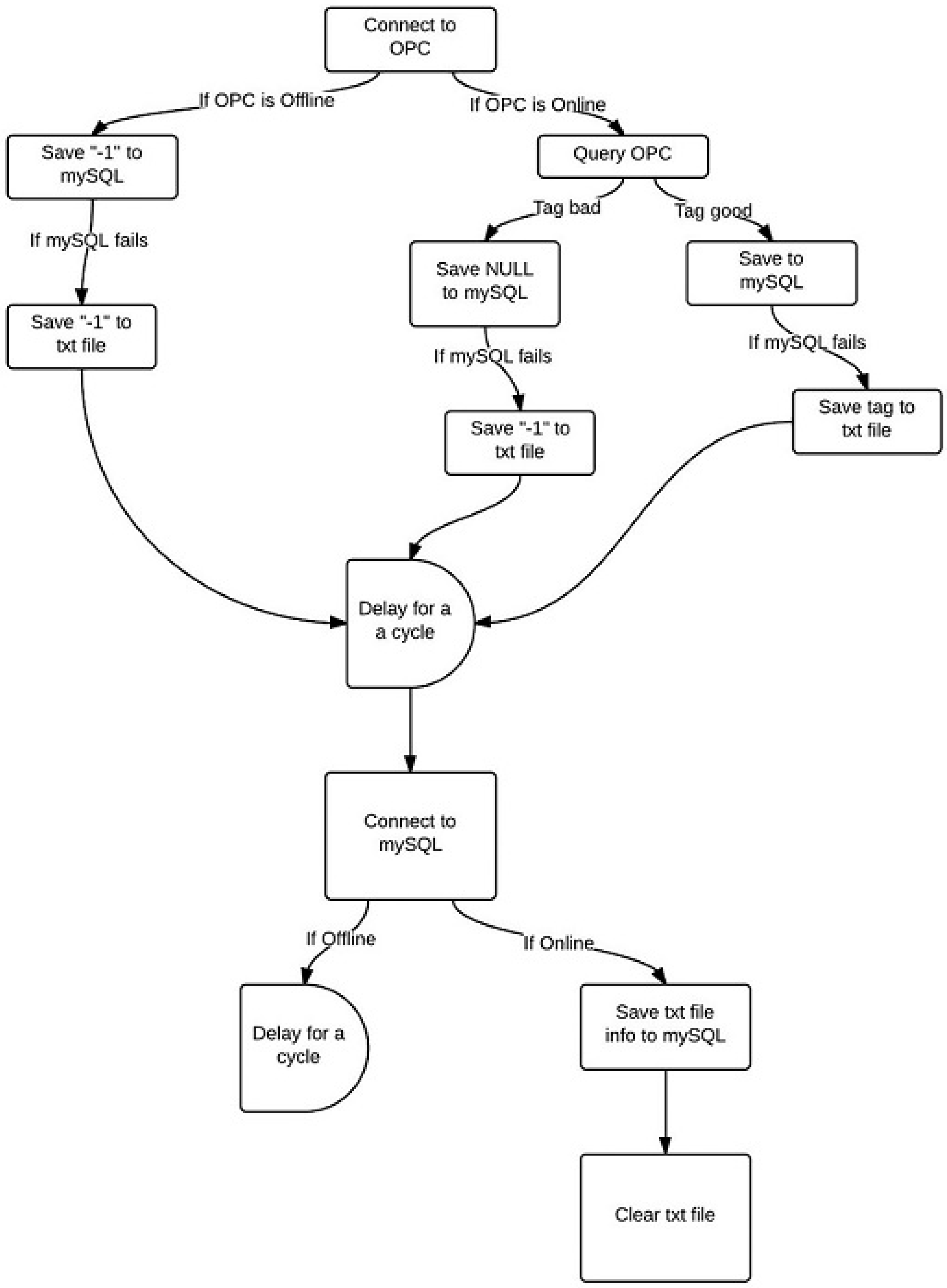


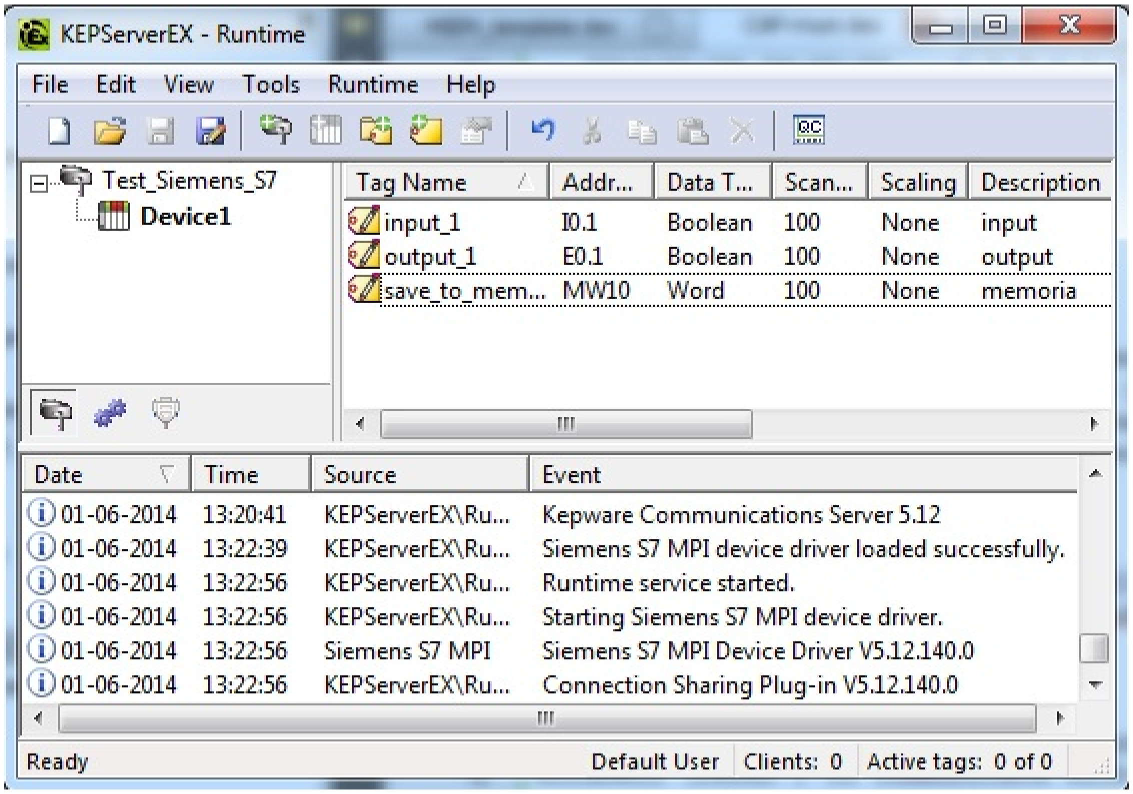
Figure 4.1: Schematic of the information flow on the OPC program

Table 4.1: Pump operation schedule

start datetime	Work duration [minutes]	Pump Speed
2014/01/20 00:00:00	1	2
2014/01/20 01:00:00	1	2
2014/01/20 03:00:00	1	2
2014/01/20 04:00:00	0.5	1
2014/01/20 05:00:00	0	0

optimal schedule and speed to operate the equipment, providing an easy reading and integration within the program. An extract of this database is presented on table 4.1 .

In order to test the developed software, a laboratory installation was set, resorting to a local OPC server and to a PLC simulator. This implementation was based on the Kepware software, which is a communication and interoperability software. This software supports an array of open standards, as well as proprietary communication protocols, which made it ideal for testing and debugging. It was however a trial version with a two hours limitation, which impose boundaries as to its utilization. An example of the KEPSeverEX running, is presented in figure 4.2.



Tag Name	Addr...	Data T...	Scan...	Scaling	Description
input_1	I0.1	Boolean	100	None	input
output_1	E0.1	Boolean	100	None	output
save_to_mem...	MW10	Word	100	None	memoria

Date	Time	Source	Event
01-06-2014	13:20:41	KEPServerEX\Ru...	Kepware Communications Server 5.12
01-06-2014	13:22:39	KEPServerEX\Ru...	Siemens S7 MPI device driver loaded successfully.
01-06-2014	13:22:56	KEPServerEX\Ru...	Runtime service started.
01-06-2014	13:22:56	KEPServerEX\Ru...	Starting Siemens S7 MPI device driver.
01-06-2014	13:22:56	Siemens S7 MPI	Siemens S7 MPI Device Driver V5.12.140.0
01-06-2014	13:22:56	KEPServerEX\Ru...	Connection Sharing Plug-in V5.12.140.0

Ready   Default User   Clients: 0   Active tags: 0 of 0

Figure 4.2: Example of the KEPwareServerEX running

The following step was to install the same software used in the ADDP's station. This station uses the RSLinx Classic by Rockwell [44] to communicate with their Allen-Bradley PLCs. A simulation of the same type of management used in ADDP's plant was also set in the laboratory previously discussed above. The RSLinx version used was the Gateway version, which was at the time the most complete version available. Features both the integration of an OPC server, an OPC client and communication with the RSLogix PLC Emulator, being one of the main programs to manage Allen-Bradley's PLCs. This program is also certified to compile under the OPC Foundation's specifications, granting an easy integration with a third party non proprietary application.

The RSLogix 500 software [58], was then used to write a simple Ladder program, which activated an output, given certain input information, as well as featured an auto resetting counter, as it can be perceived in figure 4.3. In this image, CTU means counter, incremented by the toggling a button. This counter would then reset after three actions, by virtue of the equal (EQU) function implemented below. The Ladder program also features by last, an ordinary input/output direct cause/effect routine.

The RSLogix Emulate 500 [59] was used as a PLC simulator to run the previously explained Ladder routine, making it possible for the OPC server to actually connect to an object, and execute read and write actions.

Figure 4.4 shows a simple way to confirm that everything is working properly, by using an OPC client application. This application connects to the OPC server, displays the selected variables, and it also has the ability to write to the devices (*e.g.* pumps), through the server. Neither the OPC server and client, nor the ladder program used to test this module represent any specific system, they are merely a way simulate simple communications that would later be substituted by effective applications.

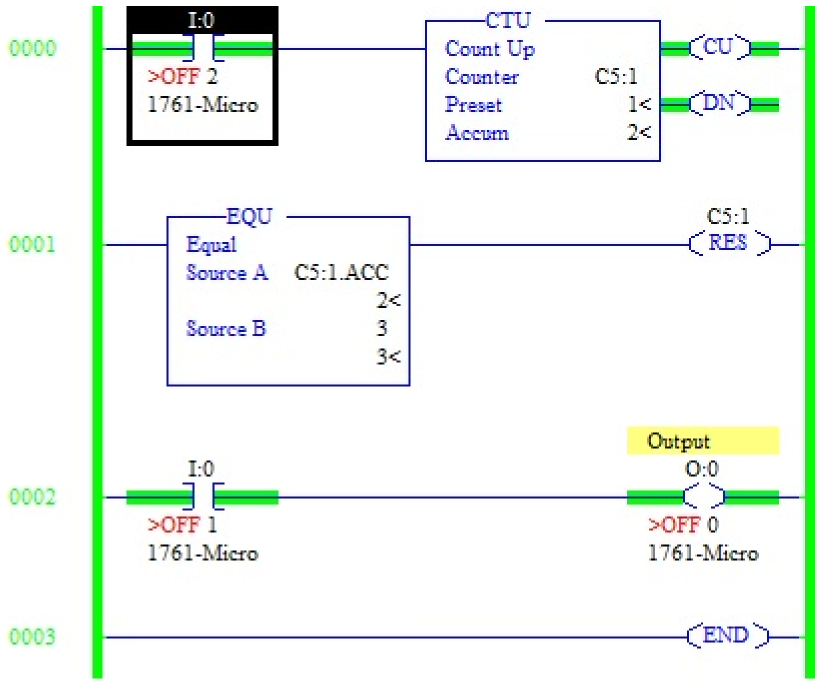
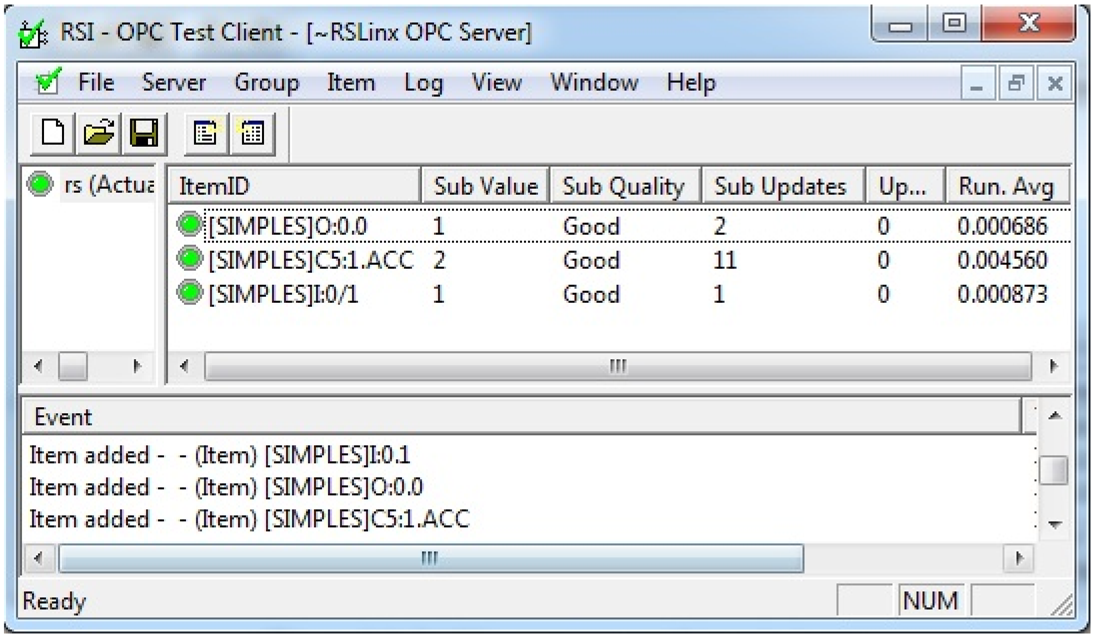


Figure 4.3: Ladder program used on a test run

Using this debugging method insured that the developed programs, both to read and write, were error free as well as insured their flexibility as to the changes on the number and type of tags.

All the programs referred were exhaustively tested in laboratory, prior to their


RSI - OPC Test Client - [~RSLinx OPC Server]
File Server Group Item Log View Window Help
rs (Actua
Ready
NUM

ItemID	Sub Value	Sub Quality	Sub Updates	Up...	Run. Avg
[SIMPLES]O:0.0	1	Good	2	0	0.000686
[SIMPLES]C5:1.ACC	2	Good	11	0	0.004560
[SIMPLES]I:0/1	1	Good	1	0	0.000873

```
Event
Item added -  - (Item) [SIMPLES]I:0.1
Item added -  - (Item) [SIMPLES]O:0.0
Item added -  - (Item) [SIMPLES]C5:1.ACC
```

Figure 4.4: OPC client test by RSLinx

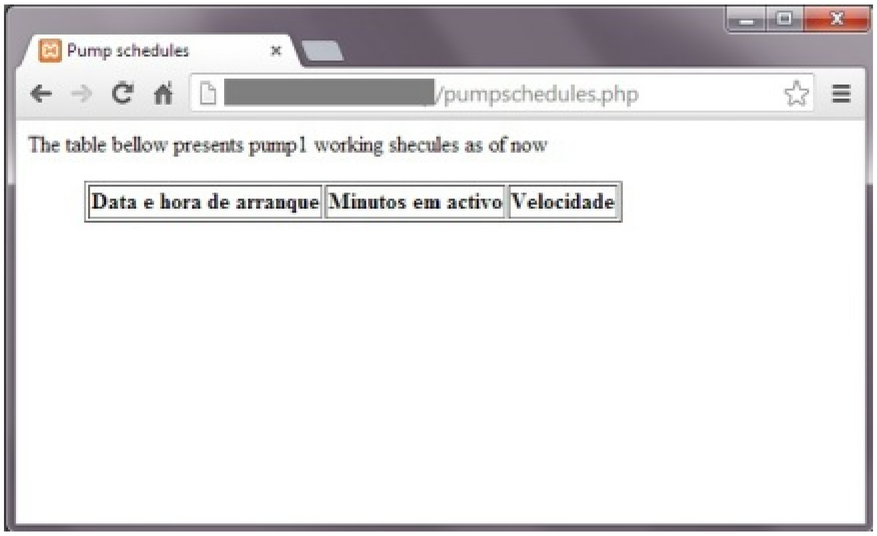
installation on the Águas do Douro e Paiva (ADDP) Water Supply System (WSS), even though they just suggested the control schedule, not being actually connected to the pumps or valves. These results were encouraging to the continuation of the project.

4.1.1 Manual control alternative

As stated in chapter 3.2, this project is still in an early stage. Due to this this fact, an alternative to the fully automated command was requested. The prerequisites of this alternative were that it should demand no installation on any device and it should display the information about the pump and valve optimum control to the plant's operators in an straightforward structure.

The clearest solution for this problem was to set an website that would display the database's tables in any computer browser. Using the MySQL to PHP linking library [60], a complete display of the projects data base, displayed in figure 4.5, is available in every computer with access to the local work network. Note that the main software tool detailed in section 4.4, is still required so that the forecasting and optimizing tools can be run.

The results must be then approved by the plant's manager before being sent to the operators. This means that the feedback cycle is not closed, so, in the following iteration, the program will not assume that its suggestions were followed by the operators.



The table bellow presents pump1 working shecules as of now

Data e hora de arranque	Minutos em activo	Velocidade

Figure 4.5: Online display of pump controls

4.2 Water demand forecast

The five methods discussed in chapter, 3.3, (classical additive and multiplicative decomposition, additive and multiplicative exponential smoothing, ARIMA methods) were tested by running a 24 hour forecast. The data on which they were based, was the previous eight weeks of hourly records from the water consumption of a WSS.

Table 4.2 shows the Mean Absolute Percentage Error (MAPE) of each forecast, against the reference values, that is, the actually recorded water consumption on the forecasted period. This error is calculated according to equation

$$\text{MAPE} = mean(|\frac{y_i - \hat{y}_i}{y_i}| * 100), \tag{4.1}$$

given y_i, the ith record, and $\hat{y}_i$, the ith forecast. MAPE constitutes the main cost function used in this thesis. The maximum deviation from the reference values, also patent in table 4.2, is found by determining the maximum percentage error, before calculating the mean, being a secondary selection tool. Tables 4.2 and 4.3 show those two criteria, used to assert the global accuracy of the results discussed below in this section.

Water consumption features seasonal characteristics due to several factors such as weather, *e.g.*, air temperature, rain fall, humidity, or even agricultural activities, among others. These features can be translated on a yearly general trend, which is designated as the cycle on seasonal adjustment, as illustrated on figure 4.6. This trend has significant importance when planning long-term strategies, or producing long-term forecasts as presented on, chapter 2.3, for dimensioning new stations or even strategizing water resources. However, water demand also displays an expressive weekly seasonality, which translates to each day of the week being evidently different from each other but with a repeating pattern, visible in figure 4.7. This picture shows a month of water demand, taken from the recorded data on which the results are based. Given that only eight week of data were available the cycle component was neglected, due to insufficient data for it's computation.

Table 4.2: Mean error on the results of the application of all the methods to a 24h forecast with relation to the recorded data

Method	MAPE[%]	Max. Deviation[%]
Holt Winters Additive Exponential Smoothing	9.30	22.35
Holt Winters Multiplicative Exponential Smoothing	12.66	37.58
ARIMA(4,1,3)(0,1,1)$_{168}$	5.22	20.06
Classical Additive Decomposition	6.15	17.12
Classical Multiplicative Decomposition	6.18	16.87
ETS Additive	6	30.01
ETS Multiplicative	12.85	36.98
ETS Additive 24 hour frequency	10.62	27.40

As a result of this inference, and considering that the aim of this study is to obtain a short-term forecast, priority was given to a 168 entries per season frequency rate. The number 168 was chosen by analysing the given data, which is consists of eight weeks of hourly data, hence $freq = 24 * 7$, 24 hours a day, and 7 days a week. A frequency of 24 entries per season, considering hourly data, was also tested for the exponential smoothing method, but, and as predicted, the results were not acceptable, derived from the fact that this frequency of seasonality is not adequate to the nature of the data.

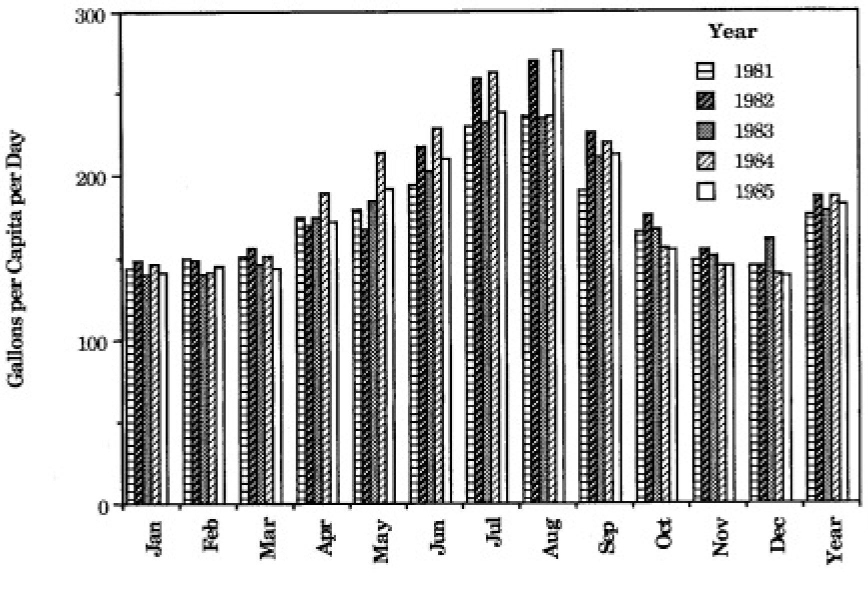


Figure 4.6: Monthly and annual water use by year [61]

The two criteria were used as a tool to describe the fitness of the model to forecast. By plotting the forecasts obtained with each method explained in the previous chapter, against the reference values set as the bold dark blue line on figure 4.8, it becomes clear which methods should be discarded and which deserve further analysis. The line in blue represents the Holt-Winters Multiplicative Exponential Smoothing whose forecast is notably over the reference values of the recorded data, so it will be acceptable to be discarded, for it is clearly the worst fit. This decision is also supported by the error

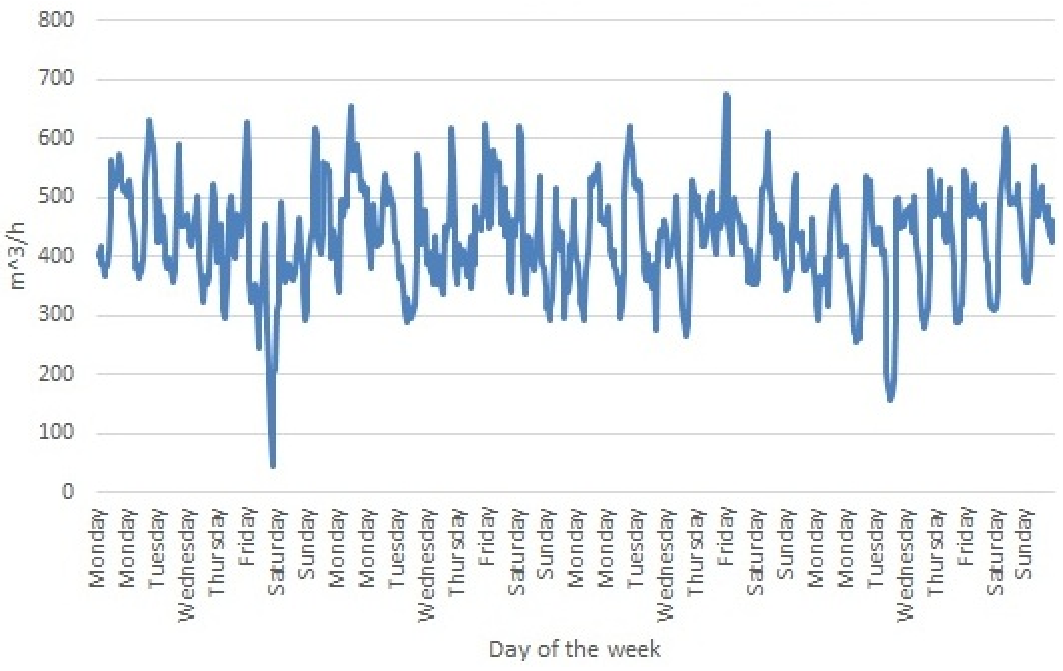


Figure 4.7: Weekly trend on a month of water demand

compilation on table 4.2 giving a MAPE of 12.85%. The same criterion can be used to exclude the Holt-Winters Additive Exponential Smoothing which only achieved a MAPE of 9.30%.

Both these two methods used constants α, β and γ to smooth the forecast. The values used were, respectively, $\alpha = 1$, $\beta = 0$, $\gamma = 1$ and $\alpha = 0.74$, $\beta = 0$, $\gamma = 1$. These values were obtained by running an optimization routine based on the equations 3.15 and 3.16, which run several iterations starting with the standard initial state, which is based on equations

$$L_s = \frac{1}{s}\sum_{i=1}^{s} y_i, \tag{4.2}$$

$$b_s = \frac{1}{s}[\frac{y_{s+1} - y_1}{s} + \frac{y_{s+2} - y_2}{s} + ... + \frac{y_{2s} - y_s}{s}], \tag{4.3}$$

and

$$S_i = y_i + L_s \tag{4.4}$$

for the additive model. The multiplicative model uses the same set of equations 4.2, 4.3 although using equation

$$S_i = \frac{y_i}{L_s} \tag{4.5}$$

for the seasonal section. For these to work, at least one complete seasonal cycle is needed. Finally, the best parameters are then chosen by computing the least mean square error given between the actual series and the smoothed series.

The ETSs, presented on the graph in figure 4.8, are time series forecasting methods applied with different parameters, the automation of its implementation is supplied by R project library accessible at [62]. These methods are based on the Holt-Winters

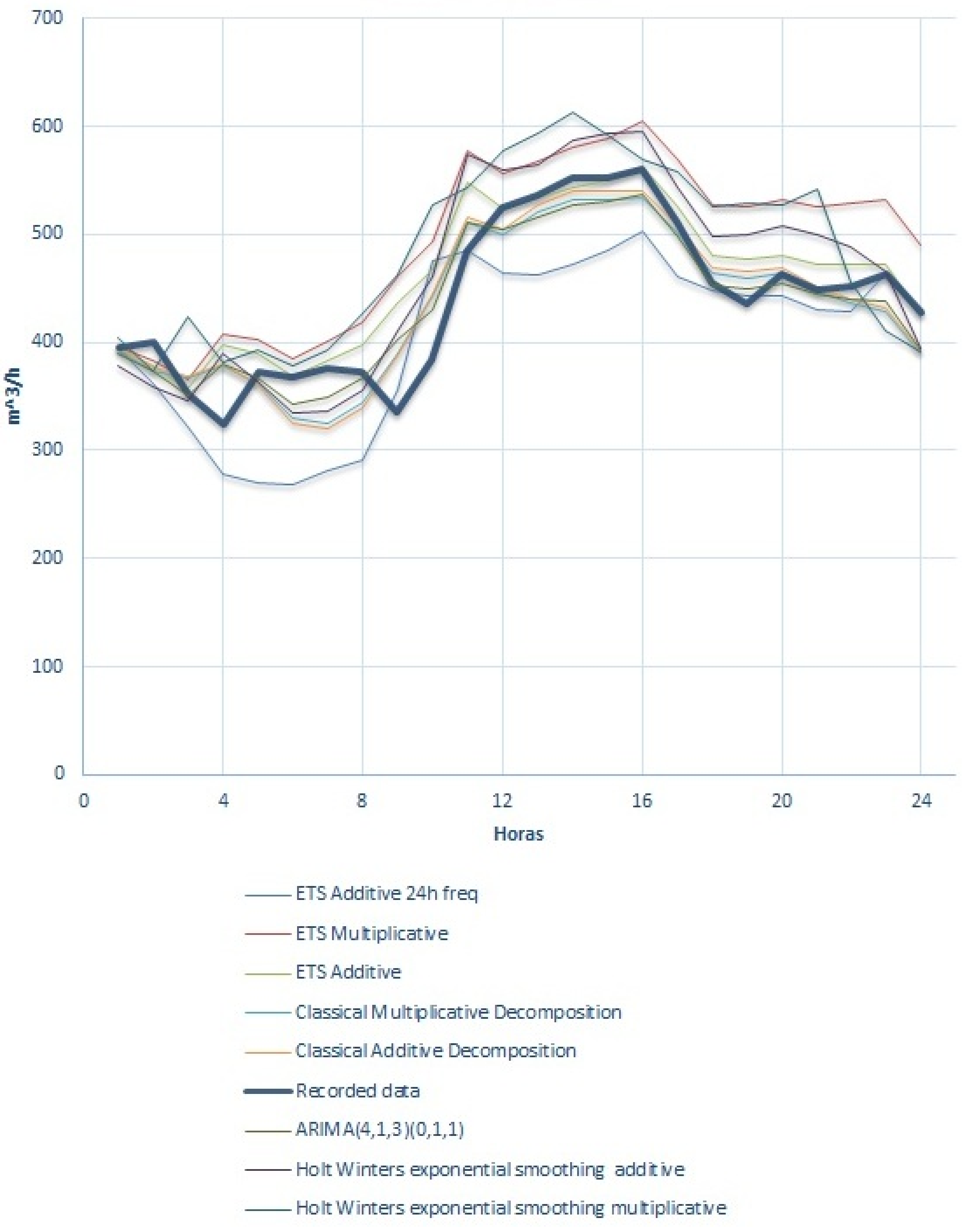


Figure 4.8: 24 hour forecast for each method

exponential smoothing, following, however, a different path from the previously in what concerns the parameters initialization. On the ETSs the initial state for the optimization of the equation parameters, similar to Holt Winter's equations, is predicted based on the maximization of the series log-likelihood function.

The likelihood function is a statistic method, based on the probability density function from the time series, which is inserted on the algorithm. Ultimately, this means that, although the ETS and the Holt Winters exponential smoothing methods are similar, their parameter finding algorithms will differ on the initialization routines, as well as in the parameter selection objectives, resulting in different forecasts. The ETS will in general achieve better results, due to the fact that it will often find the global minimum point of the algorithm, contrary to the Holt Winters routine which may be held in local minima.

The ETSs were run on the additive and multiplicative forms, applying a seasonal frequency of 168 hours, as done for the previous methods, achieving a MAPE of 6% and 12.85% respectively. The second algorithm, ETS Multiplicative, was discarded for it clearly does not properly model the series, running constantly several steps above the reference. As the additive model was much more promising, despite the fact that at some points it escalated to maximum deviations of 30%, which may be alarming for it may induce error on modelling the WSS profile. This method is more easily understood on figure 4.8 and table 4.2. Given these attributes, the additive method was not discarded, and was later plotted against the selection of figure 4.9.

Another experiment was run with the ETS algorithm to assert the presupposition explained previously, which stated that this type of time series, water demand, did not have a 24 hour per season frequency. This experiment resulted on a forecast with a 10.62% MAPE being a fairly bad interpretation of the series. This was expected, for this seasonality would mean that no correlation was found between the days of the adjoining weeks, which was already confirmed to exist.

Models composed of ARIMA without differentiation were tested but not included on this result's layout due to clearly not representing a model of the data, presenting instead a Mean Absolute Percentage Errors (MAPEs) superior to 10%. As previously stated, the raw data does not comply with the ARIMA requisites, as it is not stationary, leading henceforth to inappropriate modelling. Therefore, some pre processing was first endured in order for it to work.

As explained on the previous chapter, one of the conditions of the ARIMA model is that the data input must be stationary, *i.e.*, without trend or seasonality. One of the techniques to make a stationary series is differentiation. On the particular case of water demand, the data is first differentiated, and then seasonally differentiated, meaning differentiation in relation to the seasonality frequency. This process will output the plot on figure 3.10, a stationary series, identifiable by its resemblance to white noise.

Following this data handling, several models were tested by R, whom uses the AIC as the selection criterion to find the model which provides the minimum AIC. This method of optimization is explained in detail on chapter 3.3.

It was concluded that the model that best represented the data supplied was the ARIMA(4,1,3)(0,1,1)$_{168}$, meaning an standard component of Auto Regressive (AR) of 4, differentiation (diff) of 1 and Moving Average (MA) of 3, as well as a seasonal component composed by an AR factor of 0, a diff of 1 and a MA of 1. The equation

$$\begin{gathered} y_t = c_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \theta_3 e_{t-3} + \Theta_1 e_{t-168} + \theta_1\Theta_1 e_{t-169} + \theta_2\Theta_1 e_{t-170} + \theta_3\Theta_1 e_{t-171} + \\ + (1+\phi_1) y_{t-1} + (\phi_2 - \phi_1) y_{t-2} + (\phi_3 - \phi_2) y_{t-3} + (\phi_4 - \phi_3) y_{t-4} + \\ - \phi_4 y_{t-5} + y_{t-168} - (1+\phi_1) y_{t-169} + (\phi_1 - \phi_2) y_{t-170} + \\ + (\phi_2 - \phi_3) y_{t-171} + (\phi_3 - \phi_4) y_{t-172} + \phi_4 y_{t-173} \end{gathered} \tag{4.6}$$

Table 4.3: Mean error on the results of the application of the selected methods to a 24h forecast with relation to the recorded data

Method	MAPE[%]	Max. Deviation[%]
Classical Additive Decomposition	6.15	17.12
Classical Multiplicative Decomposition	6.18	16.87
ETS Additive	6	30.01
ARIMA(4,1,3)(0,1,1)$_{168}$	5.22	20.06

represents the development of the ARIMA modelling equation

$$ARIMA(p,d,q)(P,D,Q)_m. \tag{4.7}$$

This was in fact the method which provided the best MAPE at 5.22%, making an interesting approach to the series, although deviating 20% on some points of the 24 hour forecast. It led to the two more interesting time series models tested, the Classical Additive Decomposition and the Classical Multiplicative Decomposition, despite being two of the simplest ways of analysing a time series, often used as teaching examples. They did, in fact, provide an excellent model of the fed data, achieving a MAPE of 6.15% and 6.18% respectively. These values were only slightly higher than the ARIMA, which was the best model of the study in terms of Mean Absolute Percentage Error (MAPE).

Having discarded the four less suited methods, an easier analysis of the remaining is provided by figure 4.9, in which a augmented view of the interest area is provided. This was done removing the area between 0 and 300 m^3/h so that a zoom could be applied on the vertical axis, allowing a better view of the model's behaviour.

All the remaining forecasts have very similar Mean Absolute Percentage Errors (MAPEs) oscillations between 5.22% and 6.20%, as is made evident on table 4.3. That said, the focus will be turned to the maximum deviation.

From the remaining methods, the worst fit is still the ETS model that, although not having the worst mean error, deviates severely from the reference data, meaning it won't suit the application. This leaves the, simple to compute, classical decomposition methods and the R provided ARIMA modelling, which although having a 1% lower MAPE, does not quite model the series so well in terms of maximum deviation, an important part for the type of application the team was designing. Having in mind that this forecasting tools will be included in part of a designed proprietary software, several questions must be raised as to the legal inclusion of R computing on such software, which shall be discussed on section 4.2.1.

By discarding all the already refered methods, the choice was left between the very similar additive and multiplicative classical decomposition. After implementing and comparing the two methods above, the empirical data confirmed what the theory proposed by the bibliography, namely Douglas [15], by presenting an MAPE of 6.18% for the multiplicative method and 6.15% for the additive method.

This result, and such small error difference, was expected, given that in the data used, the variation component and the level were not absolutely proportional as stated above. However, there is a more significant point to take into account, the maximum error local reference of 17.11% on the additive method versus 16.87% on the multiplicative method, which is an important factor as, considering the nature of the application, gives

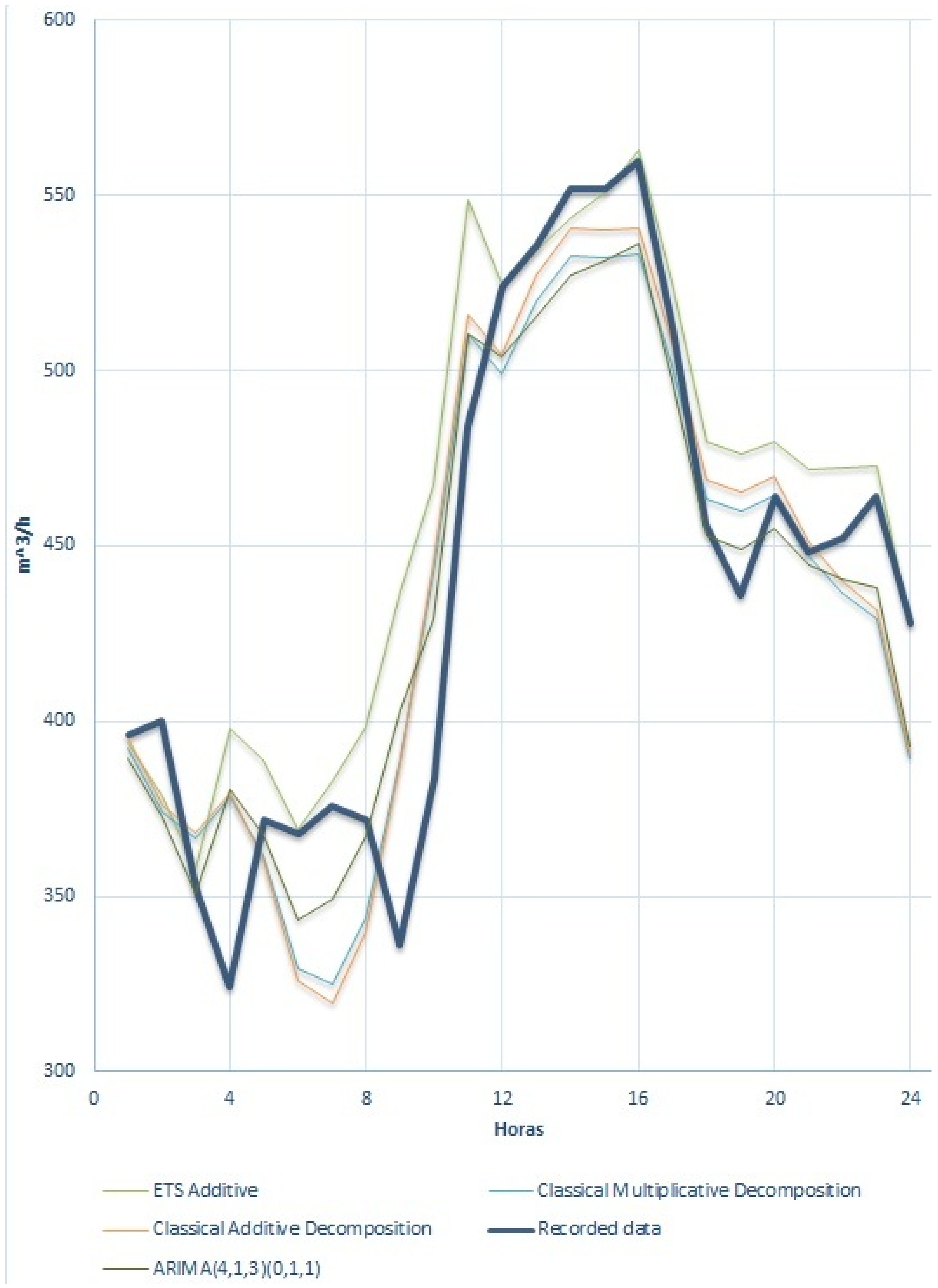


Figure 4.9: 24 hour forecast for each of the best suited methods

a higher degree of confidence to the user, and provides a better base for the optimization algorithm. Due to this result, the choice will lean towards the second method, the Classical Multiplicative Decomposition.

It's worth remembering that, although the comparisons made in this section are based on a 24 hour forecast, typically when forecasting a time series with seasonal component the forecast is produced for a complete period, in this particular case a week. However, as a result of the optimization of the pumping schedules relying solely on the following 24 hours, the remainder of the forecast was ignored.

4.2.1 Implementation of water demand forecast automation

One of the objectives proposed at the start, was to develop an automated water demand forecast module. In order to achieve that goal two options were available. The first was to use an linking library to the R project [51], described in chapter 3.3, and the second was implement from scratch the selected method on a C++ routine.

The R library for C# (R.NET) provided at [63], provides an easy way to integrate the .NET framework with the R statistical language. This allowed to write R routines such as the ones detailed in chapter 3.3, the codes 3.1, 3.2, 3.3 or 3.6. However this would require for the R program to be installed on the machine in which this routine runs. It also rose questions regarding the commercial use of such application by virtue of being under the GNU General Public License (GPL)v2 license. Although using R would be the ideal solution, because of these concerns the alternative route was taken.

This route would be to implement the Classical Multiplicative Decomposition time series method, seeing that implementing the ARIMA algorithm would be extremely time consuming. This alternative would not condition the performance of the forecasting tool in a significant manner, as debated in chapter 4.2, as the best performance model run by R achieved 5.22% Mean Absolute Percentage Error (MAPE), which compared to the Classical Additive Decomposition at 6.18% represents a difference of 1%.

Having considered this difference negligible, the Classical Multiplicative Decomposition time series method, was translated to C++, having been based on the Microsoft Excel implementation previously used. The input data is acquired by querying the project's database, enabling the program to run and finally save the produced forecast to the same database.

The automation of the forecasting module means that independently of the station in which it is implemented, a new model will be generated according to the new conditions, those being primarily the stations historical data. It also means that every time the program is run, typically every 24 hours, the parameters of the statistical model are recomputed taking into account the recent data inserted in the database (data in between executions).

It is worth mentioning that the recalculation of the model parameters takes only a few minutes. For instance, in a computer equipped with an intel core i5 and 4GB of RAM, working on a local network connection of 54Mbps, it took on average one minute based on a historical data of eight weeks. However it may vary according to the database size and data frequency. This particular test was executed using hourly data from the previous eight weeks, due to data quality constrains. Nevertheless, this variable can by tuned on the main GUI presented on figure 4.10 in chapter 4.4.

4.3 Project's Database

A database is, by definition an organized collection of data, stored on a local or remote machine called database server. Software applications such as MySQL [39], PostgreSQL [64], Microsoft Access [65] or SQlite [66], amongst others, are between some of the most popular applications of database management systems.

In this project a MySQL database management system was used, primarily for complying with the requirements necessary, being an open source software with commercial alternatives, as well as for being a popular application, extensively tested by all the team member's even before the project..

The database, at this stage is not normalized nor optimized. Although this is not an ideal solution, due to the possibility that, with the program's increase in size, it can became resource wasteful, it has the advantage of being easy to understand and to connect to. This benefit allows the separation of the project into different modules, which would in turn allow them to have different recording intervals without requiring any further development on the program's side.

In sum, to operate the fully running project E-Pumping [4], three different database schemas (aggregations of tables with the same structure) are needed. The schema's, table's or even column's naming can be entered dynamically into the program, meaning that there is no requirement for any specific naming. However, there are some requirements relativity to number and data type of the entries. Bellow it is detailed the rules set to the water consumption notes, which allow the correct functioning of the program.

For each of the nodes, *i.e.* consumption points, on a network, one table with the following columns must be created. The listed specifications must be followed:

1. **Datetime** - column of type datetime, being the key of the table (an unique index of each table that uniquely identifies an entry), stores the date and time of the entry registration;

2. **Caudal** - column of type float, stores the data from the recorded flows needed for the predictions;

3. **temp** - column of type float, stores the data from the recorded ambient temperatures needed for neural predictions;

4. **humid** - column of type float, stores the data from the recorded humidity needed for neural predictions;

5. **pluv** - column of type float, stores the data from the recorded pluviosity needed for neural predictions;

6. **prev_ets** - column of type float, stores the data from the time series prediction;

7. **prev_nn** - column of type float, stores the data from the neural network prediction;

8. **err_prev_ets** - column of type float, stores the difference between predicted flow (timeseries) and real flow(not yet implemented);

9. **err_prev_nn** - column of type float, stores the difference between predicted flow(neural networks) and real flow(not yet implemented);

This is the general database for the complete project. As so, there are several columns not needed for the modules developed in this thesis, such as the **temp**, **humid**, **pluv**, **prev_nn**,**err_prev_nn**. These contain information required for the forecast module based on Artificial Neural Network (ANN) algorithms develop by José Vagos.

For driving the pumps, another schema was created, whose tables should resemble table 4.1. The number of tables in that schema should equal the pumps to control. Their composition was set as follows:

1. **Datetime** - column of type datetime, being the key of the table, stores the start time of the pump operation;
2. **work_duration_minutes** - column of type float, stores the data from the optimisation process of the duration of pump activity;
3. **pump_speed** - column of type float, stores the data for the pump speed control;

The last schema developed, was the repository for valve control, which is similar in composition to the pump control, despite the fact that the valves work with discrete control, i.e, on / off control. This means that it does not need to contain a column for the valve's speed control, being therefore simpler.

1. **Datetime** - column of type datetime, being the key of the table, stores the start time of the valve operation;
2. **work_duration_minutes** - column of type float, stores the data from the optimisation process of the duration of valve activity.

4.4 Project's Graphical User Interface (GUI)

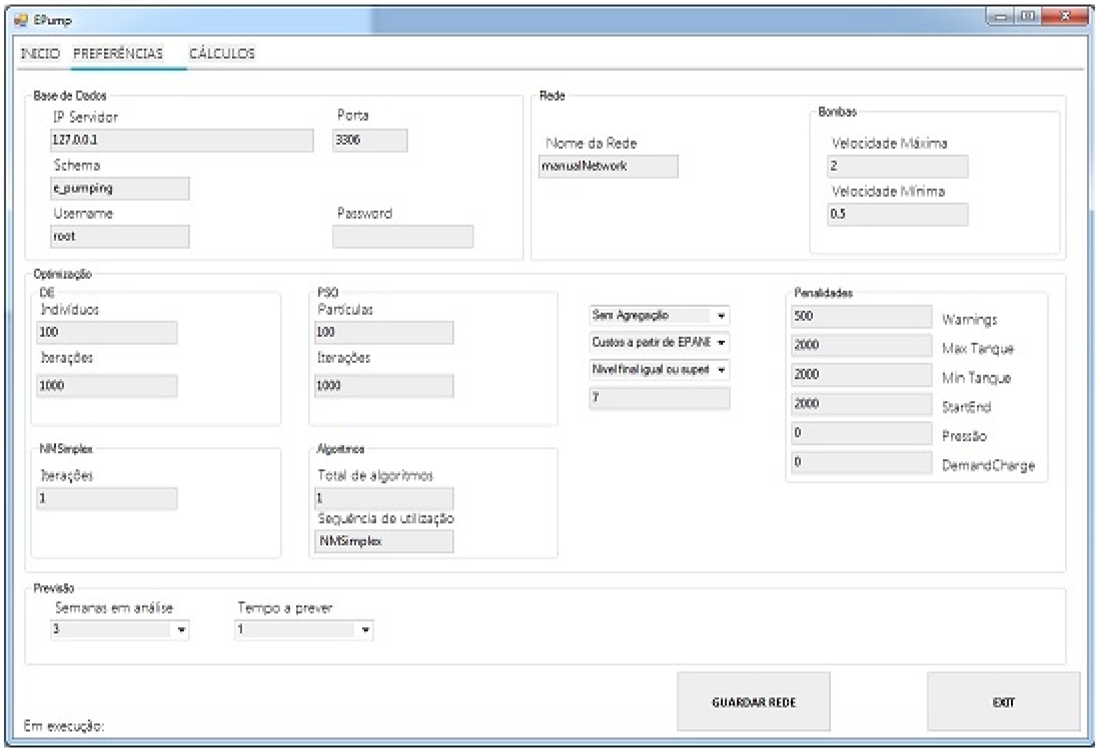
In order to provide an easy visualization and operation over all the modules within the project, an user friendly interface which integrates all of the modules, as well as all the configurations necessary, was developed by the team member Bruno Pereira in his master's thesis [67], also associated with the E-Pumping project [4].

The window displayed in figure 4.10 presents the screen associated with the configurations settings. This configuration includes the data base location, username and password, optimization algorithm's options and the forecast tool configuration of time of analysis and time of forecast.

Other tabs in this interface allow the graphical visualization of the pumps operation's schedules, exposed on figure 4.11, as well as the graphical representation of the forecasts produced, showed in figure 4.12. These features will aid in the decision making procedure, by enabling the user to select the best forecast and critically analysed the control schedule suggested by the optimization module.

This interface provides a valuable aid to the end user, who doesn't need to understand the code implemented in order to comprehend the program suggestions, a crucial design feature requested by the project manager.

To follow in the detail the use of the software, it's user's manual was added to Appendix A.


EPump

INICIO PREFERÊNCIAS CÁLCULOS

Base de Dados

IP Servidor	127.0.0.1
Porta	3306
Schema	e_pumping
Username	root
Password	

Rede

Nome da Rede	manualNetwork

Bombas

Velocidade Máxima	2
Velocidade Mínima	0.5

Optimização

DE

Indivíduos	100
Iterações	1000

PSO

Partículas	100
Iterações	1000

Sem Agregação

Custos a partir de EPANE

Nivel final igual ou super

7

Penalidades

500	Warnings
2000	Max Tanque
2000	Min Tanque
2000	StartEnd
0	Pressão
0	DemandCharge

NMSimplex

Iterações	1

Algoritmos

Total de algoritmos	1
Sequência de utilização	NMSimplex

Previsão

Semanas em análise	3
Tempo a prever	1

GUARDAR REDE EXIT

Em execução:

Figure 4.10: Configurations selection menu

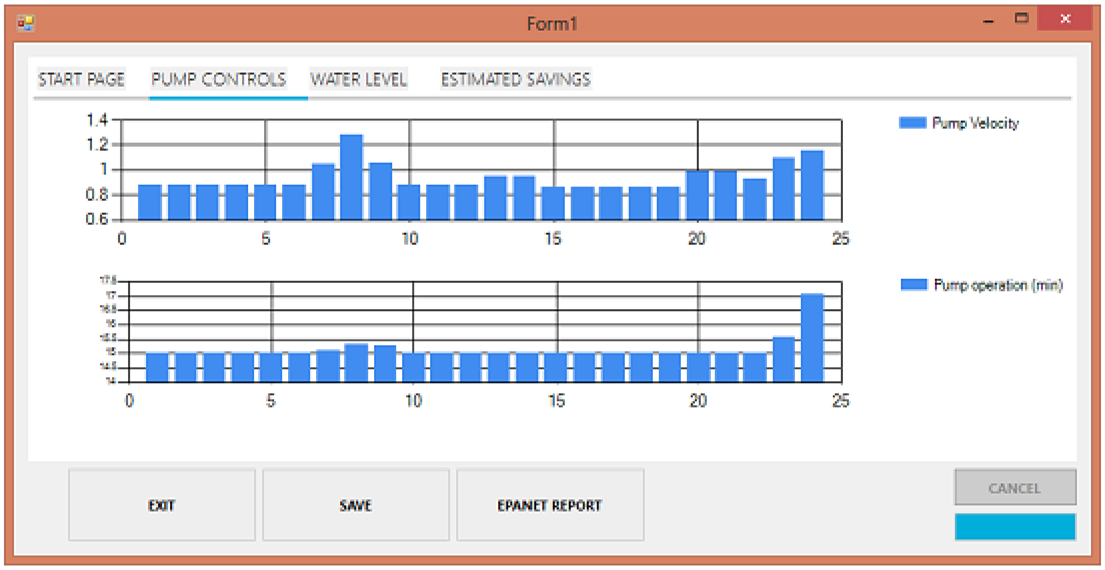
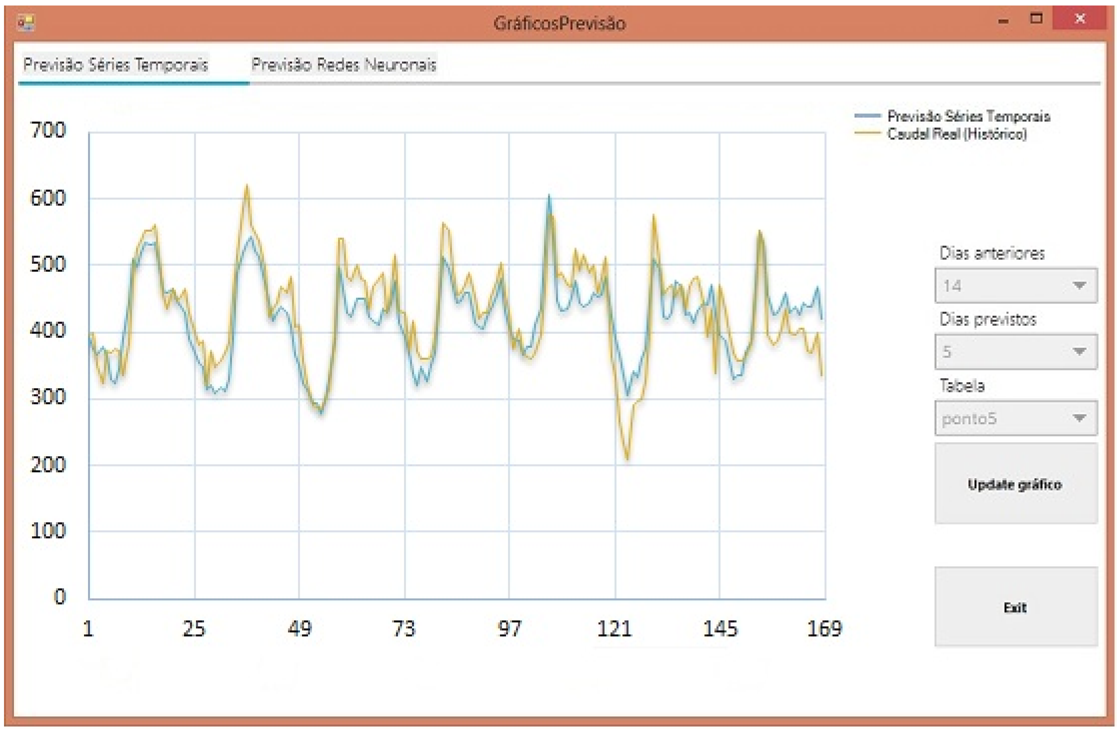


Figure 4.11: Graphical representation of the pumps operation's schedules



Dias anteriores: 14
Dias previstos: 5
Tabela: ponto5

Figure 4.12: Graphical representation of the forecast

Chapter 5

Final Remarks

5.1 Conclusions

Most Water Supply System (WSS) in Portugal still use rudimentary or empirical knowledge to manage their stations. The technology transfer project E-Pumping [4], in which this thesis inserts itself, was set to develop a management tool in order to reduce cost in the operation of those stations, not by improving energetic efficiency, but by taking advantage of the energy's timetable prices.

Within the project, one of this thesis objectives was to analyse the state-of-the-art developments up to date, following recent breakthroughs in this field of studies. Another goal was to develop a water demand forecasting tool based on statistical models, more specifically using time series methods. One last main objective was to develop a local network communication solution with a backup communication system based on Global System for Mobile Communication (GSM), in order to guarantee 100% uptime, that would interface with an existent SCADA system. To complement this solution the development of a web tool for remote monitoring was also targeted. The testing and validation of these tools was also set as main objective, as well as their connection to the general module, the main GUI, which would consolidate all the components of the project.

In order to accomplish these goals, the thesis was divided into different sections, being the first the communication and control solution, which was, in turn, subdivided into two different subsections: a fully automated solution based on the OPC protocol and a read-only solution based on database files sent by the company's system who volunteered as a pilot for the project. This last objective depended on the station's operators to manually control the system, by virtue of the company's requisites regarding permissions clearance.

The second section was related to a different part of the project, prior to the optimization of the system, which consisted on the development of a forecasting tool based on the time series classical multiplicative decomposition method, automated resorting a C++ routine and fed by the project's database.

Two different OPC servers were used to test the supervision and control tool developed, the KEPwareEX [68] and the RSLinx by Rockwell [44]. The first was an universal tool, compatible with most industrial automation systems. The second a proprietary system designed to connect to the Allen-Bradley's equipment present in the facility and whose system was used as a pilot installation. Both confirmed the flexibility of the tool developed over C# environment during the trial period.

Regarding the forecasting tool, several time series methods were tested, reaching the conclusion that, for this specific type of data, water demand, a less sophisticated solution, specifically the time series classical multiplicative decomposition method, offered a more reliable output than the ARIMA model often used in similar situations. This conclusion was based on the fact that, although the forecast produced by the multiplicative method had a higher Mean Absolute Percentage Error (MAPE), it had only a 1% of deviation from the ARIMA's forecast model, and was significantly simpler and lighter to implement. The multiplicative method also forecasted the water demand with higher accuracy in terms of spike predictions, which represented a key feature for the project scope, making it the method which best met the project's requirement of fast response under real time usage.

The final tool developed was able to compute a model, given a database with the specifications from chapter 4.3, automatically taking into account new data inserted into the database at each day, every time it is executed. The parameters recalculation is a computational light procedure, taking a short amount of time, so as to not impact the normal behaviour of the plant. This allowed the tool to be autonomous, being able to auto-correct itself every day, and flexible, having the capacity of being installed at any water plant.

The aforementioned goal of producing a fully functional software was successfully achieved. Through this thesis two modules of the project were developed and validated, along with their connection to the main Graphical User Interface (GUI) for software handling, concluding their integration with the final project design. However, the backup tool based on GSM communication, to improve the data gathering reliability, was left unattended due to lack of interest from the project's sponsor. Also, the remote monitoring web tool was only partly implemented due to several project changes, which impacted considerably the project's time line.

The full automation of the project was also left incomplete due to fact that the software was still on a trial stage. To reach this long-term objective, it still needs to go through stress testing, as well as withstand a long-term pilot installation, to clear out any problems.

This thesis developments assisted in the creation of an application that will contribute significantly for the country's economic panorama by enabling water distribution companies to reduce their running costs and, consequently, improve the market competition, which ultimately will reduce the water bill to the end costumer.

5.2 Future work

Despite the achievements of this thesis, there are some extra points which would add value to this work and to the project. One of them would be the development of a GSM accessible OPC client, resorting to low power and low cost equipments and based on Microchip's technology [69]. This tool would enable to establish wireless communications with spread out stations without wired infrastructures, as well as reliability improvements on stations that are already connected via local network.

The water demand forecasting tool could also be improved by adding state-of-the-art algorithms such as Artificial Neural Networks (ANNs) and Genetic Algorithms based forecasts. This would improve the tool's accuracy, given the availability of an extensive history of water demand and weather data.

Further testing and debugging of the software was left to a later stage. These should be attended prior to the full automation of the software.

Moreover, the development of a portable application for supervision of the running environment would complement the main GUI, by enabling the plant manager to individually attend to several ramifications of his system from anywhere at any time.

Bibliography

[1] A. I. Wilson, "Hydraulic Engineering and Water Supply," *The Oxford Handbook of Engineering and Technology in the Classical World*, 2009.

[2] EPAL, "Relatório de sustentabilidade 2010."

[3] S. M. Bunn and L. Reynolds, "The energy-efficiency benefits of pump-scheduling optimization for potable water supplies," *IBM Journal of Research and Development*, vol. 53, pp. 5:1–5:13, may 2009.

[4] Telesensor, "E-pumping." http://www.telesensor.com/products/default.html. Accessed: 04/06/2014.

[5] O. Foundation, "Opc foundation." https://opcfoundation.org/. Accessed: 17/04/2013.

[6] T. M. Walski, D. V. Chase, D. A. Savic, 1960, and I. Haestad Methods, *Water distribution modeling.* Waterbury, CT :: Haestad Press, 1st ed., reprinted with corrections june, 2001. ed., c2001. Web page: `www.haestadmethods.com`.

[7] Martins *et al.*, "Modelagem computacional como ferramenta para estudos de eficiência energética no saneamento," *VI SEREA - Seminário Iberoamericano sobre Sistemas de Abastecimento Urbano de Água.*

[8] B. M. Kiselychnyk, O. and H. Werner, "Overview of energy efficient control solutions for water supply systems," *Andrii Mykhaylo Ostrogradskiy*, pp. 40 – 46, 2009.

[9] V. R. Segovia and A. Theorin, "History of Control History of PLC and DCS," vol. 15, 2013.

[10] U. Enste, "OPC Unified Architecture," vol. 59, pp. 397–404, July 2011.

[11] Macecontractors, "Macecontractors." http://www.macecontractors.com/page/aqualia-mace-awarded-al-ain-and-eastern-region-contract. Accessed: 04/06/2014.

[12] S. Zhou, T. McMahon, a. Walton, and J. Lewis, "Forecasting daily urban water demand: a case study of Melbourne," *Journal of Hydrology*, vol. 236, pp. 153–164, Sept. 2000.

[13] A. Jain, A. K. Varshney, and U. C. Joshi, "Short-Term Water Demand Forecast Modelling at IIT Kanpur Using Artificial Neural Networks," pp. 299–321, 2002.

[14] J. S. Cedo Maksimovic, Francesco Calomino, *Water Supply Systems.* No. vol. 15, Springer Berlin Heidelberg, 1996.

[15] M. K. Douglas C. Montgomery, Cheryl L. Jennings, *Introdution to Time Seires Analysis and Forecasting.* Wiley Inter-Science, 2008.

[16] J. Bougadis, K. Adamowski, and R. Diduch, "Short-term municipal water demand forecasting," *Hydrological Processes*, vol. 19, pp. 137–148, Jan. 2005.

[17] P. Nastos, A. Paliatsos, K. Koukouletsos, I. Larissi, and K. Moustris, "Artificial neural networks modeling for forecasting the maximum daily total precipitation at athens, greece," *Atmospheric Research*, vol. 144, no. 0, pp. 141 – 150, 2014. Perspectives of Precipitation Science - Part {II}.

[18] J. Caiado, "Forecasting water consumption in Spain using univariate time series models," MPRA Paper 6610, University Library of Munich, Germany, Sept. 2007.

[19] J. F. F. da Silva, "Modelação e previsão de utilizações de Água usando uma ferramenta de optimização para a estimação de parâmetros," *Universidade do Minho, Departamento de Engenharia Civil*, no. 33, 2008.

[20] R. Abdel-Aal, "Univariate modeling and forecasting of monthly energy demand time series using abductive and neural networks," *Computers & Industrial Engineering*, vol. 54, pp. 903–917, May 2008.

[21] J. L. Torres, M. D. Blas, A. D. Francisco, and A. Garc, "Forecast of hourly average wind speed with ARMA models in Navarre (Spain)," vol. 79, pp. 65–77, 2005.

[22] Y. Dazhi, P. Jirutitijaroen, and W. M. Walsh, "Hourly solar irradiance time series forecasting using cloud cover index," *Solar Energy*, vol. 86, no. 12, pp. 3531–3543, 2012.

[23] U. EPA, "Epanet project." `http://www.epa.gov/nrmrl/wswrd/dw/epanet.html`. Accessed: 26/03/2013.

[24] 7-technologies, "7-technologies." http://www2.schneider-electric.com/sites/corporate /en/products-services/former-brands/7-technologies/7t-is-now-se.page. Accessed: 26/04/2013.

[25] Derceto, "Aquadapt." http://www.derceto.com/Products-Services/Derceto-Aquadapt /About-Derceto-Aquadapte. Accessed: 06/05/2013.

[26] C. W. Gellings, "Program on technology innovation: Electric efficiency through water supply technologies - a roadmap," *Electric Power Research Institute*, June 2009.

[27] B. Coelho, "Energy resourses optimization in water supply networks," 2011.

[28] K. E. Atkinson and J. Wiley, *An Introduction To Numerical Analysis Second Edition Kendall E. Atkinson.*

[29] J. M. E. Carlos León, Sergio Martín and J. Luque, "Explore: Hybrid Expert System for Water Networks Management," 2009.

[30] X. Zhuan and X. Xia, "Optimal operation scheduling of a pumping station with multiple pumps," *Applied Energy*, vol. 104, no. 0, pp. 250 – 257, 2013.

[31] Schneider-Electric, “Aquis.” http://www.schneider-electric.com/products/ww/en/5100-software/5125-information-management/61417-aquis-software/. Accessed: 04/05/2013.

[32] Bentley, “Watercad.” http://www.bentley.com/en-US/Products/WaterCAD/. Accessed: 04/05/2013.

[33] Bentley, “Watergems.” http://www.bentley.com/en-US/Products/WaterGEMS/. Accessed: 04/05/2013.

[34] Bentley, “Darwin-scheduler.” http://www.bentley.com/en-US/Products/WaterGEMS/Darwin-Scheduler.htm. Accessed: 04/05/2013.

[35] Bentley, “Pipe renewal planner.” http://www.bentley.com/en-US/Products/ WaterGEMS/Pipe+Renewal+Planner.htm.htm. Accessed: 04/05/2013.

[36] Bentley, “Darwin calibrator module.” http://www.bentley.com/en-US/Products/WaterGEMS/Darwin-Calibrator.htm. Accessed: 04/05/2013.

[37] Bentley, “Scadaconnect.” http://www.bentley.com/en-US/Products/WaterGEMS/SCADAConnect-Overview.htm. Accessed: 04/05/2013.

[38] Z. Chen, S. Grasby, K. Osadetz, and P. Fesko, “Historical climate and stream flow trends and future water demand analysis in the Calgary region, Canada,” *Water Science & Technology*, vol. 53, p. 1, jun 2006.

[39] Oracle, “Mysql.” http://www.mysql.com. Accessed: 20/05/2013.

[40] L. A. Rossman, *Epanet 2 users manual.* EPA-Environmental Protection Agency, September 2000.

[41] S. Karnouskos, D. Savio, P. Spiess, D. Guinard, V. Trifa, and O. Baecker, “Real-world Service Interaction with Enterprise Systems in Dynamic Manufacturing Environments,”

[42] M. OPC, “Matrikon opc.” http://www.matrikonopc.com/. Accessed: 20/04/2013.

[43] Movicon, “Movicon.” http://www.movicon.com.au/. Accessed: 04/06/2013.

[44] R. Automation, “Rslinx.” https://www.rockwellautomation.com/rockwellsoftware/design/rslinx/overview.page. Accessed: 20/04/2013.

[45] Microsoft, “System.data.oledb namespace.” http://msdn.microsoft.com/en-us/ library/system.data.oledb.aspx. Accessed: 20/08/2013.

[46] Microsoft, “System.data.odbc namespace.” http://msdn.microsoft.com/en-us/ library/system.data.odbc.aspx. Accessed: 20/08/2013.

[47] S. Razbash, D. Schmidt, Z. Zhou, Y. Khan, C. Bergmeir, E. Wang, and M. R. J. H. Robhyndmanmonashedu, “Package forecast,” 2014.

[48] H. C. Harrison, “An intelligent business forecasting system,” pp. 229–236.

[49] NCSS, “Decomposition forecasting.” http://www.ncss.com/wp-content/themes/ ncss/pdf/Procedures/NCSS/Decomposition_Forecasting.pdf. Accessed: 17/02/2013.

[50] M. BALCILAR, “Classical decompositon,” pp. 1–6, 2007.

[51] CRAN, “The r project.” http://cran.r-project.org/. Accessed: 17/03/2013.

[52] R. J. Hyndman, “Cran task view: Time series analysis.” http://cran.r-project.org/web/views/TimeSeries.html. Accessed: 17/03/2013.

[53] J. E. M. I. T. Robert B. Cleveland, William S. Cleveland, “Stl: A seasonal-trend decomposition procesure based on loess,”

[54] R. J. Hyndman, “Cran task view: Time series analysis.” http://stat.ethz.ch/R-manual/R-patched/library/stats/html/stl.html. Accessed: 17/03/2013.

[55] R. J. Hyndman, “Forecasting: principles and practice.” https:// www.otexts.org/fpp/8/1. Accessed: 17/05/2013.

[56] R. F. Nau, “Identifying the numbers of ar or ma terms.” http:// people.duke.edu/ rnau/411arim3.htm. Accessed: 20/04/2013.

[57] R. J. Hyndman, “Automatic Time Series Forecasting : The forecast Package for R,” vol. 27, no. 3, 2008.

[58] R. Automation, “Rslogix 500.” https://www.rockwellautomation.com/rockwellsoftwar e/design/rslogix500/overview.page. Accessed: 20/04/2013.

[59] R. Automation, “Rslogix emulate.” http://www.rockwellautomation.com/rockwellsoft ware/design/rslogixemulate/overview.page. Accessed: 20/04/2013.

[60] MySQL, “Mysql native driver for php.” http://dev.mysql.com/downloads/connector/p hp-mysqlnd/. Accessed: 20/04/2013.

[61] R. C. Griffin and C. Chang, “Seasonality in Community Water Demand,” vol. 16, no. 2, pp. 207–217, 1991.

[62] R. J. Hyndman, “Ets: Exponential smoothing state space model.” http://www.inside-r.org/packages/cran/forecast/docs/ets. Accessed: 17/04/2013.

[63] H. M. Abe Kosei, J. Perraud, “R.net.” http://rdotnet.codeplex.com/. Accessed: 20/04/2013.

[64] postgresql, “postgresql.” http://www.postgresql.org/. Accessed: 20/01/2013.

[65] Microsoft, “Microsoft access.” http://office.microsoft.com/en-us/access/. Accessed: 20/01/2013.

[66] sqlite, “sqlite.” http://www.sqlite.org/. Accessed: 20/01/2013.

[67] B. Pereira, “Análise e optimização de sistemas de abastecimento de água.” http://hdl.handle.net/10773/11977, 2013.

[68] KEPware, “Kepwareex.” http://www.kepware.com/. Accessed: 20/01/2013.

[69] Microchip, “Microchip.” http://www.microchip.com/. Accessed: 02/06/2014.

Appendix A

E-Pumping User's Manual

E-Pumping

USER´S MANUAL

universidade de aveiro

The general purpose of this manual, within the project "Modelação e desenvolvimento de ferramentas numéricas para implementação em sistemas de abastecimento de água", is the demonstration of the capabilities of the developed software and to help the Telesensor corporation to use and continue the development of the solutions presented.
This document is made specifically to the project and, therefore, cannot be spread out or used by elements external to the project without written authorization of its authors.

António Andrade-Campos
Bernardete Coelho
Bruno Pereira
Daniel Correia
José Santos
José Vagos

Department of Mechanical Engineering
University of Aveiro
Aveiro, Portugal

December 2013

Contents

1 — INTRODUCTION

1.1 What is E-Pumping

E-Pumping is a computer programme that analyses water supply systems (WSSs) and provides better controls for both pumps and valves in order to minimise energy costs associated to the operation of the networks. This programme contains three main modules:

- Optimisation module, used to test distinct solutions and, consequently, through optimization methodologies find the best operational control for the network.
- Hydraulic modelling module, used to analyse the network operation with distinct solutions provided by the optimisation module and compute the costs associated to such operation controls.
- Demand prediction module, used to predict water consumptions. These predictions will be used in the process of optimisation of the network operational controls.

The link between all modules when running E-Pumping is represented in figure 1.1

The characteristics of the network, provided by the E-Pumping user, are processed by the programme in order to determine the optimal control operations of such network. Such information must be provided in a specific text file format which can be automatically generated after designing the structure of the network and introducing, element by element, all necessary data for the hydraulic modelling. After a first analysis of the modelled network, to determine the initial energy costs, an iterative cycle between the hydraulic modelling module and the optimisation module begins until find the best controls for the minimisation of the network energy costs.

As the scheme of figure 1.1 shows, E-Pumping is provided of a database capable of receiving, saving and processing real-time data from and within the network to be optimised. Some data, such as historical and real-time water consumptions, are used by the prediction module in order to allow modelling the future behaviour of the networks.

E-Pumping is designed to be a multi-purpose tool, allowing the user to work with any desired section of a water system. The user can benefit from the set of E-Pumping modules simultaneously in order to search for the most cost-effective operational conditions or, alternatively, the E-Pumping user can also make use of the hydraulic modelling module separately to check, for instance, the network behaviour under specific conditions (such as new water consumption provided by the demand prediction module). This optimisation tool is also designed to be used in parallel with real-time systems (such as SCADA systems), allowing optimal operational

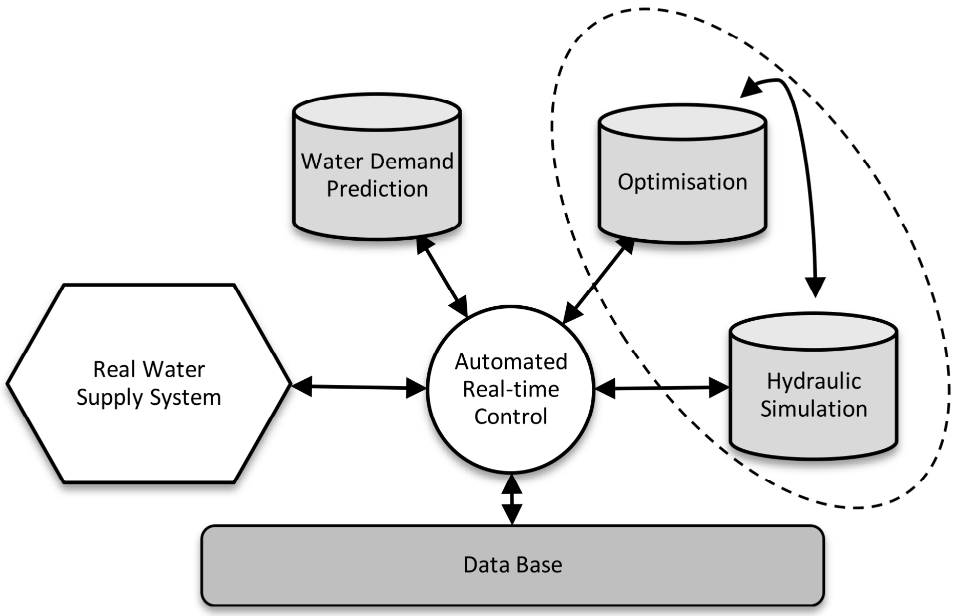


Figure 1.1: Scheme representing the connection between E-Pumping modules.

adjustments to possible variations in the networks.

A Database (DB) is also included in E-Pumping software for storing both meteorological and water consumption data and also for saving the networks characteristics and controls settings.

1.2 Hydraulic modelling capabilities

E-Pumping hydraulic modelling module is based on EPANET 2.0 (Rossman, 2000). Thus, all EPANET hydraulic modelling capabilities are also available in E-Pumping software. Such capabilities includes (Rossman, 2000):

- Computation of friction headloss using the Hazen-Williams, Darcy-Weishback or Chezy-Manning formulas;
- Computation of minor headlosses;
- Modelling of both constant- and variable-speed pumps, allowing to compute pumping energy and costs;
- Modelling of several types of valves including shutoff, check, pressure regulating and flow control valves;
- Modelling of storage tanks with distinct shapes;
- Considering multiple demand categories with distinct time patterns at nodes;
- Modelling network operations based on tank levels or time controls;
- Etc.

The water networks design is made by accessing EPANET graphical user interface (GUI) and the hydraulic modelling results can also be accessed by means of the same GUI.

This module presents no limitations on the networks sizes for modelling and future optimisation when desired.

1.3 Demand prediction capabilities

The prediction module of E-Pumping conjugates the advantages of time-series forecasting methods and also artificial intelligence methods using artificial neural networks.

This module evaluates historical and real-time data, both meteorological and water demand data, in order to predict future water consumptions. Such predicted consumptions are sent to the hydraulic modelling and optimisation modules in order to generate the most adequate future controls for the minimum operational energy costs of pumps and valves.

1.4 Optimisation capabilities

The optimisation module of E-Pumping provides three distinct optimisation algorithms which can be used sequentially and can be called several times according to the user requirements. This capability of easily apply distinct optimisation strategies contributes to a better adaptation of the module to distinct water supply systems.

E-Pumping optimisation module is able to receive the initial operational conditions (pumps speeds and operating times of both pumps and valves) and then, after multiple iterations, to provide the best solution for the operation of the tested network, minimising the associated energy costs.

The main optimisation capabilities of E-Pumping are:

- Optimisation using distinct algorithms in sequential or individual strategies;
- Optimisation strategies can be decided by the user (otherwise default options are used);
- Possible to optimise (i) variable-speed pumps (by changing speeds and operating times), (ii) constant-speed pumps (by changing operating times) and (iii) valves (by changing operating times also);
- Possible to select the elements to be optimised;
- Etc.

1.5 About this manual

E-Pumping user's manual is comprised of five main chapters:

Chapter 1 introduces E-Pumping description summarising its main capabilities in terms of hydraulic simulation, optimisation and demand prediction.

Chapter 2 provides a tutorial on how to install the software and how to quickly start using it. Readers unfamiliar with the basics of water consumption prediction or optimisation might wish to review Chapter 4 and 5 before working through the tutorial. The use of the hydraulic modelling module requires some knowledge by the user on how to use EPANET 2.0. Therefore, the user might also wish to review first EPANET user manual (Rossman, 2000).

Chapter 3 explains the organisation of E-Pumping workspace, describing the functions of the various menu options and buttons.

Chapter 4 describes how to use the water demand prediction module and defines the forecasting methods included in E-Pumping prediction module.

Chapter 5 explains how to use the optimisation module and describes each optimisation algorithm such as how to combine them in order to reach better results.

2 — QUICK START TUTORIAL

2.1 Installing E-Pumping

E-Pumping is designed to run under Windows 8/7/Vista operating systems. It is distributed as a single file, EPumping.exe, which contains a self-extracting programme. However, E-Pumping software requires a previous installation of Microsoft Visual C++ 2010 Redistributable Package. Also the installation of EPANET 2.0 is recommended.
To run E-Pumping:

1. Double click EPumping.exe and select the directory for the files extraction. A folder named EPumping is automatically generated in the chosen directory;
2. If the machine intended to run EPumping does not have a Visual Studio installation, download and install the Microsoft Visual C++ 2010 Redistributable Package, freely available at `http://www.microsoft.com/en-us/download/details.aspx?id=5555`.

To remove E-Pumping from the computer, the user simply needs to delete the folder where the installation was extracted.

2.2 Example network

In this tutorial, a simple network is used to quickly demonstrate how to use E-Pumping in order to minimise the costs associated to the operation of such network. The presented network (see Figure 2.1) contains one example of each element possible to control by the optimisation module: (i) a variable-speed pump (pump1), (ii) a constant-speed pump (pump2) and (iii) a valve.

2.2.1 Drawing the network and modelling

In order to start drawing the network for modelling, the user must open EPANET 2.0. After the EPANET window opens, the user must draw the structure of the network as represented on Figure 2.1 and then insert, for each element, the main characteristics described on Tables 2.1, 2.2 and 2.3 (the user may wish to read first the quick start tutorial of EPANET 2.0 of chapter 2 of EPANET User's Manual (Rossman, 2000)).

The network of this tutorial is constituted of (i) one water source, represented by a reservoir with 210 m of head; (ii) one tank for the water storage, with an elevation of 250 m, a diameter of 20 m and a maximum and minimum levels of 5 and 1 m respectively; (iii) three nodes of water consumption with associated 24-hour water demand patterns; (iv) one node with water entering

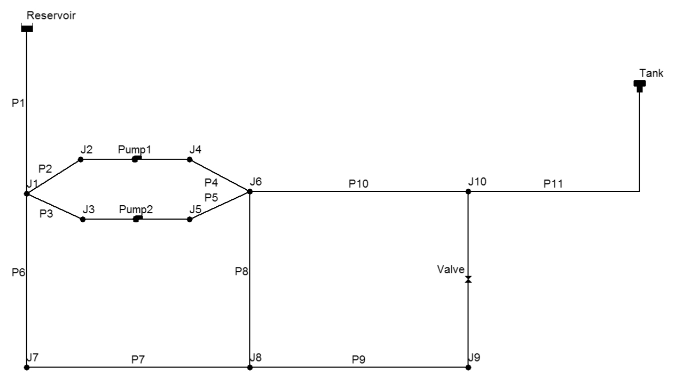


Figure 2.1: Structure of the example of network used in the tutorial.

Table 2.1: Characteristics of all nodes that constitutes the example of network used in the tutorial.

Node	Elevation (m)	Base demand (CMH)	Demand pattern
J1	0	0	-
J7	200	-10	Pat1
J6	0	0	-
J8	200	50	Pat2
J10	215	30	Pat3
J9	150	60	Pat3
J2	0	0	-
J3	0	0	-
J4	0	0	-
J5	0	0	-
Tank	250	n.a.	n.a.
Reservoir	210	n.a.	n.a.

n.a. = not applicable

the network (negative base demand) also with an associated pattern; (v) a variable-speed pump with a 24-hour speed pattern associated; (vi) a fixed-speed pump; and (vii) a throttle control valve with a 100 mm diameter and a minor head loss coefficient of 10. The pumps are similar, presenting the same pump characteristic curve represented by a single point given by 45 m of head for a flow of 130 m^3/h. A pattern for the energy price during the 24 hours and an efficiency curve (see values for the curve in Table 2.4) are also associated to each pump in order to allow the computation of the operational costs.

In order to proceed with an extended period (24 hours, in this case) simulation of the network, the total duration (see **Data » Options » Times**) must be set to 24:00. The hydraulic, pattern and reporting time-steps should be set to 1:00. Respecting to the hydraulics options (see **Data » Options » Hydraulics**), the user must choose the Hazen-Williams headloss formula (H-W), use cubic meter per hour for flow units (CMH) and set the option 'Continue' if unbalanced. For more details on the network and other modelling options, the user may wish to consult the complete Input file available in Appendix A.

After introducing all the network characteristics and setting the simulation options, it is important to define the hourly controls (hourly due to the 1-hour time-steps, in this case) for both pumps and valve. Such controls, provided in Table 2.5, should be entered in the **Simple Controls Editor** section of EPANET. By default, EPANET sets pumps and valves open at the beginning of each time-step considering their normal settings (status, in case of valves or fixed-speed pumps

Table 2.2: Characteristics of all links that constitutes the example of network used in the tutorial.

Link	Length (m)	Diameter (mm)	Roughness
P1	1	150	100
P6	1500	150	100
P7	1000	150	100
P8	1500	150	100
P10	1500	200	100
P11	1000	120	100
P9	1500	100	100
P2	1	150	100
P4	1	150	100
P3	1	150	100
P5	1	150	100

and speed, in case of variable-speed pumps), so the "OPEN" controls provided in table 2.5 are not needed for the network modelling. However, as explained in the next section, such control statements are essential for the optimisation process.

Finally, the user can run the network model for the 24-hour period (**Project » Run Analysis**) and check if no warnings occur. If warning messages appear, it means that some information should not have been correctly entered. In this case, the user may wish to compare the model details with all the data contained in file available in Appendix A.

In order to view the results of the hydraulic simulation for the selected period, the user can export a Report file by selecting **Report » Full** and then saving the file with the name *manualNetwork.rpt*. Such report contains not only the results for the energy usage but also hourly results of flow, velocity and headloss for links and pressure for nodes (see Appendix B). Alternatively, the user may wish to view such results using other EPANET options such as tables or graphs.

2.2.2 Preparing for the optimisation module

When the EPANET model of the network is finalised and running without warning messages, it is time to prepare a file that is going to be used by the optimisation module. Such file is the input file (also referred as INP), which contains all the network model characteristics in a specific format (see Appendix A). EPANET automatically exports such file. Thus, at this stage, the user should export the model by selecting the option **File » Export » Network...** and then save the file in the folder "OPT" of E-Pumping main folder with the name *manualNetwork.inp*.

Before starting to optimise the modelled network, the user must open the exported input file (*manualNetwork.inp*) using a text editor (such as Notepad) and insert the following options (in case they are missing):

1. In section [REPORT]:

   Status Full
   Energy Yes
   Nodes All
   Links All

2. In section [OPTIONS]:

   Unbalanced Continue
   Hydraulics Save HydFileName

After inserting the previous options and saving such changes, the input file is ready to be used by the E-Pumping optimisation module.

Table 2.3: Temporal patterns for the (i) pump speed, (ii) energy price and (iii) water demand used in the tutorial network.

Time-step	**Pump1 speed**	**Energy price**	**Pat1**	**Pat2**	**Pat3**
0-1h	1	0.007	0.5	0	0.3
1-2h	1	0.007	0.5	0	0.3
2-3h	1	0.007	0.5	0	0.3
3-4h	1	0.007	1	0	0.3
4-5h	1	0.007	1	0	0.3
5-6h	1	0.007	0.5	0	0.3
6-7h	1	0.007	0.5	0	1
7-8h	1	0.007	0.5	1	1
8-9h	1	0.01	0.5	1	0.6
9-10h	1	0.01	0.5	1	0.6
10-11h	1	0.01	0.5	1	0.6
11-12h	1	0.01	1	1	0.8
12-13h	1	0.01	1	1	0.8
13-14h	1	0.01	0.5	1	0.8
14-15h	1	0.01	0.5	1	0.6
15-16h	1	0.01	0.5	1	0.6
16-17h	1	0.01	0.5	1	0.6
17-18h	1	0.01	1	1	0.6
18-19h	1	0.01	1	0.5	0.8
19-20h	1	0.01	1	0	1
20-21h	1	0.01	1.2	0	1
21-22h	1	0.01	1	0	0.8
22-23h	1	0.007	0.5	0	0.6
23-24h	1	0.007	0.5	0	0.3

Table 2.4: Points for the efficiency curve of each pump of the tutorial network.

Q (m^3/h)	η **(%)**
50	60
70	68
100	73
130	75
160	74
200	70
240	64

2.3 Running network optimisation

2.3.1 Setting optimisation options

Before starting to run the optimisation module, some optimisation options must be set. Therefore, the following steps should be performed by the user:

1. Open E-Pumping, select the *PREFERÊNCIAS* Menu and, in the *Rede* box, introduce the name of the network file (*manualNetwork*).
2. In *Optimização* box, start with the section *Algoritmos*, setting the number of algorithms as 3 and the sequence as *DE PSO NMSimplex*. Such options lead the optimisation module to sequentially use the 3 distinct algorithms available in E-Pumping software.
3. For both DE and PSO boxes, set the number of individuals/particles as 100. For the network of this tutorial, the number of individuals/particles should never be inferior to 96 (24 steps $\times$ (2 $\times$ number of variable-speed pumps $+$ number of fixed-speed pumps $+$ number of valves)).

Table 2.5: Control statements for both pumps and valve of the tutorial network.

Pump1	**Pump2**	**Valve**
LINK Pump1 1.0 AT TIME 0.0	LINK Pump2 OPEN AT TIME 0.0	LINK Valve OPEN AT TIME 0.0
LINK Pump1 1.0 AT TIME 1.0	LINK Pump2 OPEN AT TIME 1.0	LINK Valve OPEN AT TIME 1.0
LINK Pump1 1.0 AT TIME 2.0	LINK Pump2 OPEN AT TIME 2.0	LINK Valve OPEN AT TIME 2.0
LINK Pump1 1.0 AT TIME 3.0	LINK Pump2 OPEN AT TIME 3.0	LINK Valve OPEN AT TIME 3.0
LINK Pump1 1.0 AT TIME 4.0	LINK Pump2 OPEN AT TIME 4.0	LINK Valve OPEN AT TIME 4.0
LINK Pump1 1.0 AT TIME 5.0	LINK Pump2 OPEN AT TIME 5.0	LINK Valve OPEN AT TIME 5.0
LINK Pump1 1.0 AT TIME 6.0	LINK Pump2 OPEN AT TIME 6.0	LINK Valve OPEN AT TIME 6.0
LINK Pump1 1.0 AT TIME 7.0	LINK Pump2 OPEN AT TIME 7.0	LINK Valve OPEN AT TIME 7.0
LINK Pump1 1.0 AT TIME 8.0	LINK Pump2 OPEN AT TIME 8.0	LINK Valve OPEN AT TIME 8.0
LINK Pump1 1.0 AT TIME 9.0	LINK Pump2 OPEN AT TIME 9.0	LINK Valve OPEN AT TIME 9.0
LINK Pump1 1.0 AT TIME 10.0	LINK Pump2 OPEN AT TIME 10.0	LINK Valve OPEN AT TIME 10.0
LINK Pump1 1.0 AT TIME 11.0	LINK Pump2 OPEN AT TIME 11.0	LINK Valve OPEN AT TIME 11.0
LINK Pump1 1.0 AT TIME 12.0	LINK Pump2 OPEN AT TIME 12.0	LINK Valve OPEN AT TIME 12.0
LINK Pump1 1.0 AT TIME 13.0	LINK Pump2 OPEN AT TIME 13.0	LINK Valve OPEN AT TIME 13.0
LINK Pump1 1.0 AT TIME 14.0	LINK Pump2 OPEN AT TIME 14.0	LINK Valve OPEN AT TIME 14.0
LINK Pump1 1.0 AT TIME 15.0	LINK Pump2 OPEN AT TIME 15.0	LINK Valve OPEN AT TIME 15.0
LINK Pump1 1.0 AT TIME 16.0	LINK Pump2 OPEN AT TIME 16.0	LINK Valve OPEN AT TIME 16.0
LINK Pump1 1.0 AT TIME 17.0	LINK Pump2 OPEN AT TIME 17.0	LINK Valve OPEN AT TIME 17.0
LINK Pump1 1.0 AT TIME 18.0	LINK Pump2 OPEN AT TIME 180.0	LINK Valve OPEN AT TIME 18.0
LINK Pump1 1.0 AT TIME 19.0	LINK Pump2 OPEN AT TIME 19.0	LINK Valve OPEN AT TIME 19.0
LINK Pump1 1.0 AT TIME 20.0	LINK Pump2 OPEN AT TIME 20.0	LINK Valve OPEN AT TIME 20.0
LINK Pump1 1.0 AT TIME 21.0	LINK Pump2 OPEN AT TIME 21.0	LINK Valve OPEN AT TIME 21.0
LINK Pump1 0.0 AT TIME 22.0	LINK Pump2 CLOSED AT TIME 22.0	LINK Valve CLOSED AT TIME 22.0
LINK Pump1 0.0 AT TIME 23.0	LINK Pump2 CLOSED AT TIME 23.0	LINK Valve OPEN AT TIME 23.0

4. For the number of iterations of DE and PSO, 300 is enough to obtain good results for the network of the tutorial. The iterations number of NMSimplex algorithm can be set larger, such as 5000. In this case, NMSimplex is able to stop when the best value is found.
5. In the *Penalidades* box, set the Warnings penalty coefficient as 500. Such value must be much larger than the initial energy cost for the network operation (in this case, the initial cost is 13.30).

After setting the enumerated options, the user must click the button "*GUARDAR REDE*" and then select which pumps and valves of the modelled network are going to be optimised.

Finally, the user must go to *CÁLCULOS* Menu and click the button "*Optimização*" in order to start the network optimisation.

2.3.2 Viewing results

When the network optimisation is completed, a report file containing the simulation results of the optimised network is automatically generated. The user can access such file by clicking the "*Abrir Report*" button in the *Resultados* box. Alternatively, the user can open the input file of the optimised network by clicking the button "*Abrir Rede Optimizada*" and, after running the network, view all the results using EPANET tools. In case the user only wants to view the controls results for pumps and valves, those can be easily accessed by clicking the button "*Gráficos Optimização*" in the *Resultados* box.

In the case of using a database to save all the information concerned all modules, it is possible to see graphically the results obtained by the optimisation module, including the variable speed of the pumps or/and the valves.

2.4 Working with predicted data

E-Pumping allows to automatically transfer predicted water consumptions from the prediction module to the input file of the network to be optimised. Firstly, an appropriate database for the network being analysed must be developed (see Section 2.5 on how to develop a database).

When a database is already available, before starting the optimisation process, the user must go to *PREFERÊNCIAS* Menu in order to define the prediction options. After setting the optimisation options and the Database location in the *Base de Dados* box, the user must set the number of weeks to use in the prediction module as well as the period of time to be predicted in the *Previsão* box. Then the user must click the button "*GUARDAR REDE*" and go to *CÁLCULOS* Menu. In the *Previsão de Consumos* box of the *CÁLCULOS* Menu, the user must choose the prediction method and click in the correspondent button ("*Previsão Séries Temporais*" or "*Previsão Redes Neuronais*").
If the user executes both predictions, it is possible to compare both by pressing the button "*Gráficos Previsão*" on the *Resultados* box of the *CÁLCULOS* Menu, which opens a new window with graphical information of the predictions. In such window, the user can view the predictions by both methods, and for all the consumption points. After selecting the consumption point and time interval, the user must click the button "*Update Gráfico*".
Back to the *CÁLCULOS* Menu window, the user should choose the prediction method to use in the optimisation by selecting the desired one in the combo-box available in the *Optimização* box. The user must then click the button "*Optimização*" in the *Optimização* box to start the optimisation process. After the optimisation process has ended, the user can verify the pump and valve controls by clicking the button "*Gráficos Optimização*" in the *Resultados* box. In the new window, the user must select the pump or valve and press the button "*Update gráfico*".
The user can also, at any time, view the data content of the databases used by clicking the button "*Bases de Dados*" in the *Resultados* box.

2.5 Developing a database

In order to operate, E-Pumping needs to use a Database to store and retrieve data. For correct functioning, the database needs three different schemas. One for consumption nodes, one for pumps and one for valves.
The schema for consumption nodes can be created with any desired name. In this example, the created schema was called "e_ pumping". The total number of tables in this schema is equal to the total of consumption nodes in the input file created. Each table must have a total of nine columns, these being:

Datetime - column of type datetime, being the key of the table;
Caudal - column of type float, stores the data from the recorded flows needed for the predictions;
temp - column of type float, stores the data from the recorded and predicted ambient temperatures needed for neural predictions;
humid - column of type float, stores the data from the recorded and predicted humidity needed for neural predictions;
pluv - column of type float, stores the data from the recorded and predicted pluviosity needed for neural predictions;
prev_ets - column of type float, stores the data from the time series prediction;
prev_nn - column of type float, stores the data from the neural network prediction;
err_prev_ets - column of type float, for storing the difference between predicted flow(time series) and real flow(not yet implemented);
err_prev_nn - column of type float, for storing the difference between predicted flow(neural networks) and real flow(not yet implemented).

The schema for pumps needs to be created with a name associated with the schema from the consumption nodes. To the name of the previous schema, the name needs the addition of "_pump_control". In this example, the created schema was called "e_pumping_pump_control".

The total number of tables in this schema is equal to the total of pumps in the input file created. Each table must have a total of three columns, these being:

start_datetime - column of type datetime, being the key of the table, correspondent to the start time of the pump operation;

work_duration_minutes - column of type float, stores the data from the optimisation process of the duration of pump activity.

pump_speed - column of type float, stores the data from the optimisation process of the speed of the pump.

The schema for valves needs to be created with a name associated with the schema from the consumption nodes. To the name of the previous schema, the name needs the addition of "_valve_control". In this example, the created schema was called "e_pumping_valve_control". The total number of tables in this schema is equal to the total of valves in the input file created. Each table must have a total of two columns, these being:

start_datetime - column of type datetime, being the key of the table, correspondent to the start time of the valve operation;

work_duration_minutes - column of type float, stores the data from the optimisation process of the duration of valve activity.

3 — E-PUMPING WORKSPACE

3.1 Overview

When opening E-Pumping, the user must enter the access password (see Figure 3.1), provided together with the software delivery, in order to have access to E-Pumping workspace.

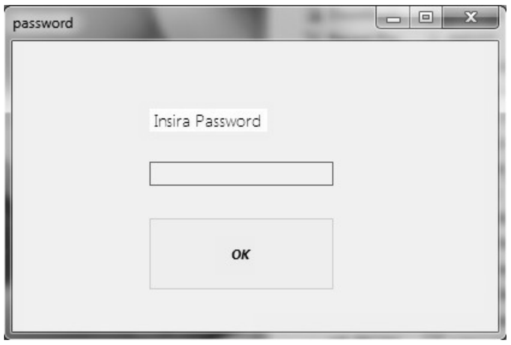


Figure 3.1: Starting window of E-Pumping.

The workspace of E-Pumping is simple, trying to present itself as an intuitive interface. E-Pumping workspace comprises (i) an initial presentation page of E-Pumping (see Figure 3.2) and (ii) two main menus: *PREFERÊNCIAS* (Preferences Menu) and *CÁLCULOS* (Calculation and Results Menu). The first one serves for the user select some options related to the network being optimised as well as some analysis options. The second one is used essentially to proceed with the optimisation process and data treatment.

Each menu from the menu bar is organised by distinct boxes. A description of both menus and its contents is provided in the following section.

3.2 Menu bar

3.2.1 Preferences Menu

- ***Base de Dados* Box**
  In this section, the user must insert the data that allow accessing the database. The IP address and the TCP/IP port of the MySQL Server must be entered, as well as the default

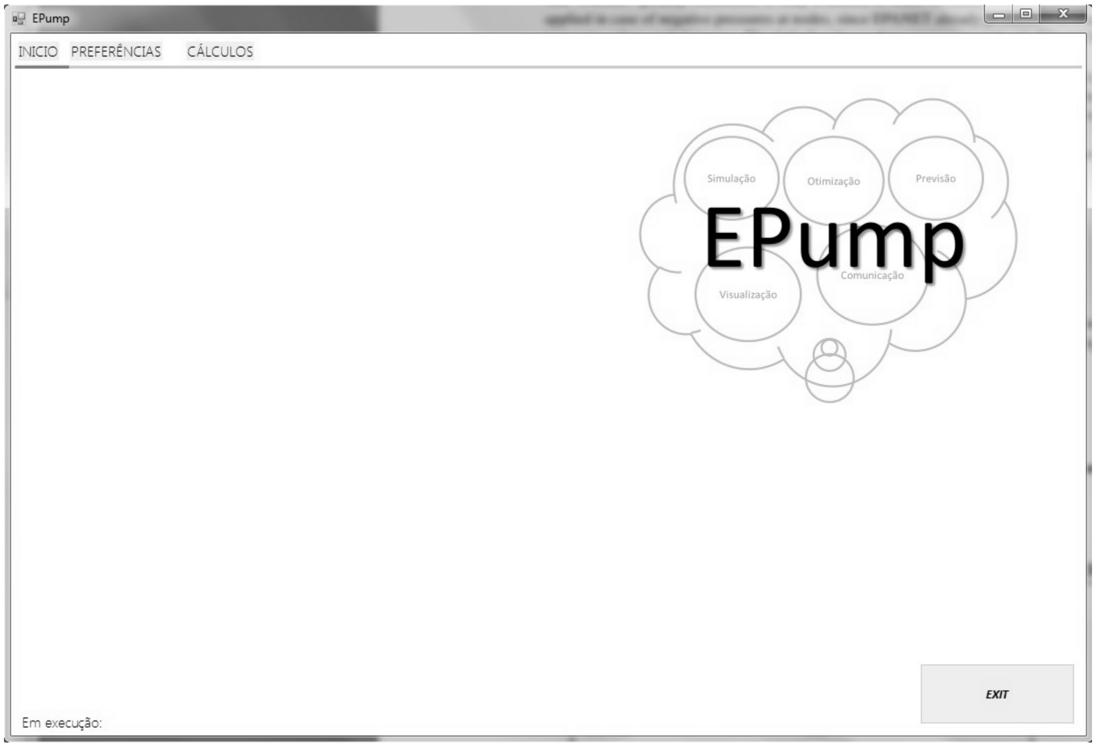


Figure 3.2: Initial page of E-Pumping.

schema to be used when the connection to server is established and finally, the user name and password to be used for the connection.

- ***Rede* Box**

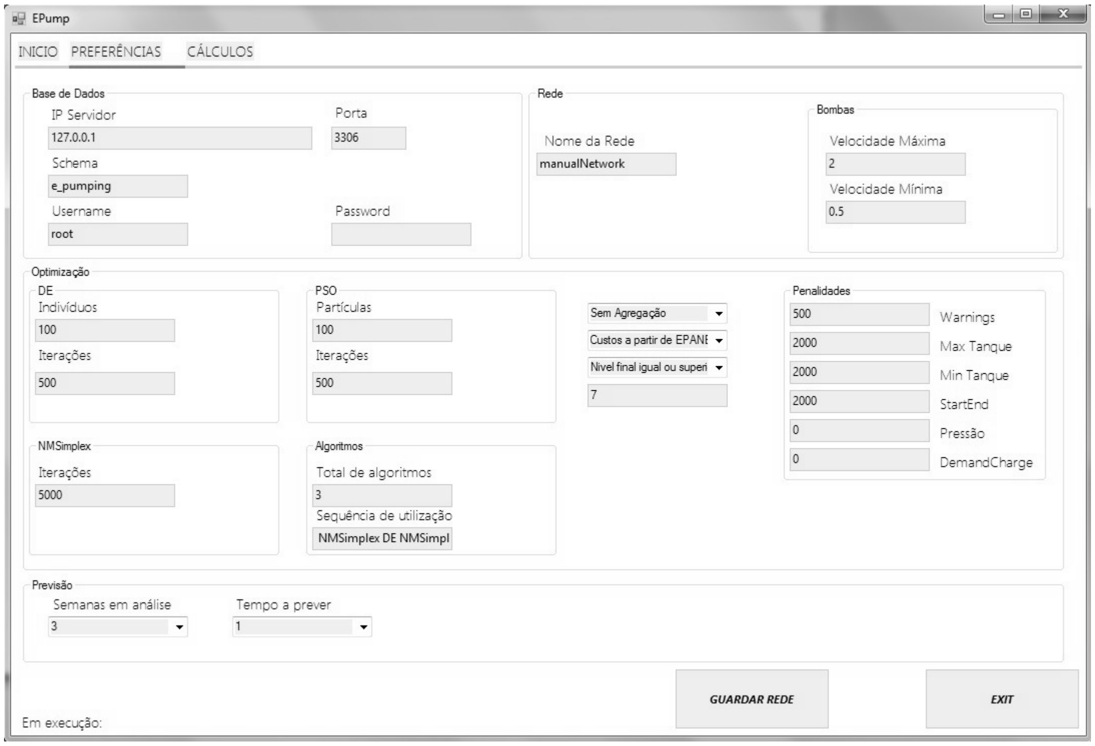
  Nome da Rede
  Place to insert the name given to the input file of the network to be analysed. Do not include the file extension (*.inp).

  Bombas
  Section to insert maximum and minimum relative speeds allowed for pumps. Such values are only used to limit the search space of the variables related to pumps speed. By default, the maximum/minimum values are set as double/half the relative speed of pumps (in relation to their nominal speed), i.e, $2.0/0.5$. Usually, pumps are not allowed to operate outside this values. However, if the user wants to, can always change such option.

- ***Optimização* Box**

  DE
  Section to insert the number of individuals and iterations to be considered by the DE algorithm. The default values are 100 individuals and 500 iterations.

  PSO
  Section to insert the number of particles and iterations to be considered by PSO algorithm. The default values are 100 particles and 500 iterations.



Base de Dados
IP Servidor: 127.0.0.1
Porta: 3306
Schema: e_pumping
Username: root
Password:

Rede
Nome da Rede: manualNetwork

Bombas
Velocidade Máxima: 2
Velocidade Mínima: 0.5

Optimização
DE
Indivíduos: 100
Iterações: 500

PSO
Partículas: 100
Iterações: 500

Sem Agregação
Custos a partir de EPANE
Nivel final igual ou superi
7

Penalidades
500 Warnings
2000 Max Tanque
2000 Min Tanque
2000 StartEnd
0 Pressão
0 DemandCharge

NMSimplex
Iterações: 5000

Algoritmos
Total de algoritmos: 3
Sequência de utilização: NMSimplex DE NMSimpl

Previsão
Semanas em análise: 3
Tempo a prever: 1

Em execução:

Figure 3.3: Preferences Menu of E-Pumping

NMSimplex
Section to insert the number of iterations to be considered by NMSimplex algorithm. The default value is 5000 iterations.

Algoritmos
Section to decide the optimisation strategy to be used by the optimisation module. The user must enter the number of sequences and define the desired sequence by introducing the abbreviations DE, PSO and/or NMSimplex separated by a space (example: to use sequentially the DE, PSO and then DE again, the number of sequences must be 3 and the desired sequence must be set as "DE PSO DE"). By default, the number of sequences is 3 and is defined by "DE PSO NMSimplex".

First combo box list
This list contains two possible options of calculation for the optimisation module: the first one, "*Com agregação*", allows the module to aggregate/group the optimisation variables as explained bellow; the second one, "*Sem agregação*", does not allow the variables aggregation during the optimisation process.
The optimisation variables aggregation consist in grouping two or more time-steps into one when conditions of consumption and energy price are maintained constant. Following the example demonstrated on Table 3.1, in the first three steps, both energy price and the consumption patterns Pat1, Pat2 and Pat3 are maintained constant, meaning that, during this period, the optimisation module can consider only one step representing the operation between 0 and 3 hours. This kind of strategy allows reducing the number of optimisation variables, reducing the processing time. The default option is set to no variables aggregation - "*Sem agregação*".

Table 3.1: Explanation on how the variables aggregation option works. The consecutive time-steps represented by blue or green colour are treated as a unique time-step by the optimisation module, reducing the number of optimisation variables. In this example, instead of 24 time-steps, the optimisation module works with 16 time-steps (11 1-hour time-steps + 5 grouped time-steps).

Time-step	Pump1 speed	Energy price	Pat1	Pat2	Pat3
0-1h	1	0.007	0.5	0	0.3
1-2h	1	0.007	0.5	0	0.3
2-3h	1	0.007	0.5	0	0.3
3-4h	1	0.007	1	0	0.3
4-5h	1	0.007	1	0	0.3
5-6h	1	0.007	0.5	0	0.3
6-7h	1	0.007	0.5	0	1
7-8h	1	0.007	0.5	1	1
8-9h	1	0.01	0.5	1	0.6
9-10h	1	0.01	0.5	1	0.6
10-11h	1	0.01	0.5	1	0.6
11-12h	1	0.01	1	1	0.8
12-13h	1	0.01	1	1	0.8
13-14h	1	0.01	0.5	1	0.8
14-15h	1	0.01	0.5	1	0.6
15-16h	1	0.01	0.5	1	0.6
16-17h	1	0.01	0.5	1	0.6
17-18h	1	0.01	1	1	0.6
18-19h	1	0.01	1	0.5	0.8
19-20h	1	0.01	1	0	1
20-21h	1	0.01	1.2	0	1
21-22h	1	0.01	1	0	0.8
22-23h	1	0.007	0.5	0	0.6
23-24h	1	0.007	0.5	0	0.3

Second combo box list

This list contains three possible options for the energy costs computation. E-Pumping is able to compute energy costs by itself ("*Custos calculados*"), using the data obtained during the hydraulic simulation of the network, which decreases the required computational time. This option of costs computation is not dependent on the hydraulics results file generated during the hydraulic simulation. E-Pumping can also use EPANET results to obtain the energy costs instead of computing them ("*Custos a partir do EPANET*"). This second option is set by default. The third option of this list, "*Ambos os custos*", consists in the costs computation by both modes (using EPANET results and computed by E-Pumping) which checks if the two calculation methods are in accordance.

When option "*Hydraulics Save HydFileName*" is not set in the input file (see Section 2.2.2 on how to prepare the INP), the costs computation option "*Custos calculados*" must be set. Otherwise, the optimisation process can result in a error after a number of iterations.

Third combo box list

This list contains options related to tanks constraints. E-Pumping allows the user to choose the desired conditions for the tanks water levels at the end of the simulation period.

The first option available in this list, "*Com restrição de continuidade*", sets the continuity constraint to all tanks, forcing the final level of each tank to be the same as the beginning of the simulation period.

The second option, "*Sem restrição de continuidade*", does not apply any tank constraint, meaning that, at the end of the simulation, the tanks water levels can be at their minimum

allowed. This is the most probable solution since it usually means less pumps operation when tanks are emptying.
The third option of the list, "*Nível final fixo*" forces a final tank level chosen by the user (the desired level, in meters, must be entered in the box below the combo box lists), for all existent tanks. This option can be useful for analysis of one-tank networks when the user needs to restrict the tank to a certain level at a specific time of the day.
The final option ("*Nível final igual ou superior ao nível inicial*"), which is set as default, applies the most common condition to this kind of problem. When this option is set, the final water level of each tank is allowed to be equal or superior to the initial level (level at the beginning of the simulation).

Penalidades
This section deals with penalty coefficients to change the penalty value applied when the specific operational conditions are violated: (i) occurrence of warning messages due to troubles in the hydraulic simulation, (ii) violation of a maximum water level allowed for a tank, (iii) violation of a minimum level allowed for a tank, (iv) violation of the constraint selected in the third combo list for the final water level of tanks, (v) violation of minimum allowed pressure (set as zero) and (vi) increase of demand charge, an additional cost due to the pumps operation at maximum power.
The "Warnings" penalty is set by default as 500. However, the dimension of such value must always be much higher than the initial operational costs of each network being analysed. For example, in case a certain network presents an initial cost in the order of hundreds, a penalty coefficient in the order of thousands is desired. Otherwise, some non-feasible control solutions for the network being optimised can be provided. Alternatively, the user can choose a larger value in order to be used in a larger number of cases (for instance 10000).
The "Max/Min Tanque" and "Start/End" penalty coefficients are set by default as 2000. If the optimised solution for the network control is not respecting the tanks constraints as desired, the user may wish to increase these coefficients and repeat the optimisation process.
The "Pressão" penalty coefficient is only available in order to reinforce the constraint applied in case of negative pressures at nodes, since EPANET already produces warning messages when such happens. The default value is then set to zero and only EPANET verification of pressure values is considered.
The "DemandCharge" penalty coefficient is set as zero by default in order to do not apply penalisation in case of additional costs due to pumps operating at maximum power. If the user wish to avoid control solutions with pumps operating at maximum power, a positive value must be set (such as 1000).
For more information on the penalties description and on how these are applied, the user may wish to read section 5.3.

- ***Previsão* Box**
  In this section, the first combo box list ("*Semanas em análise*") allows the user to select the number of weeks to be used by the prediction module for the historical data analysis; this selection is always related to the last weeks, i.e., last 3 weeks, last 6 weeks or last 12 weeks). The second combo box list ("*Tempo a prever*") corresponds to the duration of time to be predicted (in days); by default, this option is set to 1 day.

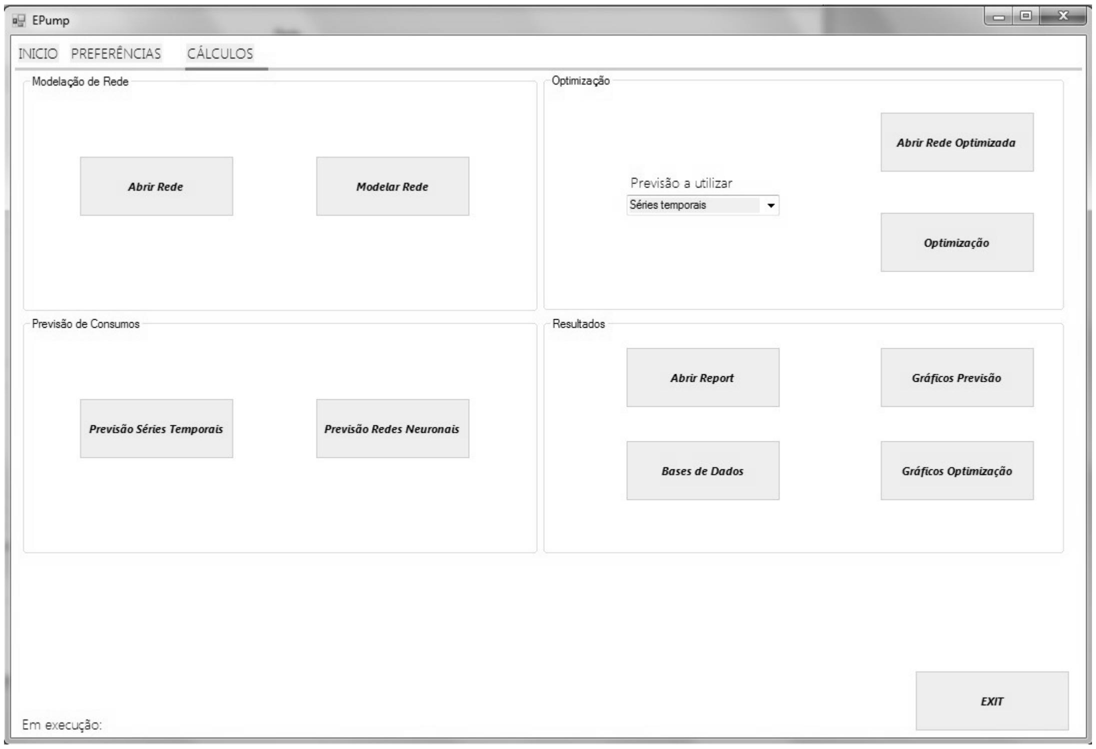


Figure 3.4: Calculation and Results Menu of E-Pumping

3.2.2 Calculation and Results Menu

- ***Modelação da rede* Box**
  This section allows the user to open the initial network (network before the optimisation process) with EPANET by just clicking the button "*Abrir Rede*". A second button, "*Modelar Rede*", automatically performs the hydraulic simulation of the selected network using EPANET.

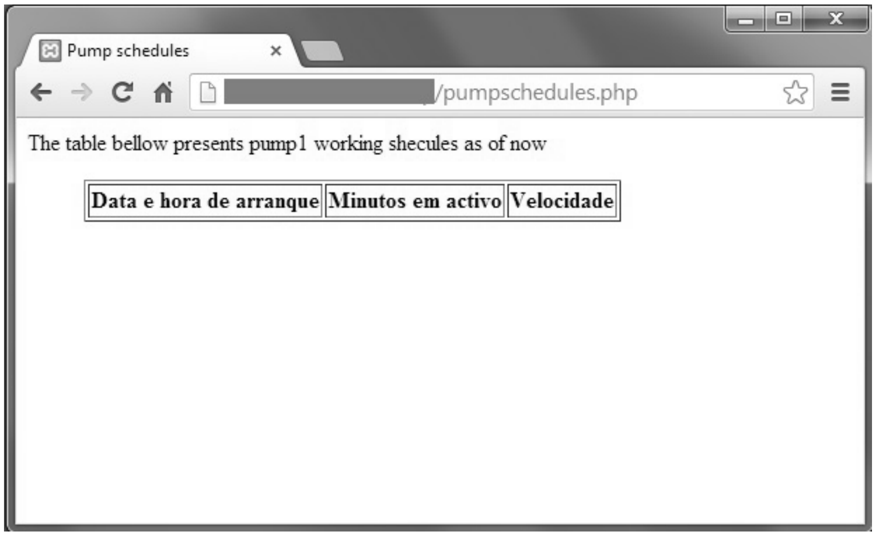
- ***Previsão de consumos* Box**
  This section allows the user to perform the water demand prediction for all the consumption nodes of the network. Each button initialise the prediction process using the correspondent forecasting method: (i) Time Series ("*Previsão Séries Temporais*") and/or (ii) Artificial Neural Networks ("*Previsão Redes Neuronais*").

- ***Optimização* Box**
  In this section, the user must first select the predicted results to be used in the optimisation by selecting one of the available options of the combo box list: (i) consumptions obtained with the Time Series method ("*Séries temporais*"), (ii) consumptions obtained with Artificial Neural Networks ("*Redes Neuronais*") or (iii) consumptions obtained by the average of both methods ("*Ambos os métodos*").
  The button "*Optimização*" automatically starts the optimisation process by using the optimisation strategy selected by the user in *PREFERÊNCIAS* Menu.
  After the optimisation finishes, the input file of the optimised network (network being operated with the best found controls) becomes available. Such input file can be easily accessed by clicking the button "*Abrir Rede Optimizada*", which opens the file using EPANET interface.

- ***Resultados* Box**
  This section allows the user to view prediction and optimisation results, as well as to access the database.
  The button "*Abrir Report*" opens the report file of the optimised network, generated by EPANET after the hydraulic simulation of the network under the optimised control conditions.
  The button "*Bases de Dados*" accesses the database of all consumption points of the network, as well as the database of both pumps and valves controls.
  The buttons "*Gráficos Previsão*" and "*Gráficos Optimização*" allows to access, respectively, results of the consumption prediction and of the control optimisation, both in a graphical format.

3.3 Online controls viewer

An additional simple tool was developed to work in parallel with E-Pumping in order to allow the easy access by the operators to the optimised controls for pumps and valves obtained by E-Pumping. Such tool consists in an online display containing the updated suggestion of control (see Figure 3.5).



The table bellow presents pump1 working shecules as of now

Data e hora de arranque	Minutos em activo	Velocidade

Figure 3.5: Online display of pump controls. The first column provides date and time to turn the pump on; the second column provides the duration of such operation; and the third column the pump speed to be used.

4 — DEMAND PREDICTION MODULE

4.1 Working with this module

Before starting to use E-Pumping prediction module, the user must develop an appropriate database for the network being analysed (see Section 2.5 on how to develop a database). Then, the user can open E-Pumping and select the *PREFERÊNCIAS* Menu in order to insert the name of the network in the *Rede* box and also the access information for the database in the *Base de Dados* box. After entering the desired number of weeks to be analysed by the prediction module and the number of days to be predicted in the *Previsão* box, the user must save the preferences by clicking the button "*GUARDAR REDE*".

In order to start the process of water demand prediction, the user must go to the *Previsão de Consumos* box in the *CÁLCULOS* Menu and click in the buttons to start the prediction with the correspondents forecasting methods. After both predictions are finalised, the user can view the results for all the consumption points of the network by accessing the data base clicking the button "*Bases de Dados*" in the *Resultados* box. For an easier visualisation and comparison of both methods performance, prediction results in a graphical format are also provided by E-Pumping and can be accessed by clicking the button "*Gráficos Previsão*" in the *Resultados* box.

4.2 Time series

Time series forecasting is the use of a model to predict future values based on previously observed values. While regression analysis is often employed in such a way as to test theories that the current values of one or more independent time series affect the current value of another time series, this type of analysis of time series focuses on comparing values of time series at different points in time. Models for time series data can have many forms and represent different stochastic processes. The time series model applied by E-Pumping prediction module is the multiplicative exponential smoothing model.

Time series data themselves are a sequence of observations. The observed phenomenon may be an essentially random process, or it may be an orderly, but noisy, process. Whereas in the simple moving average model, the past observations are weighted equally, exponential smoothing model assigns exponentially decreasing weights over time.

Exponential smoothing was first suggested by Robert Goodell Brown in 1956, and then expanded by Charles C. Holt in 1957.

The raw data sequence to be analysed is often represented by x_t, and the output of the exponential smoothing algorithm is commonly written as s_t, which may be regarded as a best estimate of what the next value of x will be. The simplest form of the exponential smoothing algorithm is given by:

$$\begin{aligned} s_1 &= x_0 \\ s_t &= \alpha x_{t-1} + (1-\alpha)s_{t-1} = s_{t-1} + \alpha(x_{t-1} - s_{t-1}), \qquad t > 1 \end{aligned} \tag{4.1}$$

where α is the smoothing factor and $0 < \alpha < 1$. In other words, the smoothed statistic s_t is a simple weighted average of the previous observation x_{t-1} and the previous smoothed statistic s_{t-1}. The term smoothing factor applied to α here is something of a misnomer, as larger values of α actually reduce the level of smoothing, and in the limiting case with $\alpha = 1$ the output series is just the same as the original series (with lag of one time unit). Simple exponential smoothing is easily applied, and it produces a smoothed statistic as soon as two observations are available. This is the main advantage of this kind of forecasting method when compared with methods based on artificial neural networks.

Values of α close to one have less of a smoothing effect and give greater weight to recent changes in the data, while values of α closer to zero have a greater smoothing effect and are less responsive to recent changes. There is no formally correct procedure for choosing α. Sometimes the statistician's judgement is used to choose an appropriate factor. Alternatively, a statistical technique may be used to optimise the value of α.

Unlike some other smoothing methods, this technique does not require any minimum number of observations to be made before it begins to produce results. In practice, however, a "good average" will not be achieved until several samples have been averaged together (see, for instance, wikipedia for more general details).

4.3 Artificial neural networks

There are many types of artificial neural networks (ANN). An artificial neural network is a computational simulation of a biological neural network. These models mimic the real life behaviour of neurons and the electrical messages they produce between input (such as from the eyes or nerve endings in the hand), processing by the brain and the final output from the brain (such as reacting to light or from sensing touch or heat). There are other ANNs which are adaptive systems used to model things such as environments and population.

ANNs apply the principle of function approximation by example, meaning that they learn a function by looking at examples of this function.

If an ANN is to be able to learn a problem, it must be defined as a function with a set of input and output variables supported by examples of how this function should work.

Artificial neurons are similar to their biological counterparts. They have input connections which are summed together to determine the strength of their output, which is the result of the sum being fed into an activation function. Though many activation functions exist, the most common is the sigmoid activation function, which outputs a number between 0 (for low input values) and 1 (for high input values). The resultant of this function is then passed as the input to other neurons through more connections, each of which are weighted. These weights determine the behaviour of the network.

In the human brain the neurons are connected in a seemingly random order and send impulses asynchronously. If we wanted to model a brain this might be the way to organise an ANN, but since we primarily want to create a function approximate, ANNs are usually not organised like this. When we create ANNs, the neurons are usually ordered in layers with connections going between the layers (see Figure 4.1). The first layer contains the input neurons and the last layer contains the output neurons. These input and output neurons represent the input and output variables of the function that we want to approximate. Between the input and the output layer a number of hidden layers exist and the connections (and weights) to and from these hidden layers determine how well the ANN performs. When an ANN is learning to approximate a function, it is shown examples of how the function works and the internal weights in the ANN are slowly adjusted so as to produce the same output as in the examples. The hope is that when the ANN is shown a new set of input variables, it will give a correct output.

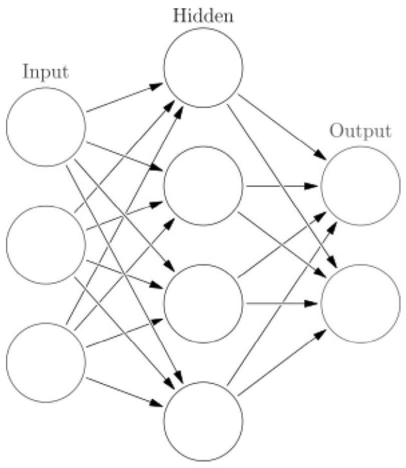


Figure 4.1: Scheme representing the layers of an artificial neural network. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another

When training an ANN with a set of input and output data, we wish to adjust the weights in the ANN, to make the ANN give the same outputs as seen in the training data. On the other hand, we do not want to make the ANN too specific, making it give precise results for the training data, but incorrect results for all other data. When this happens, we say that the ANN has been over-fitted.

The training process can be seen as an optimisation problem, where we wish to minimise the mean square error of the entire set of training data. This problem can be solved in many different ways, ranging from standard optimisation heuristics, through gradient descent algorithms like back-propagation.

The most used algorithm is the back-propagation training algorithm, which is the method used by E-Pumping prediction module. The training of the ANN performed by E-Pumping uses a set of data consisting on: date and time, air temperature, humidity, amount of rain fall and registered water demand.

The applied methodology considers a total of three layers (the input layer with 30 neurons, one hidden layer with 300 neurons and the output layer with 1 neuron) for the neural network.

The model of ANN used by the E-Pumping prediction module is based on the free libraries of Steffen Nissen named Fast Artificial Neural Network that could be found in `http://leenissen.dk/fann/wp`. These libraries have a GNU LGPL licence.

5 — OPTIMISATION MODULE

5.1 Working with this module

Before starting to use the optimisation module of E-Pumping, the user must open the input file of the network to be optimised with a text editor (such as Notepad) and ensure that the following steps are fulfilled:

1. Allow more EPANET iterations if the system is unbalanced - In section [OPTIONS] verify if the option "*Unbalanced Continue*" is set (the option "*Unbalanced Continue n*", with *n* = number of iterations, can also be set);
2. Save a unique hydraulic file instead of generating temporary hydraulic files in each iteration of EPANET - In section [OPTIONS] insert "*Hydraulics SAVE hydraulicFilename*"; this option allows the optimisation module to perform the number of optimisation iterations as desired, avoiding particular EPANET errors during the optimisation process; after introducing this option, the user may verify an error report message when opening the file with EPANET interface, however, such message can be ignored by clicking "OK" button;
3. Save energy results - In section [REPORT] enter the option "*Energy Yes*"; such option is crucial for the optimisation module in order to verify the costs related to certain operational decisions;
4. Save a complete status report - In section [REPORT] enter the option "*Status Full*"; the status report also provides warning messages that can occur during the hydraulic simulation; when such messages occur, the optimisation module do not accept the corresponding solution;
5. Make sure that specific control information for pumps and valves to optimise are correctly entered:
   - Each element to be optimised has associated a simple TIME control statement for each time-step, following the example of Table 2.5; it is important to notice that, for variable-speed pumps, the number of steps of the speed pattern must be equal to the number of controls (i.e., for instance, in a simulation period of 24 hours, if pattern time-steps of half an hour are used, then the number of controls for each element must be 48); variables to be used in the optimisation process are automatically obtained from such TIME controls and patterns information; if the user want to include some additional control for the system operation, instead of using TIME controls type, must use CLOCKTIME controls, in order not to cause conflicts during

the optimisation process;

- For each variable-speed pump, a speed pattern must be associated and the control settings must correspond to the speed at each hour (according to the speed pattern) instead of the status open/closed; in case of fixed-speed pumps and valves, no speed patterns should be associated and the control settings must indicate the status of each element (open or closed) at each hour of the simulation period).

After finalising the previous steps, the input file is ready to be used by the E-Pumping optimisation module. Thereby, the user should start deciding which optimisation options are the most adequate to the network that is going to be optimised. Some options can be kept with the default values provided by E-Pumping. However, some other options must take into account the characteristics of the network, meaning that an update of such options must be done for each new modelled network. The importance of such options update is related to the dependency of the optimisation problem on the network characteristics, as described on Section 5.3.

The options dependent on the network characteristics, which can be accessed on *PREFERÊNCIAS* Menu (see Chapter 3), are listed and explained below:

Individuals/Particles must be chosen according to the number of variables. The number of variables is dependent on the number of pumps and valves to be optimised and also on the number of time-steps defined by the user (usually, in this kind of problem, 1-hour time-steps are used). The number of individuals/particles must always exceed or, at least, be equal to the number of variables that can be determined by:

$$N_{variables} = N_{time-steps} \times (2 \times N_{VSpumps} + N_{FSpumps} + N_{valves}) \tag{5.1}$$

where

$N_{time-steps}$ is the number of time-steps,
$N_{VSpumps}$ the number of variable-speed pumps to be optimised,
$N_{FSpumps}$ the number of fixed-speed pumps to be optimised and
N_{valves} the number of valves to be optimised.

Warnings coefficient can never be a value close or inferior to the initial operational cost associated to the network to be optimised. The ideal value for this coefficient would be far superior to the initial cost (for example, with an initial daily cost of 10 €, the user can select a warning coefficient of 500 or even 1000).

Iterations number is an option that influences how long the optimisation module takes to complete the optimisation process, i.e., a larger number of iterations means more processing time. At the same time, a larger number of iterations can allow the optimisation algorithms to reach better results. Usually, the optimisation of more complex networks also implies larger number of iterations, since the optimisation problem difficulty increases.

5.2 Detecting common problems

During the optimisation process, errors related to some kind of misconfiguration can occur. Bellow, a list of common mistakes is available:

1. Input file not in the correct location of E-Pumping;
2. Input file not correctly configured - you must check the following steps (see also how to prepare correctly the input file on subsection 2.2.2):
   - Section **[REPORT]** must include de option for Energy and Status;
   - Section **[OPTIONS]** must include the option to continue the iterative process if the system is unbalanced and the option to save a hydraulic file;
   - Variable-speed pumps has always to have an associated **speed pattern**;
   - Section **[CONTROLS]** must contain TIME controls for each element to be optimised; the number of time controls for each element must be equal to the number of

time-steps; only the elements to be optimised can have TIME controls associated; for other elements control, other type of control must be used (CLOCKTIME controls, for example); the control speed settings for variable-speed pumps must correspond to its speed and for fixed-speed pumps or valves must correspond to their status (open/closed);
- **Energy patterns** must be associated to each pump to optimise;
- Water consumption junctions may have a value for the **base demand** and an associated **demand pattern**.

3. Some option from *PREFERÊNCIAS* Menu is **missing or** contain **invalid values** (see sections 2.3.1 and 5.1 on how to set such options);

Even without occurring any error during the simulation, the user may verify that the solution obtained by the optimisation module for the network operation was not improved. This can be associated to one of two main reasons:

1. The initial network modelled was already operating under optimal conditions and no improvements can be made;
2. Some optimisation options can be not well adjusted to the network being optimised. Firstly, the user should verify if the number of individuals/particles for DE/PSO is adequate to the number of variables. Also the penalty coefficient for warnings must be checked. Then, the user may try to increase the number of iterations allowed for the optimisation algorithms and check for improvements.

5.3 Optimisation problem description

An optimisation problem consists in the minimisation (or maximisation) of a certain function (Objective Function) dependent on particular variables (optimisation variables) that will determine the value of the function. Such function can also be subjected to a number of constraints which can be related to the variables boundaries or even to some particularities of the function. The specific problem of water supply systems operational optimisation consists in the minimisation of the total costs related to pumps operation. Constraints related to both variables and system operational requirements must be considered in this kind of problem. Thus, the main optimisation terms, applied to this specific problem, can be described as follows:

Objective Function - Corresponds to the total cost associated to the operation of the water network; E-Pumping determines the value of the Objective Function by running the network in EPANET 2.0;

Optimisation variables - Correspond to the operating times of the elements being controlled in the network (pumps or/and valves) and also to the relative rotational speeds considered for the variable-speed pumps; the control of such elements is directly related to the operational costs, thus, this variables must be changed until find the best control for the minimum cost;

Constraints - Correspond to limitations imposed to the problem, such as: (i) limits for the values of the optimisation variables (the maximum/minimum speed allowed for pumps and dimension of the time-steps); (ii) limits for the water level in tanks; (iii) pressure constraints; and (iv) constraints related to flow continuity and head loss in the network.

Mathematically, this optimisation problem can be, in a general way, described by:

$$\begin{array}{lll} \min_{\mathbf{x}} & f(\mathbf{x}), & \mathbf{x} = x_1, x_2, ..., x_{nvar} \\ \text{subject to:} & h_k(\mathbf{x}) = 0, & k = 1, ..., l; \\ & g_j(\mathbf{x}) \leq 0, & j = 1, ..., m; \\ & x_{min} < x < x_{max} & \end{array} \tag{5.2}$$

where $f(\mathbf{x})$ represents the objective function subject to the bounds of n_{var} decision variables (x_{min} and x_{max}) and subject to l equality constraints ($h_k(\mathbf{x})$) and m inequality constraints ($g_j(\mathbf{x})$).

The optimisation variables are defined by the relative pump speed (fraction between the actual speed and the nominal speed) at each time-step of the simulation and also by the fraction of time of pump operation for the correspondent time-step. In the case of fixed-speed pumps and valves, the variables are only defined by the fraction of operating time in the correspondent time-step. Therefore, the number of decision variables, n_{var}, can be determined by:

$$n_{var} = n_{steps} \times (2 \times n_{VSpumps} + n_{FSpumps} + n_{valves}) \tag{5.3}$$

where

n_{steps} is the number of time-steps,
$n_{VSpumps}$ the number of variable-speed pumps to be optimised,
$n_{FSpumps}$ the number of fixed-speed pumps to be optimised and
n_{valves} the number of valves to be optimised.

In E-Pumping computation, all the variables are transformed in order to take values between 0 and 1, where 0 represents the minimum relative speed or the minimum operating time and 1 the respective maximums.

Respecting the constraints, by default, the equation that describes the constraint for the water level in each tank at the end of the simulation, is given by :

$$L_{i,final} - L_{i,initial} \leq 0, \qquad i = 1, 2, ..., t \tag{5.4}$$

where $Li_{i,final}$ and $Li_{i,initial}$ are, respectively, the final and the initial water level of each i tank and t is the number of tanks. If the user

The constraint for the maximum water level allowed is given by:

$$L_i - L_{i,max} \leq 0, \qquad i = 1, 2, ..., t \tag{5.5}$$

and for the minimum water level allowed by:

$$L_{i,min} - L_i \leq 0, \qquad i = 1, 2, ..., t \tag{5.6}$$

where L_i is the actual water level in tank i, and $L_{i,max}$ and $L_{i,min}$ are, respectively, the maximum and the minimum water levels of each tank i. In case of these constraints violation, the objective function is penalised through the use of the exterior penalties method:

$$F(\mathbf{X}) = f(\mathbf{X}) + r_h \sum_{k=1}^{l} \left(h_k(\mathbf{X})\right)^2 + r_g \sum_{j=1}^{m} \left(\max\{0, g_j(\mathbf{X})\}\right)^2 \tag{5.7}$$

where F represents the augmented penalised objective function and r_h and r_g are the penalty coefficients.

Constraints related to limits of operation of pumps, pressure, flow continuity and head loss are handled by EPANET which retrieves warning messages in case of violation of such constraints. Such warning messages are also treated by E-Pumping as penalties in order to discard control solutions that induce such warnings.

E-Pumping deals with all the network information in parallel with EPANET in order to, in consecutive iterations, search for the best control conditions for the network being analysed.

This entire optimisation process described above is completely automatic in E-Pumping, allowing the user to quickly reach the best network operational controls at the minimum cost allowed, while satisfying all the requirements of the network.

A general scheme describing how the optimisation module works is provided in Figure 5.1. The linkage between the optimisation module and EPANET 2.0 calls, several times, a number of functions exported by EPANET toolkit.

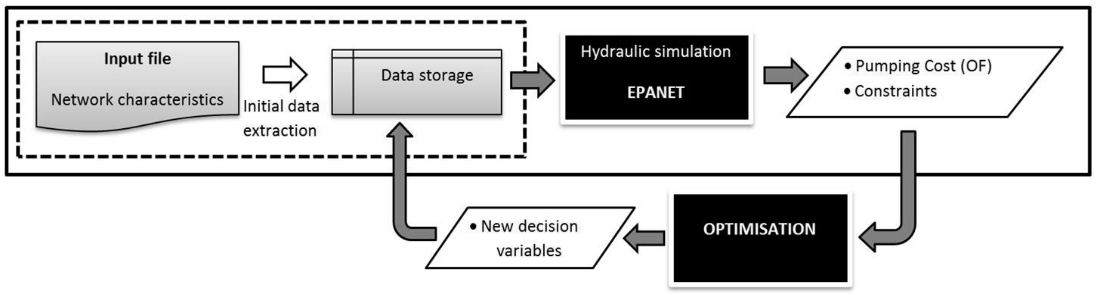


Figure 5.1: General scheme representing the linkage between the optimisation module and the hydraulic simulator EPANET 2.0. The grey-filled arrows represent the iterative cycle that occurs during the optimisation process.

At an initial stage, E-Pumping automatically reads and stores all the network information and the characteristics of all elements of the network (such as pumps, valves, etc.) available in the input file. Afterwards, a matrix containing all the variables necessary for the optimisation is generated. This matrix includes relative speeds (or settings) and operating time rates for both pumps and valves control. The matrix of the variables is then used in the iterative process of optimisation (represented by the grey-filled arrows of Figure 5.1). The optimisation variables are automatically sent to EPANET that performs the hydraulic analysis, provides the daily operational costs and constraint functions values. These results are then sent to the optimisation module to search for better variables in order to reduce the costs. The fit of the new variables provided by the optimisation algorithm is always checked by the hydraulic simulator and the iterative process only ends when the stop criterion (maximum number of iterations or optimal value found) is reached.

5.4 Optimisation algorithms

5.4.1 Differential Evolution - DE

The Differential Evolution (DE) is a meta-heuristic evolutionary algorithm initially introduced by Storn and Price (Storn and Price, 1995) that generates new solutions by combining members of the population.

A basic variant of the DE algorithm works by having a population of candidate solutions (called agents). These agents are moved around in the search-space by using simple mathematical formulae to combine the positions of existing agents from the population. If the new position of an agent is an improvement it is accepted and forms part of the population, otherwise the new position is simply discarded.

The DE algorithm combines simple arithmetic operators with recombination operators, mutation and crossover to evolve the randomly generated population into a final solution. The main advantages of the DE algorithm are its simple structure, easiness of use and good final results. DE is used for multidimensional real-valued functions although it does not use the gradient of the problem being optimised, which means DE does not require for the optimisation problem to be differentiable as is required by classic optimisation (Rocca *et al.*, 2011). DE can therefore also be used in optimisation problems that are not even continuous, are noisy, change over time, etc. Consequently, it can be used in hydraulic systems. The initial NP-sized population is generated randomly in a D-dimensional search space using:

$$x_{j,min} < x_{j,i,0} = rand_j[0,1](x_{j,max} - x_{j,min}) + x_{j,min} < x_{j,max} \tag{5.8}$$

with $i = 1,2,...,NP$ and $j = 1,2,...,D$. $x_{j,min}$ and $x_{j,max}$ are the lower and upper bounds, respectively, and $rand_j[0,1]$ is a randomly generated number between 0 and 1. Consecutively, the algorithm performs the mutation operation in which, for each target vector $x_{i,G}$ in a given generation G, a disturbed vector $x_{i,G+1}$ (also called donor vector) is generated accordingly to:

$$x_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}). \tag{5.9}$$

The weight coefficient (also called mutation factor) F, considered equal to 0.7 in E-Pumping, is a positive real number between 0 and 2 that controls the amplification of the differential variation. r_2 and r_3 are randomly integers from $1,2,...,NP$, but different from i. Depending on the mutation strategy, r_1 can either be a random integer or the best member of the population. In E-Pumping, a random integer is used. Upon approaching the mutation operation, the crossover operation is performed. In binomial crossover, the candidate vectors $x_{i,G+1}$ are generated by:

$$x_{i,G+1} = \begin{cases} x_{i,G+1} & \text{if } rand_j[0,1] \le CR \\ x_{i,G} & \text{if } rand_j[0,1] > CR. \end{cases} \tag{5.10}$$

The crossover probability CR, that is a value between $[0,1]$ defines the fraction of parameters that will be copied from the mutated vector. In E-Pumping $CR = 0.9$. Finally, the perturbed vector $x_{i,G+1}$ is compared with $x_{i,G}$. If $x_{i,G+1}$ has a better fitness, it replaces $x_{i,G}$ in the $G+1$ generation. Otherwise $x_{i,G}$ is kept. This process is repeated until a stopping criterion is satisfied, including a maximum number of evaluations of the objective function.

Being $\mathbf{x} \in \mathbb{R}^n$ a candidate solution (agent) in the population, the basic DE algorithm can then be described as follows:

1. Initialise all agents $\mathbf{x}$ with random positions in the search-space.
2. Until a termination criterion is met (e.g. number of iterations performed, or adequate fitness reached), repeat the following:
   - For each agent $\mathbf{x}$ in the population do:
     - Pick three agents $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$ from the population at random, they must be distinct from each other as well as from agent $\mathbf{x}$
     - Pick a random index $R \in \{1,\ldots,n\}$ (n being the dimensionality of the problem to be optimised).
     - Compute the agent's potentially new position $\mathbf{y} = [y_1,\ldots,y_n]$ as follows:
       (a) For each i, pick a uniformly distributed number $r_i \equiv U(0,1)$
       (b) If $r_i <$ CR or $i = R$ then set $y_i = a_i + F(b_i - c_i)$ otherwise set $y_i = x_i$
       (c) (In essence, the new position is outcome of binary crossover of agent $\mathbf{x}$ with intermediate agent $\mathbf{z} = \mathbf{a} + F(\mathbf{b} - \mathbf{c})$.)
     - If $f(\mathbf{y}) < f(\mathbf{x})$ then replace the agent in the population with the improved candidate solution, that is, replace $\mathbf{x}$ with $\mathbf{y}$ in the population.
3. Pick the agent from the population that has the highest fitness or lowest cost and return it as the best found candidate solution.

In E-Pumping optimisation module, the DE code proposed by Tetsuyuki Takahama is used (Takahama and Sakai, 2006).

5.4.2 Particle Swarm Optimisation - PSO

Particle swarm optimisation (PSO) is a population based stochastic optimisation technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behaviour of bird flocking or fish schooling.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialised with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far (the fitness value is also stored). This value is called pbest. Another "best" value that is tracked by the particle swarm optimiser is the best value, obtained so far by any particle in the neighbours of the particle. This location is called lbest. when a particle takes all the population as its topological neighbours, the best value is a global best and is called gbest.

The particle swarm optimisation concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations.

In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimisation has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Formally, let f be the cost function which must be minimised. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the objective function value of the given candidate solution. The gradient of f is not known. The goal is to find a solution a for which $f(a) \leq f(b)$ for all b in the search-space, which would mean a is the global minimum. Maximisation can be performed by considering the function $h = -f$ instead.

Let S be the number of particles in the swarm, each having a position $x_i \in \Re^n$ in the search-space and a velocity $v_i \in \Re$. Let p_i be the best known position of particle i and let g be the best known position of the entire swarm. A basic PSO algorithm is then:

1. For each particle $i = 1, \ldots, S$ do:
   - Initialise the particle's position with a uniformly distributed random vector;
   - Initialise the particle's best known position to its initial position: $p_i \leftarrow x_i$;
   - If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$;
   - Initialise the particle's velocity: v_i
2. Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:

```
• For each particle $i = 1, \ldots, S$ do:
    – Pick random numbers: $r_g$;
    – For each dimension $d = 1, \ldots, n$ do:
        (a) Update the particle's velocity: $v_{i,d}$;
    – Update the particle's position: $x_i$;
    – If $(f(x_i) < f(p_i)$ do:
        (a) Update the particle's best known position: $p_i \leftarrow x_i$;
        (b) If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$;
3. Now $g$ holds the best found solution.
```

5.4.3 Nelder-Mead Simplex - NMSimplex

A downhill simplex[1] method for finding a local minimum of a function of several variables has been devised by Nelder and Mead. For two variables, a simplex is a triangle, and the method is a pattern search that compares function values at the three vertices of a triangle. The worst vertex, where $f(x,y)$ is largest, is rejected and replaced with a new vertex. A new triangle is formed and the search is continued. The process generates a sequence of triangles (which might have different shapes), for which the function values at the vertices get smaller and smaller. The size of the triangles is reduced and the coordinates of the minimum point are found.

The algorithm is stated using the term simplex (a generalised triangle in N dimensions) and will find the minimum of a function of N variables. It is effective and computationally compact. The method does not require any derivative information, which makes it suitable for problems with non-smooth functions. It is widely used to solve parameter estimation and similar statistical problems, where the function values are uncertain or subject to noise. It can also be used for problems with discontinuous functions, which occur frequently in statistics and experimental mathematics. Unlike modern optimisation methods, the Nelder-Mead heuristic can converge to a non-stationary point unless the problem satisfies stronger conditions than are necessary for modern methods.

A general algorithm of the Nelder-Mead method includes the following steps:

1. **Order** according to the values at the vertices:
   $f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots \leq f(\mathbf{x}_{n+1})$
2. Calculate $\mathbf{x}_o$, the center of gravity of all points except $\mathbf{x}_{n+1}$.
3. **Reflection**
   Compute reflected point $\mathbf{x}_r = \mathbf{x}_o + \alpha(\mathbf{x}_o - \mathbf{x}_{n+1})$
   If the reflected point is better than the second worst, but not better than the best, i.e.: $f(\mathbf{x}_1) \leq f(\mathbf{x}_r) < f(\mathbf{x}_n)$,
   then obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the reflected point $\mathbf{x}_r$, and go to step 1.
4. **Expansion**
   If the reflected point is the best point so far, $f(\mathbf{x}_r) < f(\mathbf{x}_1)$,
   then compute the expanded point $\mathbf{x}_e = \mathbf{x}_o + \gamma(\mathbf{x}_o - \mathbf{x}_{n+1})$
   If the expanded point is better than the reflected point, $f(\mathbf{x}_e) < f(\mathbf{x}_r)$
   then obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the expanded point $\mathbf{x}_e$, and go to step 1.
   Else obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the reflected point $\mathbf{x}_r$, and go to step 1.
   Else (i.e. reflected point is not better than second worst) continue at step 5.
5. **Contraction**

[1]This method is called simplex due to the use of a geometrical simplex. It is not related to the simplex method used in linear programming

Here, it is certain that $f(\mathbf{x}_r) \geq f(\mathbf{x}_n)$
Compute contracted point $\mathbf{x}_c = \mathbf{x}_o + \rho(\mathbf{x}_o - \mathbf{x}_{n+1})$
If the contracted point is better than the worst point, i.e. $f(\mathbf{x}_c) < f(\mathbf{x}_{n+1})$
then obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the contracted point $\mathbf{x}_c$, and go to step 1.
Else go to step 6.

6. **Reduction**
   For all but the best point, replace the point with
   $\mathbf{x}_i = \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1)$ for all i $\in \{2, \ldots, n+1\}$. go to step 1.

α, γ, ρ and σ are respectively the reflection, the expansion, the contraction and the shrink coefficient. Standard values are $\alpha = 1$, $\gamma = 2$, $\rho = -1/2$ and $\sigma = 1/2$. For the reflection, since $\mathbf{x}_{n+1}$ is the vertex with the higher associated value among the vertices, we can expect to find a lower value at the reflection of $\mathbf{x}_{n+1}$ in the opposite face formed by all vertices point $\mathbf{x}_i$ except $\mathbf{x}_{n+1}$. For the expansion, if the reflection point $\mathbf{x}_r$ is the new minimum along the vertices we can expect to find interesting values along the direction from $\mathbf{x}_o$ to $\mathbf{x}_r$. Concerning the contraction: If $f(\mathbf{x}_r) > f(\mathbf{x}_n)$ we can expect that a better value will be inside the simplex formed by all the vertices $\mathbf{x}_i$. Finally, the reduction handles the rare case that contracting away from the largest point increases f, something that cannot happen sufficiently close to a non-singular minimum. In that case we contract towards the lowest point in the expectation of finding a simpler landscape.

For more details concerning the Nelder-Mead simplex method, see non-linear optimisation books or even wikipedia.

5.5 Optimisation strategies

It is commonly accepted in the scientific community that there exists no unique algorithm robust enough to deal with every possible situation. The conclusion of some researchers is being that, although any problem can be solved by at least one optimisation algorithm, no individual algorithm is able to solve all the problems (Ponthot and Kleinermann, 2006).

In order to alleviate convergence difficulties, the so-called cascade optimisation strategies have been used. In this strategy, a multi-stage procedure is applied, in which various optimisation algorithms are activated one after the other in a pre-specified sequence. Of course, the aim of this strategy is to take advantage of the strength of each selected algorithm. It should also be noted that the sequence can be either of a deterministic type or of an evolutionary type, or it can even combine gradient and evolutionary algorithms (Ponthot and Kleinermann, 2006).

E-Pumping optimisation module can also solve a problem by using a cascade strategy that can be defined by the user. The strategy can be created by specifying, in *Optimização* box of the *PREFERÊNCIAS* Menu, the number of sequences and the correspondent sequencing by setting the abbreviations of the desired algorithms separated by spaces and following the desired order. E-Pumping automatically performs the specified strategy by running each algorithm sequentially, starting with a randomly selected initial population and then using the best solution obtained as an initial solution of the following algorithm.

Bibliography

Ponthot, J.-P., Kleinermann, J.-P. (2006), A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation, Vol. 195, pp. 5472-5508.

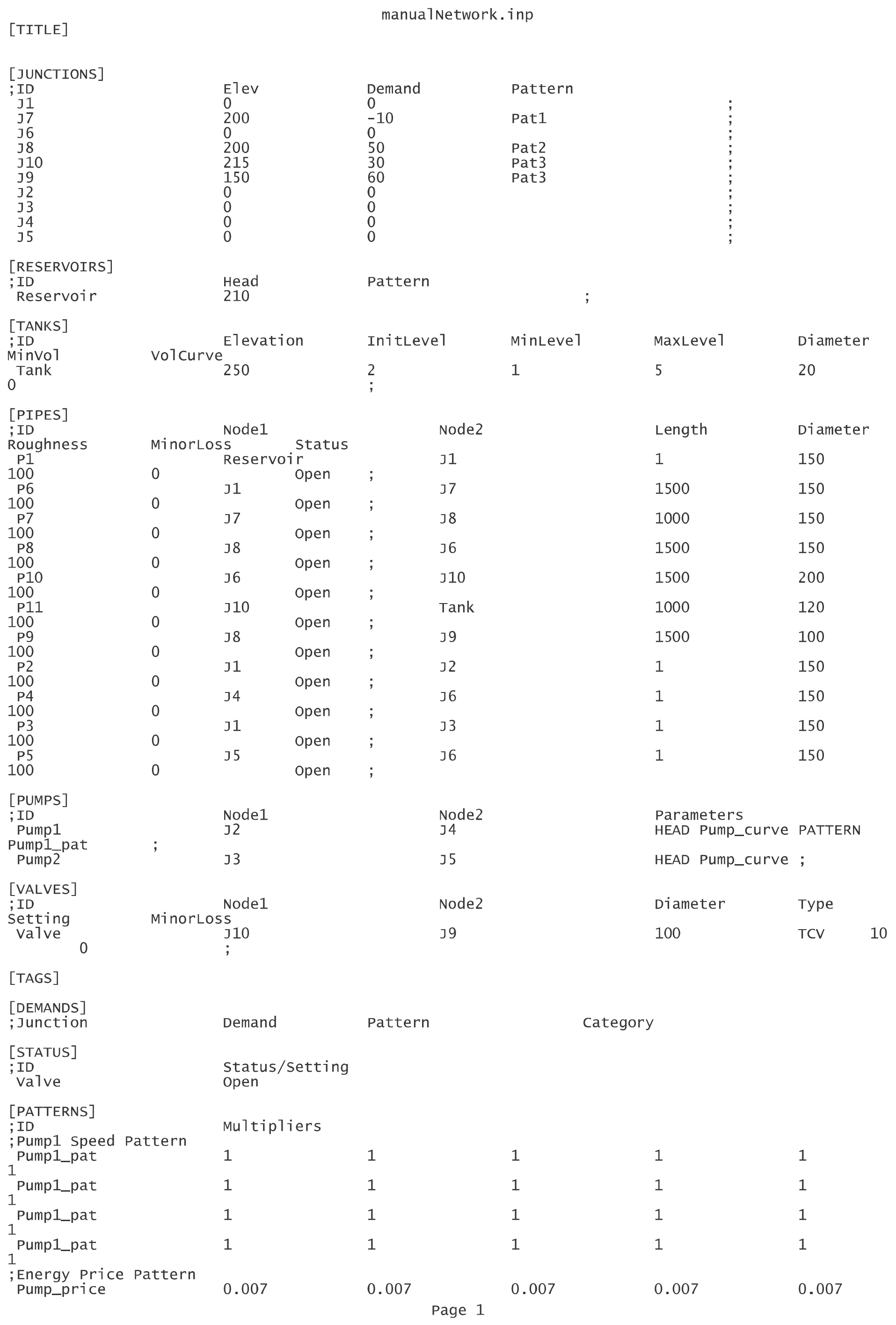
Rocca, P., Oliveri, G., Massa, A. (2011), Differential Evolution as Applied to Electromagnetics, IEEE Antennas and Propagation Magazine, Vol. 53(1), pp. 38-49.

Rossman, L. A. (2000), EPANET 2 Users manual, EPA-Environmental Protection Agency: Cincinnati. Available online: `http://nepis.epa.gov/Adobe/PDF/P1007WWU.pdf`

Storn, R., Price, K. (1995), Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, International Computer Science Institute: Berkeley.

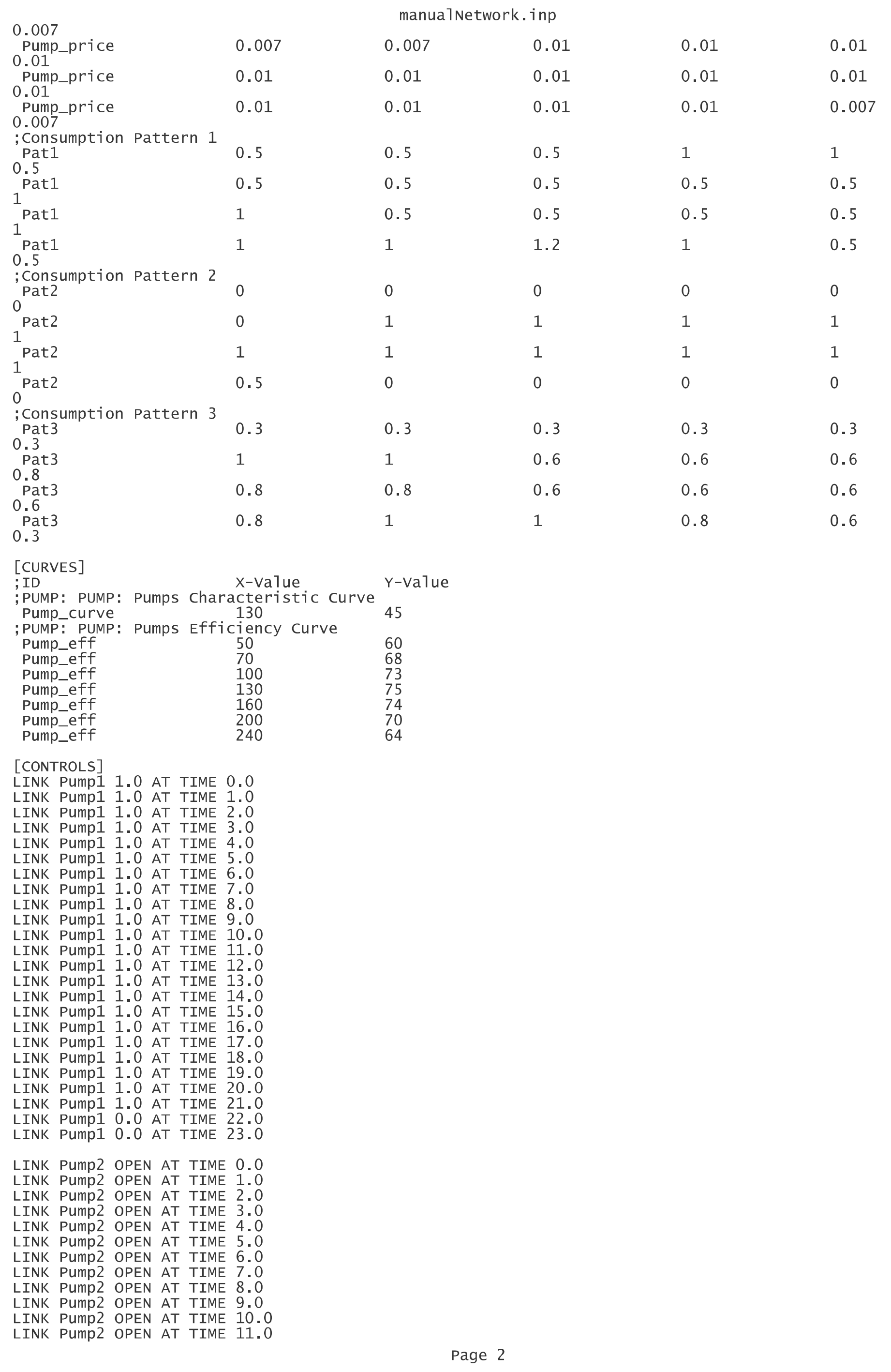
Takahama, T., Sakai, S. (2006), Constrained optimization by the "Constrained differential evolution with gradient-based mutation and feasible elites", Proc. of the 2006 IEEE Congress on Evolutionary Computation, pp. 308-315.

A — Input file



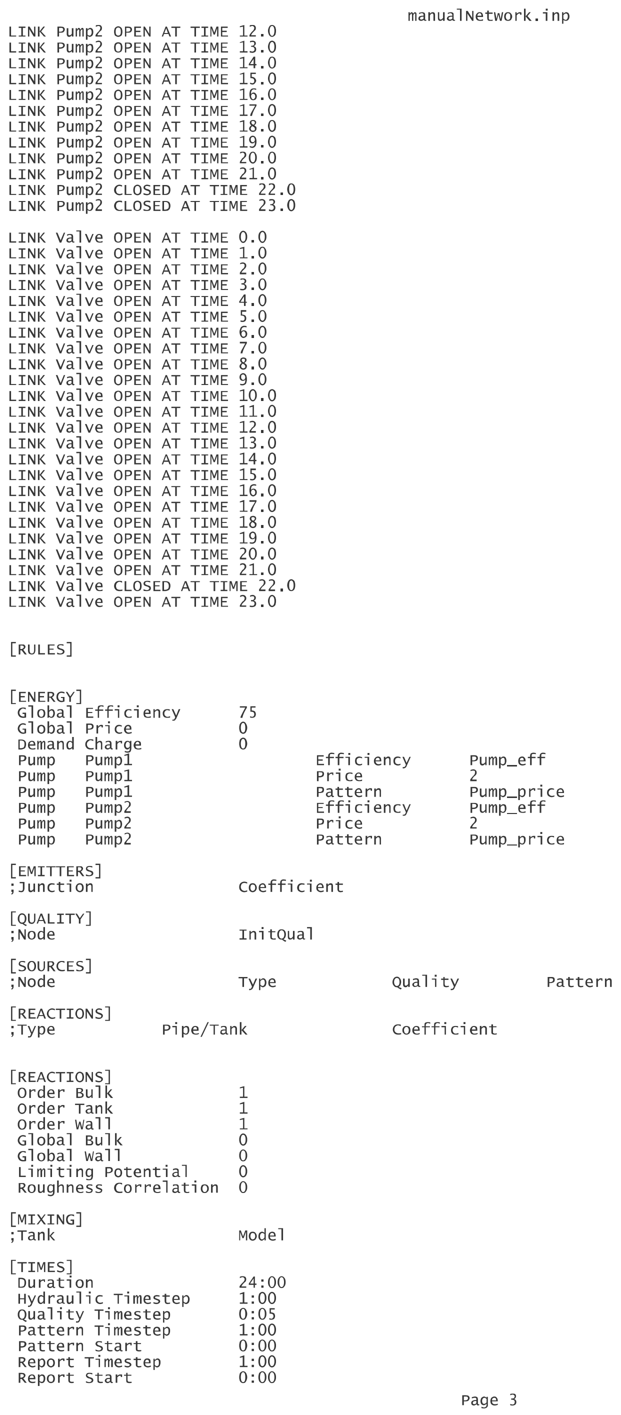
```
                                          manualNetwork.inp
[TITLE]


[JUNCTIONS]
;ID                     Elev            Demand          Pattern
 J1                     0               0                                       ;
 J7                     200             -10             Pat1                    ;
 J6                     0               0                                       ;
 J8                     200             50              Pat2                    ;
 J10                    215             30              Pat3                    ;
 J9                     150             60              Pat3                    ;
 J2                     0               0                                       ;
 J3                     0               0                                       ;
 J4                     0               0                                       ;
 J5                     0               0                                       ;

[RESERVOIRS]
;ID                     Head            Pattern
 Reservoir              210                                     ;

[TANKS]
;ID                     Elevation       InitLevel       MinLevel        MaxLevel        Diameter
MinVol          VolCurve
 Tank                   250             2               1               5               20
0                                       ;

[PIPES]
;ID                     Node1                   Node2                   Length          Diameter
Roughness       MinorLoss       Status
 P1                     Reservoir               J1                      1               150
100             0               Open    ;
 P6                     J1                      J7                      1500            150
100             0               Open    ;
 P7                     J7                      J8                      1000            150
100             0               Open    ;
 P8                     J8                      J6                      1500            150
100             0               Open    ;
 P10                    J6                      J10                     1500            200
100             0               Open    ;
 P11                    J10                     Tank                    1000            120
100             0               Open    ;
 P9                     J8                      J9                      1500            100
100             0               Open    ;
 P2                     J1                      J2                      1               150
100             0               Open    ;
 P4                     J4                      J6                      1               150
100             0               Open    ;
 P3                     J1                      J3                      1               150
100             0               Open    ;
 P5                     J5                      J6                      1               150
100             0               Open    ;

[PUMPS]
;ID                     Node1                   Node2                   Parameters
 Pump1                  J2                      J4                      HEAD Pump_curve PATTERN
Pump1_pat       ;
 Pump2                  J3                      J5                      HEAD Pump_curve ;

[VALVES]
;ID                     Node1                   Node2                   Diameter        Type
Setting         MinorLoss
 Valve                  J10                     J9                      100             TCV     10
        0               ;

[TAGS]

[DEMANDS]
;Junction               Demand          Pattern                 Category

[STATUS]
;ID                     Status/Setting
 Valve                  Open

[PATTERNS]
;ID                     Multipliers
;Pump1 Speed Pattern
 Pump1_pat              1               1               1               1               1
1
 Pump1_pat              1               1               1               1               1
1
 Pump1_pat              1               1               1               1               1
1
 Pump1_pat              1               1               1               1               1
1
;Energy Price Pattern
 Pump_price             0.007           0.007           0.007           0.007           0.007
                                               Page 1
```

Figure A.1: Input file (page 1) of the network used in the tutorial provided on chapter 2 of this manual.



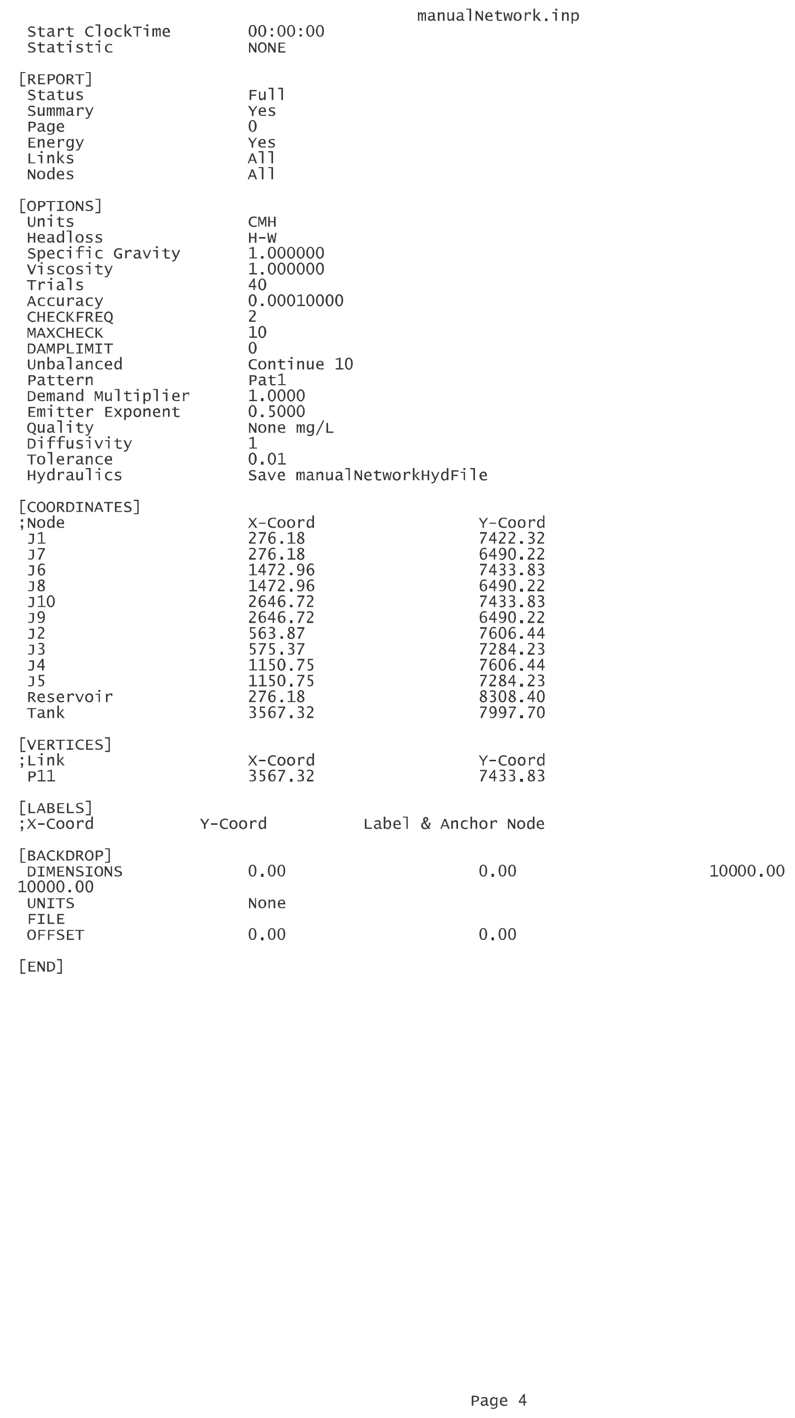
```
                                        manualNetwork.inp
0.007
 Pump_price             0.007           0.007           0.01            0.01            0.01
0.01
 Pump_price             0.01            0.01            0.01            0.01            0.01
0.01
 Pump_price             0.01            0.01            0.01            0.01            0.007
0.007
;Consumption Pattern 1
 Pat1                   0.5             0.5             0.5             1               1
0.5
 Pat1                   0.5             0.5             0.5             0.5             0.5
1
 Pat1                   1               0.5             0.5             0.5             0.5
1
 Pat1                   1               1               1.2             1               0.5
0.5
;Consumption Pattern 2
 Pat2                   0               0               0               0               0
0
 Pat2                   0               1               1               1               1
1
 Pat2                   1               1               1               1               1
1
 Pat2                   0.5             0               0               0               0
0
;Consumption Pattern 3
 Pat3                   0.3             0.3             0.3             0.3             0.3
0.3
 Pat3                   1               1               0.6             0.6             0.6
0.8
 Pat3                   0.8             0.8             0.6             0.6             0.6
0.6
 Pat3                   0.8             1               1               0.8             0.6
0.3

[CURVES]
;ID                     X-Value         Y-Value
;PUMP: PUMP: Pumps Characteristic Curve
 Pump_curve             130             45
;PUMP: PUMP: Pumps Efficiency Curve
 Pump_eff               50              60
 Pump_eff               70              68
 Pump_eff               100             73
 Pump_eff               130             75
 Pump_eff               160             74
 Pump_eff               200             70
 Pump_eff               240             64

[CONTROLS]
LINK Pump1 1.0 AT TIME 0.0
LINK Pump1 1.0 AT TIME 1.0
LINK Pump1 1.0 AT TIME 2.0
LINK Pump1 1.0 AT TIME 3.0
LINK Pump1 1.0 AT TIME 4.0
LINK Pump1 1.0 AT TIME 5.0
LINK Pump1 1.0 AT TIME 6.0
LINK Pump1 1.0 AT TIME 7.0
LINK Pump1 1.0 AT TIME 8.0
LINK Pump1 1.0 AT TIME 9.0
LINK Pump1 1.0 AT TIME 10.0
LINK Pump1 1.0 AT TIME 11.0
LINK Pump1 1.0 AT TIME 12.0
LINK Pump1 1.0 AT TIME 13.0
LINK Pump1 1.0 AT TIME 14.0
LINK Pump1 1.0 AT TIME 15.0
LINK Pump1 1.0 AT TIME 16.0
LINK Pump1 1.0 AT TIME 17.0
LINK Pump1 1.0 AT TIME 18.0
LINK Pump1 1.0 AT TIME 19.0
LINK Pump1 1.0 AT TIME 20.0
LINK Pump1 1.0 AT TIME 21.0
LINK Pump1 0.0 AT TIME 22.0
LINK Pump1 0.0 AT TIME 23.0

LINK Pump2 OPEN AT TIME 0.0
LINK Pump2 OPEN AT TIME 1.0
LINK Pump2 OPEN AT TIME 2.0
LINK Pump2 OPEN AT TIME 3.0
LINK Pump2 OPEN AT TIME 4.0
LINK Pump2 OPEN AT TIME 5.0
LINK Pump2 OPEN AT TIME 6.0
LINK Pump2 OPEN AT TIME 7.0
LINK Pump2 OPEN AT TIME 8.0
LINK Pump2 OPEN AT TIME 9.0
LINK Pump2 OPEN AT TIME 10.0
LINK Pump2 OPEN AT TIME 11.0
                                              Page 2
```

Figure A.2: Input file (page 2) of the network used in the tutorial provided on chapter 2 of this manual.



```
                                         manualNetwork.inp
LINK Pump2 OPEN AT TIME 12.0
LINK Pump2 OPEN AT TIME 13.0
LINK Pump2 OPEN AT TIME 14.0
LINK Pump2 OPEN AT TIME 15.0
LINK Pump2 OPEN AT TIME 16.0
LINK Pump2 OPEN AT TIME 17.0
LINK Pump2 OPEN AT TIME 18.0
LINK Pump2 OPEN AT TIME 19.0
LINK Pump2 OPEN AT TIME 20.0
LINK Pump2 OPEN AT TIME 21.0
LINK Pump2 CLOSED AT TIME 22.0
LINK Pump2 CLOSED AT TIME 23.0

LINK Valve OPEN AT TIME 0.0
LINK Valve OPEN AT TIME 1.0
LINK Valve OPEN AT TIME 2.0
LINK Valve OPEN AT TIME 3.0
LINK Valve OPEN AT TIME 4.0
LINK Valve OPEN AT TIME 5.0
LINK Valve OPEN AT TIME 6.0
LINK Valve OPEN AT TIME 7.0
LINK Valve OPEN AT TIME 8.0
LINK Valve OPEN AT TIME 9.0
LINK Valve OPEN AT TIME 10.0
LINK Valve OPEN AT TIME 11.0
LINK Valve OPEN AT TIME 12.0
LINK Valve OPEN AT TIME 13.0
LINK Valve OPEN AT TIME 14.0
LINK Valve OPEN AT TIME 15.0
LINK Valve OPEN AT TIME 16.0
LINK Valve OPEN AT TIME 17.0
LINK Valve OPEN AT TIME 18.0
LINK Valve OPEN AT TIME 19.0
LINK Valve OPEN AT TIME 20.0
LINK Valve OPEN AT TIME 21.0
LINK Valve CLOSED AT TIME 22.0
LINK Valve OPEN AT TIME 23.0


[RULES]


[ENERGY]
 Global Efficiency      75
 Global Price           0
 Demand Charge          0
 Pump   Pump1                   Efficiency      Pump_eff
 Pump   Pump1                   Price           2
 Pump   Pump1                   Pattern         Pump_price
 Pump   Pump2                   Efficiency      Pump_eff
 Pump   Pump2                   Price           2
 Pump   Pump2                   Pattern         Pump_price

[EMITTERS]
;Junction               Coefficient

[QUALITY]
;Node                   InitQual

[SOURCES]
;Node                   Type            Quality         Pattern

[REACTIONS]
;Type           Pipe/Tank               Coefficient


[REACTIONS]
 Order Bulk             1
 Order Tank             1
 Order Wall             1
 Global Bulk            0
 Global Wall            0
 Limiting Potential     0
 Roughness Correlation  0

[MIXING]
;Tank                   Model

[TIMES]
 Duration               24:00
 Hydraulic Timestep     1:00
 Quality Timestep       0:05
 Pattern Timestep       1:00
 Pattern Start          0:00
 Report Timestep        1:00
 Report Start           0:00
                                               Page 3
```

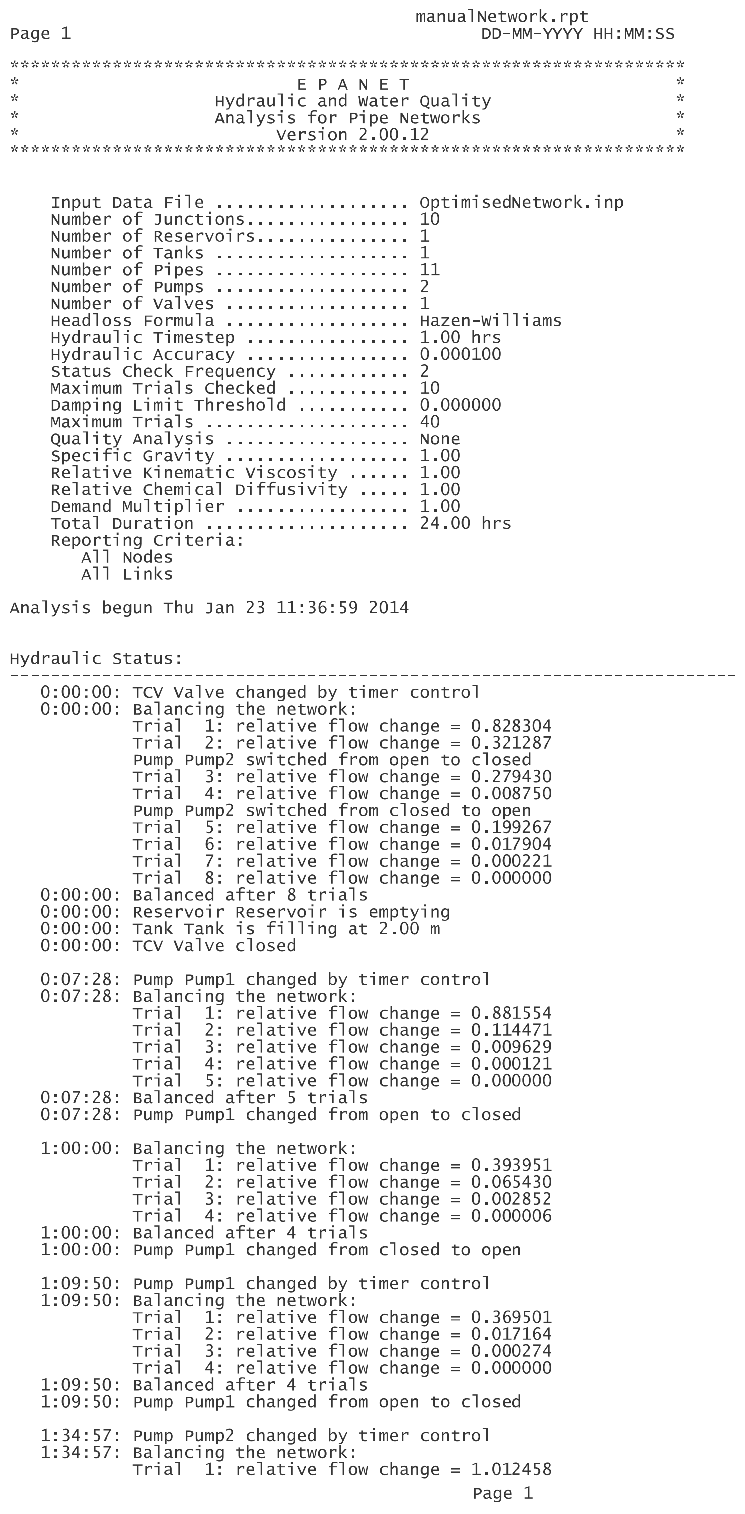
Figure A.3: Input file (page 3) of the network used in the tutorial provided on chapter 2 of this manual.



```
                                           manualNetwork.inp
 Start ClockTime        00:00:00
 Statistic              NONE

[REPORT]
 Status                 Full
 Summary                Yes
 Page                   0
 Energy                 Yes
 Links                  All
 Nodes                  All

[OPTIONS]
 Units                  CMH
 Headloss               H-W
 Specific Gravity       1.000000
 Viscosity              1.000000
 Trials                 40
 Accuracy               0.00010000
 CHECKFREQ              2
 MAXCHECK               10
 DAMPLIMIT              0
 Unbalanced             Continue 10
 Pattern                Pat1
 Demand Multiplier      1.0000
 Emitter Exponent       0.5000
 Quality                None mg/L
 Diffusivity            1
 Tolerance              0.01
 Hydraulics             Save manualNetworkHydFile

[COORDINATES]
;Node                   X-Coord                 Y-Coord
 J1                     276.18                  7422.32
 J7                     276.18                  6490.22
 J6                     1472.96                 7433.83
 J8                     1472.96                 6490.22
 J10                    2646.72                 7433.83
 J9                     2646.72                 6490.22
 J2                     563.87                  7606.44
 J3                     575.37                  7284.23
 J4                     1150.75                 7606.44
 J5                     1150.75                 7284.23
 Reservoir              276.18                  8308.40
 Tank                   3567.32                 7997.70

[VERTICES]
;Link                   X-Coord                 Y-Coord
 P11                    3567.32                 7433.83

[LABELS]
;X-Coord           Y-Coord          Label & Anchor Node

[BACKDROP]
 DIMENSIONS             0.00                    0.00                    10000.00
10000.00
 UNITS                  None
 FILE
 OFFSET                 0.00                    0.00

[END]




                                             Page 4
```

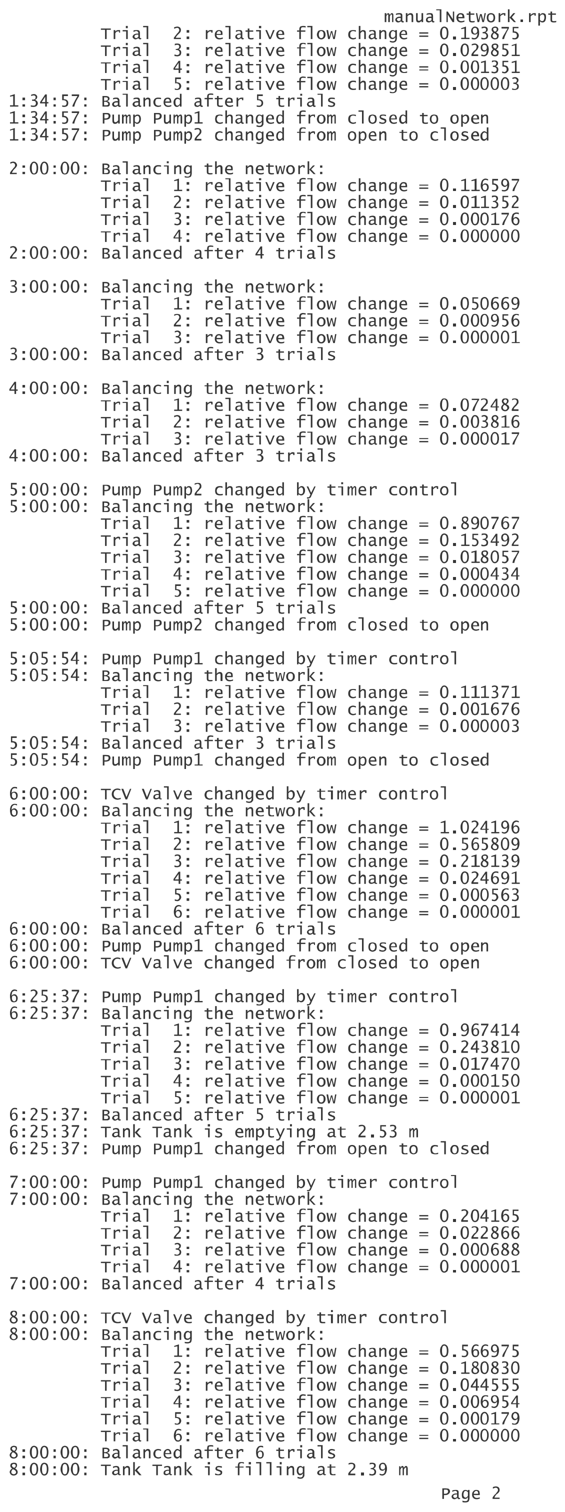
Figure A.4: Input file (page 4) of the network used in the tutorial provided on chapter 2 of this manual.

B — Report file



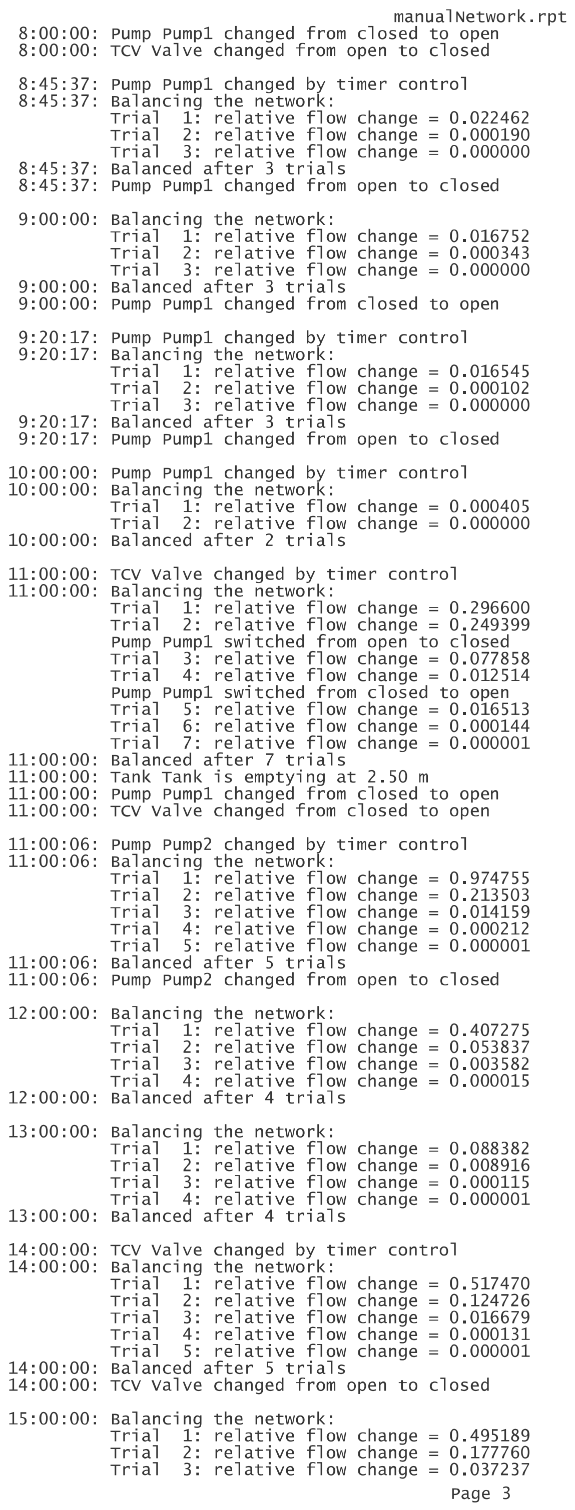
```
                                        manualNetwork.rpt
Page 1                                        DD-MM-YYYY HH:MM:SS

******************************************************************
*                           E P A N E T                          *
*                   Hydraulic and Water Quality                  *
*                   Analysis for Pipe Networks                   *
*                         Version 2.00.12                        *
******************************************************************


    Input Data File ................... OptimisedNetwork.inp
    Number of Junctions................ 10
    Number of Reservoirs............... 1
    Number of Tanks ................... 1
    Number of Pipes ................... 11
    Number of Pumps ................... 2
    Number of Valves .................. 1
    Headloss Formula .................. Hazen-Williams
    Hydraulic Timestep ................ 1.00 hrs
    Hydraulic Accuracy ................ 0.000100
    Status Check Frequency ............ 2
    Maximum Trials Checked ............ 10
    Damping Limit Threshold ........... 0.000000
    Maximum Trials .................... 40
    Quality Analysis .................. None
    Specific Gravity .................. 1.00
    Relative Kinematic Viscosity ...... 1.00
    Relative Chemical Diffusivity ..... 1.00
    Demand Multiplier ................. 1.00
    Total Duration .................... 24.00 hrs
    Reporting Criteria:
       All Nodes
       All Links

Analysis begun Thu Jan 23 11:36:59 2014


Hydraulic Status:
-----------------------------------------------------------------------
   0:00:00: TCV Valve changed by timer control
   0:00:00: Balancing the network:
            Trial  1: relative flow change = 0.828304
            Trial  2: relative flow change = 0.321287
            Pump Pump2 switched from open to closed
            Trial  3: relative flow change = 0.279430
            Trial  4: relative flow change = 0.008750
            Pump Pump2 switched from closed to open
            Trial  5: relative flow change = 0.199267
            Trial  6: relative flow change = 0.017904
            Trial  7: relative flow change = 0.000221
            Trial  8: relative flow change = 0.000000
   0:00:00: Balanced after 8 trials
   0:00:00: Reservoir Reservoir is emptying
   0:00:00: Tank Tank is filling at 2.00 m
   0:00:00: TCV Valve closed

   0:07:28: Pump Pump1 changed by timer control
   0:07:28: Balancing the network:
            Trial  1: relative flow change = 0.881554
            Trial  2: relative flow change = 0.114471
            Trial  3: relative flow change = 0.009629
            Trial  4: relative flow change = 0.000121
            Trial  5: relative flow change = 0.000000
   0:07:28: Balanced after 5 trials
   0:07:28: Pump Pump1 changed from open to closed

   1:00:00: Balancing the network:
            Trial  1: relative flow change = 0.393951
            Trial  2: relative flow change = 0.065430
            Trial  3: relative flow change = 0.002852
            Trial  4: relative flow change = 0.000006
   1:00:00: Balanced after 4 trials
   1:00:00: Pump Pump1 changed from closed to open

   1:09:50: Pump Pump1 changed by timer control
   1:09:50: Balancing the network:
            Trial  1: relative flow change = 0.369501
            Trial  2: relative flow change = 0.017164
            Trial  3: relative flow change = 0.000274
            Trial  4: relative flow change = 0.000000
   1:09:50: Balanced after 4 trials
   1:09:50: Pump Pump1 changed from open to closed

   1:34:57: Pump Pump2 changed by timer control
   1:34:57: Balancing the network:
            Trial  1: relative flow change = 1.012458
                                                  Page 1
```

Figure B.1: Report file (page 1) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



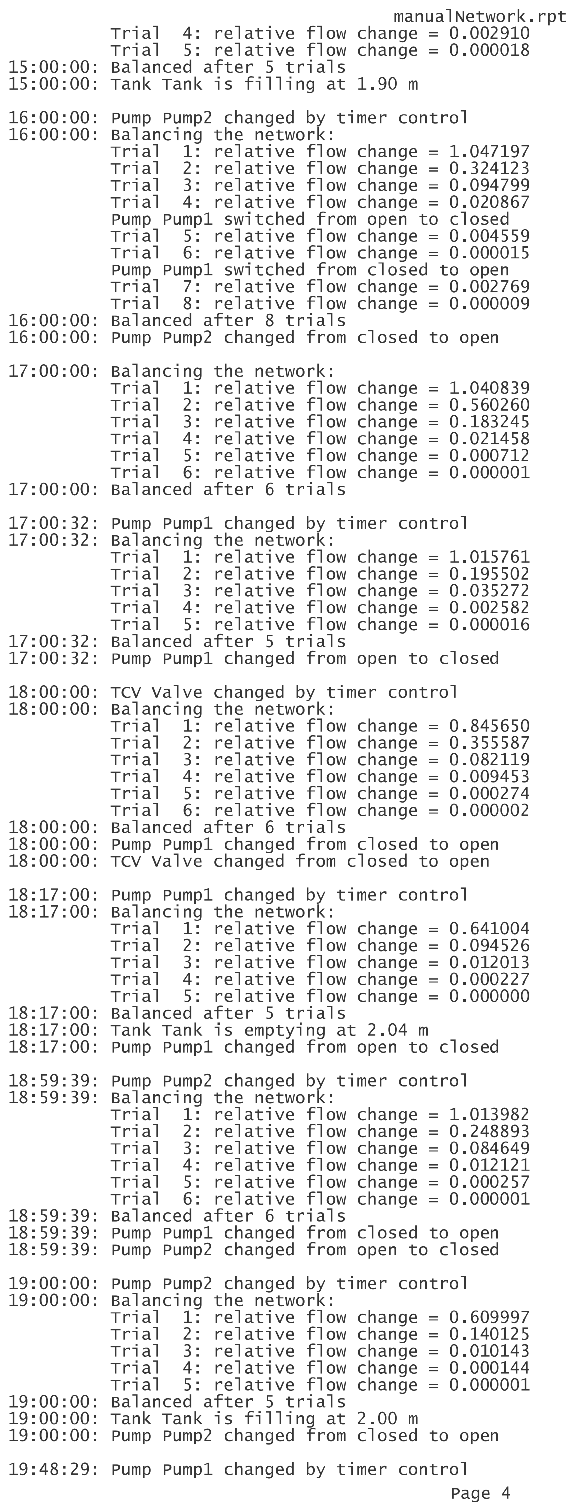
```
                                   manualNetwork.rpt
         Trial  2: relative flow change = 0.193875
         Trial  3: relative flow change = 0.029851
         Trial  4: relative flow change = 0.001351
         Trial  5: relative flow change = 0.000003
1:34:57: Balanced after 5 trials
1:34:57: Pump Pump1 changed from closed to open
1:34:57: Pump Pump2 changed from open to closed

2:00:00: Balancing the network:
         Trial  1: relative flow change = 0.116597
         Trial  2: relative flow change = 0.011352
         Trial  3: relative flow change = 0.000176
         Trial  4: relative flow change = 0.000000
2:00:00: Balanced after 4 trials

3:00:00: Balancing the network:
         Trial  1: relative flow change = 0.050669
         Trial  2: relative flow change = 0.000956
         Trial  3: relative flow change = 0.000001
3:00:00: Balanced after 3 trials

4:00:00: Balancing the network:
         Trial  1: relative flow change = 0.072482
         Trial  2: relative flow change = 0.003816
         Trial  3: relative flow change = 0.000017
4:00:00: Balanced after 3 trials

5:00:00: Pump Pump2 changed by timer control
5:00:00: Balancing the network:
         Trial  1: relative flow change = 0.890767
         Trial  2: relative flow change = 0.153492
         Trial  3: relative flow change = 0.018057
         Trial  4: relative flow change = 0.000434
         Trial  5: relative flow change = 0.000000
5:00:00: Balanced after 5 trials
5:00:00: Pump Pump2 changed from closed to open

5:05:54: Pump Pump1 changed by timer control
5:05:54: Balancing the network:
         Trial  1: relative flow change = 0.111371
         Trial  2: relative flow change = 0.001676
         Trial  3: relative flow change = 0.000003
5:05:54: Balanced after 3 trials
5:05:54: Pump Pump1 changed from open to closed

6:00:00: TCV Valve changed by timer control
6:00:00: Balancing the network:
         Trial  1: relative flow change = 1.024196
         Trial  2: relative flow change = 0.565809
         Trial  3: relative flow change = 0.218139
         Trial  4: relative flow change = 0.024691
         Trial  5: relative flow change = 0.000563
         Trial  6: relative flow change = 0.000001
6:00:00: Balanced after 6 trials
6:00:00: Pump Pump1 changed from closed to open
6:00:00: TCV Valve changed from closed to open

6:25:37: Pump Pump1 changed by timer control
6:25:37: Balancing the network:
         Trial  1: relative flow change = 0.967414
         Trial  2: relative flow change = 0.243810
         Trial  3: relative flow change = 0.017470
         Trial  4: relative flow change = 0.000150
         Trial  5: relative flow change = 0.000001
6:25:37: Balanced after 5 trials
6:25:37: Tank Tank is emptying at 2.53 m
6:25:37: Pump Pump1 changed from open to closed

7:00:00: Pump Pump1 changed by timer control
7:00:00: Balancing the network:
         Trial  1: relative flow change = 0.204165
         Trial  2: relative flow change = 0.022866
         Trial  3: relative flow change = 0.000688
         Trial  4: relative flow change = 0.000001
7:00:00: Balanced after 4 trials

8:00:00: TCV Valve changed by timer control
8:00:00: Balancing the network:
         Trial  1: relative flow change = 0.566975
         Trial  2: relative flow change = 0.180830
         Trial  3: relative flow change = 0.044555
         Trial  4: relative flow change = 0.006954
         Trial  5: relative flow change = 0.000179
         Trial  6: relative flow change = 0.000000
8:00:00: Balanced after 6 trials
8:00:00: Tank Tank is filling at 2.39 m
                                        Page 2
```

Figure B.2: Report file (page 2) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



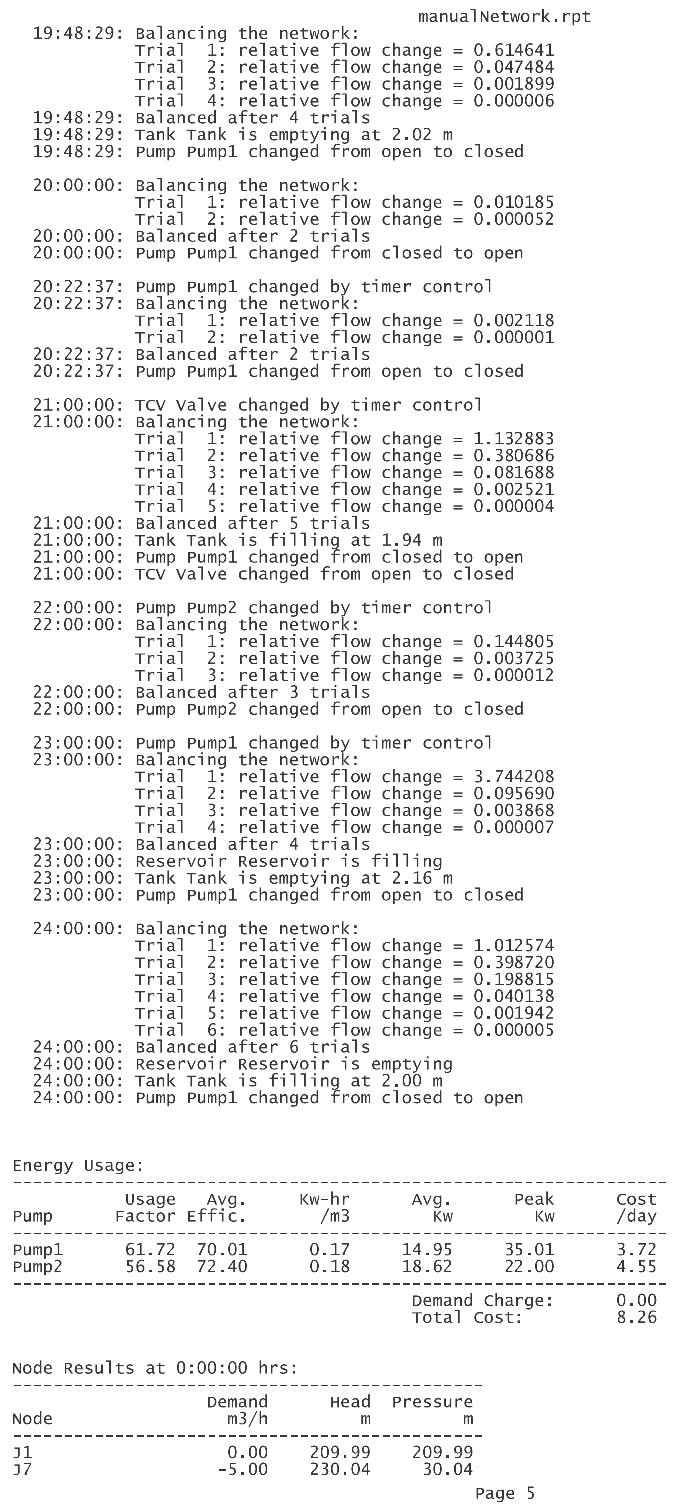
```
                                        manualNetwork.rpt
  8:00:00: Pump Pump1 changed from closed to open
  8:00:00: TCV Valve changed from open to closed

  8:45:37: Pump Pump1 changed by timer control
  8:45:37: Balancing the network:
           Trial  1: relative flow change = 0.022462
           Trial  2: relative flow change = 0.000190
           Trial  3: relative flow change = 0.000000
  8:45:37: Balanced after 3 trials
  8:45:37: Pump Pump1 changed from open to closed

  9:00:00: Balancing the network:
           Trial  1: relative flow change = 0.016752
           Trial  2: relative flow change = 0.000343
           Trial  3: relative flow change = 0.000000
  9:00:00: Balanced after 3 trials
  9:00:00: Pump Pump1 changed from closed to open

  9:20:17: Pump Pump1 changed by timer control
  9:20:17: Balancing the network:
           Trial  1: relative flow change = 0.016545
           Trial  2: relative flow change = 0.000102
           Trial  3: relative flow change = 0.000000
  9:20:17: Balanced after 3 trials
  9:20:17: Pump Pump1 changed from open to closed

 10:00:00: Pump Pump1 changed by timer control
 10:00:00: Balancing the network:
           Trial  1: relative flow change = 0.000405
           Trial  2: relative flow change = 0.000000
 10:00:00: Balanced after 2 trials

 11:00:00: TCV Valve changed by timer control
 11:00:00: Balancing the network:
           Trial  1: relative flow change = 0.296600
           Trial  2: relative flow change = 0.249399
           Pump Pump1 switched from open to closed
           Trial  3: relative flow change = 0.077858
           Trial  4: relative flow change = 0.012514
           Pump Pump1 switched from closed to open
           Trial  5: relative flow change = 0.016513
           Trial  6: relative flow change = 0.000144
           Trial  7: relative flow change = 0.000001
 11:00:00: Balanced after 7 trials
 11:00:00: Tank Tank is emptying at 2.50 m
 11:00:00: Pump Pump1 changed from closed to open
 11:00:00: TCV Valve changed from closed to open

 11:00:06: Pump Pump2 changed by timer control
 11:00:06: Balancing the network:
           Trial  1: relative flow change = 0.974755
           Trial  2: relative flow change = 0.213503
           Trial  3: relative flow change = 0.014159
           Trial  4: relative flow change = 0.000212
           Trial  5: relative flow change = 0.000001
 11:00:06: Balanced after 5 trials
 11:00:06: Pump Pump2 changed from open to closed

 12:00:00: Balancing the network:
           Trial  1: relative flow change = 0.407275
           Trial  2: relative flow change = 0.053837
           Trial  3: relative flow change = 0.003582
           Trial  4: relative flow change = 0.000015
 12:00:00: Balanced after 4 trials

 13:00:00: Balancing the network:
           Trial  1: relative flow change = 0.088382
           Trial  2: relative flow change = 0.008916
           Trial  3: relative flow change = 0.000115
           Trial  4: relative flow change = 0.000001
 13:00:00: Balanced after 4 trials

 14:00:00: TCV Valve changed by timer control
 14:00:00: Balancing the network:
           Trial  1: relative flow change = 0.517470
           Trial  2: relative flow change = 0.124726
           Trial  3: relative flow change = 0.016679
           Trial  4: relative flow change = 0.000131
           Trial  5: relative flow change = 0.000001
 14:00:00: Balanced after 5 trials
 14:00:00: TCV Valve changed from open to closed

 15:00:00: Balancing the network:
           Trial  1: relative flow change = 0.495189
           Trial  2: relative flow change = 0.177760
           Trial  3: relative flow change = 0.037237
                                             Page 3
```

Figure B.3: Report file (page 3) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



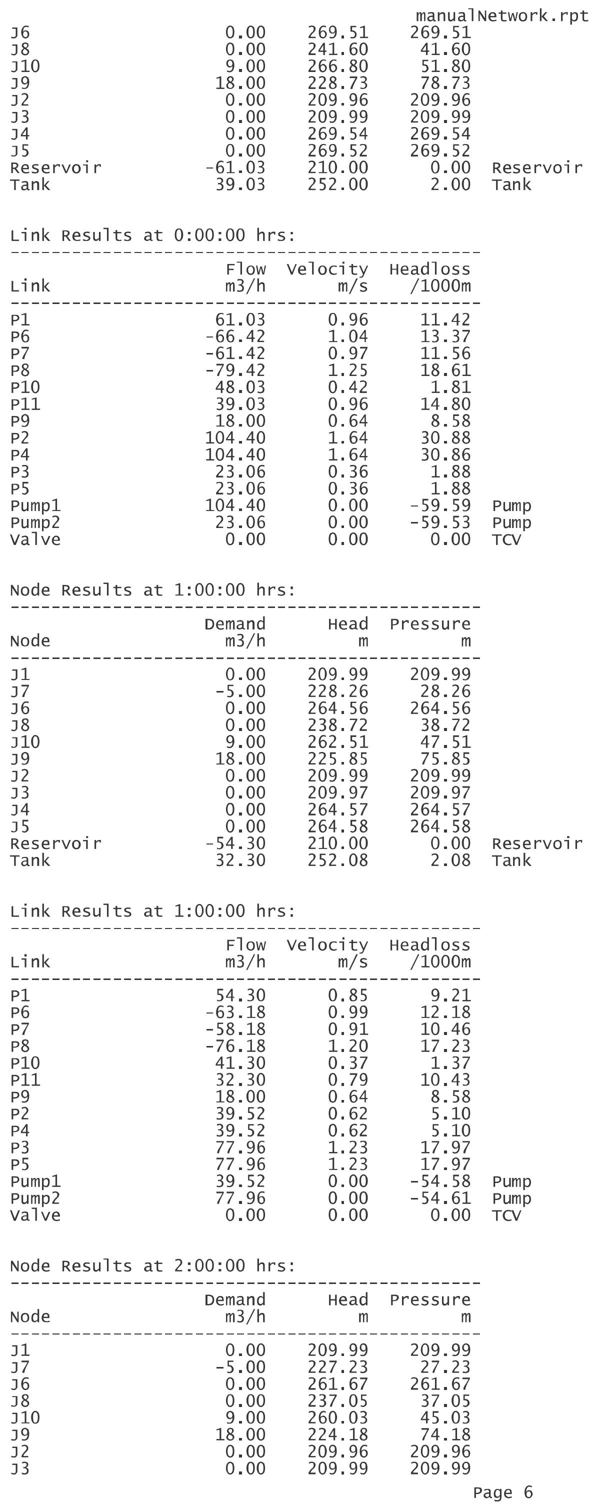
```
                                        manualNetwork.rpt
           Trial  4: relative flow change = 0.002910
           Trial  5: relative flow change = 0.000018
15:00:00: Balanced after 5 trials
15:00:00: Tank Tank is filling at 1.90 m

16:00:00: Pump Pump2 changed by timer control
16:00:00: Balancing the network:
           Trial  1: relative flow change = 1.047197
           Trial  2: relative flow change = 0.324123
           Trial  3: relative flow change = 0.094799
           Trial  4: relative flow change = 0.020867
           Pump Pump1 switched from open to closed
           Trial  5: relative flow change = 0.004559
           Trial  6: relative flow change = 0.000015
           Pump Pump1 switched from closed to open
           Trial  7: relative flow change = 0.002769
           Trial  8: relative flow change = 0.000009
16:00:00: Balanced after 8 trials
16:00:00: Pump Pump2 changed from closed to open

17:00:00: Balancing the network:
           Trial  1: relative flow change = 1.040839
           Trial  2: relative flow change = 0.560260
           Trial  3: relative flow change = 0.183245
           Trial  4: relative flow change = 0.021458
           Trial  5: relative flow change = 0.000712
           Trial  6: relative flow change = 0.000001
17:00:00: Balanced after 6 trials

17:00:32: Pump Pump1 changed by timer control
17:00:32: Balancing the network:
           Trial  1: relative flow change = 1.015761
           Trial  2: relative flow change = 0.195502
           Trial  3: relative flow change = 0.035272
           Trial  4: relative flow change = 0.002582
           Trial  5: relative flow change = 0.000016
17:00:32: Balanced after 5 trials
17:00:32: Pump Pump1 changed from open to closed

18:00:00: TCV Valve changed by timer control
18:00:00: Balancing the network:
           Trial  1: relative flow change = 0.845650
           Trial  2: relative flow change = 0.355587
           Trial  3: relative flow change = 0.082119
           Trial  4: relative flow change = 0.009453
           Trial  5: relative flow change = 0.000274
           Trial  6: relative flow change = 0.000002
18:00:00: Balanced after 6 trials
18:00:00: Pump Pump1 changed from closed to open
18:00:00: TCV Valve changed from closed to open

18:17:00: Pump Pump1 changed by timer control
18:17:00: Balancing the network:
           Trial  1: relative flow change = 0.641004
           Trial  2: relative flow change = 0.094526
           Trial  3: relative flow change = 0.012013
           Trial  4: relative flow change = 0.000227
           Trial  5: relative flow change = 0.000000
18:17:00: Balanced after 5 trials
18:17:00: Tank Tank is emptying at 2.04 m
18:17:00: Pump Pump1 changed from open to closed

18:59:39: Pump Pump2 changed by timer control
18:59:39: Balancing the network:
           Trial  1: relative flow change = 1.013982
           Trial  2: relative flow change = 0.248893
           Trial  3: relative flow change = 0.084649
           Trial  4: relative flow change = 0.012121
           Trial  5: relative flow change = 0.000257
           Trial  6: relative flow change = 0.000001
18:59:39: Balanced after 6 trials
18:59:39: Pump Pump1 changed from closed to open
18:59:39: Pump Pump2 changed from open to closed

19:00:00: Pump Pump2 changed by timer control
19:00:00: Balancing the network:
           Trial  1: relative flow change = 0.609997
           Trial  2: relative flow change = 0.140125
           Trial  3: relative flow change = 0.010143
           Trial  4: relative flow change = 0.000144
           Trial  5: relative flow change = 0.000001
19:00:00: Balanced after 5 trials
19:00:00: Tank Tank is filling at 2.00 m
19:00:00: Pump Pump2 changed from closed to open

19:48:29: Pump Pump1 changed by timer control
                                             Page 4
```

Figure B.4: Report file (page 4) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



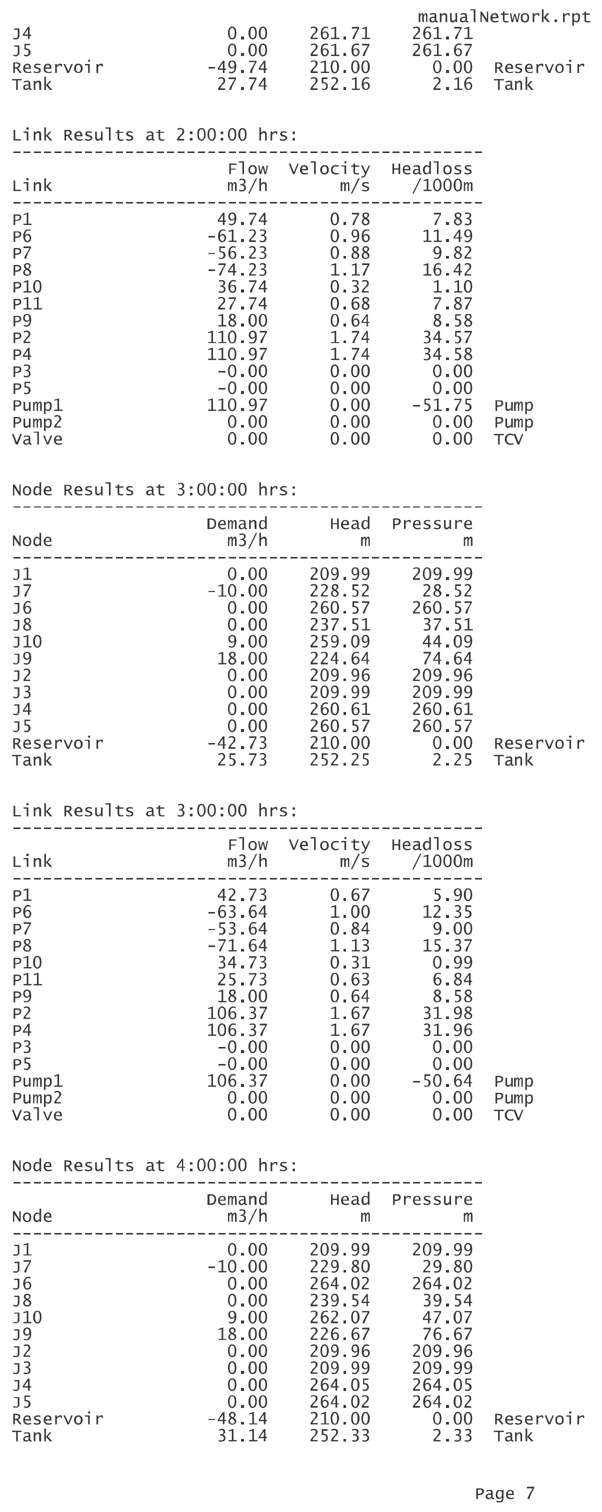
```
                                         manualNetwork.rpt
  19:48:29: Balancing the network:
            Trial  1: relative flow change = 0.614641
            Trial  2: relative flow change = 0.047484
            Trial  3: relative flow change = 0.001899
            Trial  4: relative flow change = 0.000006
  19:48:29: Balanced after 4 trials
  19:48:29: Tank Tank is emptying at 2.02 m
  19:48:29: Pump Pump1 changed from open to closed

  20:00:00: Balancing the network:
            Trial  1: relative flow change = 0.010185
            Trial  2: relative flow change = 0.000052
  20:00:00: Balanced after 2 trials
  20:00:00: Pump Pump1 changed from closed to open

  20:22:37: Pump Pump1 changed by timer control
  20:22:37: Balancing the network:
            Trial  1: relative flow change = 0.002118
            Trial  2: relative flow change = 0.000001
  20:22:37: Balanced after 2 trials
  20:22:37: Pump Pump1 changed from open to closed

  21:00:00: TCV Valve changed by timer control
  21:00:00: Balancing the network:
            Trial  1: relative flow change = 1.132883
            Trial  2: relative flow change = 0.380686
            Trial  3: relative flow change = 0.081688
            Trial  4: relative flow change = 0.002521
            Trial  5: relative flow change = 0.000004
  21:00:00: Balanced after 5 trials
  21:00:00: Tank Tank is filling at 1.94 m
  21:00:00: Pump Pump1 changed from closed to open
  21:00:00: TCV Valve changed from open to closed

  22:00:00: Pump Pump2 changed by timer control
  22:00:00: Balancing the network:
            Trial  1: relative flow change = 0.144805
            Trial  2: relative flow change = 0.003725
            Trial  3: relative flow change = 0.000012
  22:00:00: Balanced after 3 trials
  22:00:00: Pump Pump2 changed from open to closed

  23:00:00: Pump Pump1 changed by timer control
  23:00:00: Balancing the network:
            Trial  1: relative flow change = 3.744208
            Trial  2: relative flow change = 0.095690
            Trial  3: relative flow change = 0.003868
            Trial  4: relative flow change = 0.000007
  23:00:00: Balanced after 4 trials
  23:00:00: Reservoir Reservoir is filling
  23:00:00: Tank Tank is emptying at 2.16 m
  23:00:00: Pump Pump1 changed from open to closed

  24:00:00: Balancing the network:
            Trial  1: relative flow change = 1.012574
            Trial  2: relative flow change = 0.398720
            Trial  3: relative flow change = 0.198815
            Trial  4: relative flow change = 0.040138
            Trial  5: relative flow change = 0.001942
            Trial  6: relative flow change = 0.000005
  24:00:00: Balanced after 6 trials
  24:00:00: Reservoir Reservoir is emptying
  24:00:00: Tank Tank is filling at 2.00 m
  24:00:00: Pump Pump1 changed from closed to open


Energy Usage:
----------------------------------------------------------------
           Usage   Avg.     Kw-hr      Avg.      Peak      Cost
Pump      Factor Effic.       /m3        Kw        Kw      /day
----------------------------------------------------------------
Pump1      61.72  70.01      0.17     14.95     35.01      3.72
Pump2      56.58  72.40      0.18     18.62     22.00      4.55
----------------------------------------------------------------
                                       Demand Charge:      0.00
                                       Total Cost:         8.26

Node Results at 0:00:00 hrs:
----------------------------------------------
                      Demand      Head  Pressure
Node                    m3/h         m         m
----------------------------------------------
J1                      0.00    209.99    209.99
J7                     -5.00    230.04     30.04
                                                 Page 5
```

Figure B.5: Report file (page 5) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



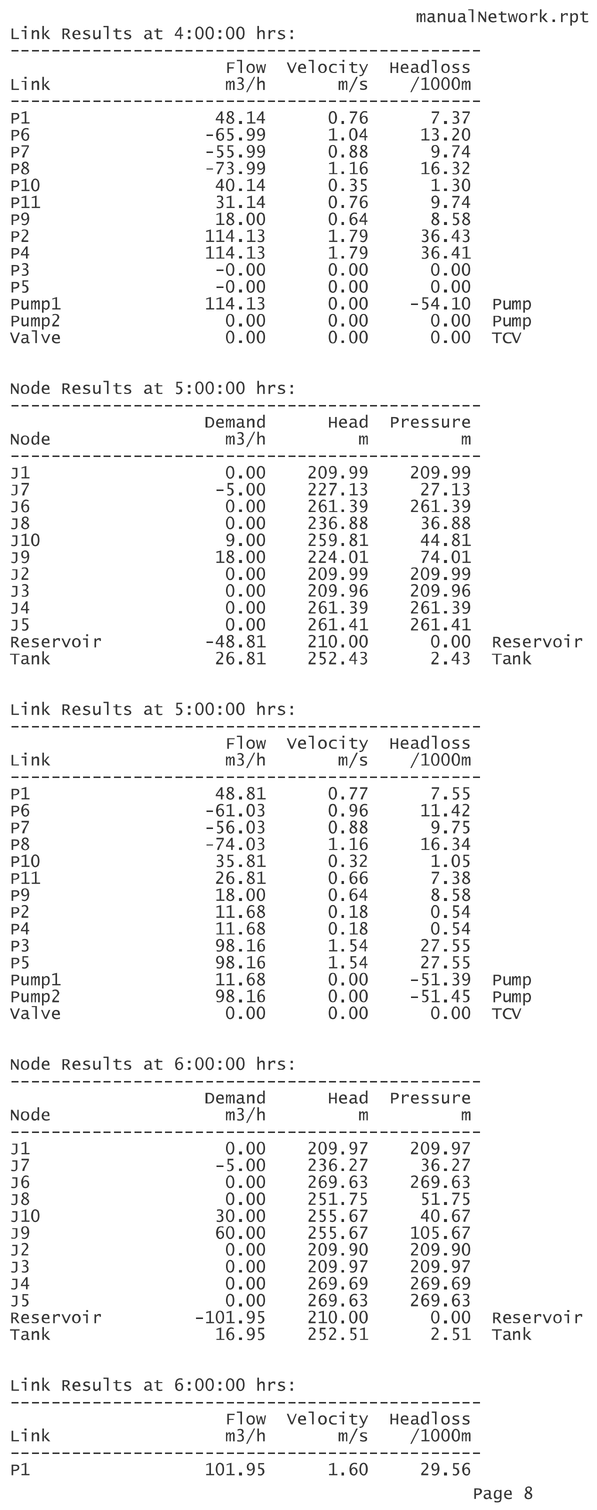
```
                                       manualNetwork.rpt
J6                   0.00    269.51    269.51
J8                   0.00    241.60     41.60
J10                  9.00    266.80     51.80
J9                  18.00    228.73     78.73
J2                   0.00    209.96    209.96
J3                   0.00    209.99    209.99
J4                   0.00    269.54    269.54
J5                   0.00    269.52    269.52
Reservoir          -61.03    210.00      0.00  Reservoir
Tank                39.03    252.00      2.00  Tank

Link Results at 0:00:00 hrs:
----------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
----------------------------------------------
P1                  61.03      0.96     11.42
P6                 -66.42      1.04     13.37
P7                 -61.42      0.97     11.56
P8                 -79.42      1.25     18.61
P10                 48.03      0.42      1.81
P11                 39.03      0.96     14.80
P9                  18.00      0.64      8.58
P2                 104.40      1.64     30.88
P4                 104.40      1.64     30.86
P3                  23.06      0.36      1.88
P5                  23.06      0.36      1.88
Pump1              104.40      0.00    -59.59  Pump
Pump2               23.06      0.00    -59.53  Pump
Valve                0.00      0.00      0.00  TCV

Node Results at 1:00:00 hrs:
----------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
----------------------------------------------
J1                   0.00    209.99    209.99
J7                  -5.00    228.26     28.26
J6                   0.00    264.56    264.56
J8                   0.00    238.72     38.72
J10                  9.00    262.51     47.51
J9                  18.00    225.85     75.85
J2                   0.00    209.99    209.99
J3                   0.00    209.97    209.97
J4                   0.00    264.57    264.57
J5                   0.00    264.58    264.58
Reservoir          -54.30    210.00      0.00  Reservoir
Tank                32.30    252.08      2.08  Tank

Link Results at 1:00:00 hrs:
----------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
----------------------------------------------
P1                  54.30      0.85      9.21
P6                 -63.18      0.99     12.18
P7                 -58.18      0.91     10.46
P8                 -76.18      1.20     17.23
P10                 41.30      0.37      1.37
P11                 32.30      0.79     10.43
P9                  18.00      0.64      8.58
P2                  39.52      0.62      5.10
P4                  39.52      0.62      5.10
P3                  77.96      1.23     17.97
P5                  77.96      1.23     17.97
Pump1               39.52      0.00    -54.58  Pump
Pump2               77.96      0.00    -54.61  Pump
Valve                0.00      0.00      0.00  TCV

Node Results at 2:00:00 hrs:
----------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
----------------------------------------------
J1                   0.00    209.99    209.99
J7                  -5.00    227.23     27.23
J6                   0.00    261.67    261.67
J8                   0.00    237.05     37.05
J10                  9.00    260.03     45.03
J9                  18.00    224.18     74.18
J2                   0.00    209.96    209.96
J3                   0.00    209.99    209.99
                                            Page 6
```

Figure B.6: Report file (page 6) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



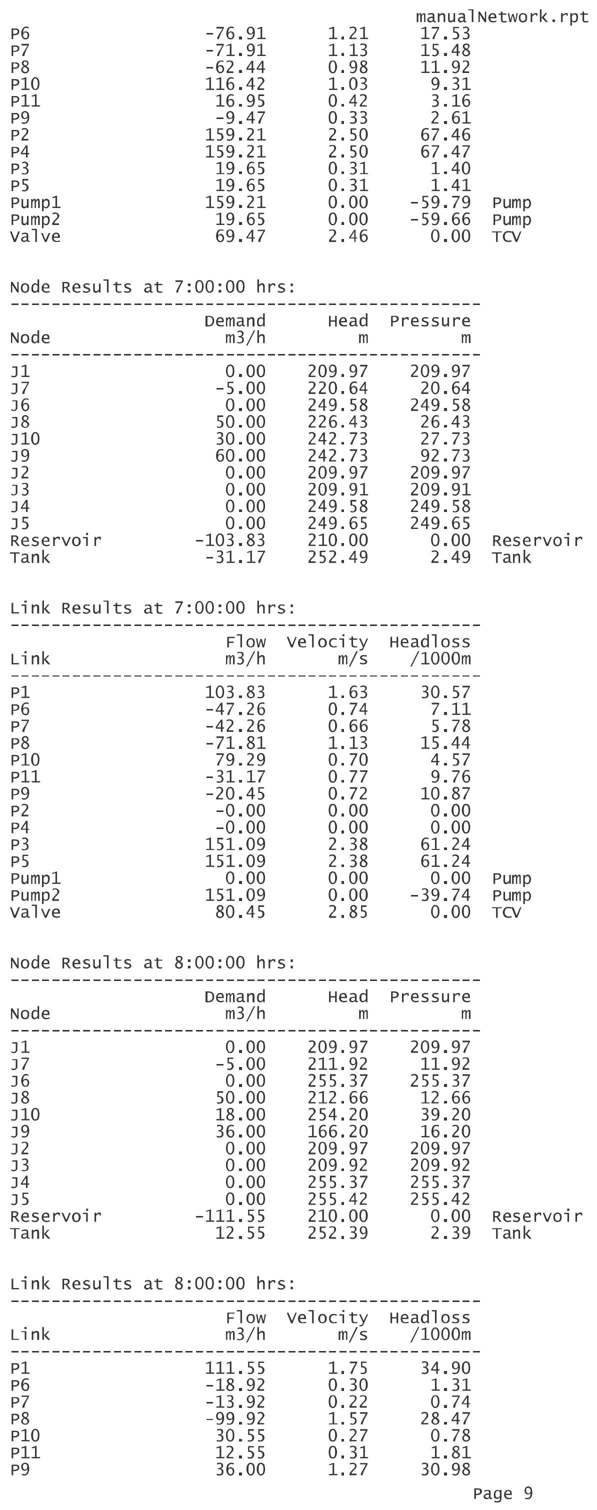
```
                                        manualNetwork.rpt
J4                    0.00    261.71    261.71
J5                    0.00    261.67    261.67
Reservoir           -49.74    210.00      0.00  Reservoir
Tank                 27.74    252.16      2.16  Tank

Link Results at 2:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   49.74      0.78      7.83
P6                  -61.23      0.96     11.49
P7                  -56.23      0.88      9.82
P8                  -74.23      1.17     16.42
P10                  36.74      0.32      1.10
P11                  27.74      0.68      7.87
P9                   18.00      0.64      8.58
P2                  110.97      1.74     34.57
P4                  110.97      1.74     34.58
P3                   -0.00      0.00      0.00
P5                   -0.00      0.00      0.00
Pump1               110.97      0.00    -51.75  Pump
Pump2                 0.00      0.00      0.00  Pump
Valve                 0.00      0.00      0.00  TCV

Node Results at 3:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.99    209.99
J7                  -10.00    228.52     28.52
J6                    0.00    260.57    260.57
J8                    0.00    237.51     37.51
J10                   9.00    259.09     44.09
J9                   18.00    224.64     74.64
J2                    0.00    209.96    209.96
J3                    0.00    209.99    209.99
J4                    0.00    260.61    260.61
J5                    0.00    260.57    260.57
Reservoir           -42.73    210.00      0.00  Reservoir
Tank                 25.73    252.25      2.25  Tank

Link Results at 3:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   42.73      0.67      5.90
P6                  -63.64      1.00     12.35
P7                  -53.64      0.84      9.00
P8                  -71.64      1.13     15.37
P10                  34.73      0.31      0.99
P11                  25.73      0.63      6.84
P9                   18.00      0.64      8.58
P2                  106.37      1.67     31.98
P4                  106.37      1.67     31.96
P3                   -0.00      0.00      0.00
P5                   -0.00      0.00      0.00
Pump1               106.37      0.00    -50.64  Pump
Pump2                 0.00      0.00      0.00  Pump
Valve                 0.00      0.00      0.00  TCV

Node Results at 4:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.99    209.99
J7                  -10.00    229.80     29.80
J6                    0.00    264.02    264.02
J8                    0.00    239.54     39.54
J10                   9.00    262.07     47.07
J9                   18.00    226.67     76.67
J2                    0.00    209.96    209.96
J3                    0.00    209.99    209.99
J4                    0.00    264.05    264.05
J5                    0.00    264.02    264.02
Reservoir           -48.14    210.00      0.00  Reservoir
Tank                 31.14    252.33      2.33  Tank

                                          Page 7
```

Figure B.7: Report file (page 7) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



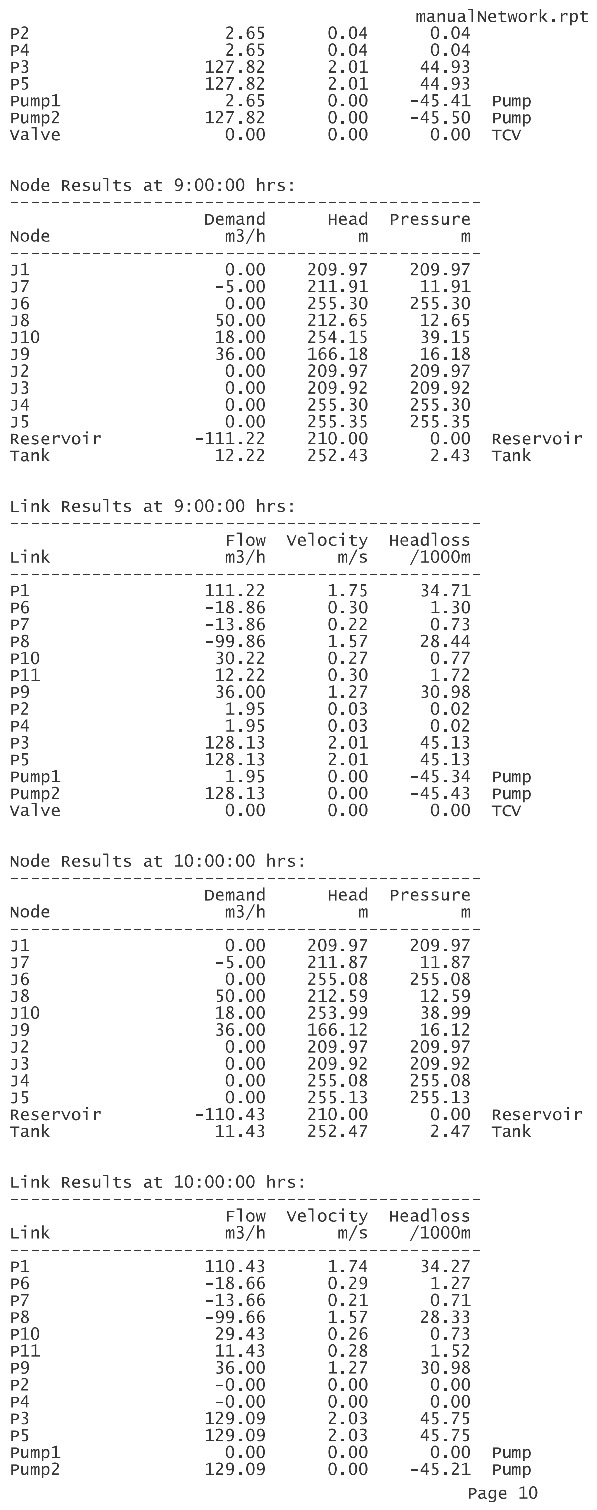
```
                                                  manualNetwork.rpt
Link Results at 4:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   48.14      0.76      7.37
P6                  -65.99      1.04     13.20
P7                  -55.99      0.88      9.74
P8                  -73.99      1.16     16.32
P10                  40.14      0.35      1.30
P11                  31.14      0.76      9.74
P9                   18.00      0.64      8.58
P2                  114.13      1.79     36.43
P4                  114.13      1.79     36.41
P3                   -0.00      0.00      0.00
P5                   -0.00      0.00      0.00
Pump1               114.13      0.00    -54.10  Pump
Pump2                 0.00      0.00      0.00  Pump
Valve                 0.00      0.00      0.00  TCV

Node Results at 5:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.99    209.99
J7                   -5.00    227.13     27.13
J6                    0.00    261.39    261.39
J8                    0.00    236.88     36.88
J10                   9.00    259.81     44.81
J9                   18.00    224.01     74.01
J2                    0.00    209.99    209.99
J3                    0.00    209.96    209.96
J4                    0.00    261.39    261.39
J5                    0.00    261.41    261.41
Reservoir           -48.81    210.00      0.00  Reservoir
Tank                 26.81    252.43      2.43  Tank

Link Results at 5:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   48.81      0.77      7.55
P6                  -61.03      0.96     11.42
P7                  -56.03      0.88      9.75
P8                  -74.03      1.16     16.34
P10                  35.81      0.32      1.05
P11                  26.81      0.66      7.38
P9                   18.00      0.64      8.58
P2                   11.68      0.18      0.54
P4                   11.68      0.18      0.54
P3                   98.16      1.54     27.55
P5                   98.16      1.54     27.55
Pump1                11.68      0.00    -51.39  Pump
Pump2                98.16      0.00    -51.45  Pump
Valve                 0.00      0.00      0.00  TCV

Node Results at 6:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.97    209.97
J7                   -5.00    236.27     36.27
J6                    0.00    269.63    269.63
J8                    0.00    251.75     51.75
J10                  30.00    255.67     40.67
J9                   60.00    255.67    105.67
J2                    0.00    209.90    209.90
J3                    0.00    209.97    209.97
J4                    0.00    269.69    269.69
J5                    0.00    269.63    269.63
Reservoir          -101.95    210.00      0.00  Reservoir
Tank                 16.95    252.51      2.51  Tank

Link Results at 6:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                  101.95      1.60     29.56
                                           Page 8
```

Figure B.8: Report file (page 8) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



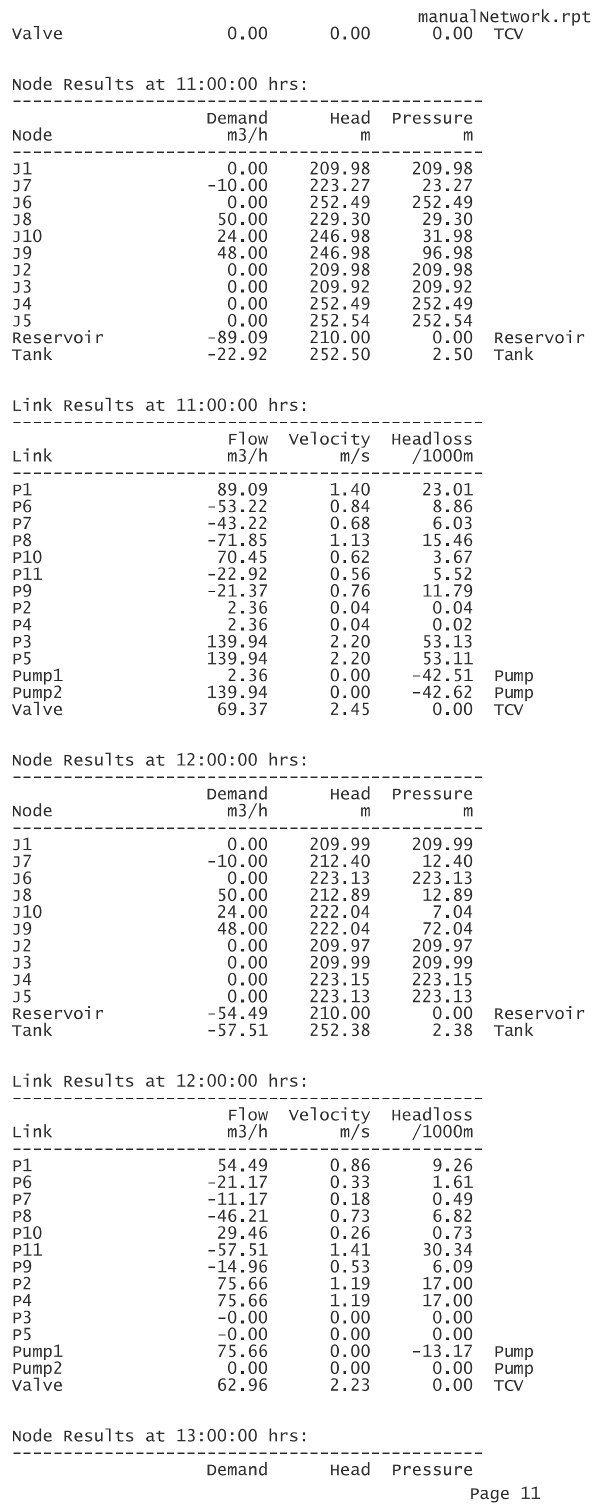
```
                                          manualNetwork.rpt
P6                  -76.91       1.21     17.53
P7                  -71.91       1.13     15.48
P8                  -62.44       0.98     11.92
P10                 116.42       1.03      9.31
P11                  16.95       0.42      3.16
P9                   -9.47       0.33      2.61
P2                  159.21       2.50     67.46
P4                  159.21       2.50     67.47
P3                   19.65       0.31      1.40
P5                   19.65       0.31      1.41
Pump1               159.21       0.00    -59.79  Pump
Pump2                19.65       0.00    -59.66  Pump
Valve                69.47       2.46      0.00  TCV

Node Results at 7:00:00 hrs:
----------------------------------------------
                    Demand       Head  Pressure
Node                  m3/h          m         m
----------------------------------------------
J1                    0.00     209.97    209.97
J7                   -5.00     220.64     20.64
J6                    0.00     249.58    249.58
J8                   50.00     226.43     26.43
J10                  30.00     242.73     27.73
J9                   60.00     242.73     92.73
J2                    0.00     209.97    209.97
J3                    0.00     209.91    209.91
J4                    0.00     249.58    249.58
J5                    0.00     249.65    249.65
Reservoir          -103.83     210.00      0.00  Reservoir
Tank                -31.17     252.49      2.49  Tank

Link Results at 7:00:00 hrs:
----------------------------------------------
                      Flow   Velocity  Headloss
Link                  m3/h        m/s    /1000m
----------------------------------------------
P1                  103.83       1.63     30.57
P6                  -47.26       0.74      7.11
P7                  -42.26       0.66      5.78
P8                  -71.81       1.13     15.44
P10                  79.29       0.70      4.57
P11                 -31.17       0.77      9.76
P9                  -20.45       0.72     10.87
P2                   -0.00       0.00      0.00
P4                   -0.00       0.00      0.00
P3                  151.09       2.38     61.24
P5                  151.09       2.38     61.24
Pump1                 0.00       0.00      0.00  Pump
Pump2               151.09       0.00    -39.74  Pump
Valve                80.45       2.85      0.00  TCV

Node Results at 8:00:00 hrs:
----------------------------------------------
                    Demand       Head  Pressure
Node                  m3/h          m         m
----------------------------------------------
J1                    0.00     209.97    209.97
J7                   -5.00     211.92     11.92
J6                    0.00     255.37    255.37
J8                   50.00     212.66     12.66
J10                  18.00     254.20     39.20
J9                   36.00     166.20     16.20
J2                    0.00     209.97    209.97
J3                    0.00     209.92    209.92
J4                    0.00     255.37    255.37
J5                    0.00     255.42    255.42
Reservoir          -111.55     210.00      0.00  Reservoir
Tank                 12.55     252.39      2.39  Tank

Link Results at 8:00:00 hrs:
----------------------------------------------
                      Flow   Velocity  Headloss
Link                  m3/h        m/s    /1000m
----------------------------------------------
P1                  111.55       1.75     34.90
P6                  -18.92       0.30      1.31
P7                  -13.92       0.22      0.74
P8                  -99.92       1.57     28.47
P10                  30.55       0.27      0.78
P11                  12.55       0.31      1.81
P9                   36.00       1.27     30.98
                                                 Page 9
```

Figure B.9: Report file (page 9) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



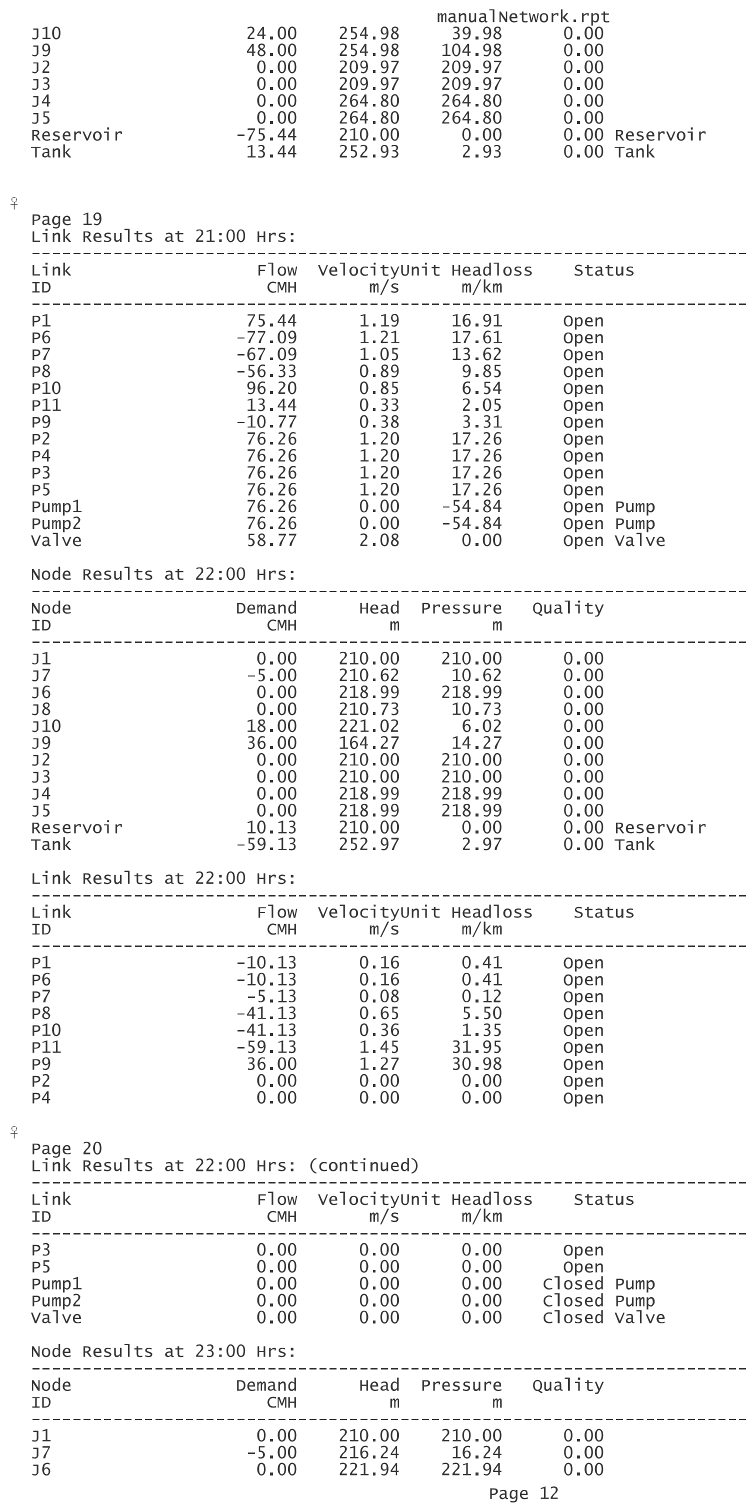
```
                                       manualNetwork.rpt
P2                  2.65      0.04      0.04
P4                  2.65      0.04      0.04
P3                127.82      2.01     44.93
P5                127.82      2.01     44.93
Pump1               2.65      0.00    -45.41  Pump
Pump2             127.82      0.00    -45.50  Pump
Valve               0.00      0.00      0.00  TCV

Node Results at 9:00:00 hrs:
----------------------------------------------
                  Demand      Head  Pressure
Node                m3/h         m         m
----------------------------------------------
J1                  0.00    209.97    209.97
J7                 -5.00    211.91     11.91
J6                  0.00    255.30    255.30
J8                 50.00    212.65     12.65
J10                18.00    254.15     39.15
J9                 36.00    166.18     16.18
J2                  0.00    209.97    209.97
J3                  0.00    209.92    209.92
J4                  0.00    255.30    255.30
J5                  0.00    255.35    255.35
Reservoir        -111.22    210.00      0.00  Reservoir
Tank               12.22    252.43      2.43  Tank

Link Results at 9:00:00 hrs:
----------------------------------------------
                    Flow  Velocity  Headloss
Link                m3/h       m/s    /1000m
----------------------------------------------
P1                111.22      1.75     34.71
P6                -18.86      0.30      1.30
P7                -13.86      0.22      0.73
P8                -99.86      1.57     28.44
P10                30.22      0.27      0.77
P11                12.22      0.30      1.72
P9                 36.00      1.27     30.98
P2                  1.95      0.03      0.02
P4                  1.95      0.03      0.02
P3                128.13      2.01     45.13
P5                128.13      2.01     45.13
Pump1               1.95      0.00    -45.34  Pump
Pump2             128.13      0.00    -45.43  Pump
Valve               0.00      0.00      0.00  TCV

Node Results at 10:00:00 hrs:
----------------------------------------------
                  Demand      Head  Pressure
Node                m3/h         m         m
----------------------------------------------
J1                  0.00    209.97    209.97
J7                 -5.00    211.87     11.87
J6                  0.00    255.08    255.08
J8                 50.00    212.59     12.59
J10                18.00    253.99     38.99
J9                 36.00    166.12     16.12
J2                  0.00    209.97    209.97
J3                  0.00    209.92    209.92
J4                  0.00    255.08    255.08
J5                  0.00    255.13    255.13
Reservoir        -110.43    210.00      0.00  Reservoir
Tank               11.43    252.47      2.47  Tank

Link Results at 10:00:00 hrs:
----------------------------------------------
                    Flow  Velocity  Headloss
Link                m3/h       m/s    /1000m
----------------------------------------------
P1                110.43      1.74     34.27
P6                -18.66      0.29      1.27
P7                -13.66      0.21      0.71
P8                -99.66      1.57     28.33
P10                29.43      0.26      0.73
P11                11.43      0.28      1.52
P9                 36.00      1.27     30.98
P2                 -0.00      0.00      0.00
P4                 -0.00      0.00      0.00
P3                129.09      2.03     45.75
P5                129.09      2.03     45.75
Pump1               0.00      0.00      0.00  Pump
Pump2             129.09      0.00    -45.21  Pump
                                            Page 10
```

Figure B.10: Report file (page 10) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



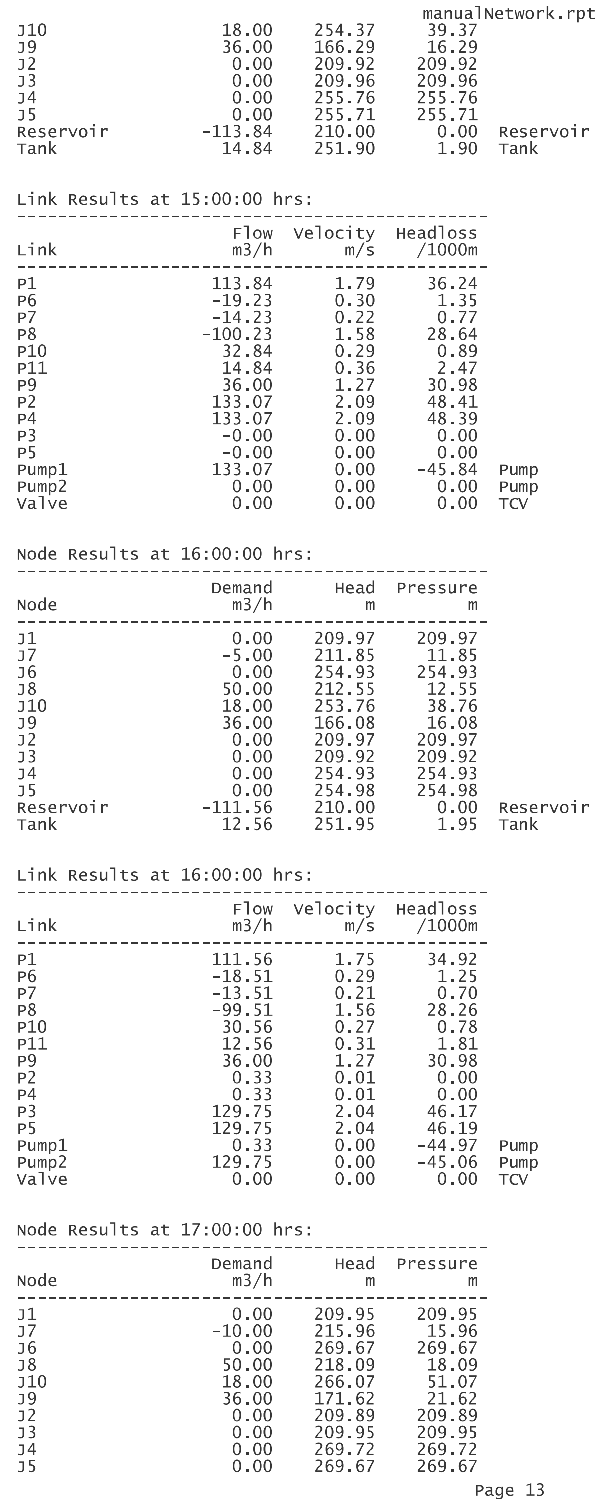
```
                                                  manualNetwork.rpt
Valve                 0.00      0.00      0.00  TCV

Node Results at 11:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.98    209.98
J7                  -10.00    223.27     23.27
J6                    0.00    252.49    252.49
J8                   50.00    229.30     29.30
J10                  24.00    246.98     31.98
J9                   48.00    246.98     96.98
J2                    0.00    209.98    209.98
J3                    0.00    209.92    209.92
J4                    0.00    252.49    252.49
J5                    0.00    252.54    252.54
Reservoir           -89.09    210.00      0.00  Reservoir
Tank                -22.92    252.50      2.50  Tank

Link Results at 11:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   89.09      1.40     23.01
P6                  -53.22      0.84      8.86
P7                  -43.22      0.68      6.03
P8                  -71.85      1.13     15.46
P10                  70.45      0.62      3.67
P11                 -22.92      0.56      5.52
P9                  -21.37      0.76     11.79
P2                    2.36      0.04      0.04
P4                    2.36      0.04      0.02
P3                  139.94      2.20     53.13
P5                  139.94      2.20     53.11
Pump1                 2.36      0.00    -42.51  Pump
Pump2               139.94      0.00    -42.62  Pump
Valve                69.37      2.45      0.00  TCV

Node Results at 12:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.99    209.99
J7                  -10.00    212.40     12.40
J6                    0.00    223.13    223.13
J8                   50.00    212.89     12.89
J10                  24.00    222.04      7.04
J9                   48.00    222.04     72.04
J2                    0.00    209.97    209.97
J3                    0.00    209.99    209.99
J4                    0.00    223.15    223.15
J5                    0.00    223.13    223.13
Reservoir           -54.49    210.00      0.00  Reservoir
Tank                -57.51    252.38      2.38  Tank

Link Results at 12:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   54.49      0.86      9.26
P6                  -21.17      0.33      1.61
P7                  -11.17      0.18      0.49
P8                  -46.21      0.73      6.82
P10                  29.46      0.26      0.73
P11                 -57.51      1.41     30.34
P9                  -14.96      0.53      6.09
P2                   75.66      1.19     17.00
P4                   75.66      1.19     17.00
P3                   -0.00      0.00      0.00
P5                   -0.00      0.00      0.00
Pump1                75.66      0.00    -13.17  Pump
Pump2                 0.00      0.00      0.00  Pump
Valve                62.96      2.23      0.00  TCV

Node Results at 13:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
                                               Page 11
```

Figure B.11: Report file (page 11) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



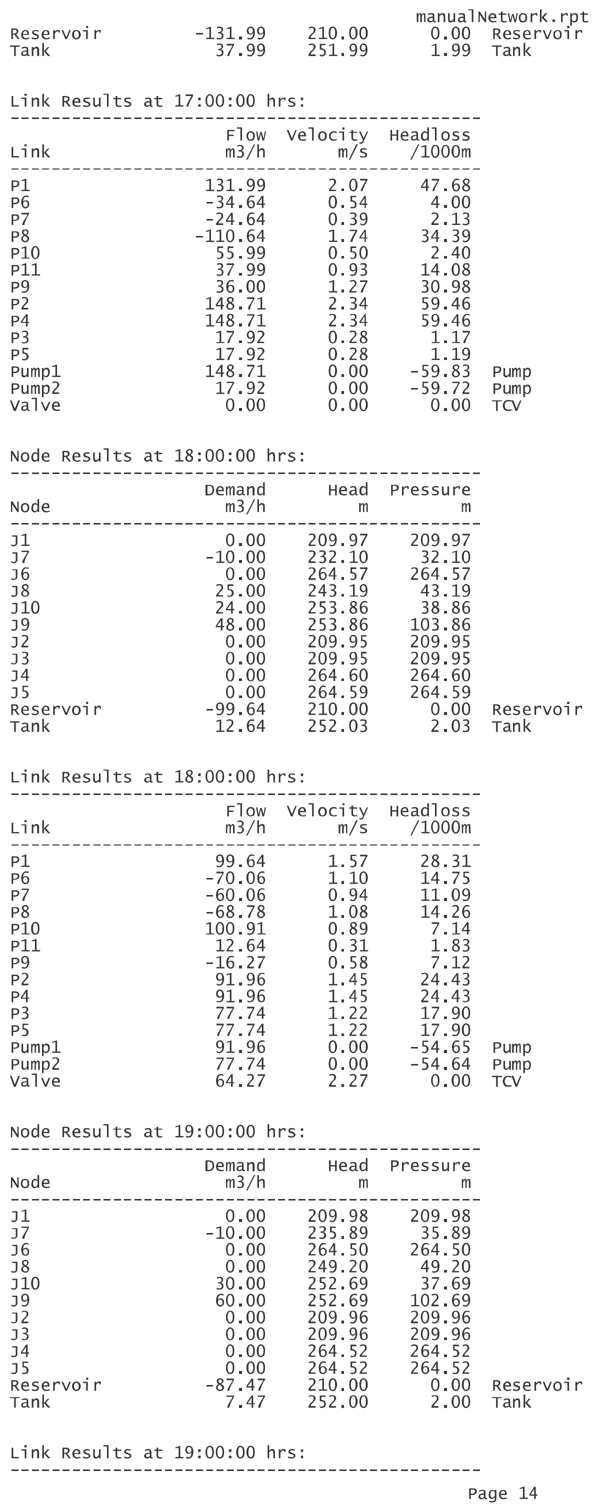
```
                                        manualNetwork.rpt
 J10                  24.00    254.98     39.98      0.00
 J9                   48.00    254.98    104.98      0.00
 J2                    0.00    209.97    209.97      0.00
 J3                    0.00    209.97    209.97      0.00
 J4                    0.00    264.80    264.80      0.00
 J5                    0.00    264.80    264.80      0.00
 Reservoir           -75.44    210.00      0.00      0.00 Reservoir
 Tank                 13.44    252.93      2.93      0.00 Tank


♀
 Page 19
 Link Results at 21:00 Hrs:
 ----------------------------------------------------------------------
 Link                  Flow  VelocityUnit Headloss    Status
 ID                     CMH       m/s      m/km
 ----------------------------------------------------------------------
 P1                   75.44      1.19     16.91      Open
 P6                  -77.09      1.21     17.61      Open
 P7                  -67.09      1.05     13.62      Open
 P8                  -56.33      0.89      9.85      Open
 P10                  96.20      0.85      6.54      Open
 P11                  13.44      0.33      2.05      Open
 P9                  -10.77      0.38      3.31      Open
 P2                   76.26      1.20     17.26      Open
 P4                   76.26      1.20     17.26      Open
 P3                   76.26      1.20     17.26      Open
 P5                   76.26      1.20     17.26      Open
 Pump1                76.26      0.00    -54.84      Open Pump
 Pump2                76.26      0.00    -54.84      Open Pump
 Valve                58.77      2.08      0.00      Open Valve

 Node Results at 22:00 Hrs:
 ----------------------------------------------------------------------
 Node                Demand      Head  Pressure   Quality
 ID                     CMH         m         m
 ----------------------------------------------------------------------
 J1                    0.00    210.00    210.00      0.00
 J7                   -5.00    210.62     10.62      0.00
 J6                    0.00    218.99    218.99      0.00
 J8                    0.00    210.73     10.73      0.00
 J10                  18.00    221.02      6.02      0.00
 J9                   36.00    164.27     14.27      0.00
 J2                    0.00    210.00    210.00      0.00
 J3                    0.00    210.00    210.00      0.00
 J4                    0.00    218.99    218.99      0.00
 J5                    0.00    218.99    218.99      0.00
 Reservoir            10.13    210.00      0.00      0.00 Reservoir
 Tank                -59.13    252.97      2.97      0.00 Tank

 Link Results at 22:00 Hrs:
 ----------------------------------------------------------------------
 Link                  Flow  VelocityUnit Headloss    Status
 ID                     CMH       m/s      m/km
 ----------------------------------------------------------------------
 P1                  -10.13      0.16      0.41      Open
 P6                  -10.13      0.16      0.41      Open
 P7                   -5.13      0.08      0.12      Open
 P8                  -41.13      0.65      5.50      Open
 P10                 -41.13      0.36      1.35      Open
 P11                 -59.13      1.45     31.95      Open
 P9                   36.00      1.27     30.98      Open
 P2                    0.00      0.00      0.00      Open
 P4                    0.00      0.00      0.00      Open

♀
 Page 20
 Link Results at 22:00 Hrs: (continued)
 ----------------------------------------------------------------------
 Link                  Flow  VelocityUnit Headloss    Status
 ID                     CMH       m/s      m/km
 ----------------------------------------------------------------------
 P3                    0.00      0.00      0.00      Open
 P5                    0.00      0.00      0.00      Open
 Pump1                 0.00      0.00      0.00    Closed Pump
 Pump2                 0.00      0.00      0.00    Closed Pump
 Valve                 0.00      0.00      0.00    Closed Valve

 Node Results at 23:00 Hrs:
 ----------------------------------------------------------------------
 Node                Demand      Head  Pressure   Quality
 ID                     CMH         m         m
 ----------------------------------------------------------------------
 J1                    0.00    210.00    210.00      0.00
 J7                   -5.00    216.24     16.24      0.00
 J6                    0.00    221.94    221.94      0.00
                                          Page 12
```

Figure B.12: Report file (page 12) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



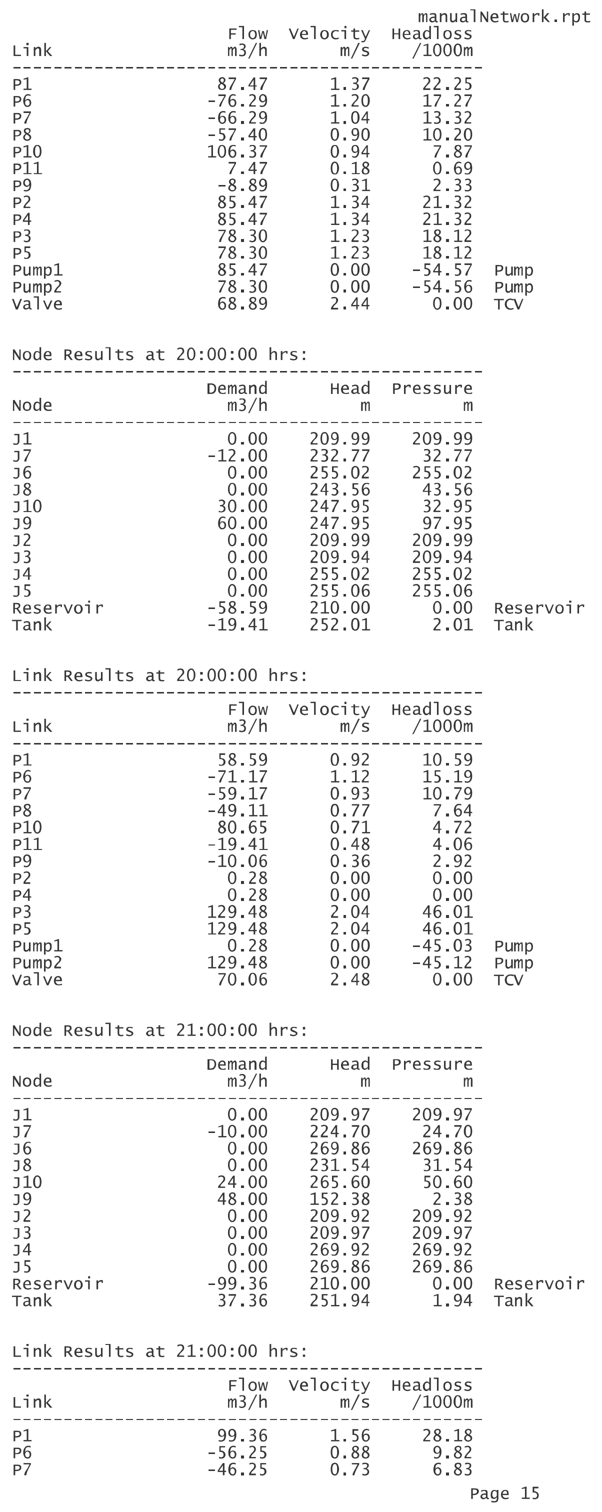
```
                                          manualNetwork.rpt
J10                 18.00    254.37     39.37
J9                  36.00    166.29     16.29
J2                   0.00    209.92    209.92
J3                   0.00    209.96    209.96
J4                   0.00    255.76    255.76
J5                   0.00    255.71    255.71
Reservoir         -113.84    210.00      0.00  Reservoir
Tank                14.84    251.90      1.90  Tank

Link Results at 15:00:00 hrs:
----------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
----------------------------------------------
P1                 113.84      1.79     36.24
P6                 -19.23      0.30      1.35
P7                 -14.23      0.22      0.77
P8                -100.23      1.58     28.64
P10                 32.84      0.29      0.89
P11                 14.84      0.36      2.47
P9                  36.00      1.27     30.98
P2                 133.07      2.09     48.41
P4                 133.07      2.09     48.39
P3                  -0.00      0.00      0.00
P5                  -0.00      0.00      0.00
Pump1              133.07      0.00    -45.84  Pump
Pump2                0.00      0.00      0.00  Pump
Valve                0.00      0.00      0.00  TCV

Node Results at 16:00:00 hrs:
----------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
----------------------------------------------
J1                   0.00    209.97    209.97
J7                  -5.00    211.85     11.85
J6                   0.00    254.93    254.93
J8                  50.00    212.55     12.55
J10                 18.00    253.76     38.76
J9                  36.00    166.08     16.08
J2                   0.00    209.97    209.97
J3                   0.00    209.92    209.92
J4                   0.00    254.93    254.93
J5                   0.00    254.98    254.98
Reservoir         -111.56    210.00      0.00  Reservoir
Tank                12.56    251.95      1.95  Tank

Link Results at 16:00:00 hrs:
----------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
----------------------------------------------
P1                 111.56      1.75     34.92
P6                 -18.51      0.29      1.25
P7                 -13.51      0.21      0.70
P8                 -99.51      1.56     28.26
P10                 30.56      0.27      0.78
P11                 12.56      0.31      1.81
P9                  36.00      1.27     30.98
P2                   0.33      0.01      0.00
P4                   0.33      0.01      0.00
P3                 129.75      2.04     46.17
P5                 129.75      2.04     46.19
Pump1                0.33      0.00    -44.97  Pump
Pump2              129.75      0.00    -45.06  Pump
Valve                0.00      0.00      0.00  TCV

Node Results at 17:00:00 hrs:
----------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
----------------------------------------------
J1                   0.00    209.95    209.95
J7                 -10.00    215.96     15.96
J6                   0.00    269.67    269.67
J8                  50.00    218.09     18.09
J10                 18.00    266.07     51.07
J9                  36.00    171.62     21.62
J2                   0.00    209.89    209.89
J3                   0.00    209.95    209.95
J4                   0.00    269.72    269.72
J5                   0.00    269.67    269.67
                                       Page 13
```

Figure B.13: Report file (page 13) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



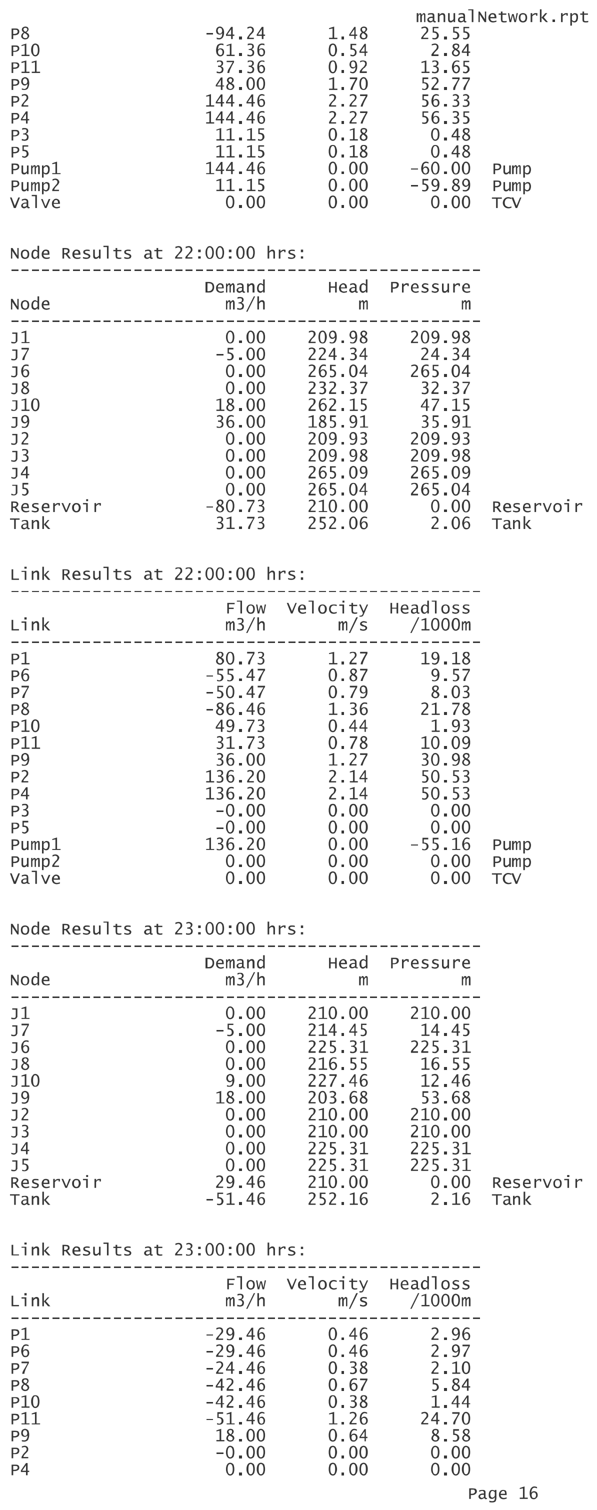
```
                                          manualNetwork.rpt
Reservoir         -131.99    210.00      0.00  Reservoir
Tank                37.99    251.99      1.99  Tank

Link Results at 17:00:00 hrs:
---------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
---------------------------------------------
P1                 131.99      2.07     47.68
P6                 -34.64      0.54      4.00
P7                 -24.64      0.39      2.13
P8                -110.64      1.74     34.39
P10                 55.99      0.50      2.40
P11                 37.99      0.93     14.08
P9                  36.00      1.27     30.98
P2                 148.71      2.34     59.46
P4                 148.71      2.34     59.46
P3                  17.92      0.28      1.17
P5                  17.92      0.28      1.19
Pump1              148.71      0.00    -59.83  Pump
Pump2               17.92      0.00    -59.72  Pump
Valve                0.00      0.00      0.00  TCV

Node Results at 18:00:00 hrs:
---------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
---------------------------------------------
J1                   0.00    209.97    209.97
J7                 -10.00    232.10     32.10
J6                   0.00    264.57    264.57
J8                  25.00    243.19     43.19
J10                 24.00    253.86     38.86
J9                  48.00    253.86    103.86
J2                   0.00    209.95    209.95
J3                   0.00    209.95    209.95
J4                   0.00    264.60    264.60
J5                   0.00    264.59    264.59
Reservoir          -99.64    210.00      0.00  Reservoir
Tank                12.64    252.03      2.03  Tank

Link Results at 18:00:00 hrs:
---------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
---------------------------------------------
P1                  99.64      1.57     28.31
P6                 -70.06      1.10     14.75
P7                 -60.06      0.94     11.09
P8                 -68.78      1.08     14.26
P10                100.91      0.89      7.14
P11                 12.64      0.31      1.83
P9                 -16.27      0.58      7.12
P2                  91.96      1.45     24.43
P4                  91.96      1.45     24.43
P3                  77.74      1.22     17.90
P5                  77.74      1.22     17.90
Pump1               91.96      0.00    -54.65  Pump
Pump2               77.74      0.00    -54.64  Pump
Valve               64.27      2.27      0.00  TCV

Node Results at 19:00:00 hrs:
---------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
---------------------------------------------
J1                   0.00    209.98    209.98
J7                 -10.00    235.89     35.89
J6                   0.00    264.50    264.50
J8                   0.00    249.20     49.20
J10                 30.00    252.69     37.69
J9                  60.00    252.69    102.69
J2                   0.00    209.96    209.96
J3                   0.00    209.96    209.96
J4                   0.00    264.52    264.52
J5                   0.00    264.52    264.52
Reservoir          -87.47    210.00      0.00  Reservoir
Tank                 7.47    252.00      2.00  Tank

Link Results at 19:00:00 hrs:
---------------------------------------------
                                         Page 14
```

Figure B.14: Report file (page 14) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



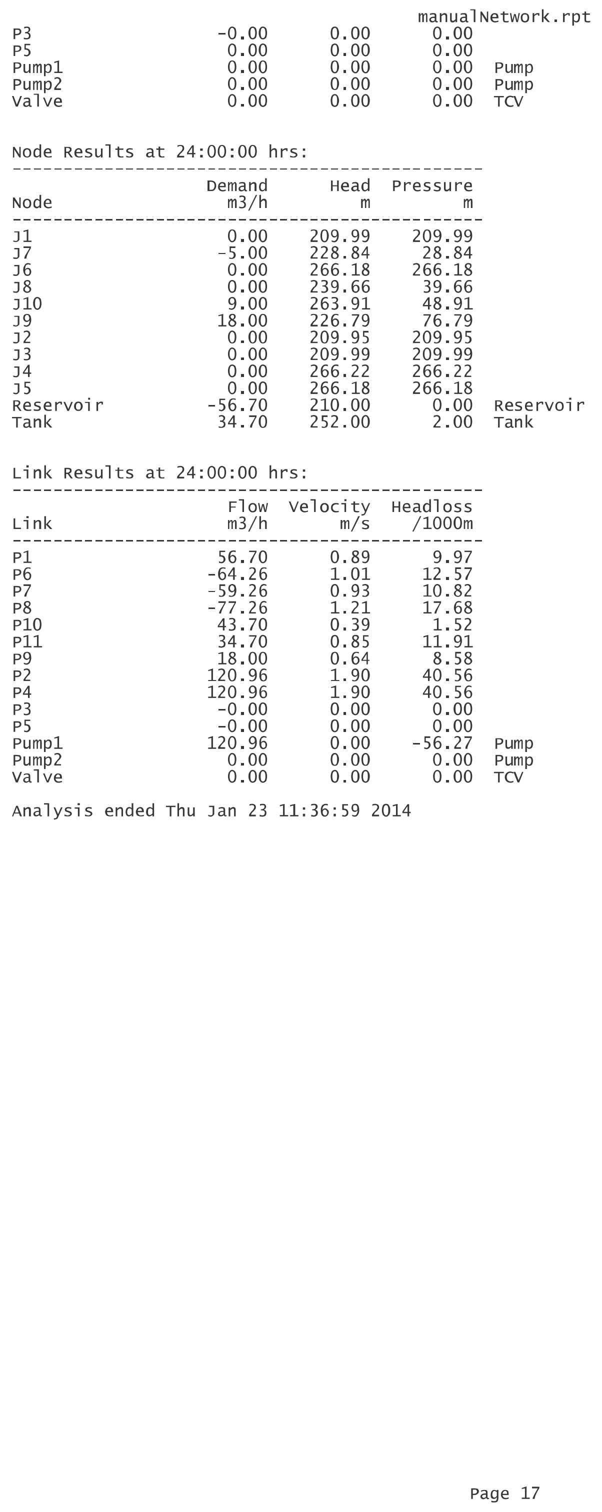
```
                                      manualNetwork.rpt
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
---------------------------------------------
P1                  87.47      1.37     22.25
P6                 -76.29      1.20     17.27
P7                 -66.29      1.04     13.32
P8                 -57.40      0.90     10.20
P10                106.37      0.94      7.87
P11                  7.47      0.18      0.69
P9                  -8.89      0.31      2.33
P2                  85.47      1.34     21.32
P4                  85.47      1.34     21.32
P3                  78.30      1.23     18.12
P5                  78.30      1.23     18.12
Pump1               85.47      0.00    -54.57  Pump
Pump2               78.30      0.00    -54.56  Pump
Valve               68.89      2.44      0.00  TCV

Node Results at 20:00:00 hrs:
---------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
---------------------------------------------
J1                   0.00    209.99    209.99
J7                 -12.00    232.77     32.77
J6                   0.00    255.02    255.02
J8                   0.00    243.56     43.56
J10                 30.00    247.95     32.95
J9                  60.00    247.95     97.95
J2                   0.00    209.99    209.99
J3                   0.00    209.94    209.94
J4                   0.00    255.02    255.02
J5                   0.00    255.06    255.06
Reservoir          -58.59    210.00      0.00  Reservoir
Tank               -19.41    252.01      2.01  Tank

Link Results at 20:00:00 hrs:
---------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
---------------------------------------------
P1                  58.59      0.92     10.59
P6                 -71.17      1.12     15.19
P7                 -59.17      0.93     10.79
P8                 -49.11      0.77      7.64
P10                 80.65      0.71      4.72
P11                -19.41      0.48      4.06
P9                 -10.06      0.36      2.92
P2                   0.28      0.00      0.00
P4                   0.28      0.00      0.00
P3                 129.48      2.04     46.01
P5                 129.48      2.04     46.01
Pump1                0.28      0.00    -45.03  Pump
Pump2              129.48      0.00    -45.12  Pump
Valve               70.06      2.48      0.00  TCV

Node Results at 21:00:00 hrs:
---------------------------------------------
                   Demand      Head  Pressure
Node                 m3/h         m         m
---------------------------------------------
J1                   0.00    209.97    209.97
J7                 -10.00    224.70     24.70
J6                   0.00    269.86    269.86
J8                   0.00    231.54     31.54
J10                 24.00    265.60     50.60
J9                  48.00    152.38      2.38
J2                   0.00    209.92    209.92
J3                   0.00    209.97    209.97
J4                   0.00    269.92    269.92
J5                   0.00    269.86    269.86
Reservoir          -99.36    210.00      0.00  Reservoir
Tank                37.36    251.94      1.94  Tank

Link Results at 21:00:00 hrs:
---------------------------------------------
                     Flow  Velocity  Headloss
Link                 m3/h       m/s    /1000m
---------------------------------------------
P1                  99.36      1.56     28.18
P6                 -56.25      0.88      9.82
P7                 -46.25      0.73      6.83
                                          Page 15
```

Figure B.15: Report file (page 15) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



```
                                      manualNetwork.rpt
P8                  -94.24      1.48     25.55
P10                  61.36      0.54      2.84
P11                  37.36      0.92     13.65
P9                   48.00      1.70     52.77
P2                  144.46      2.27     56.33
P4                  144.46      2.27     56.35
P3                   11.15      0.18      0.48
P5                   11.15      0.18      0.48
Pump1               144.46      0.00    -60.00  Pump
Pump2                11.15      0.00    -59.89  Pump
Valve                 0.00      0.00      0.00  TCV

Node Results at 22:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.98    209.98
J7                   -5.00    224.34     24.34
J6                    0.00    265.04    265.04
J8                    0.00    232.37     32.37
J10                  18.00    262.15     47.15
J9                   36.00    185.91     35.91
J2                    0.00    209.93    209.93
J3                    0.00    209.98    209.98
J4                    0.00    265.09    265.09
J5                    0.00    265.04    265.04
Reservoir           -80.73    210.00      0.00  Reservoir
Tank                 31.73    252.06      2.06  Tank

Link Results at 22:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   80.73      1.27     19.18
P6                  -55.47      0.87      9.57
P7                  -50.47      0.79      8.03
P8                  -86.46      1.36     21.78
P10                  49.73      0.44      1.93
P11                  31.73      0.78     10.09
P9                   36.00      1.27     30.98
P2                  136.20      2.14     50.53
P4                  136.20      2.14     50.53
P3                   -0.00      0.00      0.00
P5                   -0.00      0.00      0.00
Pump1               136.20      0.00    -55.16  Pump
Pump2                 0.00      0.00      0.00  Pump
Valve                 0.00      0.00      0.00  TCV

Node Results at 23:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    210.00    210.00
J7                   -5.00    214.45     14.45
J6                    0.00    225.31    225.31
J8                    0.00    216.55     16.55
J10                   9.00    227.46     12.46
J9                   18.00    203.68     53.68
J2                    0.00    210.00    210.00
J3                    0.00    210.00    210.00
J4                    0.00    225.31    225.31
J5                    0.00    225.31    225.31
Reservoir            29.46    210.00      0.00  Reservoir
Tank                -51.46    252.16      2.16  Tank

Link Results at 23:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                  -29.46      0.46      2.96
P6                  -29.46      0.46      2.97
P7                  -24.46      0.38      2.10
P8                  -42.46      0.67      5.84
P10                 -42.46      0.38      1.44
P11                 -51.46      1.26     24.70
P9                   18.00      0.64      8.58
P2                   -0.00      0.00      0.00
P4                    0.00      0.00      0.00
                                        Page 16
```

Figure B.16: Report file (page 16) containing the network simulation results of the tutorial provided on chapter 2 of this manual.



```
                                        manualNetwork.rpt
P3                   -0.00      0.00      0.00
P5                    0.00      0.00      0.00
Pump1                 0.00      0.00      0.00  Pump
Pump2                 0.00      0.00      0.00  Pump
Valve                 0.00      0.00      0.00  TCV

Node Results at 24:00:00 hrs:
----------------------------------------------
                    Demand      Head  Pressure
Node                  m3/h         m         m
----------------------------------------------
J1                    0.00    209.99    209.99
J7                   -5.00    228.84     28.84
J6                    0.00    266.18    266.18
J8                    0.00    239.66     39.66
J10                   9.00    263.91     48.91
J9                   18.00    226.79     76.79
J2                    0.00    209.95    209.95
J3                    0.00    209.99    209.99
J4                    0.00    266.22    266.22
J5                    0.00    266.18    266.18
Reservoir           -56.70    210.00      0.00  Reservoir
Tank                 34.70    252.00      2.00  Tank

Link Results at 24:00:00 hrs:
----------------------------------------------
                      Flow  Velocity  Headloss
Link                  m3/h       m/s    /1000m
----------------------------------------------
P1                   56.70      0.89      9.97
P6                  -64.26      1.01     12.57
P7                  -59.26      0.93     10.82
P8                  -77.26      1.21     17.68
P10                  43.70      0.39      1.52
P11                  34.70      0.85     11.91
P9                   18.00      0.64      8.58
P2                  120.96      1.90     40.56
P4                  120.96      1.90     40.56
P3                   -0.00      0.00      0.00
P5                   -0.00      0.00      0.00
Pump1               120.96      0.00    -56.27  Pump
Pump2                 0.00      0.00      0.00  Pump
Valve                 0.00      0.00      0.00  TCV

Analysis ended Thu Jan 23 11:36:59 2014




                                                 Page 17
```

Figure B.17: Report file (page 17) containing the network simulation results of the tutorial provided on chapter 2 of this manual.