

**DESIGNING A DYNAMIC THERMAL AND
ENERGY SYSTEM SIMULATION SCHEME FOR
CROSS INDUSTRY APPLICATIONS**

W. Bouwer

Thesis submitted in partial fulfilment of the requirements for the degree

PHILOSOPHIAE DOCTOR

in the Faculty of Engineering

Department of Mechanical Engineering

Potchefstroomse Universiteit vir Christelike Hoër Onderwys

November 2004

ABSTRACT

Title: Designing a dynamic integrated thermal and energy system simulation scheme for cross industry applications

Author: Werner Bouwer

Promoter: Prof. E.H. Mathews

Department: Mechanical Engineering

Degree: Philosophiae Doctor

Keywords: Thermal simulation; Heating ventilation and air-conditioning (HVAC) system simulation; Ventilation and cooling (VC) system simulation; Dynamic; Integrated; Control simulation; Component models; Simulation engine; User interface; Building thermal and energy systems; Mine thermal and energy systems; Simulation procedure; Mass flow procedure; System simulation scheme.

The South African economy, which is largely based on heavy industry such as minerals extraction and processing, is by nature very energy intensive. Based on the abundance of coal resources, electricity in South Africa remains amongst the cheapest in the world. Whilst the low electricity price has contributed towards a competitive position, it has also meant that our existing electricity supply is often taken for granted. The economic and environmental benefits of energy efficiency have been well documented. Worldwide, nations are beginning to face up to the challenge of sustainable energy - in other words to alter the way that energy is utilised so that social, environmental and economic aims of sustainable development are supported.

South Africa as a developing nation recognises the need for energy efficiency, as it is the most cost effective way of meeting the demands of sustainable development. South Africa, with its

unique economic, environmental and social challenges, stands to benefit the most from implementing energy efficiency practices. The *Energy Efficiency Strategy for South Africa* takes its mandate from the South African *White Paper on Energy Policy*. It is the first consolidated governmental effort geared towards energy efficiency practices throughout South Africa. The strategy allows for the immediate implementation of low-cost and no-cost interventions, as well as those higher-cost measures with short payback periods. An initial target has been set for an across sector energy efficiency improvement of 12% by 2014.

Thermal and energy system simulation is globally recognised as one of the most effective and powerful tools to improve overall energy efficiency. However, because of the usual extreme mathematical nature of most simulation algorithms, coupled with the historically academic environment in which most simulation software is developed, valid perceptions exist that system simulation is too time consuming and cumbersome. It is also commonly known that system simulation is only effective in the hands of highly skilled operators, which are specialists in their prospective fields. Through previous work done in the field, and the design of a dynamic thermal and energy system simulation scheme for cross industry applications, it was shown that system simulation has evolved to such an extent that these perceptions are not valid any more.

The South African mining and commercial building industries are two of the major consumers of electricity within South Africa. By improving energy efficiency practices within the building and mining industry, large savings can be realised. An extensive investigation of the literature showed that no general suitable computer simulation software for cross industry mining and building thermal and energy system simulation could be found. Because the heating, ventilation and air conditioning (HVAC) of buildings, closely relate to the ventilation and cooling systems of mines, valuable knowledge from this field was used to identify the requirements and specifications for the design of a new single cross industry dynamic integrated thermal and energy system simulation tool.

VISUALQEC was designed and implemented to comply with the needs and requirements identified. A new explicit system component model and explicit system simulation engine, combined with a new improved simulation of mass flow through a system procedure, suggested a marked improvement on overall simulation stability, efficiency and speed. The commercial usability of the new simulation tool was verified for building applications by

doing an extensive building energy savings audit. The new simulation tool was further verified by simulating the ventilation and cooling (VC) and underground pumping system of a typical South African gold mine. Initial results proved satisfactory but, more case studies to further verify the accuracy of the implemented cross industry thermal and energy system simulation tool are needed. Because of the stable nature of the new VISUALQEC simulation engine, the power of the simulation process can be further extended to the mathematical optimisation of various system variables.

In conclusion, this study highlighted the need for new simulation procedures and system designs for the successful implementation and creation of a single dynamic thermal and energy system simulation tool for cross industry applications. South Africa should take full advantage of the power of thermal and energy system simulation towards creating a more energy efficient society.

SAMEVATTING

Titel: Designing a dynamic integrated thermal and energy system simulation scheme for cross industry applications

Outeur: Werner Bouwer

Promotor: Prof. E.H. Mathews

Departement: Meganiese Ingenieurswese

Graad: Philosophiae Doctor

Sleuteltermes: Thermal simulation; Heating ventilation and air-conditioning (HVAC) system simulation; Ventilation and cooling (VC) system simulation; Dynamic; Integrated; Control simulation; Component models; Simulation engine; User interface; Building thermal and energy systems; Mine thermal and energy systems; Simulation procedure; Mass flow procedure; System simulation scheme.

Die Suid-Afrikaanse ekonomie is grootliks gebaseer op swaar industrie soos mineraalontginning en - prosesseering. Hierdie industrie en prosesse is baie energie intensief. As gevolg van die volop natuurlike steenkoolbronne, bly elektrisiteit in Suid-Afrika van die goedkoopste ter wereld. Terwyl hierdie goedkoop elektrisiteitsprys verseker dat Suid-Afrika kompetend bly, het dit tot gevolg dat die bestaande energy toevoer gereeld as vanselfsprekend aanvaar word. Die ekonomiese- en omgewings verwante voordele wat, 'n verhoging in energie effektiwiteit tot gevolg het, is reeds goed gedokumenteer.

Wêreldwyd begin nasies die uitdaging van volgehoue ontwikkeling ondersoek. In ander woorde, die manier waarop energie gebruik word sodat sosiale, omgewings en ekonomiese doelwitte wat volgehoue ontwikkeling inhou bevoordeel kan word. Suid-

Afrika, as ontwikkelende land, erken die noodnagheid van energie effektiwiteit. Energiedoeltreffendheid is bewys as die mees koste effektiewe metode om aan die vereistes van volgehoue ontwikkeling te voldoen. Suid-Afrika met sy unieke ekonomiese-, omgewings- en sosiale uitdagings, kan die meeste baat by die implementering van energie effektiwiteits praktyke. Die *Energy Efficiency Strategy for South Africa* kry sy mandaat van die Suid Afrikaanse *White Paper on Energy Policy*. Dit is die eerste gekonsolideerde regeringspoging gefokus op die toepassing van energie effektiwiteitspraktyke regdeur Suid Afrika. Hierdie strategie maak voorsiening vir die onmiddellike implementering van lae koste en geen koste aksies, as ook hoër koste aksies met langer terugbetaal periodes. 'n Aanvanklike energiebesparings teiken van 12% is daargestel vir 2014.

Termiese- en energiestelsel simulاسie word globaal erken as een van die kragtige en mees effektiefste gereedskapstukke beskikbaar om algehele energie effektiwiteit te verhoog. As gevolg van die gewoonlik intensiewe wiskundige natuur van meeste van die simulاسie algoritmes, gekoppel aan die historiese akademiese omgewing waarin hierdie simulاسie pakette ontwikkel is, bestaan geldige negatiewe persepsies. Daar bestaan ook 'n persepsie dat stelselsimulاسie te tydrowend en omslagtig is. Dit word ook algemeen aanvaar dat stelselsimulاسie slegs effektief is 'n die hande van 'n hoogs geleerde operateur, gewoonlik 'n spesialis in sy spesifieke veld. Deur die toepassing van vorige kennis, en die ontwerp van 'n dinamiese termiese en energie stelselsimulاسie skema vir kruis industrie toepassings, is bewys dat stelselsimulاسie tot so 'n mate gevorder het dat die algemene persepsies nie meer van toepassing is nie.

The Suid-Afrikaanse mynbou- en kommersiële gebou industrie is twee van die grootste gebruikers van elektrisiteit in Suid-Afrika. Deur die verbetering van energie effektiwiteits praktyke binne die mynbou- en gebou industrie, kan groot potensiële besparings gerealiseer word. 'n Omvattende ondersoek in die literatuur het getoon dat geen algemene gepaste rekenaar gereedskap vir die simulاسie van myn en gebou termiese en energie stelsels beskikbaar is nie. Aangesien die simulاسie van stelsels in geboue die simulاسie van mynstelsels navolg, kon waardevolle ondervinding uit hierdie veld gebruik word om vereistes en spesifikasies vir die ontwerp van 'n ten volle geïntegreerde, dinamiese kruis industrie termiese en energie stelsel gereedskap daar te stel.

VISUALQEC is ontwerp en geïmplementeer om aan hierdie vereistes en behoeftes soos geïdentifiseer te voldoen. 'n Nuwe eksplisiete simulase engine, gekombineer met 'n nuwe verbeterde simulase van massa vloei deur 'n stelsel prosedure, het 'n merkbare verbetering op algehele simulase stabiliteit, effektiwiteit en spoed tot gevolg gehad.

Die kommersiële bruikbaarheid van die nuwe simulase gereedskapstuk is vir gebou toepassings geverifieer. 'n Omvattende gebou energie besparings oudit is gedoen. Die nuwe simulase program is verder geverifieer deur die simulase van die verkoelings- en ventilasie stelsel en ondergrondse pompstelsel van 'n tipiese Suid Afrikaanse myn. Aanvanklike resultate het voldoende resultate gelewer, maar meer gevalle studies word benodig om die akkuraatheid verder te bevestig. As gevolg van die stabiele natuur van die nuwe VISUALQEC simulase engine, kan die simulase proses verder uitgebrei word tot die wiskundige optimering van die verskeie stelsel komponente.

In samevatting beklemtoon hierdie studie die nodigheid vir nuwe simulase prosedure en stelsel ontwerp vir die suksesvolle implementering en ontwikkeling van 'n enkel dinamiese termiese- en energie stelsel simulase pakket. Suid Afrika moet die volle krag, voordele en potensiaal van termiese- en energie stelsel simulase tot bevordering van 'n energie effektiewe nasie aangryp.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Prof. E.H. Mathews for the opportunity to perform this study. His guidance throughout has been of great value and I am grateful for his contribution to my ongoing development.

Many thanks also to the following people whose contributions throughout the course of this study have been invaluable:

- All my colleagues at Transfer of Energy, Momentum and Mass International (Pty) Ltd. for their continued input. A special thank you to D.C Arndt, D.T Claassen, M. den Boef and M.F Geysler for their knowledge, interest and help in the success of this study.
- Transfer of Energy, Momentum and Mass International (Pty.) Ltd. for the use of their simulation software QUICKCONTROL and various simulation resources.

Lastly, I would like to thank my wife, parents, family and friends. Without whose ongoing support this study would not have been possible.

TABLE OF CONTENTS

ABSTRACT	i
SAMEVATTING	iv
ACKNOWLEDGEMENTS	vii
TABLE OF CONTENTS	viii
NOMENCLATURE	xii
GRAPHICAL SYMBOLS	xiii
LIST OF FIGURES	xv
LIST OF TABLES	xvii

CHAPTER 1 – INTRODUCTION	1
---------------------------------	----------

1.1	Background	2
1.2	Sources of primary energy and electricity within South Africa.....	2
1.3	A growing demand for electricity.....	5
1.4	Focus on sustainable development	6
1.5	Gearing towards a more energy efficient South Africa	7
1.6	Thermal and energy system simulation	8
1.7	The impact of the South African mining industry.....	8
1.8	The building simulation field	11
1.9	Cross industry simulation technology.....	12
1.10	The need for this work	14
1.11	The contribution of this work.....	15
1.12	Outline of this work	15
1.13	References	17

2.1	An introduction to system simulation.....	21
2.2	Thermal and energy system simulation	22
2.3	Traditional approach to simulating thermal and energy system	23
2.4	Explicit mathematical modeling of system components	26
2.5	A traditional scheme for simulating large thermal and energy systems	29
2.6	Further system simulation concepts	31
2.7	The building simulation field and the basics of system simulation tools	32
2.8	Relating building and mine thermal and energy systems	33
2.9	Building simulation tools for cross industry system simulation	35
2.10	Designing successful thermal and energy system simulation tools.....	36
2.11	Criteria and requirements for successful system simulation tools	36
2.12	Dynamic integrated thermal and energy system simulation tools of the future.....	38
2.13	References	40

3.1	Introduction	44
3.2	Elements of an integrated thermal and energy system simulation tool	44
3.3	An object-orientated programming philosophy in system simulation design.....	48
3.4	A flexible, cross industry system simulation tool design	48
3.5	A new user interface design	54
3.6	A new simulation engine design	57
3.7	Simulating mass flow through integrated thermal and energy systems	60
3.8	Resolving further system simulation pitfalls.....	65
3.9	An illustrative example	65
3.10	Physical implementation of the new system simulation tool.....	68
3.11	References	69

CHAPTER 4 – BUILDING INDUSTRY VERIFICATION	70
---	-----------

4.1	Introduction.....	71
4.2	Verification procedure.....	72
4.3	Case study: Telkom Data Building.....	72
4.3.1	System description.....	73
4.3.2	Current state of the system.....	75
4.3.3	System simulation configuration.....	77
4.3.4	System operation verification.....	80
4.3.5	Retrofit options and further verification results.....	81
4.4	Building savings potential and conclusion.....	87
4.5	References.....	91

CHAPTER 5 – MINING INDUSTRY VERIFICATION	92
---	-----------

5.1	Introduction.....	93
5.2	Verification procedure.....	93
5.3	Case study: Kopanang.....	95
5.3.1	System description.....	95
5.3.2	System simulation configuration.....	98
5.3.3	Simplification of configured simulation model.....	99
5.3.4	System operation verification.....	100
5.3.5	Verification results.....	101
5.4	Conclusion.....	105

CHAPTER 6 – CONCLUSION	106
-------------------------------	------------

6.1	Efficiency and saving through cross industry application of simulation.....	107
6.2	Towards a more efficient South African building industry.....	108
6.3	Other applications for simulation in the building industry.....	108
6.4	Towards a more efficient South African mining industry.....	108
6.5	Other applications for simulation in the mining industry.....	109
6.6	Future work and further recommendations.....	110
6.7	Conclusion.....	111



















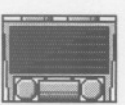



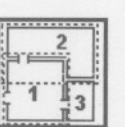





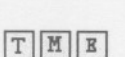

APPENDICES

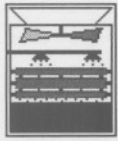
Appendix A – System component models	112
Appendix B – VISUALQEC	148
Appendix C – Telkom Data Building system specification and results.....	188
Appendix D – Kopanang system specification and results	201

NOMENCLATURE

P_{fan}	Pressure setup of fan
Q_{fan}	Flow rate of fan
P_{duct}	Pressure drop through duct
Q_{duct}	Flow rate through duct
Q_c	Cooling capacity
m_l	Mass flow of liquid
A	Empirical constant
B	Empirical constant
T_{wb}	Wet bulb temperature
T_{li}	Temperature of the liquid in
T_{le}	Temperature of the liquid exiting
a_0, a_1, a_2, a_3	Empirical coefficients

GRAPHICAL SYMBOLS

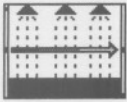
	Climate		Water dam
	Air Source		Water source
	Air t-piece converge		Water t-piece converge
	Air t-valve converge		Water t-valve converge
	Air t-piece diverge		Water t-piece diverge
	Air t-valve diverge		Water t-valve diverge
	Air heater		Water-water exchanger
	Air fan		Water pump
	Air damper		Water pipe
	Air-cooled chiller		Water-cooled chiller
	Air-air heat exchanger		Water cooling/heating coil
	Building zone		Water storage tank
	Controller - scheduler		Controller - STEP
	Controller - PID		Controller converge
	Sensors		Controller diverge



Water-air cooling tower



Pelton turbine



Water-air bulk air cooler



Water manual valve



Water valve



Air system connection



Water system connection



Control system connection

LIST OF FIGURES

- Figure 1.1 Primary energy sources used in the world
- Figure 1.2 Primary energy sources used in South Africa
- Figure 1.3 Energy sources used for electricity generation in South Africa
- Figure 1.4 Prediction of projected world energy consumption
- Figure 1.5 South African energy demand forecast
- Figure 1.6 Breakdown of electricity in South African buildings
- Figure 1.7 A typical modern building HVAC system
- Figure 1.8 A typical mine VC system
-
- Figure 2.1 Block diagram of a basic thermal and energy system configuration
- Figure 2.2 (a) A simple system consisting of a fan and duct
- Figure 2.2 (b) The flow rate balances the pressure drop through the duct with the pressure setup of the fan
- Figure 2.3 A graphical representation of an explicit cooling tower model
- Figure 2.4 Graphical representation of a thermal and energy system layout
- Figure 2.5 A typical modern HVAC system
- Figure 2.6 A typical mine VC system
-
- Figure 3.1 Critical elements of an integrated thermal and energy system simulation tool
- Figure 3.2 The VISUAL-DOE user interface
- Figure 3.3 The QUICKCONTROL user interface
- Figure 3.4 Typical “real” system layout
- Figure 3.5 Typical “virtual” system layout
- Figure 3.6 The system component
- Figure 3.7 System connections
- Figure 3.8 Thermal and energy system ports and properties
- Figure 3.9 The system simulation engine
- Figure 3.10 The dynamic integrated thermal and energy system simulation scheme
- Figure 3.11 The user interface requirements
- Figure 3.12 The drag and drop of pre-constructed components

-
- Figure 3.13 Representation of the implementation of the thermal and energy simulation interface
- Figure 3.14 Diagram of a mine surface cooling plant
- Figure 3.15 Solution to simulating mass flow
- Figure 3.16 Diagram of a typical HVAC/VC system
-
- Figure 4.1 Building simulation model air circuit
- Figure 4.2 Building simulation model water circuit
- Figure 4.3 Office chiller energy consumption
- Figure 4.4 Ground floor chiller energy consumption
- Figure 4.5 Summer simulated temperatures with set point drift
- Figure 4.6 Winter simulated temperatures with set point drift
- Figure 4.7 Summer simulated temperatures with fan and chillers scheduling
- Figure 4.8 Winter simulated temperatures with fan and chillers scheduling
-
- Figure 5.1 Schematic representation of the surface cooling plant cycle
- Figure 5.2 Schematic representation of the underground pumping network
- Figure 5.3 Complete integrated mine thermal and energy system of case mine
- Figure 5.4 Simplified integrated mine thermal and energy simulation model for case mine
- Figure 5.5 Verification of thermal characteristics of system pre-cooling towers
- Figure 5.6 Verification of thermal characteristics of system condenser towers
- Figure 5.7 Verification of thermal characteristics of system water-cooled chillers
- Figure 5.8 Verification of dam level of 38 level chilled water dam
- Figure 5.9 Verification of dam level of 38 level hot water dam
- Figure 5.10 Verification of dam level of 75 level settle dam
- Figure 5.11 Verification of pumping and cooling load

LIST OF TABLES

Table 3.1	VISUALQEC control components
Table 3.2	VISUALQEC system components
Table 3.3	VISUALQEC flow components
Table 4.1	Telkom Data Building description
Table 4.2	Building simulation climate data
Table 4.3	Building set point drift settings
Table 4.4	Building fan scheduling times
Table 4.5	Building chiller scheduling times
Table 4.6	Cost of current building operation vs. cost of repaired coil
Table 4.7	Cost savings of the office individual retrofits
Table 4.8	Financial analysis of the simulated office individual retrofits
Table 4.9	Cost savings of the first floor and ground floor retrofits
Table 4.10	Financial analysis of the first floor and ground floor retrofits
Table 4.11	Cost savings of the office combined retrofits
Table 4.12	Financial analysis of the office combined retrofits

CHAPTER 1

INTRODUCTION

The South African economy is very energy intensive. An abundance of coal and subsequent low electricity price has had the negative effect that existing energy and electricity supplies are often taken for granted. The future growth of the South African economy is heavily dependant on the increased awareness and application of energy efficiency practices. Thermal and energy system simulation is globally recognised as one of the most effective and powerful tools to improve overall energy efficiency. By providing the South African energy-consuming sector with an easy to use, mathematically stable, economically efficient and accurate thermal and energy system simulation tool, significant strides towards achieving governmental energy efficiency targets will be made.

1.1 Background

The South African economy, which is largely based on heavy industry such as minerals extraction (mining) and processing, is by nature very energy intensive [1]. Based on an abundance of coal resources [2], electricity in South Africa remains among the cheapest in the world. Whilst this historically low electricity price has largely contributed towards creating a globally competitive South African economy, it has also meant that existing and future electricity and energy supplies are often taken for granted [3] and unnecessarily wasted. With the extent of possible electricity shortages in the near future, this situation cannot continue.

1.2 Sources of primary energy and electricity within South Africa

Although classified as a third world developing country [4], South Africa boasts a well-developed electricity generation and supply infrastructure. In order to better understand the needs of the South African energy and electricity sector, it is important to understand how South Africa relates to the rest of the world in terms of primary energy sources as well as sources of energy used for the generation of electricity. According to the International Energy Agency (IEA) [5], the most often used primary source of energy in the world is oil (36%), followed by coal (23%), gas (21%), renewable (11%) and nuclear (7%). See Figure 1.1.

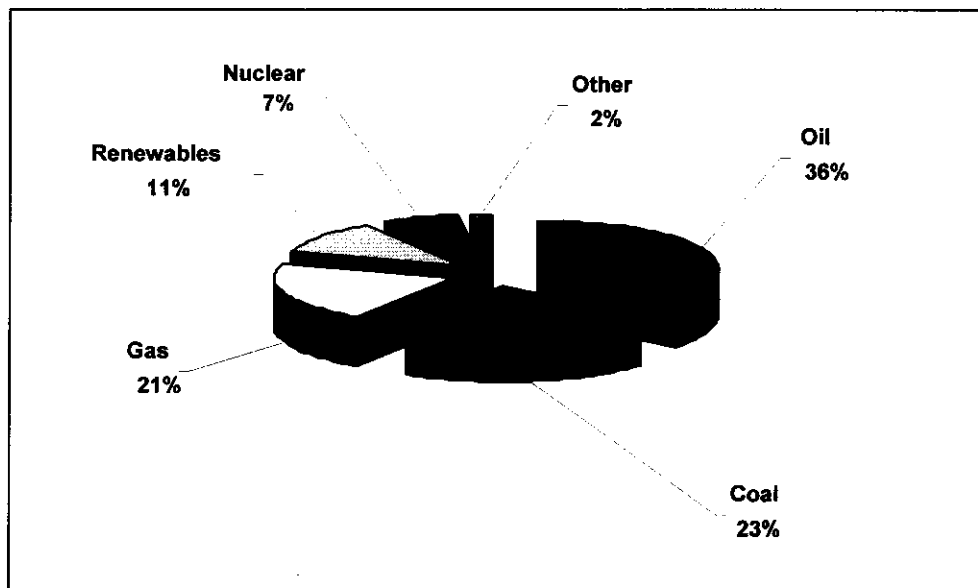


Figure 1.1: Primary energy sources used in the world

According to information supplied by the South African Department of Minerals and Energy (DME) [3], the profile of sources of primary energy used within South Africa, looks notably different to that of the rest of the world (Figure 1.1). Because of large natural coal resources, coal (73%) forms the main source of primary energy. Crude oil (17%) and natural gas (2%) resources are limited and consequently have to be imported. Renewable (5%) energy does play a limited but significant role, particularly in large hydroelectric power generation. Rich uranium deposits scattered throughout South Africa make nuclear (3%) energy a further viable source of primary energy. See Figure 1.2

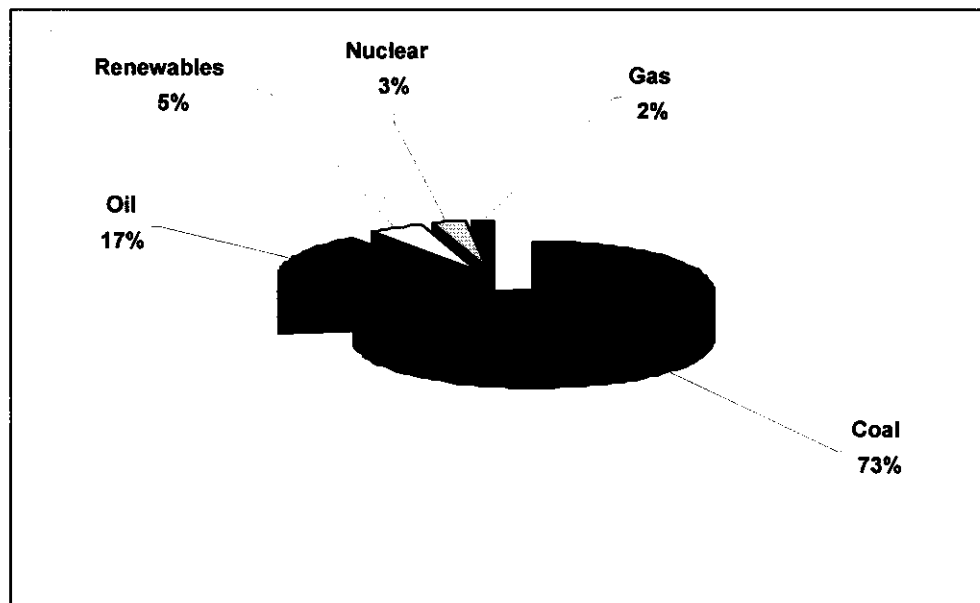


Figure 1.2: Primary energy sources used in South Africa

For the energy sources used in the generation of electricity, South Africa again differs from trends common to the rest of the world. According to IEA [5], the most common source of primary energy used for the generation of electricity in the world are coal (37%), followed by hydroelectric (29%), gas (22%), nuclear (7%) and oil (4%).

In South Africa, this picture again looks different. According to information submitted by local electricity generators to the National Energy Regulator of South Africa (NER) [2], coal (93%) by far forms the main source of energy used for the generation of electricity. Coal is furthermore supplemented to a far lesser extent by nuclear (5%) and hydroelectric (2%) energy sources. In South Africa, the one source of energy used for the generation of electricity

that is notably absent is gas. This is attributed to the lack of any large or developed natural gas fields within the South African borders. See Figure 1.3.

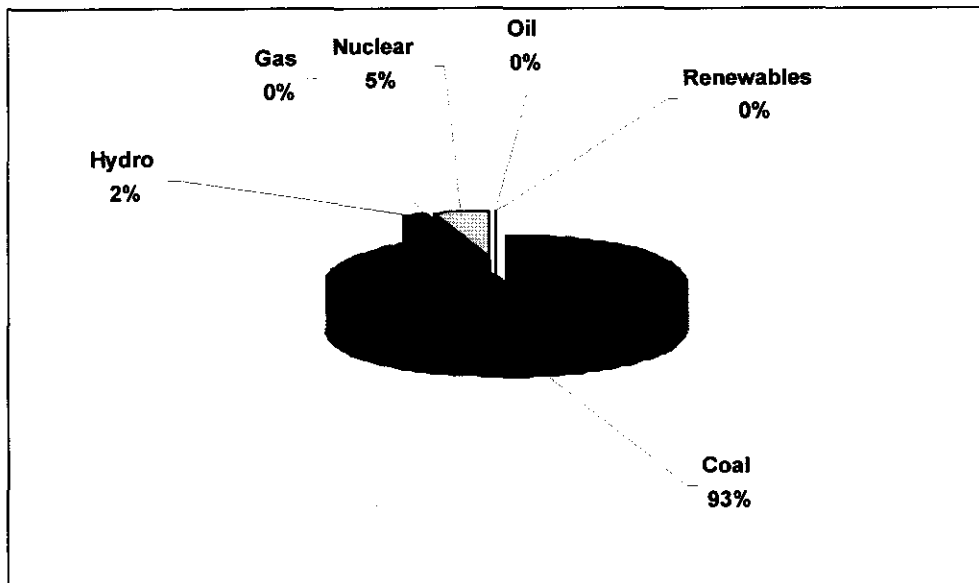


Figure 1.3: Energy sources used for electricity generation in South Africa

From Figure 1.1 and Figure 1.2 it is evident that South Africa is heavily dependent on coal as primary source of energy. Coal furthermore largely form the main source of energy for the generation of electricity throughout the South African (Figure 1.3) electricity generation sector. The overwhelming extent (93%) too which coal is used, is mainly attributed to an abundance of natural coal resources scattered throughout South Africa. According to data from the DME, South African coal production in 2002 was 242.7 million short tons (mmst). In 2001, South Africa was the world's sixth largest coal producer behind (in order) China, United States, Australia, India and Russia [4].

South Africa is in the unfavourable position that it has no other large developed natural gas or oil field resources to substitute or downscale its current overwhelming dependence on coal. The ready availability and low price of coal have thus contributed towards an economic environment wherein the unit price of electricity in South Africa can be counted as amongst the cheapest in the world. Although this historically low electricity price does play a significant role in the continued positive growth of the South African economy, it also create an environment were energy is often taken for granted and unnecessarily wasted. A further undesirable side effect of the low electricity price has been that energy efficiency practices

have been largely neglected and have frequently been demoted to make way for “priority” considerations, such as plant expansions and the increases in production throughput [6].

1.3 A growing demand for electricity

Not only is South Africa faced with the challenge of better managing its existing natural energy resources, it is also confronted with the ever increasing electricity demands of a growing and nation. The United States Department of Energy [6] predicts that the world primary energy consumption will increase by 59% over the period 1990 to 2020. The highest growth is expected in third world developing countries such as South Africa. The electricity demand in developing countries during the 1980’s has grown by more than 11% per year [7]. See Figure 1.4.

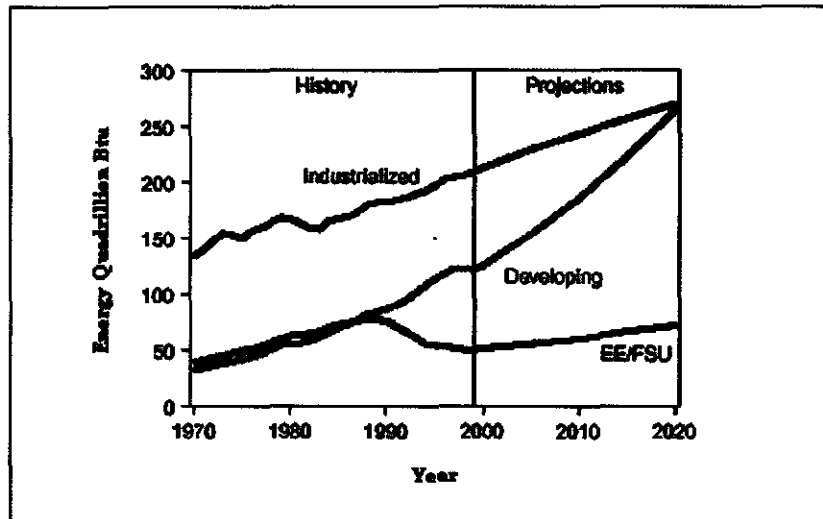


Figure 1.4: Prediction of projected world energy consumption

South Africa is currently in the fortunate position that surplus electricity supply and peak demand capacity does exist. If this upward electricity consumption and demand trend however persists, South Africa could possibly face an energy crisis in the near future. See Figure 1.5. Eskom, which is by far the largest generator and supplier of electricity in South Africa, projects these electricity shortages within the next five years. With large scale rural electrification projects currently undertaken by Eskom, the projected energy shortages could even be earlier.

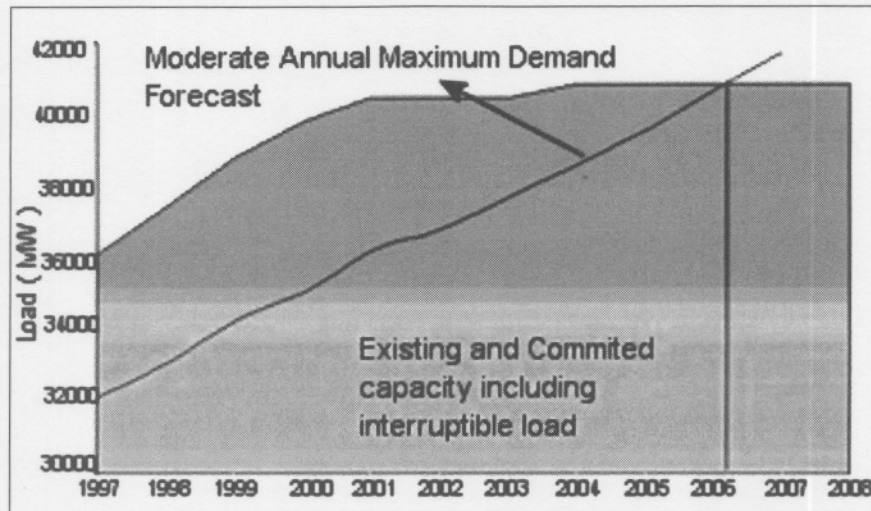


Figure 1.5: South African energy demand forecast

At a cost of R30 billion for the construction of a new power station, Eskom seeks cost efficient and effective methods to manage its current available electricity supply capacity for longer [18,19]. The possible energy crisis together with the cost of increasing electricity supply capacity for Eskom, only further strengthens the need to change the existing way in which energy and electricity within South Africa are utilised. Implementing and regulating energy efficiency practices will better prepare South Africa for the threat of possible future energy shortages.

To ensure future sustained economical growth within South Africa, it is imperative that South Africa starts implementing and benefiting from largely overlooked energy efficiency practices. It is also important that South Africa starts managing the existing natural energy resources with greater care towards future sustainability. In recent years energy efficiency has attracted more interest within South Africa, and a number of initiatives and projects have proven the merits and benefits of enhanced energy performance [6,7].

1.4 Focus on sustainable development

The social, economic and environmental benefit of energy efficiency has been well documented [8]. Worldwide, nations are beginning to face up to the challenge of sustainable energy – in other words to alter the way that energy is utilised so that social, environmental and economic aims of sustainable development are supported.

In 2002, the World Summit on Sustainable Development, held in Johannesburg, South Africa, recognised energy efficiency as a key tool to enhance clean energy development and to mitigate the negative effects of energy use upon the environment. It is also one of the only effective ways to manage existing natural resources towards greater sustainability.

The benefits of energy efficiency upon the environment are self-evident. These benefits are of particular relevance, as South Africa, through the use of coal as primary energy source, remains one of the highest emitters of the Greenhouse gas CO₂ per capita in the world. At a local level the problems of SO₂ and smoke emissions have been the focus of concern for many communities living adjacent to heavily industrialised areas. By implementing energy efficiency practices both the macroscopic and microscopic aspects of atmospheric pollution will be addressed. A *Draft White Paper on the Promotion of Renewable and Clean Energy Development* [8] further outlines these benefits.

South Africa, with its unique economic, environmental and social challenges; reconstruction and development program (RDP), stand to benefit the most from implementing energy efficiency practices. By implementing energy saving and best management practices, South Africa will prolong the life of its existing natural resources, mitigate negative environmental impacts and contribute significantly to averting the electricity shortages projected for the near future.

1.5 Gearing towards a more energy efficient South Africa

The *Energy Efficiency Strategy for South Africa* [9] takes its mandate from the *South African White Paper on Energy Policy* [10]. It is the first consolidated governmental effort geared towards energy efficiency practices throughout South Africa. The strategy allows for the immediate implementation of low-cost and no-cost interventions, as well as those higher cost measures with short payback periods. A target has been set for an across sector energy efficiency improvement of 12% by 2014.

Measures to reach this target include economic and legislative means, information activities, energy labels, energy performance standards, energy audits, energy management and the promotion of energy efficiency technologies.

1.6 Thermal and energy system simulation

With the coming of the computer age and its ability to solve continuous and discrete time systems, numerical simulation of systems, fluid flow, thermodynamics, aerodynamics etc. have developed rapidly. Today various powerful solution algorithms and simulation tools [11] are available that can be used to simulate just about any conceivable system or problem.

Thermal and energy system simulation is globally recognised as one of the most effective and powerful tools to improve overall energy system efficiency. However, because of the usual extreme mathematical nature of most simulation algorithms [12], coupled with the historically academic environment in which most simulation software are developed, valid negative perceptions exist that system simulation is too time consuming, unstable and often cumbersome. It is also commonly known that system simulation is only effective in the hands of highly skilled operators, which are specialists in their prospective fields [23,31]. By providing the South African energy consuming sectors with an easy to use, mathematically stable, economically efficient and accurate simulation tool, significant strides towards achieving the proposed energy efficiency targets [8,9,10], and creating a more energy efficient South Africa can be made.

Through an extensive literature survey, previous system simulation knowledge, and the design of a cross-industry dynamic thermal and energy system simulation scheme, it is shown that system simulation has evolved to such an extent that the negative common perceptions towards system simulation are no longer valid. South Africa, seeking methods to improve energy efficient practices, should take full advantage of the power of thermal and energy system simulation.

1.7 The impact of the South African mining industry

Mining is one of South Africa's biggest industries, along with manufacturing, trade and agriculture [13]. It is also one of the largest consumers of electricity within South Africa. The South African mining industry has been the mainstay of the South African economy for over a century [13]. Gold and diamonds are the two highly valued commodities, which were largely instrumental in the development of the country's infrastructure and the establishment of secondary industry during the first half of the twentieth century.

Gold mining has played a pivotal role in the economic development of the domestic economy, contributing about 4% in broad macro-economic terms to the gross domestic product (GDP) in 1996. This is substantially down from the 17% direct contribution recorded in 1980 when the gold price peaked [14]. Although the relative importance of the gold mining industry has fluctuated over the last decade with the performance of the gold price, gold mining still contributes about 4% directly to the South African GDP. Taking into consideration the indirect contribution to the economy, the creation of secondary industry and the multiplier effects, gold mining's total contribution today remains closer to 10%.

In 1996 the South African mining industry alone consumed 23.4% or 34,831.40 GWh of the total electricity supplied by Eskom [15]. Taking into account an average cost of 12.27 c/kWh the total cost amounts to a staggering R4200 million per annum [16]. Because of the energy-intensive nature of mining operations and the historically low per unit price of electricity, energy efficiency practices throughout the mining industry are generally neglected.

Electricity satisfies more than 95% of the average mine's energy requirements and constitutes a substantial portion of its working cost. For the average deep level mine the percentage of working cost typically varies from 10% to 13% [18]. With the financial viability and profit margin of mines directly related to the influence of the mineral prices and consequent sales, the need and benefits of more efficient cost effective mining activities becomes increasingly apparent.

With the construction of a new power station at a cost of around R30 billion, Eskom seeks efficient and effective methods to better utilise its current available electricity capacity. Recognising the need for, and benefits of energy efficiency, Eskom has embarked on a demand side management (DSM) program, within the mining industry, to motivate large consumers to manage their electricity demand better [18,19,37]. With new cost based tariff structured driving the DSM programme, Eskom is essentially forcing large consumers to change towards more cost and energy efficient practices.

The future of the South African mining industry lies in improving the economic effectiveness of the overall mining operation [17]. The main functions of a working mine are augmented by a multiplicity of essential auxiliary activities. These include the use of ventilation and cooling (VC) systems, pumping systems and various maintenance services. These activities, systems

and services are all heavy, energy and electricity intensive consumers. It is through the efficient design and control of these thermal and energy systems that the full potential for financial and environmental benefits of applied energy efficiency practices can be realised.

Computer simulation has been proven as a powerful tool that can be used to reduce overall system costs e.g. VC system; pump system, underground thermal and therefore overall mining operating costs. Although simulation has been available as a process analysis tool since the 1960's, its usage has been generally limited to the manufacturing and industrial processing industries [20]. Computer simulation involves creating a computer model of a real or proposed process or system. The model allows the engineer or operator to evaluate the system or process behaviour under various conditions or (what-if) scenarios that takes place over time [22,24].

System simulations are generally classed as either being of a static or dynamic nature. In a dynamic simulation there are changes in operating variables and conditions with respect to time and these are integrated into various feedback loops. Historically, the analysis of mine VC systems was of a static state nature. However, to evaluate the true operation of VC and thermal systems in mines, a dynamic simulation is needed. If true dynamic or real system operation can be simulated the potential for saving on the system operational costs of the mining industry alone can amount to thousands of rands.

The only method to effectively and efficiently evaluate, design, re-design and implement mine VC and thermal system control is through the use of a comprehensive, dynamic, fully integrated, thermal and energy system simulation tool. A comprehensive international survey showed that no dynamic integrated mine thermal and energy system simulation tool is available in the world today. The only thermal mine simulation found was ENVIRON [21], developed by the Council for Scientific and Industrial Research (CSIR) in South Africa. ENVIRON is however static and does not solve a mine in an integrated fashion over time.

With Eskom and governmental efforts [9] forcing the South African mining industry towards implementing more energy efficient practices, the need for the development and implementation of a fully integrated dynamic thermal and energy system simulation scheme and tool to be used for dynamic mine VC system simulation becomes increasingly apparent. By aiding the implementation of enhanced energy efficiency practices within the South

African mining sector through thermal and energy system simulation, South Africa will already realise a large proportion of the targeted energy saving of 12% by 2012.

1.8 The building simulation field

An important parameter for a well-designed, economic building is its thermal efficiency. In South Africa, studies have shown that as much as 57% of the total municipal electricity is utilised in commercial and industrial buildings [23]. See Figure 1.6. According to statistics provided by the South African Department of Minerals and Energy (DME) [3], at its worst, up to 74% of electricity goes directly towards the heating, ventilation and air conditioning (HVAC) of these commercial and industrial buildings.

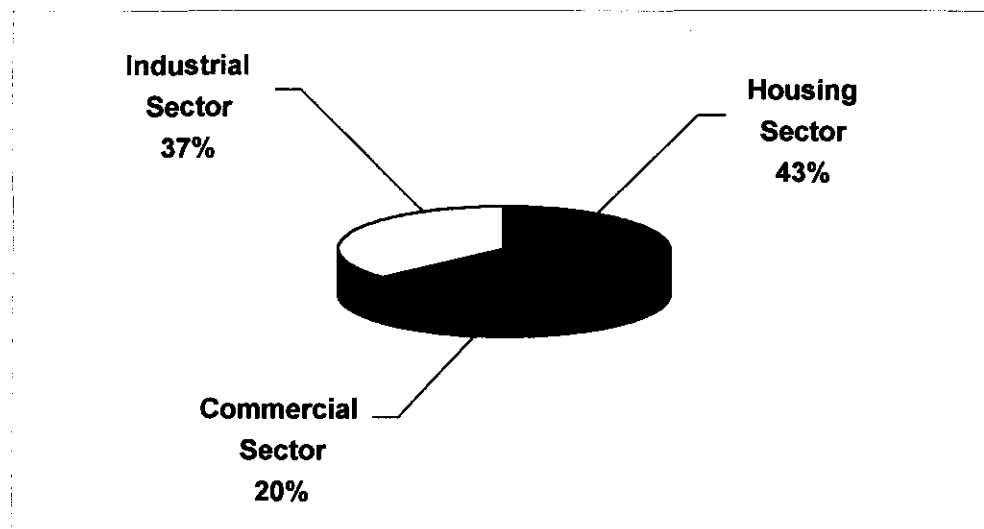


Figure 1.6: Breakdown of electricity in South African buildings

In the first half of the last century building designers paid scant attention towards the thermal characteristics of buildings. Energy was cheap, environmental concerns were generally ignored and the design and implementation of inefficient HVAC systems were common. The result was that buildings and their HVAC systems were unnecessarily wasteful, inefficient and extremely expensive to maintain. In the 1980's a move towards more cost efficient, environmentally aware buildings necessitated a more scientific and careful approach towards building HVAC system retrofit and design [25]. This opened the vast field of building and HVAC system simulation. Building and HVAC system simulation tools endeavour to predict

the dynamic response of the building HVAC system, i.e. indoor air conditions, system operation points and overall system energy consumption.

Some of the best-known building system simulation tools include APACHE [26], CABERETS [27], HVACSIM+ [28], HVAC-DYNAMIC [29], TRNSYS [30], DOE-2 [31] and QUICKCONTROL [32]. With effort, QUICKCONTROL has previously been used to successfully solve a pilot mining problem for Eskom [19]. The author however states, "it was very inefficient trying to use building software for mining applications". The potential for applying knowledge gained from building system simulation towards the creation of a more general cross-industry thermal and energy system simulation tool is huge.

1.9 Cross industry simulation technology

In essence, mine VC or thermal and energy systems are the same as building HVAC or thermal and energy systems. See Figure 1.7 for the layout of a typical modern building HVAC system.

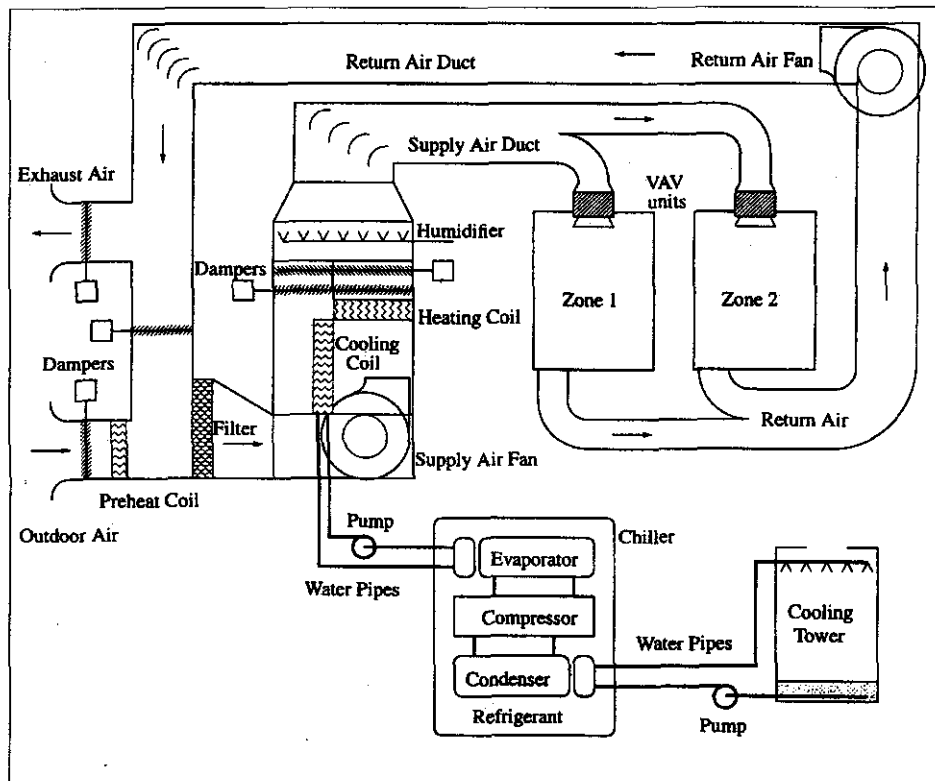


Figure 1.7: A typical modern building HVAC system

The typical modern building HVAC system (Figure 1.7) consists of four major flow networks. There is the air network formed by the ducts, filters, dampers, fan, etc.; the water coolant circuit driven by a pump; the condenser cooling tower circuit and the refrigerant circuit in the chiller. The chiller is responsible for cooling the ambient outside air to the required supply air temperature of the HVAC system. The ducts, filters, dampers and fan control the flow of the chilled and return air from and too the various building zones.

Figure 1.8 shows a schematic diagram of a typical mine VC or thermal and energy system layout.

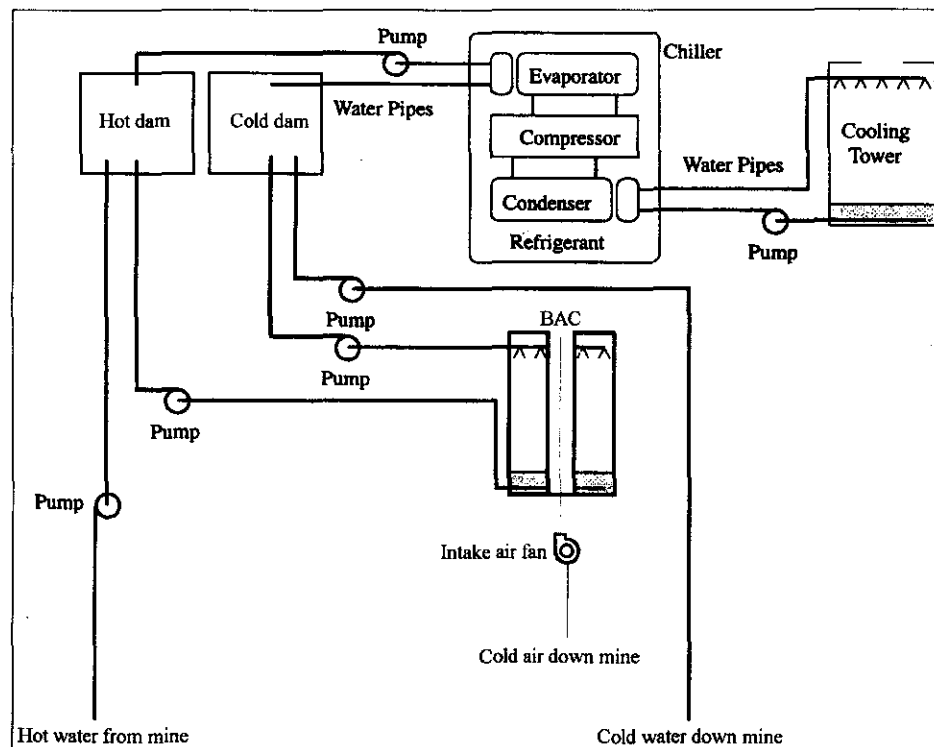


Figure 1.8: A typical mine VC system

As with HVAC systems in buildings, four major networks can be observed (Figure 1.8). The air network formed by the bulk air cooler (BAC) and intake fan, the water network driven by pumps, the condenser cooling tower and the refrigerant network of the chiller or refrigeration plant.

When Figures 1.7 and Figure 1.8 are compared, it can be seen that the thermal and energy systems of mines (VC) closely relate to the thermal and energy networks in buildings

(HVAC). Essentially, only the size of the required system components and performance requirements are different. It is through this comparison that it is possible to cross integrate building HVAC concepts with mining VC concepts into a single dynamic thermal and energy system simulation tool. By addressing the common negative perceptions as highlighted by section 1.6, a mathematically stable, economically efficient and accurate cross-industry thermal and energy system simulation scheme can be created.

Having been extensively verified for building applications [33,34,35,36,37], and exposed to the field of mine VC system simulation, QUICKCONTROL was identified as having the potential for contributing to the design and implementation of a single cross-industry dynamic integrated thermal and energy system simulation tool. This new thermal and energy system simulation scheme forms part of a new system simulation tool specifically created for the design and implementation of energy efficiency practices for both building and mining applications within South Africa.

1.10 The need for this work

The literature survey showed that:

1. The South African economy is by nature very energy intensive.
2. Based on an abundance of coal resources, electricity in South Africa remains among the cheapest in the world. This has the negative effect that the existing electricity supply is often taken for granted.
3. The energy consumption of the world is set to rise dramatically in the coming years, especially in developing countries such as South Africa.
4. The electricity consumption in South Africa has risen drastically in the previous 10 years, and will continue to do so in the future.
5. If current energy inefficient practices continue, South Africa will face an energy crisis in the near future.
6. Energy efficiency is going to play a large role in the energy policies of the South African government and the sustainable development of the economy. The South African government has set an initial overall energy efficiency improvement of 12% by 2014.

7. Thermal and energy system simulation is globally recognised as one of the most effective and powerful tools to improve energy efficiency. However, valid negative perceptions exist that system simulation is too time consuming, unstable and often cumbersome.
8. The thermal and energy systems used by the South Africa mining industry are large consumers of electricity.
9. Through the use of thermal and energy system simulation, definite potential for improving energy efficiency in the mining industry exist.
10. Although such tools in various forms exist, none could be found that solve the common negative perceptions and dynamic requirements.
11. Commercial and industrial buildings are large consumers of electricity.
12. Building thermal and energy systems (HVAC) share large similarities with mining thermal and energy systems (VC).
13. By combining building simulation knowledge, with mining requirements, a single dynamic integrated thermal and energy system simulation scheme for cross-industry application can be developed.

Therefore, the need was established to develop a mathematically stable, economically efficient and accurate cross-industry thermal and energy system simulation scheme that can be used to improve the overall energy efficiency across multiple South Africa energy consuming sectors.

1.11 The contribution of this work

Thus, this work focused on the need for an easy to use, mathematically stable, economically efficient and accurate cross-industry dynamic thermal and energy system simulation tool. The work identified the benefits, requirements and current shortfalls of such tools. It proposes the development of a new dynamic integrated thermal and energy system simulation scheme, to be implemented in a single system simulation tool that can be used across various thermal and energy consuming industries i.e. building and mining thermal and energy industry.

Specific attention was given to addressing the historic negative issues and perceptions towards thermal and energy system simulation. Such a simulation scheme was designed and implemented in a single tool, VISUALQEC. The tool and simulation scheme was tested on

building and mining applications, which in turn identified certain limitations, and identified further requirements for additional work on the subject.

1.12 Outline of this work

This study aims to follow a path from basic principles and practise in thermal and energy system simulation, through the development and design of a cross-industry dynamic integrated system simulation scheme, its tool, the verification and the implications thereof.

Chapter 2 serves as introduction to concepts, theory and traditional practices in system simulation. A look at the contributions made by the building simulation industry and the impact it has on the simulation of thermal and energy systems of mines. Criteria and requirements for a successful cross-industry system simulation tool are also discussed.

Chapter 3 follows the design of this cross-industry dynamic integrated thermal and energy system simulation tool. The design is based on principles and practices discussed in chapter 2. A brief discussion on the physical implementation of the system simulation tool is also presented.

Chapter 4 serves as verification for the cross-industry simulation tool discussed and implemented in chapter 3. A detailed verification on the thermal characteristics and energy consumption of the case building is presented.

Chapter 5 serves as verification for the cross-industry simulation tool discussed and implemented in chapter 3. Detailed verification on the thermal characteristics and energy consumption of the case mine is presented.

Chapter 6 serves as conclusion and discussion on possible improvements, application and future work to be done to improve the field of cross-industry thermal and energy system simulation.

1.13 References

- [1] Department of Minerals and Energy of the Republic of South Africa, *Integrated Energy Plan for the Republic of South Africa*, <http://www.dme.gov.za/>, 2003.
- [2] National Electricity Regulator, *Electricity Supply Statistics for South Africa*, 2001.
- [3] Department of Minerals and Energy of the Republic of South Africa, *Annual Report from the Department of Minerals and Energy of the Republic of South Africa*, <http://www.dme.gov.za/>, 2001.
- [4] Energy Information Administration, Department of Energy of America, *South African Country Analysis Brief*, <http://www.eia.doe.gov/>, 2004.
- [5] International Energy Agency, *Key World Energy Statistics from IEA*, 2002.
- [6] United States Department of Energy, *Energy Highlights*, <http://www.eia.doe.gov/>, 2001.
- [7] Lavine M., Gadgil A., Meyers A., Sathaye J., Stafurik J., Wilbanks T., *Energy efficiency, developing nations and eastern Europe: A report to the US working group on global energy efficiency*", International Institute for Energy Conservation, June 2001.
- [8] Department of Minerals and Energy of the Republic of South Africa, *The Draft White Paper on the Promotion of Renewable and Clean Energy Development*, <http://www.dme.gov.za/>, 2002.
- [9] Department of Minerals and Energy of the Republic of South Africa, *Draft Energy Efficiency Strategy of the Republic of South Africa*, <http://www.dme.gov.za/>, 2004.
- [10] Department of Minerals and Energy of the Republic of South Africa, *South African White Paper on Energy Policy* <http://www.dme.gov.za/>, 1998.
- [11] Law A.M., Kelton W.D., *Simulation Modeling and Analysis*, McGraw-Hill, 1991.
- [12] Kheir N.A., *Systems Modeling and Computer Simulation*, Editor, Marcel Dekker Inc., 1988.
- [13] Goldberg I., *South Africa's Mineral Industry*, pp. 1-12, 1992.
- [14] Chamber of Mines of South Africa, *The Importance of Gold Mining to South Africa*, <http://www.bullion.org.za/>, 2001.
- [15] Department of Mineral and Energy, *Energy balance for 1996*, <http://www.dme.gov.za/energy/>, 2001
- [16] National Electricity Regulator, *Electricity Supply Statistics for South Africa: Overview of electricity prices*, pp.13, 1998.

-
- [17] Rymon-Lipinski W.K., *Challenges of mining at great depth*, Department of Mineral and Energy, <http://www.dme.gov.za/>, 2001
- [18] Lane I.A., Delpont J., *Load Audits and Simulations to develop DSM Potential in the Mining Sector*, 1996.
- [19] Els R., *Potential for Load Shifting in Ventilation and Cooling Systems*, Thesis presented in partial fulfilment of the requirements for the degree Master of Engineering, 2000.
- [20] McIntosh Scott L., *How computer simulations reduce costs in the underground mining industry*, Technical Paper, McIntosh Redpath Engineering, 1999.
- [21] Oldenburg M., Vagenas V., Clement S., *Evaluation of ESA Simulation Tools for Mining Applications*, Laurentian University Mining Automation Laboratory (CDN).
- [22] Lebrun J., *Simulation of HVAC systems*, Renewable Energy, Vol. 5, Part 2, pp. 1151-1158, 1994.
- [23] Andersen J.J., *Cape Town Electricity Load Study and End-Use Segmentation*, Proceedings of the Seminal and Main Steering Committee Meeting Demand Side Management and Related Projects, DMEA, 1993.
- [24] Arndt D.C., *Integrated dynamic simulations of large thermal systems*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 2000.
- [25] Bevington R., Rosenfeld, A.H., *Energy for buildings and homes*, Scientific American, pp. 39-45, 1990.
- [26] Irving S.J., *APACHE – An integrated approach to thermal and HVAC system analysis*, International Journal of Ambient Energy, Vol. 7, pp. 129-136, 1986.
- [27] Samuel A.E., Chia T.H., *Simulation of the full and part load energy consumption of HVAC systems of buildings*, Building and Environment, Vol. 18, pp. 207-218, 1983.
- [28] Park C., Clark D.R., Kelly G.E., *HVACSIM Buildings, Systems and Equipment Simulation Program: Building Loads Calculation*, Gaithersburg, MD20899, 1986.
- [29] Heitz M., Ogard O., Vnovakovic, Brustad G., *HVAC-DYNAMIC – a training simulator for dynamic analysis of HVAC plants*, Modeling, Identification and Control, Vol. 10, pp. 159-164, 1989.
- [30] Beckman W.A., Broman L., Fiksel A., Klein S.A., Lindberg E., Schuler M., Thorton J., *TRNSYS the most complete solar energy system modeling and simulation software*, In AAM, editor, Renewable Energy Climate, Change Energy and Environment, World Renewable Energy Congress, Reading, UK, 1994.
-

-
- [31] Birdsall B., Buhl W.F., Ellington K.L., Erdam A.E., Winkelmann F.C., *Overview of the DOE-2 building energy analysis program*, Technical report: Simulation Research Group, Lawrence Berkeley Laboratory, University of California, 1990.
- [32] Mathews E.H., van Heerden E., Arndt D.C., *A tool for integrated HVAC, building, energy and control analysis Part1: overview of QUICKCONTROL*, Building and Environment, Vol. 34, pp. 429-449, 1999.
- [33] Mathews E.H., Arndt D.C., Piani C.B., van Heerden E., *Developing cost efficient control strategies to ensure optimal energy use and sufficient indoor comfort*, Applied Energy, Vol. 66/2, pp. 135-159, 2000.
- [34] van Heerden E., Mathews E.H., *Applying an Integrated Building and HVAC Simulation Tool to Successfully Predict Energy Savings in Buildings*, Fifth International Conference for the International Building Performance Simulation Association, 1997.
- [35] Mathews E.H., Malan A.G., Arndt D.C., *HVAC Control strategies to enhance comfort and minimise energy usage*, Energy and Buildings, Accepted for publication, 2000.
- [36] Rousseau P.G., Mathews E.H., *Needs and trends in integrated building and HVAC thermal design tools*, Building and Environment, Vol. 28, No. 4, pp. 439-452, 1993.
- [37] Arndt D.C., *“Further extension, verification and application of integrated building HVAC system and control simulation”*, Thesis presented in partial fulfilment of the requirements for the degree Masters of Engineering, University of Pretoria, 1997.
- [38] Els R., *Energy Evaluations and Load Shift Feasibility in South African Mines*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, Potchefstroom University, 2002.

CHAPTER 2

THERMAL AND ENERGY SYSTEM SIMULATION

Simulation is the execution of a model, represented by a computer program that gives information about a physical system being investigated. Since the advent of the computer age the numerical simulation of continuous and discrete time systems has developed rapidly. Today powerful and general solution algorithms are available which may be used to simulate any conceivable kind of system. A discussion of the most important principles, practices, consideration, criteria and requirements when designing a thermal and energy system simulation tool form the main objective of this chapter.

2.1 An introduction to system simulation

A system is defined as a collection of independent components whose performance parameters are interrelated. All the components together form a single unified whole or system. Simulation is defined as the process of attempting to predict aspects of the behaviour of some system by creating an approximate, usually mathematical or logical, model of it.

Computer simulation is thus the execution of a model, represented by a computer program that gives information about a physical system being investigated. System simulation thus means observing a synthetic system that imitates the performance of a real system. Since the advent of the computer age in the late 1960's, the numerical simulation of continuous and discrete time systems has developed rapidly [1].

Today powerful simulation tools are available which may be used to simulate any conceivable kind of system. These simulation tools are actually user interfaces to powerful mathematical routines that solve mathematical models of the simulated system and its various system components. Well known simulation tools includes GPSS, SIMAN, SIMSCRIPT II.5, SLAM II, ACSL, APROS, ARTIFEX, Arena, AutoMod, C++SIM, CSIM, FluidFlow, Gepasi, JavSim, MJX, MedModel, Multiverse, NETWORK, OPNET Modeler, POSES++, Simulat8, PowerSim, QUEST, REAL, SHIFT, SIMPLE++, SMPL, SimBank, SimPlusPlus, TIERRA, Witness, Javsim and SPICE [2]. These simulation tools are usually implemented in such a way that a system and its various individual system components are presented in a logical, relation based, block diagrammatic manner [3,4]. See Figure 2.1.

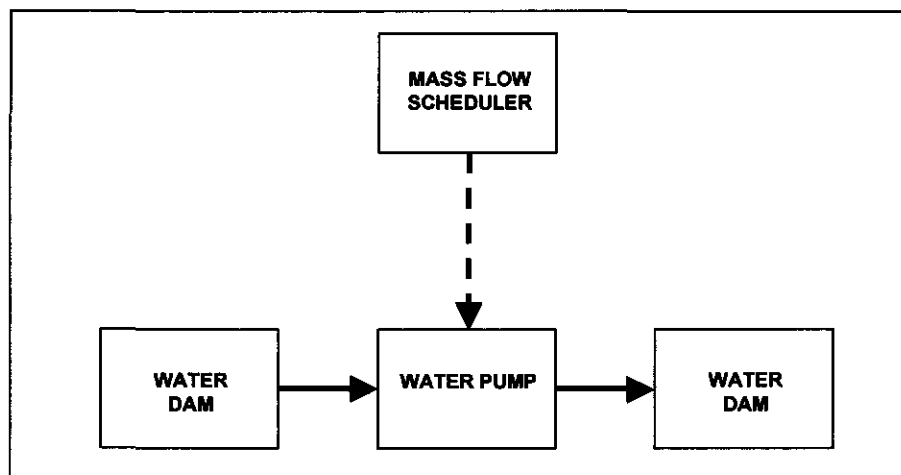


Figure 2.1: Block diagram of a basic thermal and energy system configuration

In most simulation tools the system simulation process progresses in three main stages and follows a predetermined, fixed system simulation scheme. Initially a pre-processing stage is used for specifying the system and its system component details. Then follows the actual core processing stage where the mathematical solution of the system and individual system components are computed. Lastly, a post-processing stage presents the simulated system and its results in a palatable form to the user.

Traditionally, because of the mathematical intensity of especially the core processing stage, computer system simulation is often considered slow and unstable. Furthermore, the more system components that form part of a simulated system, the more complex and mathematically unstable a possible solution of the simulated system can become.

2.2 Thermal and energy system simulation

Thermal and energy system simulation is the calculation of system operating variables (such as pressures, temperatures, and flow rates of energy and fluids) of specific points in a thermal system operating at a steady state [5]. Thermal and energy system simulation presumes knowledge of the performance characteristics of all system components, as well as equations for the thermodynamic properties of the working substances. Each system component thus has a unique mathematical representation or model, based on the performance characteristics of that component, that calculates operating variables at a specified point. These system component models can either be described by implicit or explicit mathematical equations.

Thermal and energy system simulation may be used at the design stage to help achieve an improved thermal and energy system design. It is also used on existing systems when there is a known operating problem or a possible improvement or control (energy efficiency) strategy is being considered. The effect on the system of changing a system component can be examined before real changes, with usually substantial financial implications, are made.

This ensures that the required operating conditions within the system, together with the required financial viability and maintenance of the system can be achieved. Thermal and energy system simulation is thus an invaluable tool for both the technical and financial management of thermal and energy consuming systems. Traditionally, the mathematical equations for performance characteristics of the system components of a thermal and energy

system and the thermodynamic properties, along with energy and mass balances across all system and system component boundaries, form a set of simultaneous equations relating the various system component operational variables. The mathematical description of thermal and energy system simulation is that of solving these simultaneous equations, many of which may be non-linear and possibly more than one solution.

2.3 Traditional approach to simulating thermal and energy system

Traditionally, system simulation implies solving sets of equations, which model various components of a system. Stoecker [5] remarks, “Thermal system simulation is the calculation of operating variables (such as pressures, temperatures, and flow rates of energy and fluids) in a thermal system operating at a steady state.” A more general point of view also includes unsteady conditions and particularly emphasizes the evolution of system variables with time. Stoecker [5] gives an example of a simple thermal and energy system consisting of a single fan and a duct layout. See Figure 2.2.

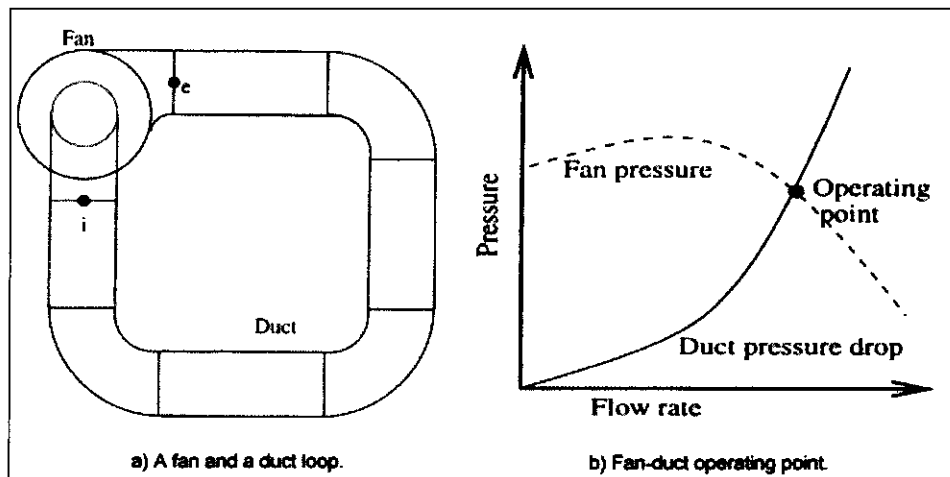


Figure 2.2: (a) A simple system consisting of a fan and a duct

(b) The flow rate balances the pressure drop through the duct with the pressure setup of the fan

At steady state the flow rate, Q , must balance the pressure drop through the duct P_{duct} , with pressure setup of the fan, P_{fan} . The relationship between Q and P is the mathematical model of the element. In general the relationship between flow and pressure is nonlinear. For example

$$P_{fan} = 0.3 - 0.2Q_{fan}^2 \quad (2.1)$$

the user should have a very broad mathematical understanding of the specific system being simulated.

An efficient method [10] for solving the system will analyse equations 2.3 to 2.6 and detect that only the following two equations are actually required.

$$P = 0.3 - 0.2Q^2 \quad (2.7)$$

$$P = 0.0625 + 0.653Q^{1.8} \quad (2.8)$$

To solve these equations by hand, one is substituted into the other to reduce the set to a single equation for the single unknown, Q

$$0.3 - 0.2Q^2 = 0.0625 + 0.653Q^{1.8} \quad (2.9)$$

P is then calculated from equations 2.7 and 2.8. Equation 2.9 implies the solution is one of the roots of

$$0 = -0.2375 + 0.653Q^{1.8} + 0.2Q^2 \quad (2.10)$$

The standard Newton-Raphson method [6] for solving an equation $y = y(x) = 0$ is to start with an arbitrary guess value, x_b , and to repeatedly calculate an updated guess, x_c , through

$$x_c = x_b - \frac{y(x_b)}{y'(x_b)} \quad (2.11)$$

To solve more than one equation, for example

$$f_1(x_1, x_2, x_3) = 0 \quad (2.12)$$

$$f_2(x_1, x_2, x_3) = 0 \quad (2.13)$$

$$f_3(x_1, x_2, x_3) = 0 \quad (2.14)$$

requires solving the set

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} \begin{bmatrix} x_{1,t} - x_{1,c} \\ x_{2,t} - x_{2,c} \\ x_{3,t} - x_{3,c} \end{bmatrix} \quad (2.15)$$

for the updated guess values $x_{1,c}$, $x_{2,c}$, $x_{3,c}$.

The matrix of partial derivatives, or Jacobian as it is known, must be evaluated for every updating step at the current guess values $x_{1,t}$, $x_{2,t}$, $x_{3,t}$. The evaluation of the Jacobian and subsequent solution of the equations to determine the new guess values is the main computational burden. This Newton-Raphson procedure lies at the heart of most equation solvers used in traditional simulation tools. Variations exist which evaluate the Jacobian numerically or guess it iteratively. The solution obtained by Stoecker [5] with this technique is $P_{fan} = P_{duct} = 0.25$ kPa, and $Q_{fan} = Q_{duct} = 0.5$ m³/s.

2.4 Explicit mathematical modeling of system components

A necessary preliminary step, before the simulation of a thermal and energy system can begin, is almost invariably that of modeling some characteristic(s) of the system components or processes. The simulation operation almost always uses data in equation form; see equations 2.1 and equations 2.2. This conversion of data to equation form is referred to as the mathematical modeling of system components.

The ability to express thermodynamic properties in equation form is valuable in work with thermal systems. It is true that property equations abound, but accurate ones are usually complex. In many cases it is possible to use some classical thermodynamic property relationship combined with classic mass- and heat-transfer theory to suggest an additional term that can be added to a simple or ideal system model relation equation.

In the past, many system components themselves involved complex, iterative and non-explicit mathematical equation solutions. With these large sets of implicit system component models, a simulated system easily became mathematically unstable.

Until implicit equation solution methods becomes unconditionally stable, it is important to ensure that all system components are explicitly modelled. Explicit meaning that for every set of input variables a safe, stable set of output variables must be calculated.

To illustrate the explicit mathematical modelling of thermal and energy system components, the explicit thermodynamic model for a cooling tower is presented. Figure 2.3 is a schematic representation of the required model parameters.

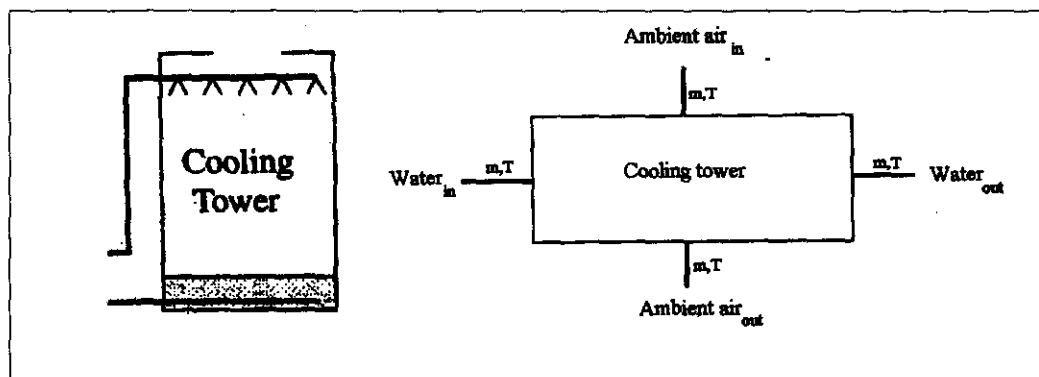


Figure 2.3: Schematic representation of an explicit cooling tower model

Data from the manufacturer's catalogue is used to accurately predict the explicit equation of the cooling capacity, Q_c , of the cooling tower model. The cooling capacity and mass flow, m , are plotted together. The relation between these two parameters is represented by the equation

$$Q_c = A(T_{wb})m_l^B \quad (2.16)$$

with Q_c the cooling capacity, m_l the mass flow of water (liquid) through the tower and T_{wb} the wet bulb temperature of the ambient air. From equation 2.16 the power (B) remains constant over a range of various cooling towers. The constant (A) is a function of the wet bulb temperature (T_{wb}) of the air flowing through the cooling tower. Equation 2.17 describes this function.

$$A = C(T_{li})T_{wb} + D(T_{li}) \quad (2.17)$$

The gradient (C) and constant (D) of equation 2.17 are functions of the water inlet temperature T_{ii} . Equation 2.18 and 2.19 represent the functions for C and D.

$$C = a_0 T_{ii} + a_1 \quad (2.18)$$

$$D = a_2 T_{ii} + a_3 \quad (2.19)$$

The coefficients a_0, a_1, a_2, a_3 are empirical curve fit coefficients. By substituting equation 2.18 and equation 2.19 into equation 2.16 an empirical equation that predicts the performance of a cooling tower is given by equation 2.20.

$$Q_c = [(a_0 + a_1 T_{ii}) T_{wb} + a_2 T_{ii} + a_3] m_i^B \quad (2.20)$$

With the cooling capacity (Q_c) known, an outlet temperature (T_{ie}) and leaving air enthalpy (h_{ae}) can be calculated using equations 2.21 and 2.22 respectively.

$$T_{ie} = T_{ii} - \frac{Q_c}{m_i c_p} \quad (2.21)$$

$$h_{ae} = h_{ai} + \frac{Q_c}{m_a} \quad (2.22)$$

The exiting air temperature can now be calculated using psychrometric relations if we assume the leaving air to be completely saturated. Further illustrations of this process are represented in Appendix A.

The importance of accurate explicit mathematical modelling of the various system components in the stability and ease of use of system simulation tools cannot be overemphasised. In an integrated thermal and energy simulation tool, it is important that these mathematical models are derived in such a manner that any given input conditions produce an appropriate and stable output condition. If we refer to the cooling tower model and Stoecker's [5] example, in equation 2.1, 2.2 and 2.20 an input parameter produces a corresponding output condition. Many methods exist to determine the most appropriate and accurate method to represent the internal processes of various system components. Some system models are invariably more complicated in their derivation and solution than others. It is the work of the simulation scheme and tool designer to decide and implement the most appropriate system

component models for his required application. It is important to note that this, together with the overall system simulation solution scheme, is usually where the success of the implemented system simulation tool is decided. Implicit equations for system component models cause simulations to be slow and usually unstable, explicit equations ensure stable results with a substantial gain in simulation computation speed (no need for iterative solvers).

2.5 A traditional scheme for simulating large thermal and energy systems

After the various component models for all relevant real system components have been found, the mass flow and thermal response of these models are numerically and explicitly solved. Traditionally the various system component models have to be solved in the correct order to maintain the various balances. To set up and build a given system configuration, the simulation tools' user interface is used. The system connections between the various components are graphically established by connecting a specific port of one component to a specific port of another component. Most integrated simulation software proceeds in this fashion. See Figure 2.4 for a graphical representation of a thermal and energy system layout. Detailed designs of the various elements of a system component are described in chapter 3.

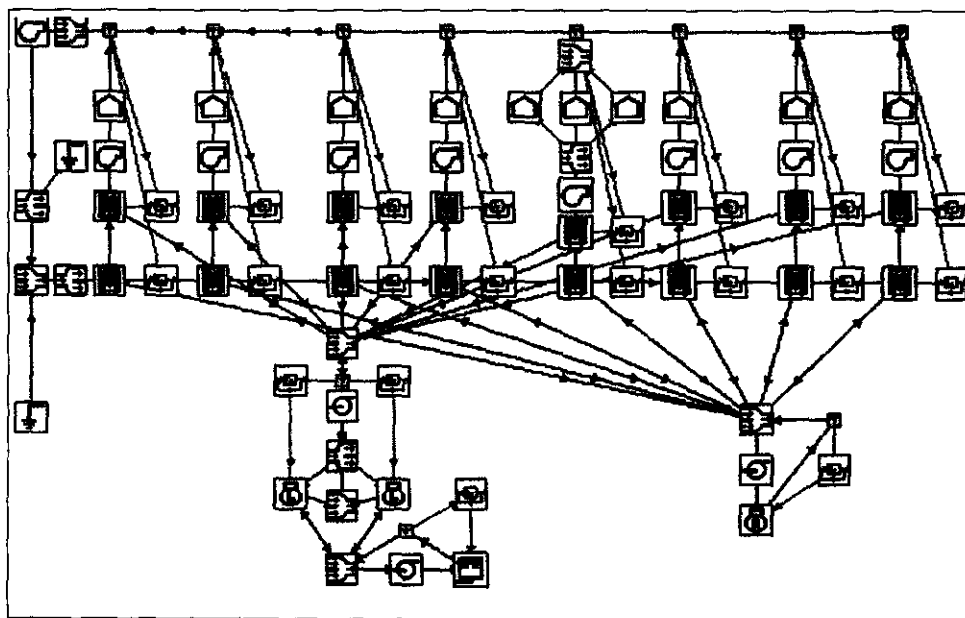


Figure 2.4: Graphical representation of a thermal and energy system layout

From Figure 2.4, the system connections establish the system configuration and system component relationships but do not inherently predict the order of system component

solution. Also, at this stage, the unknown variables still need to be determined. When using a traditional numerical solution scheme such as the Newton-Raphson method, as described in section 2.3, the order in which the various system components are solved is extremely important. To determine the order of component solution, various graph theory techniques are available.

One of the various methods to analyse the correct order in which the system components have to be solved, is that of the Tarjan Depth-First-Search algorithm. Tarjan devised a recursive algorithm to transform any graph (such as the block diagram of a thermal and energy system layout) into a spanning tree.

Van Heerden [3] explores this and various other techniques to determine the correct solution sequence and minimum system unknown variables. A previous version of the QUICKCONTROL [39] simulation engine used this technique with varied success. Having to find this absolute correct order of system component solution places further mathematical instabilities on the thermal and energy system simulation.

The traditional simulation scheme for simulating large thermal and energy systems, when using system solution methods such as Newton-Raphson, can thus be summarised as follows:

1. Configure system component models for all relevant system components.
2. Establish system connections between components.
3. Establish correct order of system component solution.
4. Numerically solve the system unknowns.

This traditional simulation scheme is used in most thermal and energy system simulation tools today. Because of the intense mathematical nature of most stages of this process, this scheme, has however in most cases been proven to be cumbersome, time consuming, computationally complex and most often, unstable.

When using this traditional simulation scheme, the efficiency, speed and stability of the simulation is directly related to the size of the configured system. The larger the simulated system the more difficult, slower and unstable these simulation platforms become.

2.6 Further system simulation concepts

A system simulation can either be classified as continuous or discrete. In a continuous system, the flow through the system is that of a continuum, e.g. a fluid or even solid particles, flowing at such rates relative to particle sizes that the stream can be considered as a continuum. In discrete systems, the flow is treated as a certain number of integers [5]. The analysis of the flow of people through a supermarket involving the time spent at various areas and the checkout counter is considered an example of a discrete system.

In a deterministic system simulation the input variables are precisely specified. In stochastic analysis the input conditions are uncertain, either being completely random or, more commonly, following some probability distribution. In simulating the performance of a steam-electric generating plant that supplies both process steam and electric power to a facility, for example, a deterministic analysis starts with one specified value of steam demand along with one specified value of the power demand. A stochastic analysis might begin with some probability description of the steam and power demands [5].

In a dynamic system simulation there are changes of operating variables and conditions with respect to time, in steady state system simulations the system component operating variables remain constant. Dynamic analysis is used for such purposes as the study of the effect of system control in order to achieve greater precision and to avoid unstable system operating conditions. A dynamic system simulation is thus capable of delivering a more “realistic” model of real world processes and systems.

Integrated system simulation [16,17,18,19] refers to the way in which components in a simulated system react towards other components in the same simulation. In a fully integrated system simulation one system component will have an immediate or delayed response on all the other related or connected system components. To simulate the effect of control effectively, it is extremely important to have a fully integrated and dynamic system simulation scheme.

2.7 The building simulation field and the basics of system simulation tools

A simulation tool is a computer program, which allows the user to carry out the simulation scheme (see section 2.5) in an efficient and convenient manner. In the HVACSIM+ program [11], every type of subsystem or system component has a mathematical model, which is coded in a FORTRAN [12] subroutine. The parameters and simulation variables are passed to the routine through FORTRAN common blocks and as subroutine parameters.

HVACSIM+ was developed by the United States Government and is based on the TARP [13] program. It follows a hierarchical, modular approach and integrates with multi-step formulae the exact and detailed differential equations of the various components of the system. QUICKCONTROL [39] was developed by TEMM International (Pty) Ltd. and is based on work done by Rousseau [16], Lombard [15] and van Heerden [17]. Coded in PASCAL, it follows the hierarchical natural block structure of the building HVAC system. This leads to a close correspondence between the mathematical model and the real physical system. The solution can proceed in much the same manner, as the physical system will operate; i.e. each subsystem detects its input conditions and transforms them to output conditions.

In both simulation tools the user selects a system model by specifying the subsystem type and gives the initial guess values and parameters of the model by filling in a data sheet. The interconnections between subsystems are specified in the form of a list saying for example “variable i of subsystem j equals variable m of subsystem n .”

The numerical methods employed within these simulation tools are techniques for solving large numbers of simultaneous non-linear algebraic equations, algorithms for integrating stiff ordinary differential equations and the interpolation of sampled data. To solve the nonlinear implicit equations, HVACSIM+ and QUICKCONTROL [39] uses the powerful public domain routine SNSQ, which is based on a quasi-Newton method with the Jacobian, determined via the Broyden method [8,9].

This procedure has been developed over the last couple of decades and provides a powerful and general method for solving the resulting sets of nonlinear equations. It has good convergence properties but, in common with all other methods for solving nonlinear equations, requires reasonable accurate initial guess values.

Many other system simulation tools use similar techniques in their simulation scheme. There are however other methods of solutions which aim to produce faster more stable answers without compromising the solution accuracy. The future of efficient, effective and accurate thermal and energy system simulation lies in finding and implementing techniques that solve only sets of stable system equations.

Through initial work done by Arndt [14], QUICKCONTROL [39] was previously enabled to solve and simulate complex thermal and energy systems without solving any integrated set of differential system or systems of equations. The stability and robustness of this proposed simulation scheme ensures more efficient, accurate and inherently solvable system simulations.

2.8 Relating building and mine thermal and energy systems

In essence, mine thermal and energy systems closely relate and imitate building HVAC systems. Consider the schematic of a typical building variable air volume HVAC system as shown in Figure 2.5.

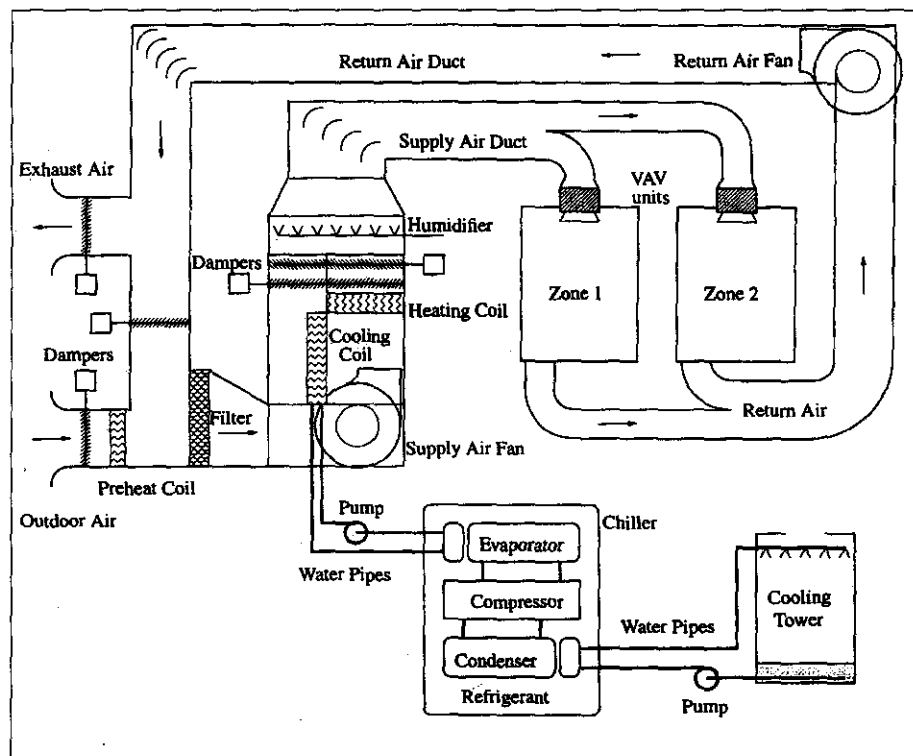


Figure 2.4: A typical modern HVAC system

Figure 2.4 shows that the system consists of four major flow networks. There is the air network formed by the ducts, filters, dampers, fan, etc.; the water coolant circuit driven by a pump; the condenser cooling tower circuit and the refrigerant circuit in the chiller. The networks are constructed from the various simulation elements by specifying the interconnections between the elements, e.g. the output of duct x is connected to the input of fan y, etc. These interconnected input and output ports are the nodes of the system. The ducts, fans, zones, etc. couple the nodes to each other. The thermodynamic properties of the fluid and mass flow rate at the nodes describe the state of the fluid at the nodes [5,17]. Figure 2.6 shows a schematic diagram of a typical mine VC or thermal and energy system layout.

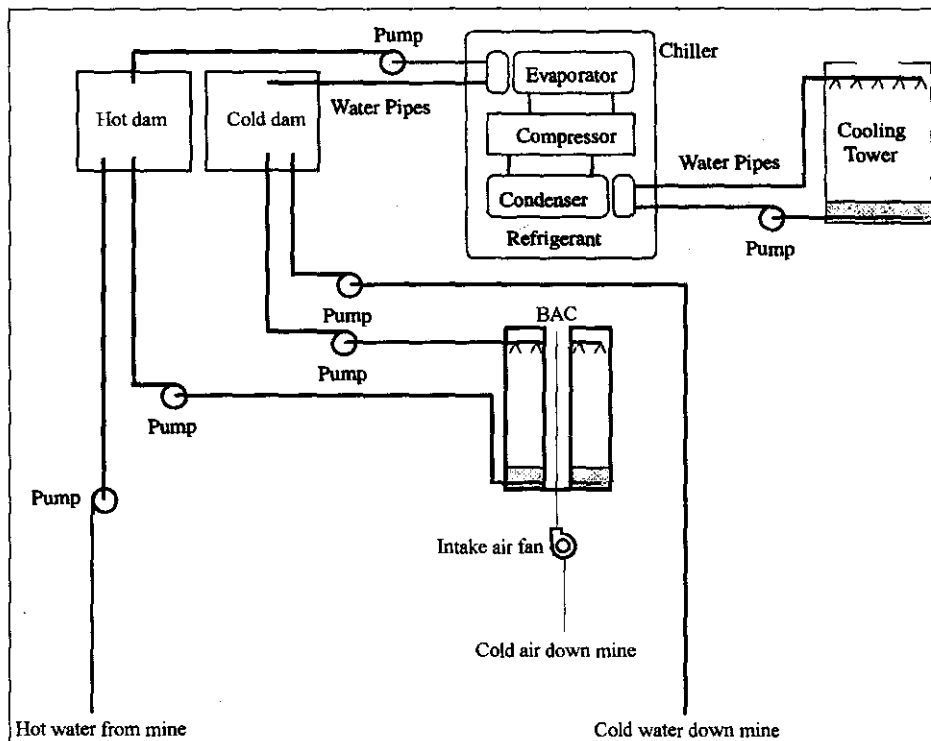


Figure 2.6: A typical mine VC system

As with HVAC systems in buildings (Figure 2.5), four major networks can be observed. The air network formed by the bulk air cooler (BAC) and intake fan, the water network driven by pumps, the condenser cooling tower and the refrigerant network of the chiller or refrigeration plant. Some layouts can also contain secondary cooling networks underground. This is mostly the case in deep mines where virgin rock temperature can reach 60°C at depths of about 4 km [18].

2.9 Building simulation tools for cross industry system simulation

There are many building and system analysis tools presently available. These tools can be primarily divided into two groups [19], energy analysis and system simulation tools. The primary function of energy analysis tools is to estimate system energy consumption. Various HVAC systems can be compared by means of typical system index figures to reveal the system with the lowest energy usage. These tools are normally based on load calculation methods where system energy usage is calculated by simplified consumption models.

Building system simulation tools endeavour to predict the dynamic response of the HVAC system and the building, i.e. indoor air quality (IAQ) and conditions, system-operating points and detailed energy consumption. In general these tools are component-based [17] which makes them more flexible. Some of the better-known building energy analysis tools are, BLAST [20], DOE-2 [21], VisualDOE [22], E-CUBE [23], AXCESS [24], COM-TECH [25], HAP E-20 [26], HAP 40 [27], TAS [28], TRACE [29], ENERPAS [30], ENERWIN [31] and BSIM2000 [32]. Building system simulation tools include APACHE [33], CABARETS [34], HVACSIM+ [11], HVAC-DYNAMIC [35], SPARK [36], TRNSYS [37], GEMS [38] and QUICKCONTROL [39].

When Figures 2.5 and 2.6 are compared, it can be seen that the ventilation and cooling (VC) systems of mines are related to the heating ventilation and cooling (HVAC) system networks in buildings (HVAC). It is through this comparison that it is possible to integrate building HVAC concepts into a general thermal and energy simulation scheme for application in both building and mining applications. The only obvious differences for mining applications being the quasi-constant internal reef loads, no effect from changing solar radiation, type of equipment, equipment size, water storage capacity, safety requirements and system operating points.

QUICKCONTROL [39] has previously been used to solve a pilot mining problem for Eskom [40]. Being a dynamic integrated and flexible simulation platform, concepts and ideas similar to those used by QUICKCONTROL [39] form the basis for a new dynamic integrated thermal and energy system simulation solution scheme. This scheme can be integrated into a stable, accurate and efficient cross-industry system simulation tool.

2.10 Designing successful thermal and energy system simulation tools

A common mistake when designing a dynamic integrated thermal and energy system simulation tool is the over emphasis of engineering and technical detail. In the past, scant attention has been given towards the efficient interaction of user and simulation tool. This is especially apparent in some of the better-known building and general simulation software tools such as DOE-2 [21,22,37]. Arndt [14] notes that it's almost necessary for all users to have at least one PhD before attempting to use these simulation tools.

There are various reasons for the difficult nature of these simulation tools. The main reason however, is that most of these tools were designed by research scientists and used most often only for research purposes [15,17]. With low cost powerful computers and a common operating system now available, simulation tools must be geared more towards commercial practices and use where time is of the essence and cost usually governs the success of a project.

For an effective, accurate and powerful simulation tool, a balance must thus be obtained between the technical complexity and the required practical implementation of the system simulation tool. Without this balance a powerful system simulation platform will be of no use to a non-research environment.

2.11 Criteria and requirements for successful system simulation tools

There is no purpose in developing any integrated thermal system simulation tool if the requirements set forth by the intended user cannot be satisfied. Most traditional system simulation tools do not satisfy the basic requirements of the commercial energy designer or energy contractor. These system simulation tools are normally developed by research institutions, which focus more on the mathematical integrity of the solution methods and accuracy of results than the practical results needed for a system simulation tool to be successfully used in a commercial environment [14].

If the South African energy-consuming sectors are to benefit from thermal and energy system simulation, these critical criteria and requirements have to be addressed. The following is a

discussion of the most important aspects to be addressed when developing a commercial thermal and energy system simulation tool.

1. It is very important that the system simulation tool must be component based in order to configure any system type. An option to include future system component models must also be included. The simulation tool must make use of the easiest possible method to quickly and efficiently configure a new or existing system. The tool must further allow for typical easy inclusion of pre-constructed system units and complete system types to decrease the configuration and construction time by means of default components and collections or sets of components.
2. Mathematically, component models must be a combination of fundamental principles and empirical correlation coefficients. This minimizes the required component input data. Scale (size) factors and typical system curves must be used to obtain the regression coefficients. It must also be possible that only one operating point needs to be specified in order to calibrate components. This operating point can either be from the equipment manufacturer or a measured value from the existing system. To ensure stability, all component models must be explicit.
3. The simulation tool must be fully integrated and component based to allow for the identification of all problem areas (total or component) and overall integrity during the verification and calibration stage. In this way the components, which do not perform according to the original specification can be identified and recalibrated. In doing so the accuracy of each component within the simulation can be ensured, thus assuring results and realistic performance predictions.
4. The simulation models or components within the integrated simulation tool must be dynamic, e.g. bypass control on water valves through cooling towers. This allows the accurate calculation of the transient response of all the system conditions and power consumption at any simulation time step. To investigate

the effect of control action and system capacity, the components must be solved in an integrated fashion.

5. Any control configuration, sensor or controller must be possible. The sensors and controllers must therefore also be component based to allow for any or all control strategies. Typical pre-constructed control units must be readily available as part of the tool. These configurations must be implemented with the easiest and most efficient methods available.
6. The simulation scheme must make use of an explicit equation philosophy on system component and simulation solution scheme level. This decreases execution time and markedly increases the component and simulation stability. The ideal for an integrated simulation tool is to use only explicit equations to calculate component conditions. With techniques similar to those suggested by Arndt [14], stable, accurate explicit simulations are a possibility.

2.12 Dynamic integrated thermal and energy simulation tools for the future

Various traditional techniques for simulating complex thermal and energy systems exist. The evaluation of the set of partial derivatives or Jacobian and subsequent solution of the equations by means of the Newton-Raphson procedure lies at the heart of most of the equation solvers used in simulation tools today. However, the numerical nature of these solvers makes the solution of the simulation process unpredictable and inherently unstable.

With new simulation techniques continuously being developed by an active building simulation community, the explicit and inherently stable solutions of large thermal and energy systems become increasingly possible. By creating simulation techniques that uses less mathematically intensive solution techniques throughout the system simulation process, the desired speed and stability can be achieved.

One of these techniques is changing the way in which the mass flow within a thermal and energy system is handled. The benefit of not solving but specifying mass flow across all system components within a configured simulation system is but one of the powerful means of stabilising the mathematical instabilities in thermal and energy system simulation.

Because of the close relation between building HVAC and mine VC systems these techniques can be efficiently integrated into to the development of a single, cross-industry, dynamic, integrated thermal and energy system simulation scheme and tool.

With the implementation of basic theoretical concepts, requirements and criteria as outlined by this chapter, the design and implementation of such a tool was made. Chapter 3 takes a detailed look at the creation and design of such a tool, VISUALQEC.

2.13 References

- [1] Law A.M., Kelton W.D., *Simulation modeling and analysis*, McGraw-Hill, 1991.
- [2] Vladimirescu S., Zhang K., Newton A.R., Pederson D.O., *SPICE VERSION 2.5 USER'S GUIDE*, Berkeley, CA. 94720.
- [3] Kheir N.A, *Systems Modeling and Computer Simulation*, Editor, Marcel Dekker Inc., 1988.
- [4] Cellier F.E., *Continuous Systems Modeling*, Springer-Verlag, 1991.
- [5] Stoecker W.F., *Design of Thermal Systems*, McGraw-Hill Book Company, 3rd Edition, 1989.
- [6] Chua L.O., Lin P.M., *Computer-Aided Analysis of Electronic Circuits*, Prentice-Hall Inc., 1975.
- [7] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., *Numerical Recipes in C, The Art of Scientific Computing*, Cambridge, 1992.
- [8] Wait R., *The Numerical Solution of Algebraic Equations*, John Wiley & Sons, 1979.
- [9] Churchouse R.F., *Handbook of Applicable Mathematics*, John Wiley & Sons, 1981.
- [10] Klein S.A., Alvarado F.L., *EES, Engineering Equation Solver*, F-Chart Software, 4406 Fox Bluff Road, Middleton, Wisconsin, 53562.
- [11] Park C., Clark D.R., Kelly G.R., *HVACSIM Buildings, Systems and Equipment Simulation Program: Building, Loads Calculation*, Gaithersburg, MD 20899, 1989.
- [12] The FORTRAN Company, <http://www.fortran.com>, 2001.
- [13] Walton G.N., *Thermal Analysis Research Program Reference Manual*, 1983.
- [14] Arndt D.C., *Integrated dynamic simulations of large thermal systems*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 2000.
- [15] Lombard C., *Two-port simulation of HVAC systems, and object orientated approach*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 1996.
- [16] Rousseau P.G., *Integrated building and HVAC thermal simulation*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 1994.

-
- [17] van Heerden E. *Integrated simulation of building thermal performance, HVAC system and control*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 1997.
- [18] Rymon-Lipinski W.K., *Challenges of mining at great depth*, Department of Mineral and Energy, <http://www.dme.gov.za/>, 2001.
- [19] Mathews E.H., van Heerden E., Arndt D.C., *A tool for integrated HVAC, building, energy and control analysis Part 1: overview of QUICKCONTROL*, Building and Environment, Vol. 34. pp. 429-449, 1999.
- [20] Lawrie L.K., *Day-to-Day use of energy software*, Energy Engineering, Vol. 89, pp.41-51, 1992.
- [21] Birdsall B., Buhl W.F., Ellington K.L., Erdam A.E., Winkelmann F.C., *Overview of the DOE-2 building energy analysis program*, Technical report: Simulation Research Group, Lawrence Berkeley Laboratory, University of California, 1990.
- [22] *VisualDOE*, Eley Associates, 142 Minna Street, San Francisco, California, 94105, USA, 2001.
- [23] Pedreya D., Bush V., *PCCUBE simulates hourly electric demand and energy consumption*, Energy Engineering, Vol. 89, pp. 23-37, 1992.
- [24] Howell R.H., Sauer H.J., *Energy Efficiency and conservation of building HVAC system using AXCESS energy analysis*, In second annual conference on ENERGY, Rolla, pp. 7-9, 1975.
- [25] Fuller K.E., Rohmund I. *Comtech for commercial building system analysis*, Energy Engineering, Vol. 89, pp. 6-22, 1992.
- [26] Acclimatisation and Development group, *Overview of the hourly analysis program Carrier*, 12 Ruede Paris-78230, LE PECQ, Available from Carrier SA, 1992.
- [27] HAP (Hourly Analysis Program) v4.0, Carrier Corporation, Software Systems, TR-1, P.O Box 4808, Syracuse, New York 13211, USA.
- [28] Gough M.C.B., *Component Based building energy and system simulation*, International Journal of Ambient Energy, Vol. 7, pp. 137-143, 1986.
- [29] Althof J., *Marketing and productivity opportunities of computer aided system design*, Energy Engineering, Vol. 84, pp. 4-29, 1987.
- [30] ENERPASS, Enermodal Engineering, 650 Riverbend Drive, Kitchener, Ontario, Canada.
- [31] ENERWIN, Texas A&M University, College Station, Texas, 77843-3137, USA.
- [32] BSIM2000, Danish Building Research Institute, P.O Box 115, Hoersholm, DK-2970, Denmark.
-

- [33] Irving S.J., *APACHE – An integrated approach to thermal and HVAC system analysis*, International Journal of Ambient Energy, Vol. 7, pp. 129-136, 1986.
- [34] Samuel A.E., Chia T.H., *Simulation of the full and part load energy consumption of HVAC systems*, Building and Environment, Vol. 18, pp. 207-218, 1983.
- [35] Heitz M., Ogard O., Vnovakovic, Brustad G., *HVAC-DYNAMIC – a training simulator for dynamic analysis of HVAC plants*, Modeling, Identification and Control Vol. 10, pp. 159-164, 1989.
- [36] SPARK (Simulation Problem Analysis and Research Kernel), Lawrence Berkeley National Laboratory, Mail Stop 90-3147, 1 Cyclotron Road, Berkeley, California 94720, USA.
- [37] Beckman W., Broman L., Fiksel A., Klein S.A., Lindberg E., Schuler M., Thornton J., *TRNSYS the most complete solar energy system modeling and simulation*, In AAM, editor, Renewable Energy Climate, Change Energy and Environment, World Renewable Energy Congress, Reading, UK, 1994.
- [38] Winkelmann F., *Advances in building energy simulation in North America*, Energy and Buildings, Vol. 10, pp. 161-173, 1988.
- [39] Mathews E.H., van Heerden E., Arndt D.C., *A tool for integrated HVAC, building, energy and control analysis*, Building and Environment, Vol. 34, pp. 429-449, 1999.
- [40] Els R., *Potential for Load Shifting in Ventilation and Cooling Systems*, Thesis presented in partial fulfilment of the requirements for the degree Master of Engineering, University of Pretoria, 2000.

CHAPTER 3

A NEW CROSS INDUSTRY SYSTEM SIMULATION TOOL

Thermal and energy system simulation is globally recognised as one of the most effective and powerful tools to improve overall energy efficiency. Valid negative perceptions does however exist that system simulation is too time consuming, mathematically unstable and often cumbersome. It is commonly known that system simulation is only effective in the hands of highly skilled operators. Through the design of a new cross industry thermal and energy system simulation tool it is shown that system simulation has evolved to such an extent that these common negative perceptions are no longer valid.

3.1 Introduction

In line with South African governmental efforts towards greater energy efficiency practices, outlined by the *Energy Efficiency Strategy for South Africa* [1] and *White Paper on Energy Policy* [2]. Together with efforts initiated by Eskom to support their demand side management program (DSM). The needs for a new, easy to use, mathematically stable, efficient and accurate cross industry thermal and energy system simulation tool was identified. VISUALQEC and its underlying system simulation scheme were specifically designed to support these energy efficiency efforts as well as to address some of the traditional thermal and energy system simulation pitfalls and shortcomings.

Currently most simulation tools for mining applications are of a static nature. However, VISUALQEC was designed to address the dynamic requirements of accurate thermal and energy system simulation of mining ventilation and cooling (VC) systems. Because of proven previous successes, VISUALQEC was designed with a simulation tool structure similar to that of QUICKCONTROL [3] and uses proven explicit system component models based on work by Rousseau [4], van Heerden [5], Lombard [6], and Arndt [7].

3.2 Elements of an integrated thermal and energy system simulation tool

An integrated thermal and energy system simulation tool consist of many interconnected and related software elements. However, three critical elements always govern the success of any integrated thermal and energy system simulation tool. These critical elements are, the system components and their simulation models (1), the system simulation engine (2) and the user interface (3). See Figure 3.1.

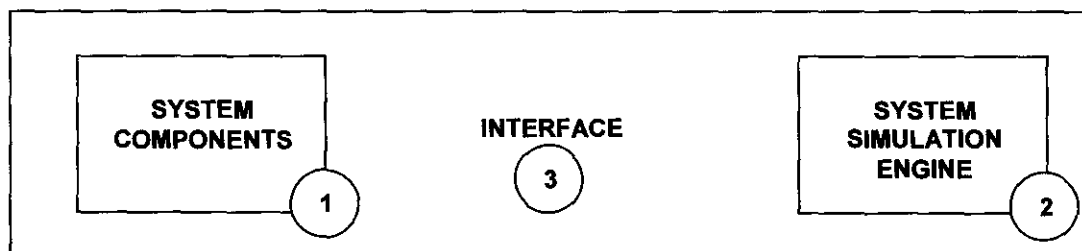


Figure 3.1: Critical elements of an integrated thermal and energy system simulation tool

The system components (1) (Figure 3.1) are the “virtual” equivalent of the “real” system components. Every type of system component contains a simulation model that simulates the thermodynamic process of that individual type of equipment. The system simulation engine (2) (Figure 3.1) is that part of the integrated thermal and energy system simulation tool not visible to the user. The simulation engine forms the driving force behind the simulated solutions generated by the system simulation tool.

The engine encapsulates all the required mathematical routines, simulation procedures, and logical programming needed for the successful completion of an integrated system simulation. Over the years, most of the work done in the field of simulation concentrated on the design and implementation of mathematically stable and successful system simulation engines. The interface (3) (Figure 3.1) is that part of the system simulation tool visible to the user. The user interface is responsible for the configuration, re-configuration and conclusion of information needed or results supplied by or to the integrated system simulation process.

To create a successful integrated thermal and energy system simulation tool all three of the critical elements (Figure 3.1) need to be designed with an equal amount of flexibility, usability, and mathematical and logical stability as main objective. In the past, a common over-emphasis was placed on the technical and engineering (mathematical) detail of the various elements. The design of the system components, as well as the system simulation engine typically received most of the development resources.

This technical over-emphasis created valid negative perceptions towards simulation. System simulation is commonly believed to be very time consuming, usually unstable (due to complexity of over accurate numerical solution techniques and implicit equations of component models) and often cumbersome and difficult to use. As discussed in section 2.10 and 2.11, the inherent limitations on computer graphical representation and the major involvement of research institutions in the development of these simulation tools in the past, interface design historically received scant attention.

Some of the well-known building thermal and energy system simulation tools both too varying degrees suffer from these ill-designed interfaces. See Figure 3.2 for the out-dated VISUAL-DOE [8] user interface and Figure 3.3 for the more powerful QUICKCONTROL user interface.

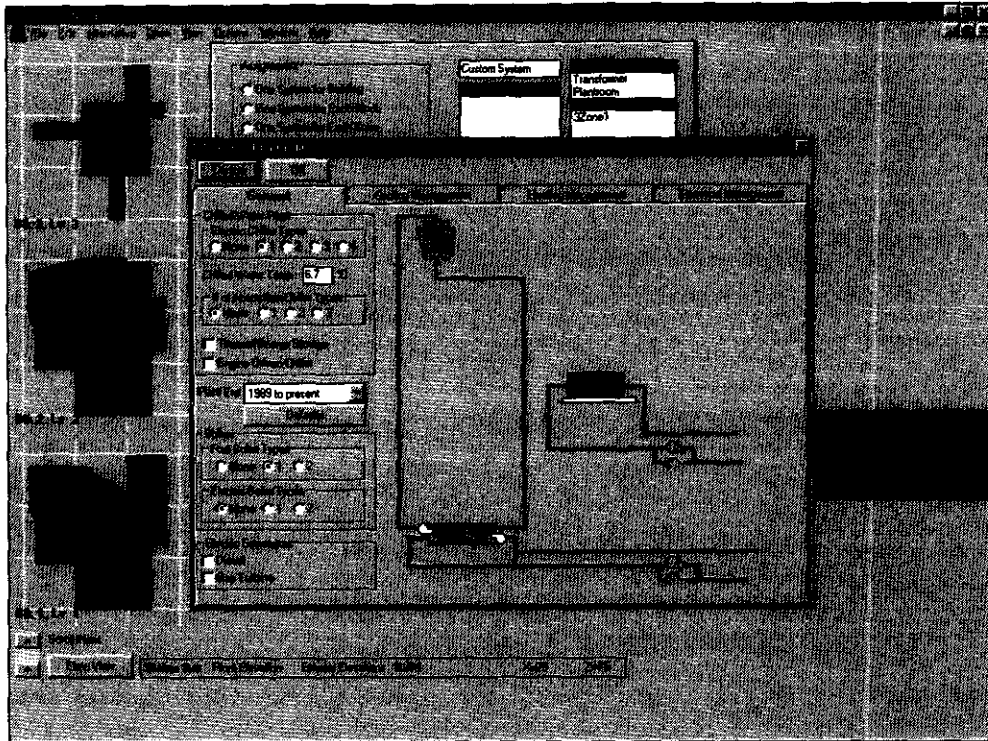


Figure 3.2: The VISUAL-DOE user interface

In VISUAL-DOE the user configures the required heating ventilation and cooling (HVAC) system by selecting an approximate appropriate system configuration from a database of pre-constructed system configurations. Although this method of system construction is very effective when the required system configuration matches an existing one, the VISUAL-DOE interface allows little flexibility in expanding these common systems to uncommon configurations or into new standard systems. To configure and calibrate the various system components the user also require detailed and accurate data as well as knowledge of how the various real system components function.

Originally designed to aid in the simulation of building control retrofits, QUICKCONTROL allows the user full flexibility towards building any required system configuration. However, to obtain a stable simulation environment, QUICKCONTROL also requires detailed and accurate system component data. In most cases, this accurate system component data is not readily available. QUICKCONTROL has the further limitation that no easy access is available to the user to view the resulting simulated system conditions.

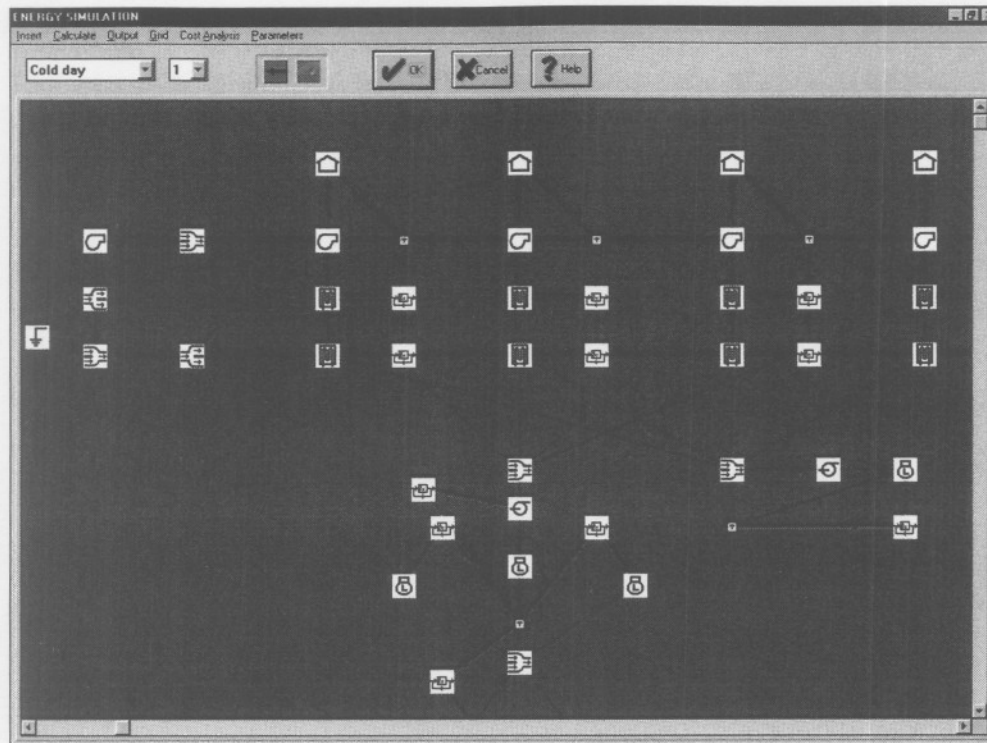


Figure 3.3: The QUICKCONTROL user interface

Figure 3.2 and Figure 3.3 emphasises the importance of ensuring a balance between usability and compute-ability when designing any system simulation tool. A mathematically stable thermal and energy system simulation tool is of no use if the easy and understandable access to the power of the simulation engine and the resulting simulation cannot be attained. Because of the dynamic cross-industry requirements of the new system simulation tool, special attention needs to be given to ensure the most user-friendly and logical interaction between the three critical main simulation tool elements.

To create a successful dynamic integrated thermal and energy system simulation tool, the requirements and criteria identified by section 2.10 and 2.11 need to be obtained. In QUICKCONTROL the system simulation engine (2) (Figure 3.1) and user interface (3) (Figure 3.1) were identified as being the two main tool elements that stop QUICKCONTROL, in its current implementation, from being used as a successful cross industry thermal and energy system simulation tool. VISUALQEC was thus designed to meet all the specified criteria and requirements. The following sections describe the design and implementation philosophy, as well as the specific requirements and detail, of the improved system simulation engine and user interface.

3.3 An object-orientated programming philosophy in system simulation design

In the last decade the object-orientated method [9,10] for software tool construction has become the method of choice for the design and implementation of large integrated and complicated computer software tools. In contrast to the old algorithmic approach, which viewed a software tool as a data processing unit, the idea behind object-orientated programming (OOP) is that software should emulate real world objects.

In object-orientated programming the distinction between data and algorithm is more diffuse. The data and the methods are regarded as equally important, each forming an integral part of the software object. With this design philosophy, objects such as thermal system and tool components can be represented and described as a set of attributes and actions. This close correspondence between software objects and physical hardware is ideal for programming integrated system simulation tools.

Just as with real world thermal and energy system components, each system simulation component is viewed as a unique, independent and self-calculating (component model) entity or object. Furthermore, with OOP it is possible to implement all the elements of a simulation tool, i.e. system components, simulation engine and interface, as separate interlocking objects. This new modular (object) flexibility of all tool elements within a system simulation tool vastly improves upon the old global integrated algorithmic programming philosophies.

In the past, changing the system simulation engine usually meant that the existing tool interface and system components had to be rewritten as well. With modular flexibility through OOP, the changed module can be replaced without having any, or substantially little, effect on the existing tool structures. Making future simulation tool maintenance, further tool expansions and new system components or simulation scheme integrations easy. VISUALQEC was implemented to make full use of the power and advantages of OOP.

3.4 A flexible, cross industry system simulation tool design

The success of all system simulation tools is inherently dependent on the efficient design of the basic simulation tool structures and simulation procedures. Because of the power of OOP, the thermal and energy system simulation tool designer has the means to “virtually” create

software objects or elements that closely resemble the “real” world elements. One of the most critical elements of any system simulation tool is the system components. Visually, system components are those individual building blocks that represent a specific “real” system component within the “virtual” simulation world. For every “real” system component within a specific energy system layout, there should thus be a comparative “virtual” system component within the simulation tool environment (interface). See Figure 3.4 and Figure 3.5.

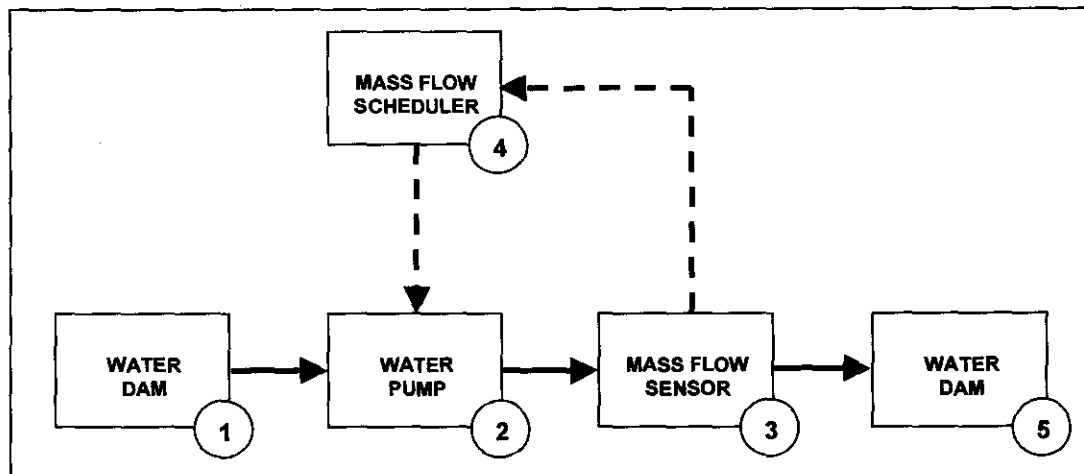


Figure 3.4: Typical “real” system layout

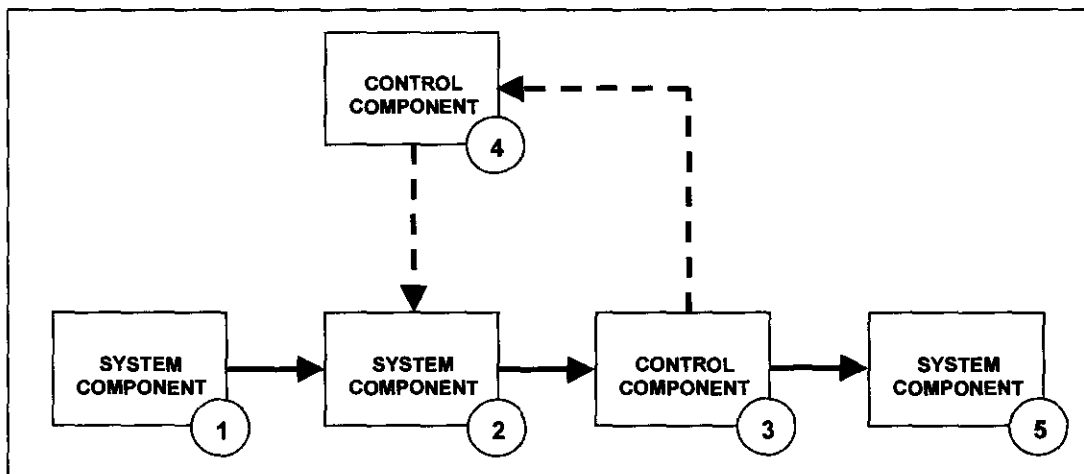


Figure 3.5: Typical “virtual” system layout

Figure 3.5 shows the graphical representation of the “real” system of Figure 3.4 on the system simulation user interface. From Figure 3.5, it can be seen that a typical system layout consists of various system components (1,2,3,4,5) and system connections. Every individual system

component within a system simulation tool consists of four major parts. These four parts are all critical for the configuration and simulation of the individual system component. These four critical parts are, IN ports (1), the component interface (2), the system component model (3) and the various OUT ports (4). See Figure 3.6.

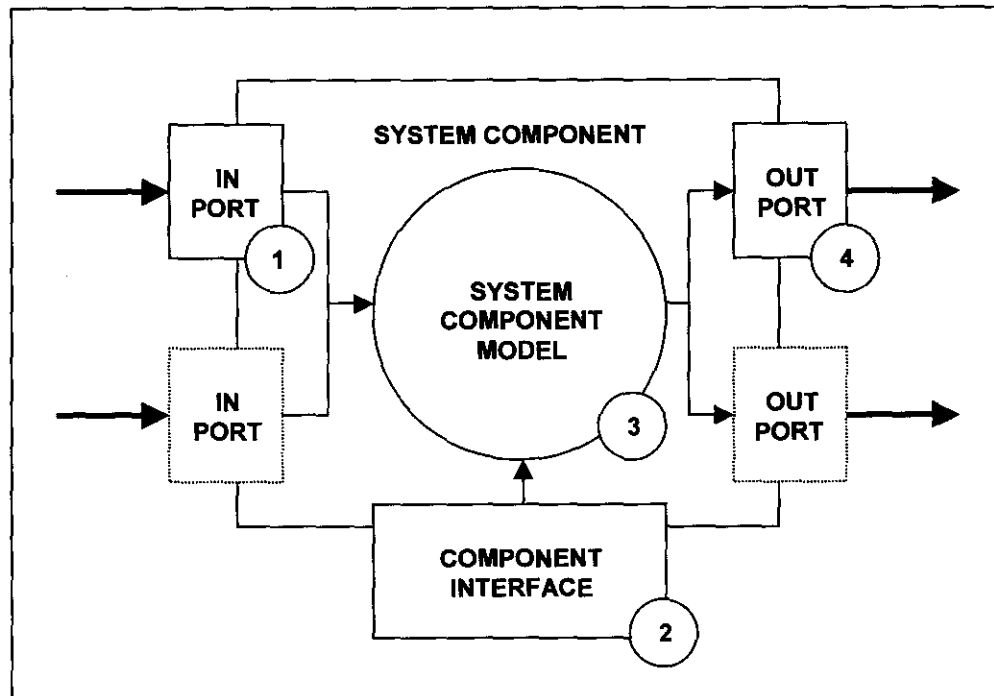


Figure 3.6: The system component

A port within a system component (Figure 3.6) is that entity that holds certain properties or values, usually of a specific nature, to be used within the system components internal simulation model (3). Mathematically, ports hold the values for the various variables needed to “solve” the system component model. In a thermal and energy system simulation tool, IN ports (1) hold the control, water and air specific properties that are used in the solution of the thermodynamic models (3) of the various thermal and energy system components. The resulting values from the solution of the system component model (3) are then passed onto the system components OUT ports (4).

Thus, from Figure 3.6 a typical system component functions as follows. The IN ports (1) and component interface (2) provide the necessary input and model calibration values to the system component model (3). After the solution of the system component model for every time interval, the resulting values are placed into the system components OUT ports (4).

These OUT port (4) values are transferred to the IN ports (1) of the various other system components within a specific system layout by means of system connections. See Figure 3.7.

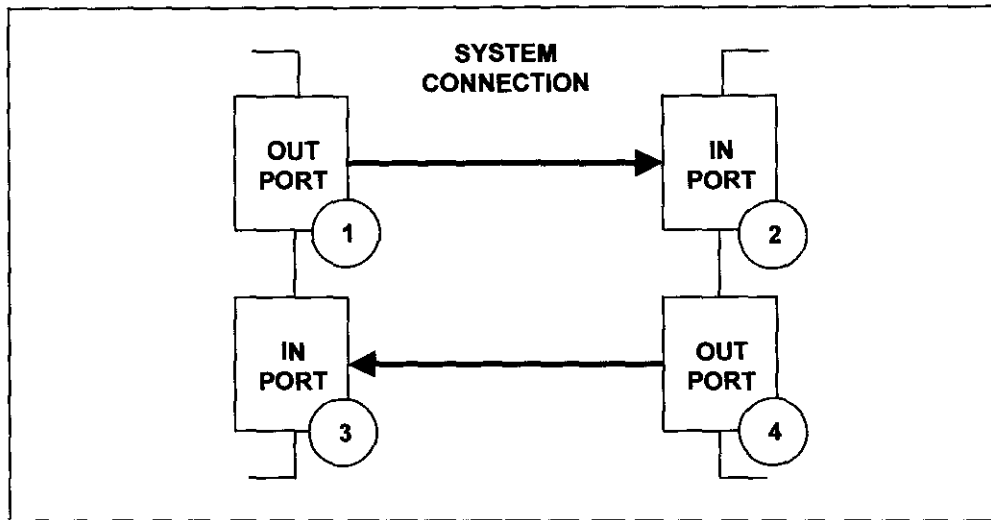


Figure 3.7: System connections

Figure 3.8 shows the typical types and properties of ports that are found within a thermal and energy system simulation components.

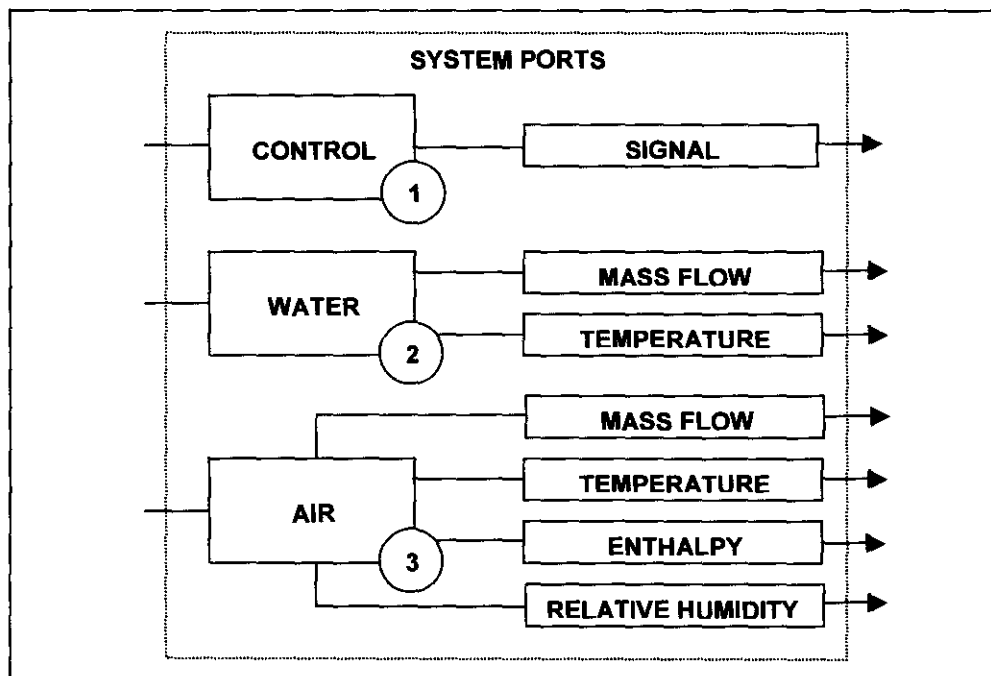


Figure 3.8: Thermal and energy system ports and properties

After the various system components have been selected, configured (calibrated) and ports connected. The system components and system connections are used by the system simulation engine too simulate the current system configuration over time. The system simulation engine essentially consists of four major parts. The system control simulator (2), mass flow simulator (3), system component simulator (4) and the dynamic integrated thermal and energy system simulation scheme (1). See Figure 3.9.

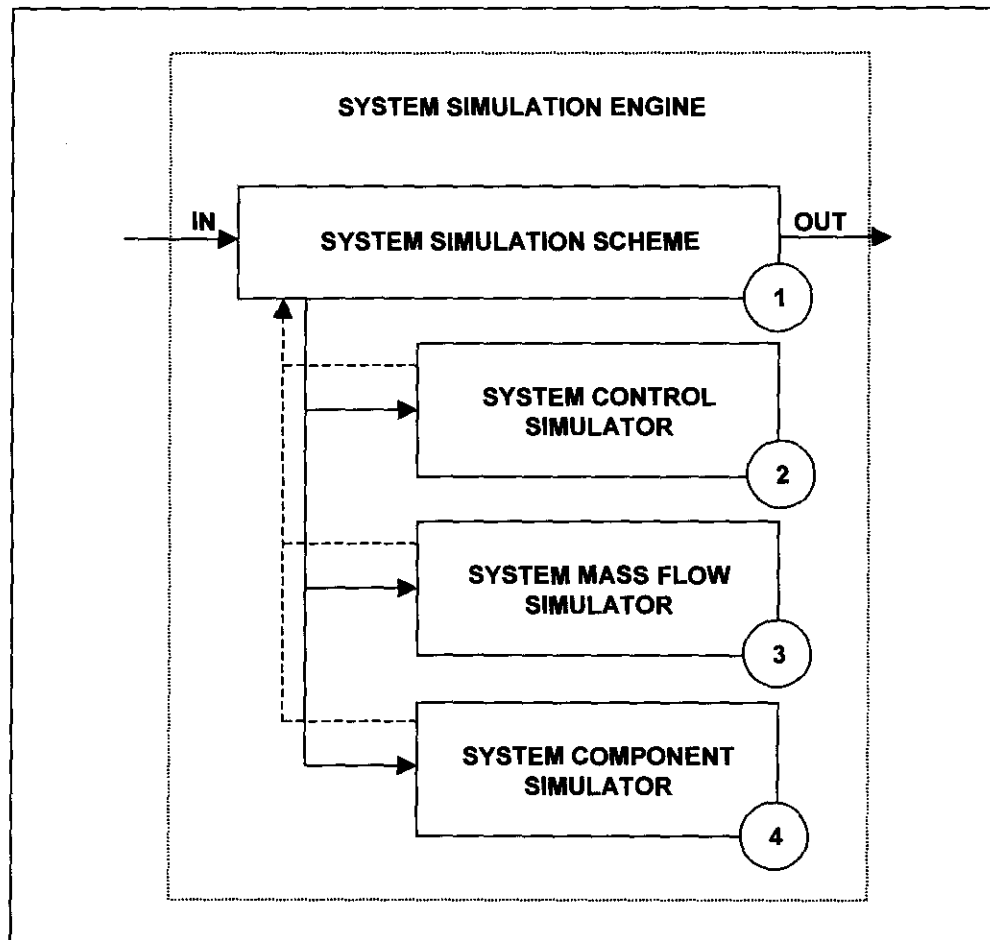


Figure 3.9: The system simulation engine

The system simulation engine uses the system simulation scheme (1) to manage the various parts (2,3,4) of the simulation process. The system simulation scheme ensures that the correct system components and connections are distributed at the right time too the correct next engine part within the simulation. Each of the various simulator parts i.e. control (2), mass flow (3) and system component (4) are then solved with their own internal best method mathematical or logical processes.

The VISUALQEC system simulation engine follows the following fixed thermal and energy system simulation scheme. See Figure 3.10.

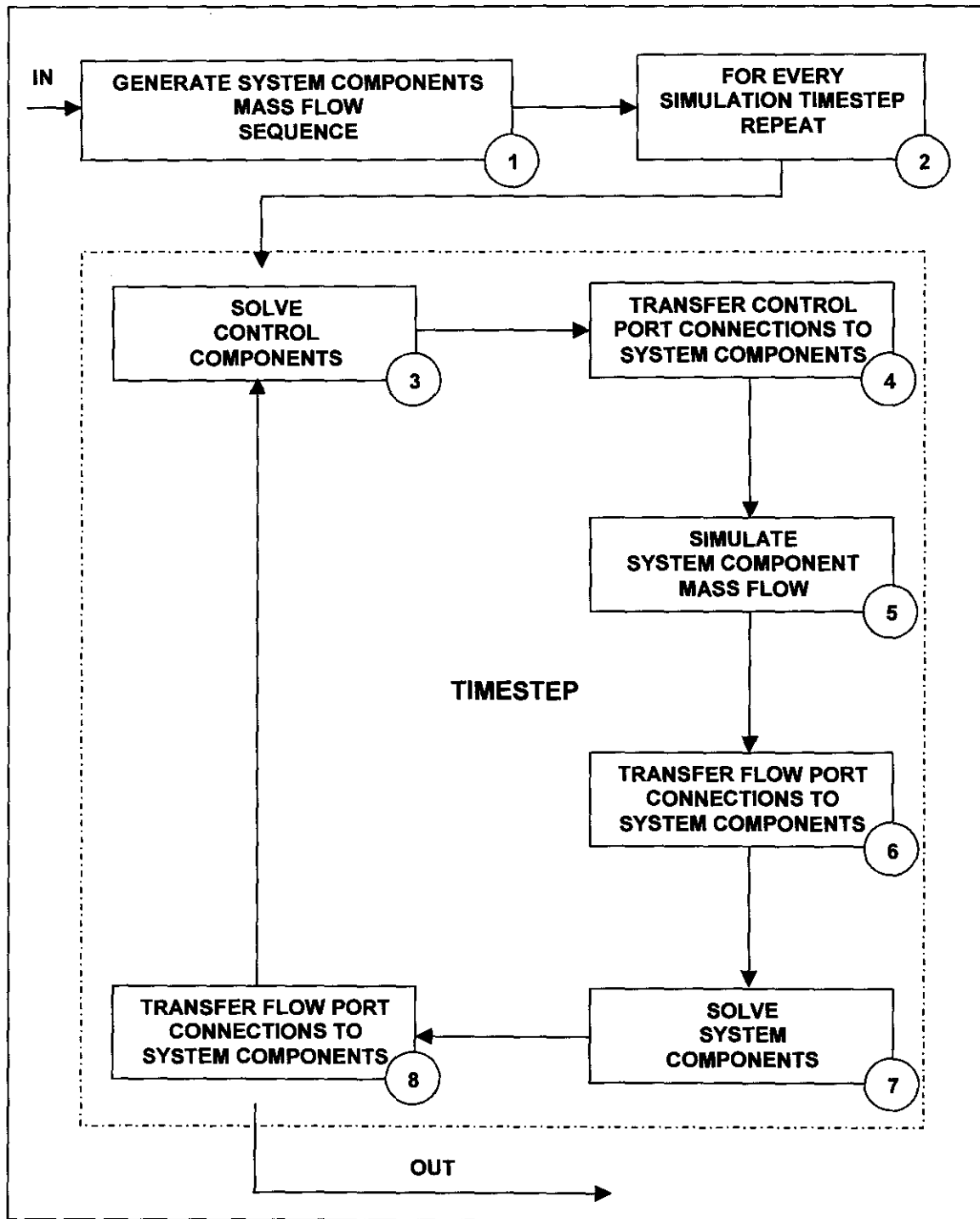


Figure 3.10: The dynamic integrated thermal and energy system simulation scheme

3.5 A new user interface design

The simulation user interface is that part of the simulation tool available to the user for interaction. The user interface is responsible for the configuration, reconfiguration and presentation of information needed or results supplied by or to the simulation engine element. The requirements as discussed in section 2.10 and 2.11 state that the user interface must be component based in order to configure any system type or configuration. It must include an option to allow the easy integration of future system components. The user interface should also supply the user with the easiest possible methods to configure new or existing system components. The interface must allow for the typical easy inclusion of pre-constructed system units and complete system types to decrease the configuration and construction time. See Figure 3.11.

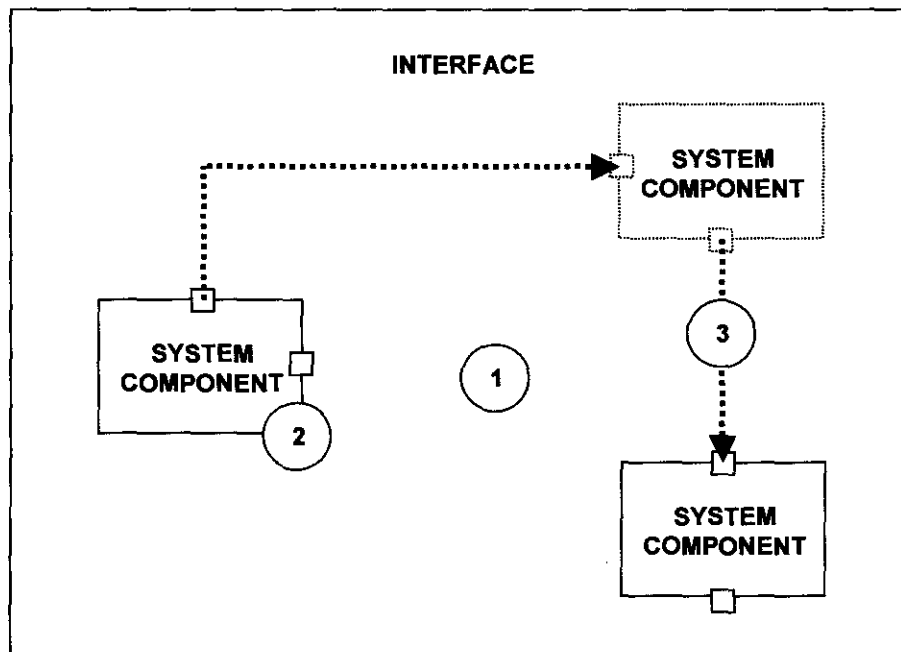


Figure 3.11: The user interface requirements

In the simulation tool the user interface is responsible for managing four separate processes. These processes are system configuration (1) (Figure 3.11), system component configuration (2), simulation parameter configuration and end-result interpretation (3). These processes all work together to configure, reconfigure and interpret the proposed solution provided by the simulation engine. It is the work of the user interface to complete and integrate these processes in the fastest most efficient manner possible.

various existing simulations tools, such as QUICKCONTROL, the user is prompted to select a system component group. The component is then configured according to the make, performance characteristics and operating conditions, by selecting the appropriate configuration from a pre-constructed component database. Through the use of the WIN32 API, the drag and drop of pre-configured system components, onto the project workspace is made possible. This eliminates the need for any system component database. This addresses the need for an easy expandable system model collection.

By moving away from a system component database a further increase of overall simulation tool stability is gained. Some of the other advantages include no version control on databases, no incorrect database connections, no database management and no database errors. By designing each component as a separate entity contained within itself the necessary coefficients, variables and information to correctly and accurately specify all of itself (object orientated), it is possible to totally move away from any database. See section 3.3. By implementing the above design as interface for a cross industry thermal and energy system simulation tool, the efficient simulation of these various thermal systems is possible. For a pictorial overview of the implementation of the user interface for VISUALQEC see Appendix B. Figure 3.13 a summary representation of the implementation of the proposed interface.

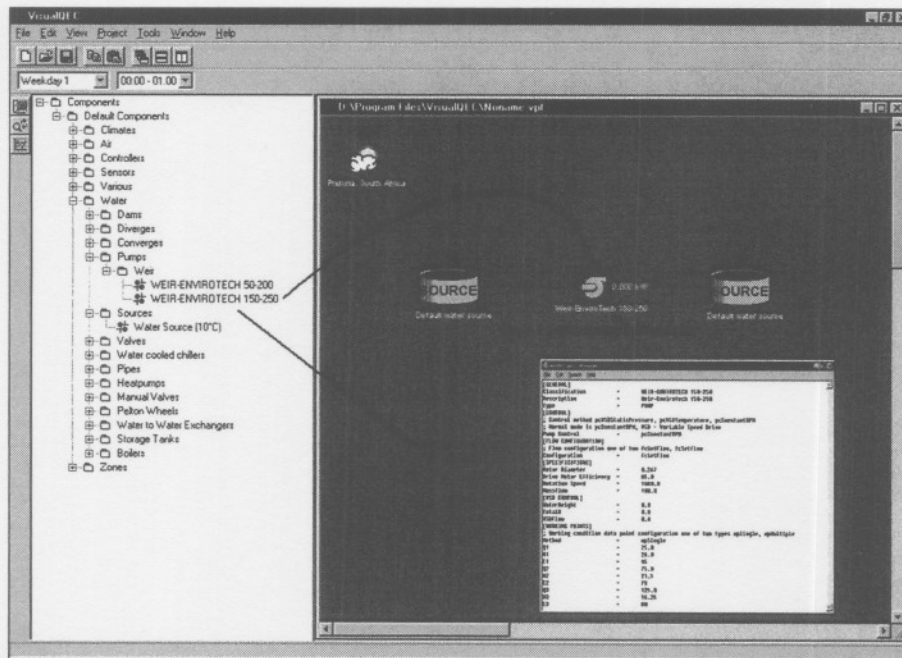


Figure 3.13: Representation of the implementation of the thermal and energy simulation interface

3.6 A new simulation engine design

The simulation engine can be presented as that part of the integrated system simulation tool not visible to the user. The simulation engine forms the driving force behind the simulated solution for the simulation tool. The engine encapsulates all the required mathematical routines, simulation procedures, and logical programming needed for the successful completion of an integrated simulation. It can be viewed as the central and mathematically most involved stage of the simulation process. Over the years, most of the work done in the field of simulation concentrated on the design and implementation of successful simulation engines. As discussed in section 2.9, various open source and public domain engines exist.

From section 2.10 and 2.11, the requirements for a successful system simulation engine state that, mathematically, system component models must be a combination of fundamental principles and empirical correlation coefficients. The simulation procedure, scheme and system component models must make use of explicit equation methods to ensure mathematical stability and simulation solvability. The simulation engine must furthermore be able to simulate any control strategy efficiently and accurately. Typical pre-constructed control units must be readily available as part of the overall simulation process.

The simulation engine stands central to the simulation process. Without the engine, no result for a configured simulation can be generated. The simulation engine can be divided into two main parts: the system component models and the system solution model. The former contain the internal mathematics upon which all system components in the simulation are based. To ensure a stable system simulation, it is imperative that these models must consist mainly of fundamental thermodynamic and first principle equations for calculating operating conditions based on performance and empirical correlation coefficients for the required component.

The detailed derivation of all these models falls outside the scope of this study. For more information see Rousseau [4], Lombard [6], van Heerden [5] and Arndt [7], with further reference to Stoecker [11] for the way in which these system component models are derived. Appendix A gives a summary of the most important system component models used within the thermal and energy system simulation field. Table 3.1 and 3.2 contain a summary of the various system control and system components found within VISUALQEC.







	Controller – scheduler		Controller – step
	Controller – PID		Controller converge
	Sensors		Controller diverge

Table 3.1: VISUALQEC control components

	Climate		Water dam
	Air source		Water source
	Air t-piece converge		Water t-piece converge
	Air t-valve converge		Water t-valve converge
	Air t-piece diverge		Water t-piece diverge
	Air t-valve diverge		Water t-valve diverge
	Air heater		Water-water exchanger
	Air fan		Water pump
	Air damper		Water pipe
	Air-cooled chiller		Water-cooled chiller
	Air-air heat exchanger		Water cooling/heating coil
	Building zone		Water storage tank
	Water-air cooling tower		Pelton turbine
	Water-air bulk air cooler		Water manual valve
	Water valve		

Table 3.2: VISUALQEC system components

Traditionally, the simulation process contained within the system simulation scheme consists of first solving the mass flow through the system and then numerically, or explicitly, solving the thermal characteristic and energy consumption of the various system components. In the new cross industry system simulation tool the simulation process essentially stays the same.

At the beginning of every simulation time interval the mass flow, and then the thermal characteristics and energy consumption, are solved. The new unique contribution is the way in which mass flow through the system is approached. In the past a numerical solution scheme for solving the mass flow of a network was used. This becomes an involved and extremely cumbersome process to be done at each time interval of the simulation process. Because of the numerical nature, some system configurations are unstable, or take far too long to converge to a stable or accurate solution. The need to replace this unstable solution scheme with some or other stable simulation algorithm becomes apparent.

The new system simulation engine implemented in VISUALQEC makes use of such a new procedure. This new procedure follows a logical flow procedure, which guides the user to inherently specify the unknown flow variables throughout a network. At the beginning of the simulation process a system component order is generated through which the simulation engine proceeds at the start of each new time interval of the simulation process. For a deeper understanding of the theory and implementation of this mass flow simulation procedure see section 3.7.

From Figure 3.10 the basic simulation scheme without the simulation of dynamic elements such as control consist of the following:

1. Graphically build the required configuration from the existing system components.
2. Generate the mass flow simulation sequence through the system (1).
3. At the beginning of each time interval of the simulation timeline, simulate the system mass flow (5).
4. Solve each explicit system component for a number of iterations within a specific time interval (7). This allows each system component to reach steady state.
5. Repeat steps 3 and 4 for every number of time intervals required for the simulation.

For the new system simulation engine, as can be seen from the above steps, nowhere is it necessary for any or set of differential equations relating the system variables to be solved. This explicit system simulation scheme dramatically improves the stability, speed and simulation efficiency without compromising accuracy.

To simulate control dynamically, a time constant is added to each steady state component model. Van Heerden [3] discusses various other methods for simulating dynamic control. The complete system simulation scheme (Figure 3.10) shows that the system control models are solved at the beginning of each time interval. This method is used by QUICKCONTROL and was found to be the easiest, mathematically stable solution available. The proposed simulation engine designed for VISUALQEC can be classified as a discrete, deterministic, integrated and dynamic cross-industry system simulation tool.

3.7 Simulating mass flow through integrated thermal and energy systems

To mathematically solve any theoretical “real” flow network is quite cumbersome, complex and often very time consuming. Therefore many thermal and energy system simulation tools [5] utilise the approach where flow is specified in the various system connections by the system components and not calculated for each time step. In constant mass flow networks the mass flow rate for each system component is specified by that system component itself. However, for a dynamic system simulation scheme, such as required by VISUALQEC, the mass flow rates could change over time due to the effect of control on certain system components i.e. valves, pumps, fans etc. These dynamic system components thus have an effect on the mass flow through the various other static system components.

These dynamic system components must therefore be utilised to set-up the correct, changing mass flow through the different network branches for all the other static system components. QUICKCONTROL makes use of such a flow specifying technique but does not allow for open loop network systems. The techniques used in QUICKCONTROL are also limited to typical flow networks mostly found only in the building industry. The need for creating a cross-industry, efficient and mathematically stable mass flow simulation procedure capable of simulating any complex open and closed loop network efficiently, is apparent. The following system component types are used to set up flow in a configured system network: pumps, valves, fans, converges and diverges. All these system components are referred to as flow

components. Pumps, valves and fans are the only system components that can furthermore specify mass flow and only one of them is needed per open or closed loop. Converges and diverges are used to add up flows or to specify mass flow fractions. Table 3.3 contains a summary of the various flow components found in VISUALQEC.














	Air t-piece converge		Water t-piece converge
	Air t-valve converge		Water t-valve converge
	Air t-piece diverge		Water t-piece diverge
	Air t-valve diverge		Water t-valve diverge
	Air fan		Water pump
	Air damper		Water valve
			Water manual valve

Table 3.3: VISUALQEC flow components

To set up a given system configuration, the user interface is used. The various system components, including the flow components (table 3.3), can be dragged and dropped into position. The system components of various ports are now connected to each other in the direction of the mass flow by a system connection. As discussed in section 3.3, system connections connect the outlet condition and mass flow of the previous system component to the inlet condition of the next system component (Figure 3.14). This implies that a system component receives its input for each time step from the system connection connected to its inlet port. The system conditions, mass flow and control values must be solved for each time step in the following sequence:

1. Obtain all the control output signals according to the conditions of the previous time step and store it in the control system connections.
2. Use the mass flow simulation procedure to set the flow of all the system connections for the current time step.
3. Now utilise an energy solver to solve all the various system components and obtain their output conditions.

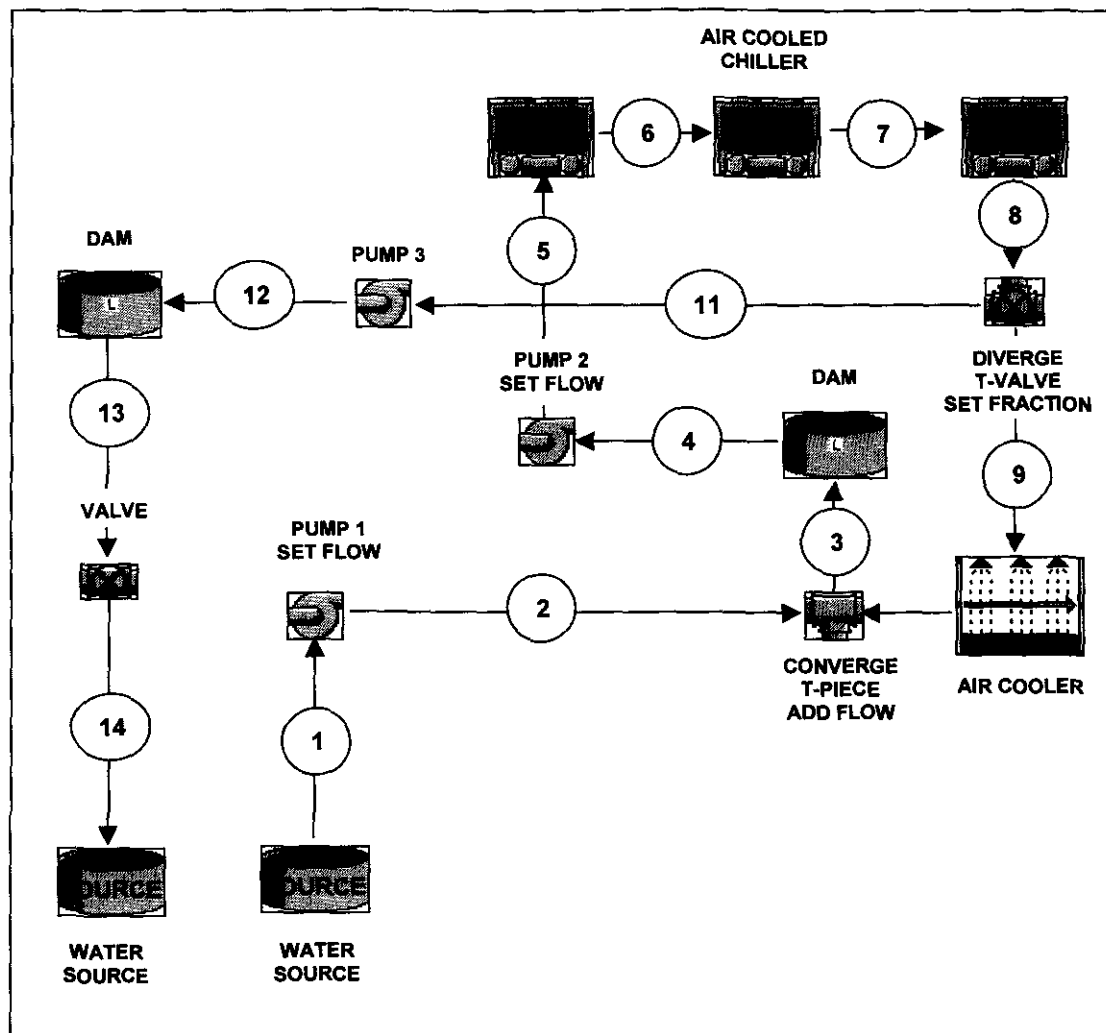


Figure 3.14: Diagram of a mine surface cooling plant

The mass flow in the configured system can be set for each time step by implementing the following mass flow simulation procedure:

1. **Pumps (Set flow):** Search for all the pumps set as flow components in the system. Pumps can be either flow components or just system components for energy calculation purposes. Set the flows of the pumps to the respective control component flows. Search forwards and backwards from each pump until the search finds another flow component or system break (e.g. dam, source) and set the flow of the system connections found to the flow specified by that pump.
2. **Valves and fans:** Follow the same procedure used by the pumps to specify the flows of the system connections.

3. Converge (Set fractions): Get the mass flow value from the system connection connected to the outlet port and calculate the inlet mass flows according to the specified control component fractions. Search backwards from the system connections connected to the inlet ports of the converge until the search finds a flow component or system break and set the flow of all the system connections found to the respective flow fractions of the converge.
4. Diverge (Set fractions): Get the mass flow value from the system connections connected to the inlet port and calculate the outlet port mass flow according to the specified control component fractions. Search forward from the system connections connected to the outlet port of the diverge until the search finds a flow component or system break and set the flow of all the system connections found to the respective flow fractions of the diverge.
5. Converge (Add flow): Get the mass flow values from the two system connections connected to the inlet ports and add them up. Search forwards from the system connections connected to the outlet port of the converge until the search finds a flow component or system break and set the flow of all the system connections found to the added flows of the converge.
6. Diverge (Add flow): Get the mass flow values from the two system connections connected to the outlet ports and add them up. Search backwards from the system connections connected to the inlet port of the diverge until the search finds a flow component or system break and set the flow of all the system connections found to the added flow of the diverge.

When configuring a system, the user must decide on the following depending on the flow network and configured flow control strategies:

1. Pumps and fans can either be used as flow components or just system components.
2. Diverges and converges can either be used to add flows or set fractions. In VISUALQEC, this distinction is made by t-valves and t-pieces. Any diverge or converge of type t-valve sets fractions whilst any diverge or converge of type t-piece add flow.

The converges and diverges must either be solved in the correct sequence to eliminate iterations, or the procedure must be run a number of times to solve all the system connections.

To illustrate the mass flow simulation procedure, consider the diagram of a typical mine surface cooling plant (Figure 3.14). Applying the procedure as discussed, generates the following sequence of system components and system connections. See Figure 3.15.

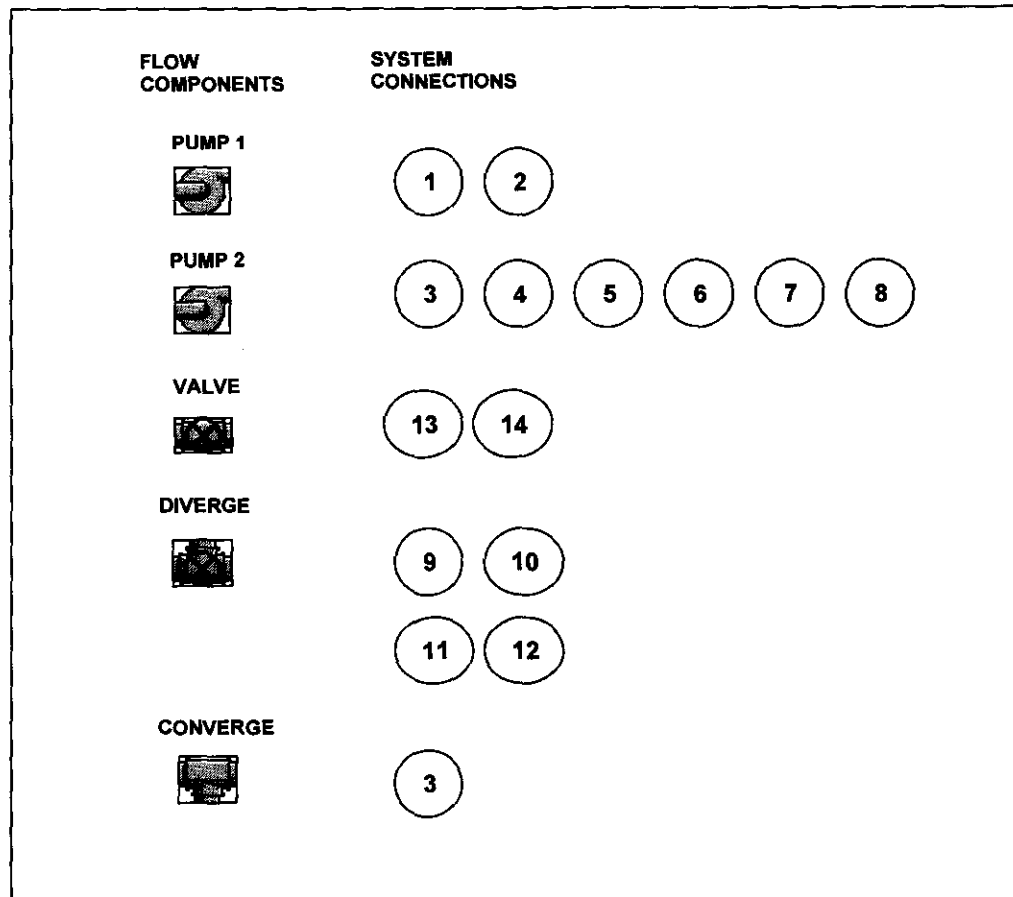


Figure 3.15: Solution to simulating mass flow

Flows are set to the system connections by the flow components in the sequence of the list. The flow values of the flow components can either be set constant over time like in the example of Figure 3.14 or the flow values can change due to changing system control outputs. The mass flow rates of pumps, valves and fans can be controlled from system conditions. This is practically achieved by adding a control sensor and controller loop to the output connection of a system component. Diverges and converges which are set on set fraction, can also be controlled from system conditions. Pump 3 in Figure 3.14 is utilised as a system component and not a flow component to include the thermal performance and energy consumption of that pump in the simulation. This simulation of mass flow procedure is effectively implemented within the VISUALQEC system mass flow simulator (3)(Figure 3.9).

3.8 Resolving further system simulation pitfalls

Traditionally, to solve any thermal and energy system, the system simulation engine is at some stage required to solve the set of numerical equations that describe the relationship of the various system components (section 2.3). Usually, numerical techniques such as Newton-Raphson are used to solve these sets of equations. These equations are most often non-linear implicit equations, which often doesn't have an easy, trivial solution.

QUICKCONTROL, for example uses, the powerful public domain routine SNSQ, which is based on a quasi-Newton method with the Jacobian, determined via the Broyden method [12,13]. This procedure has been developed over the last couple of decades and provides a powerful and general method for solving sets of non-linear equations. It has good convergence properties but, in common with all other methods for solving non-linear equations, requires a reasonable initial guess value for the solution. To comply with the requirements for a cross-industry thermal and energy system simulation tool as outlined by section 2.10 and 2.11, the need arises for the stable, explicit solution of the system.

Because all system components implemented for VISUALQEC make use of explicit steady state component models, as well as not solving but simulating the mass flow through the various system components, it is possible to "not" solve any set of differential or systems of equations, as is traditionally the case. This new unique simulation procedure allows every system component, through iteration, to reach steady state. A fast, stable, accurate solution of the system energy can then be achieved. By using this method, the very calculation intensive, and often unstable and time consuming numerical solver needed for the solution of the system, can be replaced. VISUALQEC through its design makes use of no numerical solver. This could only be achieved through the explicit modelling of system components, as well as the simulation of the mass flow through the system.

3.9 An illustrative example

To summarise the design of the new cross-industry thermal and energy system simulation tool and accompanying simulation scheme (Figure 3.10) the following small example is presented. See Figure 3.16. Figure 3.16 shows a diagram of a typical simple HVAC/VC system.

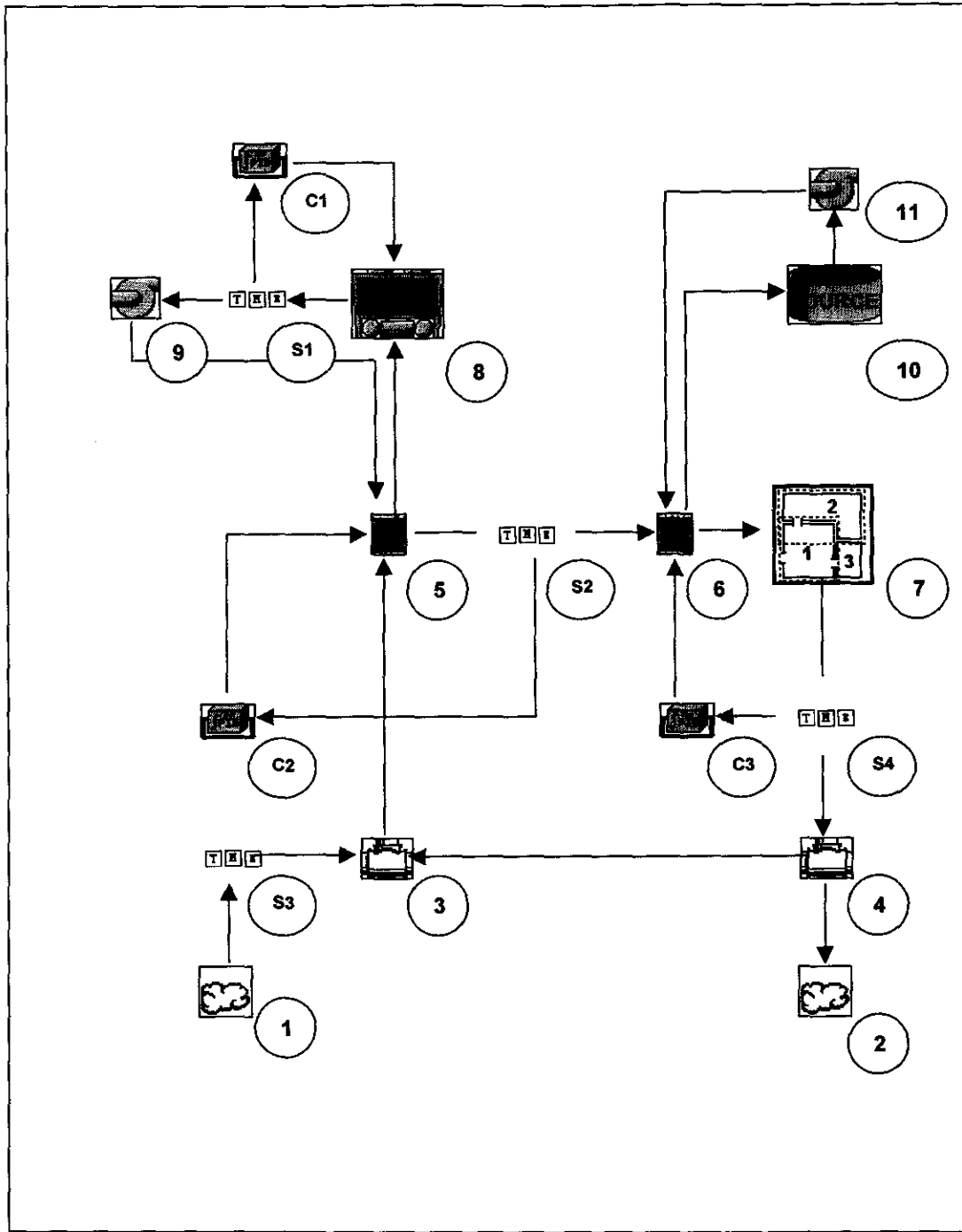


Figure 3.16: Diagram of a typical HVAC/VC system

Figure 3.16 consists of three main loops. There is the main air loop formed by the climate (1), diverge (4), converge (3), heating/cooling coil (5), heating/cooling coil (6) and the zone (7). The warm water loop formed by the heating/cooling coil (6), water source (10) and water pump (11). The cold-water loop formed by the heating/cooling coil (5), the chiller (8) and the

cold water pump (9). As discussed in section 3.4 and diagrammatically shown in Figure 3.10, the simulation process proceeds as follows:

1. The system control components i.e. sensors and controllers are sent to the system control simulator (2).
2. The various flow components i.e. air diverges, converges and in this example the flow specifying zone is sent to the mass flow simulator (3).
3. All remaining system components are sent to the system component simulator (4).
4. The mass flow simulator uses the new simulation of mass flow procedure introduced in section 3.7 to generate the sequence in which the various system components and system connections mass flow need to be simulated.
5. The initial starting value for all the system control and system components are set by the system control simulator, system mass flow simulator and system component simulator.
6. The system control simulator takes the input port values of the various sensors (S1, S2, S3, S4) and according to the specified control models (C1, C2, C3) generate the appropriate output port values.
7. These output port control values are transferred via the connected system connections to the connected in ports of the system components.
8. The system mass flow simulator runs through the simulation mass flow sequence generated at the start of the simulation.
9. The various connected system connections mass flow values are set to the specified mass flow values.
10. Using the new iterative steady state solution method discussed in section 3.8, the system component models (5,6,7,8,9,11) thermal and energy is calculated.
11. The calculated values from the system component models are transferred to the system component out ports. These out port values are transferred to the connected system connection in ports.
12. The system simulation scheme repeats the procedure from step 6 to 11 until the required number of time steps has been reached.

After the simulation, the “state” or thermal condition of every system component at a specific moment in the simulation process can be viewed graphically.

3.10 Physical implementation of the new system simulation tool

VISUALQEC was designed and developed to specifically cater for the needs and requirements identified for an efficient cross-industry thermal and energy system simulation tool. VISUALQEC was developed over a two-year period in an object orientated programming environment, namely Delphi 5,6 [10]. VISUALQEC also serves as testament to the success of the proposed new system simulation scheme, its implementations and concepts discussed throughout this study. Specific attention to a new procedure for simulating mass flow, as well as a new iterative system component solution was given. These two changes ensure an unconditional mathematically stable and fast simulation. For a pictorial overview of the implementation of VISUALQEC and some code extracts see Appendix B.

TEMM International (Pty) Ltd. sponsored the development of VISUALQEC in the interest of enhancing thermal and energy efficiency practices across multiple energy consuming industries. Chapter 4 and 5 shows the verification and application of the new simulation tool in both the building as well as the mining industry.

3.11 References

- [1] Department of Minerals and Energy of the Republic of South Africa, *Draft Energy Efficiency Strategy of the Republic of South Africa*, <http://www.dme.gov.za/>, 2004.
- [2] Department of Minerals and Energy of the Republic of South Africa, *South African White Paper on Energy Policy* <http://www.dme.gov.za/>, 1998.
- [3] TEMM International (Pty) Ltd., The Boardwalk, Lakeside, Haymeadow Street, Faerie Glen, South Africa, 0043, 2001.
- [4] Rousseau P.G., *Integrated building and HVAC thermal simulation*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 1994.
- [5] van Heerden E. *Integrated simulation of building thermal performance, HVAC system and control*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 1997.
- [6] Lombard C., *Two-port simulation of HVAC systems, and object orientated approach*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 1996.
- [7] Arndt D.C., *Integrated dynamic simulations of large thermal systems*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 2000.
- [8] *VisualDOE*, Eley Associates, 142 Minna Street, San Francisco, California, 94105, USA, 2001.
- [9] Visual C/C++, Microsoft Corporation, <http://www.microsoft.com>, 2001.
- [10] Delphi 5, Borland Corporation, <http://www.borland.com>, 2000.
- [8] Els R., *Potential for Load Shifting in Ventilation and Cooling Systems*, Thesis presented in partial fulfilment of the requirements for the degree Master of Engineering, University of Pretoria, 2000.
- [11] Stoecker W.F., *Design of Thermal Systems*, McGraw-Hill Book Company, 3rd Edition, 1989.
- [12] Wait, R., *The Numerical Solution of Algebraic Equations*, John Wiley & Sons, 1979.
- [13] Churchouse, R.F., *Handbook of Applicable Mathematics*, John Wiley & Sons, 1981.

CHAPTER 4

BUILDING INDUSTRY VERIFICATION

Improving building heating, ventilation and air-conditioning (HVAC) system energy efficiency translates to monetary savings for the building owner. Although the benefits of increased energy efficiency is very important, energy saving measures should never compromise the indoor air quality (IAQ) of any building. To support ESKOM and the South African governmental efforts towards improving energy efficiency practices. The need for a new, easy to use, mathematically stable, efficient and accurate thermal and energy system simulation tool was identified. A new cross-industry thermal and energy system simulation tool, called VISUALQEC was developed to meet these requirements. To validate the success of VISUALQEC, a building energy retrofit study, using VISUALQEC and its underlying system simulation scheme, for a commercial building is presented.

4.1 Introduction

Improving building heating, ventilation and air-conditioning (HVAC) system energy efficiency translates to monetary savings for the building owner and less greenhouse gases being released into the atmosphere [1]. Although the benefits of increased energy efficiency is very important, energy saving measures should never compromise the indoor air quality (IAQ) of any building. The reason is that IAQ has a direct effect on the productivity of the building occupants. The cost associated with poor IAQ and the resulting loss of productivity far outweighs savings due to increased energy efficiency and the reduction of energy consumption.

Popular belief in the past was that good IAQ and energy efficiency were in direct conflict. In South Africa, studies by TEMM International (Pty) Ltd. have shown that approximately 50% of energy used by the commercial sector goes directly towards their HVAC systems [2]. This statistic clearly indicates that through improved HVAC system efficiency, a tremendous potential for saving on energy exists. A cost-effective way to improve the energy efficiency of a building HVAC system, without compromising indoor comfort, is by implementing better control [3]. However, when changing the control strategy of a system it is often difficult to predict the resulting changes in system energy consumption and the indoor comfort.

To achieve these predictions, a thermal and energy system simulation tool that can efficiently and accurately simulate the building with its HVAC system and controls in an integrated fashion, is required [4]. Although there are many system simulation tools available, they do not satisfy the requirements for integrated, stable and accurate system simulation needed by a typical commercial energy consultant. The need for a new, easy to use, mathematically stable, efficient and accurate thermal and energy system simulation tool was identified. A new cross-industry thermal and energy system simulation tool, called VISUALQEC was developed to meet these requirements.

The verification and validation of a thermal and energy system simulation tool forms an integral part of ensuring the applicability and accuracy of a system simulation to real-life situations. To this end, a validation energy retrofit study, using VISUALQEC and its underlying system simulation scheme for a commercial building, is presented. In this case study, more emphasis was placed on the ease of doing such a retrofit study using

VISUALQEC. A more detailed verification of individual HVAC system components was done for the mining case study, which is discussed in chapter 5.

4.2 Verification procedure

Usually, a comparison of the results of a “calibration and verification” simulation with actual measured system conditions will indicate the accuracy of the created simulation model. The “calibration” simulation ensures that the current status of the building HVAC system can be simulated correctly, so that cost savings and energy efficiency improvements are realistic. The configured simulation model is considered adequately calibrated when the predicted daily total building demand load is within a 10% range of the actual measured building electricity demand.

The “verification” simulation is performed in order to verify the accuracy of the simulation model’s energy consumption over a typical year. The yearly energy consumption and maximum electricity demand of the simulation is then compared to the actual measured building results. The best way to ensure that the simulation model represents the real building is to verify the energy consumption of the various HVAC system components, and to check whether the simulated trends of the energy profiles are realistic.

Energy consulting group HVAC International (Pty) Ltd was contracted by TFMC (Pty) Ltd to investigate the potential for energy savings throughout the Telkom Campus. To further verify the benefits and practical implications of using VISUALQEC, VISUALQEC was used as main system simulation tool. The results and conclusion of this retrofit study is presented as sufficient proof and verification of the successful application of VISUALQEC to the building industry.

4.3 Case study: Telkom Data Building

The Telkom Data Building is situated in the central business district (CBD) of Pretoria, South Africa and forms part of the Telkom Campus group. The building can be divided into four main building sections, each of which is used differently and is supplied by it’s own air-conditioning (HVAC) system. These four main sections are, an office section, a ground floor

area, first floor area and a lower ground section. Table 4.1 is a summary of the most important building and HVAC system details for the Data Building.

BUILDING DESCRIPTION	
Building name	Data building
Building description	Commercial building
Building location	Pretoria CBD
Number of floors	16
HVAC system	Multi zone
Cooling plant	Water cooled and air cooled
Air distribution	Variable air volume (VAV) and Constant air volume (CAV)
Control System	Building management system (BMS)

Table 4.1: Telkom Data Building description

4.3.1 System description

Office area

The office area is situated on floors 2 through 13. Each floor is used as office space in an open plan arrangement. The main plant room of this zone is situated on the 14th floor. The chillers and the main air-handling unit (AHU) are in this room. The cooling towers are on the same level on the roof.

The system is designed as a full fresh air system. Outdoor air is fed to the main cooling coil where it is cooled, and then supplied to induction units situated on the sides of each floor. The main cooling coil is a free coil, which means that the coil is not controlled and that it operates continuously at full capacity (the coil is not equipped with a bypass damper).

Electric heaters heat the air when required. In addition to the main AHU, the zone is equipped with induction units and ceiling mounted fan coil units (FCU's). The induction units are supplied from the main AHU and are situated on the sides of the floors, and the FCU's re-circulate the air inside the zone and is situated in the centre of the floors. Three water-cooled chillers cool the water. The chillers supply the main cooling coil, the induction units and the FCU's with chilled water. The condenser water is cooled by three cooling tower located on the roof of the building.

First floor

The first floor area consists mainly of office space. There is one small room used for equipment (called Dark Room). The air conditioning of the office area is done by three AHU's. One of the AHU's is in a plant room on the first floor, and the other two are on the podium. The units consist of a cooling coil, fixed speed fan and electrical heaters. Two air-cooled chillers situated on the podium cool the water. The equipment room is conditioned by two down-blow units, and is fed of the main system. The chilled water to these units is supplied from the chillers.

Ground floor

The ground floor of the building is used for office space and technical equipment. The air is conditioned by 10 down blow units located in a service compartment next to the outer wall of the zone. Return air is fed to the down-blow units through a common return air ceiling void. In the units the air is cooled by cooling coils and humidified if required. No heating is installed. The chilled air is then supplied to the zone through a pressurized under floor area. Four air-cooled chillers are situated on the podium above the first floor area.

Lower ground printing section

The lower ground area is used for printing of the telephone invoices for the Telkom's customers. The area is mainly a large open plan workspace where the printing equipment is located. A few smaller areas like offices and storerooms also form part of the area, as well as a separate conference room.

The area is supplied by four AHU's. The printing area of the lower ground section is supplied by a variable air volume system. Two air handlers with variable speed drive (VSD) fans, a cooling coil, economiser cycle and electric heaters feed the area. The conference room is supplied by a similar system. The fourth air handler supplies a storeroom. This unit consists of a cooling coil and constant speed fan, with a constant fresh air supply. One air-cooled chiller located on the podium supplies chilled water.

4.3.2 Current state of the system

During a building energy audit it is not unusual to find that the HVAC system does not operate according to designed intentions. This was also the case in this building. The differences between the designed and the current system, as well as any potential problems, are discussed in this section. It must be stressed that the aim of the study was not to find problems with the system, but to make recommendations as to the theoretical energy savings potential of the building through simulation.

Office area

As stated previously, the office area was designed as a full fresh air system with a free running coil. The fresh air is supplied to the air handler via a set of grills with fixed openings. During the winter the system did not operate as a full fresh air system. The maintenance personnel closed the fresh air supply grills, and opened the doors of the service shafts in the plant room. The system therefore operated as a fixed percentage return air system, with the only fresh air supply to the building that which leaks past the closures.

The electric heaters are also turned off because of dangerous temperatures that it obtains when on. The probable reason for this is the low airflow over the heaters, which also results in less than required heating available in the zone. During the summer, the system is turned back into a full fresh air system. The system currently operates as such.

In addition, the main cooling coil is blocked. Very little airflow is let through the coil. This will have an adverse effect on the potential of the system to maintain comfort in the building. This will be because of the reduced airflow over the main coil, also because less “high pressure” air is fed to the induction units, reducing its efficiency. In contrast, it will result in a decreased load on the chillers resulting in decreased energy consumption. During the project some of the compressors of each chiller were out of commission for servicing.

First floor

The first floor area is mainly designed as office space. Additional to this there is the equipment room, but this area is very small compared to the office area and is supplied by its own system air conditioning system. The whole of the first floor area, except the equipment room, is currently unoccupied. The office area is supplied by three AHU's, one situated on the floor and two located on the podium. All three are operating.

The first floor area was originally designed as a full fresh air system. The building facilitators changed this to a return air system. One of the three AHU's is operating on full return air. The two units on the podium share a fresh air inlet. For these units the return air dampers and fresh air dampers are fully open.

Two down-blow units supply the equipment room, which require chilled water from the chillers on the podium. However, the load of the equipment room is small compared to the chiller size, which then results in chiller cycling when only the down-blow units are operating. The other three AHU's are operating to apply additional load to the chiller, in order to keep it from cycling. This results in relatively high-energy consumption in relation to the required heat load.

Ground floor

The ground floor is currently operating below designed capacity. However, there are possible plans to increase the amount of people and equipment in the near future, which will increase the load on the system.

The result of this is that more down-blow units are operating than is required. However, some of the down blow units cannot be turned off. The reason for this is because of short-circuit backflow into the unit, causing the fan to rotate backwards. If the unit is switched on, then strain is placed on the fan and fan motor, decreasing its operating life.

Lower ground printing section

The lower ground air handlers and chillers are currently operating to design conditions. Although this space is used as a printing area, it was not designed as such. When the printing machines are operational, the installed HVAC capacity is not enough to maintain comfort levels in the zone. The only change to the system from the original is the removal of humidifiers in the AHU's and the electric re-heaters in the ducts.

BMS

The main drawback of the building management system (BMS) is the extent of the data available in the BMS. The most common issue would be that most of the equipment has a control signal in the BMS, but not an operational signal (e.g. a positional, or on/off signal). Therefore, the operator can see the BMS command to the component, but not the response of the unit to the command.

4.3.3 System simulation configuration

For the Telkom Data Building, the simulation model was divided into four zones, with each individual building section being modelled as a separate zone. These four zones included:

1. Office area, ranging from the second floor to the thirteenth floor.
2. The vacated first floor area.
3. The ground floor area, including the office section and equipment section.
4. The lower ground printing section.

The following assumptions were further made to obtain the relevant zone area, occupancies and load inputs:

1. For the zone lights, assume a value of 22 W/m^2 . This figure was obtained from a TFMC lighting audit. This value also corresponds to typical design values and was verified by spot-checking in the building.

2. Assume 330 persons working in the building. This value was obtained from the building telephone directory.
3. Assume that 80% of the occupants operate PC's.

Table 4.2 shows the climate over a typical 24-hour period that was used for the verification and energy simulation of the Telkom Data Building. This data was calculated from yearly data taken over the past 20 years. Temperature and relative humidity (RH) is shown.

BUILDING CLIMATE DESCRIPTION				
HOUR	SUMMER °C	SUMMER RH%	WINTER °C	WINTER RH%
0	18.4	73.3	10.0	71.1
1	18.0	75.4	9.4	72.8
2	17.5	77.3	8.8	74.9
3	17.1	78.8	8.3	76.6
4	16.7	80.4	7.8	78.9
5	16.4	82.1	7.2	80.8
6	16.3	82.9	7.0	81.8
7	17.0	80.7	7.1	82.0
8	18.9	74.3	9.0	76.0
9	20.6	68.6	13.1	62.8
10	22.2	62.7	16.1	53.5
11	23.6	57.8	18.3	45.8
12	24.6	53.7	19.9	39.9
13	25.4	50.3	20.8	36.0
14	26.0	48.1	21.4	33.0
15	26.1	46.8	21.3	32.3
16	25.8	47.4	20.9	32.5
17	25.0	49.2	19.6	35.1
18	23.8	52.7	17.2	42.3
19	22.3	57.7	15.3	48.8
20	21.1	62.8	13.6	56.1
21	20.2	66.4	12.3	61.4
22	19.5	69.3	11.5	64.8
23	18.9	71.8	10.6	68.1

Table 4.2: Building simulation climate data.

To verify the accuracy of VISUALQEC, the dynamic integrated nature of the complete HVAC system of the Telkom Data Building was modelled and simulated. Figure 4.1 is a representation of the four main zones air circuits. Figure 4.2 is a representation of the four main zones water circuits.

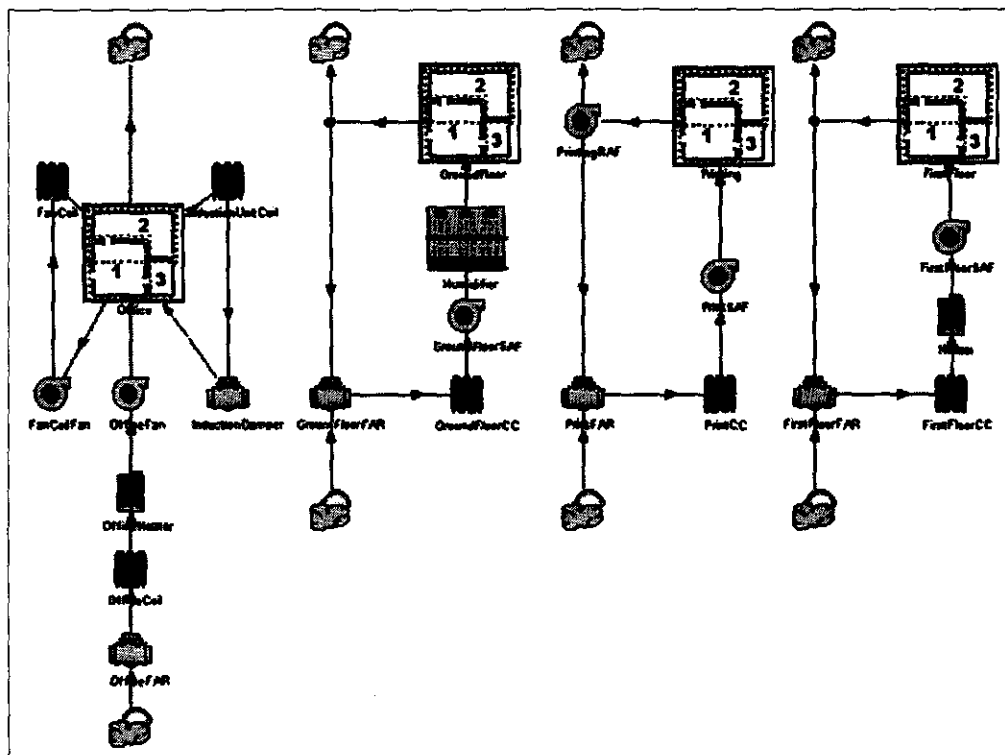


Figure 4.1: Simulation model air circuit layout

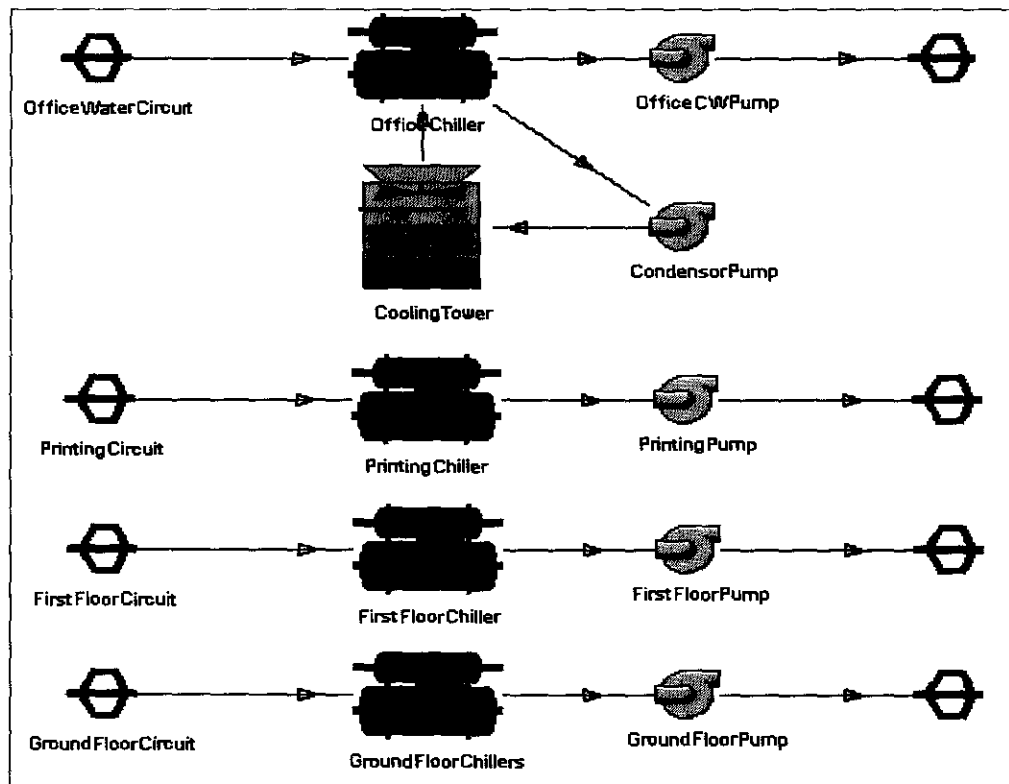


Figure 4.2: Simulation model water circuit layout

4.3.4 System operation verification

The complete the “calibration” and “verification” simulation the actual measured system data was used to verify the predictions obtained by the simulation program. A specific typical 24-hour day of system operation was chosen for calibration and verification purposes. The simulation was performed according to the exact operating schedules of the measured system. The predictions of this simulation were compared to a specific typical 24-hour day of normal operation, chosen from the most complete measured data. For this study 7 September 2004 was used. As is most often the case in commercial buildings, access and availability of measured system data is very limited.

The following figure graphically shows the energy consumption of the office area chiller. See Figure 4.3. The measured data shown is the average hourly energy consumption calculated from the typical 24-hour day of normal operation. The difference between the simulated and measured system energy consumption is 9%. This is within the 10% required accuracy band.

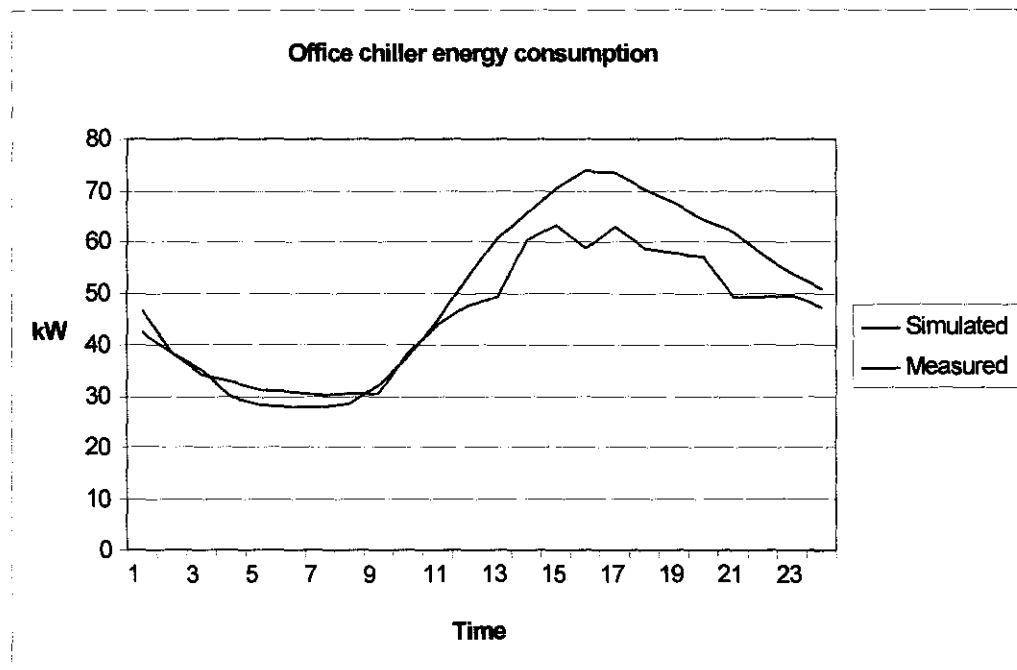


Figure 4.3: Office chiller energy consumption

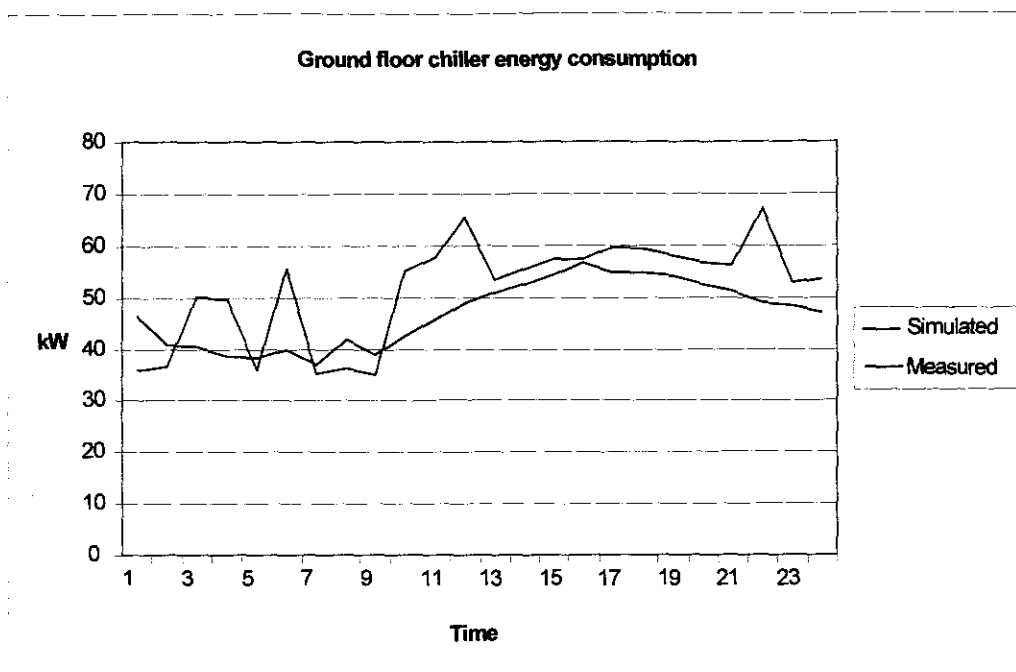


Figure 4.4: Ground floor chiller energy consumption

For the annual energy consumption, VISUALQEC predicts an electricity cost of approximately R 1, 553, 000.00, including Maximum Demand (MD). When compared to the daily energy consumption of the whole Telkom Campus group, the Data Building accounts for approximately 10% of the group energy consumption.

4.3.5 Retrofit options and further verification results

Making use of VISUALQEC, an end-user energy cost breakdown was determined. This identified the large energy consumers in the building. By investigating these consumers, possible energy efficiency measures and savings potential can be investigated. See Appendix C for more detailed layout of the Telkom Data Building, building energy cost and building HVAC system energy cost breakdown.

From the building energy cost breakdown it was found that 52.77% of the total of 7941.72 MWh of energy consumed by the building goes towards the HVAC system. It further showed that 43.40% of this total goes towards cooling and 30.61% towards the ventilation of the HVAC system. With these figures in mind, a number of retrofit, or control change options to improve these figures can be investigated. The retrofit

options investigated are discussed per main building section. The assumptions made for the specific retrofit or control change is also discussed.

Office area

The retrofits investigated in the office section include fixing of the main coil, changing of the system to a return air economiser control, evaporative cooler, light scheduling, fan scheduling, chillers scheduling and night set point setback.

1. Repair main cooling coil: Firstly, the influence of repairing the main AHU cooling coil was investigated (Currently the main cooling coil is blocked). This would give a more accurate picture of how the office system should currently be operating. The simulation model with the functioning coil was then used as the base year scenario for the rest of the office retrofit comparisons. The energy consumption of the system with the fixed coil would be more than with the blocked coil. The new airflow of the coil, and therefore the zone, was calculated from typical design airspeed of 2m/s over the coil face area.
2. Return air economiser control: Changing of the system to a return air system would constitute many equipment changes to both the building and HVAC system. This would include the placing of air ducting in the service shafts leading to the main plant room, motorised dampers to regulate return air, fresh air intake, and return air fans to maintain static pressure in the office areas. This option would most likely not be feasible from a payback period perspective, but will also have a positive effect on the comfort in the building and maintenance costs on the system. This option was also investigated and proposed by building maintenance staff in recent years.
3. Evaporative cooler: Evaporative cooler will reduce load from the chillers with the use of free cooling. This option will however have penalties from a maintenance point of view, through personnel time and equipment costs such as water treatment. It will also lead to increased water consumption.

4. **Night setback/set point drift:** Night set point setback involves the slacking of the building set point during unoccupied times. The cooling set point is raised and the heating set point is lowered during these times. The assumption was made that there would be no cost implication in this retrofit as it could be set up on the BMS by maintenance staff. The set points were drifted by the following values:

Weekday		Saturday		Sunday	
00:00 – 04:59	5°C	00:00 – 05:59	5°C	00:00 – 23:59	5°C
05:00 – 05:59	2°C	06:00 – 06:59	3°C		
06:00 – 06:59	1°C	06:00 – 06:59	1°C		
18:00 – 18:59	2°C	12:00 – 12:59	1°C		
19:00 – 19:59	3°C	13:00 – 13:59	2°C		
20:00 – 23:59	5°C	14:00 – 14:59	3°C		
		15:00 – 23:59	5°C		

Table 4.3: Building set point drift settings

The following figures show the summer and winter simulation for a weekday, Saturday and Sunday. From the figures it can be seen that the building is on set point after the set points are set to normal ranges.

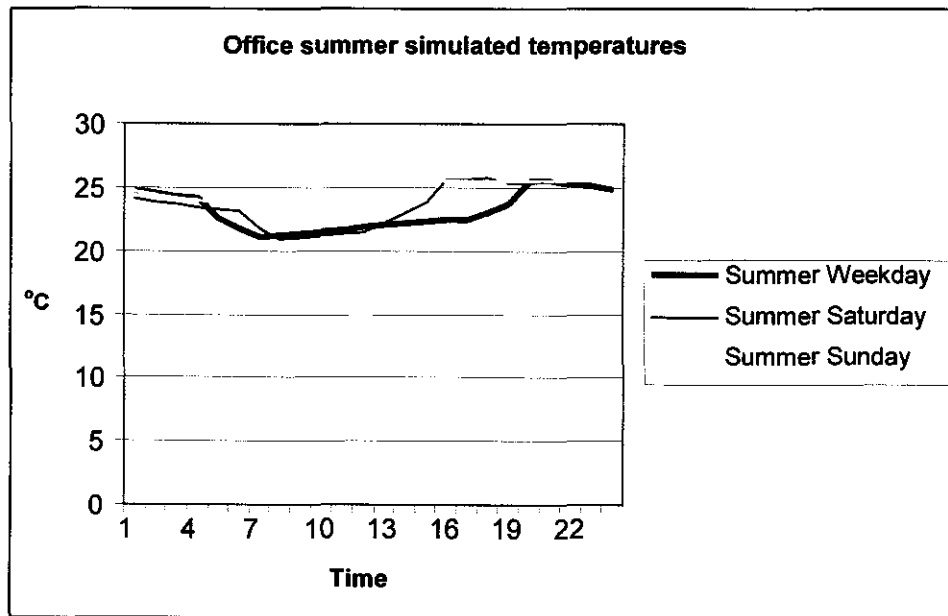


Figure 4.5: Summer simulated temperatures with set point drift

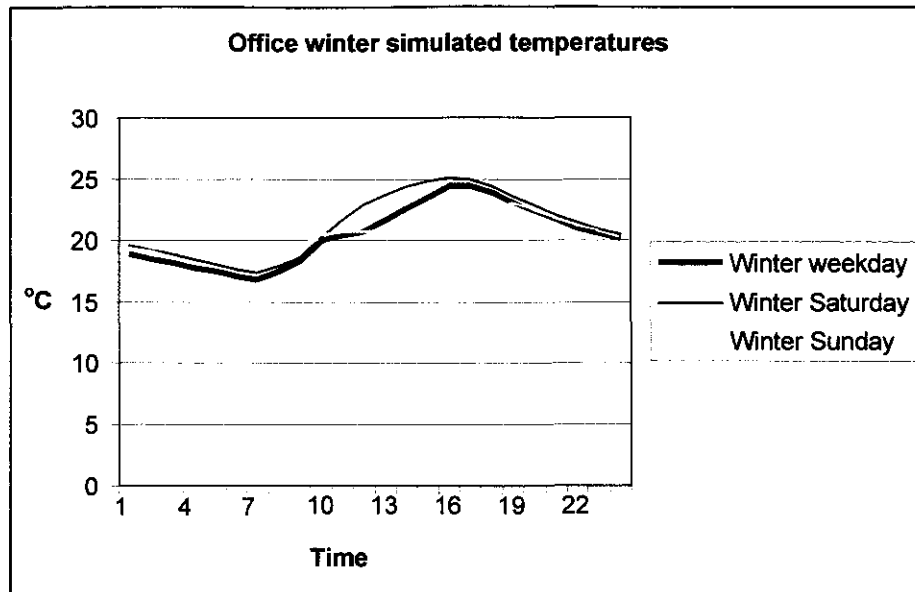


Figure 4.6: Winter simulated temperatures with set point drift

Fan scheduling: Fan scheduling involves turning the fans off during occupied times. The assumption was made that there would be no preliminary cost implication in this retrofit as it could be set up on the BMS by maintenance staff. Normally maintenance personnel are not in favour of switching equipment off, and would prefer set point drift. The following times were used:

Weekday		Saturday		Sunday	
00:00 – 04:59	OFF	00:00 – 05:59	OFF	00:00 – 23:59	OFF
05:00 - 17:59	ON	06:00 - 13:59	ON		
18:00 – 23:59	OFF	14:00 – 23:59	OFF		

Table 4.4: Building fan scheduling times

Chiller scheduling: Chiller scheduling involves turning the chillers off during occupied times. The assumption was made that there would be no preliminary cost implication in this retrofit, as it could be set up on the BMS by maintenance staff. In the simulation model the chillers were turned on one hour before the fans, to allow the chillers to lower the water temperature to the required set point. This would reduce the workload on the system as compared to the case where all equipment is turned on simultaneously. Normally maintenance personnel are not in favour of switching equipment off, and would prefer set point drift. The following times were used:

Weekday		Saturday		Sunday	
00:00 – 03:59	OFF	00:00 – 04:59	OFF	00:00 – 23:59	OFF
04:00 - 18:59	ON	05:00 - 13:59	ON		
19:00 – 23:59	OFF	14:00 – 23:59	OFF		

Table 4.5: Building chiller scheduling times

The following figures show the summer and winter simulated temperatures, for a Weekday, Saturday and Sunday case. From the figures, it is evident that the zone temperatures are in the comfort range (21°C - 24 °C) during office hours.

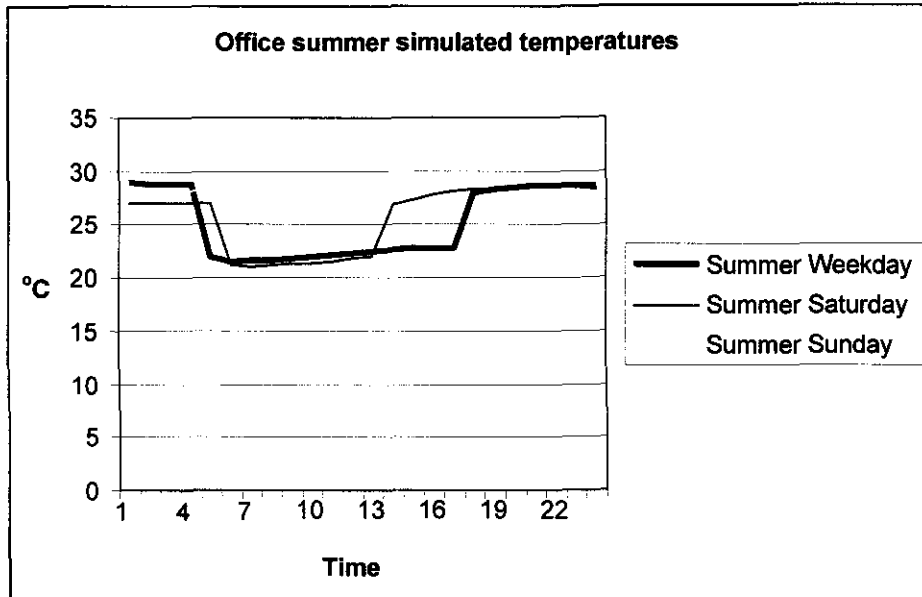


Figure 4.7: Summer simulated temperatures with fan and chillers scheduling

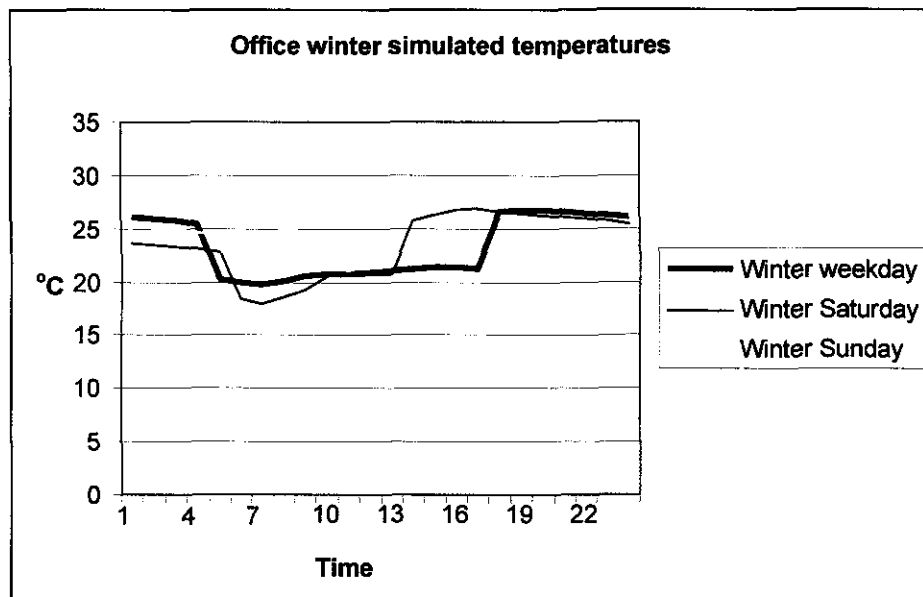


Figure 4.8: Winter simulated temperatures with fan and chillers scheduling

First floor

Currently the entire HVAC system of the first floor is operating to supply only a very small section of the floor. The office section of the floor is vacated because of health issues. There are plans, however, to move the equipment in the equipment room to another venue. There is no time scale to this plan and it may happen soon, or only after several years. There are no definite plans, known by the building facilitators, to reoccupy the floor. In the event that the equipment remains in place for some time, it may be efficient to install a dedicated system in the room and permanently switch off the remaining floor HVAC system. It is therefore proposed to install a small split unit air conditioner in the equipment room.

Ground floor

The ground floor area was originally designed as an equipment area. It has an installed cooling capacity of 1000kW. Less than half of the capacity is currently used. However, the fan of a down blow unit cannot be turned off because the large under floor supply pressure causes backflow that causes the fan to turn backwards. If the fan is turned on again, the fan and fan motor is placed under strain, which has a negative maintenance implication. Therefore, the retrofit proposed is to install a short piece of ducting away from the units to lower the amount of backflow because of the high under floor pressure. Only the number of required units required maintaining set point can then be used. There are plans to increase capacity in the ground floor area. No savings will then be possible since all units will be required. For the simulation it was assumed that the floor is currently running at half capacity, therefore half the units could be switched off.

Lower ground printing section

The lower ground HVAC system was not designed to make provision for the printing equipment currently installed in the area. The equipment therefore struggles to maintain the required temperature in the area. In addition, the printing equipment cannot be switched off even during times of no printing. These two facts require the HVAC system to operate continually and no savings is possible.

4.4 Building savings potential and conclusion

Numerous retrofit options and combinations thereof were investigated through simulation during the project. The most important financial results are displayed in the following tables. The cost calculations are based on an active energy cost of 12.01c/kWh, and a MD cost of R51.51.

The table below shows the current building electricity cost for the current building simulation model, and the building simulation with the fixed main coil. From the table it is evident that by fixing the cooling coil it will have an adverse effect on the annual building electricity cost.

Electricity cost savings			
Description	Cost (R)	Cost savings (R)	% Savings
Current building	1544055.77	-	-
Repair coil	1553633.14	-9577.37	-0.62

Table 4.6: Cost of current building operation vs. cost of repaired coil

The following tables give the retrofits pertaining to the office section. The base year scenario is the cost of the current building simulation, with the assumption that the main coil is repaired. The cost of the coil repair is therefore NOT included in the price of the retrofits listed. The retrofit installation costs are also a general figure and are not based on actual contractual quotes. It also operates on the assumption that the lights can be scheduled. The following retrofit and combination retrofit options was investigated:

1. Fresh air ratio economiser control
2. Evaporative cooler
3. Night setback/set point drift
4. Fan scheduling
5. Fan and chillers scheduling
6. Economiser, evaporative cooler, set point drift and light scheduling
7. Economiser, evaporative cooler, fan and chiller scheduling, and light scheduling

Electricity Cost Savings			
Description	Cost (R)	Cost Savings (R)	% Savings
Base year	1553633.14	-	-
Option 1	1545016.02	8617.13	0.55
Option 2	1541290.63	12342.52	0.79
Option 3	1531705.40	21927.74	1.41
Option 4	1533847.28	19785.87	1.27
Option 5	1506224.30	47408.84	3.05
Option 6	1464848.06	88785.08	5.71
Option 7	1437915.32	115717.82	7.45

Table 4.7: Cost savings of the office individual retrofits

Financial Analysis					
Description	Project Cost (R)	Direct Payback Period (Months)	Discounted Payback Period (months)	Loan Rate (%/year)	Net Present Value (R)
					Year 5
Option 1	200000.00	279	100	12.00	-168937.19
Option 2	50000.00	49	66	12.00	-5508.00
Option 3	0.00	0	0	0.00	44492.00
Option 4	0.00	0	0	0.00	44492.00
Option 5	0.00	0	0	0.00	44492.00
Option 6	250000.00	34	41	12.00	70050.36
Option 7	250000.00	26	30	12.00	167136.86

Table 4.8: Financial analysis of the simulated office individual retrofits

Tables 4.9 and 4.10 give the retrofit information for the simulation of the first floor and ground floor areas. The base year scenario is taken as the current building operation, without the repaired main office coil. The reason for this is that these specific retrofits are not dependant on the correct functioning of the main office system.

Electricity cost savings			
Description	Cost (R)	Cost savings (R)	% Savings
Base year	1544055.77	-	-
First floor	1465958.70	78097.07	5.06
Ground floor	1473539.67	70516.10	4.57

Table 4.9: Cost savings of the first floor and ground floor retrofits

Financial analysis					
Description	Project cost (R)	Direct Payback Period	Discounted Payback Period	Loan rate (%/year)	Net Present Value (R)
					Year 3
First floor	50000.00	8	9	12.00	137575.99
Ground floor	40000.00	7	8	12.00	129367.78

Table 4.10: Financial analysis of the first floor and ground floor retrofits

The following tables give the results of the total combined retrofits. These include the first floor and ground floor simulations, as well as the combined office retrofits including the set point drift, as well as fan and chiller scheduling respectively.

Electricity cost savings			
Description	Cost (R)	Cost savings (R)	% Savings
Base year	1553633.14	-	-
Option 1	1174673.28	378959.86	24.39
Option 2	1153197.63	400435.52	25.77

Table 4.11: Cost savings of the office combined retrofits

Financial analysis					
Description	Project cost (R)	Direct Payback Period	Discounted Payback Period	Loan rate (%/year)	Net Present Value (R)
					Year 3
Option 1	340000.00	11	12	12.00	570197.65
Option 2	340000.00	11	11	12.00	621778.54

Table 4.12: Financial analysis of the office combined retrofits

It is assumed that the scheduling of the equipment will have zero capital input, as an existing maintenance contract on the BMS will be able to implement these retrofits. For practical purposes it was assumed that capital costs would be covered by a loan with an interest rate of 12% p.a. The project cost prices are general values and do not form part of a cost analysis of a concept design.

From the results it can be seen that the combination of all the retrofit options holds the biggest savings potential. Unfortunately, it also has a relatively long payback period compared to most of the other retrofits. In addition, some of the retrofits are not very practical, and based on assumption that is in turn based on predicted space usage, which may change.

The combined retrofit will realise a saving of 25% on the total yearly cost, with a payback period of approximately one year. The investment will have a net present value of about R620, 000.00 over three years.

The retrofit study done on the Telkom Data Building shows that VISUALQEC and the new simulation engine was capable of efficiently completing a typical building energy efficiency saving study. VISUALQEC was found to be extremely versatile in its allowed system configurations (retrofits) as well as unconditionally stable when simulating the various system components. Since this study, VISUALQEC and or the new VISUALQEC simulation engine (chapter 3) have been consistently used by HVAC International (Pty) Ltd. [6,7] to do various other energy retrofit and thermal and energy related work. VISUALQEC thus complies with all the requirements and criteria as outlined by chapter 2. It is also the first thermal and energy system simulation tool specifically designed for use by a mostly non-academic, commercial energy-consulting community.

4.5 References

- [1] Department of Minerals and Energy of the Republic of South Africa, *The Draft White Paper on the Promotion of Renewable and Clean Energy Development*, <http://www.dme.gov.za>, 2002.
- [2] Andersen J.J., *Cape Town Electricity Load Study and End-Use Segmentation*, Proceedings of the Seminal and Main Steering Committee Meeting Demand Side Management and Related Projects, DMEA, 1993.
- [3] van Heerden E., Mathews E.H., *Applying an Integrated Building and HVAC Simulation Tool to Successfully Predict Energy Savings in Buildings*, Fifth International Conference for the International Building Performance Simulation Association, 1997.
- [4] Arndt D.C., *Integrated dynamic simulations of large thermal systems*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, University of Pretoria, 2000.
- [5] National Electricity Regulator, *Electricity Supply Statistics for South Africa*, 2001.
- [6] HVAC International (Pty) Ltd., The Boardwalk, Lakeside, Haymeadow Street, Faerie Glen, South Africa, 0043, 2001.
- [7] Geysers M.F., *A new integrated procedure for energy and analysis of buildings*, Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor, Potchefstroom University, 2003.

CHAPTER 5

MINE INDUSTRY VERIFICATION

The verification and validation of a thermal and energy system simulation tool forms an important part of ensuring the applicability of the simulation to real-life situations. Kopanang is situated on the Free State side of the Gauteng – Free State border. This mine’s primary function is the removal of gold carrying ore from the gold carrying rock body. By using VISUALQEC, the surface cooling plant and underground pumping network of a typical South African gold mine was simulated. The results of this verification form the central theme of this chapter.

5.1 Introduction

The verification and validation of a thermal and energy system simulation tool forms an important part of ensuring the applicability of the simulation to real-life situations. To this end, a validation study is presented in this chapter. The case study was done on the thermal and energy consumption systems of a typical goldmine within the South African mining industry. Details of the system layout and the accompanying verification are given in the following sections.

5.2 Verification procedure

To verify and validate a system properly, it is necessary to physically measure the relevant information needed for the system. This information includes water flow rates, temperature trends, dam levels and electricity consumption measured over a certain period, at available, or specified time intervals. The measuring equipment needed was either already installed in the system, e.g. System Control and Data Acquisition (SCADA) systems, or additionally provided.

By using VISUALQEC, the system can be modelled by configuring the required sub-systems using the different component models of the simulation tool. Starting with a typical day, the measured data of the system can be compared to the end result data acquired from the mine thermal and energy simulation tool.

To be able to verify the dynamic integrated thermal and energy system simulation tool, VISUALQEC, it was necessary to establish a proper procedure for the verification process. This procedure needed to be systematic and was broken into seven steps.

1. A detailed schematic layout of the system was needed to establish the configuration of the simulation model and to determine the various measuring points needed for the verification purposes.

2. The measurements needed for verification purposes included the measuring of all underground dam levels, temperature trends, the water flow rate from the underground system to the surface cooling plant, pumping schedules of each pumping station and the electricity usage of all the power consuming equipment.
3. Equipment that was already installed in the system, e.g. SCADA, as well as additional measuring equipment was used to perform all the necessary measurements. Typical additional equipment included temperature and mass flow measuring probes.
4. A typical working day was selected from the measured data. This data was selected together with an appropriate measuring interval to cover enough working conditions of the pumping cycle.
5. The measured data was collected and sorted into useful formats. Along with the system measurements, the representative weather temperature (T) and relative humidity (RH) was collected. The weather measurements form the driving force for most of the air component models within the integrated thermal simulation.
6. The simulation was set up with the weather conditions and various cooling and pumping cycles of the typical day. The simulation was run and the simulated data was compared with the actual measured data.
7. Corrections and modifications were made to the simulation models and the simulation was repeated until the simulated data compared well to the measured data.

By using the above verification procedure the verification process was successfully performed.

5.3 Case study: Kopanang

Kopanang is situated on the Free State side of the Gauteng - Free State border. The primary function of the mine is the removal of gold carrying rock ore from the rock body. The process of removing this ore requires a constant supply of chilled water to various underground stopes. The chilled water is also used to cool the ambient supply air temperature to the various stopes through the use of a bulk air cooler (BAC).

Two of the major consumers of energy on this mine are the refrigeration (surface cooling plant) and the underground pumping network. The surface cooling plant is used to cool water and air into and out of the mine, whilst the pumping network is used to transfer this water throughout the various underground levels and stopes back to the surface cooling plant for re-refrigeration. Other consumers of large amounts of energy, which fall outside the scope of this verification, include the compressed air cycle and the man and rock winder systems.

5.3.1 System description

For the purpose of this verification study the Kopanang system is broken into two parts, namely the surface cooling plant and the underground pumping network that includes the various underground dams on 38 and 75 levels.

Surface cooling plant

The mine has an installed cooling capacity of 35 MW in its surface plant. Six Hitachi HM-22A water-cooled chillers provide this cooling capacity. The system was originally designed to work with eight machines but the current six machines can manage the total required cooling load efficiently. The surface plant system consists of the condenser cooling towers, pre-cooling towers and storage capacity through the various water storage dams. Figure 5.1 schematically represents the surface plant.

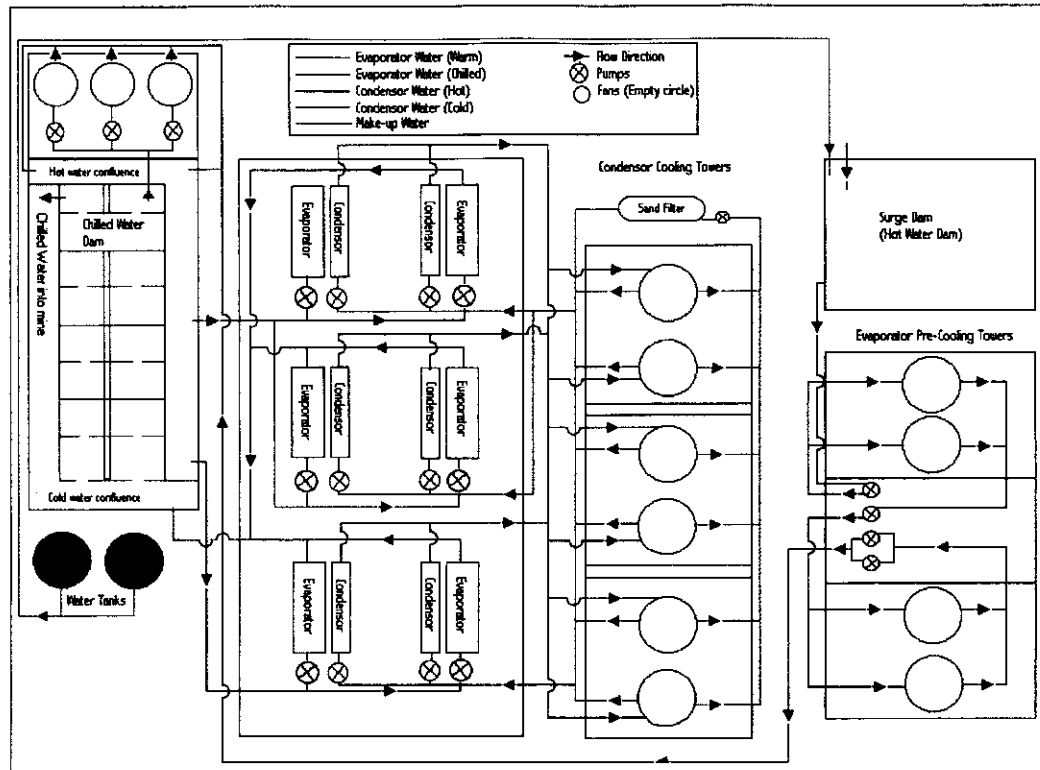


Figure 5.1: Schematic representation of the surface cooling plant cycle

The surface cooling plant system begins with hot water from the underground mine at 38 level being pumped into the surge dam. If the dam level of the surge dam is too low (usually because of evaporation) makeup water from the municipal water tanks is added to the system. The hot water from the surge dam at 27°C is then pumped through pre-cooling towers, where the ambient water temperature is lowered to about 14°C , into the hot water confluence side of the main storage dam. This water is mixed with water at 8°C that was used by the bulk air cooler (BAC) from the chilled water dam at 3°C , to cool the ambient air temperature into the mine.

The hot confluence water is then cooled by the water-cooled chillers (evaporator cycle) and returned to the cold confluence side of the main storage dam where it mixes at about 3°C with the chilled water dam. The three condenser cooling towers of the system support the cooling of the condenser water cycle of the water-cooled chillers. Appendix D contains all the specifications of the various components of the surface cooling plant.

Underground pumping network

The main dam above ground is the chilled water dam that serves as the input link for the chilled water from the surface cooling plant to the underground working areas. The hot surge dam on the surface serves as the output link between the underground working areas and the surface cooling plant. Figure 5.2 schematically represents the underground pumping network.

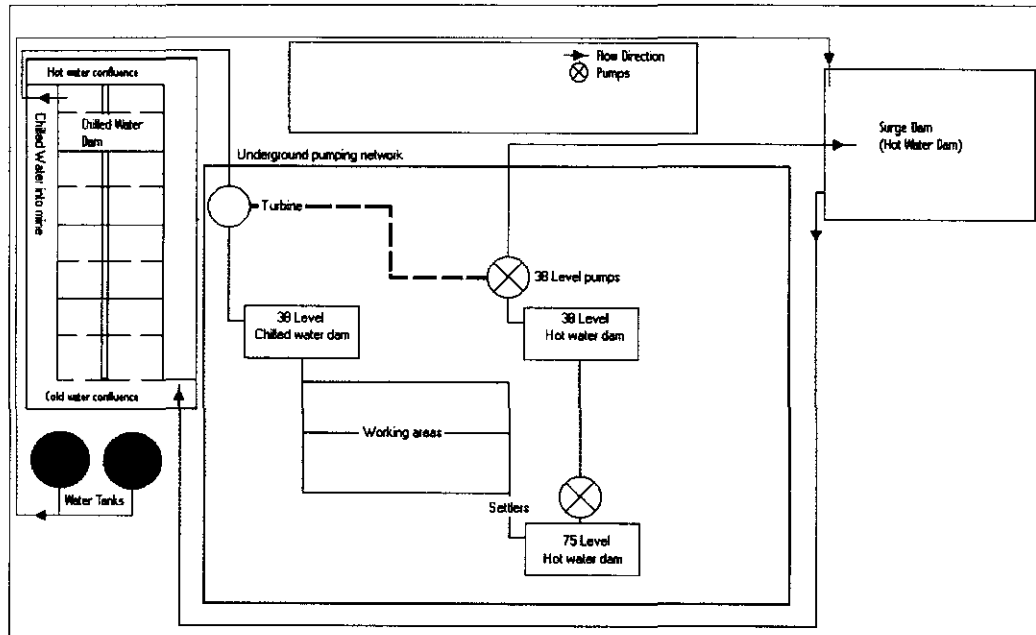


Figure 5.2: Schematic representation of the underground pumping network.

Water is sent into the mine from the chilled water dam on the surface through a turbine. The turbine generates additional energy for the water extraction pumps situated on 38 level. Water is pumped from the warm water dams on 38 level into the warm surge dam on the surface. Water is distributed through the various working levels underground where the used water then gathers into the settlers at the bottom of the mine at 75 level. The 75 level pumps then pump the water from the 75 level dams up into the hot water dams situated on 38 level.

5.3.2 System simulation configuration

To verify the accuracy of VISUALQEC, the dynamic integrated nature of both the surface cooling plant and the underground pumping network was modelled and simulated. Figure 5.3 is a representation of the complete integrated system as modelled with VISUALQEC.

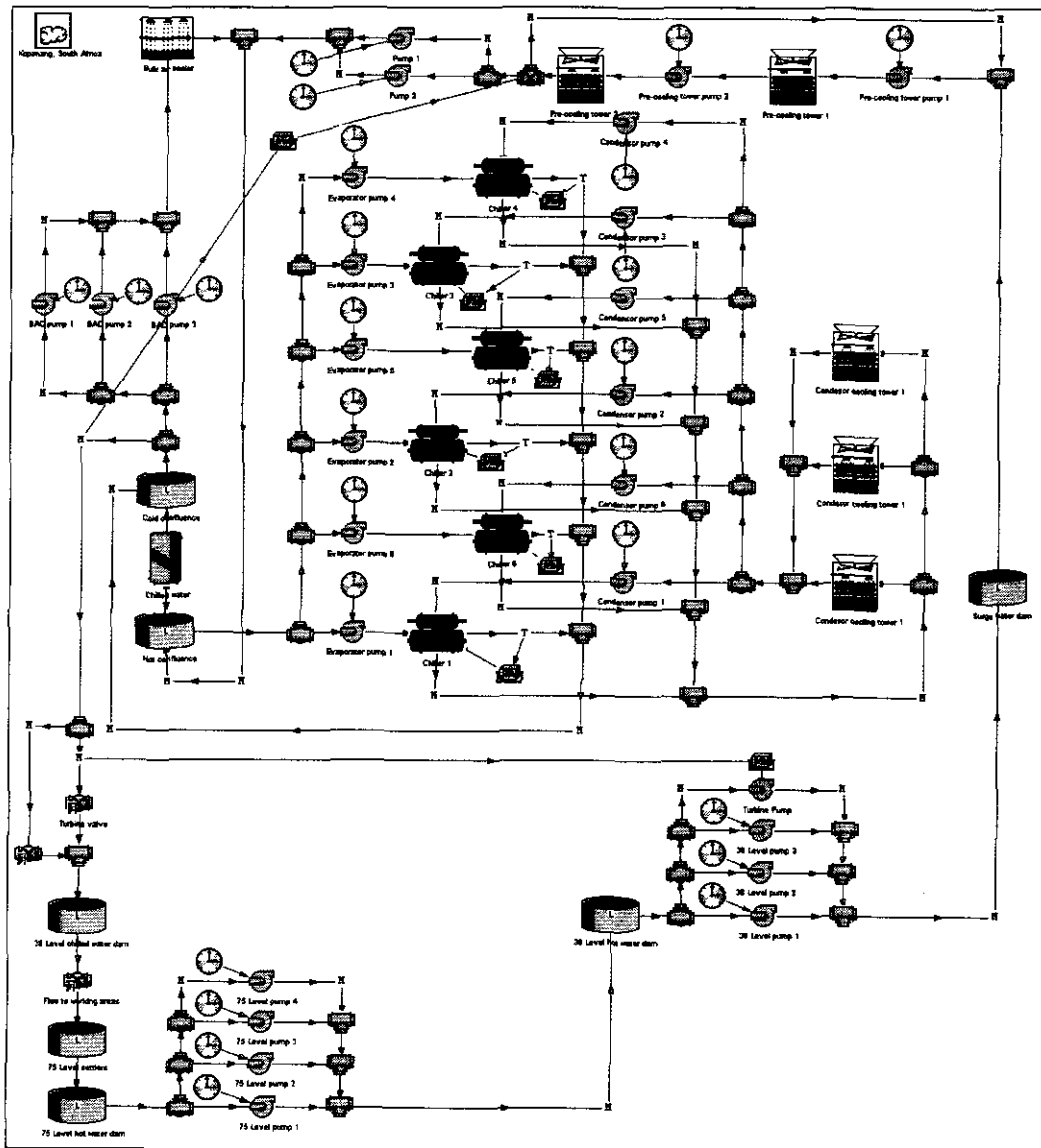


Figure 5.3: Complete integrated mine thermal and energy system of case mine

5.3.3 Simplification of configured simulation model

Because each component requires a certain amount of time to be efficiently simulated, the larger the simulation model the longer the simulation time. Because of the large nature of the Kopanang surface cooling plant and underground pumping network systems a decision was made to simplify the simulation model. Figure 5.4 shows the simplified simulation model that was used to simulate and verify the accuracy of the mine thermal and energy simulation software tool.

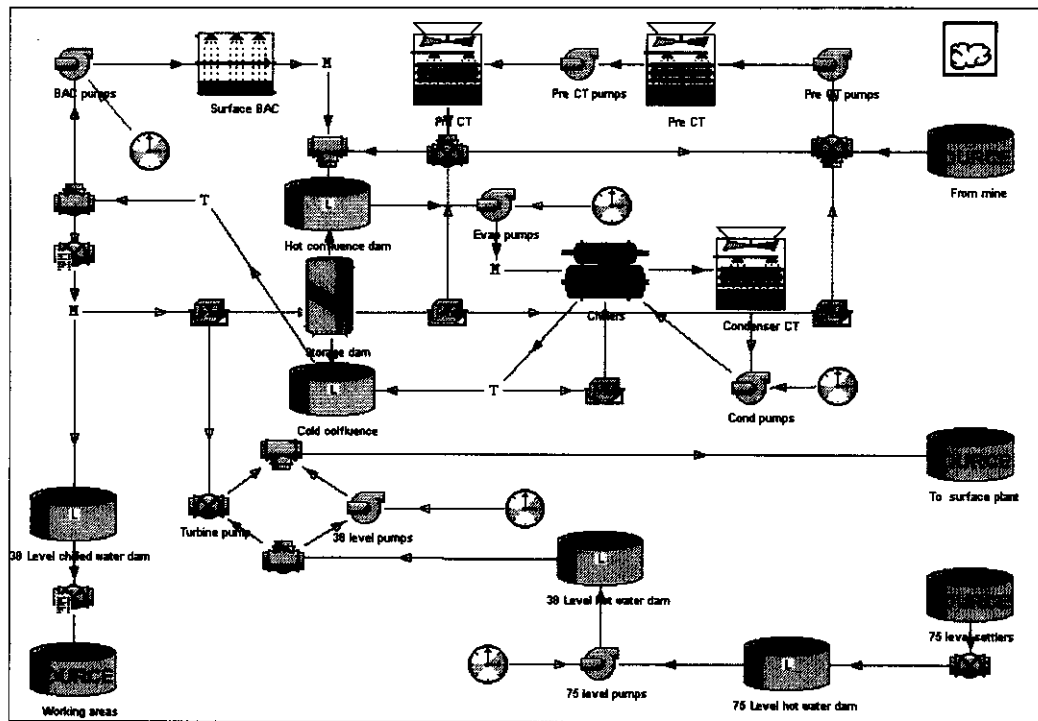


Figure 5.4: Simplified integrated mine thermal and energy simulation model of case mine

The main simplifications that were made involves the combination of the six water-cooled chillers into one controllable model (see Figure 5.4, Chillers) and the combination of the three condenser cooling towers into one representative tower (see Figure 5.4, Condenser CT). Note that on both Figure 5.3 and Figure 5.4 all the control strategies required by the various components of the system could be easily configured and implemented. Results showed that this simplification had no adverse effect on the accuracy of the simulation.

5.3.4 System operation verification

The actual measured system data was used to verify the predictions obtained by the simulation program. A specific typical 24 hour day of system operations was chosen for verification purposes. The simulation was performed according to the exact operating schedules of the measured system. The predictions of this simulation were compared to a specific typical 24 hour day of normal operation, chosen from the most complete measured data. For this study 25 July 2001 was used.

This 24 hour simulation was done to find out if it was possible to simulate the operation of the surface cooling plant and pumping network accurately, i.e. it was necessary to simulate the real life operation of the system. The end result predictions of the thermal and energy characteristics made by the simulation tool were verified against the actual measured data.

Surface cooling plant

For the surface plant the temperatures of the most important components of the system, namely the pre-cooling towers (see Figure 5.4, Pre CT), condenser towers (see Figure 5.4, Condenser CT) and water-cooled chillers (see Figure 5.4, Chillers) were verified. This was necessary to ensure the accuracy of the integrated system and the accurate prediction of the cooling load and energy consumption.

Underground pumping network

For the pumping network the various dam levels of the 38 chilled water, the 38 hot water and the 75 settler dams was verified. This verifies the accurate simulation of flow through a system over a typical 24 hour working operation cycle. The energy consumption of the pumping system was also verified. The following section describes the verification of the various systems.

5.3.5 Verification results

The entire system, as shown in Figure 5.4 was simulated with 60 time intervals per hour. This allowed for the accurate and dynamic simulation of the required controllers of the integrated systems. The following figures represent the main results obtained from the successful verification of the thermal and energy simulation abilities of the designed and implemented simulation tool.

Surface cooling plant

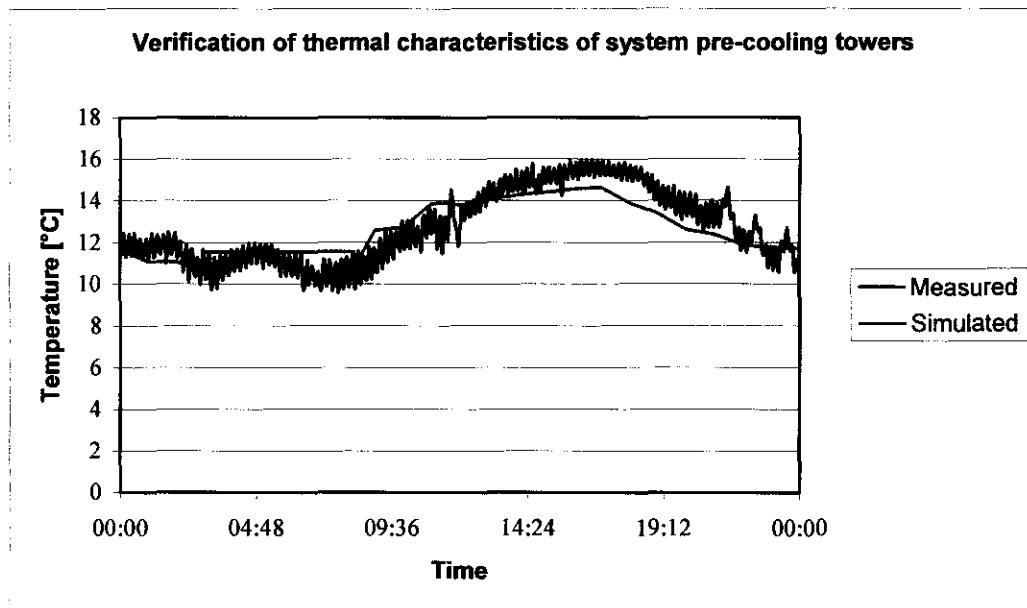


Figure 5.5: Verification of the thermal characteristics of the system pre-cooling towers

Figure 5.5 shows the simulated vs. measured results of the thermal characteristics of the system pre-cooling towers. The profile of the simulated values closely follows the measured results. The simulated system characteristics easily fall within the acceptable 10% allowed difference.

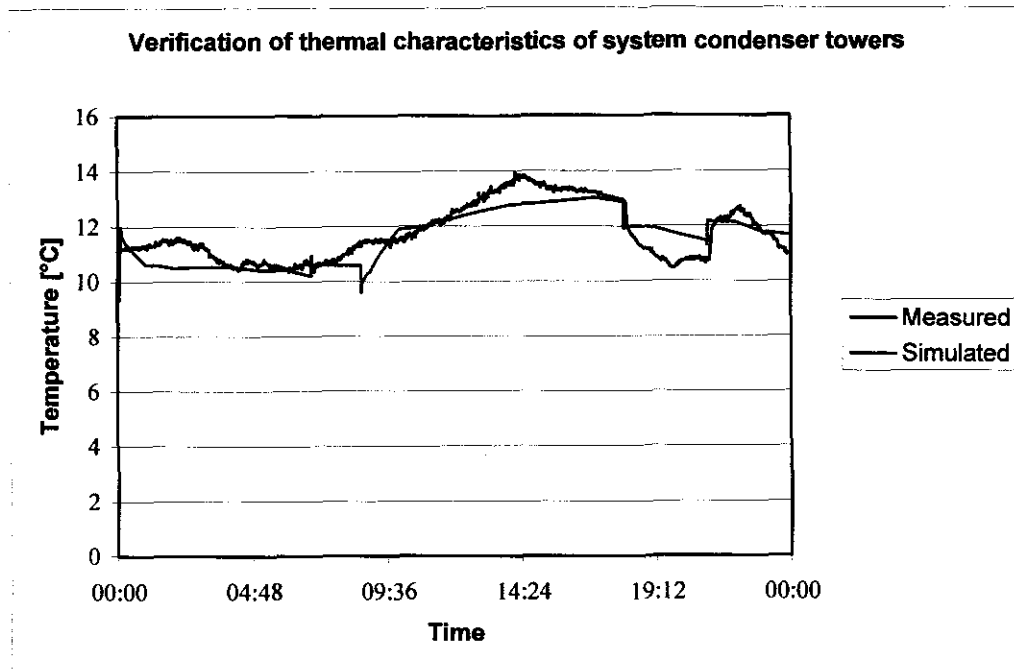


Figure 5.6: Verification of the thermal characteristics of the system condenser towers

Figure 5.6 shows the simulated vs. measured thermal characteristics of the system condenser towers. Again, the results obtained by using VISUALQEC closely follow the trends of the actual measured system.

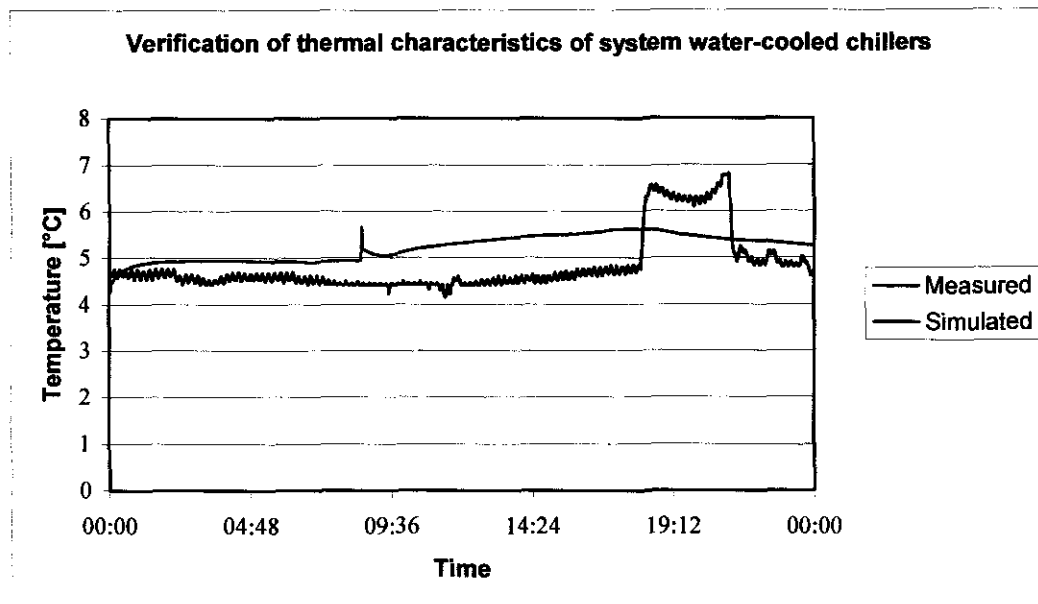


Figure 5.7: Verification of the thermal characteristics of the system water-cooled chillers

Underground pumping network

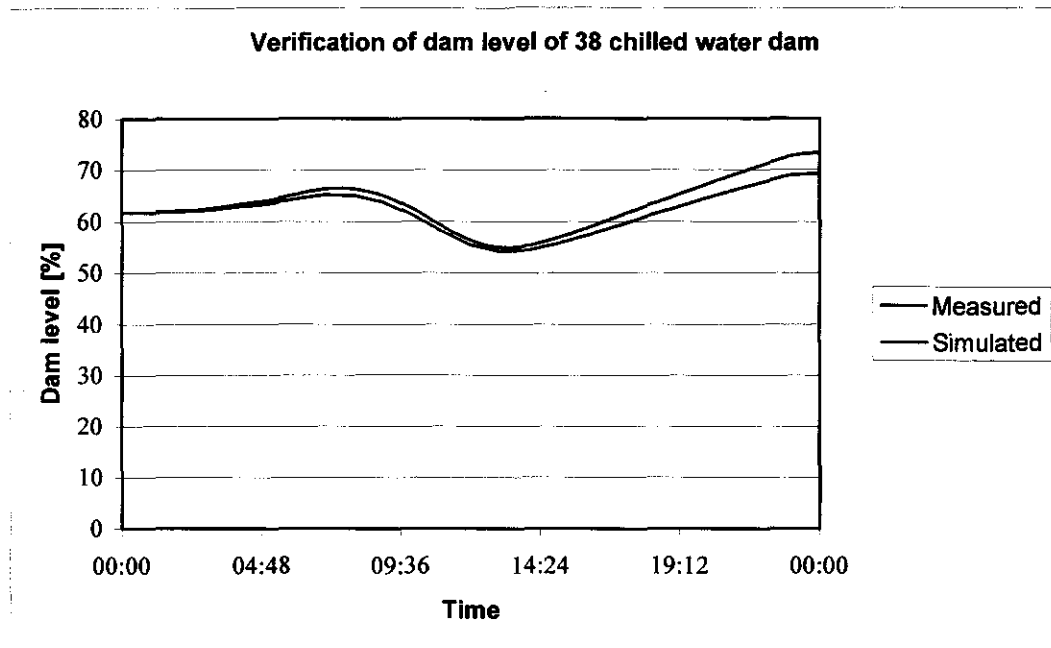


Figure 5.8: Verification of the dam level of the 38 level chilled water dam

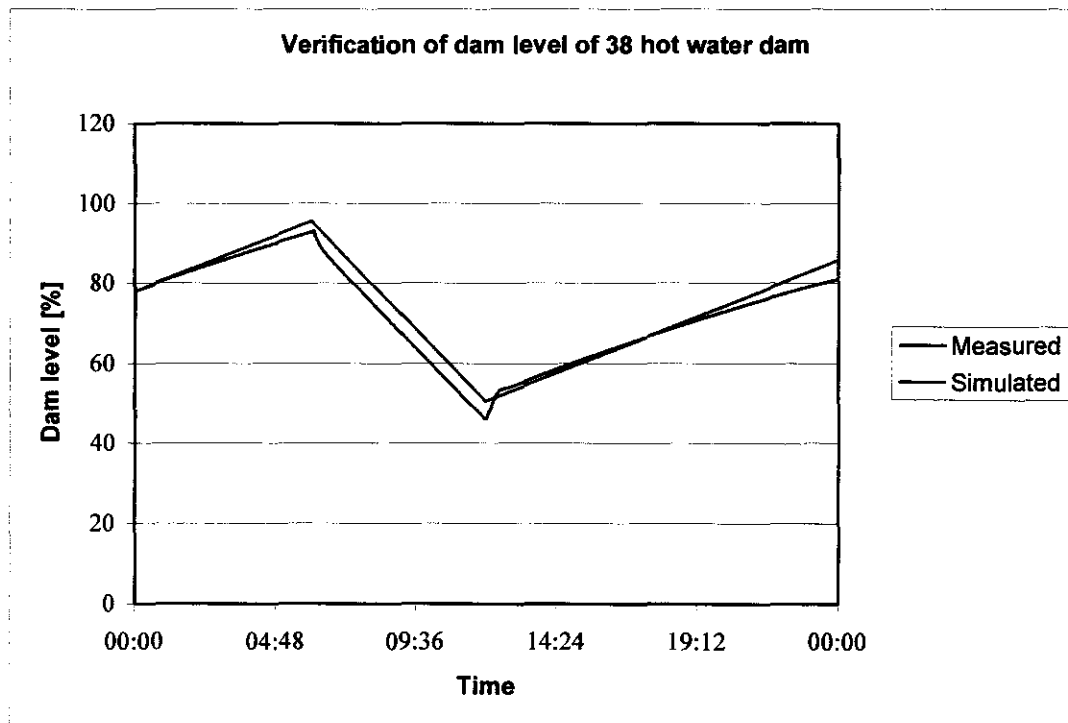


Figure 5.9: Verification of dam the level of the 38 level hot water dam

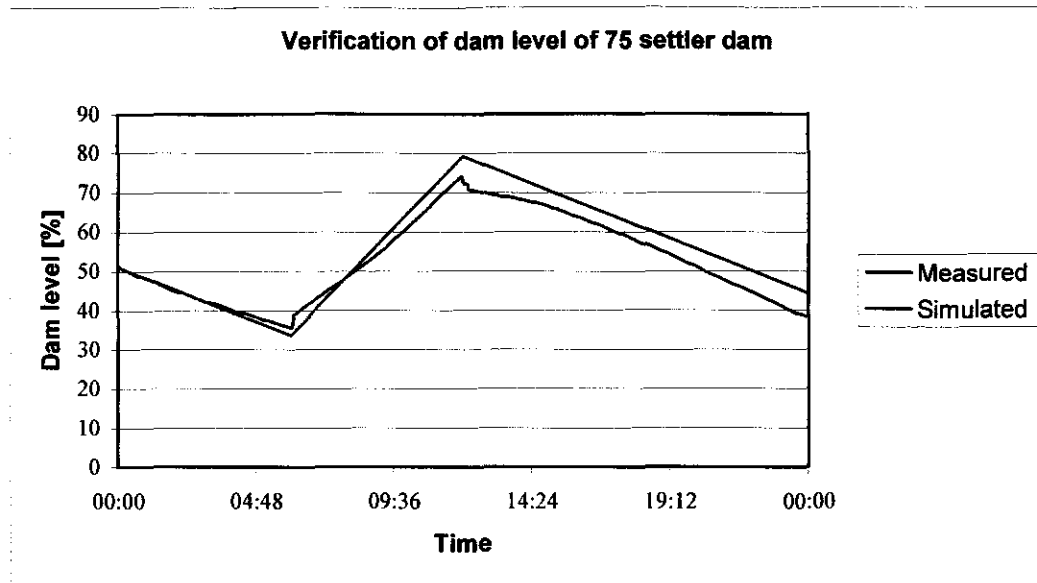


Figure 5.10: Verification of the dam level the 75 level settler dam

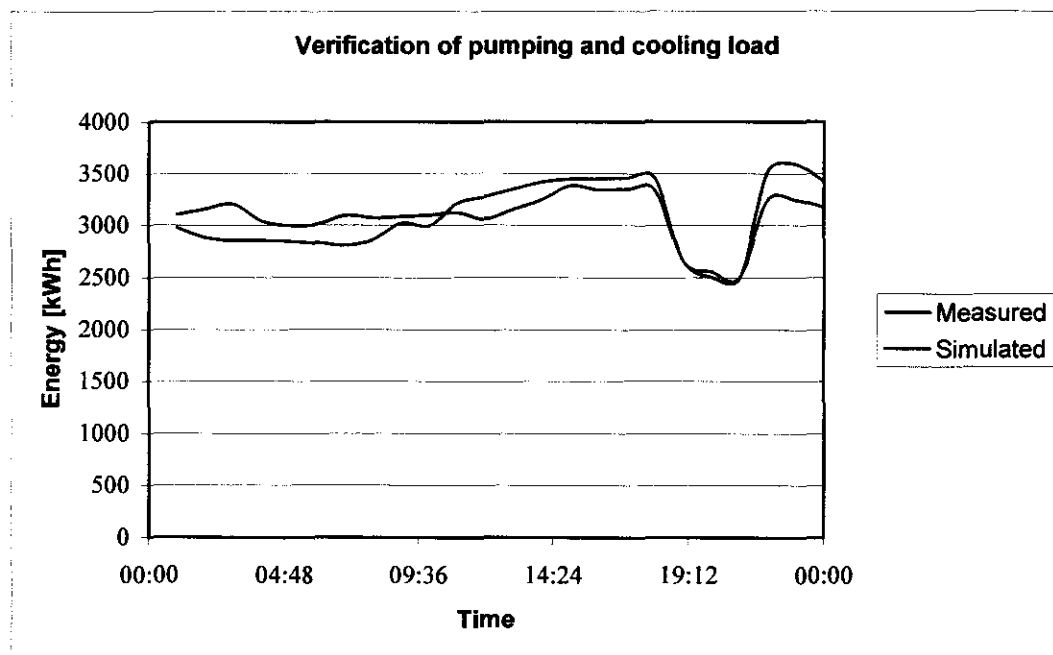


Figure 5.11: Verification of the pumping and cooling load

From Figure 5.8, 5.9 and 5.10 the verification and accuracy of the simulation of the underground pumping network are shown. It can be seen that the new mass flow procedure is able to accurately simulate the real measured system.

5.4 Conclusion

The entire surface cooling plant and underground pumping reticulation system of the Kopanang gold mine was successfully simulated with VISUALQEC. The predictions made by the tool were verified against measured data from the physical system. These measurements included dam levels, active pumps, water flows and various energy consumption values as shown in Figures 5.5 to 5.11.

Satisfactory results were obtained from this verification study. It was found that the simulation tool simulated most of the underground flow network with a maximum error of 5% and a minimum error of 0.05% for 90% of the time. The temperature verification of the surface cooling plant was also satisfactory. From a simulation tool point of view, VISUALQEC also allowed for the easy configuration of the entire Kopanang surface cooling plant and underground pumping network. VISUALQEC also efficiently simulated the entire simulation model and its various components with no marked loss of accuracy. A marked improvement on the overall stability of the simulation process was evident.

CHAPTER 6

CONCLUSION

Using existing building simulation tools as reference, coupled with trends and practice in system simulation software design, a cross industry dynamic integrated thermal and energy system simulation tool was developed. The performance and usability of VISUALQEC within the building industry was verified by completing a typical energy consulting commercial building retrofit study. VISUALQEC was furthermore verified for mining applications by simulating the most typical thermal and energy systems of a gold mine. From these verifications, new applications, possible future implementations and further improvements arise. These aspects and the conclusion of this study form the main objective of this chapter.

6.1 Efficiency and saving through cross industry application of simulation

Thermal and energy system simulation is globally recognised as one of the most effective and powerful tools to improve on overall energy system efficiency. Through simulation, South Africa, seeking ways to improve on current inadequate energy efficiency practices can benefit greatly. However, because of the usual extreme mathematical nature of most simulation tools, coupled with the historically academic environment in which most simulation software is developed, valid negative perceptions towards the commercial viable, large scale implementation of simulation technology exist.

System simulation is too often seen as unstable, cumbersome and time consuming. It is also commonly known that system simulation is only effective in the hands of highly skilled operators. For South Africa to make full use of the benefits that thermal and energy system simulation holds, these negative perceptions towards system simulation need to be addressed.

By providing the South African energy-consuming sector with a commercially easy to use, unconditional, mathematically stable system simulation tool, significant strides towards changing perceptions as well as achieving the required energy efficiency improvement targets can be made. A new cross-industry thermal and energy system simulation tool, VISUALQEC, was developed to satisfy this need.

VISUALQEC was designed with a new user interface specifically designed towards a less academic, more commercially orientated user. A new mass flow simulation procedure specifically created to increase the mathematical stability and speed of the overall thermal and energy system simulation process was also implemented. Through the application of explicit system component models, and a new iterative solution scheme, no need for a differential equation solver was required. A new iterative solution method was implemented. Further improving on overall system and mathematical stability.

VISUALQEC was proven successful for use in both commercial building and mining applications. Before VISUALQEC, no integrated thermal and energy system simulation tool specifically geared towards the dynamic simulation of mine ventilation and cooling (VC) systems existed.

6.2 Towards a more efficient South African building industry

With most of South Africa's commercial buildings being of older design, measures and solutions are sought to improve or change the current wasteful inefficient HVAC systems. Most of the available thermal and energy system simulation tools too date, were designed by academic institutions, making them often commercially inefficient, too time consuming and costly to use. For a commercial product, a greater balance between the usability, cost implications and mathematical accuracy of the simulation tool needs to be obtained.

VISUALQEC was created to fulfill this requirement. Through the explicit system component models and new iterative solution model, VISUALQEC ensures fast, accurate and stable system simulations. Unlike most other building simulation tools, VISUALQEC was specifically designed to always give the user a solution. By using the new user interface, it is also possible to easily configure any required system and or HVAC system retrofit. With such a powerful tool available to commercial energy consultants, the application of applied large-scale building energy efficiency practices becomes a reality. The South African commercial building industry can take full advantage of the power that system simulation holds towards improving building and HVAC system design and retrofit.

6.3 Other applications for simulation in the building industry

Because of the stable nature of the VISUALQEC simulation engine, new dynamic control strategies can be implemented. By integrating VISUALQEC into the building management system (BMS), the best possible optimised current system operation can be simulated and controlled.

6.4 Towards a more efficient South African mining industry

The future of the South African mining industry lies in improving the economic effectiveness of the overall mining operation [17]. The main functions of a working mine are augmented by a multiplicity of essential auxiliary activities. These include the use of ventilation and cooling (VC) systems, pumping systems and various maintenance services. These activities, systems and services are all heavy, energy and electricity intensive consumers. It is through the

efficient design and control of these thermal and energy systems that the full potential for financial and environmental benefits of applied energy efficiency practices can be realised.

An extensive literature survey showed that, to date, no suitable dynamic integrated mining thermal and energy system simulation tool existed. VISUALQEC through its cross-industry design, fulfill these requirements.

Because of the explicit solution nature of the system components and simulation solution model implemented in VISUALQEC, the fast stable solution of the dynamic integrated thermal and energy systems of mines is ensured. Because of the stable mathematical and logical methods implemented in VISUALQEC it is possible to extend the simulation process even further.

6.5 Other applications for simulation in the mining industry

As discussed, one method for increased efficiency of the thermal and energy systems of mines, resulting in higher income profit associated with lower electricity consumption, is better control of the VC system or the accommodation of ESKOM's (dynamic structures) new cost based tariff structures. One of these cost based tariff structures is real time pricing (RTP).

RTP is a pricing methodology, which sets the selling price of electricity at the marginal and transmission cost plus an added profit margin. The marginal cost of electricity is defined as the hourly market price by which electricity is generated and transferred from the transmission system to the electricity distribution system. RTP thus offers the consumer a clear economical cost signal, enabling and motivating electricity consumers to adjust their current consumption patterns to match ESKOM's objectives.

With real time pricing of electricity and a stable, fast, dynamic integrated thermal and energy system simulation environment (VISUALQEC) available, an optimisation of the daily mine systems VC control for its various systems and sub systems i.e. underground pumping can be done. This optimisation process involves the optimised solution of the objective function. In the case of RTP, this means minimising the overall daily electricity function. With the necessary system constraints implemented it will be possible to optimise the VC control strategies and electricity consumption to allow for maximum savings without compromising

or changing the existing mining operations. With better control over the daily operation of the related mining systems implemented, the financial viability and increased energy efficiency of mines can be considerably increased. The RTP process optimisation as suggested above is currently being investigated and implemented with great success by HVAC International (Pty) Ltd.

6.6 Future work and possible improvements

Although the success of the implementation of the chosen system simulation user interface and system simulation engine for VISUALQEC was verified, further improvements on the current implementation of the cross-industry thermal and energy system simulation tool can be made. The following suggestions for future work will further improve the efficiency, accuracy and usability of VISUALQEC.

1. More detailed case studies designed to test various aspects of the simulation tool are needed to more extensively verify the simulation tool and models. Although initial results on Kopanang and the Telkom Data Building were highly satisfactory, the accuracy and stability of some of the component models and new procedures still need to be verified.
2. It was found that the new process of simulating without solving mass flow drastically improved the stability of the system simulation engine. However, in some complex cases, the ease of specifying and understanding the logical flow of mass in the user interface, from a user point of view was found to be wanting. Further improvements are needed on the way the mass flow through the system is specified by the user.
3. The new explicit iterative simulation of system components has proven to drastically add to the speed and stability of the overall simulation solution. Currently, by running through a fixed number of iterations, the explicit system components are allowed to reach steady state. No order in which the system components are solved is required. However, this has the effect that the simulation engine does waste valuable iteration cycles. By calculating the order of system component solution and the required maximum of needed iterations per system component, further speed improvements can be made.

6.7 Conclusion

Energy efficiency, and in particular improvements in the consumption of electricity, will become increasingly important in the SA energy economy. Thermal and energy system simulation is globally recognised as one of the most effective and powerful tools to improve energy efficiency.

The South African mining and commercial building industries are two of the major consumers of electricity within South Africa. By improving energy efficiency practices within the building and mining industry, large potential savings can be realised. VISUALQEC was designed and implemented to comply with the needs and requirements identified.

A new explicit system component model and explicit system simulation engine, combined with a new improved simulation of mass flow through a system procedure, suggested a marked improvement on overall simulation stability, efficiency and speed. The commercial usability of the new simulation tool was verified for building applications by doing an extensive building energy savings audit. The new simulation tool was further verified by simulating the ventilation and cooling (VC) and underground pumping system of a typical South African gold mine. Initial results proved satisfactory but more case studies to further verify the accuracy of the implemented cross-industry thermal and energy system simulation tool are needed. Because of the stable nature of the new VISUALQEC simulation engine, the simulation process can be further extended to the mathematical optimisation of various system variables.

In conclusion, this study highlighted the need and creation for new simulation procedures and system designs for the successful implementation of a single dynamic thermal and energy system simulation tool for cross-industry applications. This tool also addresses the need for a dynamic thermal and energy system simulation tool for mining VC systems design and retrofit. South Africa should take full advantage of the power of thermal and energy system simulation towards creating a more energy efficient society.

APPENDIX A
SYSTEM COMPONENT MODELS

Pumps

Assumptions:

- No psychrometric property relations are needed to calculate the liquid density. The density is taken throughout as $\rho_l = 1000 \text{ kg/m}^3$. Similarly the specific heat at constant pressure will be taken throughout as $c_{pl} = 4187 \text{ J/kg}^\circ\text{C}$.
- The drive motor may not be situated within the liquid flow.
- The dynamic pressure difference over the pump is assumed to be negligible.

Parameters:

- a_j Correlation coefficients for the K_h versus K_f relation with $j = 0$ to 2 .
 b_j Correlation coefficients for the η_{pump} versus K_f relation with $j = 0$ to 2 .

Inputs:

Simulation:

- m_l Mass flow rate of liquid at inlet [kg/s].
 T_{li} Temperature of liquid entering at inlet [$^\circ\text{C}$].

Interface:

- D Rotor diameter [m].
 N Rotational speed of the pump [rpm].
 η_{motor} Efficiency of the drive motor.
 H Pressure head [m].
 q Flow rate [kg/s].
 η_{pump} Pump efficiency.

Outputs:

- dP_1 Static pressure rise [Pa].
 T_{le} Temperature of liquid leaving at outlet [$^\circ\text{C}$].
 P_{wr} Pumping power required [W].

Internal variables:

- H_1 Extra pressure head point [m].
 H_2 Extra pressure head point [m].
 K_f Dimensionless flow coefficient.
 K_h Dimensionless pressure head coefficient.
 η_{pump} Efficiency of the pump.
 η_1 Extra efficiency point.

-
- η_2 Extra efficiency point.
 Q_t Rate of heat gain to the liquid [W].

Explicit equations:

Three different pressures and efficiencies at three different flows are needed to calculate the correlation coefficients. A flow variation of 20% above and below the interface input value is assumed. The extra points are then calculated as follow:

$$H_1 = a_h(0.8q)^2 + b_h(0.8q) + c_h$$

$$\eta_1 = a_\eta(0.8q)^2 + b_\eta(0.8q) + c_\eta$$

$$H_2 = a_h(1.2q)^2 + b_h(1.2q) + c_h$$

$$\eta_2 = a_\eta(1.2q)^2 + b_\eta(1.2q) + c_\eta$$

with

$$a_h = -0.63125$$

$$b_h = 1.18125$$

$$a_\eta = -0.06188$$

$$b_\eta = 0.33625 \quad \text{for } 0 < q < 3.$$

$$a_h = -0.25914$$

$$b_h = 0.591667$$

$$a_\eta = -0.01816$$

$$b_\eta = 0.190308 \quad \text{for } 3 < q < 7.5.$$

$$a_h = -0.3248$$

$$b_h = 3.736$$

$$a_\eta = -0.00648$$

$$b_\eta = 0.133 \quad \text{for } 7.5 < q < 15.$$

$$a_h = -0.100$$

$$b_h = 2.197$$

$$a_\eta = -0.00507$$

$$b_\eta = 0.159 \quad \text{for } 15 < q < 20.$$

$$a_h = -0.02674$$

$$b_h = 0.639$$

$$a_{\eta} = -0.00089$$

$$b_{\eta} = 0.05089 \quad \text{for } 20 < q < 40.$$

$$a_h = -0.01505$$

$$b_h = 0.88875$$

$$a_{\eta} = -0.00028$$

$$b_{\eta} = 0.029263 \quad \text{for } 40 < q < 80.$$

$$a_h = -0.00164$$

$$b_h = 0.262$$

$$a_{\eta} = -0.000047$$

$$b_{\eta} = 0.010467 \quad \text{for } 80 < q < 150.$$

These values were calculated as the average values of the coefficients of a wide range of centrifugal pump curves. These results are shown in figures A.1 to A.14. c_h and c_{η} are now calculated from the one given operating point obtained from the supplier or measurements. This implies only one operating point is needed to obtain the mathematical model of a specific pump. And

$$c_h = H - (a_h q^2 + b_h q)$$

$$c_{\eta} = \eta - (a_{\eta} q^2 + b_{\eta} q)$$

For each of these three points the K_h and K_f value must be calculated as follow:

$$K_{h1} = \frac{gH_1}{N^2 D^2}$$

$$K_{f1} = \frac{m_1}{\rho_1 N D^3}$$

$$K_h = \frac{gH}{N^2 D^2}$$

$$K_f = \frac{m}{\rho_1 N D^3}$$

$$K_{h2} = \frac{gH_2}{N^2 D^2}$$

$$K_{f2} = \frac{m_2}{\rho_1 N D^3}$$

Using these values the correlation coefficients can be calculated as follow:

$$a_2 = \frac{(K_{h2} - K_{h1})(K_f - K_{f1}) - (K_{f2} - K_{f1})(K_h - K_{h1})}{(K_{f2}^2 - K_{f1}^2)(K_f - K_{f1}) - (K_{f2} - K_{f1})(K_f^2 - K_{f1}^2)}$$

$$a_1 = \frac{(K_h - K_{h1}) - (K_f^2 - K_{f1}^2)a_2}{(K_f - K_{f1})}$$

$$a_0 = K_{h1} - K_{f1}a_1 - K_{f1}^2a_2$$

$$b_2 = \frac{(\eta_2 - \eta_1)(K_f - K_{f1}) - (K_{f2} - K_{f1})(\eta - \eta_1)}{(K_{f2}^2 - K_{f1}^2)(K_f - K_{f1}) - (K_{f2} - K_{f1})(K_f^2 - K_{f1}^2)}$$

$$b_1 = \frac{(\eta - \eta_1) - (K_f^2 - K_{f1}^2)b_2}{(K_f - K_{f1})}$$

$$b_0 = \eta_1 - K_{f1}b_1 - K_{f1}^2b_2$$

These coefficients can now be used to calculate the necessary outputs with the following explicit equations:

$$K_f = \frac{m_f}{\rho_f ND^3}$$

$$K_h = a_0 + a_1 K_f + a_2 K_f^2$$

$$dP_f = K_h \rho_f N^2 D^2$$

$$\eta_{\text{pump}} = b_0 + b_1 K_f + b_2 K_f^2$$

$$Q_l = \frac{(1 - \eta_{\text{pump}}) m_f dP_f}{\rho_f \eta_{\text{pump}}}$$

$$T_{le} = T_{li} + \frac{Q_l}{m_f c_{pl}}$$

$$Pwr = \frac{dP_f m_f}{\rho_f \eta_{\text{motor}} \eta_{\text{pump}}}$$

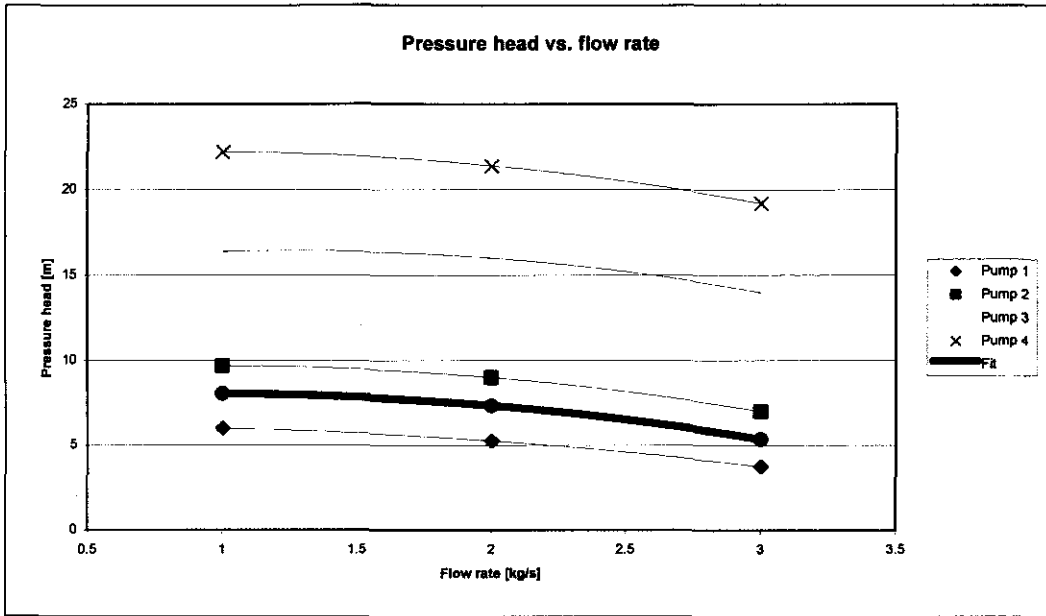


Figure A.1: Pressure head correlation for flow between 0 and 3 kg/s

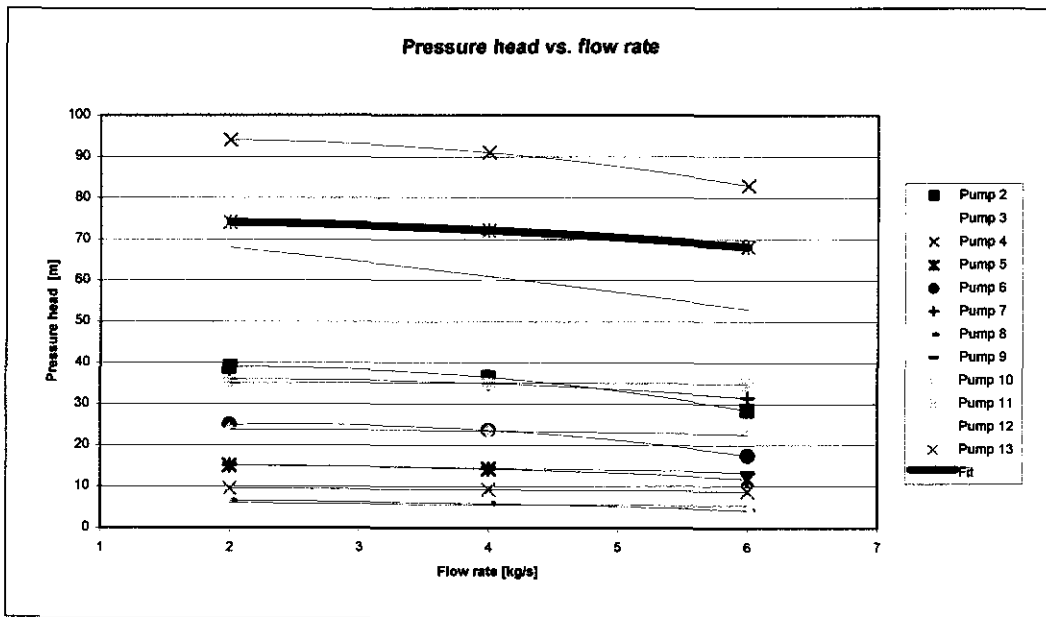


Figure A.2: Pressure head correlation for flow between 3 and 7.5 kg/s

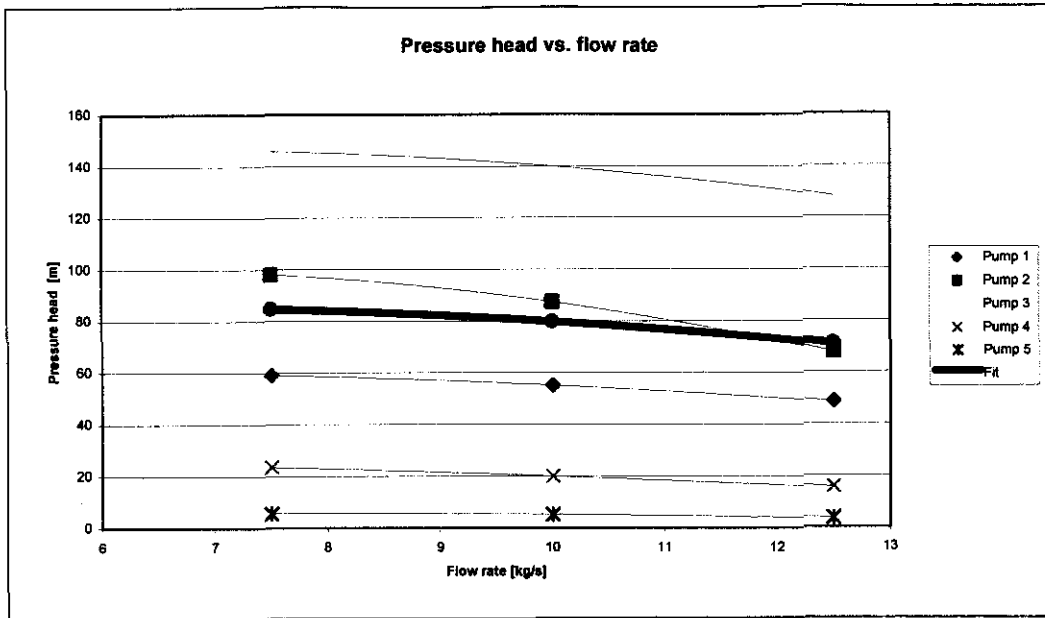


Figure A.3: Pressure head correlation for flow between 7.5 and 15 kg/s

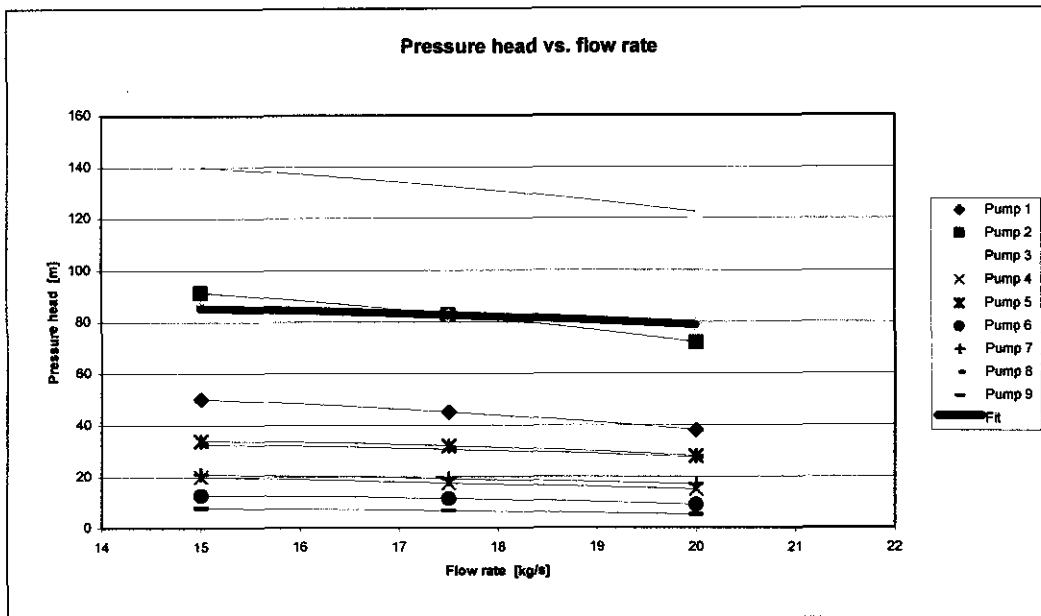


Figure A.4: Pressure head correlation for flow between 15 and 20 kg/s

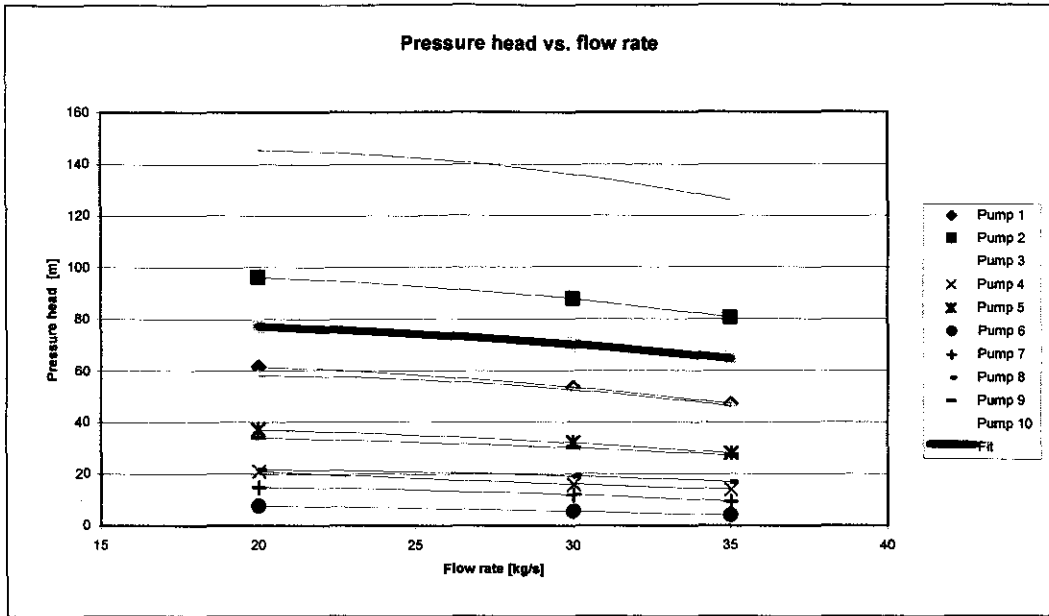


Figure A.5: Pressure head correlation for flow between 20 and 40 kg/s

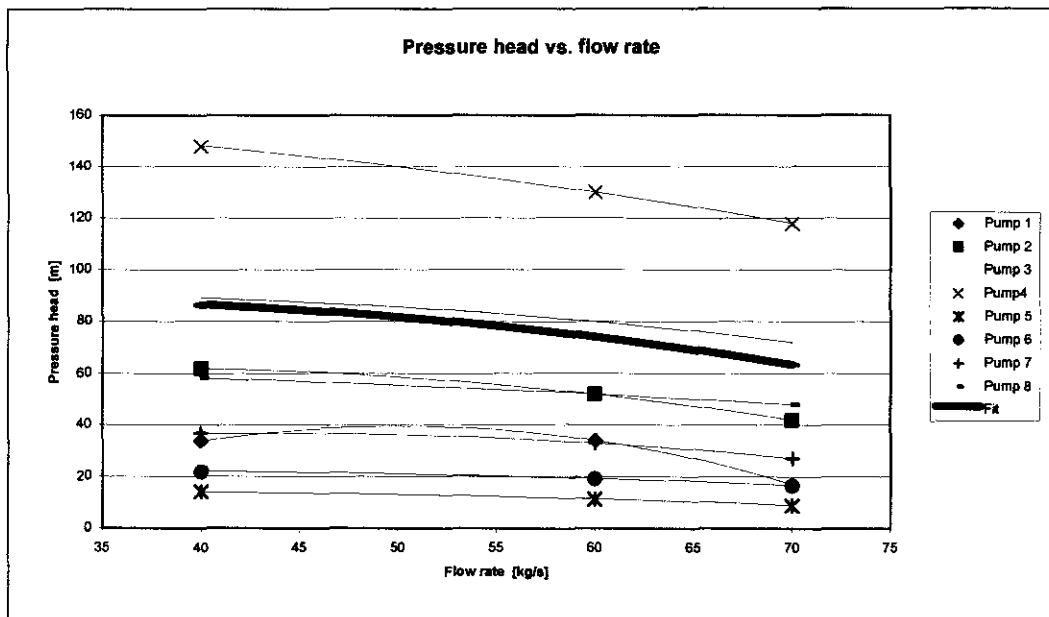


Figure A.6: Pressure head correlation for flow between 40 and 70 kg/s

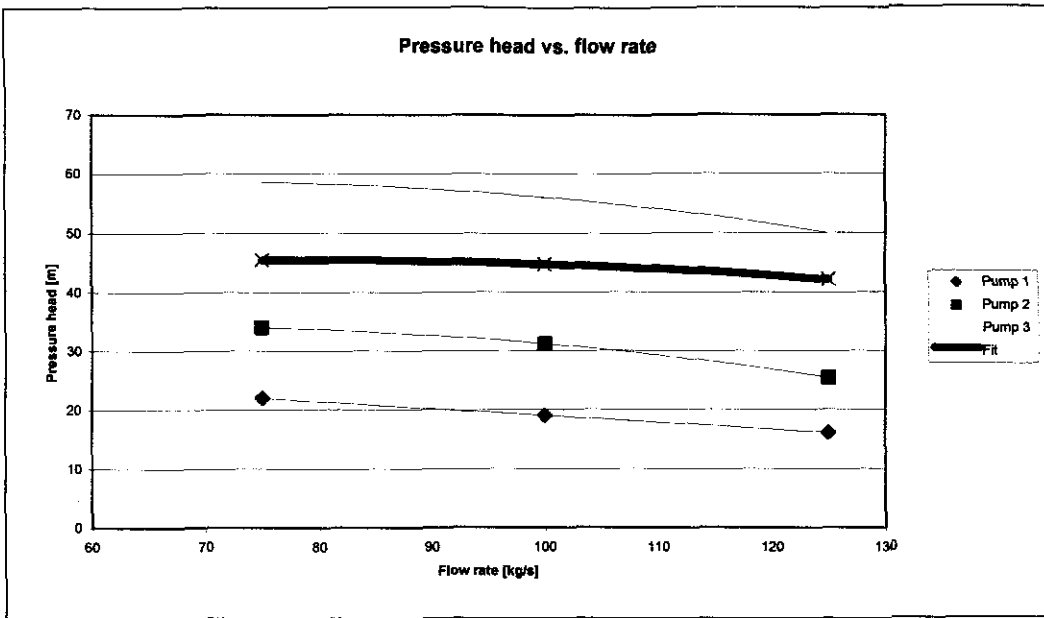


Figure A.7: Pressure head correlation for flow between 70 and 150 kg/s

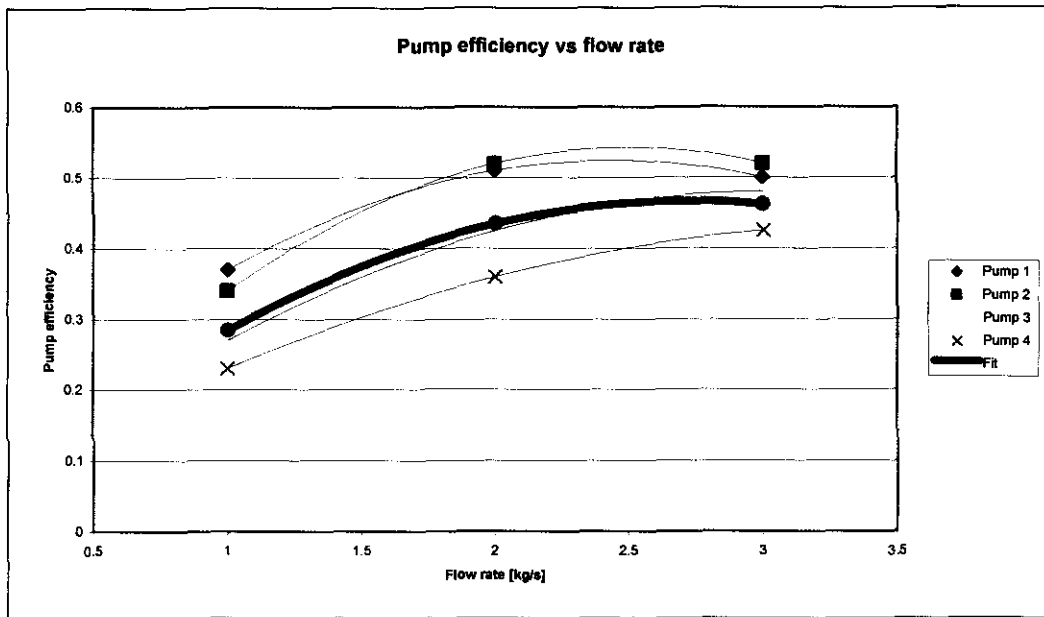


Figure A.8: Efficiency correlation for flow between 0 and 3 kg/s

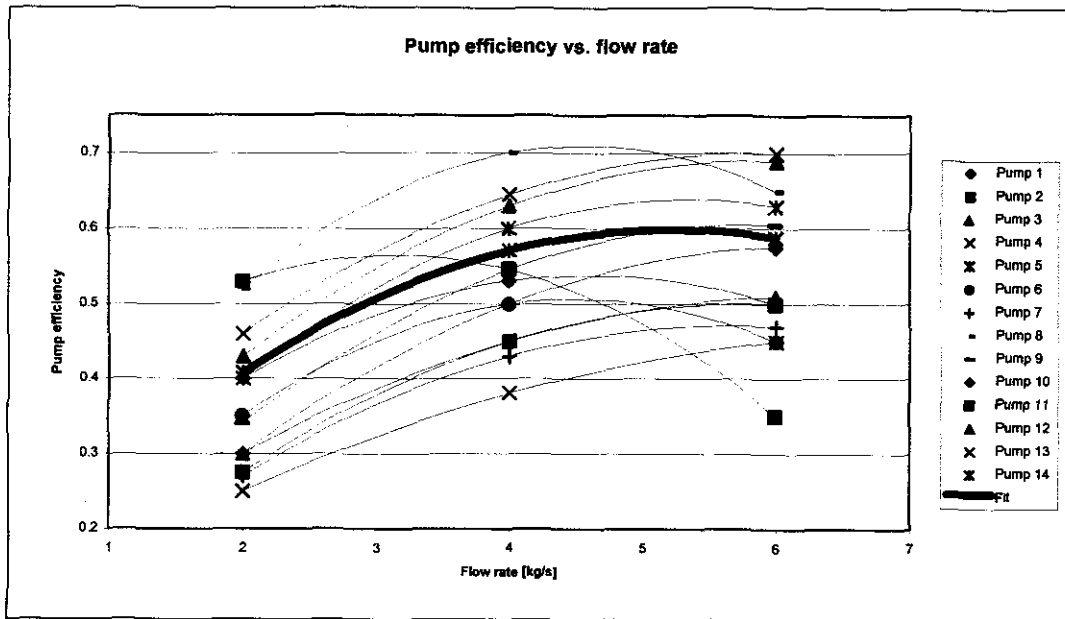


Figure A.9: Efficiency correlation for flow between 3 and 7.5 kg/s

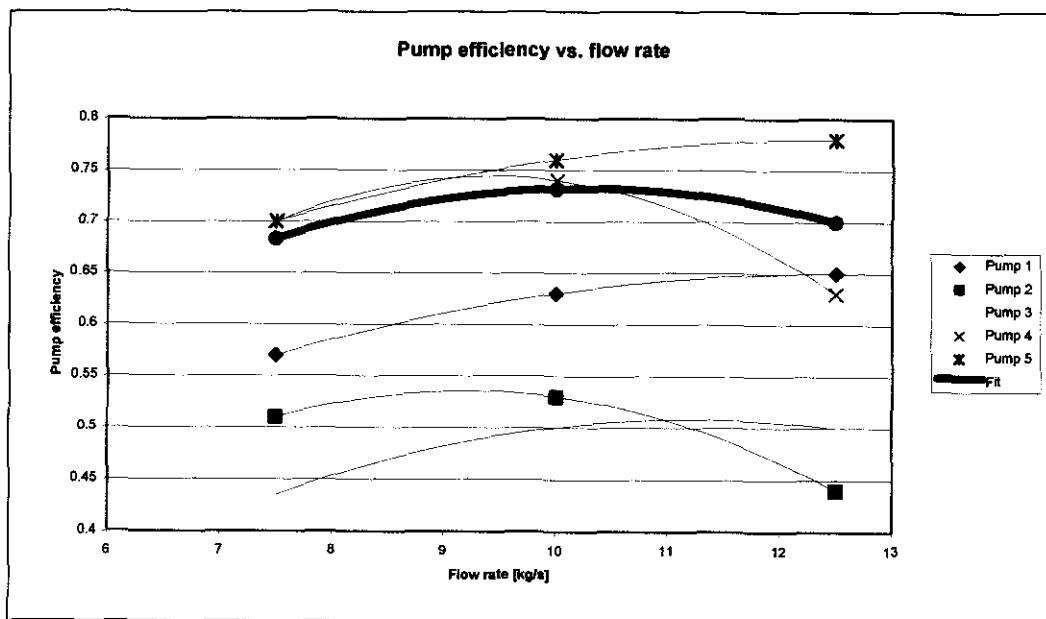


Figure A.10: Efficiency correlation for flow between 7.5 and 15 kg/s

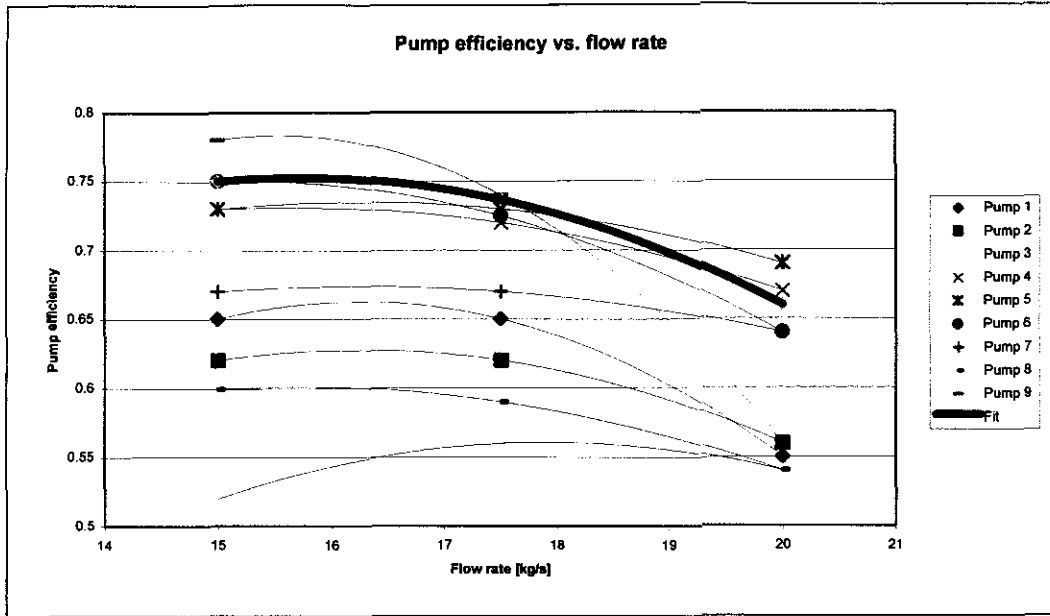


Figure A.11: Efficiency correlation for flow between 15 and 20 kg/s

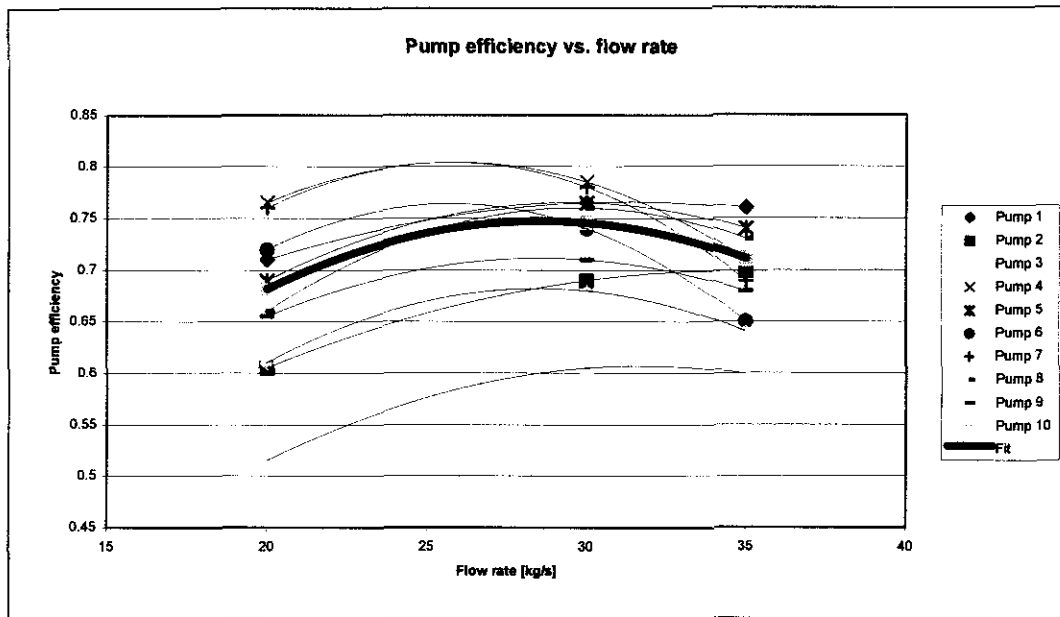


Figure A.12: Efficiency correlation for flow between 20 and 40 kg/s

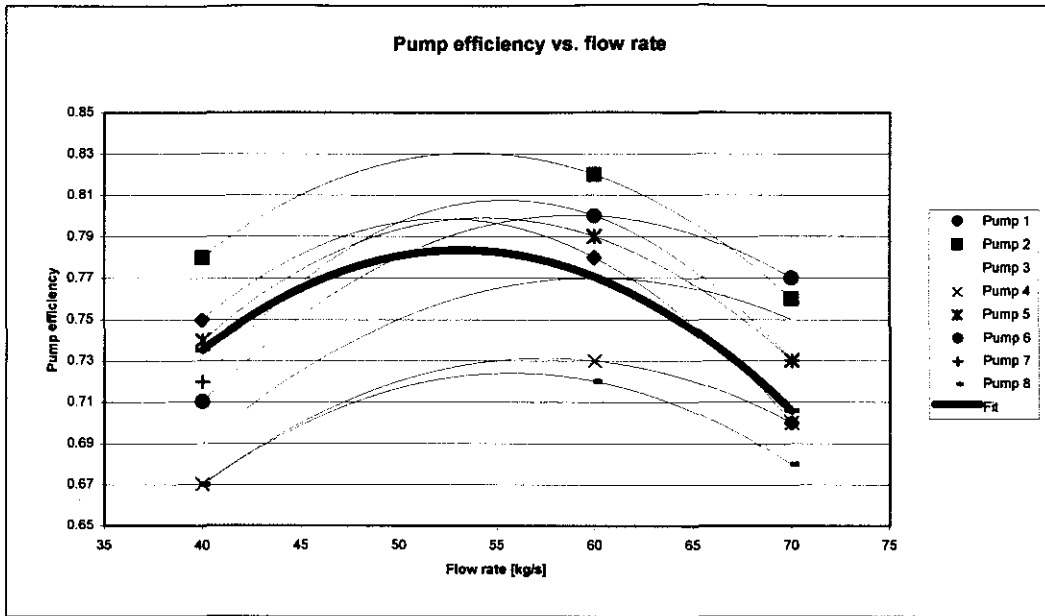


Figure A.13: Efficiency correlation for flow between 40 and 70 kg/s

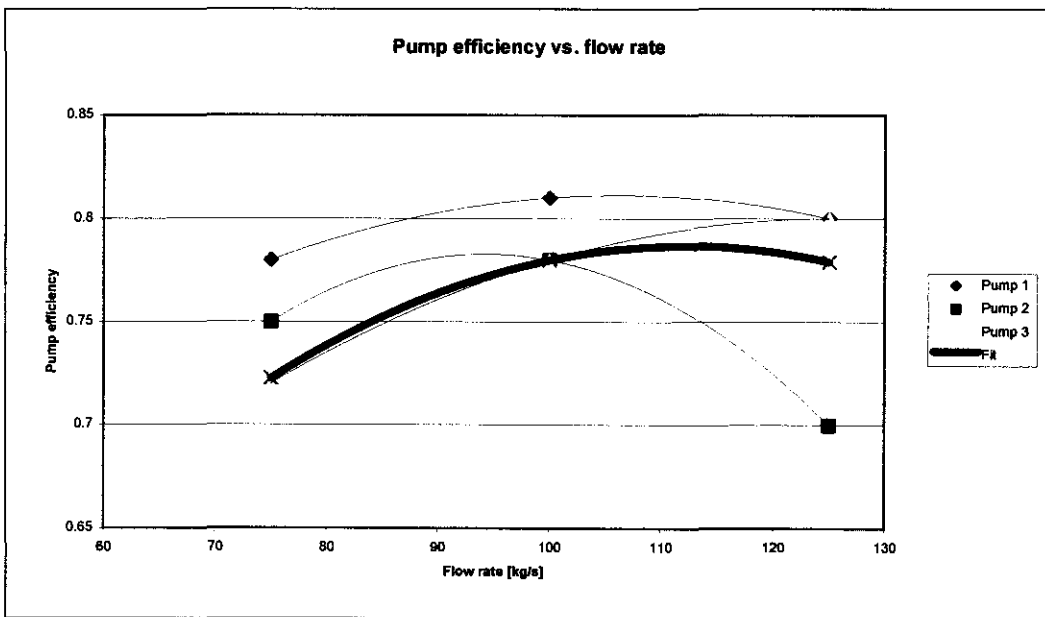


Figure A.14: Efficiency correlation for flow between 70 and 150 kg/s

Fan

Parameters:

- a_j Correlation coefficients for the K_h versus K_f relation with $j = 0$ to 2 .
 b_j Correlation coefficients for the η_{fan} versus K_f relation with $j = 0$ to 2 .

Inputs:

Simulation:

- m_a Mass flow rate of dry air at inlet [kg/s].
 w_{ai} Humidity ratio of air entering at the inlet [kg_{vapour} / kg_{dry air}].
 h_{ai} Specific enthalpy of air entering at the inlet [J/kg_{dry air}].

Interface:

- D The rotor diameter [m].
 N Rotational speed of the fan [rpm].
 η_{motor} Efficiency of the drive motor.
 P_{total} Total pressure [Pa].
 q Volume flow rate [m³/s].
 η_{fan} Fan efficiency.

Outputs:

- dP_a Static pressure rise [Pa].
 w_{ae} Humidity ratio of air entering at the inlet [kg_{vapour} / kg_{dry air}].
 h_{ae} Specific enthalpy of air entering at the inlet [J/kg_{dry air}].
 P_{wr} Power required [W].

Internal variables:

- P_1 Extra pressure points [Pa].
 P_2 Extra pressure points [Pa].
 K_f Dimensionless flow coefficient.
 K_h Dimensionless pressure head coefficient.
 η_{fan} Efficiency of the fan.
 η_1 Extra efficiency point.
 η_2 Extra efficiency point.
 ρ_a Air density [kg/m³]
 Q_a Rate of heat gain to the air [W].

Explicit equations:

Three different pressures and efficiencies at three different flows are needed to calculate the correlation coefficients. A flow variation of 10% above and below the interface input value is assumed. The extra points are then calculated as follow:

$$P_1 = a_h(0.9q)^2 + b_h(0.9q) + c_h$$

$$\eta_1 = a_\eta(0.9q)^2 + b_\eta(0.9q) + c_\eta$$

$$P_2 = a_h(1.1q)^2 + b_h(1.1q) + c_h$$

$$\eta_2 = a_\eta(1.1q)^2 + b_\eta(1.1q) + c_\eta$$

with

$$a_h = -823.114$$

$$b_h = 354.706$$

$$a_\eta = -95.3q^2 + 108.78q - 33.777$$

$$b_\eta = 18.281q^2 - 24q + 10.802 \quad \text{for } 0 < q < 1 \text{ [m}^3/\text{s].}$$

$$a_h = -147.577$$

$$b_h = 378.984$$

$$a_\eta = -0.2432q^2 + 1.4215q - 2.176$$

$$b_\eta = 0.3379q^2 - 2.2688q + 4.4056 \quad \text{for } 1 < q < 5 \text{ [m}^3/\text{s].}$$

$$a_h = -33.262$$

$$b_h = 309.9086$$

$$a_\eta = 0.0074q - 0.0801$$

$$b_\eta = -0.054q + 0.7516 \quad \text{for } 5 < q < 15 \text{ [m}^3/\text{s].}$$

$$a_h = -5.47298$$

$$b_h = 158.6346$$

$$a_\eta = -0.00299$$

$$b_\eta = 0.1219 \quad \text{for } 15 < q < 30 \text{ [m}^3/\text{s].}$$

$$a_h = -2.14676$$

$$b_h = 74.34644$$

$$a_\eta = -0.0014$$

$$b_\eta = 0.081011$$

$$\text{for } 30 < q < 60 \text{ [m}^3/\text{s].}$$

These values were calculated as the average values of the coefficients of a wide range of backward curved centrifugal fan curves or as a function of flow. The a and b coefficients as a function of flow can be seen in figures A.15 to A.20. The accuracy of this can be seen in figures A.21 to A.30. c_h and c_η are calculated from the one given operating point obtained from the supplier or measurements. This implies that only one operating point is needed to obtain the mathematical model of a specific fan. And

$$c_h = P - (a_h q^2 + b_h q)$$

$$c_\eta = \eta - (a_\eta q^2 + b_\eta q)$$

For each of these three points the K_h and K_f value must be calculated as follow:

$$K_{h1} = \frac{P_1}{\rho_a N^2 D^2}$$

$$K_{f1} = \frac{\rho_a (0.9q)}{\rho_a N D^3}$$

$$K_h = \frac{P}{\rho_a N^2 D^2}$$

$$K_f = \frac{\rho_a q}{\rho_a N D^3}$$

$$K_{h2} = \frac{P_2}{\rho_a N^2 D^2}$$

$$K_{f2} = \frac{\rho_a (1.1q)}{\rho_a N D^3}$$

Using these values the correlation coefficients can be calculated as follow:

$$a_2 = \frac{(K_{h2} - K_{h1})(K_f - K_{f1}) - (K_{f2} - K_{f1})(K_h - K_{h1})}{(K_{f2}^2 - K_{f1}^2)(K_f - K_{f1}) - (K_{f2} - K_{f1})(K_f^2 - K_{f1}^2)}$$

$$a_1 = \frac{(K_h - K_{h1}) - (K_f^2 - K_{f1}^2)a_2}{(K_f - K_{f1})}$$

$$a_0 = K_{h1} - K_{f1}a_1 - K_{f1}^2 a_2$$

$$b_2 = \frac{(\eta_2 - \eta_1)(K_f - K_{f1}) - (K_{f2} - K_{f1})(\eta - \eta_1)}{(K_{f2}^2 - K_{f1}^2)(K_f - K_{f1}) - (K_{f2} - K_{f1})(K_f^2 - K_{f1}^2)}$$

$$b_1 = \frac{(\eta - \eta_1) - (K_f^2 - K_{f1}^2)b_2}{(K_f - K_{f1})}$$

$$b_0 = \eta_1 - K_{f1}b_1 - K_{f1}^2b_2$$

These coefficients can now be used to calculate the necessary outputs with the following explicit equations:

$$\rho_a = \zeta(h_{ai}, w_{ai})$$

$$K_f = \frac{\rho_a q}{\rho_a N D^3}$$

$$K_h = a_0 + a_1 K_f + a_2 K_f^2$$

$$P_{total} = K_h \rho_a N^2 D^2$$

$$\eta_{fan} = b_0 + b_1 K_f + b_2 K_f^2$$

$$Q_a = \frac{(1 - \eta_{fan}) m_a P_{total}}{\rho_a \eta_{fan}} \quad \text{motor outside airstream}$$

$$Q_a = \frac{(1 - \eta_{fan} \eta_{motor}) m_a P_{total}}{\rho_a \eta_{fan} \eta_{motor}} \quad \text{motor inside airstream}$$

$$w_{ae} = w_{ai}$$

$$h_{ae} = h_{ai} + \frac{Q}{m_a}$$

$$P_{WR} = \frac{P_{total} m_a}{\rho_a \eta_{motor} \eta_{fan}}$$

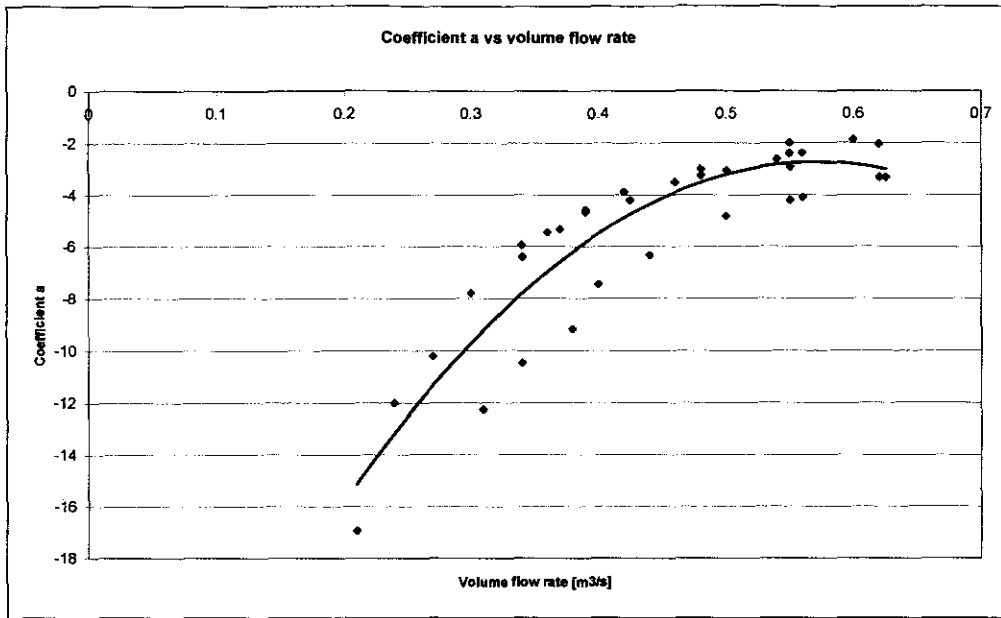


Figure A.15: Efficiency's a coefficient as a function of volume flow rate for flow between 0 and 1 m³/s

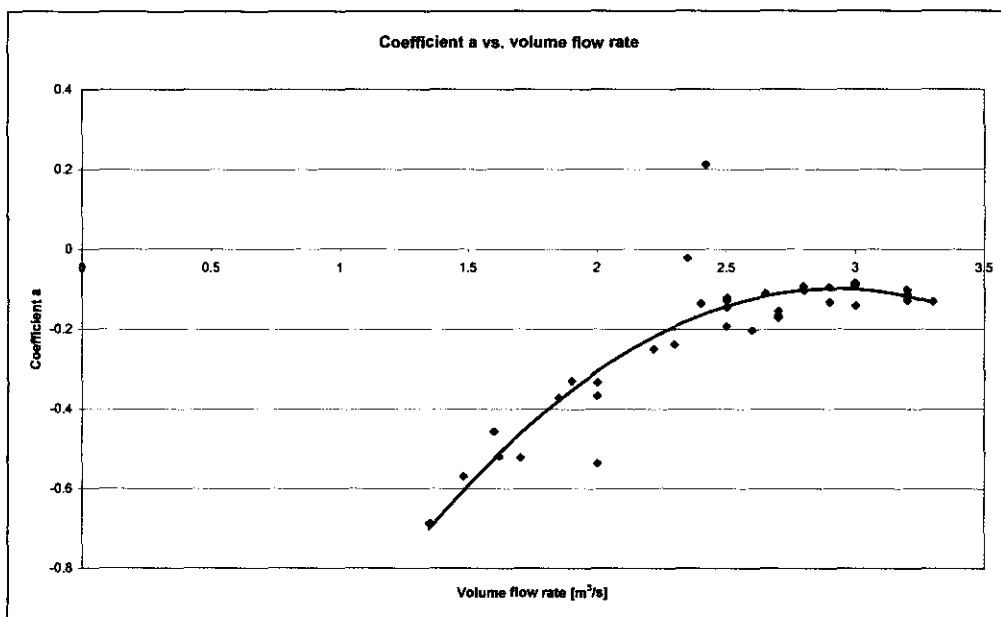


Figure A.16: Efficiency's a coefficient as a function of volume flow rate for flow between 1 and 5 m³/s

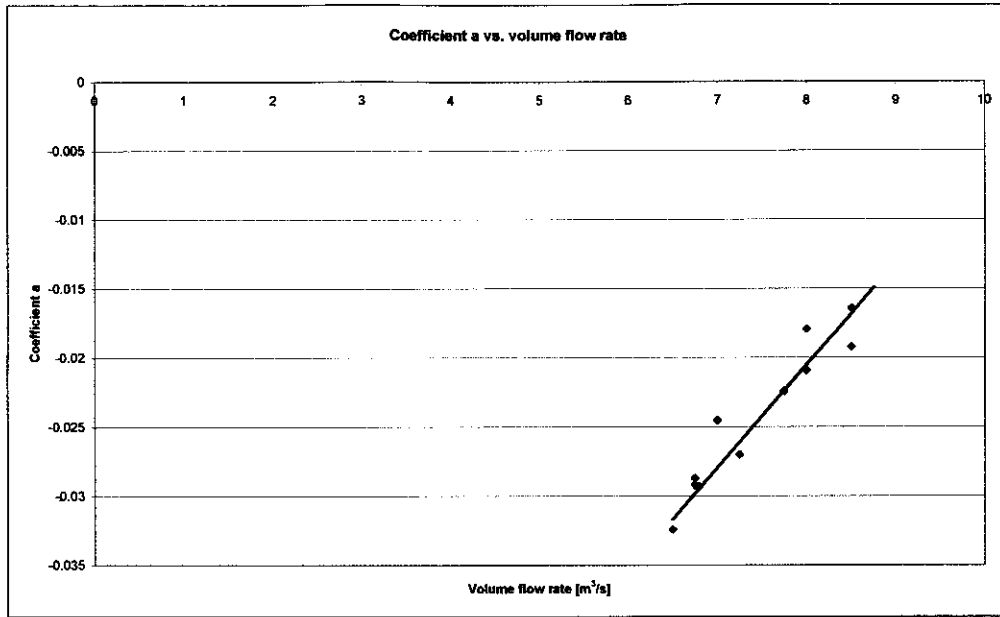


Figure A.17: Efficiency's a coefficient as a function of volume flow rate for flow between 5 and 15 m³/s

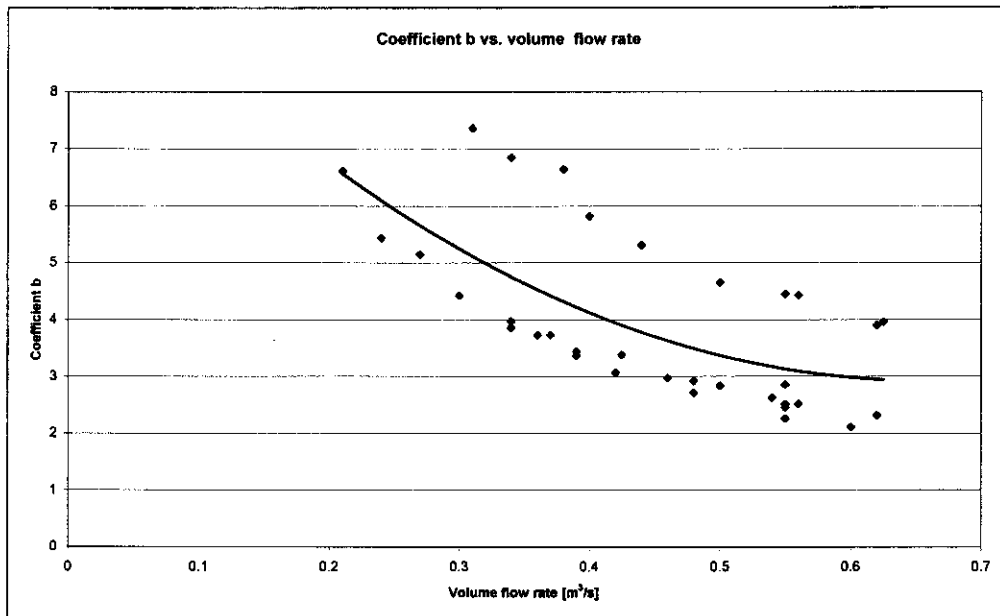


Figure A.18: Efficiency's b coefficient as a function of volume flow rate for flow between 0 and 1 m³/s

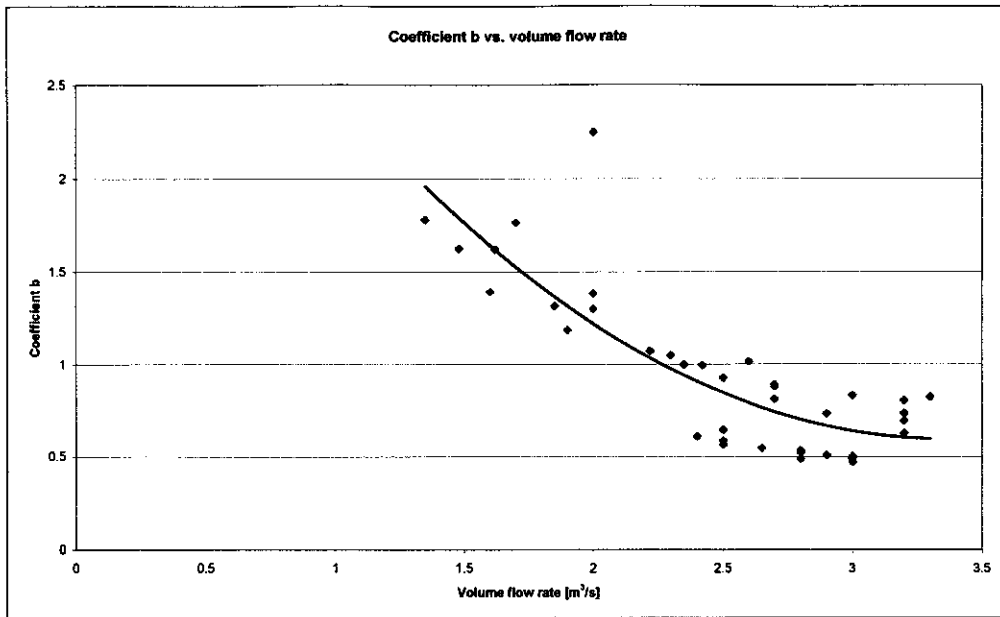


Figure A.19: Efficiency's b coefficient as a function of volume flow rate for flow between 1 and 5 m³/s

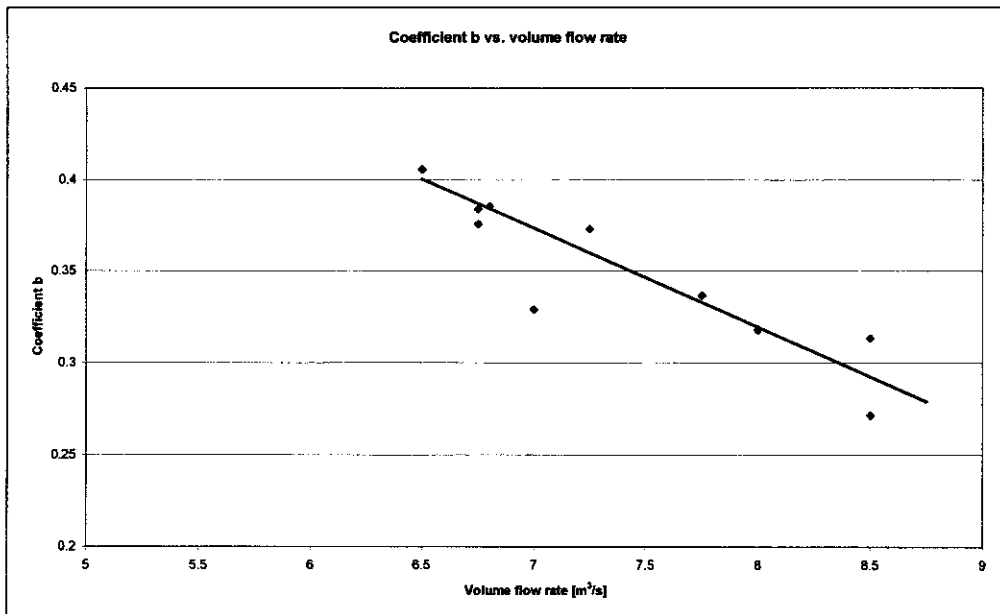


Figure A.20: Efficiency's b coefficient as a function of volume flow rate for flow between 5 and 15 m³/s

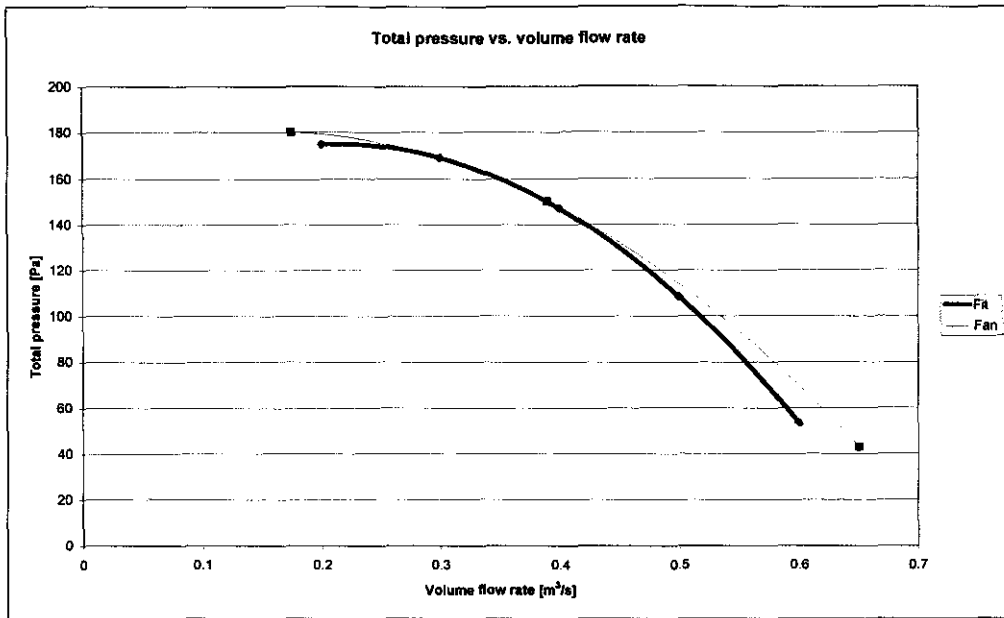


Figure A.21: Total pressure correlation for a volume flow rate between 0 and 1 m³/s

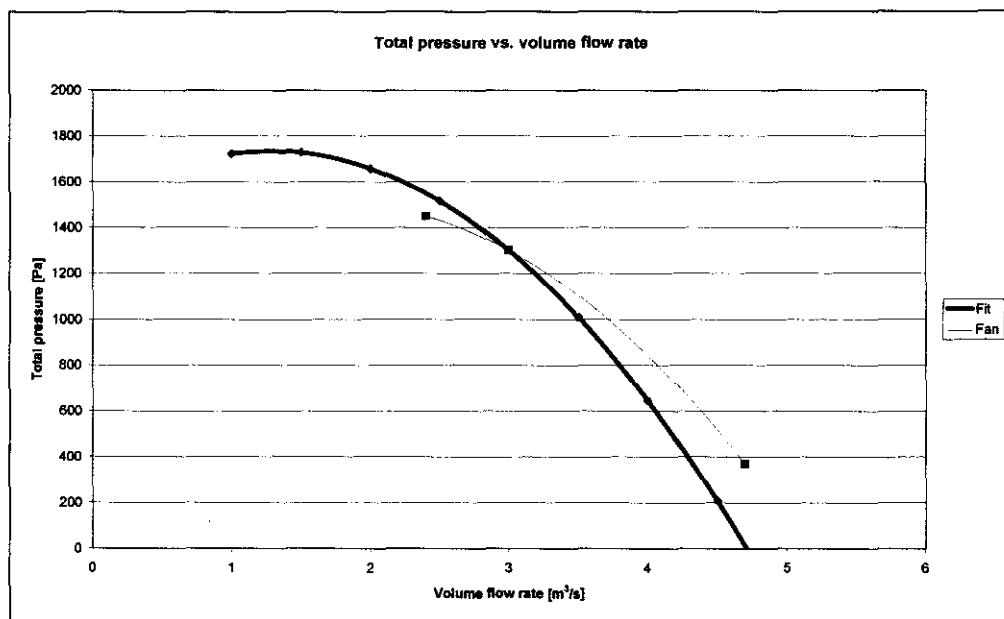


Figure A.22: Total pressure correlation for a volume flow rate between 1 and 5 m³/s

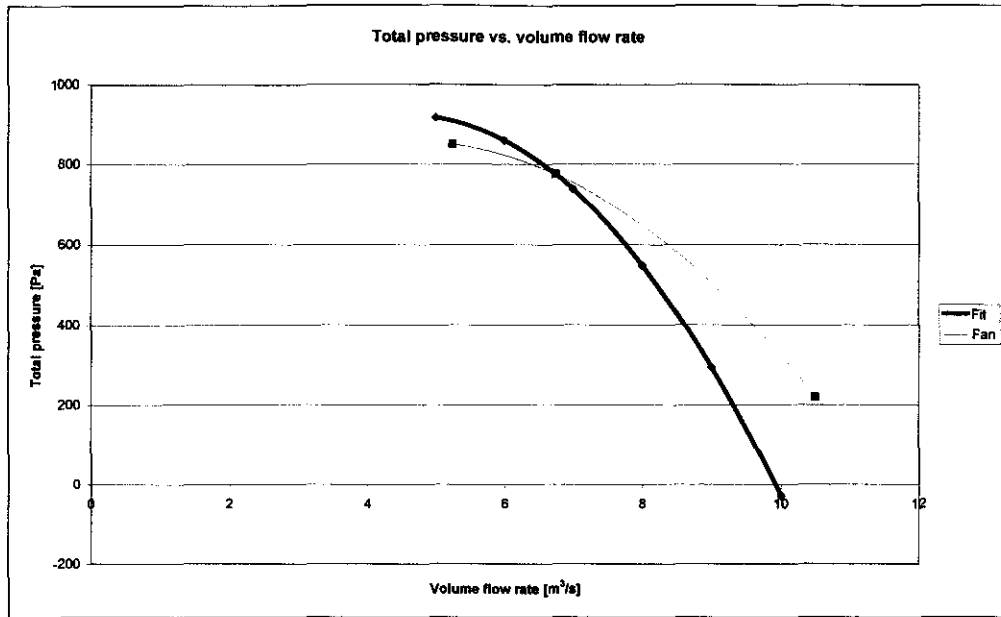


Figure A.23: Total pressure correlation for a volume flow rate between 5 and 15 m³/s

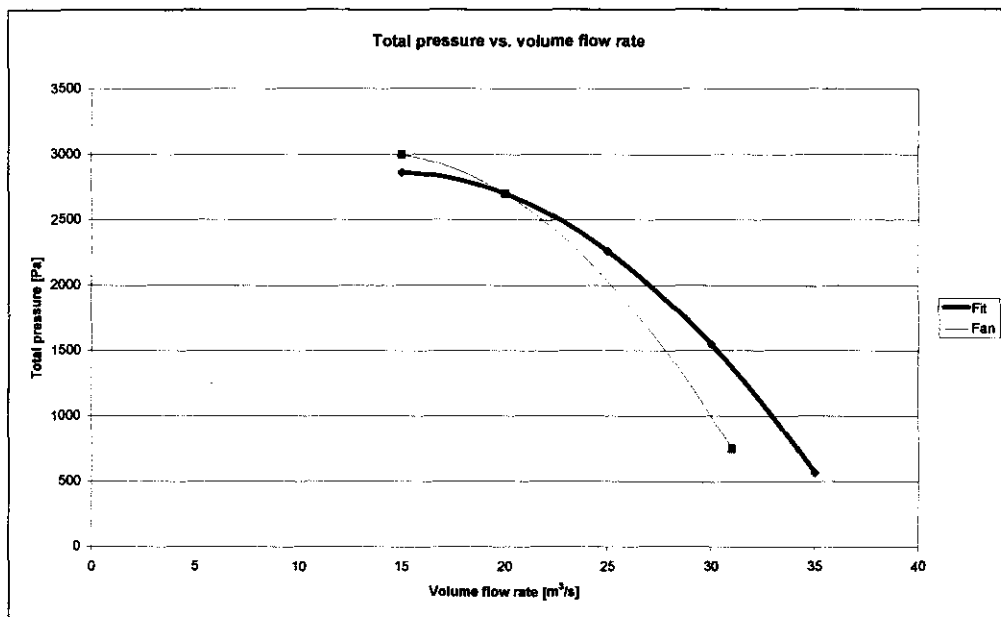


Figure A.24: Total pressure correlation for a volume flow rate between 15 and 30 m³/s

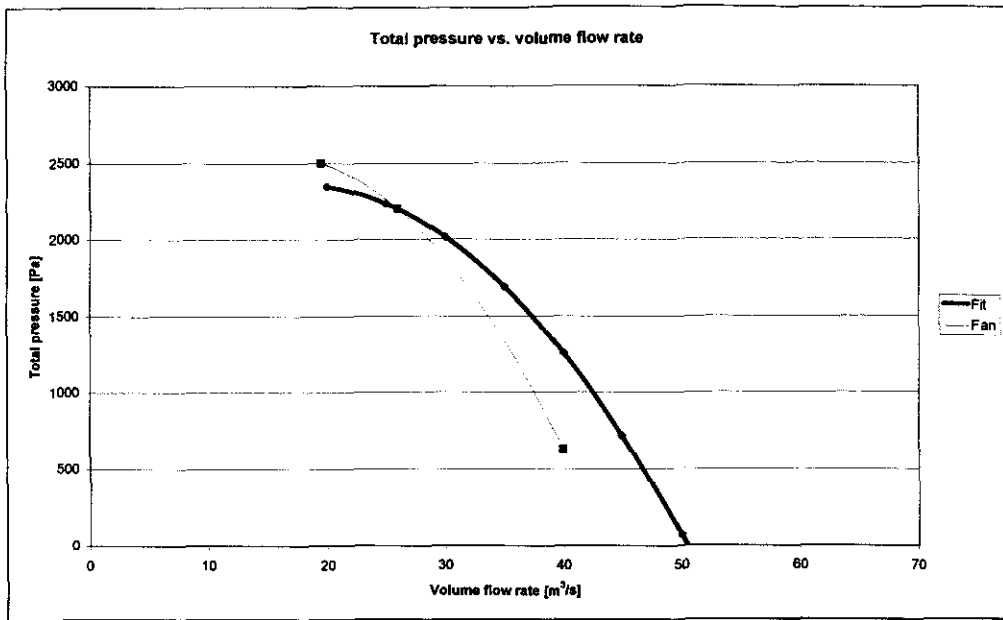


Figure A.25: Total pressure correlation for a volume flow rate between 30 and 60 m³/s

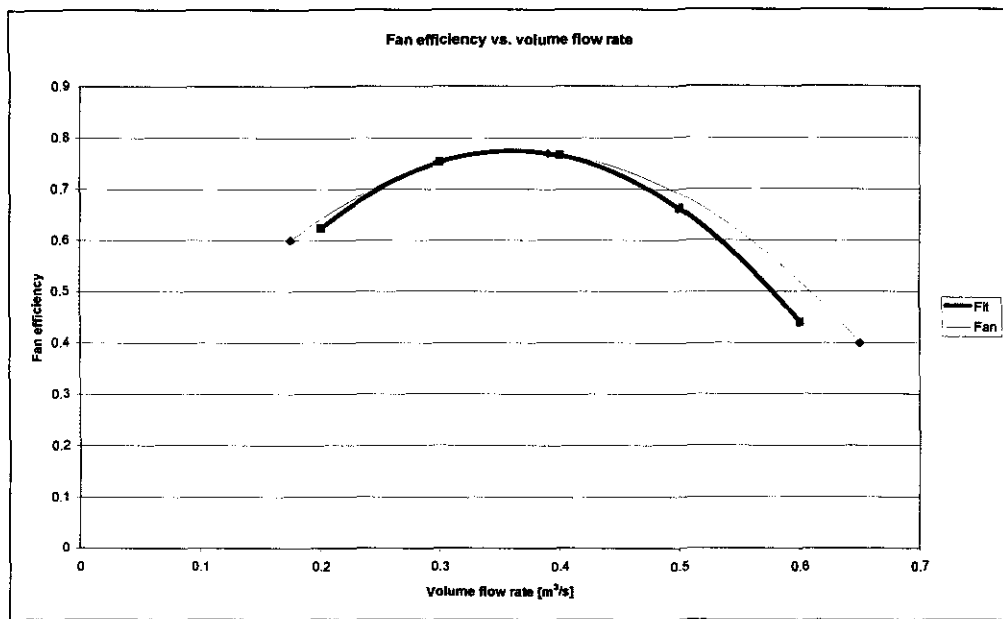


Figure A.26: Fan efficiency correlation for a volume flow rate between 0 and 1 m³/s

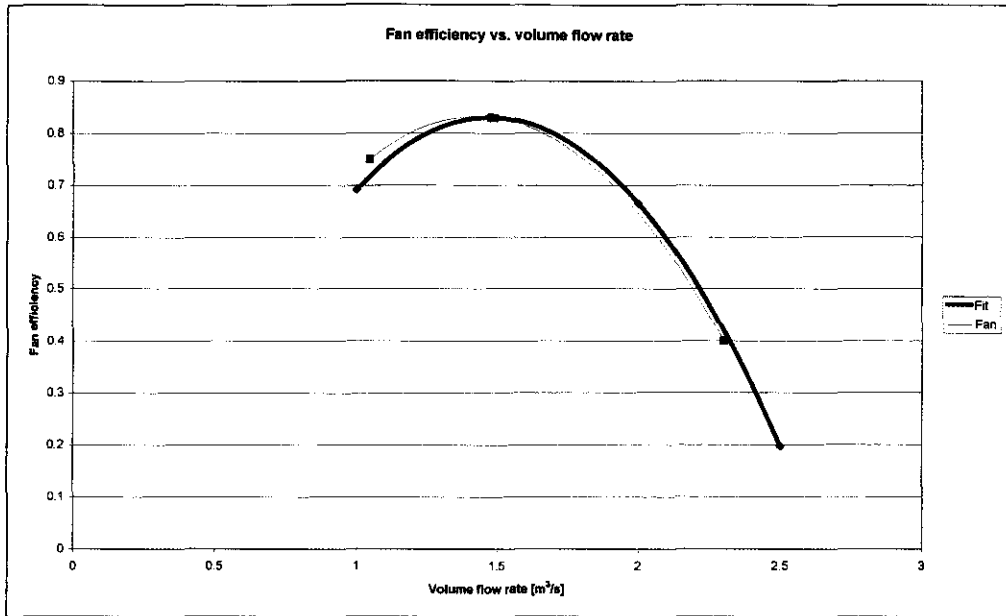


Figure A.27: Fan efficiency correlation for a volume flow rate between 1 and 5 m³/s

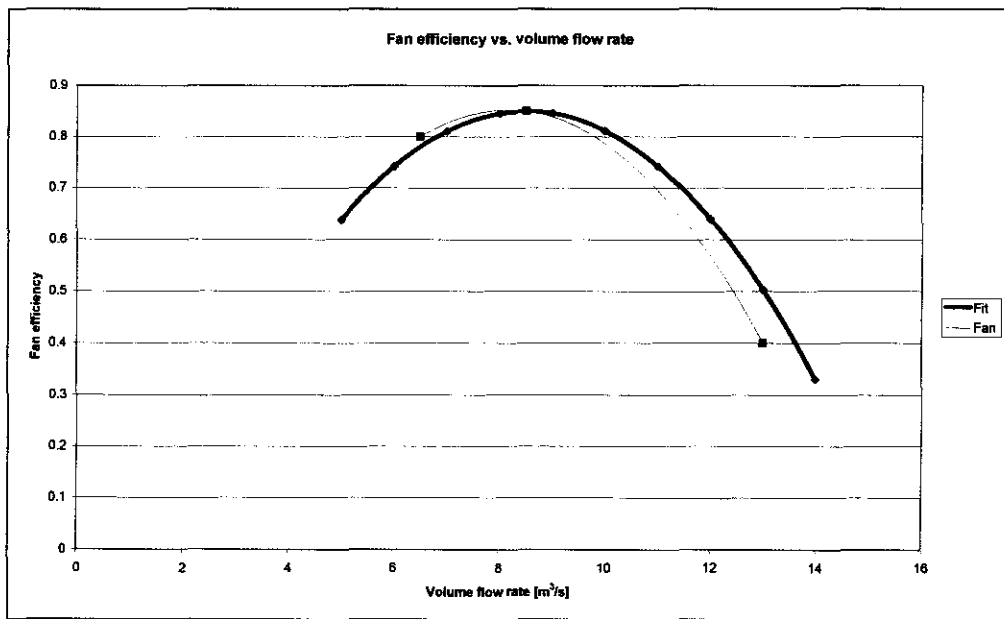


Figure A.28: Fan efficiency correlation for a volume flow rate between 5 and 15 m³/s

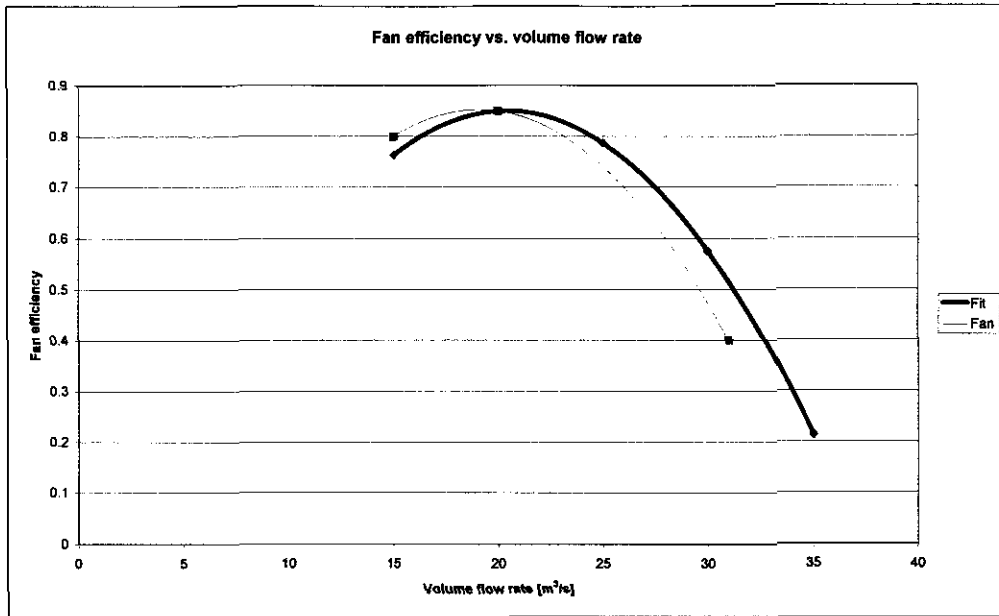


Figure A.29: Fan efficiency correlation for a volume flow rate between 15 and 30 m³/s

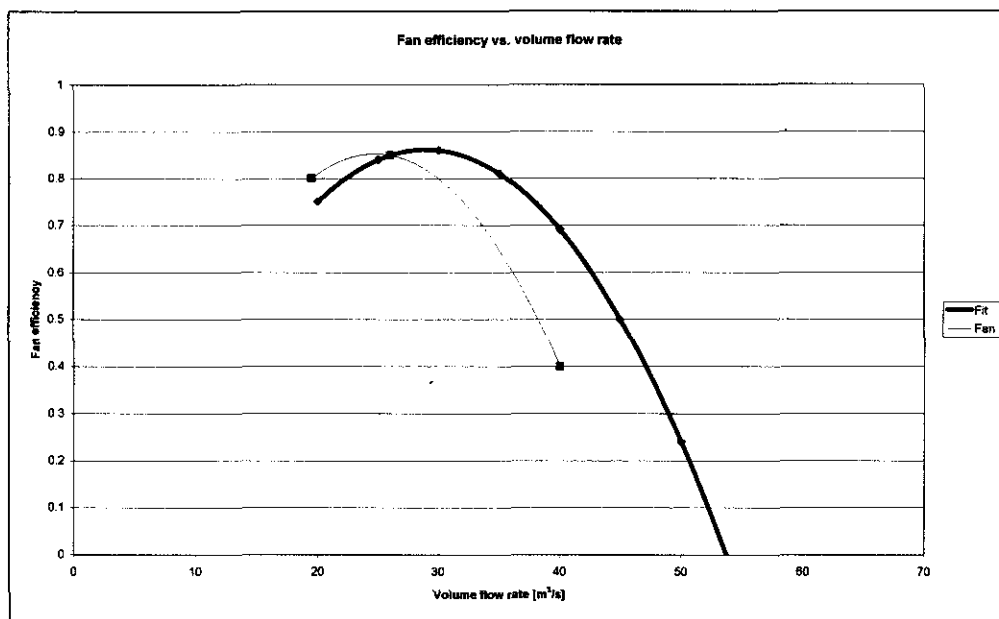


Figure A.30: Fan efficiency correlation for a volume flow rate between 30 and 60 m³/s

Air cooled liquid chiller

Parameters:

- f Active fraction of total capacity.
 a_n Regression coefficients for cooling capacity.
 b_n Regression coefficients for power.

Inputs:

Simulation:

- $m_{l(\text{evap})}$ Mass flow rate of water through the evaporator [kg/s].
 $T_{li(\text{evap})}$ Temperature of water entering evaporator [°C].
 $w_{ai(\text{cond})}$ Humidity ratio of air entering the condenser [kg_{vapour} / kg_{dry air}].
 $h_{ai(\text{cond})}$ Specific enthalpy of air entering the condenser [J/kg_{dry air}].

Interface:

- C_c Cooling capacity at the expected operational temperatures of the chiller [kW].
 P Compressor power at the expected operational temperatures of the chiller [kW].
 T_{aa} Expected dry bulb temperature of the air entering the condenser [°C].
 T_{ee} Expected temperature of water exiting evaporator [°C].

Outputs:

- $T_{le(\text{evap})}$ Temperature of water leaving evaporator [°C].
 Q_e Cooling capacity [kW].
 P_{wr} Power consumed by the compressor and fan [kW].

Internal variables:

- $T_{ai(\text{cond})}$ Dry bulb temperature of air entering condenser [°C].

Explicit equations:

$$T_{ai(\text{cond})} = \zeta(h_{ai(\text{cond})}, w_{ai(\text{cond})})$$

$$Q_e = f(a_0 + a_1 T_{le(\text{evap})} + a_2 T_{ai(\text{cond})})$$

With

$$a_2 = -0.0112C_c + 0.024$$

$$a_1 = 0.0265C_c - 0.2547$$

$$a_0 = C_c - (T_{ee} a_1 + T_{aa} a_2)$$

Coefficients a_2 and a_1 were found to be linear over a wide range of chillers with respect to cooling capacity as can be seen in figure A.31 and A.32. a_0 is calculated from the one given operating point obtained from the supplier or measurements. This implies that only one operating point is needed to obtain the mathematical model of a specific chiller.

$$P_{WR} = f(b_0 + b_1 T_{li(ewap)} + b_2 T_{at(cond)})$$

With

$$b_2 = 0.0079P + 0.3051$$

$$b_1 = 0.0184P - 0.1403$$

$$b_0 = P - (T_{ee} b_1 + T_{aa} b_2)$$

Coefficients b_2 and b_1 were found to be linear over a wide range of chillers with respect to compressor power as can be seen in figures A.33 and A.34. b_0 is calculated from the one given operating point obtained from the supplier or measurements. This implies that only one operating point is needed to obtain the mathematical model of a specific chiller.

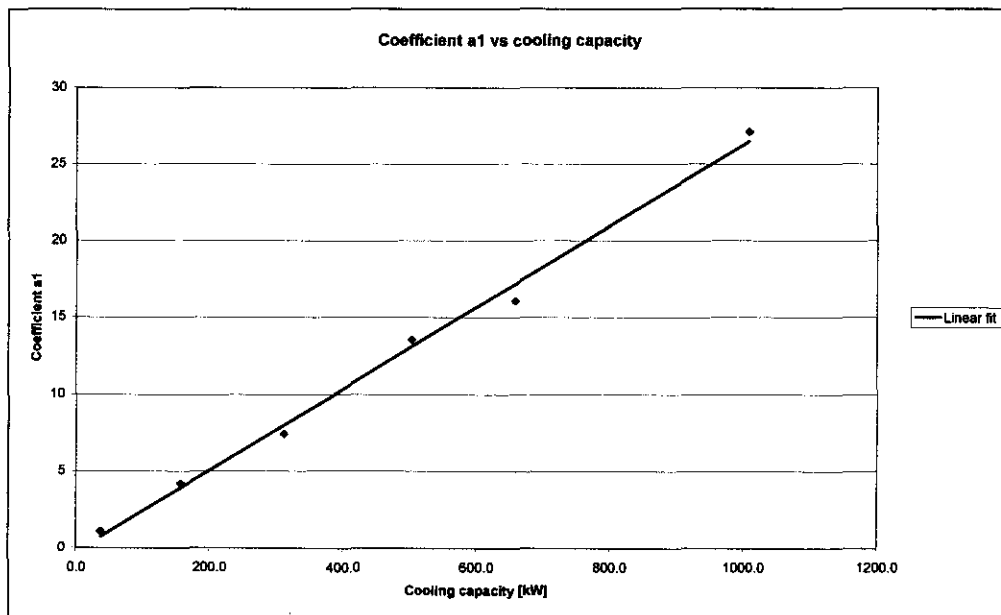


Figure A.31: Coefficient a1 as a function of cooling capacity

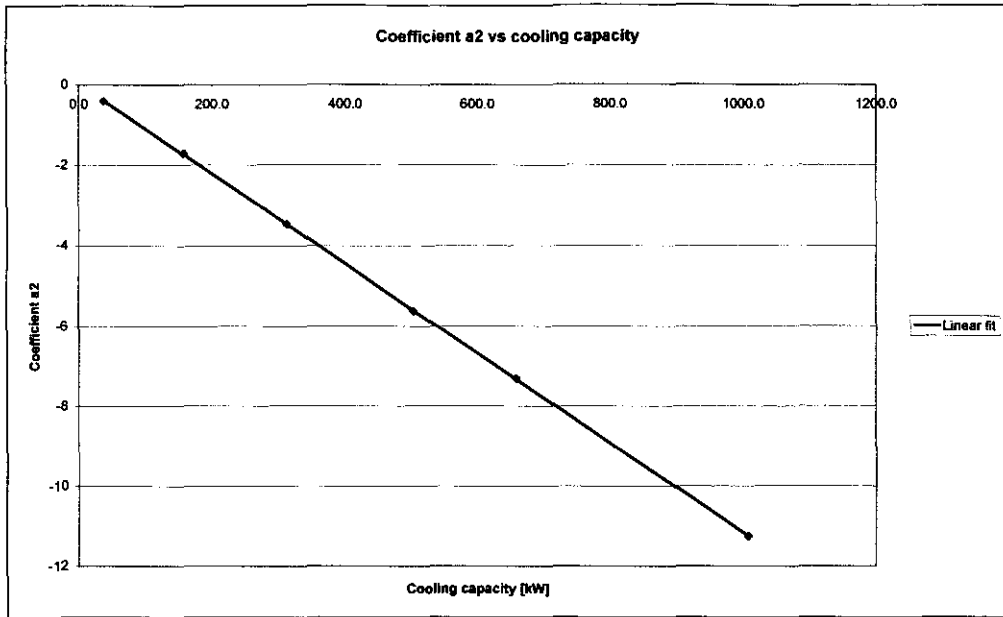


Figure A.32: Coefficient a_2 as a function of cooling capacity

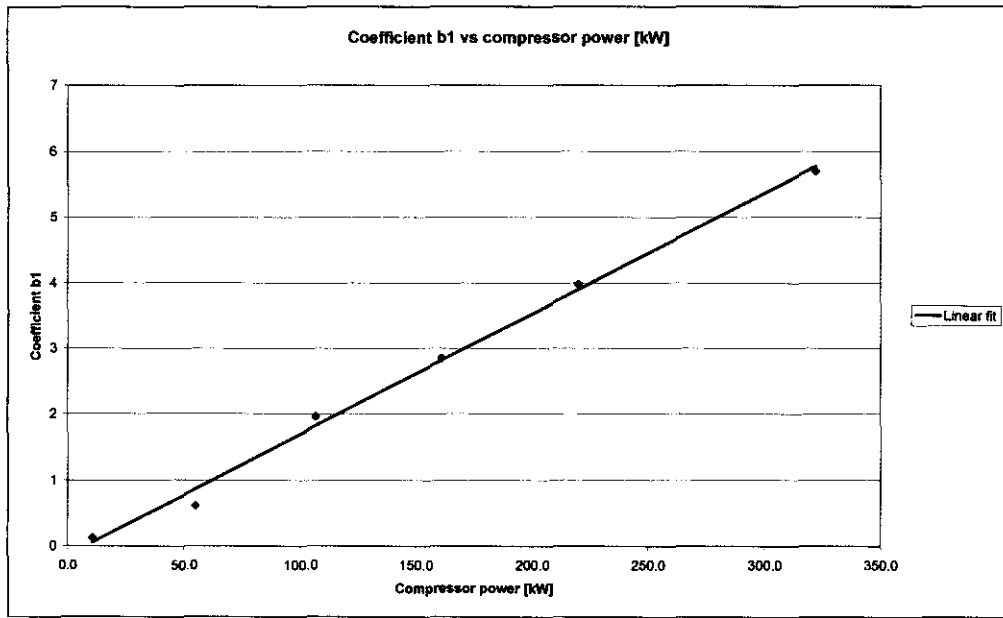


Figure A.33: Coefficient b_1 as a function of compressor power

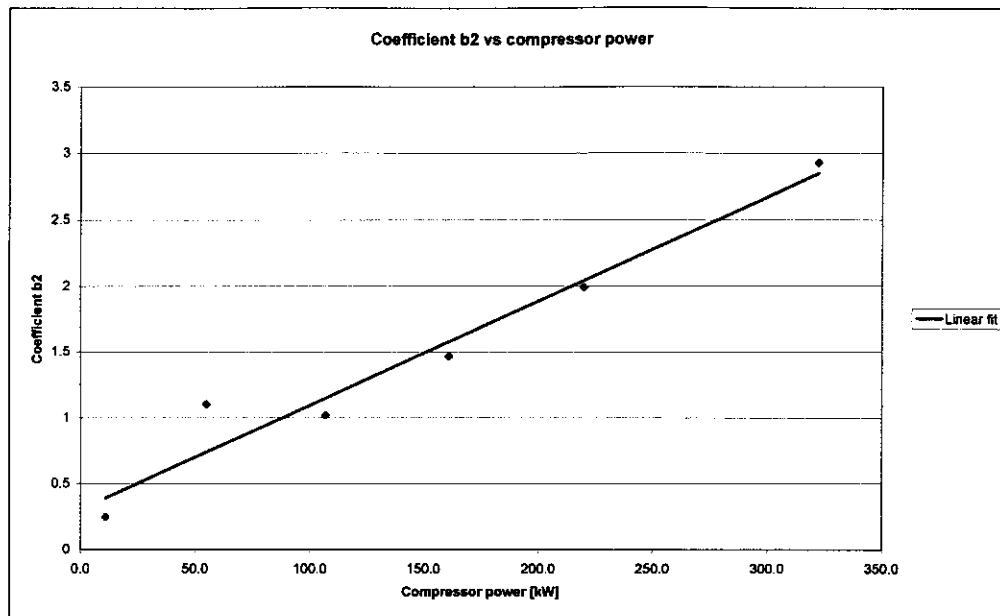


Figure A.34: Coefficient b2 as a function of compressor power

Water cooled liquid chiller

Parameters:

- f Active fraction of total capacity.
 a_n Regression coefficients for cooling capacity.
 b_n Regression coefficients for power.

Inputs:

Simulation:

- $m_{l(\text{cond})}$ Mass flow rate of water through the condenser [kg/s].
 $m_{l(\text{evap})}$ Mass flow rate of water through the evaporator [kg/s].
 $T_{li(\text{cond})}$ Temperature of water entering condenser [°C].
 $T_{li(\text{evap})}$ Temperature of water entering evaporator [°C].

Interface:

- C_c Cooling capacity at the expected operational temperatures of the chiller [kW].
 P Compressor power at the expected operational temperatures of the chiller [kW].
 T_{ce} Expected temperature of water exiting condenser [°C].
 T_{ee} Expected temperature of water exiting evaporator [°C].

Outputs:

$T_{le(cond)}$ Temperature of water leaving condenser [°C].

$T_{le(evap)}$ Temperature of water leaving evaporator [°C].

Q_e Cooling capacity [kW].

P_{wr} Power consumed by the compressor [kW].

Explicit equations:

$$Q_e = f(a_0 + a_1 T_{le(cond)} + a_2 T_{le(evap)})$$

With

$$a_2 = 0.0266Cc + 2.8714$$

$$a_1 = -0.01Cc + 0.2289$$

$$a_0 = Cc - (T_{ce}a_1 + T_{ee}a_2)$$

Coefficients a_2 and a_1 were found to be linear over a wide range of chillers with respect to cooling capacity as can be seen in figure A.35 and A.36. a_0 is calculated from the one given operating point obtained from the supplier or measurements. This implies that only one operating point is needed to obtain the mathematical model of a specific chiller.

$$P_{wr} = f(b_0 + b_1 t_{li(cond)} + b_2 t_{li(evap)})$$

With

$$b_2 = 0.007P + 0.3549$$

$$b_1 = 0.0124P + 0.4207$$

$$b_0 = P - (t_{ce}b_1 + t_{ee}b_2)$$

Coefficients b_2 and b_1 were found to be linear over a wide range of chillers with respect to compressor power as can be seen in figure A.37 and A.38. b_0 is calculated from the one given operating point from the supplier or measurements. This implies that only one operating point is needed to obtain the mathematical model of a specific chiller.

$$T_{le(cond)} = T_{li(cond)} - \frac{Q_e + P_{wr}}{m_{l(cond)}c_{pl}}$$

$$T_{le(evap)} = T_{li(evap)} - \frac{Q_e}{m_{l(evap)}c_{pl}}$$

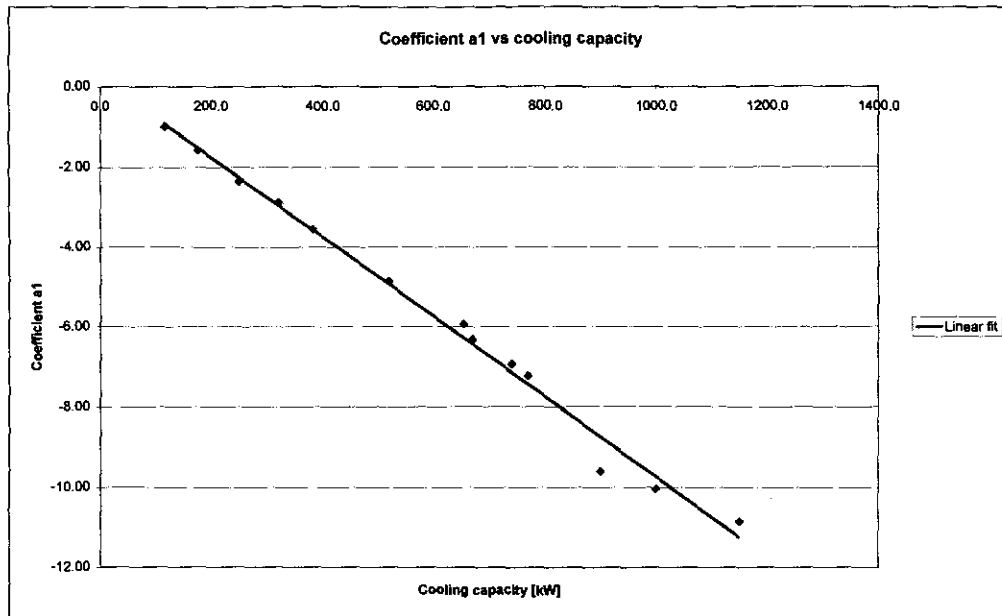


Figure A.35: Coefficient a1 as a function of cooling capacity

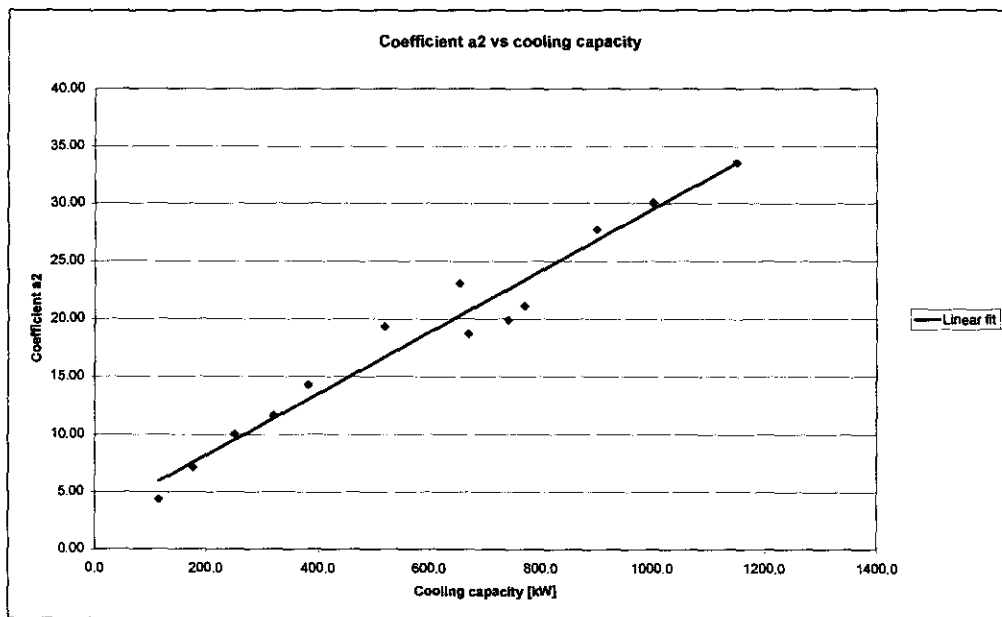


Figure A.36: Coefficient a2 as a function of cooling capacity

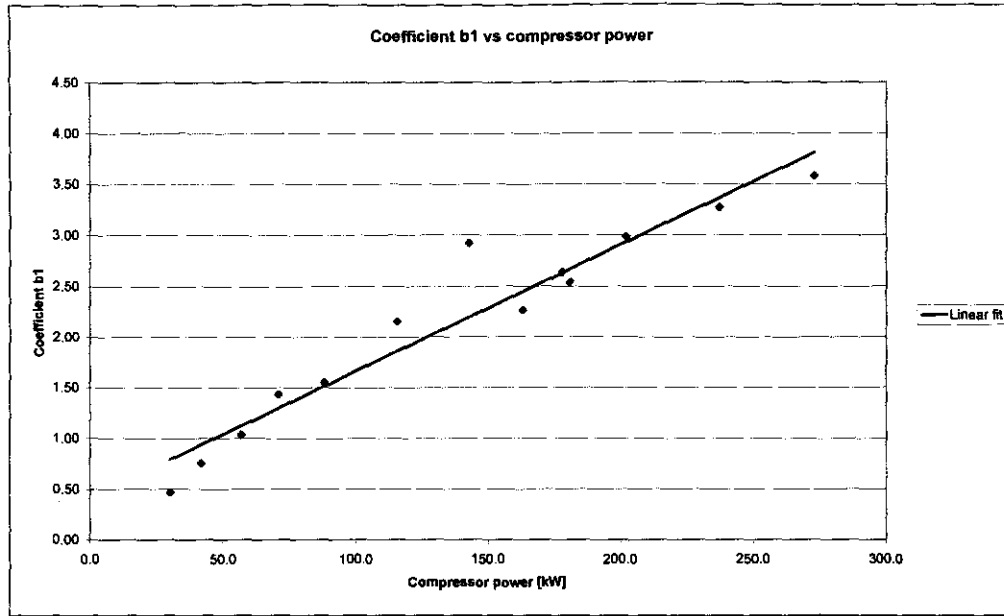


Figure A.37: Coefficient b1 as a function of compressor power

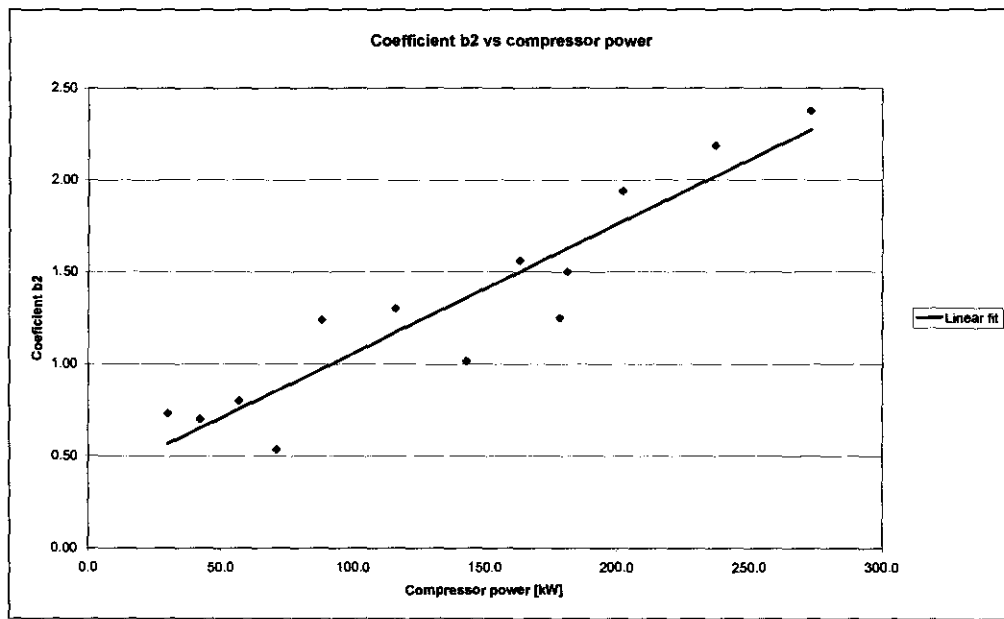


Figure A.38: Coefficient b2 as a function of compressor power

Coil

Parameters:

- A_f Coil face area [m^2].
 R Ratio A_o/A_f .
 f Fraction of m_l that is not bypassed and passes through the coil.
 a_j Correlation coefficients for the h_o versus V_a relation with $j = 0$ to 2.

Inputs:

- m_a Mass flow rate of dry air at the inlet [kg/s].
 w_{ai} Humidity ratio of air entering at the inlet [kg_{vapour}/kg_{dryair}].
 h_{ai} Specific enthalpy of air entering at the inlet [J/kg_{dryair}].
 m_l Mass flow rate of liquid at inlet [kg/s].
 T_{li} Temperature of liquid entering at inlet [$^{\circ}C$].

Outputs:

- w_{ae} Humidity ratio of air leaving at the outlet [kg_{vapour}/kg_{dryair}].
 h_{ae} Specific enthalpy of air leaving at the outlet [J/kg_{dryair}].
 T_{le} Temperature of liquid leaving at outlet [$^{\circ}C$].

Internal variables:

- A_o Total outside heat transfer area [m^2].
 ρ_a Air density [kg/m^3].
 T_{ai} Dry-bulb temperature of air at the inlet [$^{\circ}C$].
 V_a Air face velocity [m/s].
 h_o Outside surface convection heat transfer coefficient [$W/m^2^{\circ}C$].
 T_{ae} Dry-bulb temperature of air at the outlet [$^{\circ}C$].
 w_{adp} Humidity ratio of saturated air at t_{li} [$kg_{vapour}/kg_{dry air}$].
 h_{ad} Specific enthalpy of air at the outlet [$J/kg_{dry air}$].
 S Sensible heat ratio.

Explicit equations:

$$m_l = fm_l$$

$$\rho_a = \zeta(h_{ai}, w_{ai})$$

$$T_{ai} = \zeta(h_{ai}, w_{ai})$$

$$V_a = \frac{m_a}{\rho_a A_f}$$

$$h_o = a_0 + a_1 V_a + a_2 V_a^2$$

$$A_0 = RA_f$$

If $T_{li} \geq \zeta$ (T_{ai} , saturated) the coil is dry which means that $S = 1$, $w_{adp} = w_{ai}$ and $h_{ad} = h_{ai}$ and the following equations apply.

$$w_{ae} = w_{ai}$$

$$T_{ae} = \frac{\left[e^{\frac{h_o A_o}{m_a c_{pa}} \left(\frac{1}{m_a c_{pa}} - \frac{1}{m_l c_{pl}} \right)} - 1 \right] T_{li} - \left[\frac{m_a c_{pa}}{m_l c_{pl}} - 1 \right] T_{ai}}{e^{\frac{h_o A_o}{m_a c_{pa}} \left(\frac{1}{m_a c_{pa}} - \frac{1}{m_l c_{pl}} \right)} - \frac{m_a c_{pa}}{m_l c_{pl}}}$$

$$h_{ae} = \zeta(T_{ae}, w_{ae})$$

$$T_{le} = T_{li} - \frac{m_a c_{pa}}{m_l c_{pl}} (T_{ae} - T_{ai})$$

Else the coil is wet and the following equations apply

$$w_{adp} = \xi(T_{li}, \text{saturated})$$

Simultaneous equations:

$$h_{ae} = h_{ai} + \frac{h_o A_o (T_{le} - T_{ai}) - (T_{li} - T_{ae})}{m_a S \ln \left(\frac{T_{le} - T_{ai}}{T_{li} - T_{ae}} \right)}$$

$$T_{le} = T_{li} - \frac{m_a}{m_l c_{pl}} (h_{ae} - h_{ai})$$

$$T_{ae} = T_{ai} - \frac{w_{ai} - w_{ae}}{w_{ai} - w_{adp}} (T_{ai} - T_{li})$$

$$w_{ae} = \zeta(T_{ae}, h_{ae})$$

$$h_{ad} = \zeta(T_{ai}, w_{ae})$$

$$S = \frac{h_{ad} - h_{ae}}{h_{ai} - h_{ae}}$$

Explicit equation:

$$T_{le} = f T_{le} + (1 - f) T_{li}$$

Cooling tower

Parameters:

- a_j Correlation coefficients for the UA versus m_l and t_{li} relation with $j = 0$ to 5
 m_a Mass flow rate of outdoor air through the tower [kg/s]
 Pwr_{fan} Power required by the built-in fan [W]
 Pwr_{pump} Power required by the built-in pump [w]

Inputs:

- w_{ai} Humidity ration of air entering at inlet [kg vapour/kg dry air]
 h_{ai} Specific enthalpy of air entering at inlet [J/kg dry air]
 m_l Mass flow rate of liquid at inlet [kg/s]
 T_{li} Temperature of liquid entering at inlet [°C]

Outputs:

- w_{ae} Humidity ratio of air leaving at outlet [kg vapour/kg dry air]
 h_{ae} Specific enthalpy of air leaving at outlet [J/kg dry air]
 T_{le} Temperature of liquid leaving at outlet [°C]
 Pwr Input power required [W]

Internal variables:

- h_{si} Specific enthalpy of saturated air entering at t_{le} [J/kg dry air]
 h_{se} Specific enthalpy of saturated air entering at t_{li} [J/kg dry air]

Explicit equations:

$$UA = a_0 + a_1 m_l + a_2 T_{li} + a_3 m_l^2 + a_4 T_{li}^2 + a_5 m_l T_{li}$$

$$h_{se} = 10^3 (16.66326 + 4.701617 T_{li} - 0.11237 T_{li}^2 + 0.004991 T_{li}^3 - 0.17197 p_{barom} - 0.01364 p_{barom} T_{li} + 0.000493 p_{barom} T_{li}^2 - 2.9 \times 10^{-5} p_{barom} T_{li}^3)$$

$$Pwr = Pwr_{fan} + Pwr_{pump}$$

Simultaneous equations:

$$h_{si} = 10^3 (16.66326 + 4.701617 T_{le} - 0.11237 T_{le}^2 + 0.004991 T_{le}^3 - 0.17197 p_{barom} - 0.01364 p_{barom} T_{le} + 0.000493 p_{barom} T_{le}^2 - 2.9 \times 10^{-5} p_{barom} T_{le}^3)$$

$$Q_a = UA \left[\frac{(h_{si} - h_{ai}) - (h_{se} - h_{ae})}{\ln \left(\frac{h_{si} - h_{ai}}{h_{se} - h_{ae}} \right)} \right]$$

$$Q_a = m_a (h_{ae} - h_{ai})$$

$$Q_a = m_l c_{pl} (T_{li} - T_{le})$$

Explicit equations:

$$T_{ae} = \xi(h_{ae}, \text{saturated})$$

Heater

Parameters:

Q_a^* Heater capacity [W]

Inputs:

m_a Mass flow rate of air entering the heater [kg/s]

h_{ai} Specific enthalpy of air entering heater [J/kg_{dry air}]

w_{ai} Humidity ration of air entering the heater [kg_{vapour}/kg_{dry air}]

Outputs:

h_{ae} Specific enthalpy of air leaving the heater [J/kg_{dry air}]

w_{ae} Humidity ratio of air leaving the heater [kg_{vapour}/kg_{dry air}]

Explicit equations:

$$h_{ae} = h_{ai} + \frac{Q_a}{m_a}$$

$$w_{ae} = w_{ai}$$

PID controller

Parameters:

t Present time

Δt Size of simulation time step

θ_{lo} Throttling range (proportional band) low of controlled variable

θ_{hi} Throttling range (proportional band) high of controlled variable

ϕ_{lo} Low potential of final control element

ϕ_{hi} High potential of final control element

k_I Integral control factor

k_D Derivative control factor

Inputs:

θ Value of controlled variable

Outputs:

ϕ Potential of final control element

Internal variables:

θ_{sp} Set-point value of controlled variable

ϕ_{sp}	Set-point potential of final control element
k_p	Proportional control factor
ε	Error function

Explicit equations:

$$\theta_{sp} = \frac{\theta_{lo} + \theta_{hi}}{2}$$

$$\phi_{sp} = \frac{\phi_{lo} + \phi_{hi}}{2}$$

$$k_p = \frac{\phi_{lo} - \phi_{hi}}{\theta_{lo} - \theta_{hi}}$$

$$\varepsilon_t = \theta_t - \theta_{sp}$$

if $\theta_{t-\Delta t} \leq \theta_{lo}$ then $\phi_t = \phi_{lo}$

else if $\theta_{t-\Delta t} \geq \theta_{hi}$ then $\phi_t = \phi_{hi}$

else $\phi_t = k_p \varepsilon_{t-\Delta t} + k_i \sum_{t=0}^{t-\Delta t} [\varepsilon \Delta t] + k_D \left[\frac{\varepsilon_{t-\Delta t} - \varepsilon_{t-2\Delta t}}{\Delta t} \right]$

Step controller

Parameters:

t	Present time.
Δt	Size of simulation time steps.
n	Number of steps.
$\theta_{L,1...n}$	Loading set-points of controlled variable.
$\theta_{U,1...n}$	Unloading set-points of controlled variable.
$\phi_{L,1...n}$	Loading potential steps of final control element.
$\phi_{U,1...n}$	Unloading potential steps of final control element.

Inputs:

θ	Value of controlled variable.
----------	-------------------------------

Outputs:

ϕ	Potential of final control element.
--------	-------------------------------------

Explicit equations:

If $\theta_{t-\Delta t} \geq \theta_{t-2\Delta t}$ and $\theta_{t-\Delta t} \geq \theta_{L,i}$ then $\phi_t = \phi_{L,i}$

Else if $\theta_{t-\Delta t} \leq \theta_{t-2\Delta t}$ and $\theta_{t-\Delta t} \leq \theta_{U,i}$ then

APPENDIX B
VISUALQEC IN ACTION

The logo for VisualQEC is centered within a rectangular border. It features the text "Mass Energy Simulation Solver" in a small, sans-serif font at the top. Below this, the word "VisualQEC" is written in a large, bold, sans-serif font. Underneath "VisualQEC", the words "Dynamic" and "Quickeasy" are written in a smaller, sans-serif font, positioned to the left and right of the "Q" respectively.

Mass Energy Simulation Solver
VisualQEC
Dynamic Quickeasy

Figure B.1: The VISUALQEC thermal and energy simulation platform

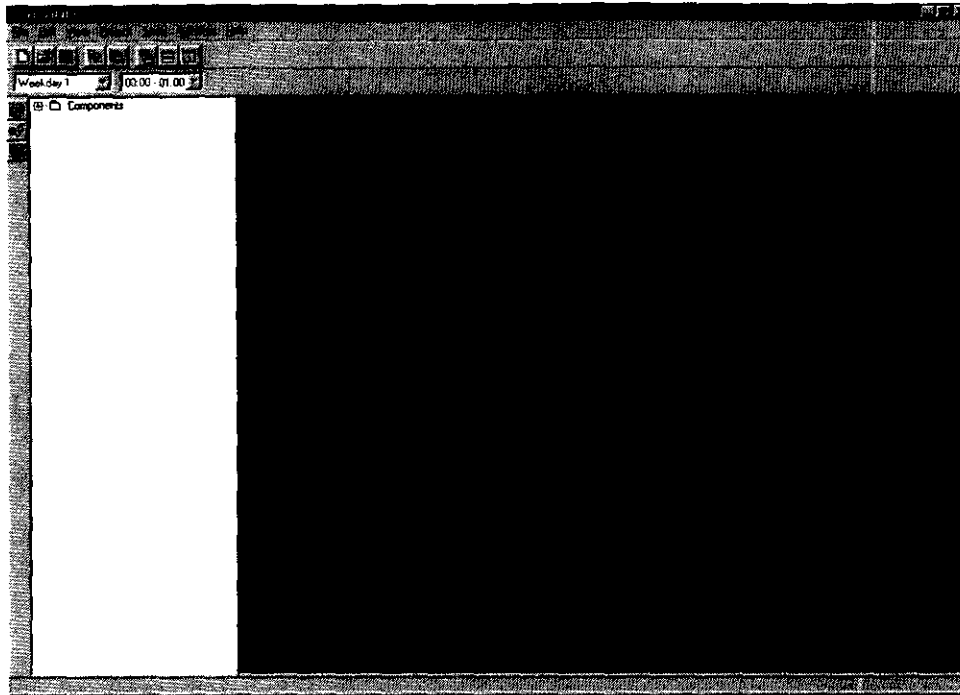


Figure B.2: The VISUALQEC user interface without the expanded component list

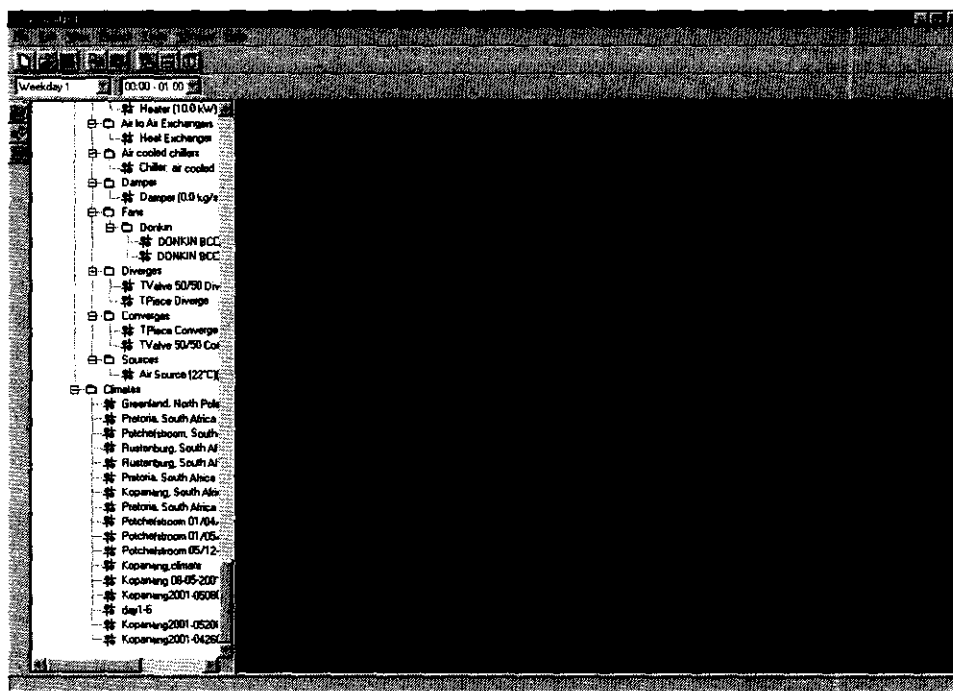


Figure B.3: The VISUALQEC user interface with expanded component list

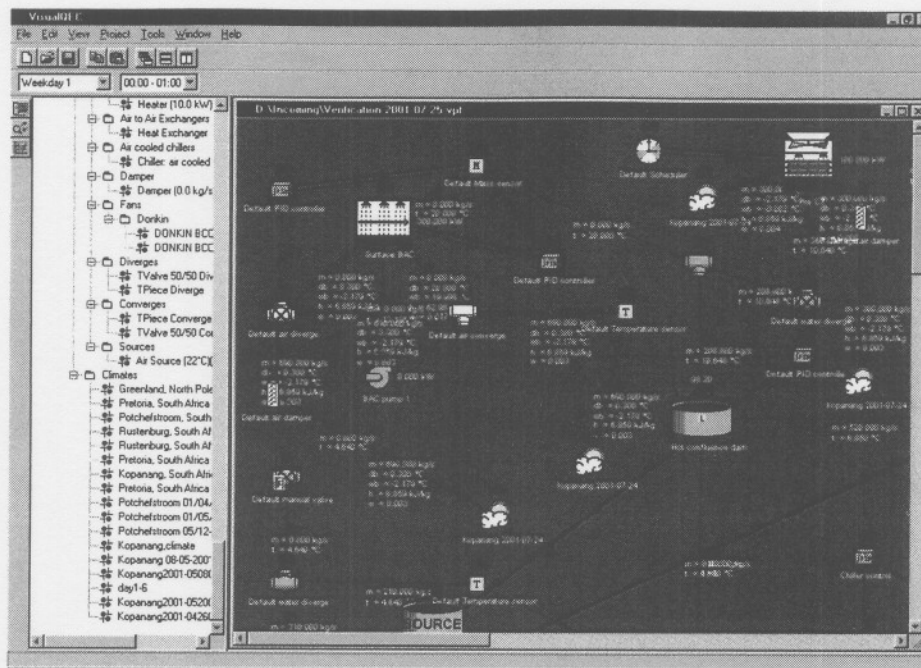


Figure B.4: A configured system on the simulation user interface

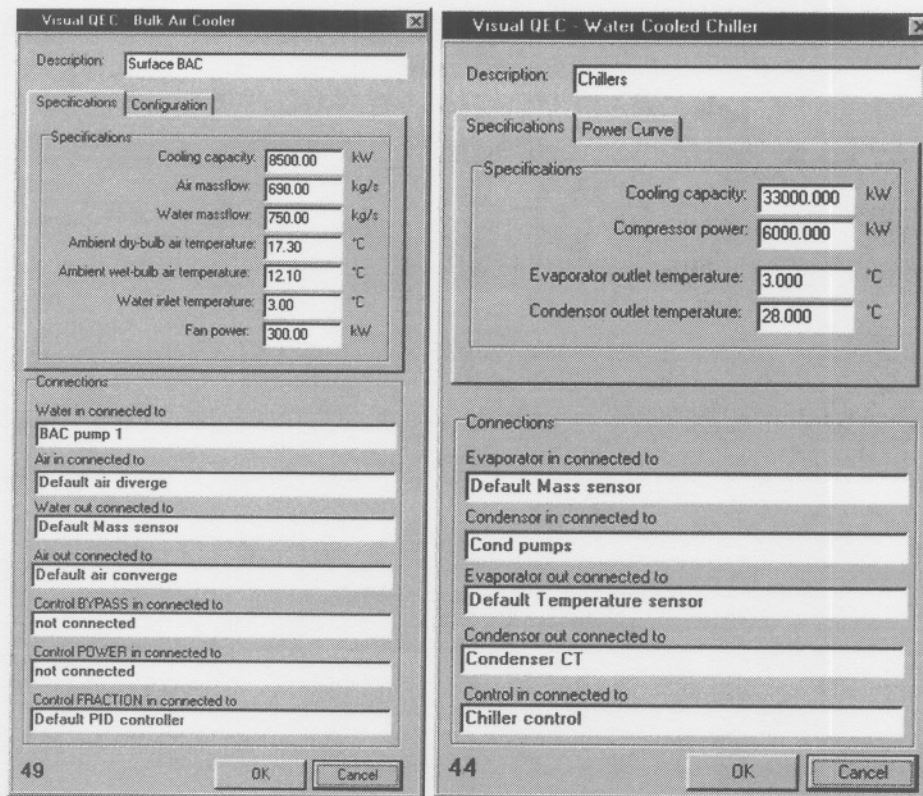


Figure B.5: Component configuration interfaces

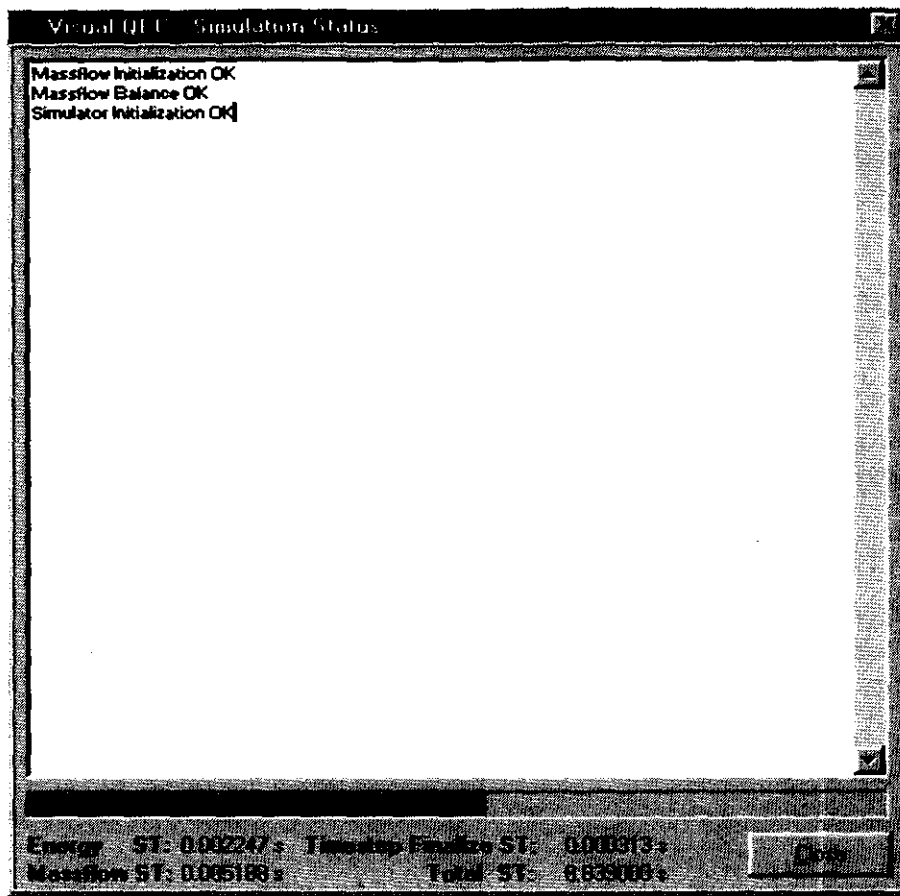


Figure B.6: Simulation status and time indicator screen

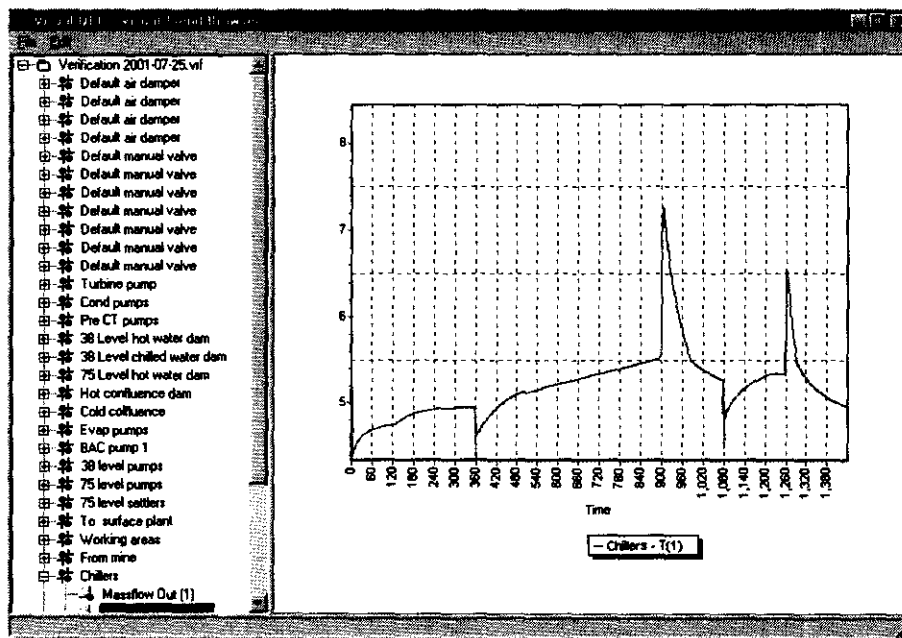


Figure B.7: Simulation trend and result browser

SYSTEM COMPONENT CODE EXTRACT (PUMP)

```
unit PumpQEC;
{*****}
{* NAME: PumpQEC *}
{* DESC: Pump component declarations. *}
{*****}
interface
uses
  Globals,
  Windows,
  Graphics,
  IniFiles,
  Forms,
  ElementQEC,
  FileHandling,
  ConnectionQEC,
  Psychrometric,
  Various,
  Classes,
  OutputQEC;

type
  TPumpDE = record
    ROTORDIAMETER : EnergyFloat;
    DRIVEMOTOREFFICIENCY : EnergyFloat;
    PUMPCONTROL : PControl;
    FLOWCONFIGURATION : FConfiguration;
    WORKINGPOINT : WPoint;
    ROTATIONSPEED : EnergyFloat;
    MASSFLOW : EnergyFloat;
    WATERHEIGHT : EnergyFloat;
    TOTALH : EnergyFloat;
    VSDFLOW : EnergyFloat;
    QPOINTS : Array[1..3] of EnergyFloat;
    HPOINTS : Array[1..3] of EnergyFloat;
    EPOINTS : Array[1..3] of EnergyFloat;
    ACOEFFICIENT : Array[1..3] of EnergyFloat;
    BCOEFFICIENT : Array[1..3] of EnergyFloat;
    ctWATER_IN : TWaterFlowConnectionQEC;
    ctWATER_OUT : TWaterFlowConnectionQEC;
    ctCONTROL_MASSFLOW : TControlConnectionQEC;
  end;

  TPumpQEC = class(TOneInOneOutElementQEC)
  protected
    PumpDE : TPumpDE;
  public
    constructor Create( QECRecord : Pointer;
      X,Y : Integer); override;
    constructor CreateCopy(aElement : TElementQEC); override;
    procedure Store ( var Stream: TFileRecordStream); override;
    constructor Load ( var Stream: TFileRecordStream); override;
    function CreateQECInterface( aElement : pointer): TForm; override;
    procedure InitializeDE; override;
    procedure HandleRecord; override;
    procedure ExportRecord(Filename : String); override;
    procedure AssignGenericLinks; override;
    procedure MakeConnection( aConnection : Pointer;
      aPort : Byte); override;
    procedure BreakConnection(aPort : Byte;
      aPortType : PortType); override;
    procedure GetOpenPortConnectionTypes( var aConnection : ConnectionSet;
      aPortType : PortType); override;
    procedure GetOpenPorts( var Ports : PortSet;
      aPortType : PortType;
      aConnection : ConnectionSet); override;
    function GetPortName( aPort : Byte) : String; override;

    function ElementStatus : EStatus; override;

    procedure SetControlableValue( aPort : Byte;
      aValue: EnergyFloat); override;
    function GetControlableValue( aPort : Byte; aTime: TimeType):EnergyFloat; override;
```

```

function DoInternalMassflow(const aTime : TimeType;
    aFlag : FType): Boolean; override;
procedure CalculateCoefficients;
function DoSimulationInitialize (const aTime : TimeType) : Boolean; override;
function DoInternalANDOutletState(const aTime : TimeType): Boolean; override;
procedure Report(const aTime : TimeType; var Stream : TFileRecordStream); override;
function GetNominalHeatingCooling(InletConnection: TConnectionQEC): EnergyFloat; override;

{private}
private
    tle0 : EnergyFloat; {Previous Timestep temperature}
end;
    {This is initialized in DoSimulationInitialize}

implementation

uses
    PumpFrm;

const
    ptIN_WATER = 0;
    ptOUT_WATER = 1;
    pdN_CONTROL_MASSFLOW = 2;

constructor TPumpQEC.Create( QECRecord : Pointer;
    X, Y : Integer);
begin
    inherited Create(QECRecord, X, Y);
    m_ELEMENTHEADER.Person := 'Martin den Boef';
    m_ELEMENTHEADER.Date := '2001/01/15';
    InitializeDE;
    HandleRecord;
    CalculateCoefficients;
end;

constructor TPumpQEC.CreateCopy(aElement : TElementQEC);
begin
    inherited CreateCopy(aElement);
    PumpDE := TPumpQEC(aElement).PumpDE;
    InitializeDE;
    CalculateCoefficients;
end;

procedure TPumpQEC.Store ( var Stream: TFileRecordStream);
begin
    inherited Store(Stream);
    Stream.Write(PumpDE, sizeof(PumpDE));
end;

constructor TPumpQEC.Load ( var Stream: TFileRecordStream);
begin
    inherited Load(Stream);
    Stream.Read(PumpDE, sizeof(PumpDE));
end;

function TPumpQEC.CreateQECInterface( aElement : pointer): TForm;
begin
    Result := TPumpQECFrm.Create(Application, self);
end;

procedure TPumpQEC.InitializeDE;
begin
    with PumpDE do
        begin
            ctWATER_IN := nil;
            ctWATER_OUT := nil;
            ctCONTROL_MASSFLOW := nil;
        end;
end;

procedure TPumpQEC.HandleRecord;
var
    QECFile : TIniFile;
begin
    QECFile := TIniFile.Create(m_FileName);
    PumpDE.PUMPCONTROL := GetControl(QECFile.ReadString('SPECIFICATIONS','CONFIGURATION','fcSetFlow'));

```

```

PumpDE.FLOWCONFIGURATION := GetFlowConfiguration(QECFile.ReadString('FLOW
CONFIGURATION','CONFIGURATION',fcSetFlow));
PumpDE.ROTORDIAMETER := QECFile.ReadFloat('SPECIFICATIONS','ROTOR DIAMETER',0.0);
PumpDE.DRIVEMOTOREFFICIENCY := QECFile.ReadFloat('SPECIFICATIONS','DRIVE MOTOR EFFICIENCY',0.0);
PumpDE.ROTATIONSPEED := QECFile.ReadFloat('SPECIFICATIONS','ROTATION SPEED',0.0);
PumpDE.MASSFLOW := QECFile.ReadFloat('SPECIFICATIONS','MASSFLOW',0.0);
PumpDE.WATERHEIGHT := QECFile.ReadFloat('VSD CONTROL','WATERHEIGHT',0.0);
PumpDE.TOTALH := QECFile.ReadFloat('VSD CONTROL','TOTALH',0.0);
PumpDE.VSDFLOW := QECFile.ReadFloat('VSD CONTROL','VSDFLOW',0.0);
PumpDE.WORKINGPOINT := GetWorkingPointMethod(QECFile.ReadString('WORKING POINTS','METHOD','wpSingle'));
PumpDE.QPOINTS[1] := QECFile.ReadFloat('WORKING POINTS','Q1',0.0);
PumpDE.QPOINTS[2] := QECFile.ReadFloat('WORKING POINTS','Q2',0.0);
PumpDE.QPOINTS[3] := QECFile.ReadFloat('WORKING POINTS','Q3',0.0);
PumpDE.HPOINTS[1] := QECFile.ReadFloat('WORKING POINTS','H1',0.0);
PumpDE.HPOINTS[2] := QECFile.ReadFloat('WORKING POINTS','H2',0.0);
PumpDE.HPOINTS[3] := QECFile.ReadFloat('WORKING POINTS','H3',0.0);
PumpDE.EPOINTS[1] := QECFile.ReadFloat('WORKING POINTS','E1',0.0);
PumpDE.EPOINTS[2] := QECFile.ReadFloat('WORKING POINTS','E2',0.0);
PumpDE.EPOINTS[3] := QECFile.ReadFloat('WORKING POINTS','E3',0.0);
QECFile.Free;
end;

procedure TPumpQEC.ExportRecord(Filename : String);
var
  QECFile : TIniFile;
begin
  inherited ExportRecord(Filename);
  QECFile := TIniFile.Create(Filename);
  QECFile.WriteString('SPECIFICATIONS','CONFIGURATION', SetControl(PumpDE.PUMPCONTROL));
  QECFile.WriteString('FLOW
CONFIGURATION','CONFIGURATION',SetFlowConfiguration(PumpDE.FLOWCONFIGURATION));
  QECFile.WriteFloat('SPECIFICATIONS','ROTOR DIAMETER',PumpDE.ROTORDIAMETER);
  QECFile.WriteFloat('SPECIFICATIONS','DRIVE MOTOR EFFICIENCY',PumpDE.DRIVEMOTOREFFICIENCY);
  QECFile.WriteFloat('SPECIFICATIONS','ROTATION SPEED',PumpDE.ROTATIONSPEED);
  QECFile.WriteFloat('SPECIFICATIONS','MASSFLOW',PumpDE.MASSFLOW);
  QECFile.WriteFloat('VSD CONTROL','WATERHEIGHT',PumpDE.WATERHEIGHT);
  QECFile.WriteFloat('VSD CONTROL','TOTALH',PumpDE.TOTALH);
  QECFile.WriteFloat('VSD CONTROL','VSDFLOW',PumpDE.VSDFLOW);
  QECFile.WriteString('WORKING POINTS','METHOD',SetWorkingPointMethod(PumpDE.WORKINGPOINT));
  QECFile.WriteFloat('WORKING POINTS','Q1',PumpDE.QPOINTS[1]);
  QECFile.WriteFloat('WORKING POINTS','Q2',PumpDE.QPOINTS[2]);
  QECFile.WriteFloat('WORKING POINTS','Q3',PumpDE.QPOINTS[3]);
  QECFile.WriteFloat('WORKING POINTS','H1',PumpDE.HPOINTS[1]);
  QECFile.WriteFloat('WORKING POINTS','H2',PumpDE.HPOINTS[2]);
  QECFile.WriteFloat('WORKING POINTS','H3',PumpDE.HPOINTS[3]);
  QECFile.WriteFloat('WORKING POINTS','E1',PumpDE.EPOINTS[1]);
  QECFile.WriteFloat('WORKING POINTS','E2',PumpDE.EPOINTS[2]);
  QECFile.WriteFloat('WORKING POINTS','E3',PumpDE.EPOINTS[3]);
  QECFile.Free;
end;
(*****
{ See base class for comments }
*****)
procedure TPumpQEC.AssignGenericLinks;
begin
  with PumpDE do
    begin
      Link_In := ctWATER_IN;
      Link_Out := ctWATER_OUT;
    end;
end;

procedure TPumpQEC.MakeConnection( aConnection : Pointer;
aPort : Byte);
begin
  with PumpDE do
    begin
      case aPort of
        ptIN_WATER : ctWATER_IN := aConnection;
        ptIN_CONTROL_MASSFLOW : ctCONTROL_MASSFLOW := aConnection;
        ptOUT_WATER : ctWATER_OUT := aConnection;
      end;
    end;
  AssignGenericLinks;
end;

```

```

procedure TPumpQEC.BreakConnection(aPort : Byte;
                                   aPortType : PortType);
begin
  with PumpDE do
  begin
    case aPort of
      ptIN_WATER : ctWATER_IN := nil;
      ptIN_CONTROL_MASSFLOW: ctCONTROL_MASSFLOW := nil;
      ptOUT_WATER: ctWATER_OUT := nil;
    end;
    end;
    AssignGenericLinks;
  end;

procedure TPumpQEC.GetOpenPortConnectionTypes( var aConnection : ConnectionSet;
                                                aPortType : PortType);
begin
  aConnection := [];
  with PumpDE do
  begin
    case aPortType of
      ptIN: begin
        if (ctWATER_IN = nil) then
          aConnection := aConnection + [ctWATER];
        if (ctCONTROL_MASSFLOW = nil) then
          aConnection := aConnection + [ctCONTROL];
        end;
      ptOUT: begin
        if (ctWATER_OUT = nil) then
          aConnection := aConnection + [ctWATER];
        end;
      end;
    end;
  end;
end;

procedure TPumpQEC.GetOpenPorts( var Ports : PortSet;
                                 aPortType : PortType;
                                 aConnection : ConnectionSet);
begin
  Ports := [];
  with PumpDE do
  begin
    if (ctWATER in aConnection) then
    begin
      case aPortType of
        ptIN: begin
          if (ctWATER_IN = nil) then
            Ports := Ports + [ptIN_WATER];
          end;
        ptOUT: begin
          if (ctWATER_OUT = nil) then
            Ports := Ports + [ptOUT_WATER];
          end;
        end;
      end;
    end;
    if (ctCONTROL in aConnection) then
    begin
      Ports := Ports + [ptIN_CONTROL_MASSFLOW];
    end;
  end;
end;

function TPumpQEC.GetPortName( aPort : Byte) : String;
begin
  case aPort of
    ptIN_WATER : Result := 'ptIN WATER';
    ptIN_CONTROL_MASSFLOW : Result := 'ptIN CONTROL MASSFLOW';
    ptOUT_WATER : Result := 'ptOUT WATER';
  end;
end;

function TPumpQEC.ElementStatus : EStatus;
begin
  with PumpDE do
  begin
    if (ctWATER_IN <> nil) and (ctWATER_OUT <> nil) then

```

```

        Result := esDefined
    else
        Result := esUndefined;
    end;
end;

procedure TPumpQEC.SetControlableValue( aPort : Byte;
                                       aValue: EnergyFloat);
begin
    with PumpDE do
        begin
            if aPort = ptIN_CONTROL_MASSFLOW then
                begin
                    MASSFLOW := aValue;
                end;
            end;
        end;
end;

function TPumpQEC.GetControlableValue( aPort: Byte; aTime: TimeType):EnergyFloat;
begin
    result := 0;
    with PumpDE do
        begin
            if aPort = ptIN_CONTROL_MASSFLOW then
                begin
                    result := ctWATER_OUT.GetMassFlow(aTime);
                end;
            end;
        end;
end;

function TPumpQEC.DoInternalMassflow(const aTime : TimeType; aFlag : FFtype): Boolean;
begin
    with PumpDE do
        begin
            if (aFlag = ffBoth) and (FLOWCONFIGURATION = fcSetFlow) then
                begin
                    ctWATER_IN.SetMassFlow(aTime, MASSFLOW);
                    ctWATER_OUT.SetMassFlow(aTime, MASSFLOW);
                end;
            if (aFlag = ffForward) and (FLOWCONFIGURATION = fcSetFlow) then
                begin
                    ctWATER_OUT.SetMassFlow(aTime, MASSFLOW);
                end
            else
                begin
                    if (aFlag = ffForward) then
                        ctWATER_OUT.SetMassFlow(aTime, ctWATER_IN.GetMassFlow( aTime ));
                    end;
                    if (aFlag = ffBackward) and (FLOWCONFIGURATION = fcSetFlow) then
                        begin
                            ctWATER_IN.SetMassFlow(aTime, MASSFLOW);
                        end
                    else
                        begin
                            if (aFlag = ffBackward) then
                                ctWATER_IN.SetMassFlow(aTime, ctWATER_OUT.GetMassFlow( aTime ));
                            end;
                            if aFlag = ffSpecial then
                                begin
                                    ctWATER_OUT.SetMassFlow(aTime, 0);
                                    ctWATER_IN.SetMassFlow(aTime, 0);
                                end;
                            end;
                        end;
                end;
        end;
    end;
    Result := TRUE;
end;

procedure TPumpQEC.CalculateCoefficients;
var
    ah, bh      : EnergyFloat;
    an, bn      : EnergyFloat;
    kh1, kh2, kh3 : EnergyFloat;
    kf1, kf2, kf3 : EnergyFloat;
    ch, cn      : EnergyFloat;
    q2, q3      : EnergyFloat;

```

```

h2,h3      : EnergyFloat;
e2,e3      : EnergyFloat;
begin
ah := 0;
bh := 0;
an := 0;
bn := 0;
kh1 := 0;
kh2 := 0;
kh3 := 0;
kf1 := 0;
kf2 := 0;
kf3 := 0;
with PumpDE do
begin
{Checks if Multiple working points can be used.}
if WORKINGPOINT = wpMultiple then
begin
{Check if 3 points on Q are different. If not change wpMultiple}
{working points into single (wpSingle).}
if (QPOINTS[1] = QPOINTS[2]) or (QPOINTS[2] = QPOINTS[3]) or (QPOINTS[1] = QPOINTS[3]) then
begin
WORKINGPOINT := wpSingle;
end
else
begin
{The following calculations are used for both the single and multiple point regressions.}
kh1 := (pl*g*HPOINTS[1])/(pl*sqr(ROTATIONSPEED)*sqr(ROTORDIAMETER));
kf1 := QPOINTS[1]/(pl*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
kh2 := (pl*g*HPOINTS[2])/(pl*sqr(ROTATIONSPEED)*sqr(ROTORDIAMETER));
kf2 := QPOINTS[2]/(pl*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
kh3 := (pl*g*HPOINTS[3])/(pl*sqr(ROTATIONSPEED)*sqr(ROTORDIAMETER));
kf3 := QPOINTS[3]/(pl*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
BCOEFFICIENT[3] := (((EPOINTS[3]/100)-(EPOINTS[2]/100))*(kf1-kf2)-(kf3-kf2)*((EPOINTS[1]/100)-
(EPOINTS[2]/100)))/
((sqr(kf3)-sqr(kf2))*(kf1-kf2)-(kf3-kf2)*(sqr(kf1)-sqr(kf2)));
BCOEFFICIENT[2] := (((EPOINTS[1]/100)-(EPOINTS[2]/100)-(sqr(kf1)-sqr(kf2))*BCOEFFICIENT[3])/(kf1-kf2);
BCOEFFICIENT[1] := (EPOINTS[2]/100) -(kf2*BCOEFFICIENT[2])-(sqr(kf2)*BCOEFFICIENT[3]);
end;
end;
{If Single working point selected, 2 other points must be calculated.}
if WORKINGPOINT = wpSingle then
begin
{Calculates curvefit coefficients for several ranges of flow [kg/s].}
if (QPOINTS[1] > 0) and (QPOINTS[1] <= 3) then
begin
{Curvefit coefficients for range}
ah := -0.63125;
bh := 1.18125;
an := -0.06188;
bn := 0.33625;
end;
if (QPOINTS[1] > 3) and (QPOINTS[1] <= 7.5) then
begin
{Curvefit coefficients for range}
ah := -0.25914;
bh := 0.591667;
an := -0.01816;
bn := 0.190308;
end;
if (QPOINTS[1] > 7.5) and (QPOINTS[1] <= 15) then
begin
{Curvefit coefficients for range}
ah := -0.3248;
bh := 3.736;
an := -0.00648;
bn := 0.133;
end;
if (QPOINTS[1] > 15) and (QPOINTS[1] <= 20) then
begin
{Curvefit coefficients for range}
ah := -0.1;
bh := 2.197;
an := -0.00507;
bn := 0.159;
end;
end;
end;

```

```

if (QPOINTS[1] > 20) and (QPOINTS[1] <= 40) then
begin
  {Curvefit coefficients for range}
  ah := -0.02674;
  bh := 0.639;
  an := -0.00089;
  bn := 0.05089;
end;
if (QPOINTS[1] > 40) and (QPOINTS[1] <= 80) then
begin
  {Curvefit coefficients for range}
  ah := -0.01505;
  bh := 0.88875;
  an := -0.00028;
  bn := 0.029263;
end;
if (QPOINTS[1] > 80) and (QPOINTS[1] <= 150) then
begin
  {Curvefit coefficients for range}
  ah := -0.00164;
  bh := 0.262;
  an := -0.000047;
  bn := 0.010467;
end;
if (QPOINTS[1] > 150) then
begin
  {Curvefit coefficients for range}
  ah := -0.0002;
  bh := 0;
  an := -0.00002;
  bn := 0.0071;
end;
{Curve fitted coefficients.}
ch := HPOINTS[1]-(ah*(sqrt(QPOINTS[1]))+bh*QPOINTS[1]);
cn := (EPOINTS[1]/100)-(an*(sqrt(QPOINTS[1]))+bn*QPOINTS[1]);
{The extra pressure,flow and efficiency points are now calculated}
{Flow range of +/- 20% of single point.}
q2 := 0.8*QPOINTS[1];
q3 := 1.2*QPOINTS[1];
{Pressure head}
h2 := ah*(sqrt(q2))+bh*(q2)+ch;
{Test if the new points are viable if not the are given the limit value.}
{The pressure head is limited to be between 0 and 200 meters.}
{The efficiency is limited to be between 0.3 and 1}
if h2 < 0 then
  h2 := 0;
if h2 > 200 then
  h2 := 200;
e2 := an*(sqrt(q2))+bn*(q2)+cn;
if e2 < 0.3 then
  e2 := 0.3;
if e2 > 1 then
  e2 := 1;
h3 := ah*(sqrt(q3))+bh*(q3)+ch;
if h3 < 0 then
  h3 := 0;
if h3 > 200 then
  h3 := 200;
e3 := an*(sqrt(q3))+bn*(q3)+cn;
if e3 < 0.3 then
  e3 := 0.3;
if e3 > 1 then
  e3 := 1;
{The following calculations are used for both the single and multiple point regressions.}
kh1 := (pi*g*HPOINTS[1])/(pi*sqrt(ROTATIONSPEED)*sqrt(ROTORDIAMETER));
kf1 := QPOINTS[1]/(pi*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
kh2 := (pi*g*h2)/(pi*sqrt(ROTATIONSPEED)*sqrt(ROTORDIAMETER));
kf2 := q2/(pi*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
kh3 := (pi*g*h3)/(pi*sqrt(ROTATIONSPEED)*sqrt(ROTORDIAMETER));
kf3 := q3/(pi*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
BCOEFFICIENT[3] := ((e3-e2)*(kf1-kf2)-(kf3-kf2)*((EPOINTS[1]/100)-e2)/
((sqrt(kf3)-sqrt(kf2))*(kf1-kf2)-(kf3-kf2)*(sqrt(kf1)-sqrt(kf2))));
BCOEFFICIENT[2] := (((EPOINTS[1]/100)-e2)-(sqrt(kf1)-sqrt(kf2))*BCOEFFICIENT[3])/(kf1-kf2);
BCOEFFICIENT[1] := e2 -(kf2*BCOEFFICIENT[2])-(sqrt(kf2))*BCOEFFICIENT[3];
end;
{The correlation coefficients are calculated next}

```

```

ACOEFFICIENT[3] := ((kh3-kh2)*(kf1-kf2)-(kf3-kf2)*(kh1-kh2))/
((sqr(kf3)-sqr(kf2))*(kf1-kf2)-(kf3-kf2)*(sqr(kf1)-sqr(kf2)));
ACOEFFICIENT[2] := ((kh1-kh2)-(sqr(kf1)-sqr(kf2)))*ACOEFFICIENT[3]/(kf1-kf2);
ACOEFFICIENT[1] := kh2-(kf2*ACOEFFICIENT[2])-(sqr(kf2)*ACOEFFICIENT[3]);
end;
end;

function TPumpQEC.DoSimulationInitialize (const aTime : TimeType): Boolean;
begin
  tle0 := InitialWaterConnectionT;
  Result := TRUE;
end;

function TPumpQEC.DoInternalANDOutletState(const aTime : TimeType): Boolean;
var
  mli, Te      : EnergyFloat;
  tli, tle     : EnergyFloat;
  hc, Area     : EnergyFloat;
  kf, kh, npump : EnergyFloat;
  ptotal       : EnergyFloat;
  Qi           : EnergyFloat;
  cpl          : EnergyFloat;
  A, B, n1, n2 : EnergyFloat;
  pwr          : EnergyFloat;
begin
  ptotal := 0;
  {The following values will be taken from the inlet connection}
  with PumpDE do
  begin
    mli := ctWATER_IN.GetMassFlow(aTime);
    tli := ctWATER_IN.GetTemperature(aTime);
    cpl := WaterSpecificCapacityT(tli);
    Te := 20; {The temperature that the fluid in the component }
              {will strive towards Either 20 if the component is indoors or the }
              {climate temperature if it is outdoors. }
    hc := 15; {Convection coefficient [W/m^2]}
    Area := 0.3; {Heat transfer area}
    {Control strategies}
    if mli = 0 then {Check for when the massflow = 0}
    begin
      pwr := 0;
      tle := (tle0-Te)*exp(-(hc*Area)/(pl*0.01*cpl))*deltaT)+Te;
      if tle < 0 then {Check if temperature is within limits}
        tle := 0;
      if tle > 100 then
        tle := 100;
      SetElementPower(aTime, pwr);
      ctWATER_OUT.SetTemperature(aTime, tle);
    end
  else
  begin
    if mli < 0 then
    begin
      if PUMPCONTROL = pcVSDStaticPressure then {VSD static pressure control}
      begin
        ptotal := WATERHEIGHT*pl*G;
      end;
      if PUMPCONTROL = pcVSDTemperature then {VSD return temperature control}
      begin
        ptotal := ((TOTALH*G*pl)/(sqr(VSDFLOW/pl)))*sqr(mli/pl);
      end;
      if (PUMPCONTROL = pcVSDStaticPressure) or (PUMPCONTROL = pcVSDTemperature) then
      begin
        A := ptotal/(pl*sqr(ROTORDIAMETER));
        B := mli/(pl*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
        n1 := (-ACOEFFICIENT[2]*B+sqrt(sqr(ACOEFFICIENT[2]*B)-
4*ACOEFFICIENT[1]*(ACOEFFICIENT[3]*B*B-A)))/(2*ACOEFFICIENT[1]);
        n2 := (-ACOEFFICIENT[2]*B-sqrt(sqr(ACOEFFICIENT[2]*B)-
4*ACOEFFICIENT[1]*(ACOEFFICIENT[3]*B*B-A)))/(2*ACOEFFICIENT[1]);
        if (n2 < 0) and (n2 > 2900) then
          n2 := 1450;
        if (n1 > 0) and (n1 <= 2900) then
          ROTATIONSPEED := n1
        else
          ROTATIONSPEED := n2;
        end;
      end;
    end;
  end;
end;

```

```

end;
kf := mli/(pi*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER*ROTORDIAMETER);
if kf < 0.000001 then
  kf := 0.000001;
if kf > 0.1 then
  kf := 0.1;
kh := ACOEFFICIENT[1]+ACOEFFICIENT[2]*kf+ACOEFFICIENT[3]*sqr(kf);
if kh < 0.000001 then
  kh := 0.000001;
if kh > 0.1 then
  kh := 0.1;
npump := BCOEFFICIENT[1]+BCOEFFICIENT[2]*kf+BCOEFFICIENT[3]*sqr(kf);
if npump < 0.3 then
  npump := 0.3;
if npump > 1 then
  npump := 1;
ptotal := kh*pi*ROTATIONSPEED*ROTATIONSPEED*ROTORDIAMETER*ROTORDIAMETER;
Ql := ((1-npump)*mli*ptotal)/(pi*npump);
tle := tli+(Ql/(mli*cpl));
if tle < tli then
  tle := tli;
if tle > 100 then
  tle := 100;
pwr := ((ptotal*mli)/(pi*(DRIVEMOTOREFFICIENCY/100)*npump));
SetElementPower(aTime, pwr);
ctWATER_OUT.SetTemperature(aTime, tle);
end;
if sTimeStepEnded in aTime.Flag then
begin
  tle0 := tle;
end;
end;
Result := TRUE;
end;

(*****)
{See base class for comments}
(*****)
function TPumpQEC.GetNominalHeatingCooling(InletConnection: TConnectionQEC):EnergyFloat;
var
  maxflow, {kg/s}
  maxpressure, {m}
  minefficiency : EnergyFloat; {%-}
  i : Integer;
begin
  with PumpDE do
  begin
    maxflow := 0;
    maxpressure := 0;
    minefficiency := 1;
    for i := 1 to 3 do
    begin
      if QPOINTS[i] > maxflow then
      begin
        maxflow := QPOINTS[i];
      end;
      if HPOINTS[i] > maxpressure then
      begin
        maxpressure := HPOINTS[i];
      end;
      if EPOINTS[i] < minefficiency then
      begin
        minefficiency := EPOINTS[i];
      end;
    end;
    result := ((maxflow/pi)*maxpressure*G*pi*(1-minefficiency/100))/1000;
  end;
end;

procedure TPumpQEC.Report(const aTime : TimeType; var Stream : TFileRecordStream);
var
  Output : OutputDE;
begin
  Output.Element := m_ELEMENT;
  with PumpDE do
  begin

```

```

        Output.M1 := ctWATER_OUT.GetMassFlow(aTime);
        Output.Tout1 := ctWATER_OUT.GetTemperature(aTime);
        Output.Power := GetElementPower(aTime);
        Stream.Write(Output, sizeof(OutputDE));
    end;
end;
end.

```

SYSTEM MASS FLOW SIMULATOR CODE EXTRACT

```

unit MassflowQEC;

interface
uses
    Globals,
    Windows,
    Classes,
    ChildWin,
    Sysutils,
    ElementQEC,
    WaterConvergeQEC,
    AirConvergeQEC,
    WaterDivergeQEC,
    AirDivergeQEC,
    ValveQEC,
    ManualValveQEC,
    PumpQEC,
    DamperQEC,
    DamQEC,
    FanQEC,
    WaterCooledChillerQEC,
    AirCooledChillerQEC,
    CoolingTowerQEC,
    BulkAirCoolerQEC,
    CoilQEC,
    WaterToWaterExchangerQEC,
    AirToAirExchangerQEC,
    StorageTankQEC;

type
    TMassflowQEC = class
    public
        {public declarations}
        m_MASSFLOWSEQUENCE : TList;
        m_MASSFLOWITERATIONS: Integer;

        constructor Create;
        procedure UpdateElementConnections;
        procedure SortMassFlowSequence; virtual;
        procedure BuildMassFlowSequence; virtual;
        procedure ReportMassFlowSequence; virtual;
        procedure SolveMassflow (aTime : TimeType);virtual;
        procedure SolveSystemFlow ( aTime : TimeType);
        procedure SolveSystemSpecialFlowCase1( aTime : TimeType);
        procedure SolveSystemSpecialFlowCase2( aTime : TimeType);
        function ValidateMassflow ( aTime : TimeType): Boolean;
        destructor Destroy; override;
        {Helper functions: WATER}
        function isWATERCONVERGETVALVE (aElement: TElementQEC): Boolean;
        function isWATERCONVERGETPIECE (aElement: TElementQEC): Boolean;
        function isWATERDIVERGETVALVE (aElement: TElementQEC): Boolean;
        function isWATERDIVERGETPIECE (aElement: TElementQEC): Boolean;
        function isPUMPSETFLOW (aElement: TElementQEC): Boolean;
        function isFANSETFLOW (aElement: TElementQEC): Boolean;
        function isDAMSETFLOW (aElement : TElementQEC): Boolean;
        function isDAMSETFLOWOVERFLOW (aElement : TElementQEC): Boolean;
        function isDAMGETFLOW (aElement : TElementQEC): Boolean;
        function isAIRCONVERGETVALVE (aElement: TElementQEC): Boolean;
        function isAIRCONVERGETPIECE (aElement: TElementQEC): Boolean;
        function isAIRDIVERGETVALVE (aElement: TElementQEC): Boolean;
        function isAIRDIVERGETPIECE (aElement: TElementQEC): Boolean;
        function isMULTIPORT(aElement : TElementQEC): Boolean;
        function getMULTIPORTBackwards ( aConnection: TConnectionQEC) : TElementQEC;
    end;

```

```

function getMULTIPOINTForwards ( aConnection: TConnectionQEC) : TElementQEC;
function getMULTIPOINTForwardsSystemSpecialFlow ( aConnection : TConnectionQEC) : TElementQEC;
private
{private declarations}
procedure SearchBackwardsSingle(var aElement : TElementQEC);
procedure SearchBackwardsDouble(var aElement : TElementQEC);
procedure SearchForwardsSingle (var aElement : TElementQEC);
procedure SearchForwardsDouble (var aElement : TElementQEC);
procedure SearchForwardsSingleSystemSpecialFlowCase1(const aTime : TimeType; var aElement : TElementQEC);
procedure SearchForwardsDoubleSystemSpecialFlowCase1(const aTime : TimeType; var aElement : TElementQEC);
procedure SearchForwardsSingleSystemSpecialFlowCase2(const aTime : TimeType; var aElement : TElementQEC);
procedure SearchForwardsDoubleSystemSpecialFlowCase2(const aTime : TimeType; var aElement : TElementQEC);

end;

implementation

uses
    Main;

var
    QECElement      : TElementQEC;
    QECCConnection   : TConnectionQEC;

{*****}
{ * NAME: Create                                     * }
{ * DESC: Creates MASSFLOW and SEQUENCE list.      * }
{*****}
constructor TMassflowQEC.Create;
begin
    m_MASSFLOWSEQUENCE := TList.Create;
    m_MASSFLOWITERATIONS := 1;
    SortMassFlowSequence;
    BuildMassFlowSequence;
    MainForm.SimulationStatus.Output.Lines.Text := 'Massflow Initialization OK';
    if MainForm.ProgramSettings.RMCheckBox.Checked = TRUE then
        begin
            ReportMassFlowSequence;
        end;
end;

{*****}
{ * NAME: isWATERCONVERGETVALVE and isAIRCONVERGETVALVE * }
{ * DESC: Boolean function returning true if TVALVE flow * }
{ * valve component                                     * }
{*****}
function TMassflowQEC.isWATERCONVERGETVALVE (aElement: TElementQEC): Boolean;
begin
    if (aElement.m_Type = 'CONVERGE:WATER') then
        begin
            if (TWaterConvergeQEC(aElement).WaterConvergeDE.CONVERGETYPE = etTVAlve) then
                begin
                    Result := TRUE;
                end
            else
                Result := FALSE;
            end
        end
    else
        Result := FALSE;
end;

function TMassflowQEC.isAIRCONVERGETVALVE (aElement : TElementQEC): Boolean;
begin
    if (aElement.m_Type = 'CONVERGE:AIR') then
        begin
            if (TAirConvergeQEC(aElement).AirConvergeDE.CONVERGETYPE = etTVAlve) then
                begin
                    Result := TRUE;
                end
            else
                Result := FALSE;
            end
        end
    else
        Result := FALSE;
end;
end;

```

```

{*****}
{* NAME: isWATERCONVERGETPIECE and isAIRCONVERGETPIECE *}
{* DESC: Boolean function returning true if TPIECE flow *}
{* valve component. *}
{*****}
function TMassflowQEC.isWATERCONVERGETPIECE (aElement: TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'CONVERGE:WATER') then
    begin
      if (TWaterConvergeQEC(aElement).WaterConvergeDE.CONVERGETYPE = etTPiece) then
        begin
          Result := TRUE;
        end
      else
        Result := FALSE;
      end
    end
  else
    Result := FALSE;
  end;
end;

function TMassflowQEC.isAIRCONVERGETPIECE (aElement : TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'CONVERGE:AIR') then
    begin
      if (TAirConvergeQEC(aElement).AirConvergeDE.CONVERGETYPE = etTPiece) then
        begin
          Result := TRUE;
        end
      else
        Result := FALSE;
      end
    end
  else
    Result := FALSE;
  end;
end;
{*****}
{* NAME: isWATERDIVERGETVALVE and isAIRDIVERGETVALVE *}
{* DESC: Boolean function returning true if TVALVE flow *}
{* valve component. *}
{*****}
function TMassflowQEC.isWATERDIVERGETVALVE (aElement: TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'DIVERGE:WATER') then
    begin
      if (TWaterDivergeQEC(aElement).WaterDivergeDE.DIVERGETYPE = etTValve) then
        begin
          Result := TRUE;
        end
      else
        Result := FALSE;
      end
    end
  else
    Result := FALSE;
  end;
end;

function TMassflowQEC.isAIRDIVERGETVALVE (aElement: TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'DIVERGE:AIR') then
    begin
      if (TAirDivergeQEC(aElement).AirDivergeDE.DIVERGETYPE = etTValve) then
        begin
          Result := TRUE;
        end
      else
        Result := FALSE;
      end
    end
  else
    Result := FALSE;
  end;
end;
{*****}
{* NAME: isWATERDIVERGETPIECE and isAIRDIVERGETPIECE *}
{* DESC: Boolean function returning true if TPIECE flow *}
{* valve component. *}
{*****}
function TMassflowQEC.isWATERDIVERGETPIECE (aElement: TElementQEC): Boolean;
begin

```

```

if (aElement.m_Type = 'DIVERGE:WATER') then
begin
  if (TWaterDivergeQEC(aElement).WaterDivergeDE.DIVERGETYPE = etTPiece) then
  begin
    Result := TRUE;
  end
  else
    Result := FALSE;
  end
else
  Result := FALSE;
end;

function TMassflowQEC.isAIRDIVERGETPIECE (aElement: TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'DIVERGE:AIR') then
  begin
    if (TAirDivergeQEC(aElement).AirDivergeDE.DIVERGETYPE = etTPiece) then
    begin
      Result := TRUE;
    end
    else
      Result := FALSE;
    end
  else
    Result := FALSE;
  end;
end;

{*****}
{* NAME: isDAMSETFLOW *}
{* DESC: Boolean function returning true if fcSetFlow flow *}
{*****}
function TMassflowQEC.isDAMSETFLOW( aElement : TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'DAM') then
  begin
    if (TDamQEC(aElement).DamDE.FLOWCONFIGURATION = fcSetFlow) then
    begin
      Result := TRUE;
    end
    else
      Result := FALSE;
    end
  else
    Result := FALSE;
  end;
end;

{*****}
{* NAME: isDAMGETFLOW *}
{* DESC: Boolean function returning true if fcGetFlow flow *}
{*****}
function TMassflowQEC.isDAMGETFLOW( aElement : TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'DAM') then
  begin
    if (TDamQEC(aElement).DamDE.FLOWCONFIGURATION = fcGetFlow) then
    begin
      Result := TRUE;
    end
    else
      Result := FALSE;
    end
  else
    Result := FALSE;
  end;
end;

{*****}
{* NAME: isDAMSETOVERFLOWFLOW *}
{* DESC: Boolean function returning true if fcSetOverFlow flow*}
{*****}
function TMassflowQEC.isDAMSETFLOWOVERFLOW( aElement : TElementQEC): Boolean;
begin
  if (aElement.m_Type = 'DAM') then
  begin
    if (TDamQEC(aElement).DamDE.FLOWCONFIGURATION = fcSetFlowOverflow) then
    begin

```

```

        Result := TRUE;
    end
    else
        Result := FALSE;
    end
    else
        Result := FALSE;
    end;

{*****}
{* NAME: isPUMPSETFLOW *}
{* DESC: Boolean function returning true if fcSetFlow flow *}
{*****}
function TMassflowQEC.isPUMPSETFLOW( aElement : TElementQEC): Boolean;
begin
    if (aElement.m_Type = 'PUMP') then
    begin
        if (TPumpQEC(aElement).PumpDE.FLOWCONFIGURATION = fcSetFlow) then
        begin
            Result := TRUE;
        end
        else
            Result := FALSE;
        end
    end
    else
        Result := FALSE;
    end;

{*****}
{* NAME: isFANSETFLOW *}
{* DESC: Boolean function returning true if fcSetFlow flow *}
{*****}
function TMassflowQEC.isFANSETFLOW( aElement : TElementQEC): Boolean;
begin
    if (aElement.m_Type = 'FAN') then
    begin
        if (TFanQEC(aElement).FanDE.FLOWCONFIGURATION = fcSetFlow) then
        begin
            Result := TRUE;
        end
        else
            Result := FALSE;
        end
    end
    else
        Result := FALSE;
    end;

{*****}
{* NAME: isMULTIPOINT *}
{* DESC: Boolean function returning true if fcSetFlow flow *}
{*****}
function TMassflowQEC.isMULTIPOINT( aElement : TElementQEC): Boolean;
begin
    if (aElement.m_Type = 'CHILLER:WATER') then
    begin
        Result := TRUE;
    end
    else
        if (aElement.m_Type = 'STORAGETANK') then
        begin
            Result := TRUE;
        end
        else
            if (aElement.m_Type = 'EXCHANGER:WATERTOWATER') then
            begin
                Result := TRUE;
            end
            else
                if (aElement.m_Type = 'EXCHANGER:AIRTOAIR') then
                begin
                    Result := TRUE;
                end
                else
                    if (aElement.m_Type = 'CHILLER:AIR') then

```

```

begin
  if (TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_IN <> nil) and
    (TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_OUT <> nil) then
  begin
    Result := TRUE;
  end
  else
    Result := FALSE;
  end
else
  if (aElement.m_Type = 'COOLINGTOWER') then
  begin
    if (TCoolingTowerQEC(aElement).CoolingTowerDE.ctAIR_IN <> nil) and
      (TCoolingTowerQEC(aElement).CoolingTowerDE.ctAIR_OUT <> nil) then
    begin
      Result := TRUE;
    end
    else
      Result := FALSE;
    end
  else
    if (aElement.m_Type = 'BULKAIRCOOLER') then
    begin
      if (TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_IN <> nil) and
        (TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_OUT <> nil) then
      begin
        Result := TRUE;
      end
      else
        Result := FALSE;
      end
    end
  else
    if (aElement.m_Type = 'COIL') then
    begin
      Result := TRUE;
    end
    else
      Result := FALSE;
    end;
end;

{*****}
{* NAME: getMULTIPORTBackwards *}
{* DESC: Returns the correct multiport towards which to move *}
{*****}
function TMassflowQEC.getMULTIPORTBackwards (aConnection: TConnectionQEC) : TElementQEC;

type ConnectionRecord = record
  ConnectionIN_1 : Integer;
  ConnectionIN_2 : Integer;
  ConnectionOUT_1 : Integer;
  ConnectionOUT_2 : Integer;
  Connection : Integer;
end;

var
  Connection : ConnectionRecord;
  aElement : TElementQEC;
begin
  with MainForm do
  begin
    aElement := aConnection.GetSTART;
    if (aElement.m_Type = 'CHILLER:WATER') then
    begin
      Connection.ConnectionIN_1 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctEVAPORATORWATER_IN.m_CONNECTION;
      Connection.ConnectionIN_2 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctCONDENSORWATER_IN.m_CONNECTION;
      Connection.ConnectionOUT_1 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctEVAPORATORWATER_OUT.m_CONNECTION;
      Connection.ConnectionOUT_2 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctCONDENSORWATER_OUT.m_CONNECTION;
    end;
    if (aElement.m_Type = 'STORAGETANK') then
    begin
      Connection.ConnectionIN_1 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_IN1.m_CONNECTION;
      Connection.ConnectionIN_2 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_IN2.m_CONNECTION;
    end;
  end;
end;

```

```

    Connection.ConnectionOUT_1 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_OUT1.m_CONNECTION;
    Connection.ConnectionOUT_2 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_OUT1.m_CONNECTION;
end;
if (aElement.m_Type = 'EXCHANGER:WATERTOWATER') then
begin
    Connection.ConnectionIN_1 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_IN1.m_CONNECTION;
    Connection.ConnectionIN_2 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_IN2.m_CONNECTION;
    Connection.ConnectionOUT_1 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_OUT1.m_CONNECTION;
    Connection.ConnectionOUT_2 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_OUT2.m_CONNECTION;
end;
if (aElement.m_Type = 'EXCHANGER:AIRTOAIR') then
begin
    Connection.ConnectionIN_1 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_IN1.m_CONNECTION;
    Connection.ConnectionIN_2 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_IN2.m_CONNECTION;
    Connection.ConnectionOUT_1 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_OUT1.m_CONNECTION;
    Connection.ConnectionOUT_2 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_OUT2.m_CONNECTION;
end;
if (aElement.m_Type = 'CHILLER:AIR') then
begin
    Connection.ConnectionIN_1 := TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctEVAPORATORWATER_IN.m_CONNECTION;
    Connection.ConnectionIN_2 := TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 := TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctEVAPORATORWATER_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 := TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'COOLINGTOWER') then
begin
    Connection.ConnectionIN_1 := TCoilingTowerQEC(aElement).CoolingTowerDE.ctWATER_IN.m_CONNECTION;
    Connection.ConnectionIN_2 := TCoilingTowerQEC(aElement).CoolingTowerDE.ctAIR_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 := TCoilingTowerQEC(aElement).CoolingTowerDE.ctWATER_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 := TCoilingTowerQEC(aElement).CoolingTowerDE.ctAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'BULKAIRCOOLER') then
begin
    Connection.ConnectionIN_1 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctWATER_IN.m_CONNECTION;
    Connection.ConnectionIN_2 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctWATER_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'COIL') then
begin
    Connection.ConnectionIN_1 := TCoilQEC(aElement).CoilDE.ctAIR_IN.m_CONNECTION;
    Connection.ConnectionIN_2 := TCoilQEC(aElement).CoilDE.ctWATER_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 := TCoilQEC(aElement).CoilDE.ctAIR_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 := TCoilQEC(aElement).CoilDE.ctWATER_OUT.m_CONNECTION;
end;
if aConnection.m_CONNECTION = Connection.ConnectionOUT_1 then
    Connection.Connection := Connection.ConnectionIN_1
else
if aConnection.m_CONNECTION = Connection.ConnectionOUT_2 then
    Connection.Connection := Connection.ConnectionIN_2;
if aConnection.m_CONNECTION = Connection.ConnectionIN_1 then
    Connection.Connection := Connection.ConnectionOUT_1
else
if aConnection.m_CONNECTION = Connection.ConnectionIN_2 then
    Connection.Connection := Connection.ConnectionOUT_2;
QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Connection.Connection];
if (QECCConnection.GetSTART.m_Type <> 'SOURCE:WATER') and
(QECCConnection.GetSTART.m_Type <> 'SOURCE:AIR') and
(isWATERCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
(isAIRCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
(isWATERDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
(isAIRDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
(QECCConnection.GetSTART.m_Type <> 'VALVE') and
(QECCConnection.GetSTART.m_Type <> 'MANUALVALVE') and

```

```

(isPUMPSETFLOW(QECCConnection.GetSTART) = FALSE) and
(isFANSETFLOW(QECCConnection.GetSTART) = FALSE) and
(QECCConnection.GetSTART.m_Type <> 'DAMPER') and
(QECCConnection.GetSTART.m_Type <> 'DAM') and
(QECCConnection.GetSTART <> aElement) then
    Result := QECCConnection.GetSTART
else
    Result := nil;
end;
end;

{*****}
{* NAME: getMULTIPORTForwards *}
{* DESC: Returns the correct multiport towards which to move *}
{*****}
function TMassflowQEC.getMULTIPORTForwards (aConnection: TConnectionQEC) : TElementQEC;

type ConnectionRecord = record
    ConnectionIN_1 : Integer;
    ConnectionIN_2 : Integer;
    ConnectionOUT_1 : Integer;
    ConnectionOUT_2 : Integer;
    Connection : Integer;
end;

var
    Connection : ConnectionRecord;
    aElement : TElementQEC;
begin
    with MainForm do
        begin
            aElement := aConnection.GetEND;
            if (aElement.m_Type = 'CHILLER:WATER') then
                begin
                    Connection.ConnectionIN_1 := TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctEVAPORATORWATER_IN.m_CONNECTION;
                    Connection.ConnectionIN_2 := TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctCONDENSORWATER_IN.m_CONNECTION;
                    Connection.ConnectionOUT_1 := TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctEVAPORATORWATER_OUT.m_CONNECTION;
                    Connection.ConnectionOUT_2 := TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctCONDENSORWATER_OUT.m_CONNECTION;
                end;
            if (aElement.m_Type = 'STORAGETANK') then
                begin
                    Connection.ConnectionIN_1 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_IN1.m_CONNECTION;
                    Connection.ConnectionIN_2 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_IN2.m_CONNECTION;
                    Connection.ConnectionOUT_1 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_OUT1.m_CONNECTION;
                    Connection.ConnectionOUT_2 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_OUT2.m_CONNECTION;
                end;
            if (aElement.m_Type = 'EXCHANGER:WATER:TOWATER') then
                begin
                    Connection.ConnectionIN_1 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_IN1.m_CONNECTION;
                    Connection.ConnectionIN_2 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_IN2.m_CONNECTION;
                    Connection.ConnectionOUT_1 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_OUT1.m_CONNECTION;
                    Connection.ConnectionOUT_2 := TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_OUT2.m_CONNECTION;
                end;
            if (aElement.m_Type = 'EXCHANGER:AIR:TOAIR') then
                begin
                    Connection.ConnectionIN_1 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_IN1.m_CONNECTION;
                    Connection.ConnectionIN_2 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_IN2.m_CONNECTION;
                    Connection.ConnectionOUT_1 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_OUT1.m_CONNECTION;
                    Connection.ConnectionOUT_2 := TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_OUT2.m_CONNECTION;
                end;
            if (aElement.m_Type = 'CHILLER:AIR') then
                begin

```

```

    Connection.ConnectionIN_1 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctEVAPORATORWATER_IN.m_CONNECTION;
    Connection.ConnectionIN_2 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctEVAPORATORWATER_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'COOLINGTOWER') then
begin
    Connection.ConnectionIN_1 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctWATER_IN.m_CONNECTION;
    Connection.ConnectionIN_2 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctAIR_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctWATER_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'BULKAIRCOOLER') then
begin
    Connection.ConnectionIN_1 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctWATER_IN.m_CONNECTION;
    Connection.ConnectionIN_2 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctWATER_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'COIL') then
begin
    Connection.ConnectionIN_1 := TCoilQEC(aElement).CoilDE.ctAIR_IN.m_CONNECTION;
    Connection.ConnectionIN_2 := TCoilQEC(aElement).CoilDE.ctWATER_IN.m_CONNECTION;
    Connection.ConnectionOUT_1 := TCoilQEC(aElement).CoilDE.ctAIR_OUT.m_CONNECTION;
    Connection.ConnectionOUT_2 := TCoilQEC(aElement).CoilDE.ctWATER_OUT.m_CONNECTION;
end;
if aConnection.m_CONNECTION = Connection.ConnectionOUT_1 then
    Connection.Connection := Connection.ConnectionIN_1
else
if aConnection.m_CONNECTION = Connection.ConnectionOUT_2 then
    Connection.Connection := Connection.ConnectionIN_2;

if aConnection.m_CONNECTION = Connection.ConnectionIN_1 then
    Connection.Connection := Connection.ConnectionOUT_1
else
if aConnection.m_CONNECTION = Connection.ConnectionIN_2 then
    Connection.Connection := Connection.ConnectionOUT_2;
QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Connection.Connection];
if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
(QECCConnection.GetEND.m_Type <> 'SOURCE:AIR') and
(isWATERCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(isAIRCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(isWATERDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(isAIRDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(QECCConnection.GetEND.m_Type <> 'VALVE') and
(QECCConnection.GetEND.m_Type <> 'MANUALVALVE') and
(isPUMPSETFLOW(QECCConnection.GetEND) = FALSE) and
(isFANSETFLOW(QECCConnection.GetEND) = FALSE) and
(QECCConnection.GetEND.m_Type <> 'DAMPER') and
(QECCConnection.GetEND.m_Type <> 'DAM') and
(QECCConnection.GetEND <> aElement) then
    Result := QECCConnection.GetEND
else
    Result := nil;
end;
end;

{*****}
{* NAME: getMULTIPORTForwardsSystemNoFlow *}
{* DESC: Returns the correct multiport towards which to move *}
{*****}
function TMassFlowQEC.getMULTIPORTForwardsSystemSpecialFlow (aConnection: TConnectionQEC) : TElementQEC;

type ConnectionRecord = record
    ConnectionIN_1 : Integer;
    ConnectionIN_2 : Integer;
    ConnectionOUT_1 : Integer;
    ConnectionOUT_2 : Integer;
    Connection : Integer;
end;

var

```

```

Connection      : ConnectionRecord;
aElement        : TElementQEC;
begin
with MainForm do
begin
aElement := aConnection.GetEND;
if (aElement.m_Type = 'CHILLER:WATER') then
begin
Connection.ConnectionIN_1 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctEVAPORATORWATER_IN.m_CONNECTION;
Connection.ConnectionIN_2 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctCONDENSORWATER_IN.m_CONNECTION;
Connection.ConnectionOUT_1 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctEVAPORATORWATER_OUT.m_CONNECTION;
Connection.ConnectionOUT_2 :=
TWaterCooledChillerQEC(aElement).WaterCooledChillerDE.ctCONDENSORWATER_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'STORAGETANK') then
begin
Connection.ConnectionIN_1 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_IN1.m_CONNECTION;
Connection.ConnectionIN_2 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_IN2.m_CONNECTION;
Connection.ConnectionOUT_1 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_OUT1.m_CONNECTION;
Connection.ConnectionOUT_2 := TStorageTankQEC(aElement).StorageTankDE.ctWATER_OUT2.m_CONNECTION;
end;
if (aElement.m_Type = 'EXCHANGER:WATERTOWATER') then
begin
Connection.ConnectionIN_1 :=
TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_IN1.m_CONNECTION;
Connection.ConnectionIN_2 :=
TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_IN2.m_CONNECTION;
Connection.ConnectionOUT_1 :=
TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_OUT1.m_CONNECTION;
Connection.ConnectionOUT_2 :=
TWaterToWaterExchangerQEC(aElement).WaterToWaterExchangerDE.ctWATER_OUT2.m_CONNECTION;
end;
if (aElement.m_Type = 'EXCHANGER:AIRTOAIR') then
begin
Connection.ConnectionIN_1 :=
TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_IN1.m_CONNECTION;
Connection.ConnectionIN_2 :=
TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_IN2.m_CONNECTION;
Connection.ConnectionOUT_1 :=
TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_OUT1.m_CONNECTION;
Connection.ConnectionOUT_2 :=
TAirToAirExchangerQEC(aElement).AirToAirExchangerDE.ctAIR_OUT2.m_CONNECTION;
end;
if (aElement.m_Type = 'CHILLER:AIR') then
begin
Connection.ConnectionIN_1 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctEVAPORATORWATER_IN.m_CONNECTION;
Connection.ConnectionIN_2 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_IN.m_CONNECTION;
Connection.ConnectionOUT_1 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctEVAPORATORWATER_OUT.m_CONNECTION;
Connection.ConnectionOUT_2 :=
TAirCooledChillerQEC(aElement).AirCooledChillerDE.ctCONDENSORAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'COOLINGTOWER') then
begin
Connection.ConnectionIN_1 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctWATER_IN.m_CONNECTION;
Connection.ConnectionIN_2 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctAIR_IN.m_CONNECTION;
Connection.ConnectionOUT_1 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctWATER_OUT.m_CONNECTION;
Connection.ConnectionOUT_2 := TCoolingTowerQEC(aElement).CoolingTowerDE.ctAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'BULKAIRCOOLER') then
begin
Connection.ConnectionIN_1 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctWATER_IN.m_CONNECTION;
Connection.ConnectionIN_2 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_IN.m_CONNECTION;
Connection.ConnectionOUT_1 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctWATER_OUT.m_CONNECTION;
Connection.ConnectionOUT_2 := TBulkAirCoolerQEC(aElement).BulkAirCoolerDE.ctAIR_OUT.m_CONNECTION;
end;
if (aElement.m_Type = 'COIL') then
begin
Connection.ConnectionIN_1 := TCoilQEC(aElement).CoilDE.ctAIR_IN.m_CONNECTION;
Connection.ConnectionIN_2 := TCoilQEC(aElement).CoilDE.ctWATER_IN.m_CONNECTION;
Connection.ConnectionOUT_1 := TCoilQEC(aElement).CoilDE.ctAIR_OUT.m_CONNECTION;

```

```

    Connection.ConnectionOUT_2:= TCoilQEC(aElement).CoilDE.ctWATER_OUT.m_CONNECTION;
end;
if aConnection.m_CONNECTION = Connection.ConnectionOUT_1 then
    Connection.Connection := Connection.ConnectionIN_1
else
if aConnection.m_CONNECTION = Connection.ConnectionOUT_2 then
    Connection.Connection := Connection.ConnectionIN_2;

if aConnection.m_CONNECTION = Connection.ConnectionIN_1 then
    Connection.Connection := Connection.ConnectionOUT_1
else
if aConnection.m_CONNECTION = Connection.ConnectionIN_2 then
    Connection.Connection := Connection.ConnectionOUT_2;
QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Connection.Connection];
if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
(QECCConnection.GetEND.m_Type <> 'SOURCE:AIR') and
(isWATERCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(isAIRCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(isWATERDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(isAIRDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
(QECCConnection.GetEND.m_Type <> 'DAM') and
(QECCConnection.GetEND <> aElement) then
    Result := QECCConnection.GetEND
else
    Result := nil;
end;
end;

{*****}
{* NAME: SearchBackwardsSingle *}
{* DESC: Procedure which searches backwards (IN->OUT) on the *}
{* connections to set/run the found elements internal *}
{* valve component. Recursive function. *}
{*****}
procedure TMassflowQEC.SearchBackwardsSingle(var aElement : TElementQEC);
var
    Counter : Integer;
    aMSElement : PMASSESEQUENCERecord;
begin
    with MainForm do
    begin
        for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
        begin
            QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
            if (QECCConnection.GetEND = aElement) and (QECCConnection.GetTYPE <> ctCONTROL) then
            begin
                if (QECCConnection.GetSTART.m_Type <> 'SOURCE:WATER') and
                (QECCConnection.GetSTART.m_Type <> 'SOURCE:AIR') and
                (isWATERCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
                (isAIRCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
                (isWATERDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
                (isAIRDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
                (QECCConnection.GetSTART.m_Type <> 'VALVE') and
                (QECCConnection.GetSTART.m_Type <> 'MANUALVALVE') and
                (isPUMPSETFLOW(QECCConnection.GetSTART) = FALSE) and
                (isFANSETFLOW(QECCConnection.GetSTART) = FALSE) and
                (QECCConnection.GetSTART.m_Type <> 'DAMPER') and
                (QECCConnection.GetSTART.m_Type <> 'DAM') then
                begin
                    aElement := QECCConnection.GetSTART;
                    New(aMSElement);
                    aMSElement^.Element := aElement.m_ELEMENT;
                    aMSElement^.Flag := ffBackward;
                    m_MASSFLOWSEQUENCE.Add(aMSElement);
                    if (isMULTIPOINT(QECCConnection.GetSTART)) then
                    begin
                        aElement := getMULTIPOINTBackwards(QECCConnection);
                        if (aElement <> nil) then
                        begin
                            New(aMSElement);
                            aMSElement^.Element := aElement.m_ELEMENT;
                            aMSElement^.Flag := ffBackward;
                            m_MASSFLOWSEQUENCE.Add(aMSElement);
                        end;
                    end;
                end;
                SearchBackwardsSingle(aElement);
            end;
        end;
    end;
end;

```

```

        end;
    end;
end;
end;
end;

{*****}
{* NAME: SearchBackwardsDouble          *}
{* DESC: Procedure which searches backwards (IN->OUT) on the *}
{* connections to set/run the found elements internal *}
{* valve components. Double is for a CONVERGE or for *}
{* a DIVERGE which splits up into two (double) paths. *}
{* Recursive function which calls the recursive single *}
{* search function.                    *}
{*****}
procedure TMassflowQEC.SearchBackwardsDouble(var aElement : TElementQEC);
var
    Counter : Integer;
    CONNECTIONCOUNT : Integer;
    CONNECTION : Array[1..2] of Integer;
    aMSElement : PMASSESEQUENCERRecord;
begin
    with MainForm do
    begin
        CONNECTIONCOUNT := 1;
        for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
        begin
            QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
            if (QECCConnection.GetEND = aElement) and (QECCConnection.GetTYPE <> ctCONTROL) then
            begin
                CONNECTION[CONNECTIONCOUNT]:= QECCConnection.m_CONNECTION;
                CONNECTIONCOUNT := CONNECTIONCOUNT + 1;
            end;
        end;
        QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[1]];
        if (QECCConnection.GetSTART.m_Type <> 'SOURCE:WATER') and
            (QECCConnection.GetSTART.m_Type <> 'SOURCE:AIR') and
            (isWATERCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (isAIRCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (isWATERDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (isAIRDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (QECCConnection.GetSTART.m_Type <> 'VALVE') and
            (QECCConnection.GetSTART.m_Type <> 'MANUALVALVE') and
            (isPUMPSETFLOW(QECCConnection.GetSTART) = FALSE) and
            (isFANSETFLOW(QECCConnection.GetSTART) = FALSE) and
            (QECCConnection.GetSTART.m_Type <> 'DAMPER') and
            (QECCConnection.GetSTART.m_Type <> 'DAM') then
        begin
            aElement := QECCConnection.GetSTART;
            New(aMSElement);
            aMSElement.Element := aElement.m_ELEMENT;
            aMSElement.Flag := ffBackward;
            m_MASSFLOWSEQUENCE.Add(aMSElement);
            if (isMULTIPOINT(QECCConnection.GetSTART)) then
            begin
                aElement := getMULTIPOINTBackwards(QECCConnection);
                if (aElement <> nil) then
                begin
                    New(aMSElement);
                    aMSElement.Element := aElement.m_ELEMENT;
                    aMSElement.Flag := ffBackward;
                    m_MASSFLOWSEQUENCE.Add(aMSElement);
                end;
            end;
            SearchBackwardsSingle(aElement);
        end;
        QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[2]];
        if (QECCConnection.GetSTART.m_Type <> 'SOURCE:WATER') and
            (QECCConnection.GetSTART.m_Type <> 'SOURCE:AIR') and
            (isWATERCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (isAIRCONVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (isWATERDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (isAIRDIVERGETVALVE(QECCConnection.GetSTART) = FALSE) and
            (QECCConnection.GetSTART.m_Type <> 'VALVE') and
            (QECCConnection.GetSTART.m_Type <> 'MANUALVALVE') and
            (isPUMPSETFLOW(QECCConnection.GetSTART) = FALSE) and

```

```

(isFANSETFLOW(QECCConnection.GetSTART) = FALSE) and
(QECCConnection.GetSTART.m_Type <> 'DAMPER') and
(QECCConnection.GetSTART.m_Type <> 'DAM') then
begin
  aElement := QECCConnection.GetSTART;
  New(aMSElement);
  aMSElement^.Element := aElement.m_ELEMENT;
  aMSElement^.Flag := ffBackward;
  m_MASSFLOWSEQUENCE.Add(aMSElement);
  if (isMULTIPOINT(QECCConnection.GetSTART)) then
  begin
    aElement := getMULTIPOINTBackwards(QECCConnection);
    if (aElement <> nil) then
    begin
      New(aMSElement);
      aMSElement^.Element := aElement.m_ELEMENT;
      aMSElement^.Flag := ffBackward;
      m_MASSFLOWSEQUENCE.Add(aMSElement);
    end;
  end;
  SearchBackwardsSingle(aElement);
end;
end;
end;

{*****}
{* NAME: SearchForwardsSingle *}
{* DESC: Procedure which searches forwards (OUT->IN) on the *}
{* connections to set/run the found elements internal *}
{* valve component. Recursive function. *}
{*****}
procedure TMassflowQEC.SearchForwardsSingle(var aElement : TElementQEC);
var
  Counter : Integer;
  aMSElement : PMASSSEQUENCERecord;
begin
  with MainForm do
  begin
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
    begin
      QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
      if (QECCConnection.GetSTART = aElement) and (QECCConnection.GetTYPE <> ctCONTROL) then
      begin
        if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
          (QECCConnection.GetEND.m_Type <> 'SOURCE:AIR') and
          (isWATERCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
          (isAIRCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
          (isWATERDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
          (isAIRDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
          (QECCConnection.GetEND.m_Type <> 'VALVE') and
          (QECCConnection.GetEND.m_Type <> 'MANUALVALVE') and
          (isPUMPSETFLOW(QECCConnection.GetEND) = FALSE) and
          (isFANSETFLOW(QECCConnection.GetEND) = FALSE) and
          (QECCConnection.GetEND.m_Type <> 'DAMPER') and
          (QECCConnection.GetEND.m_Type <> 'DAM') then
        begin
          aElement := QECCConnection.GetEND;
          New(aMSElement);
          aMSElement^.Element := aElement.m_ELEMENT;
          aMSElement^.Flag := ffForward;
          m_MASSFLOWSEQUENCE.Add(aMSElement);
          if (isMULTIPOINT(QECCConnection.GetEND)) then
          begin
            aElement := getMULTIPOINTForwards(QECCConnection);
            if (aElement <> nil) then
            begin
              New(aMSElement);
              aMSElement^.Element := aElement.m_ELEMENT;
              aMSElement^.Flag := ffForward;
              m_MASSFLOWSEQUENCE.Add(aMSElement);
            end;
          end;
          SearchForwardsSingle(aElement);
        end;
      end;
    end;
  end;
end;
end;

```

```

end;
end;

{*****}
{* NAME: SearchForwardsDouble *}
{* DESC: Procedure which searches forwards (OUT->IN) on the *}
{* connections to set/run the found elements internal *}
{* valve components. Double is for a CONVERGE or for *}
{* a DIVERGE which splits up into two (double) paths. *}
{*****}
procedure TMassflowQEC.SearchForwardsDouble(var aElement : TElementQEC);
var
  Counter : Integer;
  CONNECTIONCOUNT : Integer;
  CONNECTION : Array[1..2] of Integer;
  aMSElement : PMASSESEQUENCERecord;
begin
  with MainForm do
  begin
    CONNECTIONCOUNT := 1;
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
    begin
      QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
      if (QECCConnection.GetSTART = aElement) and (QECCConnection.GetTYPE <> ctCONTROL) then
      begin
        CONNECTION[CONNECTIONCOUNT]:= QECCConnection.m_CONNECTION;
        CONNECTIONCOUNT := CONNECTIONCOUNT + 1;
      end;
    end;
    QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[1]];
    if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
      (QECCConnection.GetEND.m_Type <> 'SOURCE:AIR') and
      (isWATERCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (isAIRCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (isWATERDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (isAIRDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (QECCConnection.GetEND.m_Type <> 'VALVE') and
      (QECCConnection.GetEND.m_Type <> 'MANUALVALVE') and
      (isPUMPSETFLOW(QECCConnection.GetEND) = FALSE) and
      (isFANSETFLOW(QECCConnection.GetEND) = FALSE) and
      (QECCConnection.GetEND.m_Type <> 'DAMPER') and
      (QECCConnection.GetEND.m_Type <> 'DAM') then
    begin
      aElement := QECCConnection.GetEND;
      New(aMSElement);
      aMSElement^.Element := aElement.m_ELEMENT;
      aMSElement^.Flag := ffForward;
      m_MASSFLOWSEQUENCE.Add(aMSElement);
      if (isMULTIPOINT(QECCConnection.GetEND)) then
      begin
        aElement := getMULTIPOINTForwards(QECCConnection);
        if (aElement <> nil) then
        begin
          New(aMSElement);
          aMSElement^.Element := aElement.m_ELEMENT;
          aMSElement^.Flag := ffForward;
          m_MASSFLOWSEQUENCE.Add(aMSElement);
        end;
      end;
      SearchForwardsSingle(aElement);
    end;
    QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[2]];
    if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
      (QECCConnection.GetEND.m_Type <> 'SOURCE:AIR') and
      (isWATERCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (isAIRCONVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (isWATERDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (isAIRDIVERGETVALVE(QECCConnection.GetEND) = FALSE) and
      (QECCConnection.GetEND.m_Type <> 'VALVE') and
      (QECCConnection.GetEND.m_Type <> 'MANUALVALVE') and
      (isPUMPSETFLOW(QECCConnection.GetEND) = FALSE) and
      (isFANSETFLOW(QECCConnection.GetEND) = FALSE) and
      (QECCConnection.GetEND.m_Type <> 'DAMPER') and
      (QECCConnection.GetEND.m_Type <> 'DAM') then
    begin
      aElement := QECCConnection.GetEND;

```

```

{* connections to set/run the found elements internal *}
{* valve components. Double is for a CONVERGE or for *}
{* a DIVERGE which splits up into two (double) paths. *}
{*****}
procedure TMassflowQEC.SearchForwardsDoubleSystemSpecialFlowCase1(const aTime : TimeType; var aElement : TElementQEC);
var
  Counter : Integer;
  CONNECTIONCOUNT : Integer;
  CONNECTION : Array[1..2] of Integer;
begin
  with MainForm do
  begin
    CONNECTIONCOUNT := 1;
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
    begin
      QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
      if (QECCConnection.GetSTART = aElement) and (QECCConnection.GetTYPE <> ctCONTROL) then
      begin
        CONNECTION[CONNECTIONCOUNT]:= QECCConnection.m_CONNECTION;
        CONNECTIONCOUNT := CONNECTIONCOUNT + 1;
      end;
    end;
    QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[1]];
    if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
      (QECCConnection.GetEND.m_Type <> 'DAM') then
    begin
      aElement := QECCConnection.GetEND;
      if (aElement.m_Type = 'VALVE') or
        (aElement.m_Type = 'MANUALVALVE') or
        (aElement.m_Type = 'PUMP') then
      begin
        aElement.DoInternalMassflow(aTime, ffSpecial);
      end
      else
      begin
        aElement.DoInternalMassflow(aTime, ffForward);
      end;
      if (isMULTIPORT(QECCConnection.GetEND)) then
      begin
        aElement := getMULTIPORTForwardsSystemSpecialFlow(QECCConnection);
        if (aElement <> nil) then
        begin
          aElement.DoInternalMassflow(aTime, ffForward);
        end;
      end;
      SearchForwardsSingleSystemSpecialFlowCase1(aTime, aElement);
    end;
    QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[2]];
    if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
      (QECCConnection.GetEND.m_Type <> 'DAM') then
    begin
      aElement := QECCConnection.GetEND;
      if (aElement.m_Type = 'VALVE') or
        (aElement.m_Type = 'MANUALVALVE') or
        (aElement.m_Type = 'PUMP') then
      begin
        aElement.DoInternalMassflow(aTime, ffSpecial);
      end
      else
      begin
        aElement.DoInternalMassflow(aTime, ffForward);
      end;
      if (isMULTIPORT(QECCConnection.GetEND)) then
      begin
        aElement := getMULTIPORTForwardsSystemSpecialFlow(QECCConnection);
        if (aElement <> nil) then
        begin
          aElement.DoInternalMassflow(aTime, ffForward);
        end;
      end;
      SearchForwardsSingleSystemSpecialFlowCase1(aTime, aElement);
    end;
  end;
end;
end;
end;

```

```

{*****}
{* NAME: SearchForwardsSingleSystemSpecialFlowCase2      *}
{* DESC: Procedure which searches forwards (OUT->IN) on the *}
{* connections to set/run the found elements internal *}
{* flow component. Recursive function.                    *}
{*****}
procedure TMassflowQEC.SearchForwardsSingleSystemSpecialFlowCase2(const aTime : TimeType; var aElement : TElementQEC);
var
  Counter : Integer;
begin
  with MainForm do
  begin
    aElement.DoInternalMassflow(aTime, ffForward);
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
    begin
      QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
      if (QECCConnection.GetSTART = aElement) and (QECCConnection.GetTYPE <> ctCONTROL) then
      begin
        if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
          (QECCConnection.GetEND.m_Type <> 'DIVERGE:WATER') and
          (QECCConnection.GetEND.m_Type <> 'DAM') and
          (QECCConnection.GetEND.m_Type <> 'VALVE') and
          (QECCConnection.GetEND.m_Type <> 'MANUALVALVE') and
          (isPUMPSETFLOW(QECCConnection.GetEND) = FALSE) then
          begin
            aElement := QECCConnection.GetEND;
            aElement.DoInternalMassflow(aTime, ffForward);
            if (isMULTIPOINT(QECCConnection.GetEND)) then
            begin
              aElement := getMULTIPOINTForwardsSystemSpecialFlow(QECCConnection);
              if (aElement <> nil) then
              begin
                aElement.DoInternalMassflow(aTime, ffForward);
              end;
            end;
            SearchForwardsSingleSystemSpecialFlowCase2(aTime, aElement);
          end;
          if (QECCConnection.GetEND.m_Type = 'DIVERGE:WATER') then
          begin
            aElement := QECCConnection.GetEND;
            aElement.DoInternalMassflow(aTime, ffForward);
            SearchForwardsDoubleSystemSpecialFlowCase2(aTime, aElement);
          end;
        end;
      end;
    end;
  end;
end;

{*****}
{* NAME: SearchForwardsDoubleSystemSpecialFlowCase2      *}
{* DESC: Procedure which searches forwards (OUT->IN) on the *}
{* connections to set/run the found elements internal *}
{* valve components. Double is for a CONVERGE or for *}
{* a DIVERGE which splits up into two (double) paths. *}
{*****}
procedure TMassflowQEC.SearchForwardsDoubleSystemSpecialFlowCase2(const aTime : TimeType; var aElement : TElementQEC);
var
  Counter : Integer;
  CONNECTIONCOUNT : Integer;
  CONNECTION : Array[1..2] of Integer;
begin
  with MainForm do
  begin
    CONNECTIONCOUNT := 1;
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
    begin
      QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
      if (QECCConnection.GetSTART = aElement) and (QECCConnection.GetTYPE <> ctCONTROL) then
      begin
        CONNECTION[CONNECTIONCOUNT] := QECCConnection.m_CONNECTION;
        CONNECTIONCOUNT := CONNECTIONCOUNT + 1;
      end;
    end;
    QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[1]];
    if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
      (QECCConnection.GetEND.m_Type <> 'DAM') and

```

```

(QECCConnection.GetEND.m_Type <> 'VALVE') and
(QECCConnection.GetEND.m_Type <> 'MANUALVALVE') and
(isPUMPSETFLOW(QECCConnection.GetEND) = FALSE) then
begin
  aElement := QECCConnection.GetEND;
  aElement.DoInternalMassflow(aTime, ffForward);
  if (isMULTIPOINT(QECCConnection.GetEND)) then
  begin
    aElement := getMULTIPOINTForwardsSystemSpecialFlow(QECCConnection);
    if (aElement <> nil) then
    begin
      aElement.DoInternalMassflow(aTime, ffForward);
    end;
  end;
  SearchForwardsSingleSystemSpecialFlowCase2(aTime, aElement);
end;
QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[CONNECTION[2]];
if (QECCConnection.GetEND.m_Type <> 'SOURCE:WATER') and
(QECCConnection.GetEND.m_Type <> 'DAM') and
(QECCConnection.GetEND.m_Type <> 'VALVE') and
(QECCConnection.GetEND.m_Type <> 'MANUALVALVE') and
(isPUMPSETFLOW(QECCConnection.GetEND) = FALSE) then
begin
  aElement := QECCConnection.GetEND;
  aElement.DoInternalMassflow(aTime, ffForward);
  if (isMULTIPOINT(QECCConnection.GetEND)) then
  begin
    aElement := getMULTIPOINTForwardsSystemSpecialFlow(QECCConnection);
    if (aElement <> nil) then
    begin
      aElement.DoInternalMassflow(aTime, ffForward);
    end;
  end;
  SearchForwardsSingleSystemSpecialFlowCase2(aTime, aElement);
end;
end;
end;

{*****}
{* NAME: UpdateElementConnections *}
{* DESC: Renumbers all elements and connections. *}
{*****}
procedure TMassflowQEC.UpdateElementConnections;
var
  Counter : Integer;
begin
  with MainForm do
  begin
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
    begin
      QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
      QECElement.m_ELEMENT := Counter;
    end;
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ConnectionList.Count-1 do
    begin
      QECCConnection := TMDIChildWin(ActiveMDIChild).ConnectionList.Items[Counter];
      QECCConnection.m_CONNECTION := Counter;
    end;
  end;
end;

{*****}
{* NAME: SortMassFlowSequence *}
{* DESC: Procedure which sorts elements in componentlist *}
{* according to the rules of flow. Flow components *}
{* must be FIRST with non flow components called last. *}
{*****}
procedure TMassflowQEC.SortMassFlowSequence;
var
  Counter : Integer;
begin
  with MainForm do
  begin
    {First move all the possible special flow cases to the back}
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
    begin

```

```

QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
if QECElement.m_Type = 'DAM' then
begin
    TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'DIVERGE:WATER' then
    begin
        if (isWATERDIVERGETPIECE(QECElement) = TRUE) then
        begin
            TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
    end;
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'DIVERGE:AIR' then
    begin
        if (isAIRDIVERGETPIECE(QECElement) = TRUE) then
        begin
            TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
    end;
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'CONVERGE:WATER' then
    begin
        if (isWATERCONVERGETPIECE(QECElement) = TRUE) then
        begin
            TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
    end;
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'CONVERGE:AIR' then
    begin
        if (isAIRCONVERGETPIECE(QECElement) = TRUE) then
        begin
            TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
    end;
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'CONVERGE:WATER' then
    begin
        if (isWATERCONVERGETVALVE(QECElement) = TRUE) then
        begin
            TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
    end;
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'CONVERGE:AIR' then
    begin
        if (isAIRCONVERGETVALVE(QECElement) = TRUE) then
        begin
            TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
    end;
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'DIVERGE:WATER' then

```

```

begin
  if (isWATERDIVERGETVALVE(QECElement) = TRUE) then
    begin
      TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
      inc(m_MASSFLOWITERATIONS);
    end;
  end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
  QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
  if QECElement.m_Type = 'DIVERGE:AIR' then
    begin
      if (isAIRDIVERGETVALVE(QECElement) = TRUE) then
        begin
          TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
          inc(m_MASSFLOWITERATIONS);
        end;
      end;
    end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
  QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
  if QECElement.m_Type = 'PUMP' then
    begin
      if (isPUMPSETFLOW(QECElement) = TRUE) then
        begin
          TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
      end;
    end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
  QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
  if QECElement.m_Type = 'VALVE' then
    begin
      TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
    end;
  end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
  QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
  if QECElement.m_Type = 'MANUALVALVE' then
    begin
      TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
    end;
  end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
  QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
  if QECElement.m_Type = 'FAN' then
    begin
      if (isFANSETFLOW(QECElement) = TRUE) then
        begin
          TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
        end;
      end;
    end;
  end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
  QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
  if QECElement.m_Type = 'DAMPER' then
    begin
      TMDIChildWin(ActiveMDIChild).ComponentList.Move(Counter,0);
    end;
  end;
end;
end;
end;
end;

{*****}
{* NAME: BuildMassFlowSequence *}
{* DESC: Sets elements into MASSFLOWSEQUENCE list with flags.*}
{* This list only need to be run once every iteration.*}
{*****}
procedure TMassflowQEC.BuildMassFlowSequence;
var

```

```

Counter      : Integer;
aMSElement  : PMASSEQUENCERecord;
begin
  UpdateElementConnections;
  with MainForm do
  begin
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
    begin
      QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
      if QCEElement.m_Type = 'DAMPER' then
      begin
        New(aMSElement);
        aMSElement^.Element := QCEElement.m_ELEMENT;
        aMSElement^.Flag := ffBackward;
        m_MASSFLOWSEQUENCE.Add(aMSElement);
        SearchBackWardsSingle(QCEElement);
        QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
        New(aMSElement);
        aMSElement^.Element := QCEElement.m_ELEMENT;
        aMSElement^.Flag := ffForward;
        m_MASSFLOWSEQUENCE.Add(aMSElement);
        SearchForwardsSingle(QCEElement);
      end;
    end;
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
    begin
      QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
      if QCEElement.m_Type = 'FAN' then
      begin
        if (isFANSETFLOW(QCEElement) = TRUE) then
        begin
          New(aMSElement);
          aMSElement^.Element := QCEElement.m_ELEMENT;
          aMSElement^.Flag := ffBackward;
          m_MASSFLOWSEQUENCE.Add(aMSElement);
          SearchBackWardsSingle(QCEElement);
          QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
          New(aMSElement);
          aMSElement^.Element := QCEElement.m_ELEMENT;
          aMSElement^.Flag := ffForward;
          m_MASSFLOWSEQUENCE.Add(aMSElement);
          SearchForwardsSingle(QCEElement);
        end;
      end;
    end;
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
    begin
      QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
      if QCEElement.m_Type = 'VALVE' then
      begin
        New(aMSElement);
        aMSElement^.Element := QCEElement.m_ELEMENT;
        aMSElement^.Flag := ffBackward;
        m_MASSFLOWSEQUENCE.Add(aMSElement);
        SearchBackWardsSingle(QCEElement);
        QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
        New(aMSElement);
        aMSElement^.Element := QCEElement.m_ELEMENT;
        aMSElement^.Flag := ffForward;
        m_MASSFLOWSEQUENCE.Add(aMSElement);
        SearchForwardsSingle(QCEElement);
      end;
    end;
    for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
    begin
      QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
      if QCEElement.m_Type = 'MANUALVALVE' then
      begin
        New(aMSElement);
        aMSElement^.Element := QCEElement.m_ELEMENT;
        aMSElement^.Flag := ffBackward;
        m_MASSFLOWSEQUENCE.Add(aMSElement);
        SearchBackWardsSingle(QCEElement);
        QCEElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
        New(aMSElement);
        aMSElement^.Element := QCEElement.m_ELEMENT;

```

```

    aMSElement^.Flag := ffForward;
    m_MASSFLOWSEQUENCE.Add(aMSElement);
    SearchForwardsSingle(QECElement);
end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'PUMP' then
    begin
        if (isPUMPSETFLOW(QECElement) = TRUE) then
        begin
            New(aMSElement);
            aMSElement^.Element := QECElement.m_ELEMENT;
            aMSElement^.Flag := ffBackward;
            m_MASSFLOWSEQUENCE.Add(aMSElement);
            SearchBackwardsSingle(QECElement);
            QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
            New(aMSElement);
            aMSElement^.Element := QECElement.m_ELEMENT;
            aMSElement^.Flag := ffForward;
            m_MASSFLOWSEQUENCE.Add(aMSElement);
            SearchForwardsSingle(QECElement);
        end;
    end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'DIVERGE:AIR' then
    begin
        if (isAIRDIVERGETVALVE(QECElement) = TRUE) then
        begin
            New(aMSElement);
            aMSElement^.Element := QECElement.m_ELEMENT;
            aMSElement^.Flag := ffBoth;
            m_MASSFLOWSEQUENCE.Add(aMSElement);
            SearchForwardsDouble(QECElement);
        end;
    end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'DIVERGE:WATER' then
    begin
        if (isWATERDIVERGETVALVE(QECElement) = TRUE) then
        begin
            New(aMSElement);
            aMSElement^.Element := QECElement.m_ELEMENT;
            aMSElement^.Flag := ffBoth;
            m_MASSFLOWSEQUENCE.Add(aMSElement);
            SearchForwardsDouble(QECElement);
        end;
    end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'CONVERGE:AIR' then
    begin
        if (isAIRCONVERGETVALVE(QECElement) = TRUE) then
        begin
            New(aMSElement);
            aMSElement^.Element := QECElement.m_ELEMENT;
            aMSElement^.Flag := ffBoth;
            m_MASSFLOWSEQUENCE.Add(aMSElement);
            SearchBackwardsDouble(QECElement);
        end;
    end;
end;
for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
begin
    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
    if QECElement.m_Type = 'CONVERGE:WATER' then
    begin

```

```

if (isWATERCONVERGETVALVE(QCElement) = TRUE) then
begin
    New(aMSElement);
    aMSElement^.Element := QCElement.m_ELEMENT;
    aMSElement^.Flag := ffBoth;
    m_MASSFLOWSEQUENCE.Add(aMSElement);
    SearchBackwardsDouble(QCElement);
end;
end;
end;
end;

{*****}
{ * NAME: ReportMassFlowSequence * }
{ * DESC: Report the sequence of elements as contained in the * }
{ * massflow sequence. * }
{*****}
procedure TMassflowQEC.ReportMassFlowSequence;
var
    Counter : Integer;
    aMSElement : PMASSESEQUENCERecord;
begin
    with MainForm do
    begin
        for Counter := 0 to m_MASSFLOWSEQUENCE.Count-1 do
        begin
            aMSElement := m_MASSFLOWSEQUENCE.Items[Counter];
            QCElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[aMSElement^.Element];
            SimulationStatus.Output.Lines.Add(InrToStr(QCElement.m_ELEMENT)+' '+QCElement.m_Description);
        end;
    end;
end;

{*****}
{ * NAME: SolveSystemSpecialFlowCase1 * }
{*****}
procedure TMassflowQEC.SolveSystemSpecialFlowCase1(aTime : TimeType);
var
    Counter : Integer;
    aElement : TElementQEC;
begin
    with MainForm do
    begin
        for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
        begin
            QCElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
            if QCElement.m_Type = 'DAM' then
            begin
                if isDAMSETFLOW(QCElement) = TRUE then
                begin
                    aElement := TDamQEC(QCElement).DamDE.ctWATER_OUT.GetEND;
                    QCElement.DoInternalMassflow(aTime, ffForward);
                    if (aElement.m_Type = 'VALVE') or
                       (aElement.m_Type = 'MANUALVALVE') or
                       (aElement.m_Type = 'DIVERGE:WATER') or
                       (aElement.m_Type = 'PUMP') then
                    begin
                        aElement.DoInternalMassflow(aTime, ffSpecial);
                    end
                    else
                    begin
                        aElement.DoInternalMassflow(aTime, ffForward);
                    end;
                    if (aElement.m_Type = 'DIVERGE:WATER') then
                    begin
                        SearchForwardsDoubleSystemSpecialFlowCase1(aTime, aElement);
                    end
                    else
                    begin
                        SearchForwardsSingleSystemSpecialFlowCase1(aTime, aElement);
                    end;
                end;
            end;
            if isDAMSETFLOWOVERFLOW(QCElement) = TRUE then
            begin
                if TDamQEC(QCElement).DamDE.ctWATER_OVERFLOW_OUT <> nil then

```

```

                begin
                    aElement := TDamQEC(QECElement).DamDE.ctWATER_OVERFLOW_OUT.GetEND;
                    QECElement.DoInternalMassflow(aTime, ffReset);
                    SearchForwardsSingle(aElement);
                end;
            end;
        end;
    end;
end;
{*****}
{* NAME: SolveSystemSpecialFlowCase2          *}
{*****}
procedure TMassflowQEC.SolveSystemSpecialFlowCase2(aTime : TimeType);
var
    Counter      : Integer;
    aElement     : TElementQEC;
begin
    with MainForm do
        begin
            for Counter := 0 to TMDIChildWin(ActiveMDIChild).ComponentList.Count-1 do
                begin
                    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[Counter];
                    if QECElement.m_Type = 'DAM' then
                        begin
                            if isDAMSETFLOWOVERFLOW(QECElement) = TRUE then
                                begin
                                    if TDamQEC(QECElement).DamDE.ctWATER_OVERFLOW_OUT <> nil then
                                        begin
                                            aElement := TDamQEC(QECElement).DamDE.ctWATER_OVERFLOW_OUT.GetEND;
                                            QECElement.DoInternalMassflow(aTime, ffForward);
                                            SearchForwardsSingleSystemSpecialFlowCase2(aTime,aElement);
                                        end;
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

{*****}
{* NAME: SolveSystemFlow                      *}
{*****}
procedure TMassflowQEC.SolveSystemFlow(aTime : TimeType);
var
    Counter      : Integer;
    aMSElement  : PMASSESEQUENCERecord;
begin
    with MainForm do
        begin
            for Counter := 0 to m_MASSFLOWSEQUENCE.Count-1 do
                begin
                    aMSElement := m_MASSFLOWSEQUENCE.Items[Counter];
                    QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[aMSElement^.Element];
                    QECElement.DoInternalMassflow(aTime, aMSElement^.Flag);
                end;
            end;
        end;
    end;
end;

{*****}
{* NAME: SolveMassFlow                        *}
{* DESC: Procedure calls the MASSFLOWSEQUENCE and it then run*}
{* the elements DoInternalMassFlowState.          *}
{*****}
procedure TMassflowQEC.SolveMassFlow(aTime : TimeType);
var
    Iterations   : Integer;
begin
    with MainForm do
        begin
            for Iterations := 1 to m_MASSFLOWITERATIONS do
                begin
                    SolveSystemFlow(aTime);
                end;
            end;
            SolveSystemSpecialFlowCase1(aTime);
            SolveSystemSpecialFlowCase2(aTime);
        end;
    end;
end;

```

```

end;
end;

{*****}
{ * NAME: ValidateMassflow * }
{ * DESC: Check Massflow Balance over setflow elements * }
{*****}
function TMassflowQEC.ValidateMassflow ( aTime : TimeType) : Boolean;
var
  Counter      : Integer;
  aMSElement  : PMASSESEQUENCERecord;
  Balance      : Boolean;
begin
  with MainForm do
  begin
    for Counter:= 1 to m_MASSFLOWITERATIONS + 1 do
    begin
      SolveSystemFlow(aTime);
    end;
    Balance := TRUE;
    for Counter := 0 to m_MASSFLOWSEQUENCE.Count-1 do
    begin
      aMSElement := m_MASSFLOWSEQUENCE.Items[Counter];
      QECElement := TMDIChildWin(ActiveMDIChild).ComponentList.Items[aMSElement^.Element];
      if QECElement.m_Type = 'PUMP' then
      begin
        if TPumpQEC(QECElement).PumpDE.ctWATER_IN.GetMassFlow (aTime) <>
          TPumpQEC(QECElement).PumpDE.ctWATER_OUT.GetMassFlow(aTime) then
          Balance := FALSE;
        end
      else
        if QECElement.m_Type = 'VALVE' then
        begin
          if TValveQEC(QECElement).ValveDE.ctWATER_IN.GetMassFlow (aTime) <>
            TValveQEC(QECElement).ValveDE.ctWATER_OUT.GetMassFlow(aTime) then
            Balance := FALSE;
          end
        else
          if QECElement.m_Type = 'MANUALVALVE' then
          begin
            if TManualValveQEC(QECElement).ManualValveDE.ctWATER_IN.GetMassFlow (aTime) <>
              TManualValveQEC(QECElement).ManualValveDE.ctWATER_OUT.GetMassFlow(aTime) then
              Balance := FALSE;
            end;
          if QECElement.m_Type = 'FAN' then
          begin
            if TFanQEC(QECElement).FanDE.ctAIR_IN.GetMassFlow (aTime) <>
              TFanQEC(QECElement).FanDE.ctAIR_OUT.GetMassFlow(aTime) then
              Balance := FALSE;
            end
          else
            if QECElement.m_Type = 'DAMPER' then
            begin
              if TDamperQEC(QECElement).DamperDE.ctAIR_IN.GetMassFlow(aTime) <>
                TDamperQEC(QECElement).DamperDE.ctAIR_OUT.GetMassFlow(aTime) then
                Balance := FALSE;
              end;
            end;
          if Balance = TRUE then
            Result := TRUE
          else
            Result := FALSE;
          end;
        end;
      end;
    end;
  end;
end;

{*****}
{ * NAME: DestroyMassFlowSequence * }
{ * DESC: Destroys and free MASSFLOWSEQUENCE list * }
{*****}
destructor TMassflowQEC.Destroy;
var
  Counter : Integer;
  aMSElement  : PMASSESEQUENCERecord;
begin
  inherited Destroy;

```

```
for Counter := 0 to m_MASSFLOWSEQUENCE.Count-1 do
begin
  aMSElement := m_MASSFLOWSEQUENCE.Items[Counter];
  Dispose(aMSElement);
end;
m_MASSFLOWSEQUENCE.Free;
m_MASSFLOWSEQUENCE := nil;
m_MASSFLOWITERATIONS := 1;
end;
end.
```

APPENDIX C
TELKOM DATA BUILDING SYSTEM SPECIFICATION

The following tables display the detailed breakdown of the energy cost of each retrofit.

Return air economiser cycle

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	563 100	299 010	952 100
MAXIMUM DEMAND	399 180	195 740	592 920
TOTAL	1 852 270	492 750	1 540 020

Evaporative cooler

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	655 667	290 529	946 196
MAXIMUM DEMAND	407 101	187 993	595 094
TOTAL	1 062 768	478 552	1 541 290

Setpoint setback

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	639 373	291 467	930 840
MAXIMUM DEMAND	407 144	193 721	600 865
TOTAL	1 046 517	485 158	1 531 705

Fan scheduling

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	637 472	293 151	930 623
MAXIMUM DEMAND	407 144	193 721	603 224
TOTAL	1 045 325	488 522	1 533 847

Chillers and fan scheduling

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	619 213	283 787	903 000
MAXIMUM DEMAND	407 853	195 371	603 224
TOTAL	1 027 066	479 158	1 506 224

Economiser, evaporative cooler, light scheduling, setpoint setback

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	598 759	273 489	872 248
MAXIMUM DEMAND	400 164	192 436	592 600
TOTAL	998 923	465 925	1 437 916

Economiser, evaporative cooler, light scheduling, fan and chiller scheduling

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	580 903	264 269	845 172
MAXIMUM DEMAND	460 149	192 595	592 744
TOTAL	981 052	456 864	1 437 916

First floor

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	620 475	285 705	906 180
MAXIMUM DEMAND	377 216	182 563	559 779
TOTAL	997 691	468 268	1 465 959

Ground floor

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	623 321	286 557	909 878
MAXIMUM DEMAND	379 909	183 952	563 661
TOTAL	1 003 230	470 309	1 473 539

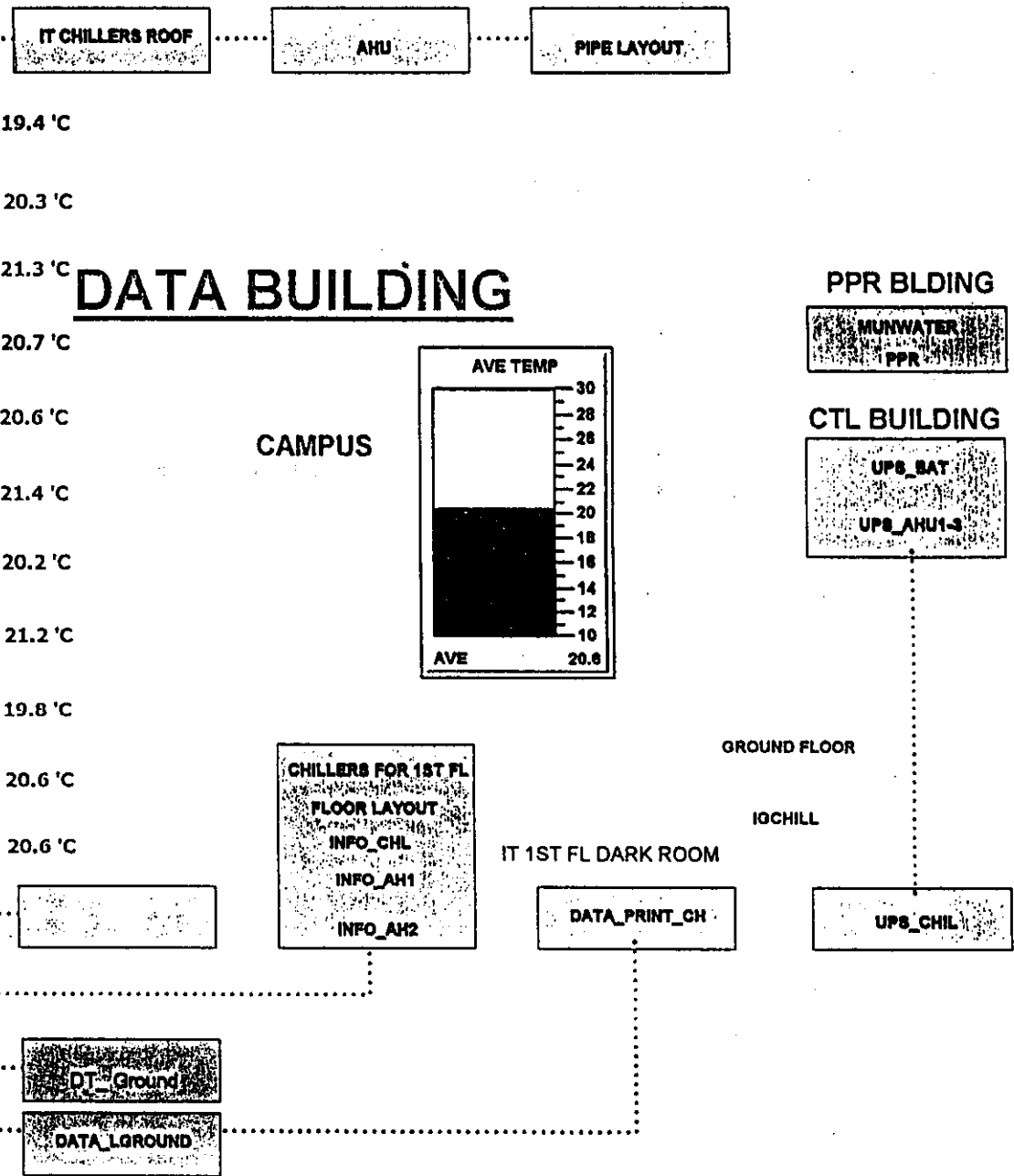
Combined retrofits with setpoint setback

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	477 729	218 185	695 912
MAXIMUM DEMAND	324 304	154 456	478 760
TOTAL	802 031	372 641	1 174 672

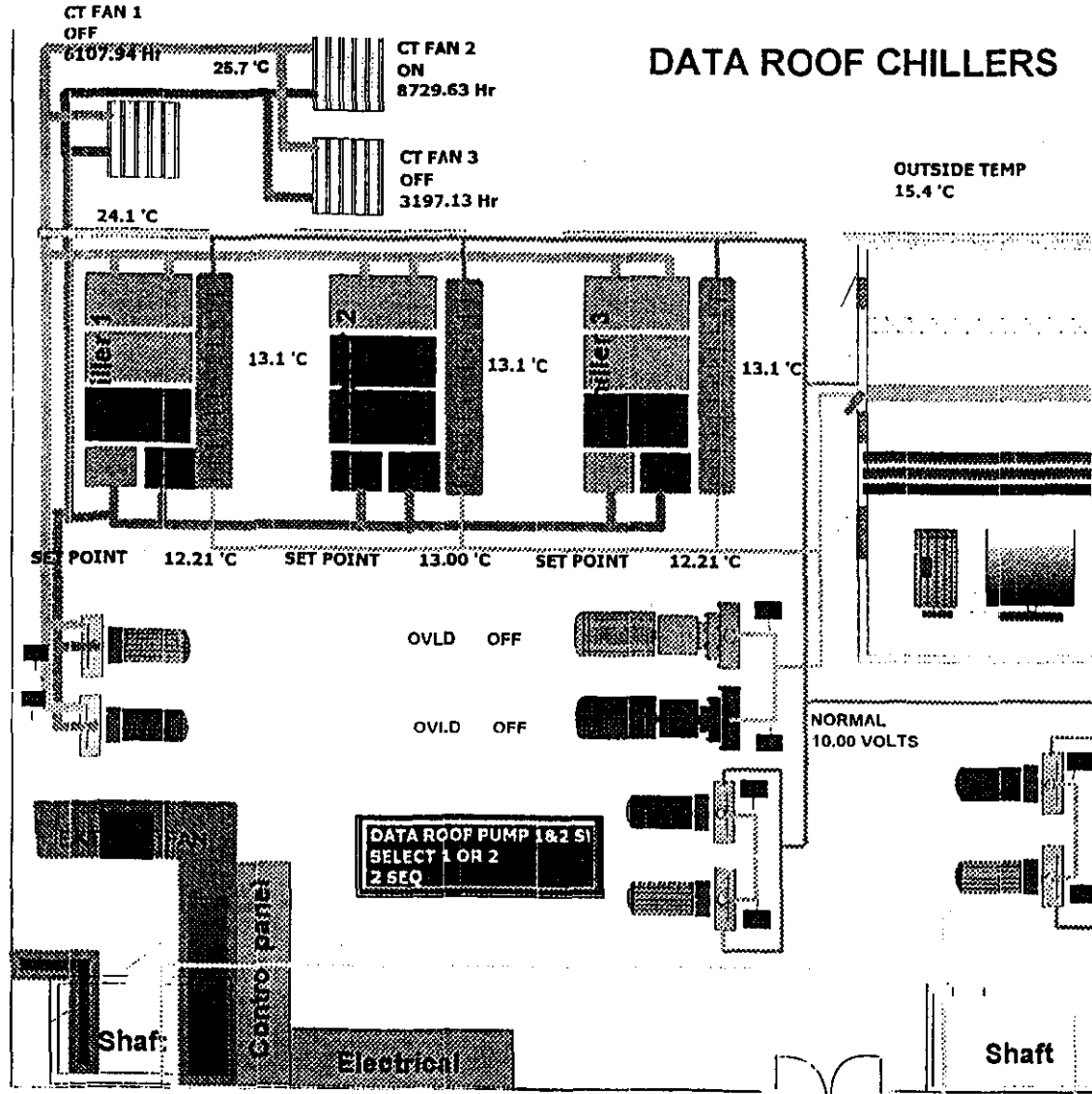
Combined retrofits, with fan and chiller scheduling

	SUMMER (R)	WINTER (R)	TOTAL (R)
ACTIVE	961 378	208 679	670 055
MAXIMUM DEMAND	326 645	156 498	483 143
TOTAL	788 023	365 175	1 153 198

ROOF					
13TH	18.7 °C	20.7 °C	18.0 °C	20.9 °C	
12TH	20.5 °C	20.7 °C	20.2 °C	21.8 °C	19.3 °C
11TH	21.7 °C	21.7 °C	21.4 °C	21.7 °C	
10TH	19.9 °C	21.9 °C	21.7 °C	20.3 °C	
9TH	21.1 °C	20.5 °C	21.4 °C	21.7 °C	19.5 °C
8TH	21.8 °C	22.2 °C	21.8 °C	21.7 °C	
7TH	19.7 °C	21.5 °C	21.3 °C	20.2 °C	18.1 °C
6TH	21.2 °C	21.7 °C	21.5 °C	21.0 °C	
5TH	20.6 °C	18.5 °C	20.9 °C	20.5 °C	
4TH	20.2 °C	21.4 °C	20.7 °C	21.5 °C	20.8 °C
3RD	20.6 °C	21.2 °C	21.0 °C	21.4 °C	19.7 °C
2ND					
1ST	23.6 °C	23.9 °C	22.2 °C	18.7 °C	19.7 °C
GRND					
LG	20.8 °C				

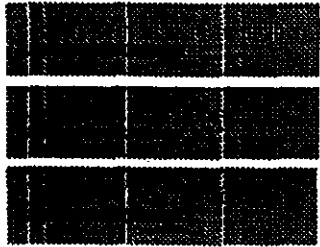


DATA ROOF CHILLERS



DATA ROOF CT 1,2&3 SWITC
SELECT 1,2 OR 3
2 SEQ

DATA ROOF CNPMP 1&2 SWI
SELECT 1 OR 2
2 SEQ



COND PUMP 1 OVL D
OFF

COND PUMP 2 OVL D
OFF

10 15 20 25 30 35
SET POINT 29.00
RETURN TEMP 25.7

SECND 6
9999.00 SEC

ANK LOW LEVEL
LOW LEVEL
NORMAL
NONE
OFF

DATA ROOF P PMP 1&2 SWITCHOV
SELECT 1 OR 2
1 SEQ

AHU HEATING OUTPUT
HEATING
NORMAL
NONE
100.0 %

data_main2

PIPE LAYOUT

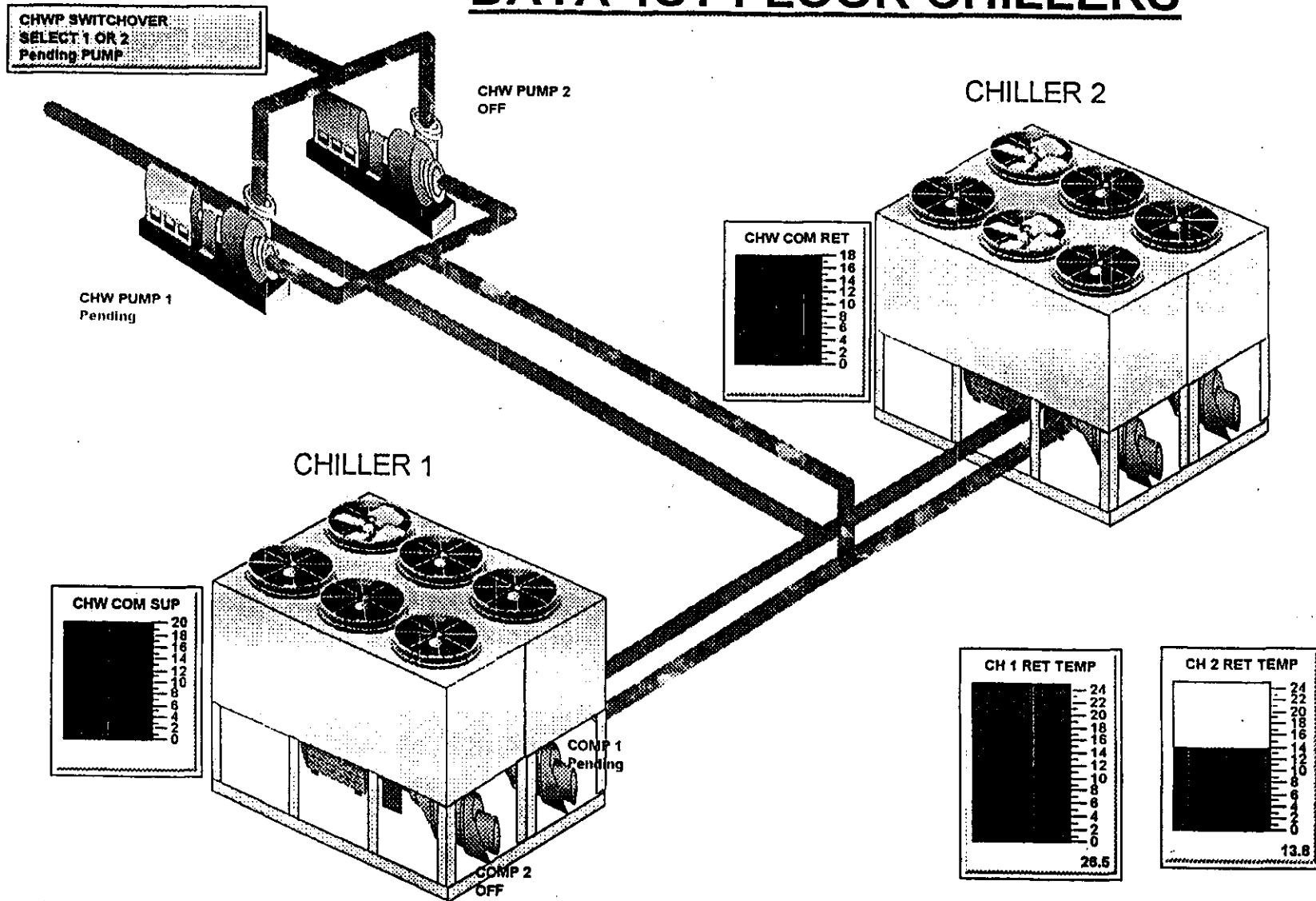


DATA ROOF PUMP 3&4 SWITCHOV
SELECT 3 OR 4
4 SEQ

13.1 °C

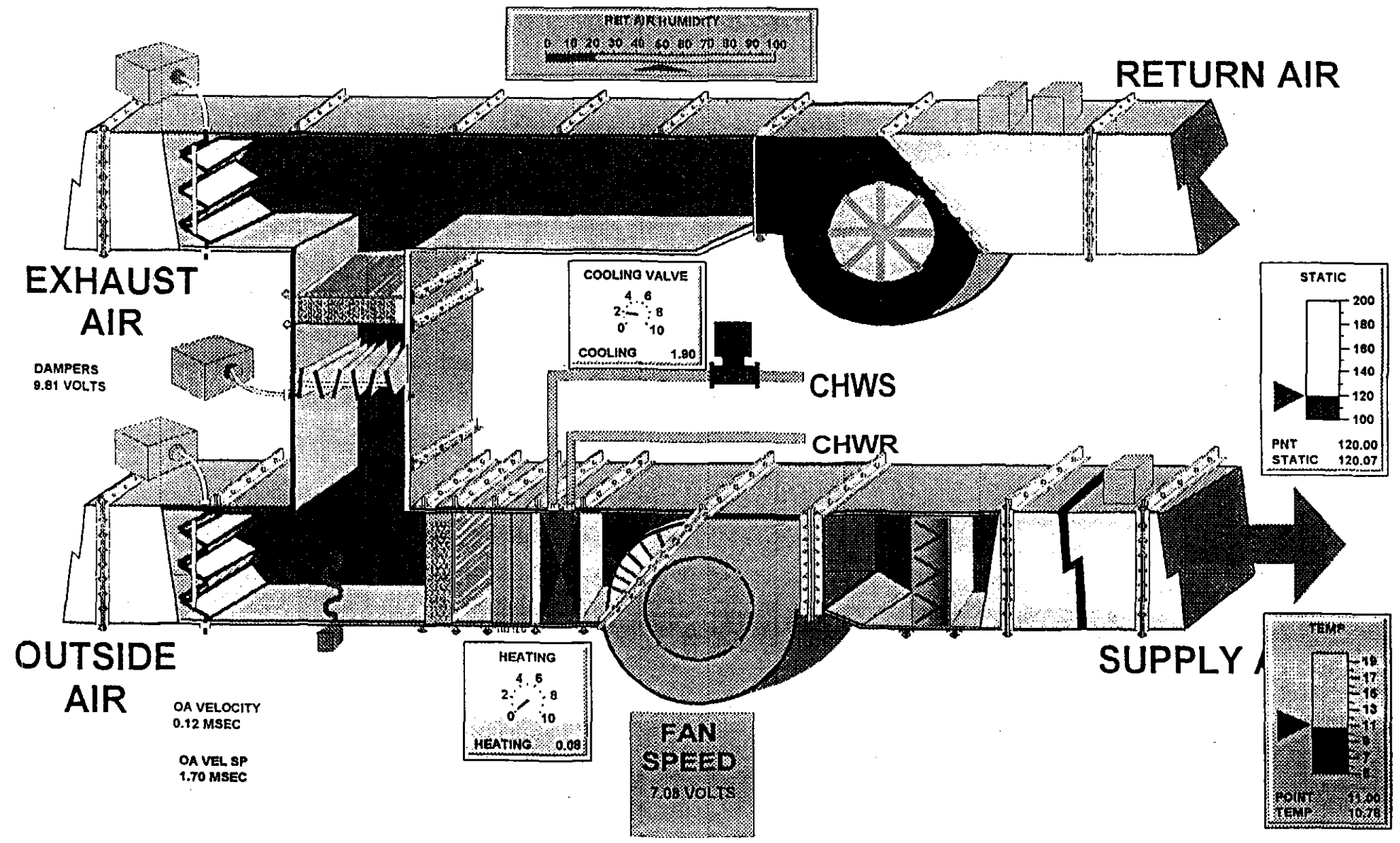
12.4 °C

DATA 1ST FLOOR CHILLERS



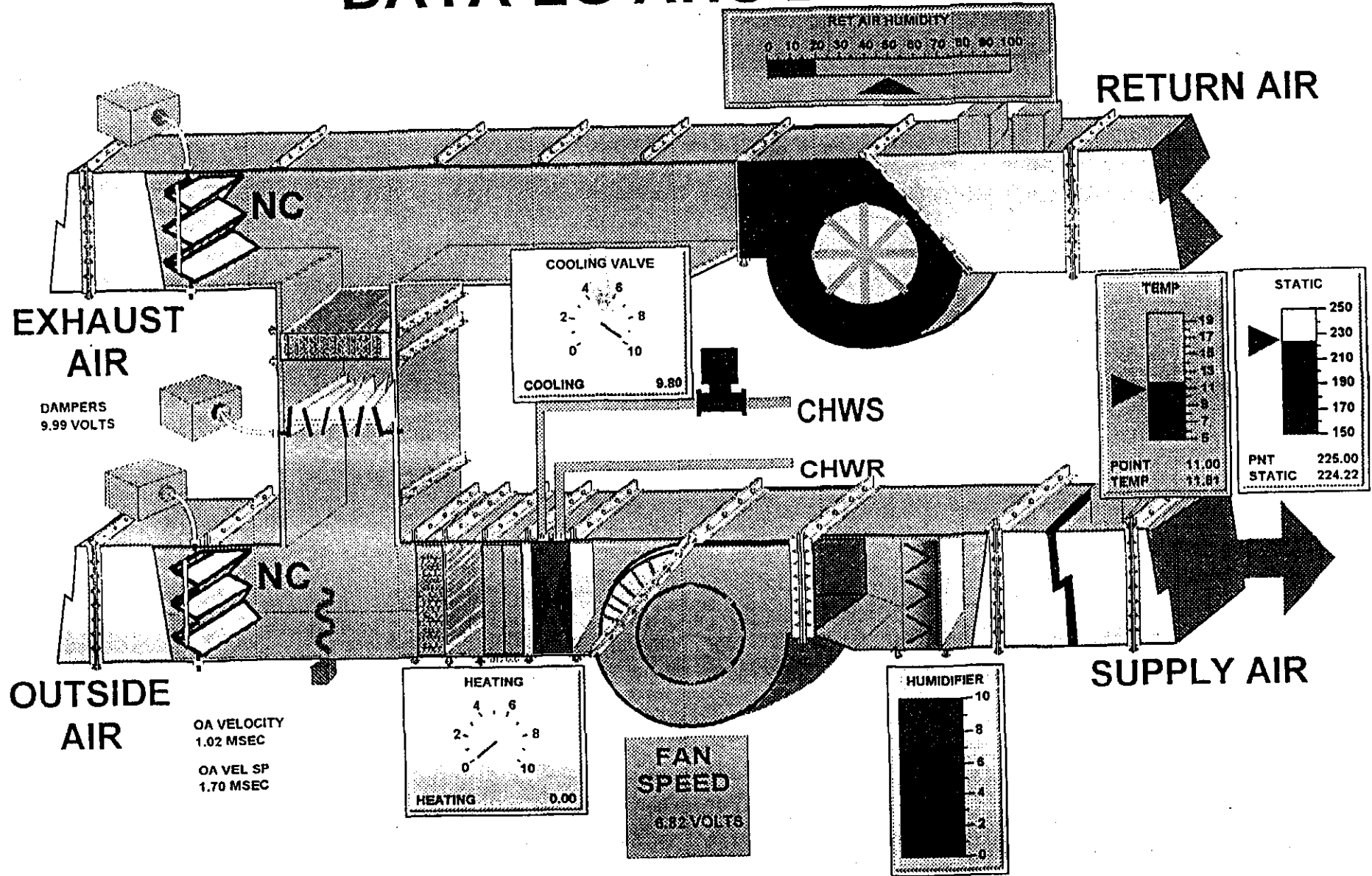
DATA LG AHU 1

195



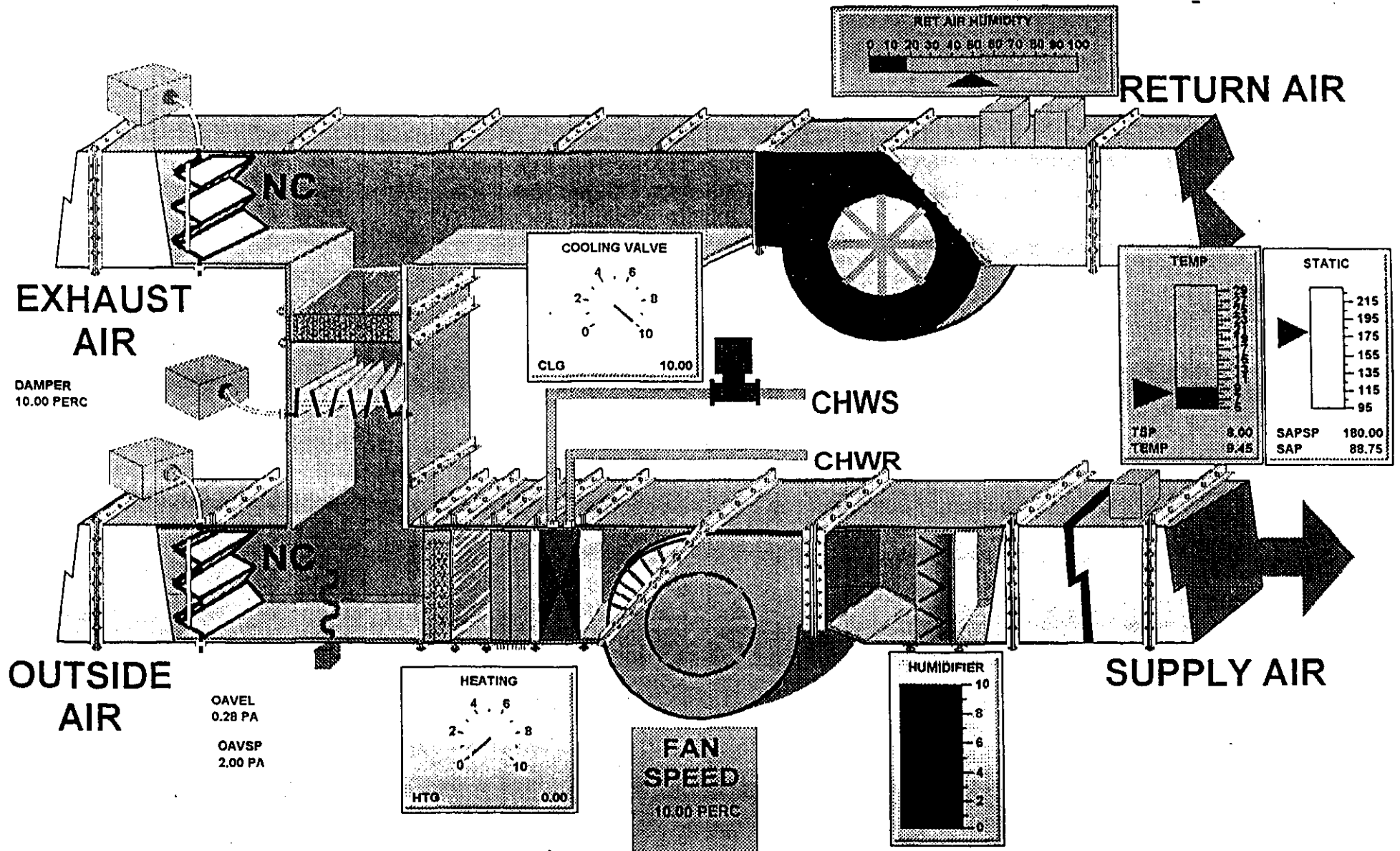
DATA LG AHU 2

196



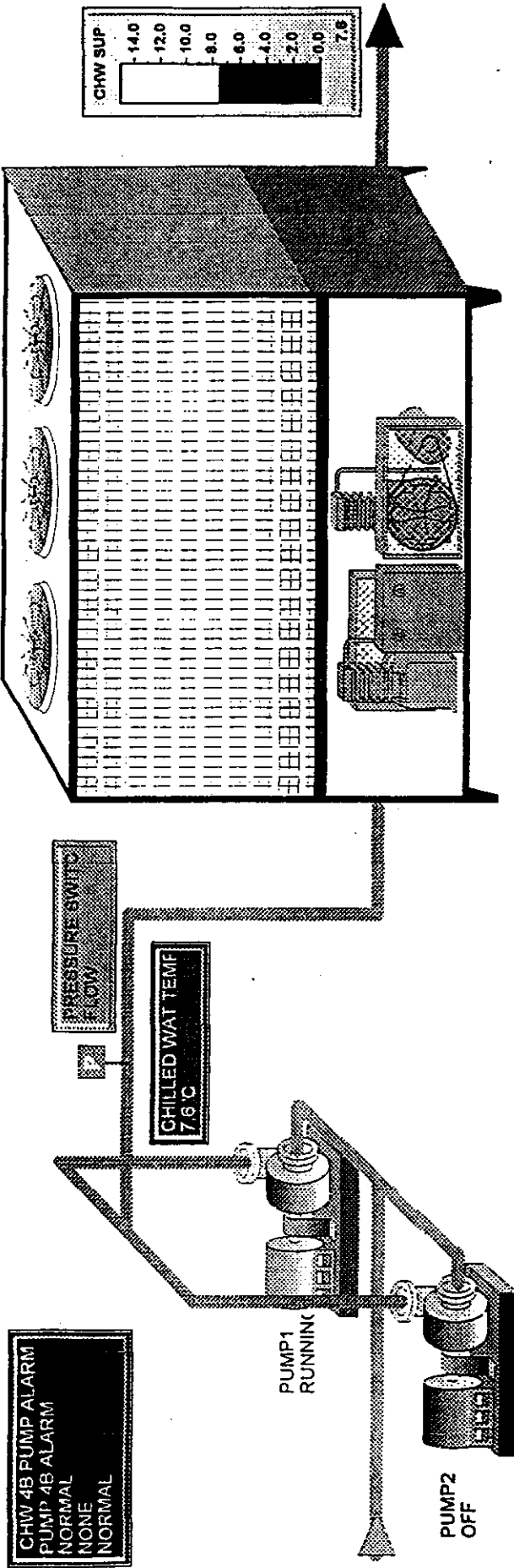
DATA LG AHU 3

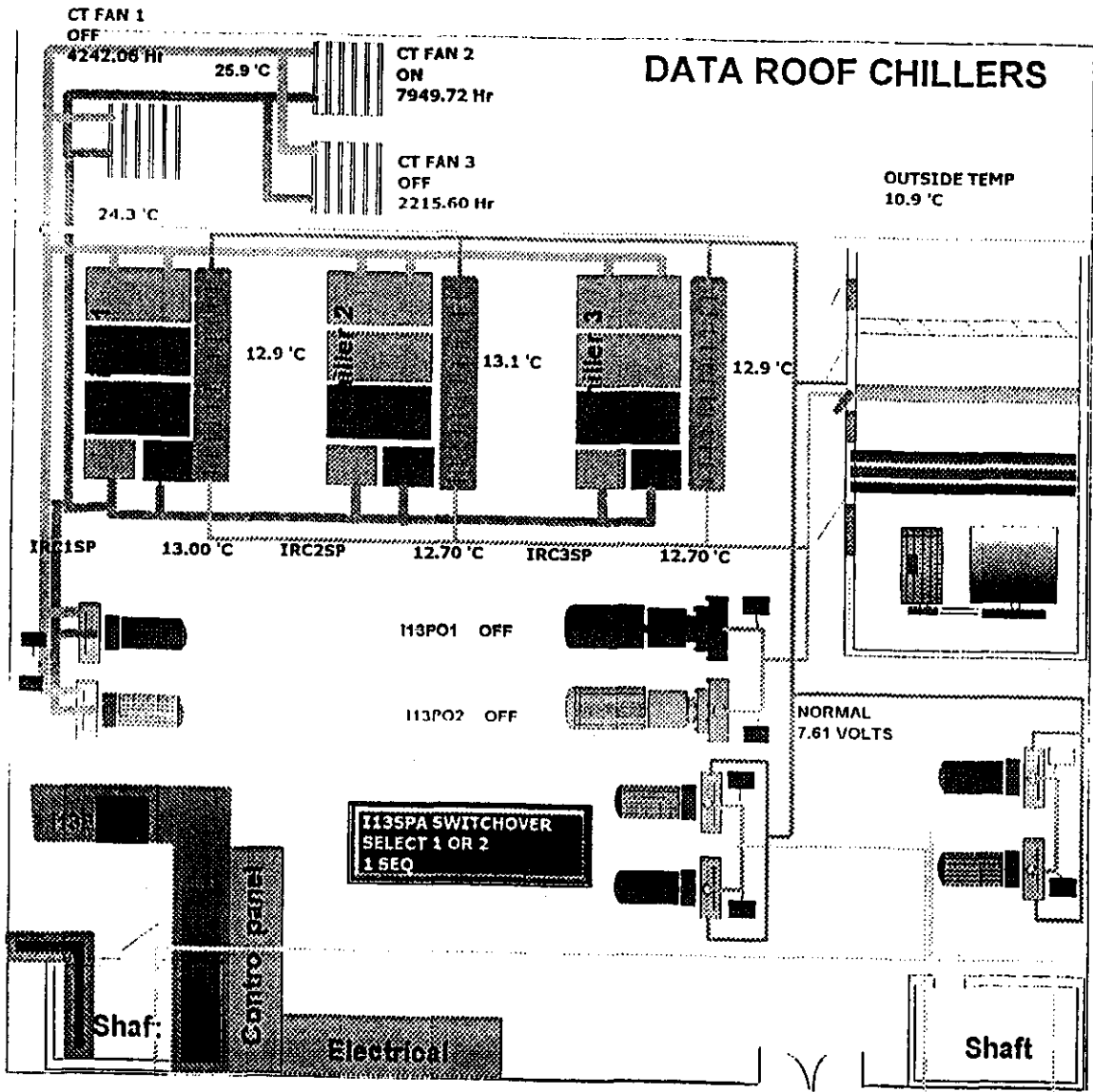
197



PRINT CHILLERS

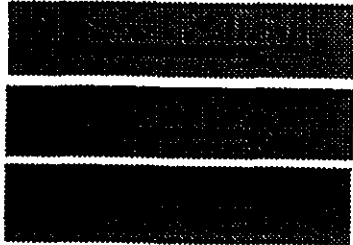
PRINT CHILLERS





I13CT SWITCHOVER
SELECT 1,2 OR 3
2 SEQ

I13CP SWITCHOVER
SELECT 1 OR 2
2 SEQ



10	15	20	25	30	35
I13CSP					28.00
I13CWR					25.9

SECD 6
9999.00 SEC

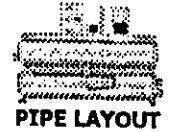
ANK LOW LEVEL
LOW LEVEL
NORMAL
NONE
OFF

I13PP SWITCHOVER
SELECT 1 OR 2
2 SEQ

I13HTG
HEATING
NORMAL
NONE
100.0 %



data_main2



PIPE LAYOUT



I13SPB SWITCHOVER
SELECT 3 OR 4
3 SEQ

13.0 °C

12.3 °C

DATA GROUND FLOOR

CHILLER
NORMAL/CHI SET PNT
NONE 5.0 °C
ON

CHILLER
CH2 SET PNT
5.0 °C

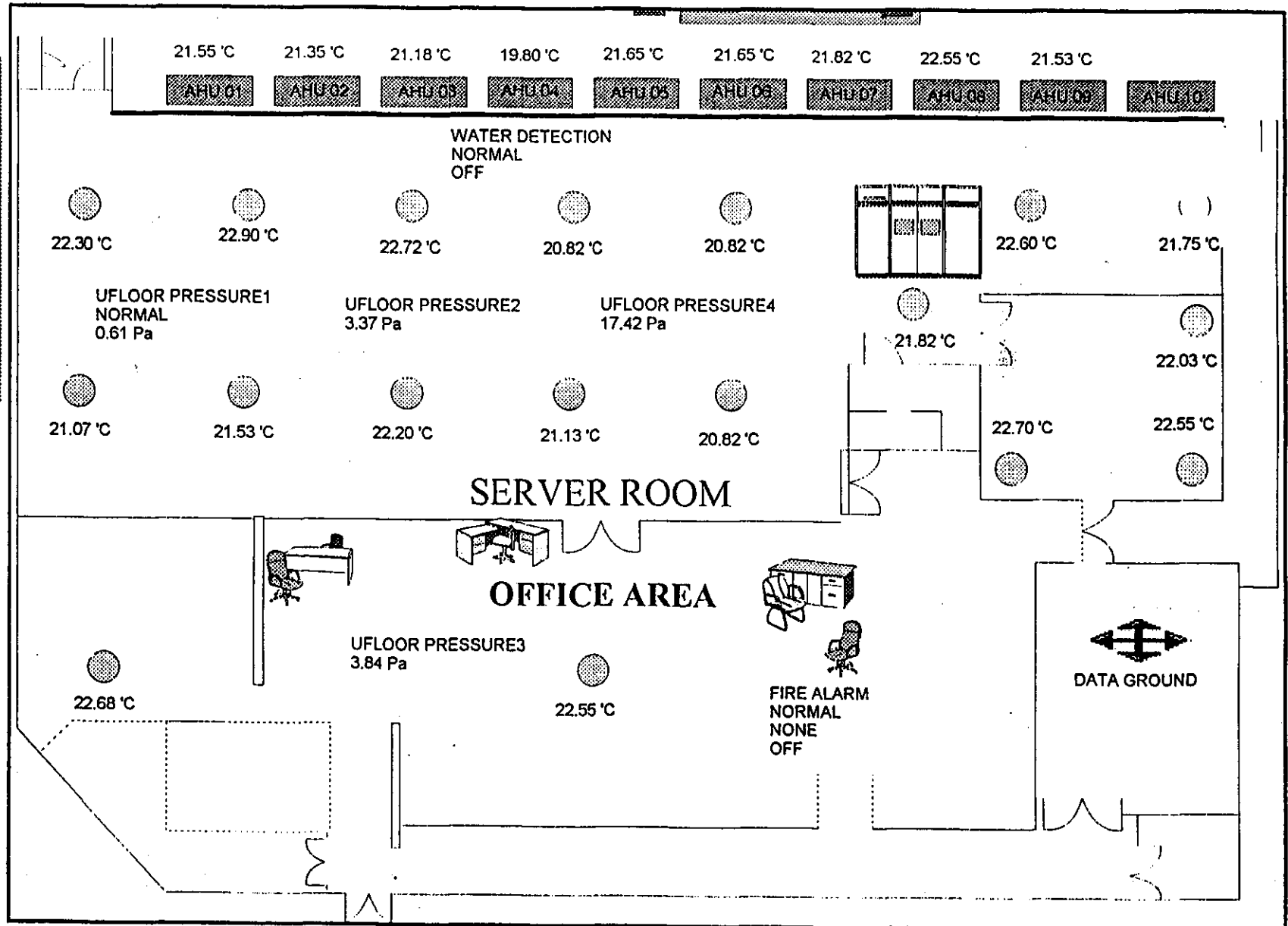
CHILLER
CH3 SET PNT
5.0 °C

CHILLER
CH3 SET PNT
5.0 °C

CHILLER
CH3 SET PNT
5.0 °C



200



APPENDIX D
KOPANANG SYSTEM SPECIFICATION AND CASE STUDY

Data for surface plant

Chilled water generator

Make	Hitachi
Model	HM - 22 A
Refrigerating duty	5983 kW
Absorbed power	1120 kW
Drive motor power	1350 kW ; 4 pole ; 6,6 kV
Capacity modulating range	100 - 20 %
Operating noise level	94 Decibel
Operating mass(Excluding motor)	60 200 kg
Impeller operating speed	78,67 Hz
Compressor oil	Castrol SW68
Refrigerant	134 A
Refrigerant charge	5500 kg
Mass of heaviest component	22500 kg

Evaporator and Condenser

Description	Evaporator	Condenser
Inlet water temperature	8.65 °C	24 °C
Outlet water temperature	3 °C	27.86 °C
Tube water velocity	2.98 m/sec.	1.89 m/sec.
Design flow rate	253 l/sec.	440 l/sec.
Design pressure drop	200 kPa	29 kPa
Evaporator/Condenser temperature	0.5 °C	31.9 °C
Evaporator/Condenser pressure	314 kPa (abs.)	784 kPa (abs)

Cooling Towers

Evaporator cooling tower

This is for two, two cell towers in series, data following being for one, two cell tower, unless noted to the contrary.

- Cooling Tower

Make	Hammon Sobelco
Design BP	87.5 kPa
Ambient Wb.	18 °C
Recirculation allowance	1.5 °C
Tower inlet wb.	19.5 °C
Tower inlet db.	28 °C
Air mass flow rate (per cell)	298.8 kg / sec.
Air volume flow rate	300 m ³ / sec. (28 / 19.5)
Air velocity in tower	2.33 m / sec.
Design water inlet temperature	27 °C
Design water outlet temperature (first stage)	21.5 °C
Calculated water outlet temperature (second stage)	20.49 °C
Water flow rate	360 l / sec.
Water loading	1.43 l / sec. / m ²

- Fan

Make	James Howden
Model	24 EFS 4
Blade material	FRP
Impeller diameter	7.3 m
Number of blades	4
Fan rotational speed	1.8 Hz
Fan total head,(sea level)	108 Pa
Fan operating blade pitch	17°

- Gearbox

Make	Hansen
Model number	RMD
Speed ratio	8.75

- Motor

Nominal power	75 kW
Power absorbed at design blade angle	42 kW
Power absorbed at max. blade angle	49.4 kW

- Pack characteristics

Pack type	Muntis C10/19, Brentwood fill
Sheet thickness	4 mm
Distance between sheets	22.5 mm
Numbers of layers of pack	3
Height of pack layer	400 mm
Pack η	- 0.64
Pack λ	1.40

- Tower design parameters

Factor of merit	0.6
KAV / L	1.93
L / G	0.62

Condenser cooling tower

- Cooling tower

This is per condenser tower.

Make	DB Thermal
Design BP	87.5 kPa
Ambient Wb.	18 °C
Recirculation allowance	0.5 °C
Tower inlet wb.	18.5 °C
Tower inlet db.	28 °C
Air mass flow rate	266.67 kg / sec.
Air volume flow rate	268 m ³ / sec. (28 / 19.5)
Air velocity in tower	1.83 m / sec.
Design water inlet temperature	27.86 °C
Design water outlet temperature	24 °C
Water side duty	7110 kW
Water flow rate	440 l / sec. / m ²
Drift loss	0.09 l / sec.
Evaporation rate	3.269 l / sec.
Bleed rate	0.82 l / sec.
Water loading	3 l / sec. / m ²

- Fan

Make	Howdan Safanco
Model number	PFT
Blade material	FRP
Impeller diameter	7.315 m
Number of blades	6
Fan rotational speed	1.083 Hz
Fan total head (sea level)	145.1 Pa
Fan operating blade pitch	13°

- Gearbox

Make	Hansen
Model number	RNE 24
Speed ratio	8.9

- Motor

Nominal power	110 kW
Power absorbed at design blade angle	56 kW
Power absorbed at max. blade angle	76.3 kW

- Pack characteristics

Pack type	Film – Big six asbestos sheet
Sheet thickness	6 mm
Distance between sheets	50 mm
Numbers of layers of pack	2
Height of pack layer	920 mm
Pack η	- 0.59
Pack λ	1.42

Bulk air-cooling towers

These values are once again the values for which the equipment was designed for, no actual values are kept on record for the specific information.

This is for the total of all three cells together, unless noted to the contrary.

Make	Hammon Sobelco
Design BP	87.5 kPa
Ambient wb.	18 °C
Recirculating allowance	0.5 °C
Tower inlet wb.	18.5 °C
Tower inlet db.	28 °C
Air mass flow rate (total)	726.9 kg / sec.
Air volume flow rates (total)	
a) (28 / 18.5)	730.6 m ³ / sec.
b) (7 / 7)	675 m ³ / sec.
c) Average	702.81 m ³ / sec.
Average velocity in towers	6.97 m / sec.
Design water inlet temperature	3.5 °C
Design outlet air temperature	7 °C
Air side duty	23696.94 kW
General loss allowance	1569 kW
Water side duty	25265.94 kW
Water flow rate (total)	750 l / sec
Water loading	2.5 l / sec. / m ²

- Pack characteristics

Pack type	Muntis C10/19
Sheet thickness	4 mm
Distance between sheets	20 mm
Numbers of layers of pack	8
Height of pack layers	400 mm
Pack η	- 0.51
Pack λ	2.21

- Fan

Make	James Howden
Model number	1200 / 2.5536 / 3064
Blade material	HA9S
Impeller diameter	3.064 m
Number of blades	8
Fan rotational speed	10.2 Hz
Fan total head	495 Pa
Fan operating blade pitch	61°

- Motor

Nominal power	250 kW
Power absorbed at 3 fan operational	177 kW

Chilled Water Dams

- Surface chilled water dam

Description	See "Appendix 6"
Number of dams	One
Capacity (Volume)	9168.255 m ³
Construction	Concrete construction as seen in appendix 6
Outside roof colour	Cream
Horizontal surface area	2037.39 m ²

- Surface hot water dam (Surge dam)

Description	Rectangular cross-section and top view
Number of dams	One rectangular cross-section and top view
Capacity (Volume)	1760.88 m ³
Construction	Concrete
Outside roof colour	No roof
Horizontal surface area	440.22 m ²

Pumps

- Condenser water pumps

Make	M & P
Model number	300 / 350 / BLE
Capacity	440 l / sec.
Head	255118 Pa
Min. allowable NPSH	74419 Pa
Impeller size	328 mm
Max. casing impeller size	381 mm
Pump speed	1480 rpm
Absorbed power (operating PT)	134 kW
Absorbed power (run out)	138 kW
Motor power	150 kW

- Evaporator water pumps

Make	M & P
Model number	250 / 300 / BLE
Capacity	253 l / sec.
Head	305041 Pa
Min. allowable NPSH	67564 Pa
Impeller size	362 mm
Max. casing impeller size	390 mm
Pump speed	1480 rpm
Absorbed power (operating PT)	95.6 kW
Absorbed power (run out)	93.25 kW
Motor power	110 kW

- Evaporator pre-cooling tower pumps

Make	M & P
Model number	350 / 400 / BLE
Capacity	360 / sec.
Head	141337 Pa
Min. allowable NPSH	34000 Pa
Impeller size	460 mm
Max. casing impeller size	546 mm
Pump speed	750 rpm
Absorbed power (operating PT)	59 kW
Absorbed power (run out)	64 kW
Motor power	75 kW

- 75 Level pump station

Description	Sulzer 32/17.5
Make and model	Sulzer 32/17.5
Impeller diameter	340 mm
Number of pumps (In parallel)	4 (Electrically driven, 2000 kW motor)
Number of stages per pump	Nine
Pump height	1127.76 m

Data for mine water pipe system

Underground dams

- Chilled water dam (38 Level)

Description	Cylindrical top view (10m dia. & 30.48m high)
Number of dams	2
Capacity (Volume)	2393.9 m ³ /dam i.e. 4787.8 m ³ total
Depth under surface	1158.24 m

- Hot water dams (38 Level)

Description	3*3 Haulage; 80*84 m
Number of dams	One
Capacity (Volume)	Unknown capacity due to mud build-up
Depth under surface	1158.24 m

- Hot water dams (75 Level)

Description	10 m Dia., 60 m High
Number of dams	Five (Two in operation)
Capacity (Volume)	4712.39 m ³ /dam
Depth under surface	2286 m

Pump stations

- 38 Level pump station

Description	Sulzer 32/17.5
Make and model	Sulzer 32/17.5
Impeller diameter	340 mm
Number of pumps (In parallel)	5 (3 Electrical & 2 Turbine driven)
Number of stages per pump	Nine
Pump height	1158.24 m

- 75 Level pump station

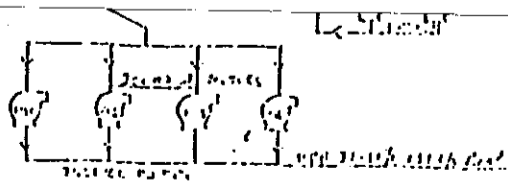
Description	Sulzer 32/17.5
Make and model	Sulzer 32/17.5
Impeller diameter	340 mm
Number of pumps (In parallel)	4 (Electrically driven, 2000 kW motor)
Number of stages per pump	Nine
Pump height	1127.76 m

Pelton wheel

This is for one turbine only.

Description	Sulzer turbine
Water flow	Design: 175 l/s Actual: 210 l/s
Water height	1158.24
Power generation	1732 kW

209



210

