

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Low frequency, in-situ vibrating sample magnetometer: Electrical systems and control software design

A thesis submitted to the University of Cape Town
for the degree of Master of Science in Engineering

Simon Gilbert Daneel Kelly

Supervised by: Dr. M. Claeys, Dr. A. Wilkinson

February, 2006

Department of Electrical Engineering
University of Cape Town



Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

Signature of Author

Cape Town

February 2006

Abstract

A low frequency vibrating sample magnetometer has been built to measure the in-situ properties of ferromagnetic catalysts. The instrument allows measurements to be taken during an experimental catalyst test run (in-situ). The vibration is performed by a motor crank arrangement at a constant frequency of 2 Hz. The software designed to control the instrument and the reaction was written in LabView which enabled a rapid prototyping approach. This thesis focuses on the software and electrical systems of the setup. Results of research conducted using this system are published separately however this document shows the relationship between the magnetic saturation and remnance and the mass of ferromagnetic material present in the reference material as well as the effect of temperature on this material.

“In the beginning was the Word, and the Word was with God, and the Word was God. He was with God in the beginning. Through him all things were made; without him nothing was made that has been made. In him was life, and that life was the light of men. The light shines in the darkness, but the darkness has not understood it.

There came a man who was sent from God; his name was John. He came as a witness to testify concerning that light, so that through him all men might believe. He himself was not the light; he came only as a witness to the light. The true light that gives light to every man was coming into the world.

He was in the world, and though the world was made through him, the world did not recognize him. He came to that which was his own, but his own did not receive him. Yet to all who received him, to those who believed in his name, he gave the right to become children of God - children born not of natural descent, nor of human decision or a husband's will, but born of God.

The Word became flesh and made his dwelling among us. We have seen his glory, the glory of the One and Only, who came from the Father, full of grace and truth.”

John1:1-14

Acknowledgements

I am grateful to many people for help, both direct and indirect, which I have received during the past two years. To Dr. Michael Claeys who leads the project and who supervised this thesis, thank you for all your help and for being so lenient with your deadlines! Thanks to Andrew Wilkinson for co-supervising this project and for the time and effort you invested. Thanks to the Catalysis Research unit of the Department of Chemical Engineering at UCT for the opportunity to work on this project and for providing funding to keep me going. Bill Randall, thanks for your help with the hardware.

To all the postgraduates in the Cat group, its been great to work with you all, thanks for the endless entertainment.

Huge thanks to my church family, I love you guys.

Lastly, thank you Jesus for the purpose you have given to my life, may your name be glorified in all the earth!

University of Cape Town

Contents

Declaration	i
Abstract	iii
Acknowledgements	vii
Table of Contents	viii
List of Figures	xix
Nomenclature	xxi
1 Introduction	1
1.1 Project Background	2
1.2 Description of Task	2
1.3 Principal Goals	3
1.4 Software Development	5
I Reactor Rig	7
2 Reactor Rig	9
2.1 Overview	10

2.2	Instrumentation and Control Requirements	11
2.3	Reactor Heating	12
2.3.1	Heating Tape	12
2.3.2	Oil Heating	13
2.3.3	Infra Red Heating	14
2.4	Mass Flow Controllers	14
2.5	Hardware Diagrams	15
3	Rig Software Design	19
3.1	Overview	20
3.2	Specifications	20
3.3	Design Approach	20
3.4	System Level Design	22
3.4.1	Overview	22
3.4.2	Module Communication and Data Transfer	22
3.4.3	Core Functions	27
3.4.4	Alarms, Events and Data Logging	28
3.5	Module Design	29
3.5.1	Overview	29
3.5.2	Analog Input and Counter Output	31
3.5.3	System Overview	33
3.5.4	General Temperature Control	33
3.5.5	Advanced Temperature Control	35
3.5.6	Water Bath	37

<i>CONTENTS</i>	xi
3.5.7 HPLC Pump	38
3.5.8 Mass Flow Controllers	39
3.5.9 Data Overview	40
3.6 Additional Software	42
3.6.1 Temperature Profile Editor	42
4 Results	45
4.1 Temperature Control Performance	46
4.2 Serial Communication	46
II Vibrating Sample Magnetometer	49
5 VSM Background	51
5.1 Basic Concept	52
5.2 Common Design Features	52
5.2.1 Vibrating Mechanisms	53
5.2.2 Detection Coil Systems	53
5.2.3 Electronic Systems	54
6 VSM Hardware Design	57
6.1 Features and Specifications	58
6.2 Constraints	59
6.3 Physical Design	61
6.3.1 Coil design, Construction and Simulation	61
6.3.2 Sample Movement	66
6.3.3 Coil Cooling	68

6.4	Electronics	68
6.4.1	Amplification	68
6.4.2	Filtering	69
6.4.3	Synchronization with Sample Movement	69
7	VSM Software Design	75
7.1	Overview	76
7.2	Specification	76
7.3	Design Process	77
7.4	Software Layout	77
7.5	Magnet Control	78
7.6	Data Acquisition and Processing	81
7.7	Configurable Measurement Sequences	83
7.8	Interface Design	87
7.9	Error Handling	88
7.10	Additional Software	89
7.10.1	Advanced VSM Data Editor	89
7.10.2	Measurement Recipe Editor	89
8	Results and Analysis	91
8.1	Signal Analysis	92
8.2	Calibration	92
8.2.1	Temperature Effects	93
8.3	Noise	93
8.4	Measurement Errors	96

<i>CONTENTS</i>	xiii
9 Conclusions	99
10 Recommendations for Future Work	101
A Software Source Code	105
B Magnet Photographs	117
C Rig Photographs	121
D Coil Winder Photographs	127
E Hardware Diagrams	133
Bibliography	136

University of Cape Town

List of Figures

2.1	Rig flow chart	10
2.2	Temperature control hardware.	16
2.3	Serial communication connections	17
3.1	System overview	22
3.2	Communication via global variables	23
3.3	Communication via message queues	24
3.4	Implementation of queue handling	25
3.5	Queue data type and possible use	26
3.6	Database communication	26
3.7	Queue manager structure.	28
3.8	Module overview	30
3.9	User interface event loop flow diagram	31
3.10	Application loop flow diagram	32
3.11	System overview interface	34
3.12	Interface layout	34
3.13	General temperature control interface	35
3.14	Reactor temperature control interface	36
3.15	Water Bath interface	38

3.16 HPLC pump module interface	39
3.17 MFC control dialog	40
3.18 Data overview interface	41
3.19 Temperature Profile Editor user interface	43
4.1 Reactor temperature control	46
6.1 Variation of magnetic field over pole face. Field set to maximum for a pole gap of 50 mm.	60
6.2 Magnet pole face showing dimensions and area of uniform field.	60
6.3 Dipole flux through single turn coil	61
6.4 Flux vs Displacement	62
6.5 Flux through two coils connected in series	62
6.6 Final coil arrangement and wiring.	63
6.7 Coil dimensions	64
6.8 Simulation arrangement	65
6.9 Simulated signals	71
6.10 Reactor movement	72
6.11 Physical layout side view (sliced through center).	72
6.12 Physical layout top view	73
6.13 Signal Conditioning	73
7.1 VSM top level diagram	78
7.2 VSM flow example	79
7.3 Field control smoothing	81
7.4 Field control comparison, the effect of smoothing.	82

7.5	The effects of windowing and padding	83
7.6	Typical recording sequence	84
7.7	Recipe flow	87
7.8	VSM module user interface	88
8.1	Recorded signal from the pickup coils.	92
8.2	Mass calibration hysteresis curves	93
8.3	Saturation and remnance vs magnetization	94
8.4	Temperature effect on the saturation and remnant magnetization.	95
8.5	Noise due to IR heating	95
8.6	Magnet noise	96
8.7	The effect of low pass filtering an un-amplified signal.	96
8.8	Hysteresis error analysis	97
A.1	Functional global implementation	105
A.2	Simplified queue manager code	106
A.3	Simplified interface loop code	106
A.4	Simplified application loop code	107
A.5	Simplified AI loop code	108
A.6	Simplified CO loop code	109
A.7	Basic PID code	110
A.8	Simplified code for advanced temperature control	111
A.9	Water bath code portion	112
A.10	HPLC functions example usage.	113

A.11 HPLC module code portion.	114
A.12 HPLC <i>set flow</i> function code.	114
A.13 MFC methods.	115
A.14 MFC module code portion: read MFC's flow.	115
A.15 MFC method: write set point.	116
B.1 Bruker B-E25V magnet, front view.	118
B.2 Bruker B-E25V magnet, oblique view.	119
B.3 Bruker B-E25V magnet showing the tapered poles. The pole face diameter is 15 cm.	120
C.1 Front view of the rig.	121
C.2 Side view of the rig.	122
C.3 Front top view of the rig showing the positioning of the reactor in the center of the magnet.	123
C.4 Close up of the reactor showing the coil cooling, coil con- nections, reactor and infra-red heating.	124
C.5 The mass flow controllers mounted on an aluminium bracket which is attached to the magnet.	125
C.6 The electronics for the rig are housed on top of the magnet power supply.	125
D.1 Top view of coil winder.	127
D.2 Side view of coil winder.	128
D.3 Front view showing the coil bobbin attached to the motor axle.	129
D.4 Wound coil attached to the motor shaft.	130
D.5 A wound coil in comparison to a South African 50 cent coin and a eight pin dip chip.	130

D.6 Motor with notched disk attached to shaft. 131

E.1 Schematic of the circuit used to drive the motor from a set point generated by the computer. This circuit is part of the winder built to wind the pick-up coils. 133

E.2 Filter schematic 134

E.3 Power supply connections 135

University of Cape Town

University of Cape Town

Nomenclature

- AI Analog Input, page 31
- CO Counter Output, page 32
- DAQ Data Acquisition, page 15
- DAQ Data aquisition, page 31
- DDE Dynamic data exchange, page 39
- DSC Data-logging and supervisory control, page 28
- EMF Electro motive force, measured in Volts, page 52
- HPLC High performance liquid chromatography, page 11
- I/O Input / Output, page 14
- MFC Mass flow controller. Controls the flow of gas based on the mass of the gas, page 10
- PID Proportional, integral and differential control, page 35
- PSD Power spectral density., page 82
- PWM Pulse width modulation, used to control the power output of the electrical heaters, page 12
- SNR Signal to noise ratio, page 69
- SSR Solid state relay, page 15
- VSM Vibrating sample magnetometer, page 3

Chapter 1

Introduction



Engineering problems are under-defined, there are many solutions, good, bad and indifferent. The art is to arrive at a good solution. This is a creative activity, involving imagination, intuition and deliberate choice.

- *Ove Arup*

1.1 Project Background

The purpose of the project, for which this thesis forms a part, is primarily to monitor the state of catalysts during a reaction, for the purpose of investigating different catalyst formulations, and secondly to investigate the effect of magnetic fields on catalyst performance.

In-situ characterisation of catalysts has become increasingly important. It is however not easy to measure or monitor physical or chemical changes in a working catalyst. The reasons for this are that the catalyst is inside a reactor and therefore not visible and neither can it be removed or sampled without interfering with the reaction. It is therefore necessary to find some property of the catalyst that can be measured without disturbing or interfering with the reaction. This property also needs to be able to provide the necessary information about the catalyst. The magnetic properties of the sample of interest fulfil these requirements since they can be measured without direct interaction with the sample and from them the necessary information about the sample can be deduced.

By performing a hysteresis measurement on the catalyst (generating a BH curve for the catalyst) inside the reactor the following properties can be measured:

- ❖ Magnetic saturation.
- ❖ Magnetic remnance.
- ❖ Coercive field.

These properties allow the calculation of other useful properties of the catalyst such as the amount of metal phase present (i.e. the degree of reduction) and the amount of ferromagnetic material present.

1.2 Description of Task

The portion of the project¹ described in this document may be divided into two sections.

¹From this point on the term *project* will refer only to that portion of the overall project that is described in this thesis.

Part I: deals with the reactor rig

The term *reactor rig* is used to describe the physical apparatus used to perform the reaction. It consists of gas supplies, regulators, gas controllers, pipes, a reactor, heating equipment, valves, pumps etc.

In order to reduce the amount of supervision required to operate the rig and to improve the rig's efficiency it is desirable to automate as much of the operation as possible. This can be done either in hardware, using commercially available temperature controllers etc. or in software, using a computer to perform the control. Since hardware systems are very expensive, require a fair amount of operator attention and are not capable of controlling all aspects of the system a preferred way of operating would be to use a computer to provide the necessary control. Hardware systems may be added to the software control to provide redundancy and improved safety.

With a single computer it is possible to control the entire rig including temperature and gas flow, other hardware via serial communication and to provide the operator with advanced displays of the past and present state of the rig such as temperature graphs.

Part II: the vibrating sample magnetometer

This portion of the project is concerned with the design and construction of the *vibrating sample magnetometer* (VSM) which is the instrument used to perform the hysteresis measurements on the catalyst. Once again, the control of the VSM is performed by the same computer. A vibrating sample magnetometer measures the magnetic properties of materials. When a material is placed within a uniform magnetic field and is mechanically vibrated, there is some magnetic flux change. This induces a voltage in the pick-up coils, which is proportional to the magnetic moment of the sample. The uniform magnetic field is generated by an electromagnet.

1.3 Principal Goals

The goals of this project can again be divided into those that apply to the rig and those that apply to the VSM.

Reactor Rig

❖ Hardware goals

1. Construct the necessary hardware required to provide temperature control.
2. Provide connectivity between the computer and the reactor rig systems such as temperature probes, pressure sensors, pumps etc.

❖ Software goals

1. Provide effective temperature control.
2. Provide control of other hardware e.g. pumps.
3. Provide a seamless, easy to use interface.
4. Incorporate data-logging to enable process variable trending.
5. Provide a single point of control for the entire system.

VSM

❖ Hardware goals

1. Manufacture the pickup coils required to measure the magnetic field produced by the samples.
2. Design the electrical system to optimise the raw signal from the coils.

❖ Software goals

1. Design and implement software to operate the VSM system.
2. Integrate the VSM control interface with the reactor rig software interface.
3. Calibrate the system and verify the dependency of the sample magnetization on the sample mass.

1.4 Software Development

It was decided to use National Instruments LabView to as the language in which to develop the software systems.

LabView is a graphical programming language from National Instruments. Included in the LabView distribution is an extensive development environment with many libraries and tools. The graphical language is named 'G'. Originally released for the Apple Macintosh in 1986, LabView is used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various flavors of UNIX, Linux, and Mac OS. [1]

Part I

Reactor Rig

University of Cape Town

Chapter 2

Reactor Rig



The great tragedy of science - the slaying of a beautiful hypothesis by
an ugly fact.

- *T H Huxley*

2.1 Overview

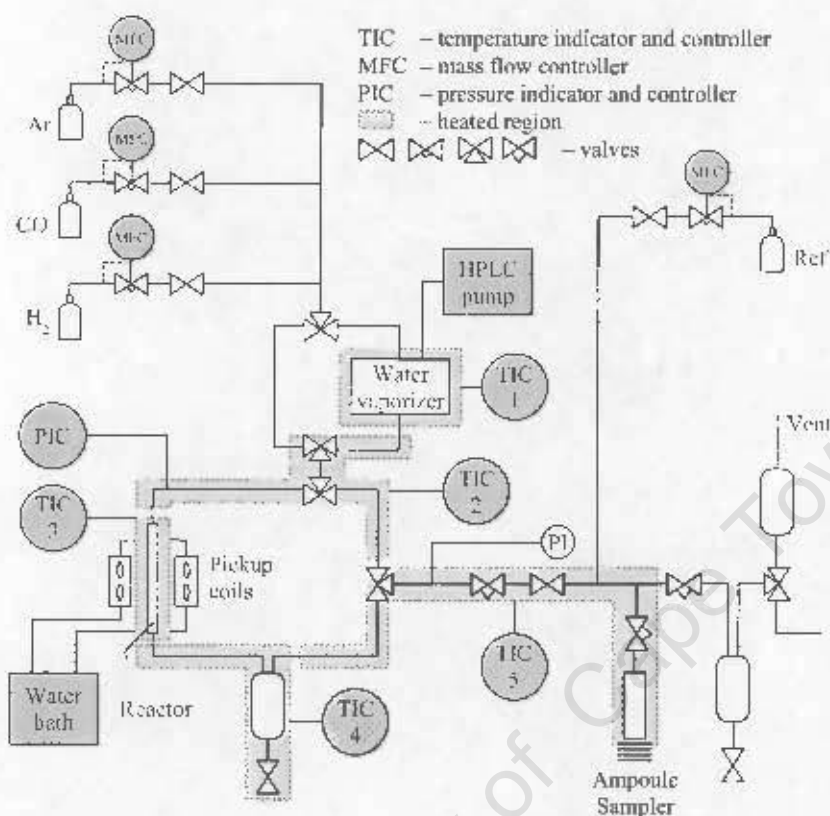


Figure 2.1: Rig flow chart

This chapter deals with the reactor rig and describes the physical aspects of the rig design making particular mention of features that are relevant to the electrical and software design.

Fig. 2.1 shows a simplified diagram of the rig layout. The rig consists of a fixed bed reactor¹ and bypass that are fed by a heated mixture of gasses. There are four controlled gas flows labeled as "MFC" (Mass Flow Controller) on the diagram which supply the gas via a network of heated piping. The five heated zones are shown on the diagram as shaded regions with dotted borders and are labeled TIC 1 to TIC 5. Throughout the rest of the document the following names will be used to describe these heated regions:

¹A fixed bed reactor uses supports for the catalyst and sample which do not move and are therefore 'fixed' as a bed in the reactor.

- ❖ TIC 1 - Water vaporizer
- ❖ TIC 2 - Feed (line)
- ❖ TIC 3 - Reactor
- ❖ TIC 4 - Hot trap
- ❖ TIC 5 - Exit (line)

Other items of interest are the HPLC (high performance liquid chromatography) pump and the Water bath with which the rig software communicates and controls. The reactor itself is a 40 cm, half inch stainless steel pipe. The catalyst is placed on a porous brass frit in the center of the reactor².

2.2 Instrumentation and Control Requirements

With reference to Fig. 2.1 the instrumentation requirements for the system are listed below. The control specifications are based on the requirements of the chemical process.

- ❖ There are four heated zones (Water Vaporizer, Feed, Hot trap, Exit) which require basic PID control (see Sec. 3.5.4 - General Temperature Control for more details). Each zone requires one thermocouple to measure the temperature. In order to enable the computer to control the temperature, one analog input and one counter output is required for each zone. The control requirements for this zone are a temperature range of 0 - 300°C with a stability of $\pm 5^\circ\text{C}$.
- ❖ The heating of the reactor requires more advanced control. (see Sec. 3.5.5 - Advanced Temperature Control). A PT100 temperature probe was used to measure the temperature which again requires one analog input and one counter output to provide computer control. The required temperature range is 0 - 450°C $\pm 0.1^\circ\text{C}$ stability.

²The brass frit is magnetically dead and therefore does not introduce any background signal (cf. Sec. 6.2)

- ❖ There are four mass flow controllers (MFC) which may be communicated with via serial communication (RS-232) (see Sec. 3.5.8 - Mass Flow Controllers).
- ❖ The pressure reading requires a single analog input in order to allow the computer to read it.
- ❖ The water bath, which is used to cool the VSM pick-up coils, is controlled via serial communication with the computer. (see Sec. 3.5.6 - Water Bath)
- ❖ The HPLC pump is likewise controlled via serial communication. (see Sec. 3.5.7 - HPLC Pump)

The main elements of these requirements are discussed in the following sections.

2.3 Reactor Heating

Of the temperature controllers, the heating of the reactor is the most important and also the most difficult. The difficulty is due to the strict heating requirements, the space limitations and also the need to minimise any electric or magnetic field produced by the heating since this will induce a signal in the pick-up coils of the VSM system and hence interfere with the signal induced by the sample.

Three different options for heating the reactor are briefly discussed here of which only the last two were implemented.

2.3.1 Heating Tape

A heating tape is a flexible heating element encased in a glass fibre insulating sheath and is the most common method of heating used in rigs of this size. The tape is wrapped around the area to be heated. The area's temperature is measured using a J type thermocouple and controlled by varying the power output of the heating tape. The power control is performed by pulse width modulation (PWM) of the heating tape's power supply.

The obvious problem with this method is that current passing through the tape would produce an electric field which due to its proximity to the coils would interfere with the signal induced in the coils by the sample. With very careful winding of the tape around the reactor it should be possible to reduce the field to almost zero by symmetrically winding the tape up and down the reactor but this would never be perfect and may require many winding attempts to achieve success. A related problem is the introduction of ferromagnetic material which would interfere with the signal from the catalyst (see Sec. 6.2 for further details). Since the heating tape would likely be ferromagnetic this would pose a serious problem.

A further problem is that the entire reactor may need to be insulated with glass wool in order to prevent heat loss. This is a problem due to the small amount of space available between the poles of the magnet (described further in Sec. 6.2).

Advantages of this method are that the full range of temperatures (up to 425°C) can be reached and that a very even temperature profile in the reactor can be achieved.

2.3.2 Oil Heating

Using hot oil to heat the reactor was considered because it does not produce any electric fields near the coils. It does however require complex physical design in order to heat the reactor without introducing large amounts of magnetic material near the sample which would interfere with the signal from the sample. The design also makes the reactor bulky and heavy since there is additional metal needed to pass the oil around the reactor, the oil itself, and the pipes supplying the oil to the reactor.

Unlike the previous method, the oil can only heat to approximately 300°C which means that an additional external heater would be required in order to reach the higher temperatures required for pre-treatment steps of the catalyst. This makes the reaction sequences more complex since the reactor would have to be moved in and out of this external heater during a reaction.

Although this method of heating was implemented it was superseded by the next method.

2.3.3 Infra Red Heating

This method uses two infra red heaters placed on either side of the reactor and outside the poles of the magnet so as to minimise the field that they experience. The field produced by the current in the heating elements is minimised by ensuring they are as far from the coils as possible. The heating elements are able to rapidly heat the reactor to well above 425°C. Cooling is also rapid due to the lack of insulation on the reactor.

Disadvantages of this system are that the mounting of the heaters is difficult due to the lack of space. Also, because of the nature of the heating it is necessary to shield the coils in order to reduce the direct heat that they would be exposed to. The temperature profile produced in the reactor by this method is not very good but could be improved by using longer heaters or by enclosing the reactor and the heaters to prevent air movement. The effect of the IR heaters on the signal from the coils is discussed in Sec. 6.4.2.

2.4 Mass Flow Controllers

The mass flow controllers (MFC's) are electrically operated valves used to control and measure the flow of gas. On this rig Brooks Instruments MFC's were used. The name MFC is derived from the method used to measure the flow of the gas through the instrument. There are three possible ways of communicating / controlling the mass flow controllers, via analog signals to each controller, individual serial communication with each controller or with the aid of a power supply provided by the manufacturers.

Using analog signals sent to and read from each individual controller, although simple, requires one analog input and one analog output for each MFC. Having four MFC's would therefore require four inputs and four outputs. Since analog I/O ports are expensive and since this method may have complications involving noise, signal conditioning, the actual control of the MFC's and also providing the MFC's with power, it was decided not to use this method.

Communicating with the controllers directly via the computer's serial ports is a possibility but would only be viable using the RS-485 standard

since this would only require one serial port. RS-232 requires a dedicated port for each communication channel and so would not be viable since the serial ports available on the computer are needed for other purposes. An additional serial card could be installed in the computer but since this method would involve the development of drivers which would take a significant amount of time, and require information regarding the communication specifications of the MFC's which is not readily available, this method was used.

Using a power supply unit (BC 154) purchased from the MFC manufacturer to communicate with the MFC's provides the computer with control over all four MFC's via a single communication channel to the power supply. The drivers needed to perform this communication were provided with the unit. Since the power supply unit provides stand alone control of the MFC's, redundancy is also introduced in the system which protects against a failure of the computer or software. The communication with, and the control of the MFC's is described in Sec. 3.5.8.

2.5 Hardware Diagrams

The following diagrams show how the hardware is arranged and the connections between the computer and the hardware.

Fig. 2.2 shows the hardware that is used to allow the computer to control the five different temperature zones. The power to each heating element is controlled by solid state relays (SSR) which are switched on and off by a digital signal from the computer. Each SSR also has a switch to allow the operator to manually switch off the power to that SSR. The purpose of these switches is to avoid accidental heating during maintenance or other similar situations.

The computer uses a National Instruments PCI-6024E DAQ (data acquisition) card which features 16 channels of analog input, two channels of analog output, a 68-pin connector and eight lines of digital I/O. This card performs all the analog input and output required by the software. The control of the SSR's is performed by the counter outputs of a National Instruments 6602 device. The 660x devices are timing and digital I/O boards for use with the PCI bus in PC-compatible computers, or PXI or

compactPCI chassis. The 6601 devices offer four 32-bit counter channels and up to 32 lines of individually configurable, TTL/CMOS-compatible digital I/O. The 6602 devices offer this capability plus four additional 32-bit counter channels.³

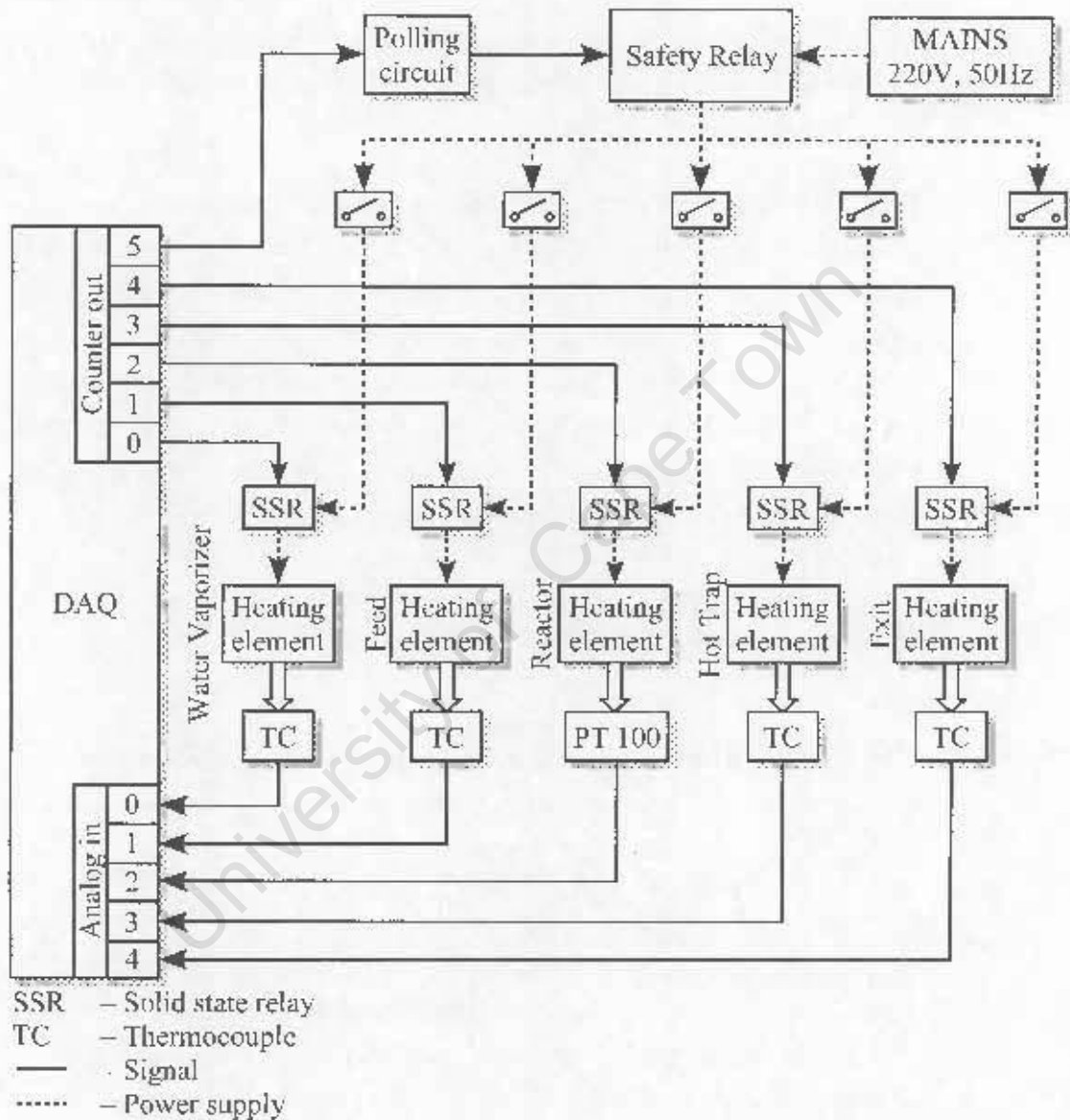


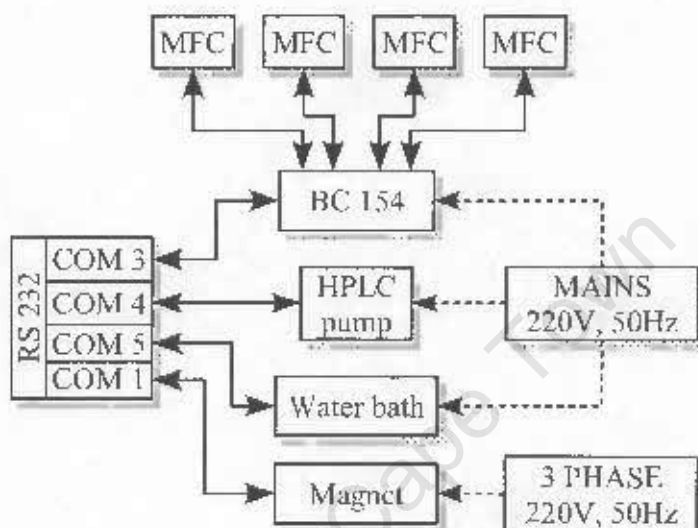
Figure 2.2: Temperature control hardware.

The function of the polling circuit is to operate the safety relay. The computer sends out a periodic pulse to indicate that it is still operating correctly. If the polling circuit stops receiving this pulse it switches the safety relay off thereby cutting power to the SSR's and preventing any

³Specifications taken for the manuals of the respective DAQ cards.

further heating from taking place. Once tripped, this relay has to be manually reset by an operator.

Fig. 2.3 shows how the various devices that are communicated with via the computer's serial ports are connected. COM ports 3, 4 and 5 are provided by a National Instruments Serial add-on board.



MFC Mass flow controller

Figure 2.3. Serial communication connections

Additional images of the rig can be found in Appendix C.

Chapter 3

Rig Software Design



The first 90 percent of the code accounts for the first 90 percent of the development time...The remaining 10 percent of the code accounts for the other 90 percent of the development time.

- Tom Cargill

3.1 Overview

This chapter deals with the design and implementation of the control and monitoring software for the reactor rig. The main emphasis is on the underlying structure of the software and on the design methods used to implement it.

3.2 Specifications

The responsibilities / specifications of the software are as follows:

- ❖ to provide a graphical user interface that displays relevant, real time information regarding the state of the rig.
- ❖ allow control of all aspects of the system via the same interface.
- ❖ the interface should be easy to use and not require any special training to operate.
- ❖ implement safety features that include alarms and automatic shut-down under hazardous conditions.
- ❖ provide data-logging of important variables to allow trending and activity recording.
- ❖ the software should be designed to be efficient and stable to allow for long running periods.

3.3 Design Approach

Spiral model

By using the spiral model [32] as the design method where successive versions of a module / function / interface are improved by assessing the previous version it was possible to start the project with very little knowledge of the final system and the capabilities of LabView. This approach allowed the capabilities of LabView and LabView design technique to be learned concurrently to the development of the software and also for the

detailed specifications of the system to develop along with the software without creating the need for large modifications in the software.

Bottom up design

By approaching the design from the bottom up, that is implementing the low level functionality such as analog input before higher level such as PID control, the core functionality of the software (e.g. temperature control) could be implemented and refined without knowledge or implementation of the higher level systems (e.g. module communication). This worked well in conjunction with the spiral model allowing the system specifications to develop as the understanding of the system grew.

Code reuse

As much as possible, code was designed in a manner that allowed and encouraged code reuse. This was achieved by making commonly used functionality available throughout the software in global functions. Templates were also used to implement common structures which could then be added to and customized on a per module / function basis eliminating the need to write the same or similar code multiple times.

Object orientated

The hardware drivers were designed in an object orientated fashion as it provides a useful model for data sharing between functions and since it allows multiple objects (sets of the same hardware) to be used simultaneously without causing clashes the in name-space.

Modular

Designing the system as a set of separate modules allows much flexibility in the final design since modules can be added and removed without the need for major changes to the system as a whole. Furthermore, by mapping the physical system to the software, the code is arranged in a logical fashion and is easy to understand and maintain.

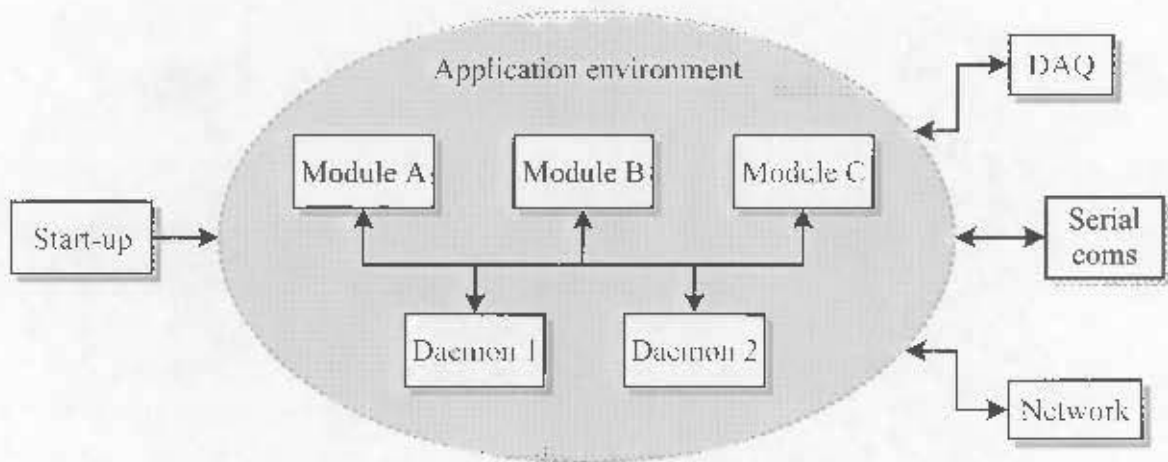


Figure 3.1: System overview

3.4 System Level Design

3.4.1 Overview

Fig. 3.1 depicts the basic design of the system. The main application is started by a dedicated function whose purpose is to initialize all the required services and to launch the modules and daemons. The modules are responsible for the core functionality of the software (e.g. temperature control) while the daemons maintain the intermodule communication and other services required by the application. Modules communicate with each other and with the daemons inside the application environment while communication outside the application environment is performed either by a dedicated module or on a per module basis. For instance, the DAQ communication is handled by the *analog in* and *counter out* modules while serial communications it performed by each module that requires it such as the *cold bath* module.

3.4.2 Module Communication and Data Transfer

A number of different methods for intercommunication between the modules are possible. Each have advantages and disadvantages and are suited to different types of communication.

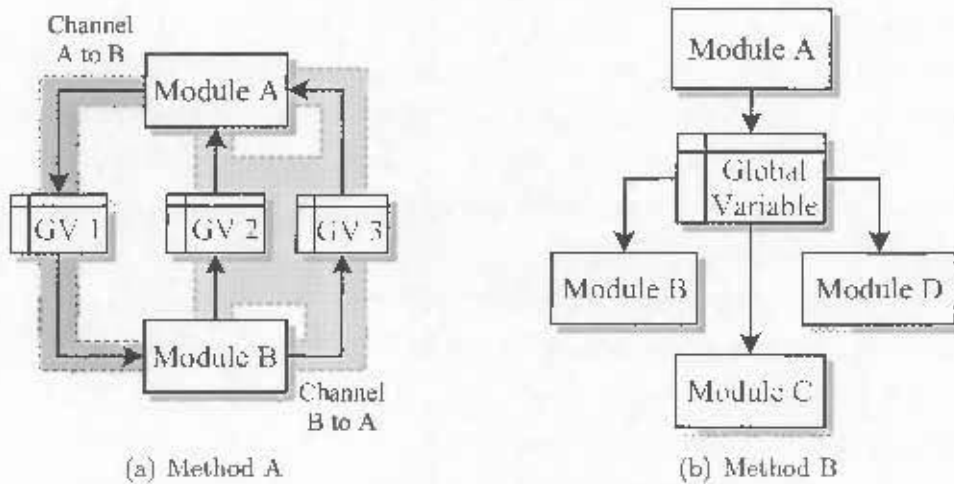


Figure 3.2: Communication via global variables

Global variables

Fig. 3.2 illustrates how communication may be achieved by using global variables. Method A uses a separate global variable for each communication channel and also for each distinct variable to be transferred. Each channel passes data in a single direction only. Method A is not a good method for general module communication but works well for data transfer channels where the reading rate is higher than the writing rate and where old data can be discarded.

Method B is a better use of global variables. A single module writes data to the variable which is then read by multiple other modules. Again, this method is not suited to high data rates and does not preserve old data. Method B is well suited for situations where one module needs to communicate a single piece of information with all other modules. An example of this is the *application run level*. A single module controls the value of the run level while all other modules rely on the run level to ensure a consistent running state throughout the application.

Both method A and B rely on polling of the global variables to see if the data has changed.

Functional globals

In LabView it is possible to implement global variables by creating what is called a *functional global*. This is essentially a global function whose data persists over multiple calls. Functional globals allow the implementation of a form of global variable which has functionality beyond that

of a normal global. For instance, by placing all the function's code inside a case structure and making one of the functions inputs a case selector the global can perform unlimited functions on its data. The most basic use of this is to create a global that has *read* and *write* modes as shown in Fig. A.1. When the mode is *read* it simply passes the data to its output. When the mode is *write* the function overwrites its stored data with the data on its input. Obviously much more complex functionality can be achieved by having more modes and by adding additional code to each case. For the write mode in the above example code could be added to compare the data in the variable with the data to be written and only update the differences or create a time-stamp that is updated at the same time as the data.

Fig. A.1 on page 105 shows how this is implemented in the code using a while loop with an uninitialized shift register to hold the data and a case structure to provide the different modes.

Message queues

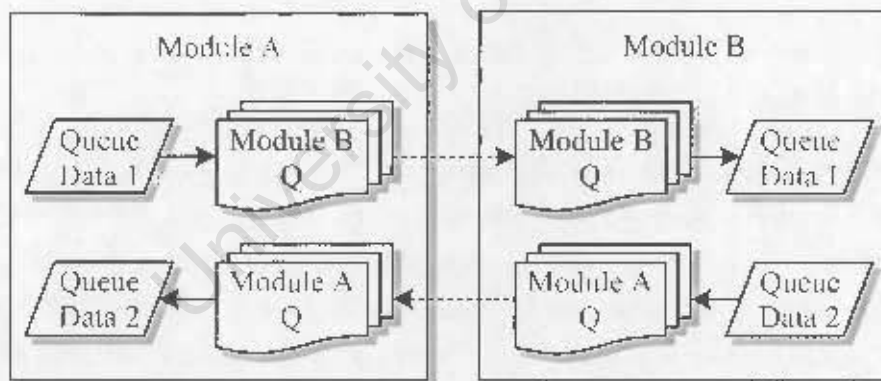
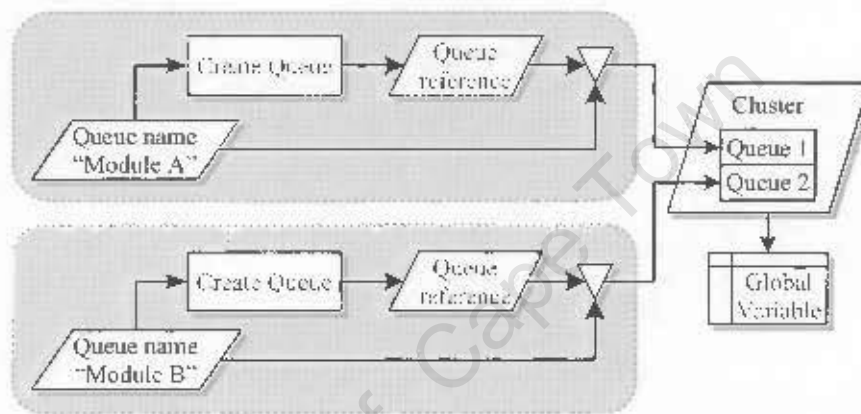


Figure 3.3: Communication via message queues

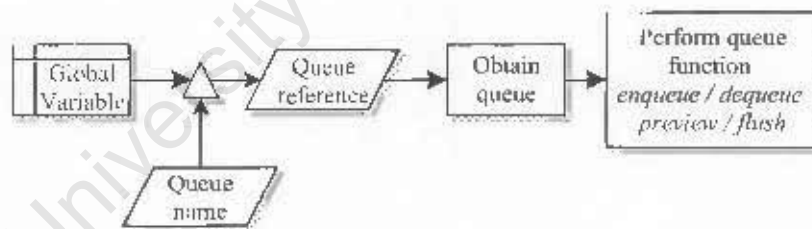
Queues can be adapted to suit almost any type of communication. They provide the underlying structure to transfer data, they do not overwrite old data, although can be made to by limiting the size of the queue, they provide error information and they do not require polling since they contain a means of notification on arrival of new data.

Fig. 3.3 shows how queue's may be used to communicate between two modules. Each module has a dedicated queue from which it expects to

receive commands. Since queue's are accessible globally by using their unique reference it is possible for any module to access the queue of any other module and hence communicate with that module. Fig. 3.4a illustrates how a queue is created and its reference stored in a global variable along with its name to provide easy access the queue references throughout the application. Modules can then access the queue by retrieving its reference from the global variable, using the queue name to select the correct reference, and then obtain the queue with the reference as show in Fig. 3.4b.



(a) Queue creation to allow global access.



(b) Queue use within a module.

Figure 3.4: Implementation of queue handling

When queue's are created they can be configured to use any data type, by creating a cluster of different data elements it is possible to transfer multiple data elements of multiple different types. For communication between modules it was decided to use the data-type shown in Fig. 3.5. Within the cluster are a string and a variant¹. The string contains a command or instruction for the receiving module to execute while the variant may contain any data that is required to perform that command. This is also shown in Fig. 3.5 where the string command is passed to

¹A variant contains flattened data of any type.

the selector of a case structure. That case contains the code to convert the variant to its correct data-type and to perform the function that is associated with that command.

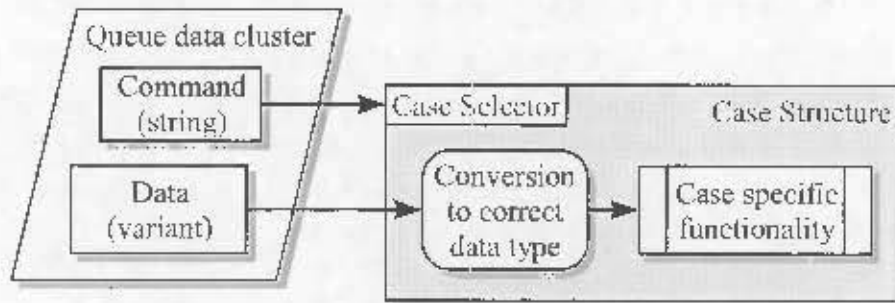


Figure 3.5: Queue data type and possible use

Centralised database

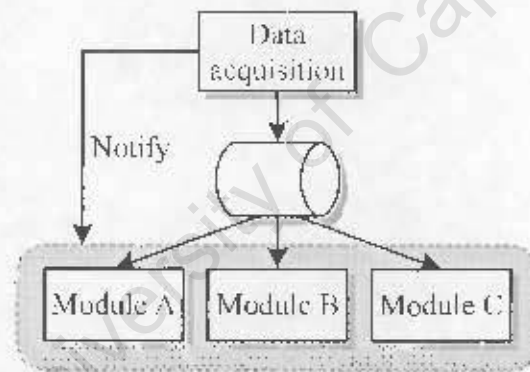


Figure 3.6: Database communication

The use of a centralised database has benefits for data transfer where data rates are not too high and where the data needs to be logged for access at later times. A good example of this would be slow temperature monitoring and control. In this case a particular temperature would be read using a DAQ which would write the data to the database and notify any modules depending on the data that new data had become available. These modules would then query the database to retrieve the data. This is illustrated in Fig. 3.6.

3.4.3 Core Functions

There are a number of core functions that are required by the system. These provide a means for retrieving queue references, pushing data onto the queue's, maintaining connections to queues, storing and retrieving application settings, etc.

The system has one daemon that is responsible for maintaining the queue connections. This is achieved by keeping an open connection to all the queues in use. This daemon is the first process started by the startup procedure and is the last process to end. It is also responsible for any cleanup or other tasks that are required when the application shuts down.

Queue manager

Fig. 3.4 shows how the queues are created and subsequently accessed. This functionality is provided by the functional global² that stores the queue references. This functional global has the following modes:

- ❖ *no-op* - performs no operation but can be used to obtain the cluster of named queue references. It is also used when the function is called for the first time in order to initialize the queues and store their references.
- ❖ *destroy* - when the application is shutdown it is necessary to destroy the queues. This mode performs this function.
- ❖ *send -> module name* - each queue has a mode of this format i.e. 'send -> module A', 'send -> module B'. These modes are used to send a command to another module. The functional global is called with the send mode for the receiving module and passed the command and associated data which are then pushed onto the queue correct for that module by the global.

Fig. 3.7 shows the structure of the queue manager which can be coded as shown in Fig. A.2 on page 106.

²see Sec. 3.4.2, Functional globals.

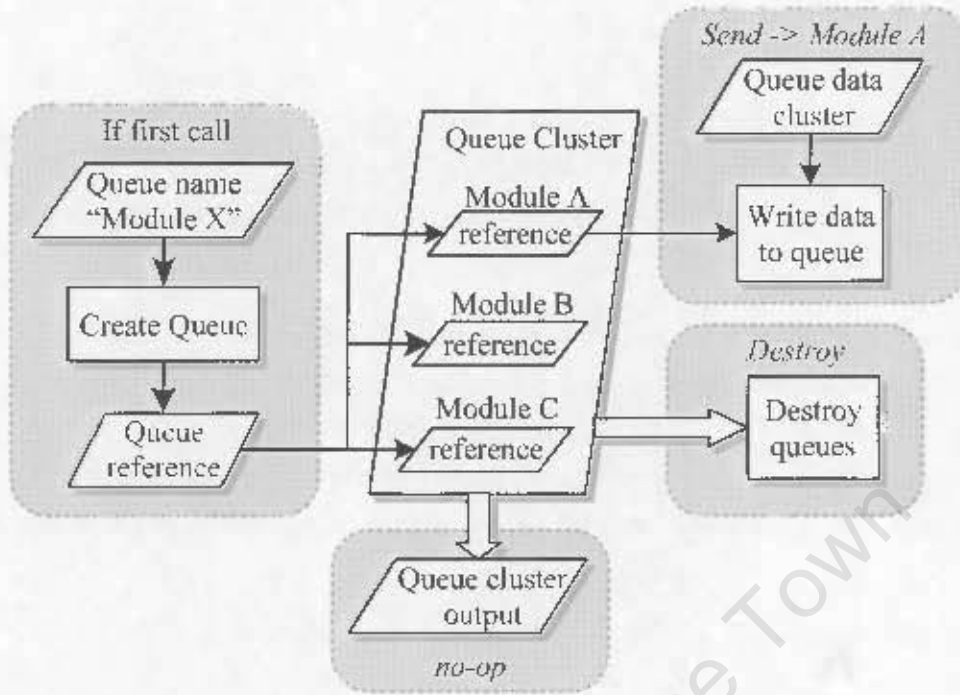


Figure 3.7: Queue manager structure.

Application settings

The application settings are stored in a *.INI*³ file which is read at startup and written to whenever settings are changed. The function that manages the settings is also a functional global that reads the *.INI* file on first call and stores the settings in a cluster that is then accessible whenever the function is called. When settings are changed they are written to the functional global which in turn writes them to the *.INI* file.

3.4.4 Alarms, Events and Data Logging

National Instruments provide an add-on module for LabView called the "data-logging and supervisory control" (DSC) module. This module gives one access to logging, event, and alarm functionality. Using the DSC module a set of *tags* are defined which represent process variables, each with associated properties (alarm levels, data type, logging settings etc.).

³Pronounced *dot-in-ee* file, a file that has a *.INI* extension and contains configuration information stored in sections that contain keys.

Data-logging

The database stores the value of these tags as a trend over time. The user then has the ability to retrieve the current value of a tag or to retrieve its trend over a certain period of time. In this application the *Analog In* module (Sec. 3.5.2) collects data and sends it to the DSC module. At the same time it sends out a notification that there is new data available. The modules that rely on this data can then query the database for the new data. An illustration of this is shown in Fig. 3.6.

Each module also periodically updates a graph on its front panel that shows a trend of the variable(s) that concern that module.

Alarms

Alarms are also handled by the DSC module. Each tag has a set of alarm conditions. When an alarm occurs (the tag value satisfies its alarm conditions) the DSC module notifies all functions that are waiting for an alarm causing them to retrieve the information about the alarm (type, value etc). Each alarm is processed by the module associated with that tag however the Data Overview module (Sec. 3.5.9) provides access to information all the alarms and also provides the ability to acknowledge alarms. Alarms are also stored in the database as part of the tag data.

Possible alarms are HI HI, HI, LO, LO LO and Bad Status. Other alarm properties are alarm priority, dead-band, and alarm message. Alarms can also be configured to auto acknowledge or to require user acknowledgement.

3.5 Module Design

3.5.1 Overview

Each module is based on the same underlying design. Generally it is only the specific functionality that is different.

The module code is divided into two sections, code that processes user interface events and code that performs the module functionality. This

is achieved by having two separate execution loops. The reason for this separation is that it provides greater flexibility and better user interface responsiveness. The loops communicate by means of the module's message queue (Sec. 3.4.2) as shown in Fig. 3.8.

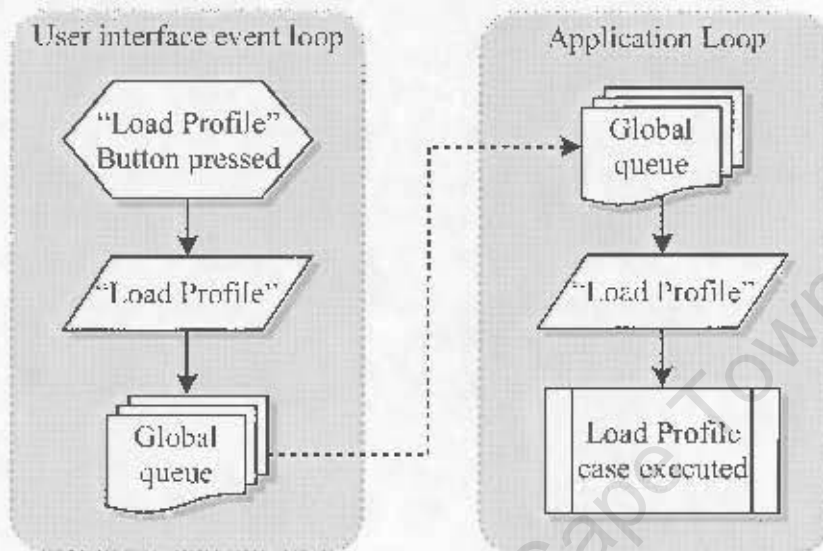


Figure 3.8: Module overview

User interface event loop

The user interface event loop contains a single event structure to catch the events generated by the user's interaction with the interface. This event structure is similar to a case structure in that it executes a particular portion of code depending on the events that occur. Fig. 3.9 shows the basic functionality of the event loop and Fig. A.3 on page 106 shows a simplified version of the code implementation for the interface event loop with the event corresponding to a change in the reactor set point selected.

The loop waits for an event, if one occurs before a specified timeout period then the event is processed (code assigned to that event is executed). If a timeout occurs then the loop executes default code. This code checks the status of the module's queue and also the run level. If the queue status is OK and the run level is equal to 1 then it returns to the beginning, otherwise it ends. Fig. 3.9 shows this process.

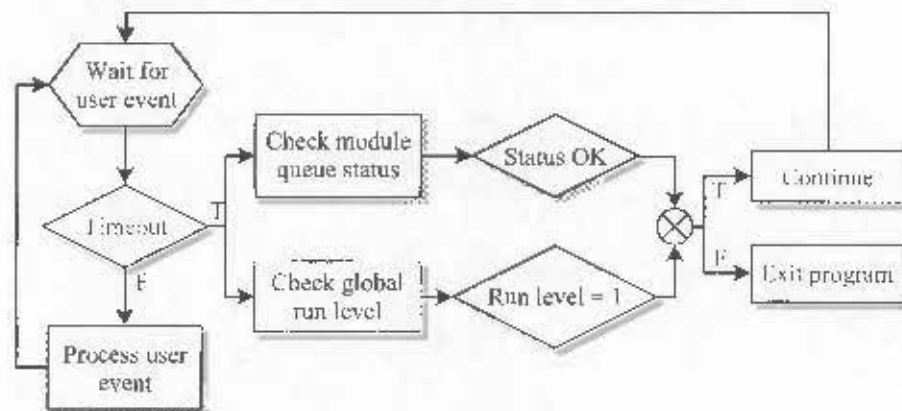


Figure 3.9: User interface event loop flow diagram

Application loop

The application loop contains all the code that provides the module with its external functionality. The loop has a command queue (not to be confused with the module message queue) that stores a list of commands still to be executed and a case structure which separates the code into basic functions including a default case to execute when the command queue is empty.

Fig. 3.10 shows the basic functionality of the application loop. If the command queue is empty, the default code is executed which queries the module's message queue for a new command. If the queue returns an error, the error handling case is executed, otherwise if a timeout occurs before a command is received then a default command is executed. If a command is received that matches one of the high level command cases then it is expanded into a set of low level commands which are then placed in the command queue, otherwise it is just placed in the command queue.

This structure enables complex commands and command sequences to be performed whilst maintaining simple and readable code. Fig. A.5 shows a simplified version of the application loop code with the *default* case selected.

3.5.2 Analog Input and Counter Output

These two modules communicate with the hardware data acquisition (DAQ) card. The Analog In (AI) module reads in data such as the

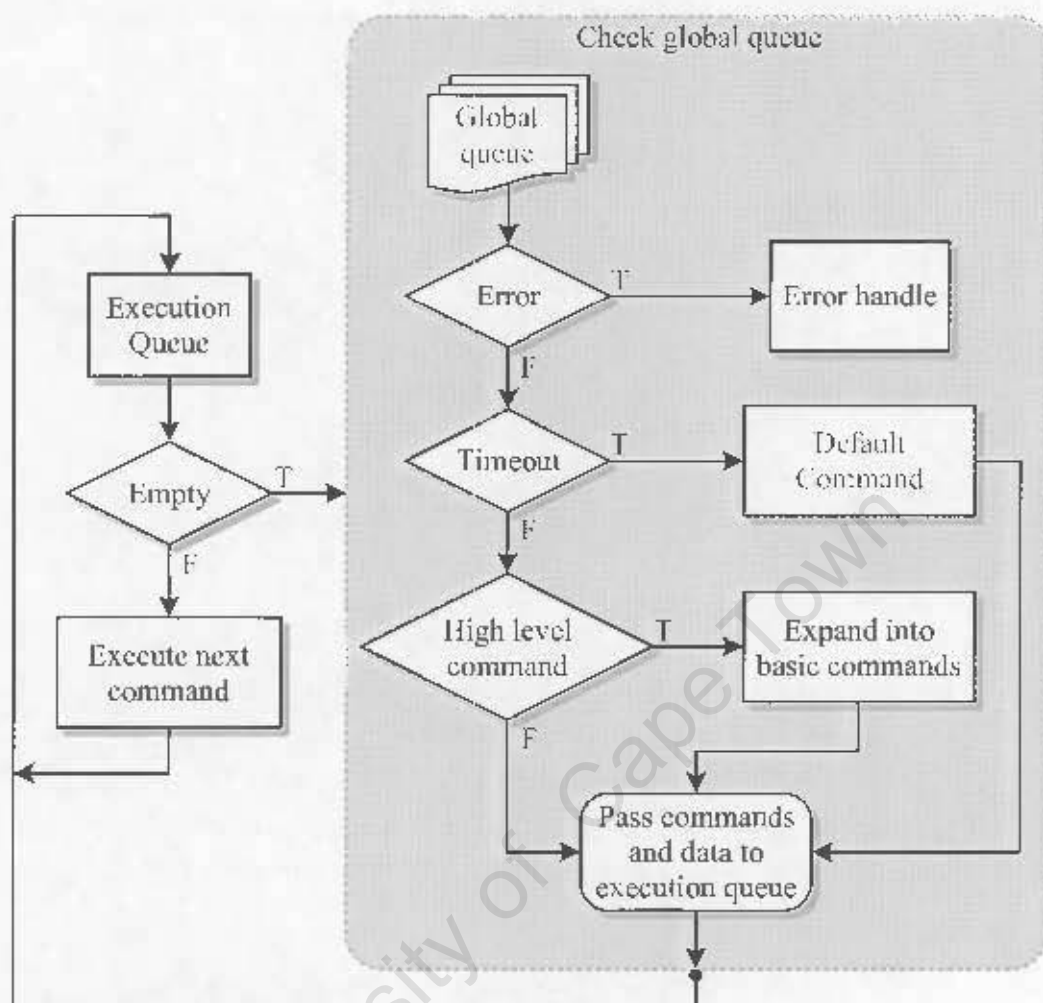


Figure 3.10: Application loop flow diagram

voltage produced by the temperature probes. The Counter Out (CO) module writes data to the DAQ, specifically it controls the state and duty cycle of the counters that switch the heating elements on and off. It is also responsible for generating the polling signal which prevents the hardware from tripping out (Sec. 2.5).

Both of their user interfaces display the raw data that is read or written and buttons that allow the user to reset or to stop the module. In general operation, the user interface is not displayed, only under debug conditions is it accessible.

In addition to the user interface event loop and the application loops described in the previous section, the AI and CO modules have a dedicated loop to perform the I/O. The reason for this is that the I/O loop needs

to execute without interruption. These I/O loops are given a higher priority to ensure this. Figs. A.5 on page 108 and A.6 on page 109 show simplified code for these I/O loops.

The AI module also processes the data that it receives by performing some smoothing and filtering to reduce noise. Once data has been read and processed, the module writes it to the DSC database (Sec. 3.4.4) and sends a command to all other modules that depend on the data to inform them that there is new data available.

The CO module reads the duty cycles for each of its outputs from dedicated global variables which it then writes to the outputs of the DAQ. The duty cycles are update once every second.

3.5.3 System Overview

The system overview module is the central interface module. It does not perform any external functionality but provides access to all other module interfaces and displays the rig flow diagram with all the main variables giving the user an overall view of the state of the rig. This module also monitors the global alarm status informing the user of any current or unacknowledged alarms.

From this interface, the user may load other module interfaces by using the mouse to click on the portion of the flow diagram associated with that module. Fig. 3.11 shows the interface for the System Overview module. The buttons on the bottom bar provide access to those modules not directly linked to the flow diagram, namely the VSM module (Sec. 7) and the Data Overview module (Sec. 3.5.9). The labeled and highlighted areas on the flow diagram (eg. H2, Feed etc) act as buttons that provide the user with access to the other modules. Fig. 3.12 shows how the user interface of the whole system is structured.

3.5.4 General Temperature Control

Four out of the five temperature controllers do not require any advanced functionality. These are TIC's 1, 2, 4, 5 on Fig. 2.1. Each of these

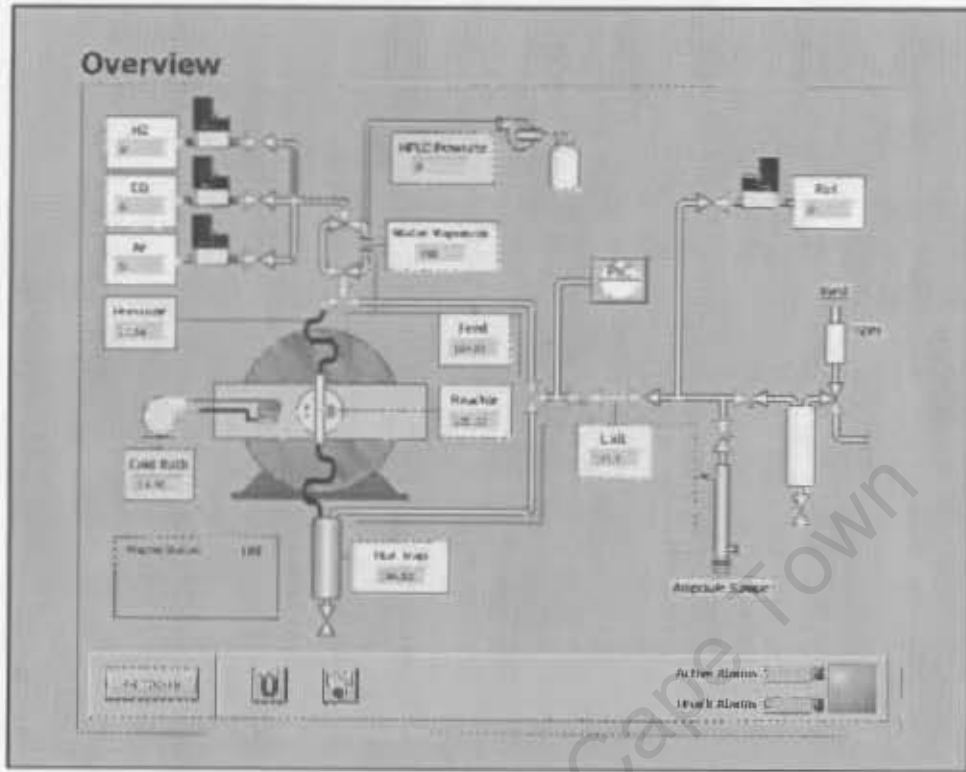


Figure 3.11: System overview interface

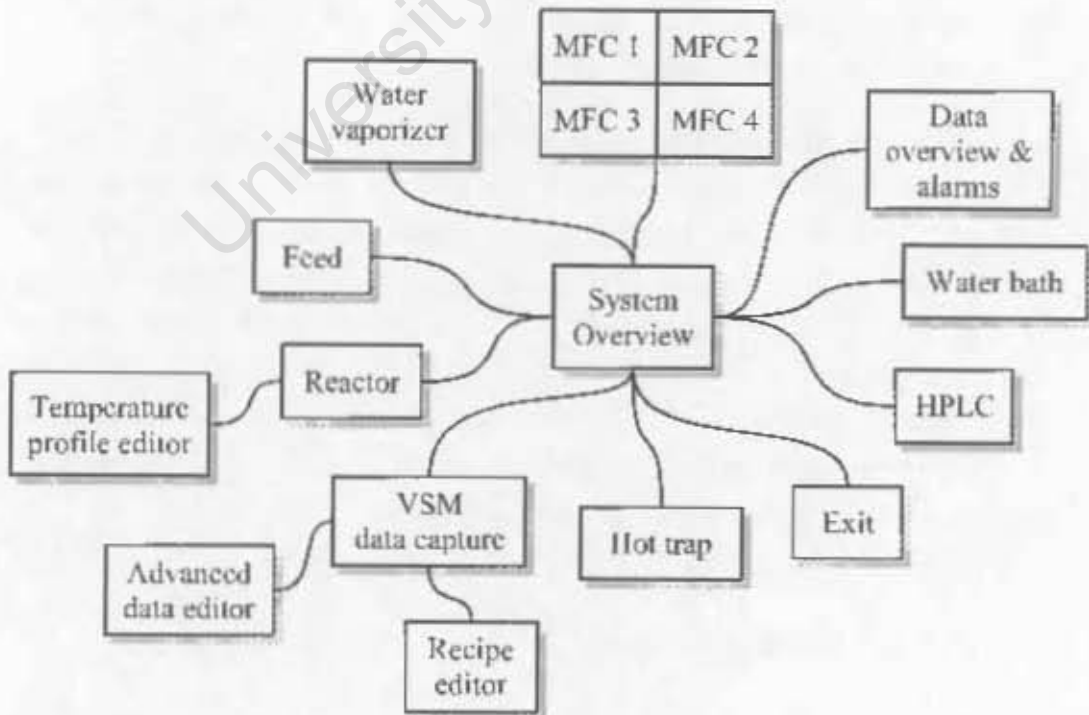


Figure 3.12: Interface layout

modules perform basic PID control on their process variable. Their interfaces display the instantaneous temperature as well as a trend of the temperature over time. The user has direct access to the PID parameters which may be changed manually or by using an auto-tuning algorithm. The user may also enable or disable heating of that particular zone.

These modules are equipped with automatic shutdown if an alarm occurs in the zone that they control. These modules do not provide any temperature profiling capability. Fig. 3.13 shows a screen shot of the interface for these temperature control modules. Fig. A.7 on page 110 shows the code used to perform the PID control.

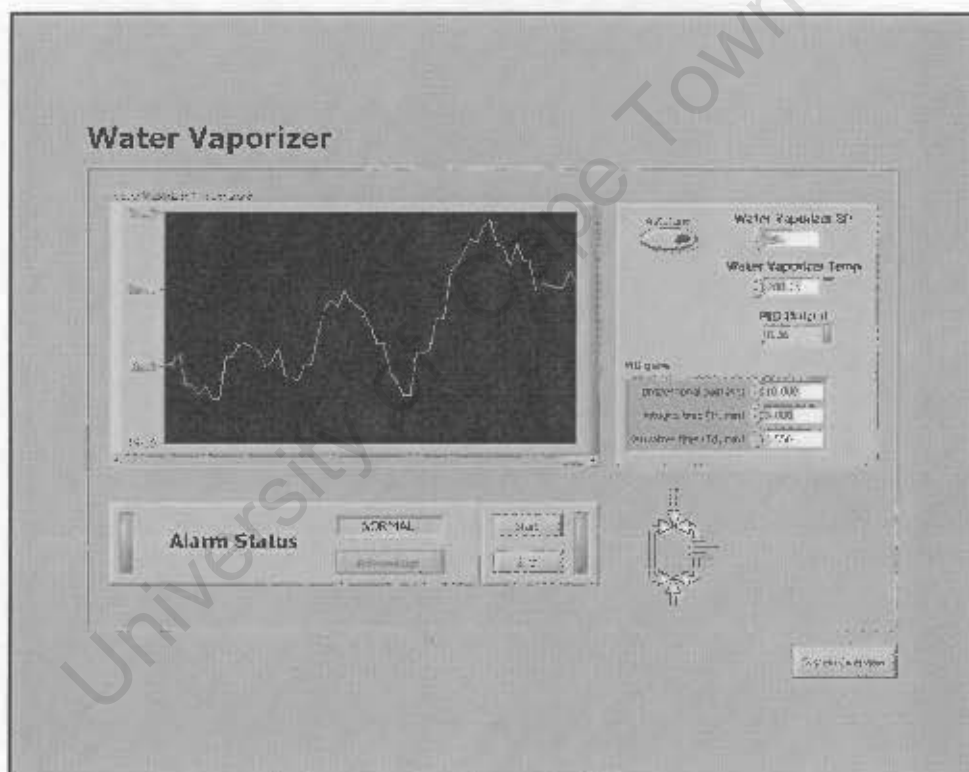


Figure 3.13: General temperature control interface.

3.5.5 Advanced Temperature Control

The reactor temperature is controlled by this module. It provides all the functionality described in the previous section whilst adding functionality that allows the user to use temperature profiles to control the temperature of the reactor. The manner in which it handles alarms is also more advanced. Furthermore the user is prevented from heating the

reactor until certain safety criteria have been met, namely that the water bath is on and is within an acceptable temperature range. The reason for this is to prevent the reactor heating from damaging the VSM coils⁴. Fig. 3.14 shows a screen shot of the Reactor temperature control module interface. This interface is reached by using the mouse to click on the reactor temperature indicator on the interface of the System Overview module (Fig. 3.11).

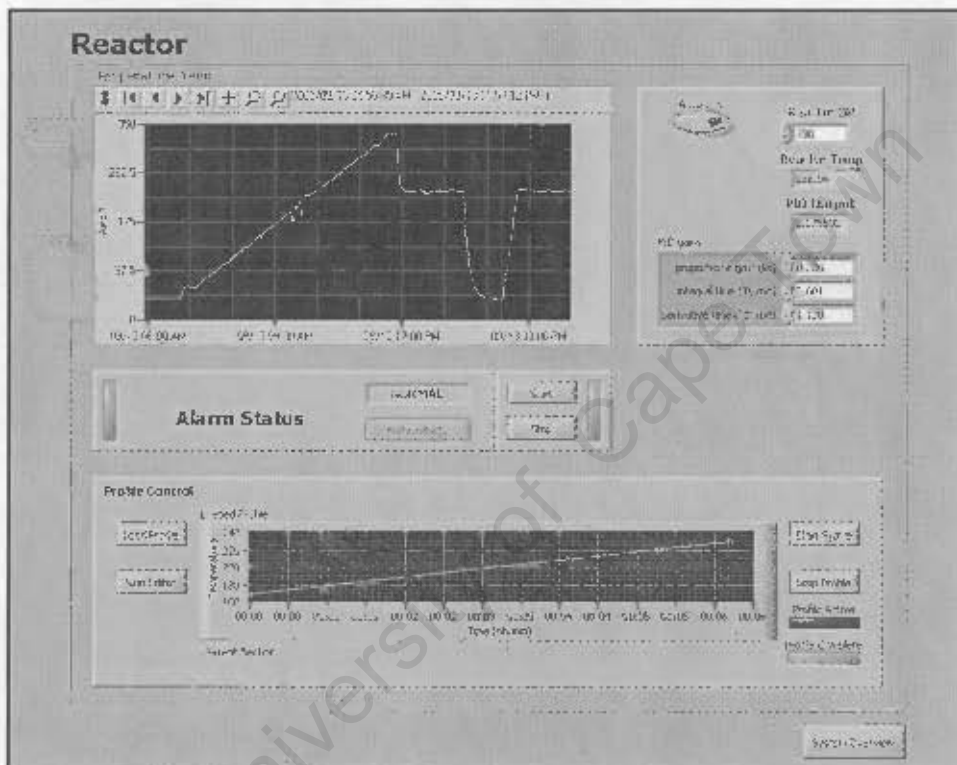


Figure 3.14: Reactor temperature control interface

The *Start* and *Stop* buttons on the user interface control whether or not the module is active (i.e. temperature control is taking place).

Temperature profiles specify the way in which the temperature should vary over time. The user is able to create profiles by using the *profile editor*⁵ which is launched by pressing the *Run Editor* button on the user interface. Once a profile has been created the user can load it into the temperature control module using the *Load Profile* button. The profile is displayed on a graph indicator which also shows the current position of the profile. When the user starts a profile, the first point is set to

⁴see Sec. 6.3.3 for information on the coils.

⁵see Sec. 3.6.1

the current value of the reactor temperature and all the other points are adjusted to ensure the slopes of all the sections are maintained.

When a profile completes, the temperature set-point is left at the last value of the profile. The user may stop a profile at any time in which case the set-point is left at the current value of the profile.

Fig. A.8 on page 111 shows a slightly simplified version of the code responsible for performing the advanced PID control.

3.5.6 Water Bath

As the name suggests, the water bath module communicates with and controls the water bath. The purpose of the water bath is to provide cooling for the VSM coils (see Sec. 6.3.3). The particular bath used for this rig is a Lauda Ecoline series bath with an RE 307 thermostat⁶. Communication between the bath and the software is achieved via the RS-232 protocol and drivers for the bath were provided by the bath manufacturer. Fig. A.9 on page 112 shows the portion of code used to update the water bath.

The user interface for this module shows the current bath temperature and settings and allows the user to change the set-point and settings that the drivers provide access to. Initially the interface had a tabbed layout in order to provide space for all the setting controls and to keep the interface uncluttered. The first tab displaying the set-point, internal and external temperatures and bath status while the second tab contained the controls and indicators for the advanced settings. Later the second tab was removed since the settings contained in it are very rarely altered.

The alarms associated with this module monitor the status of the bath and trigger if there is an error or if the bath temperature exceeds the specified limits. These alarms are monitored by the reactor module which will shutdown if they are triggered (as mentioned in the previous section). Fig. 3.15 shows a screen shot of the Water Bath module interface.

⁶The data sheet can be found on the attached CD or from <http://www.lauda.de>.

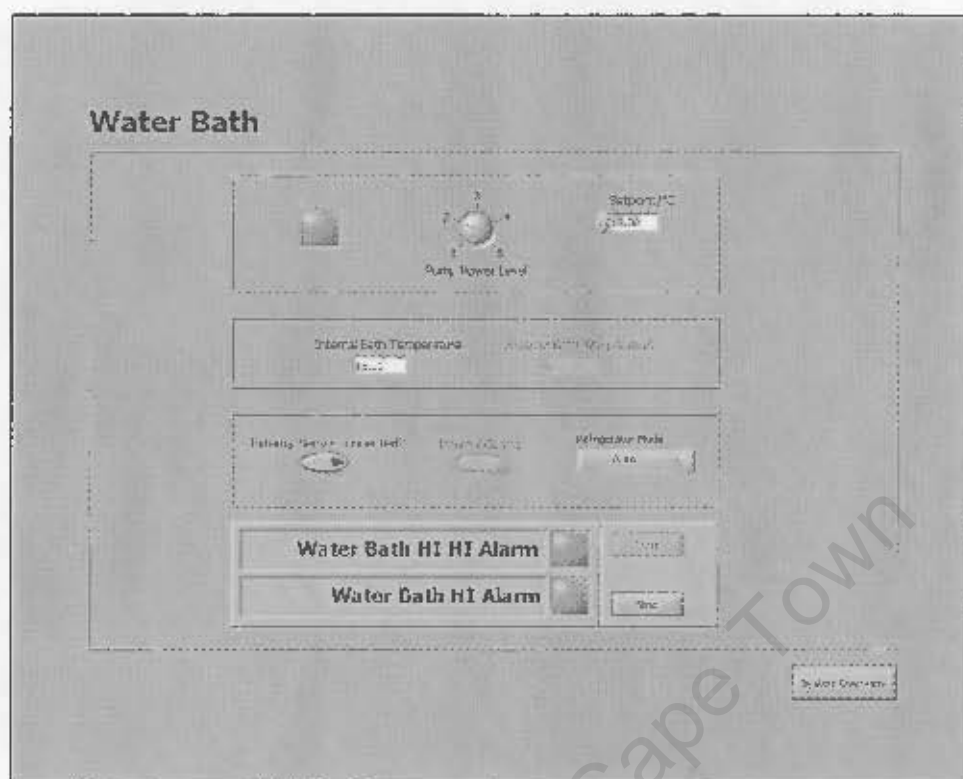


Figure 3.15: Water Bath interface

3.5.7 HPLC Pump

This module is very similar to the previous module. It communicates via RS-232 with the HPLC (High performance liquid chromatography) pump and provides the user with access to all the pumps settings and readings. Its interface is also tabbed to decrease clutter and emphasize the main settings and readings.

Drivers for the HPLC pump were designed to be object orientated and their functionality was based on the communication specifications laid out in the pump manual. Fig. 3.16 shows a screen shot of the control tab of the HPLC module interface. This tab contains only those controls and indicators of most interest. The settings tab contains additional controls and indicators to provide access to the less seldom used functionality. Note that in this screen shot the pump is switched off and hence the controls and indicators are disabled. Figures A.10, A.11 and A.12 show the HPLC functions and an example of their use in the module.

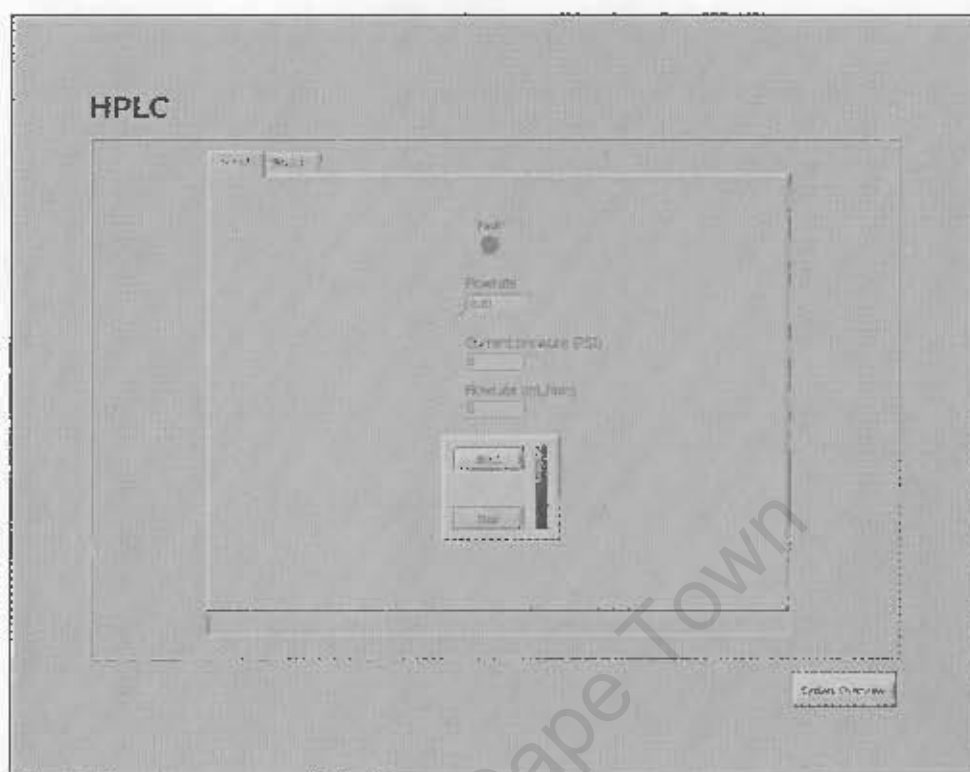


Figure 3.16: HPLC pump module interface

3.5.8 Mass Flow Controllers

The software provided with the BC 154 power supply⁷ opens a communication channel to the unit which other software on the computer can then access using the DDE⁸ interface. Although this may seem overly complex, the low development and hardware cost made it the most viable option.

The MFC module therefore communicates with this DDE software to provide the user with access to the MFC's. Although there is a single module to communicate with all four MFC's, the user experiences a separate pop-up interface for each MFC. This was done to simplify the user's interaction and to make it more intuitive. Each pop-up interface is accessed by clicking an active area on the system overview interface and shows the current set-point, flow and valve state of that MFC. Fig. 3.17 shows a screen shot of the MFC dialog that provides control over

⁷cf Sec. 2.4

⁸Dynamic Data Exchange, an interprocess communication (IPC) system built into the Macintosh, Windows, and OS/2 operating systems. DDE enables two running applications to share the same data. (<http://www.webopedia.com>)

the flow of hydrogen. This dialog is launched when the user uses the mouse to click on the hydrogen flow indicator on the System Overview interface.

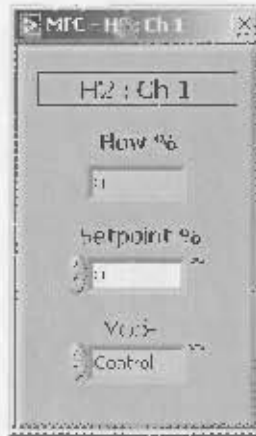


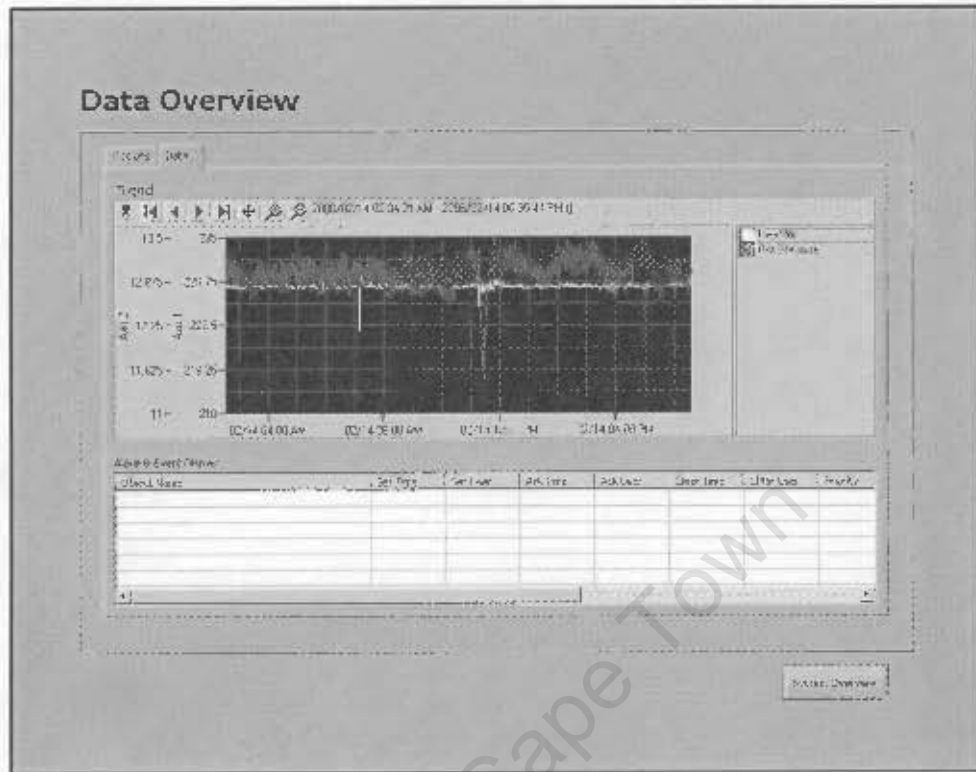
Figure 3.17: MFC control dialog

3.5.9 Data Overview

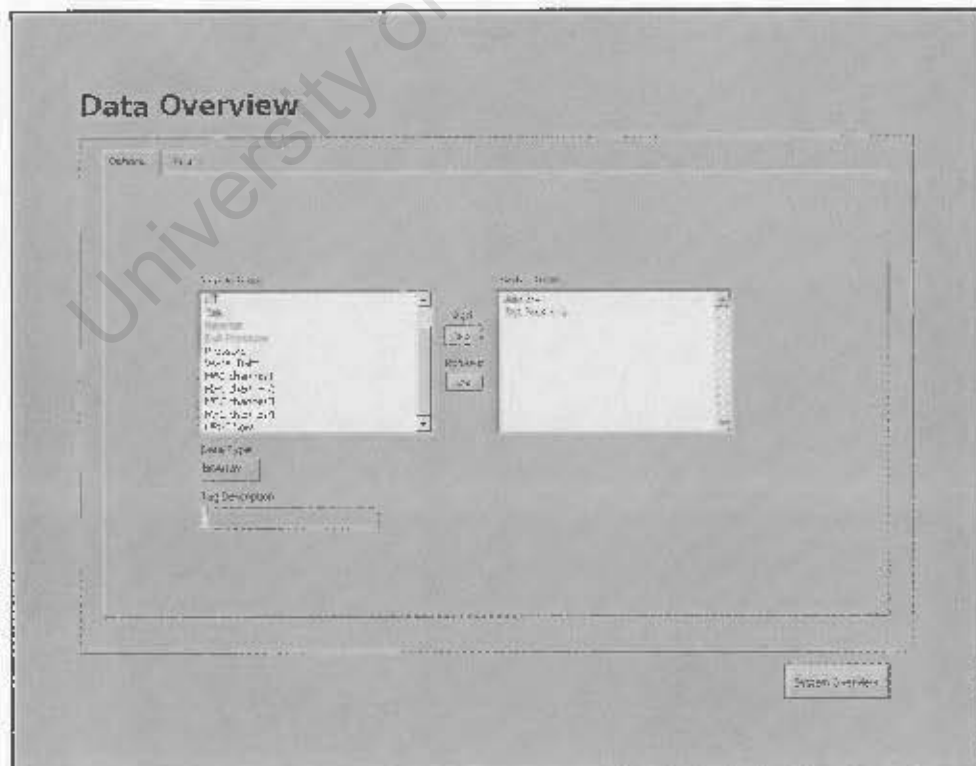
The purpose of this module is to provide the user with access to all the data simultaneously in the form of trends and as a central location for alarm acknowledgement. The interface contains a trend graph on which any combination of variables may be shown giving the user an opportunity to view relationships between the variables.

Fig. 3.18 shows both of the tabs of the data overview interface. The *options* tab allow the user to select which process variables to appear as trends on the *Trend* graph. The *Data* tab shows the *Trend* graph and also a table listing all the currently active or un-acknowledged alarms. Allowing the display of multiple variables on the same graph gives the user the opportunity to view relationships between the variables.

Alarms may be acknowledged and commented by using the mouse to right-click on the appropriate row of the table and selecting an action from the menu that appears.



(a) Data tab



(b) Options tab

Figure 3.18: Data overview interface

3.6 Additional Software

3.6.1 Temperature Profile Editor

The temperature profile editor is a separate application that can be called from the reactor control module. Its purpose is to allow the user to easily create, edit and save temperature profiles which can then be loaded into the reactor control module and used to control the temperature.

Temperature profiles are stored as an array of clusters of two elements, a time-stamp and a temperature. The time-stamp is measured from the start of the profile and the temperature value is the temperature at which the system should be at its associated time-stamp. An example of a temperature profile is shown in Table 3.1. This profile assumes the initial temperature to be 30° C. After 60 s the temperature should be 35° C and should remain there for 10 minutes before cooling back down to 30° C in a further 60 s.

	Point 1	Point 2	Point 3	Point 4
Time (s)	0	60	660	720
Temperature (°C)	30	35	35	30

Table 3.1: Temperature profile example

This format for the temperature profile suits the algorithm that executes it but requires considerable effort on the part of the user to create. The profile editor therefore allows the user to enter the profile as a set of *ramps* and *dwells*. A ramp is a portion of the profile between which the temperature is changing linearly from one temperature to another and is specified by a *ramp rate* and a *final temperature*. A dwell is a portion where the temperature is constant and is defined by a *dwell time*. Considering the same profile as shown in the example, the ramps and dwells would be as follows:

- ❖ Ramp from current temp to 35° C at 5° C/min.
- ❖ Dwell for 10 minutes.
- ❖ Ramp from current temp to 30° C at 5° C/min.

Note that none of the steps have an initial temperature defined. This means that each step gets its initial temperature from the step before. For the first step, the initial temperature is set to the current temperature of the system which eliminates the need to wait for the system to reach the initial temperature of the profile. These ramps and dwells are converted to an array of time and temperature clusters when the profile is saved. Additional functionality of the application includes:

- ❖ Display a graph of the profile.
- ❖ Allow the user to add new steps at the front, back or anywhere in the middle of the current profile.
- ❖ Allow the user to edit each step without the need to delete it and add a new one.

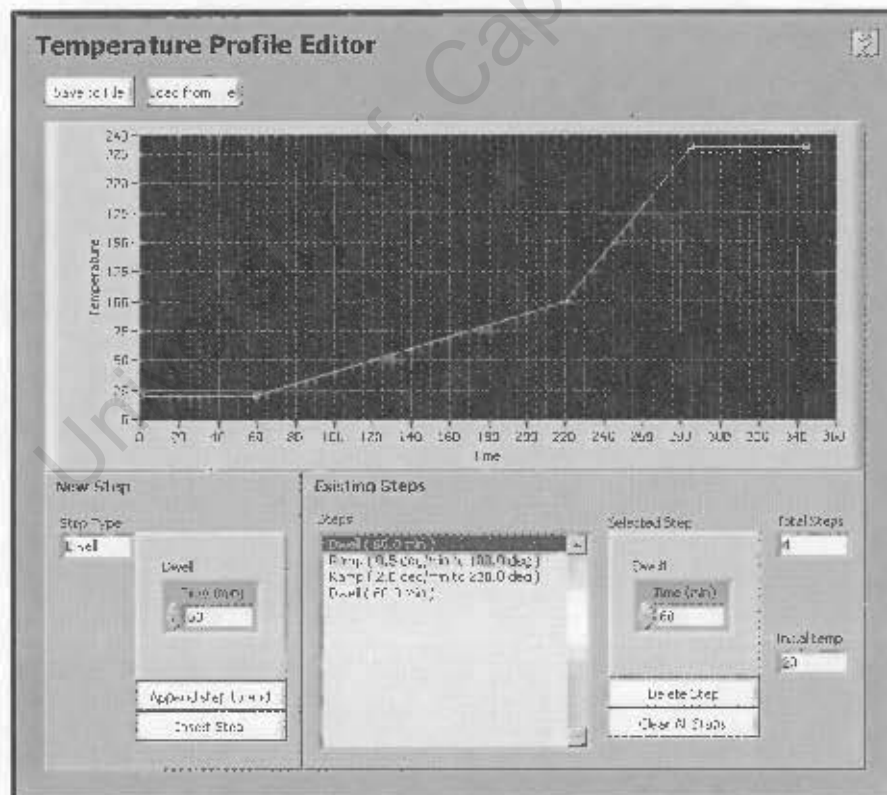


Figure 3.19: Temperature Profile Editor user interface

Fig. 3.19 shows the user interface for the Temperature Profile Editor.

Chapter 4

Results

"If email had been around before the telephone was invented, people would have said, 'Hey, forget email! With this new telephone invention I can actually talk to people!'"

- *The Executive Speechwriter Newsletter*

4.1 Temperature Control Performance

Fig. 4.1 shows the performance of the reactor temperature controller in response to a step input from 50°C to 230°C. The controller exhibits a zero degree overshoot and once settled maintains the temperature within 0.5°C as shown by the magnified graph (dotted curve with axis on right). This performance is dictated by the specific tuning parameters chosen for the controller.

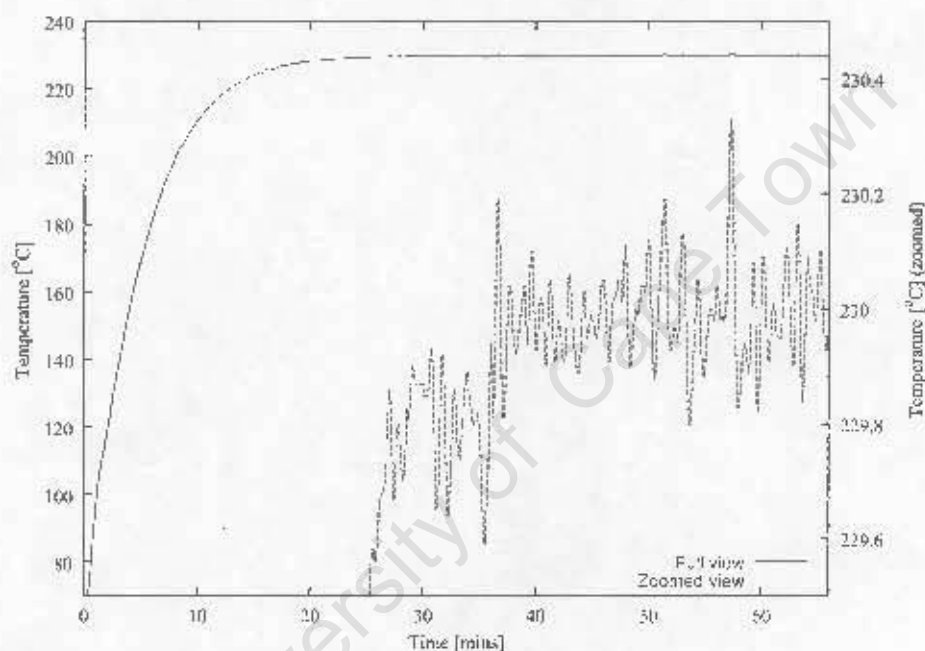


Figure 4.1: Reactor temperature control

Although the performance of the other temperature controllers is not shown they did meet the specifications, that is they were able to heat up to 300°C with a stability of $\pm 5^\circ\text{C}$.

4.2 Serial Communication

The computer communicates via RS-232 with the water bath, the HPLC pump and the MFC's. Software drivers were written for the water bath and HPLC pump. These drivers enabled the respective module to communicate with and control the devices. Initially some problems were experienced with the drivers for the HPLC pump whereby the some of

the settings on the pump were reset each time the HPLC module was started.

As stated in Sec. 3.5.8, the control of the MFC's was performed by sending commands to a software application provided by the manufacturer. Although this system worked well initially it was later discovered that when this software is left running for extended periods of time (more than a few hours) the application ceases to function. Since the MFC's need to be controlled over much longer periods of time and since control of the MFC's requires only occasional user input it was decided to manually control the MFC's.

University of Cape Town

Part II

Vibrating Sample
Magnetometer

Chapter 5

VSM Background



I do not feel obliged to believe that the same God who has endowed us with sense, reason, and intellect has intended us to forego their use.

- *Galileo Galilei*

5.1 Basic Concept

The vibrating sample magnetometer (VSM) is a well established instrument for measuring the magnetic properties of materials or more specifically for measuring the hysteresis curves of materials. In 1959 Foner [14] published a paper giving a detailed description of the design and operating characteristics of a vibrating sample magnetometer. His paper has been the starting point for the design of many VSM's since.

The principal behind vibrating sample magnetometers is Faraday's law which states that an EMF is induced in a conductor by a time varying magnetic field. In VSM's, a sample is moved with respect to stationary pickup coils (which is theoretically identical to the movement of the coils relative to the sample). The signal induced in the pickup coils is time dependant and can be measured by appropriate electronics. If it is assumed that the coils are stationary and that applied field is time independent and uniform over the range of motion of the sample, then we can say that EMF induced in the coils is due only to the changing flux produced by the moving sample.

Based on these assumptions it is possible work backward from the induced EMF to get information such as the field produced by the sample and therefore its magnetisation for a specific external field.

It should be noted here that in the past VSM's have only been used to measure *external* samples ie. samples that are not part of a reaction at the time of measurement. The significance of this is that this is the first *in-situ* VSM measurement system.

5.2 Common Design Features

All VSM's have three common features, a means of vibrating the sample under consideration in a stable, periodic manner, a set of stationary detection coils and an electronic system that allows the measurement of the signal induced in the detection coils. Some of the methods used to implement these features are described in the following sections.

5.2.1 Vibrating Mechanisms

In order to induce a current in the pickup coil a relative motion between the sample and the coil is required. Various different means of producing this motion as well as the characteristics of the motion have been used in the past. Desired characteristics of the vibrating mechanism are that it moves the sample in a stable manner and that this motion is reproducible. Some of the more common methods are listed below.

- ❖ Flanders & Doyle (1962) [12], Johansson & Nielsen (1976) [21] and Hoon & Willcock (1988) [20] describe motor and crank arrangements
- ❖ Drake & Hartland (1973) [11], Hoon (1983)[19] and Niazi *et al* (2000) [27] employ various types electromagnetic vibrators most of which are similar in design to a 'loudspeaker'.
- ❖ Other techniques include using piezoelectric or pneumatic devices.

Hoon & Willcock point out that whilst inherently simple and mechanically quiet 'loudspeaker' type designs need to be carefully stabilized and are prone to amplitude and frequency drift in the long-term. The transducer also needs to be able to dissipate relatively large amounts of power to provide sufficient displacement or to move sample arrangements of reasonable mass. Motor and crank designs are more complex mechanically but very simple electronically and provide oscillations of stable amplitude and frequency over long periods of time and can sustain high inertial loads.

5.2.2 Detection Coil Systems

The purpose of the coils is to detect the presence of the time varying magnetic field produced by the moving sample. Apart from Foucr's paper [14] which touches briefly on various coil systems, but does not present any analyses of them, there have been a number of papers published on the topic of coil design. Most notably, a simple design method based on reciprocity was presented by Mallinson in 1956 [25]. In his paper Mallinson discusses an optimal coil configuration "where four identical

N-turn coils are used, thus increasing the output, and ... providing insensitivity to magnet or external field variations." He also describes a transverse geometry which is used by Hoon & Willcock (1988) [20].

In 1976 Guy [18] elaborates on Mallinson's design in a letter written in response to a paper by Bragg and Sechra (1976) [5] who offer a complex analysis of the EMF induced in VSM coil systems.

Bowden (1972) [4] writes about coil design with specific reference to those "mounted on conventional iron cored magnets". He also presents experimental result showing a very broad saddle point for one of the arrangements he discusses.

Other papers include a detailed discussion of the general theory of detection coil systems given by Pacyna (1982) [29] which is taken further by Pacyna & Ruebenbauer (1984) [30]. Ausserlechner *et al* (1994) [2] presents a theoretical discussion of pickup systems based on magnetic multi-pole expansions.

5.2.3 Electronic Systems

In Foner's original paper [14], he describes an electronic system whose functions are:

"(1) to permit accurate calibration of the signal output obtained from the detection coils, (2) to produce a convenient dc output signal which is directly related to the input and which can be recorded (3) to provide sufficient amplification for high sensitivity operation."

Many different electronic systems can be designed to perform these functions and will vary greatly depending on the physical design of the system. The common features in most systems are:

❖ vibration control / generation

Mechanical vibration (motor etc.) is usually accomplished by using an inverter or dc motor drive while other systems use an amplified signal from a signal generator to drive a speaker type arrangement.

- ❖ synchronization between the vibration and the signal generated in the pickup coils

Not all systems require this but those that do use lock-in-amplifiers, phase lock loops or differential amplifiers and phase shifters. The purpose of this step is to isolate the signal of interest and improve the signal to noise ratio.

- ❖ signal amplification and display / recording

In general the signal produced by the coils needs to be amplified in order to get it to a reasonable level for measurement or display. The display / recording of the signal can be performed by a variety of different components from oscilloscopes and X Y recorders [14] to microcomputers [20].

University of Cape Town

Chapter 6

VSM Hardware Design



When a train goes through a tunnel and it gets dark, you don't throwaway the ticket and jump off. You sit still and trust the engineer.

- Corrie Ten Boom

This chapter will discuss the physical aspects of the design of this VSM system including an overview of the coil design by Krylov[22], a simulation of the expected signal and the electronic systems used.

6.1 Features and Specifications

The particular features desired for this vibrating sample magnetometer are somewhat different compared to the requirements for a general purpose VSM due to the specific nature of the experiments to be performed and the data that they require. Some of the specifications for the system are listed below.

- ❖ The coil arrangement should optimise space usage to allow for the minimum magnet pole gap and hence maximum field strength between the poles.
- ❖ The external magnetic field must be able to vary continuously between -2 T and +2 T to allow for magnetic saturation of typical catalyst samples.
- ❖ The coil design should maximise the sensitivity of the pickup coils whilst minimising any effects of sample offset.
- ❖ The electronic systems should amplify the signal to improve data acquisition performance.
- ❖ The electronic systems should provide filtering to improve the signal quality.
- ❖ The mechanism used to move the sample should minimise disturbances on the sample caused by excess vibration.
- ❖ The system should provide a cooling mechanism for the coils to ensure they remain at a constant, low temperature.

6.2 Constraints

Space

The constraints on the physical aspects of the system are mainly related to space. In order to enable the field to reach ± 2 T it is necessary to have a pole gap of less than 48mm. The magnet used in this instrument is a Bruker B-E25V. This magnet produces a maximum field of 2.15T with tapered pole caps and an air gap of 40mm. The magnet is powered by a Bruker BTSPS BIP 1 200/60C5 power supply which has a single output channel rated at ± 200 V, ± 60 A, 12KW with a load resistance of 3.3 ohm. The field is measured by a Projekt Elektronik Digital Teslometer FM 3000¹. Photographs of the magnet are included in Appendix B.

Since the entire coil and reactor arrangement needs to be between the poles of the magnet this requirement places a serious constraint on the coil design². A further space constraint stems from the requirement stating that the coils need to be in a homogeneous field. The magnet used in this system has a circular pole face of diameter 140 mm. Fig. 6.1 shows the homogeneity of the magnetic field between the magnet poles moving radially outward from the center of the pole face. The graph shows that, for a field variation of less than 1 %, the sample must remain within a circle of radius 45 mm and centered at the pole center. This means that the pickup coils must be designed to ensure that the sample remains within this region (marked as shaded circle on Fig.).

Ferromagnetic Material

As described in the previous chapter (Sec.5.1) it is assumed that the signal produced in the coils is solely due to the magnetization of the sample. It is therefore necessary to ensure that the amount of ferromagnetic material other than the sample that will generate a signal in the coils during a measurement is kept to a minimum. It is possible to reduce these effects by calibration but this is not the desired situation.

¹The specifications for the magnet are included in Adobe PDF format on the CD accompanying this thesis.

²The design of the reactor is beyond the scope of this document however heating aspects of it are discussed in Sec. 2.3.

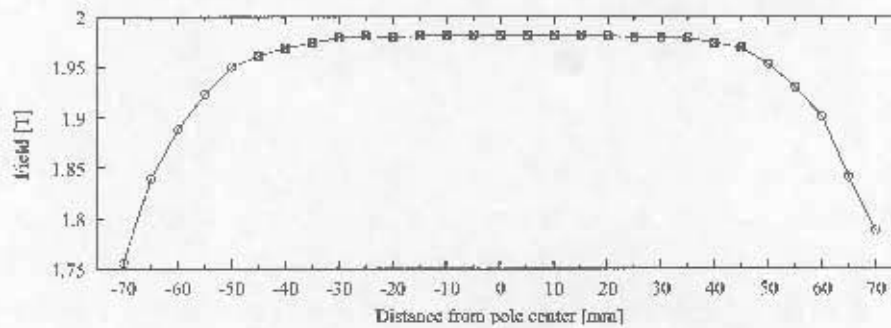


Figure 6.1: Variation of magnetic field over pole face. Field set to maximum for a pole gap of 50 mm.

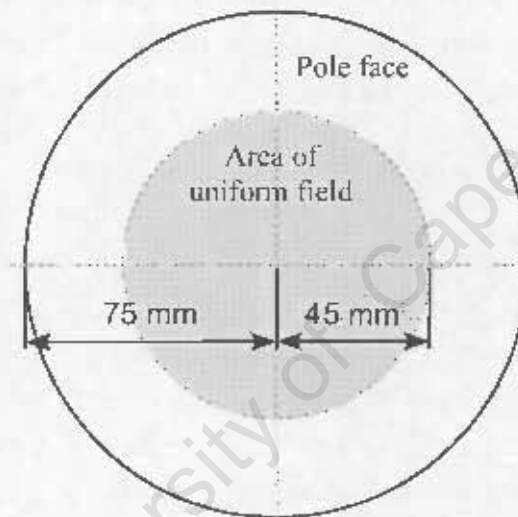


Figure 6.2: Magnet pole face showing dimensions and area of uniform field.

Temperature

The high temperatures required by the reaction introduces another constraint since the wire used to construct the coils has a maximum operating temperature of 250°C above which the wire's enamel coating melts. It is therefore necessary for the temperature of the coils to remain low. In order to achieve this some cooling must be provided for the coils.

6.3 Physical Design

This section looks at the physical aspects of the VSM design, namely the design of the pickup coil system, the coil cooling and the method for moving the sample. Results from a simulation of the expected signal induced in the coils is also included.

6.3.1 Coil design, Construction and Simulation

Design

The coil design was done by I. Krylov [22] from the University of the Western Cape. His design produces a similar results to Mallinson's [25] design although his process is slightly different. He approaches the design by analysing the flux through a single turn coil due to a magnetic dipole placed a distance x away from the plane of the coil and whose axis runs through the center of the coil as shown in Fig. 6.3.

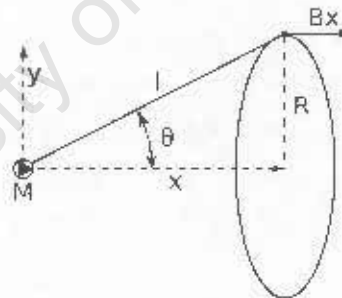


Figure 6.3: Dipole flux through single turn coil

Having produced an expression for the the flux through the coil his next step is to assume a density of turns δ and to calculate the flux for a coil of inner radius R_1 and outer radius R_2 . For the range of values of R_1 and R_2 that are practical in this case, based on the constraints stated above, he shows that the single turn approximation is very good and can be used for further calculations.

Using this approximation he numerically calculates the flux through the coil when the dipole has been shifted from the x -axis (which passes through the coil center) by a distance y . This calculation represents a single frame in the motion of the dipole. The calculation is performed

for the case where the distance between the dipole and the plane of the coil is equal, $x = R$. Fig. 6.4 shows this relationship.

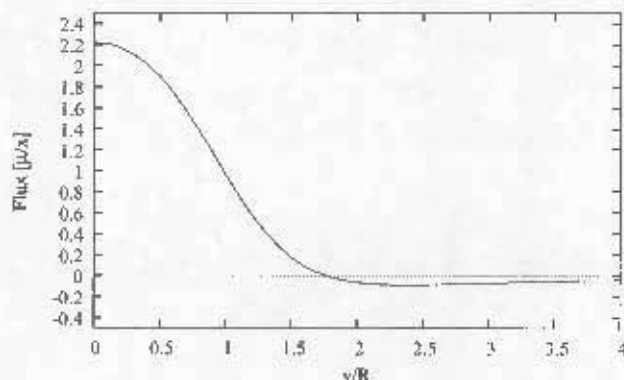


Figure 6.4: Flux vs Displacement

From Fig. 6.4 it can be seen that the maximum flux change occurs when the dipole is displaced by a distance $\frac{y}{R} \approx 2.4$. Thus for maximum sensitivity the dipole should oscillate about the center of the coil with a maximum displacement equal to $2.4R$.

He then goes on to discuss the advantages of using multiple coils. First he places a second coil in the same plane as the first with the centers of the two coils separated by a distance $2.4R$ (see Fig. 6.6 for concept). With the coils connected in series, the movement of a dipole vertically past the coils causes the total flux being linked by the two coils to be as shown in Fig. 6.5. Notice that from peak to trough, the flux variation is double that of a single coil.

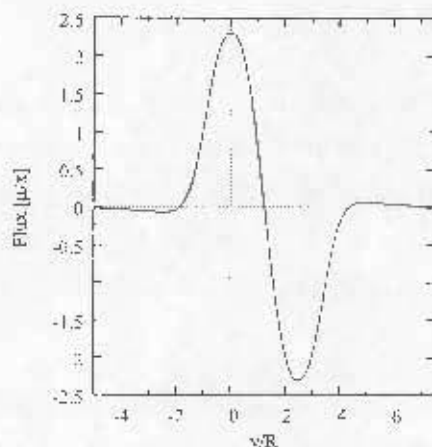


Figure 6.5: Flux through two coils connected in series

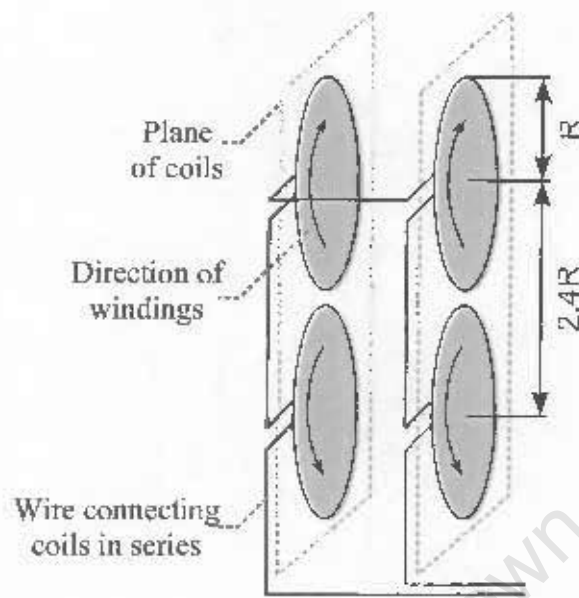


Figure 6.6: Final coil arrangement and wiring.

Finally, by adding another two coils opposite the first two, as shown in Fig. 6.6, he describes a compensating arrangement very similar to Mallinson's [25]. This arrangement reduces the effect of variations in the external field and also improves the usability of the instrument by removing the need to exactly position the sample between the two sets of coils since the sum of the perpendicular distances to each set of opposing coils will always be equal.

Krylov chooses the final dimensions of the coils as shown in Fig. 6.7.

Construction

The coil bobbins were manufactured out of aluminium by the department's mechanical workshop to the specifications shown in Fig. 6.7 above. To meet the design specifications of 10 000 turns per coil, 0.06 mm enameled copper wire was used as the winding. In order to wind coils of such high turns and with such fine wire it was necessary to design and construct a coil winder.

The coil winder consisted of a DC motor fitted with an attachment to allow the coil bobbins to be fixed to the motor axle, a hardware interface to allow the speed of the motor to be controlled by the computer and a system to generate a pulse for every full rotation of the motor. Initially

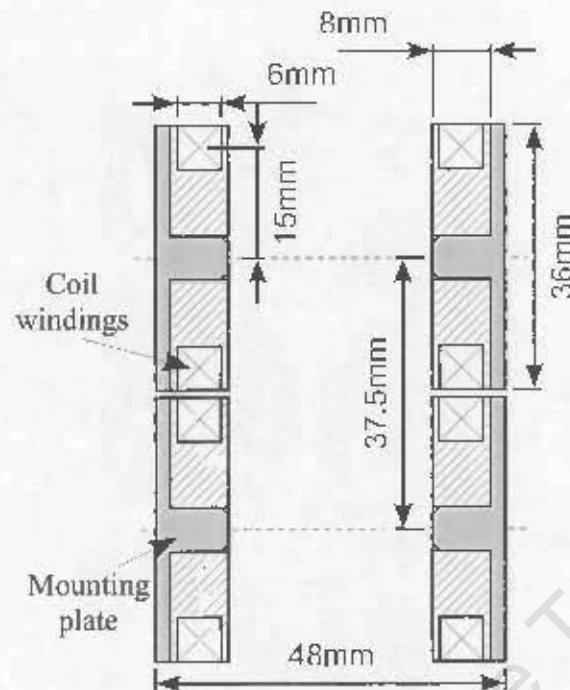


Figure 6.7: Coil dimensions

a stepper motor was used to guide the wire onto the coil however the stepper motor was not able to move in fine enough steps to produce smooth windings. To replace this the wire was guided manually. In order to prevent the wire from snapping the wire reel was mounted on bears to allow it to rotate freely. Finally a software system was written to control the speed of the motor and to count the number of turns. The software varies the speed of the motor in the following manner:

1. From standstill slowly accelerate the motor (to prevent the wire from snapping) to full speed (25Hz).
2. After 5600 turns slowly decelerate the motor to 2 Hz.
3. Remain at 2 Hz until 7000 turns and then stop.

A number of trial attempts were required to achieve find the best sequence. As step 3 states, the coils were only wound up to 7000 turns. The reason for this is that due to poor uniformity in the windings (caused by manual guidance of the wire) the volume taken up by the windings exceeded the volume calculated in the coil design. This did not cause

any problems however since the only effect was to reduce the sensitivity of the system.

The number of turns was counted using an opaque disk that contained a single notch. This disk was attached to the axle of the motor and also placed so as to interrupt the infra-red beam of an infra-red emitter / receiver pair. Thus during a rotation the beam would be broken until the notch in the disk allowed the beam to reach the receiver generating a pulse which is fed to the digital input of the computer.

To provide a more robust means of connecting the coils to the rest of the measurement system a 10 cm piece of 0.5 mm wire was soldered onto each end of the coil windings. The solder joints were glued onto the coil to ensure that no additional tension was put on the fine wire of the windings.

Images of the coil winder and the coils are shown in Appendix D.

Simulation

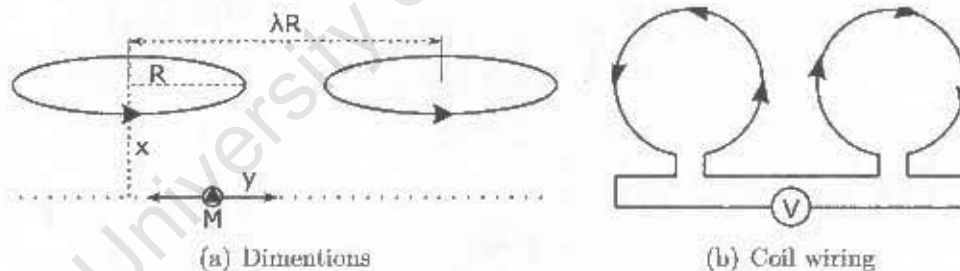


Figure 6.8: Simulation arrangement

Based on the design discussed in the previous section and by evaluating some of the mathematical expressions given by Krylov it was possible to numerically simulate the signal that could be expected from the pickup coils. The purpose of the simulation was to investigate the effect of various different dimensions on the final shape of the signal. Fig. 6.8 shows the arrangement for the simulation. As in the above section we use the case where $x = R$. λR is the separation of the two coils. The displacement of the dipole (y) is measured from the center of the left hand coil and its units are the same as that of the coil radius (R). The starting and end position of the dipole can also be varied to allow investigation

of other effects of the sample movement. It is assumed that the dipole moves sinusoidally from its start to end point and back again. The signal is calculated as follows:

$$\frac{dF}{dy} \times \frac{dy}{dt} = \frac{dF}{dt} \propto V \quad (6.1)$$

By multiplying appropriately scaled and shifted functions of $\frac{dF}{dy}$, flux vs displacement (Fig. 6.4) and $\frac{dy}{dt}$, the velocity (assumed to be sinusoidal), the shape of the signal produced by the coils can be calculated.

Fig. 6.9 shows the results of running the simulation for six different scenarios.

- ❖ Fig. 6.9a is what one would expect for a system where the spacing between the coils is between $2R$ and $2.5R$ and the dipole moves precisely between the coil centres.
- ❖ In Fig. 6.9b the dipole motion is of the same amplitude as before but its center has been shifted causing an asymmetric signal.
- ❖ In Fig. 6.9c the motion is again centered about the two coil centers but the amplitude of the motion is greater than λR ie. the sample moves past the coil centres.
- ❖ In Fig. 6.9d the dipole moves between the centers of the coils but the coil displacement has been increased to $\lambda R = 3R$.
- ❖ Finally Figs. 6.9e & 6.9f show the combined effect of the above scenarios. Notice the asymmetry in Fig. 6.9f caused by the offset motion, the individual effect of which is shown in Fig. 6.9b.

It should be noted that the effects of mutual inductance or coupling between the coils was not investigated in either the design or simulation.

6.3.2 Sample Movement

In order to induce a voltage in the coils relative motion between the coils and the sample is required. Some of the documented methods used to produce this motion are described in Sec. 5.2.1.

One of the specifications stated above was to provide a mechanism for moving the sample that would minimize any forces on the sample. The reason for this is that the samples to be measured by this instrument are delicate in that large vibrations would change the physical structure. This means that the sample has to be moved smoothly which rules out the option of having high frequency vibrations. As the sample will be moving slowly and the induced signal is proportional to the velocity it is therefore necessary to ensure that the amplitude of movement gives the largest change in flux to increase the sensitivity of the instrument. In the previous section it was stated that for the largest change in flux the sample should move between the centers of coils separated by a distance of approximately 2.5 times the radius of the coils. For the dimensions of this system the sample needs to move a total vertical distance of 37.5mm.

The final design of the system used to provide this large motion is shown in Fig. 6.10. The following points should be noted about the design:

- ❖ The reactor clamp is made of aluminium since it has a low mass and hence requires less force to be moved by the motor. Aluminium is also non-magnetic.
- ❖ The fork design of the reactor clamp ensures a minimum amount of aluminium between the poles of the magnet. Although aluminium is not magnetic, large eddy currents are induced in it when it is moved through a magnetic field. Not only does this cause interference with the signal from the sample but it also increases the amount of work needed to move the reactor³.
- ❖ The handle of the fork is fixed in place but can rotate vertically.
- ❖ The fork is oscillated vertically by means of an arm attached to a rotating crank. The crank is rotated by a geared electric motor.
- ❖ The motion of the reactor is not completely vertical, but in fact traces a small arc of a circle centered at the pivot point. The calculated error in the sample motion due to this is approximately 1% of the total vertical motion (calculated as the ratio of the horizontal displacement to the vertical displacement of the sample).

³The induced current produced in the conductor always flows in a direction such that it opposes the change that is producing it (*Lenz's law*) and hence produces a forces in the opposite direction to the motion of the arm.

- ❖ The frequency of oscillation of the motor is 1375 rpm when powered by a 50 Hz supply. When geared down by the 7.5:1 gear results in the sample arm oscillating at a maximum frequency of approximately 3 Hz. By using an inverter the frequency of the motor's power supply and therefore the frequency of the sample arm oscillations can be controlled. For this system it was decided that the sample arm should oscillate at a frequency of 2 Hz.
- ❖ The amplitude of oscillations is 37.5 mm which is the distance between the centers of the pickup coils as shown in Fig. 6.7.

6.3.3 Coil Cooling

Due to the close proximity of the reactor heating it is necessary to cool the coils to ensure that they remain at a low temperature. This is achieved by mounting the coils in an aluminum block through which cooled water is pumped. In Fig. 6.11 the cooling block is labeled while Fig. 6.12 shows the pipes that supply the cooled water to the block. The water is cooled by a water bath which is described in Sec. 3.5.6.

6.4 Electronics

Fig. 6.13 shows the path of the signal from the coils to the computer. All signal carrying cable is coaxial cable with the shielding grounded.

6.4.1 Amplification

In order to provide the best possible signal to the data acquisition card the signal is amplified and filtered before being sent to the computer. The amplification takes place in two stages, one before and one after the filter. The first amplifier stage amplifies the signal by a factor of 10 and the second stage by a factor of 2 making the overall amplification $A \times B = 20$. This results in a signal whose maximum varies between 0V and 5V depending on the sample and field strength. The amplification is performed by two Tektronics AM502 differential amplifiers.

6.4.2 Filtering

The raw signal recorded from the pickup coils is made up of three main components. The component of interest is the signal induced in the coils by the moving sample. The other two components are firstly 50Hz noise caused by the reactor heating and by the current passing through the magnet coils and secondly low frequency noise caused by the magnet. Although the 50Hz noise should not affect the signal from the sample, since it is separated by frequency and the time domain recording is not used directly for any calculations, it was decided to low pass filter the signal prior to recording it. The reason for this is that the time domain signal is displayed on the user interface and may also be used to as a means to determine if the system is working correctly (by noting the signal shape). The low frequency noise produced by the magnet may not be filtered out since it is in the same frequency range as the signal however this did not cause any problems since the level of this noise is low enough to ignore (See Sec. 8.3 for more details).

The filter is a combination of 50Hz notch filter and a 4th order low pass filter. The low pass section of the filter has its 3dB point at 15Hz and was designed to have Butterworth characteristics. The filter design was taken from [34]. Fig. 6.13 shows where the filter fits into the signal path and Fig. 8.7 on page 96 contains actual recordings showing the effect of the low pass filter on an un-amplified signal. Fig. E.2 in the Appendix shows the circuit diagram of the filter.

6.4.3 Synchronization with Sample Movement

As a further attempt to improve the signal to noise ratio (SNR) the acquisition of the signal was synchronized with the sample movement to enable the averaging of successive data captures. This method was later discarded since the SNR was acceptable and taking multiple captures for averaging increased the measurement time by too large a factor.

Note that without some synchronization or a reference signal it is only possible to find the amplitude of the sample magnetization and not its direction since the direction information is taken from the phase of the signal.

In this system the samples to be tested are of known type and therefore the direction of magnetization can be assumed.

University of Cape Town

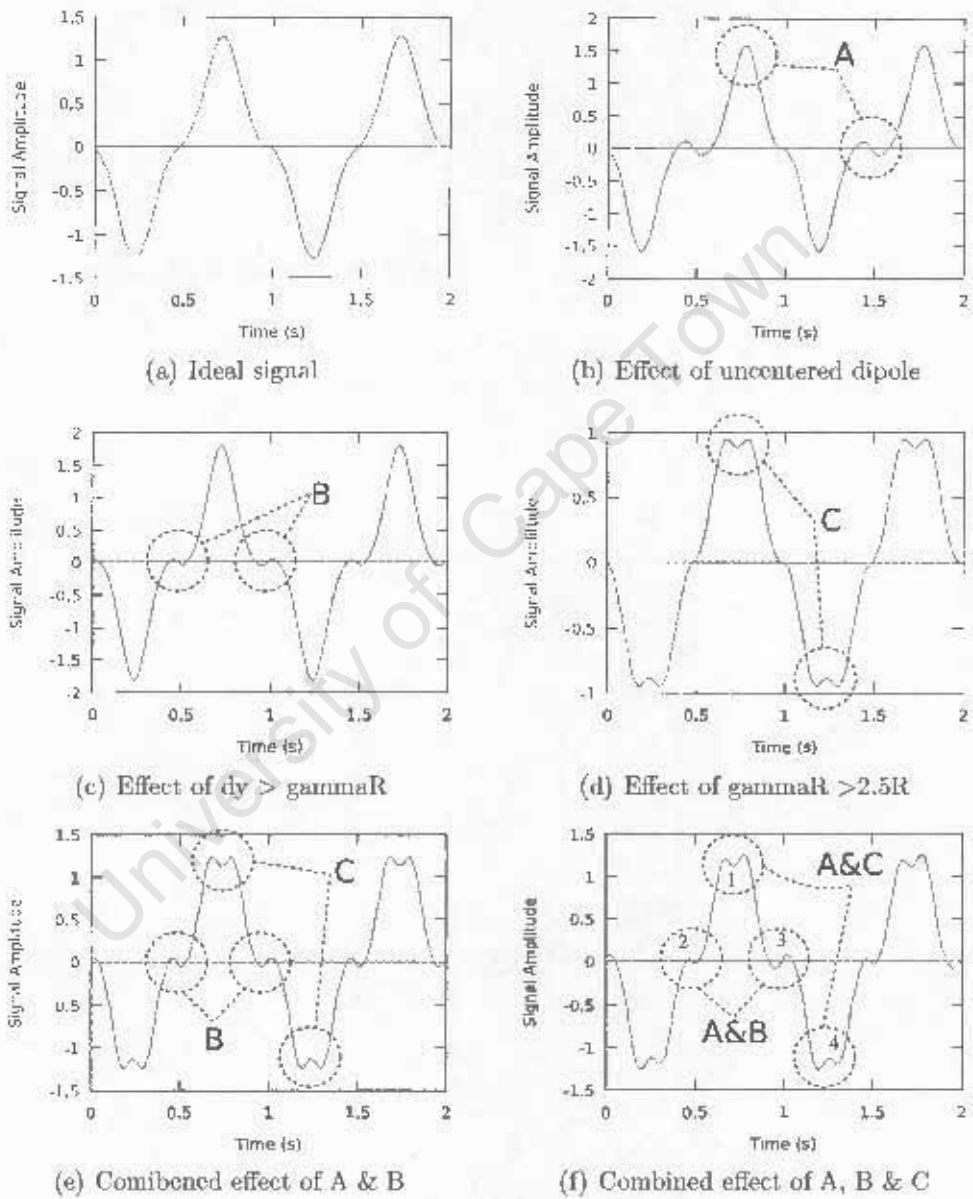


Figure 6.9: Simulated signals

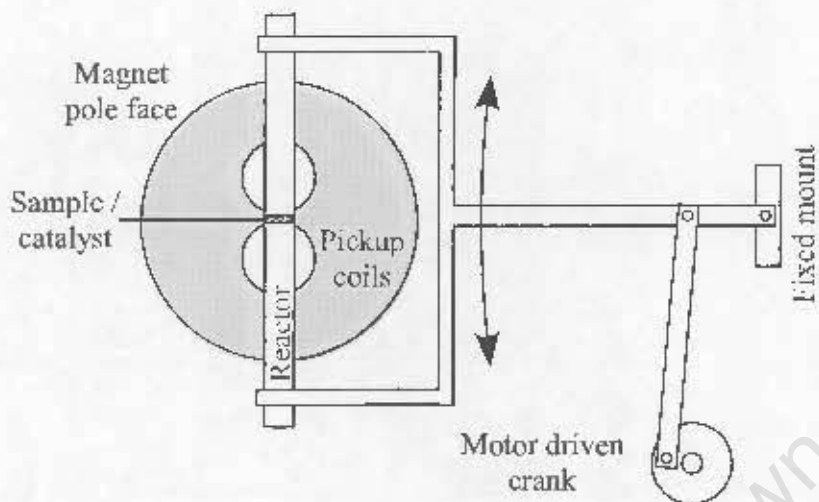


Figure 6.10: Reactor movement

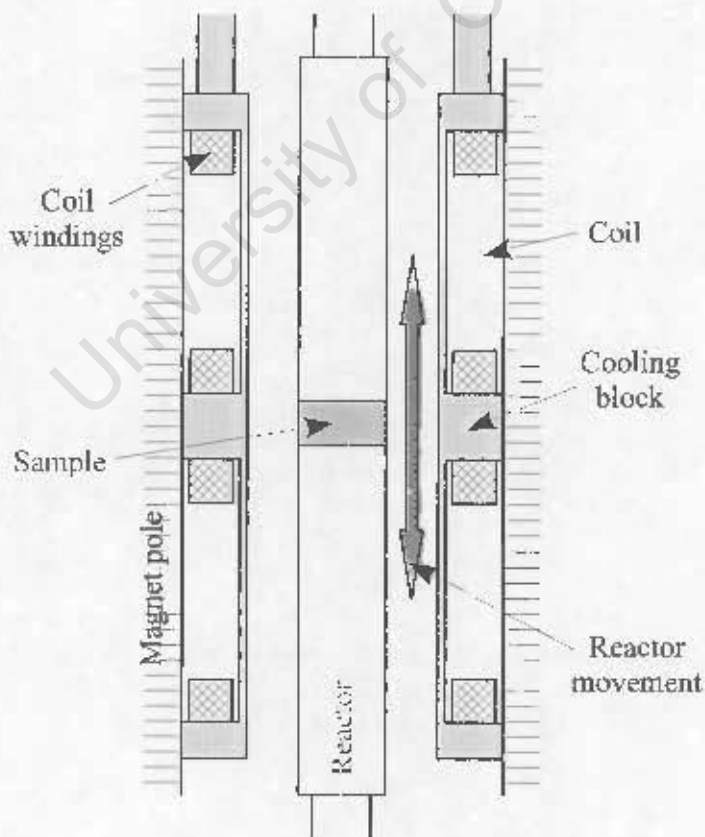


Figure 6.11: Physical layout side view (sliced through center).

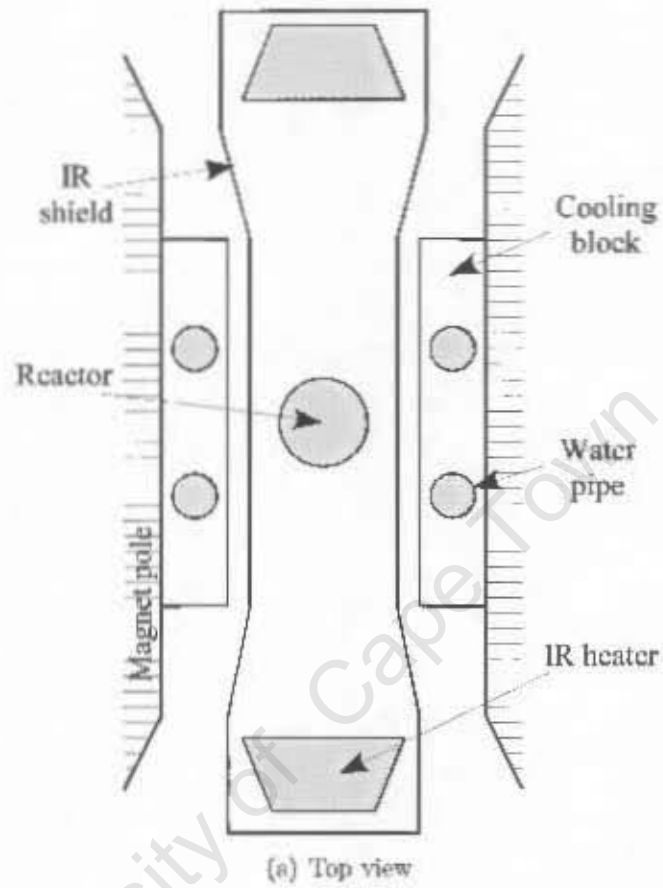


Figure 6.12: Physical layout top view

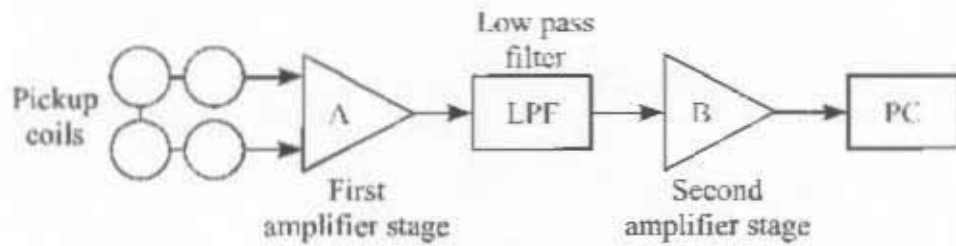


Figure 6.13: Signal Conditioning

Chapter 7

VSM Software Design



Computer science is no more about computers than astronomy is about
telescopes.

- *E. W. Dijkstra*

7.1 Overview

This chapter describes the design of the software that forms part of the vibrating sample magnetometer instrumentation system. Although this software was integrated with the rig software to form a single application, it is described separately since its functionality is completely distinct from the rig software.

In this chapter we start by looking at the software specifications and the design process before moving into the actual design and implementation. Since the controlling of the magnet is part of this software there is a section describing it. At the end of the chapter there are two short sections each describing an additional application. These were written to perform additional functions, namely edit the data recorded by this software and provide a means of writing and error checking recipes.

7.2 Specification

The VSM software is responsible for performing magnetic measurements on the sample using the hardware described in the previous chapter. This means that at the most basic level it has to be able to control the magnet and measure the voltage induced in the pick-up coils by the moving sample. This would allow the user to plot the hysteresis curve of a sample but would make it a very laborious task. It is therefore desirable for the software to perform automated measurements requiring as little user intervention as possible. The module should therefore be able to do the following:

- ❖ Communicate with and control the magnet.
- ❖ Control the movement of the sample.
- ❖ Measure the voltage induced by the sample in the coils.
- ❖ Generate a single number indication of the sample magnetisation for a specific applied field based on the recorded signal.
- ❖ Provide automatic and manual control of the measurement procedure.

- ❖ Enable the user to customise the automated measurement sequences.
- ❖ Record all relevant data to disk.

7.3 Design Process

In designing this application, the exact specifications were not well known when development started. This made it impossible to perform a once off system design. It was thus necessary to design the system from the bottom up, implementing and testing each stage and slowly building the full application as the system became better understood.

The following steps were taken in implementing this application:

1. Develop and test drivers to communicate with the magnet.
The drivers for the magnet were built based on the instruction set laid out in the magnet documentation.
2. Build an initial test application to give manual control over all aspects of the system.
This initial application allowed the system to be experimented with in order to get an understanding of how the system would work and provide a platform from which to develop the next prototype.
3. Iteratively develop and test successive versions of the software, refining it at each stage.
This iterative process of prototyping proved very successful in allowing the understanding of how the system would eventually operate to develop along with the application. It also provided a good platform from which to test and improve the hardware systems.

7.4 Software Layout

The VSM control module is designed in the same way as the modules described in Sec. 3.5. Like the Analog In and Counter Out modules, this module also contains three loops. One loop processes user interface events, another provides the application level functionality and the third

contains external functionality. Fig. 7.1 gives a graphical overview of the VSM module showing the three different loops and some of the functionality that each performs as well as the communication channels that exist between the loops.

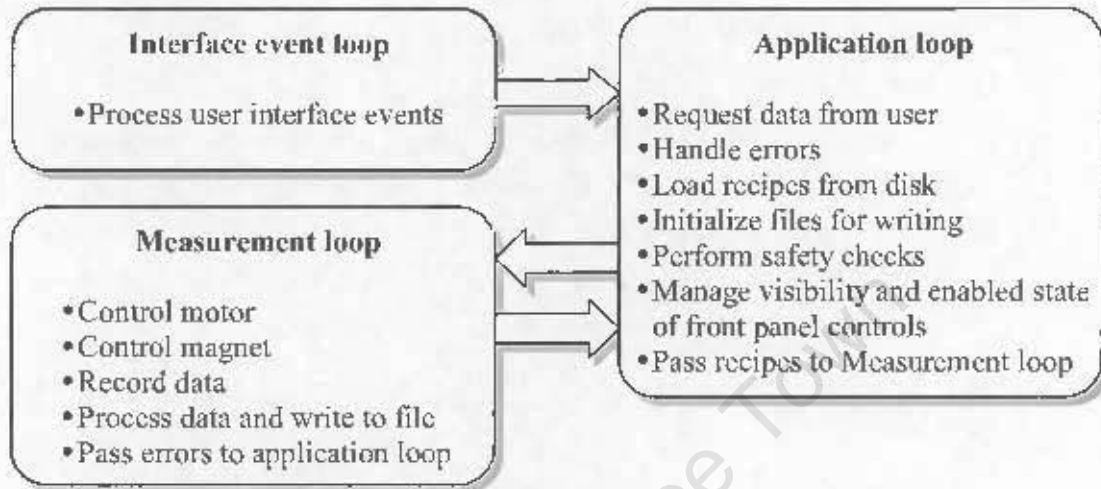


Figure 7.1: VSM top level diagram

The communication between the loops is performed using message queue's (see Sec. 3.4.2 regarding message queue's) which are consistent with the rig software design and made integration of the two applications very simple.

Fig. 7.2 shows an example of how a task would occur in the software. The functionality shown in both Fig. 7.2 and Fig. 7.1 is described further on in this chapter.

7.5 Magnet Control

As the specifications for this software state, the magnet should be controlled from the software interface. Although the magnet may be operated without the aid of a computer, the computer provides the user with a single point of control and allows the automation of the measurement process. The magnet is operated in *field control* mode (as opposed to *current control*) and the field is measured by means of a hall probe situated in the center of the pole face. Communication with the magnet is performed by using the RS-232 protocol via the computers serial

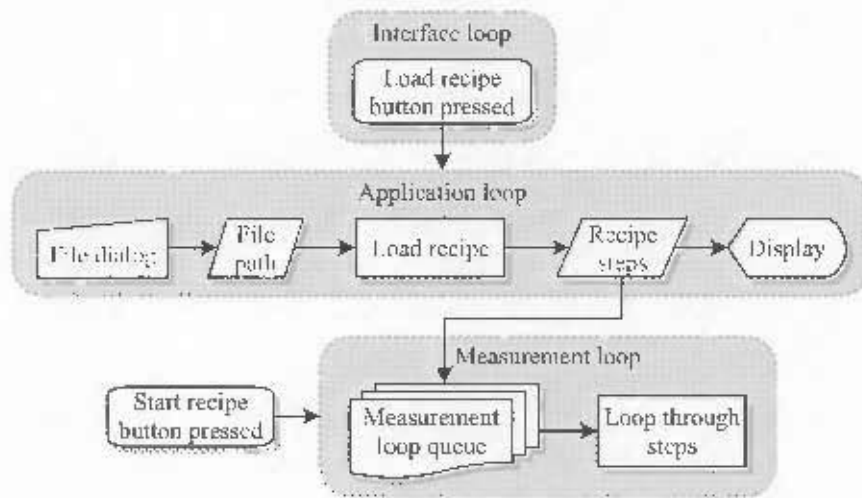


Figure 7.2: VSM flow example

ports. The basic commands used to communicate with the magnet are as follows:

- ❖ Set mode to field control / current control.
- ❖ Set power on / off.
- ❖ Set field / current set point.
- ❖ Read field / current value.

Advanced commands allow reading of the magnet error locks and state information, programming cycles and clearing errors. Since no software was provided with the magnet it was necessary to write a full set of drivers based on the information in the magnet documentation.

Initial testing of the magnet revealed the built in control of the field to be very poor producing large overshoots and oscillations. Since these oscillations produce subsidiary hysteresis¹ within the sample causing inaccurate measurement of the overall hysteresis it is necessary to implement some method of smoothing the control to ensure that the field, once increasing does not decrease until it reaches the maximum field and visa-versa.

This was achieved by exponentially stepping the set point from the current field to the desired field. The following code shows the algorithm that performs this smoothing.

¹This effect is described in Brailsford's book *Magnetic Materials* (1960) [6].

SP = Desired / final set point.

FIELD(n) = Intermediate step to get to SP.

FIELD(0) = The initial field.

sharpness = Sharpness of curve, $0 < \text{sharpness} < 1$. The smaller the value of the sharpness, the smoother the transition between the current field and the set point becomes and the number of intermediate steps required increases.

while the intermediate step is further away from the set point than 0.002 calculate the next set point as a fraction of the difference between the current set point and the final set point. Ensure that the minimum step size is 0.002.

n=0

WHILE (SP-FIELD(n) > 0.002) {

step = (**sharpness**)*(SP-FIELD(n)) # calculate the magnitude of the next intermediate step.

IF (**step** >= 0.002) **THEN** # check that the step is not less than 0.002.

FIELD(n+1) = **FIELD(n)** + **step** # calculate the next set point by adding the step to the previous set point.

ELSE

FIELD(n+1) = **FIELD(n)** + 0.002

n++

}

FIELD(n+1) = SP # set the last set point to the value of the final set point.

The result of this algorithm is shown in Fig. 7.3 for various different sharpness values where the initial field is set to zero ($\text{FIELD}(0)=0$) and the desired set point is set to one ($\text{SP}=1$). Notice that the step size is limited to a minimum of 0.002. The reason for this is that 0.002 is the smallest change in set point that will produce a change in field.

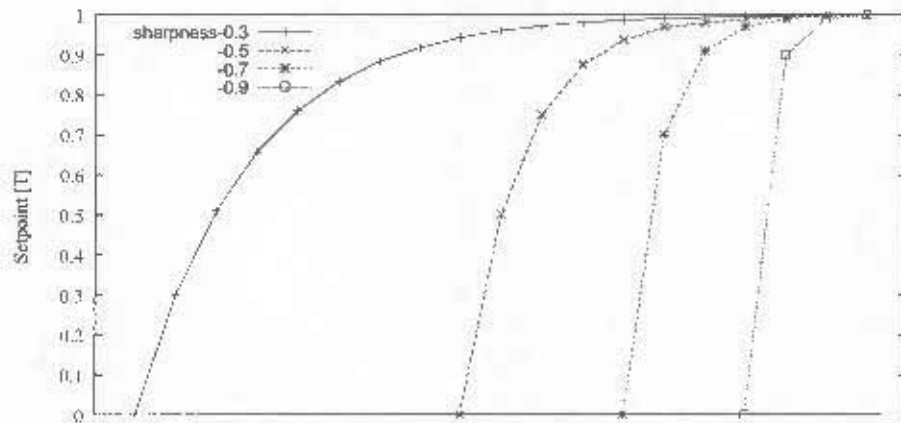


Figure 7.3: Field control smoothing

This algorithm alone was not sufficient to prevent any oscillations since the field would obviously oscillate about each of the sub-set points. In order to prevent this oscillation the magnet was not permitted to settle at each intermediate point but instead, as the field reached the set point, the set point was changed to the next set point. So the process is as follows:

1. Set field to first sub set point.
2. As the field reaches the set point, and overshoots it, set field to next sub set point.
3. Repeat step 2 until the final set point is reached.

Fig. 7.4 shows the final result of smoothing the field control compared to the built in control.

7.6 Data Acquisition and Processing

The following data I/O is required by the VSM software:

1. Recording of the signal from the coil (analog input).
2. Control of the motor used to oscillate the sample (analog out).

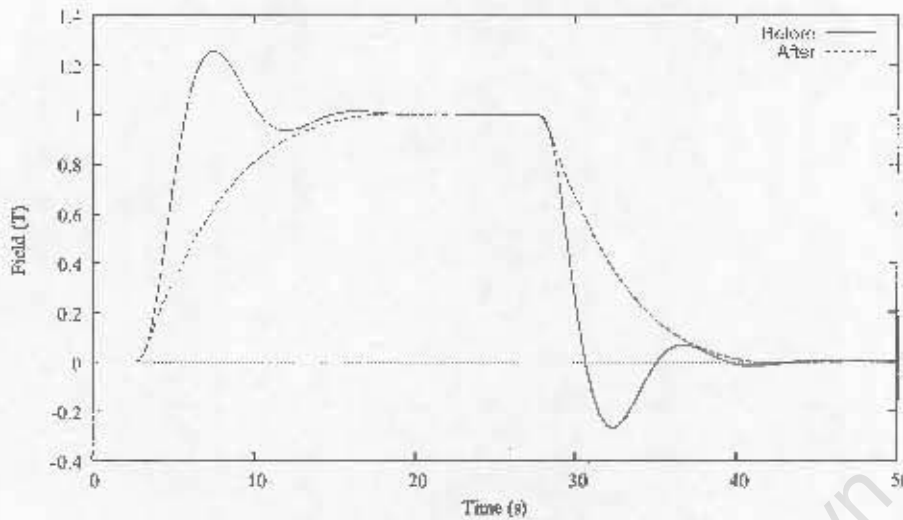


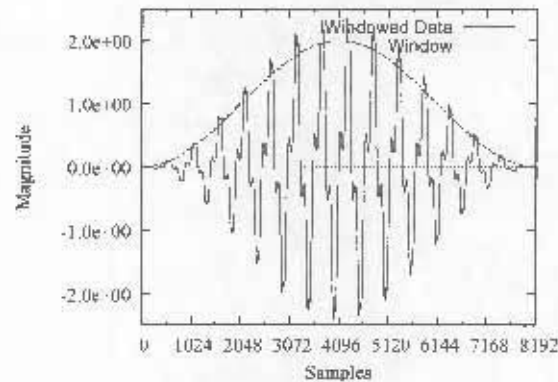
Figure 7.4: Field control comparison, the effect of smoothing.

Unlike the rig modules which rely on dedicated DAQ modules for their data I/O, the VSM software preforms its own I/O.

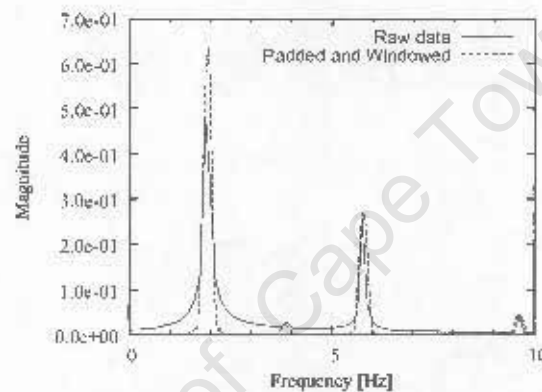
The signal from the pickup coils is recorded at a rate of 1024 samples per second. Although the frequencies of interest in the signal are low, less than 10 Hz, the resolution was kept high in order to be enable pinpointing of noise sources and also get good time data for documentation. In the future the sample rate may be decreased to as low as 20 Hz (Nyquist rate) without significant impact on the accuracy of the results providing that there are sufficient samples near the fundamental frequency to enable its magnitude to be accurately calculated.

In order to improve the signal to noise ratio of the signal (the noise being due to random noise at low frequencies) and to improve the actual frequency resolution, 8 seconds of data are recorded for each data point. This data is then windowed using a scaled hanning window and then zero padded to further improve the frequency resolution. Fig. 7.5 shows the windowed data and the effect that padding and windowing has on the power spectral density of the signal.

Once the data has been windowed and padded, the power spectral density (PSD) of the time data is calculated. From this it is possible to obtain an indication of the magnetization of the signal since the PSD shows the power present in the signal at different frequencies. By assuming that the signal of frequency $\sim 2\text{Hz}$ is solely due to the sample, the power of the



(a) Windowed time data



(b) Effect of padding and windowing

Figure 7.5: The effects of windowing and padding

signal at that frequency is then a measure of the sample's magnetization. This is the value that is used to plot the hysteresis curve of the sample. Clearly this graph is therefore not a true BH curve since this value is not the field generated by the sample. However, since the relationship between the field and this value is linear, and since the actual field value is not required by this system, this graph is sufficient (the actual value can be found by appropriately calibrating the system).

Fig. 7.6 shows a typical sequence of events for generating a single data point including the displaying of data and writing data to disk.

7.7 Configurable Measurement Sequences

In order to allow the user to run multiple different measurements on the system and to be able to customise these measurements it was decided

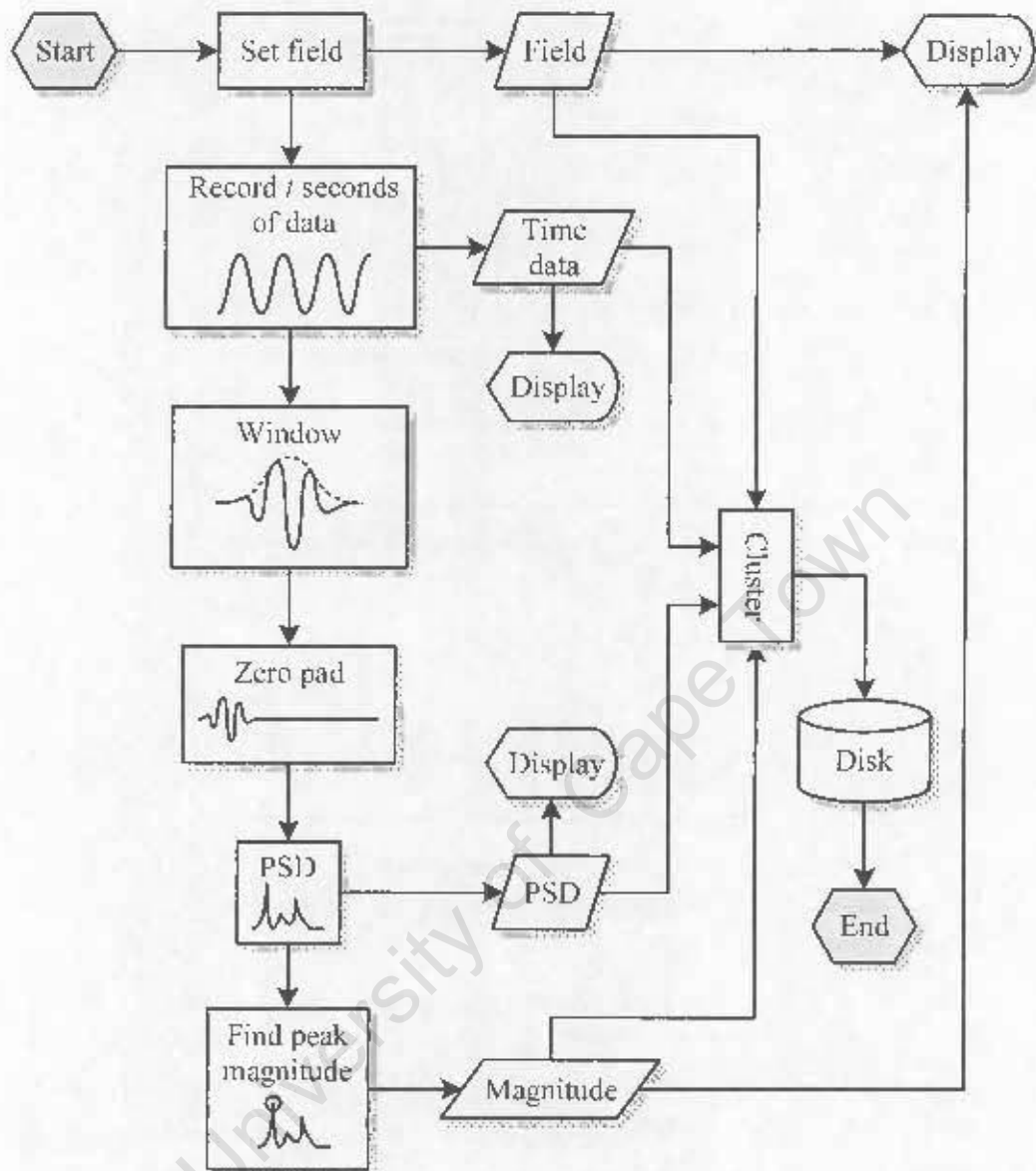


Figure 7.6: Typical recording sequence

to develop a recipe system. This system divides a measurement into a set of discrete steps which can be executed in any order. The user combines these steps to form a measurement sequence that is then saved as a recipe. When a recipe is loaded and activated, the steps are executed one after another.

Each step in a recipe is made up of the step name and a list of arguments for that step. The arguments are enclosed in brackets at the end of the step name and multiple arguments are separated by a comma.

Valid steps and their arguments are as follows:

Basic Steps

❖ *Initialize()*

Initializes the system, clearing any data from any previous measurements and opens a file in which to store the data recorded during this measurement.

❖ *Calibrate(' ' / 'daq' / 'field')*

Calibrates the system in three possible ways. If no argument is given this step performs a full calibration which involves a null recording, from which any DC offset can be found, and finding the maximum field possible, F_{max} ². The *daq* argument performs only the null recording calibration and the *field* argument only performs the maximum field calibration.

❖ *SetField(field)*

Sets the magnetic field to the value supplied by the argument in Tesla ($-F_{max} < \mathbf{field} < F_{max}$). This step uses the smooth field algorithm described in Sec. 7.5.

❖ *SetMotor(frequency)*

The motor is driven by an inverter that converts its 50Hz single phase supply to a variable frequency three phase supply used to power the motor which oscillates the reactor. Since the actual speed (angular velocity) of the motor depends on the motor specifications it was decided that the argument for this step should be the frequency of the inverter ($0\text{Hz} < \mathbf{frequency} < 50\text{Hz}$). The speed of the motor may be derived as follows.

$$\omega = \frac{f}{50} \times (\text{motor speed at } 50 \text{ Hz})$$

At the time of writing the motor on the system had a maximum speed of 1375rpm (22.91 Hz) at 50Hz and a gear of ratio 7.5:1. This means that the final frequency of oscillation of the sample is given by:

$$f_{\text{sample}} = \frac{f}{50} \times \frac{22.91}{7.5}$$

²The maximum field varies depending on the pole gap. If one sets the field above its maximum attainable value, for a particular pole gap, the magnet attempts to reach the desired field but ends up generating an over current error and shutting down. Since this would ruin a measurement it is desirable to avoid this by limiting the maximum settable field.

The frequency of oscillation can therefore vary between 0 Hz and 3 Hz.

❖ *Record(x seconds)*

This step records and processes x seconds of data before writing it to disk. This process is described in the previous section (Sec. 7.6).

❖ *Wait(x seconds)*

Waits x seconds before continuing to the next step. If $x \leq 0$ this step displays a dialog and waits for the user to acknowledge the dialog, by clicking a button on it, before continuing to the next step.

❖ *End()*

This should be the last step in a recipe. It ensures that the magnet is shutdown and that the motor is off, closes the file for writing and writes a summary file of the measurement.

Advanced Steps

❖ *Step(X, Y, Z, B)*

This step is expanded into a series of *SetField* and *Record* steps in such a way that the field is stepped from X to Y (not inclusive) in Z steps and B seconds of data are recorded at each step.

Example Recipe

```
Initialize()  
Calibrate(daq)  
SetMotor(40)  
Step(0,2,10,8)  
SetField(2)  
Record(8)  
SetMotor(0)  
End()
```

After initializing, calibrating and starting the motor, this recipe steps the field from 0T to 1.8T in steps of $\frac{2-0}{10} = 0.2T$ and records 8 seconds of data after each step. Finally the field is set to 2T and another 8 seconds of data are recorded before ending the measurement.

Fig. 7.7 shows how a recipe should be structured.

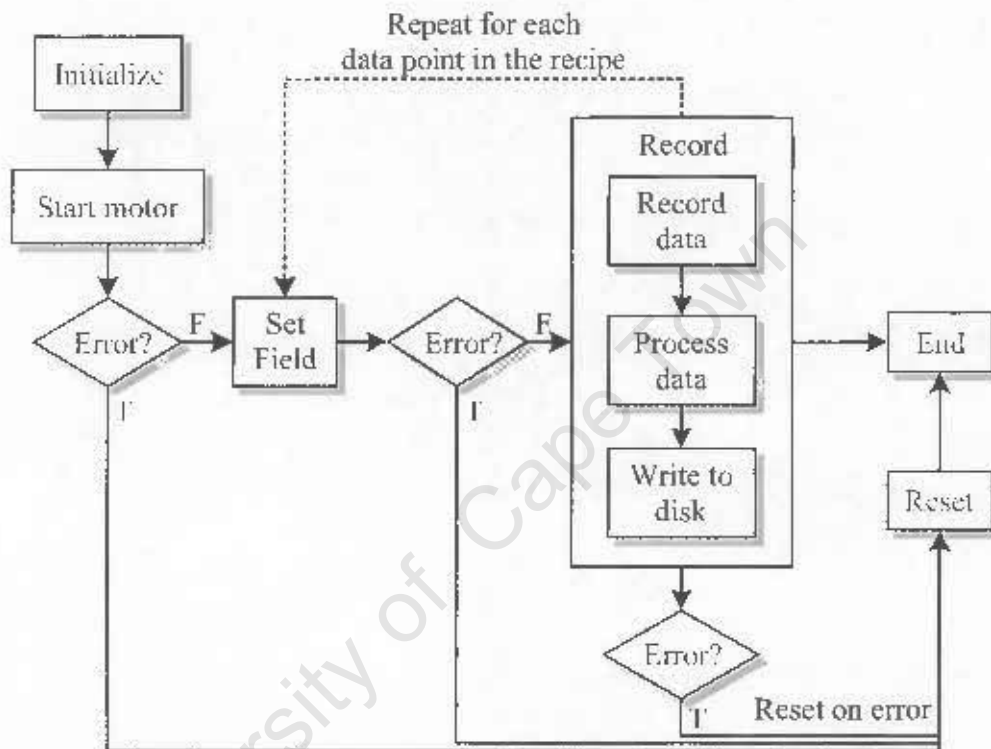


Figure 7.7: Recipe flow

7.8 Interface Design

The main elements of this applications interface as shown in Fig. 7.8 are:

- ❖ a number of graphs used to display recorded and processed data ((labeled *Time Data* and *Hysteresis Curve* respectively on Fig. 7.8) as well as a graphical representation of the steps in the currently loaded recipe (labeled *Profile*).
- ❖ various buttons to control the application including the loading of recipes, starting and aborting a recipe as well as manual control of the magnet, the motor and the recording of data.

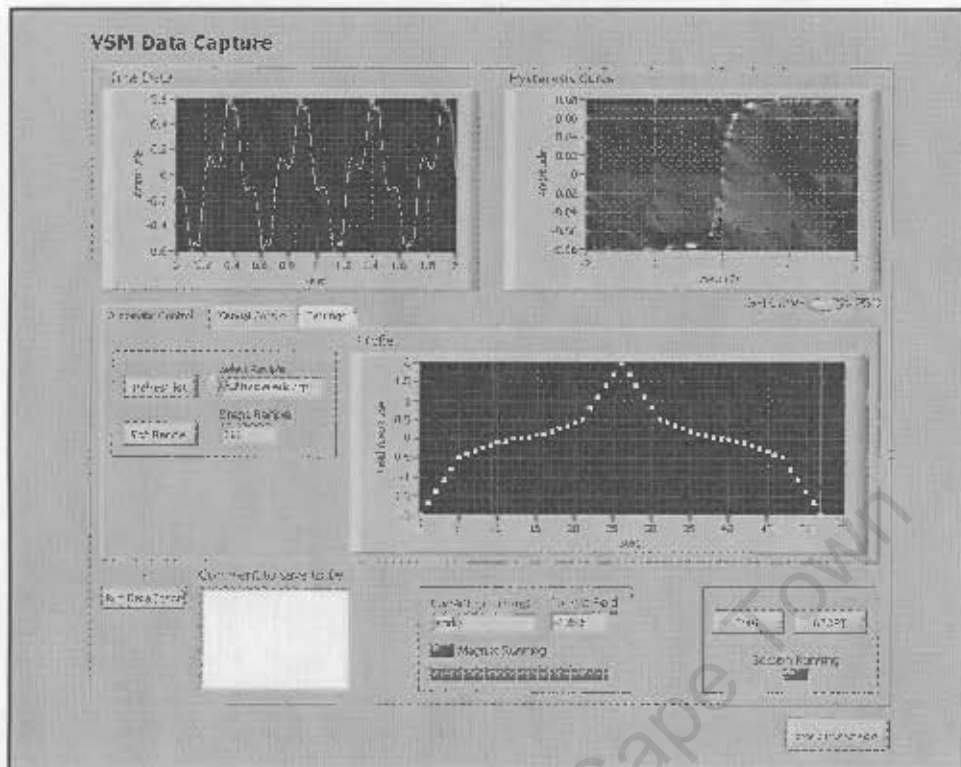


Figure 7.8: VSM module user interface

- ❖ information on the current state of the application and also the magnet.
- ❖ controls allowing the user to view and alter application settings.

In order to fit all these elements onto the interface the graphs showing the PSD and the Hysteresis curve were placed on top of each other with a switch to toggle which one is visible (in Fig. 7.8 the graph showing the hysteresis curve is visible). A tab control was also added with three pages. The first page shows the graphical representation of the current recipe and the controls for loading recipes and running the recipe editor (the active page in Fig. 7.8). The second tab contains the manual controls and is only enabled when the application is idle (not running a recipe). The last tab contains the application settings.

7.9 Error Handling

Possible errors generated by this application are:

- ❖ Application errors: errors that are generated by the application itself such as file read errors.
- ❖ Magnet errors: errors generated by the magnet drivers in response to an error message received from the magnet.
- ❖ DAQ errors: errors generated during data acquisition.
- ❖ Processing errors: errors generated during the processing of the data.

All these errors, except application errors are directly linked to the measurement process and are generated in the measurement loop. All errors are passed to the application loop which processes them accordingly. Most errors would occur during a measurement and be the result of a hardware problem. These errors would therefore mean that the current measurement can no longer take place and that the measurement should be aborted.

In the case of such an error, the measurement is automatically aborted and the user is notified via a pop up dialog. Fig. 7.7 shows that errors are only processed at the end of each recipe step.

7.10 Additional Software

7.10.1 Advanced VSM Data Editor

The VSM data editor gives the user access to the recorded data and allows them to explore the data in detail and edit the calculated portions of the data before exporting any portion of it to a spreadsheet file. The actual time data can be viewed and exported but can not be edited by the user. The PSD can be recalculated from the time data and the hysteresis curve can be replotted using any combination of data points from the PSD.

7.10.2 Measurement Recipe Editor

The recipe editor is a small application that can be launched from the main VSM window. It provides an easy way for the user to create, edit

and save recipes. Since recipes are stored as text files, any text editor could be used to perform these functions. The recipe editor however adds the ability to parse recipes and to plot a graph of the steps in the recipe. Advanced steps³ are expanded into their basic steps and the whole recipe is checked for errors.

University of Cape Town

³see Sec. 7.7

Chapter 8

Results and Analysis



“The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' ('I found it!') but rather 'hmm...that's funny...'"

- *Isaac Asimov*

8.1 Signal Analysis

Fig. 8.1 shows how the signal in the coils varies as the sample sample is oscillated between them. By comparing the shape of this signal to the signals simulated in section 6.3.1 it can be seen that the coils have been placed too far apart i.e. the distance between the coil centres is more than the 37.5 mm calculated in the design (the condition shown in Fig. 6.9d). The sample's oscillations are too large, overshooting the centers of the coil causing the effect shown in Fig. 6.9c. It may also be said that the sample's oscillations are slightly offset causing a small amount of asymmetry in the signal (Fig. 6.9b). Unfortunately since the system is already operational this is not easy to correct but does not have a large impact on the results provided the system is properly calibrated.

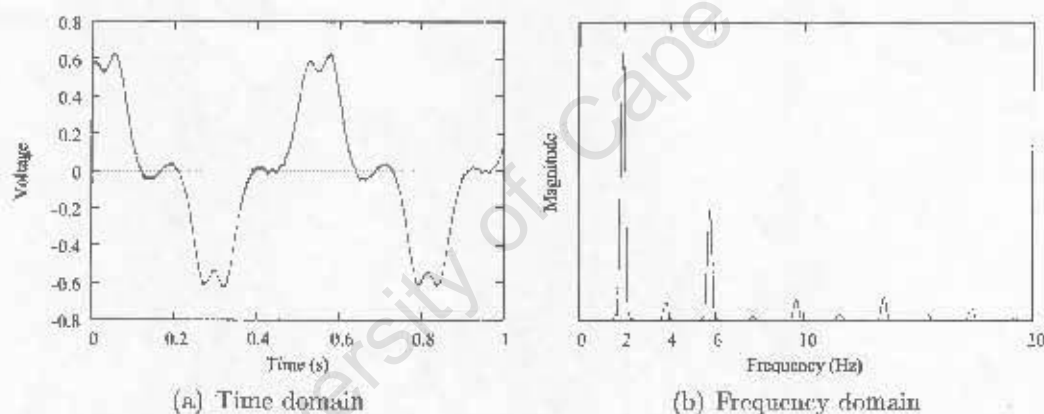


Figure 8.1: Recorded signal from the pickup coils.

Fig. 8.1b shows the PSD of the recorded signal. The signal at approximately 2 Hz is the signal that is of interest since it is the primary harmonic of the signal. Its exact position in frequency is determined by the frequency of oscillation of the sample which is controlled by the motor (see Sec. 6.3.2 - Sample Movement).

8.2 Calibration

In order to calibrate the system a full hysteresis reading was performed on three reference samples of powdered cobalt, each of different mass, placed in the reactor as well as a reading with no sample present. Fig. 8.2 shows

the full graphs from these readings. Theoretically there should be a linear relationship between the mass of the sample and the magnetization.

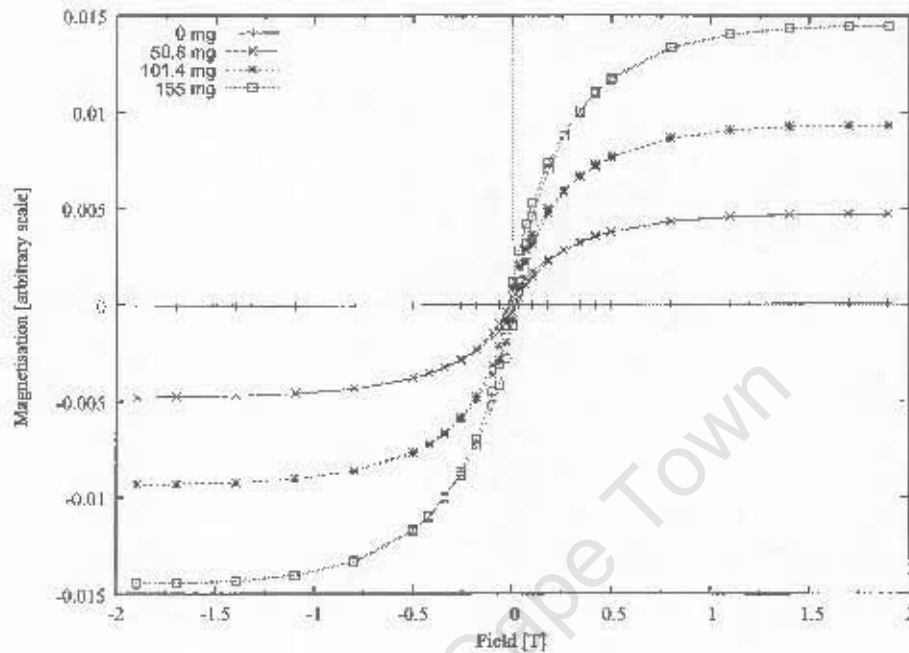


Figure 8.2: Mass calibration hysteresis curves

Fig. 8.3 shows the graphs obtained by extracting the saturation and remnant magnetization from these graphs and plotting them against the mass of the samples. The linear relationship between the mass and the magnetization is quite clear and enables a sample of unknown mass to be measured and its mass determined.

8.2.1 Temperature Effects

Using the rig and instrumentation described in this document, measurements can be taken on heated samples. The effect of temperature on the saturation and remnant magnetization of one reference sample is shown in Fig. 8.4.

8.3 Noise

Major sources of noise that interfere with the signal induced in the coils include the infra-red heating and the magnet. Both of these produce

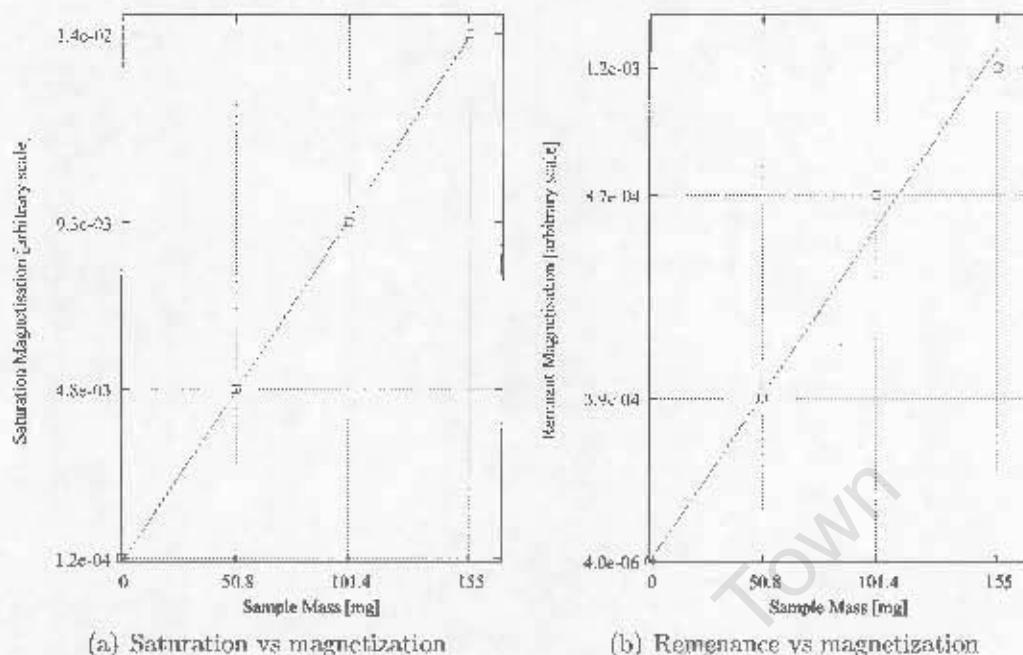


Figure 8.3: Saturation and remnance vs magnetization

large 50 Hz interference as well as some low frequency noise. The infrared heaters are pulsed with 50 Hz from the mains supply which causes portions of the time domain signal to be corrupted. This is illustrated in Fig. 8.5. Notice that the IR heating does not interfere with the low frequencies and hence does not impact the measurements.

As well as large amounts of 50 Hz noise the magnet also produces some noise at low frequencies. This may be due to small changes in the magnet current caused by the magnet's control system. Fig. 8.6a shows how the noise varies as the field is increased. The first plot shows the noise at 50 Hz while the second shows the noise at 2 Hz. Fig. 8.6b shows the PSD of the signal captured by the coils when the magnetic field is at 0.4T and there is no sample movement.

The high frequency effects of both the magnet and the IR heating are filtered out of the signal using the filter described in Sec. 6.4.1. Fig. 8.7 shows both the time domain and frequency domain plots of the signal recorded while a sample is being oscillated between the coils and the field is set to 0.4T field. There is no heating during these captures and the amplification has also been removed since the amplifiers also have some filtering built in. The signal in Fig. 8.7a & c is unfiltered while that in Fig. 8.7b & d is filtered.

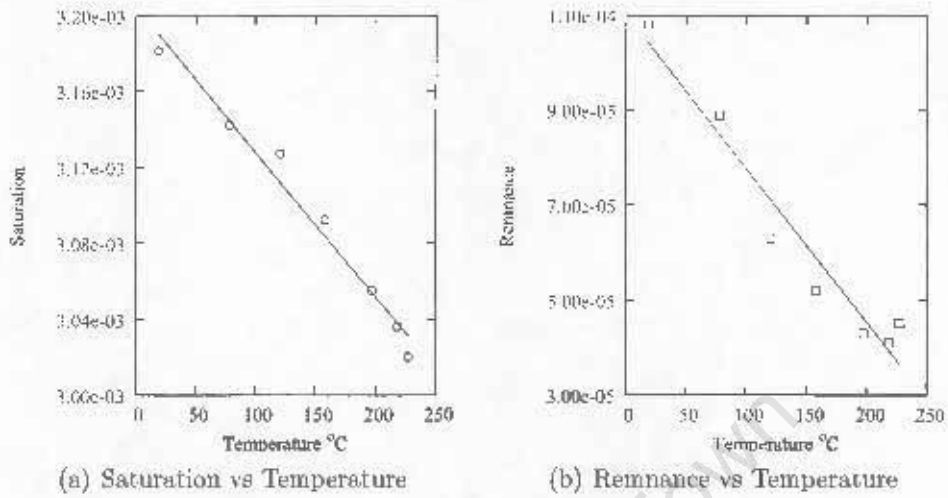


Figure 8.4: Temperature effect on the saturation and remnant magnetization.

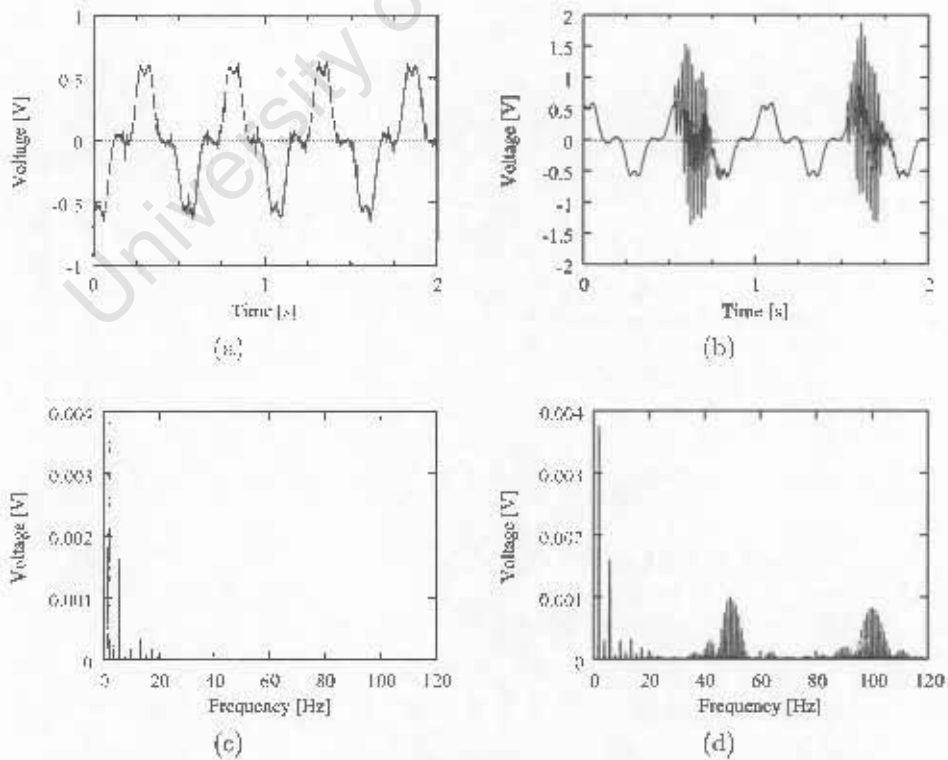


Figure 8.5: Noise due to IR heating

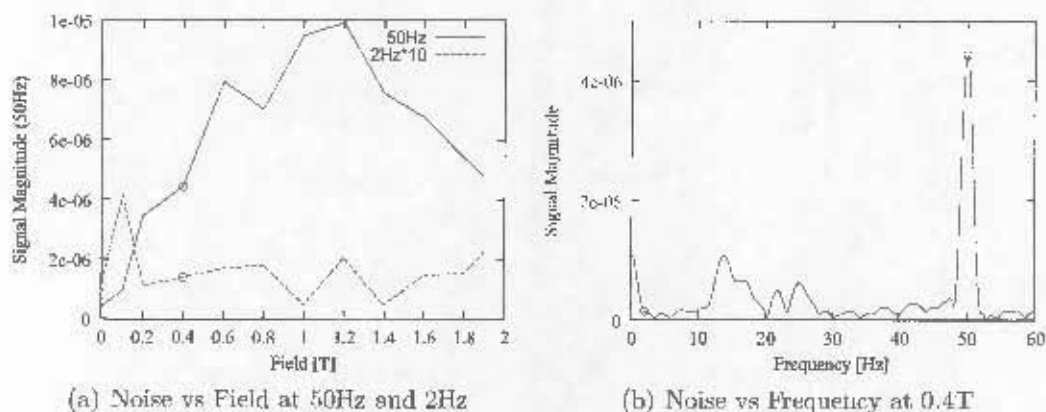


Figure 8.6: Magnet noise

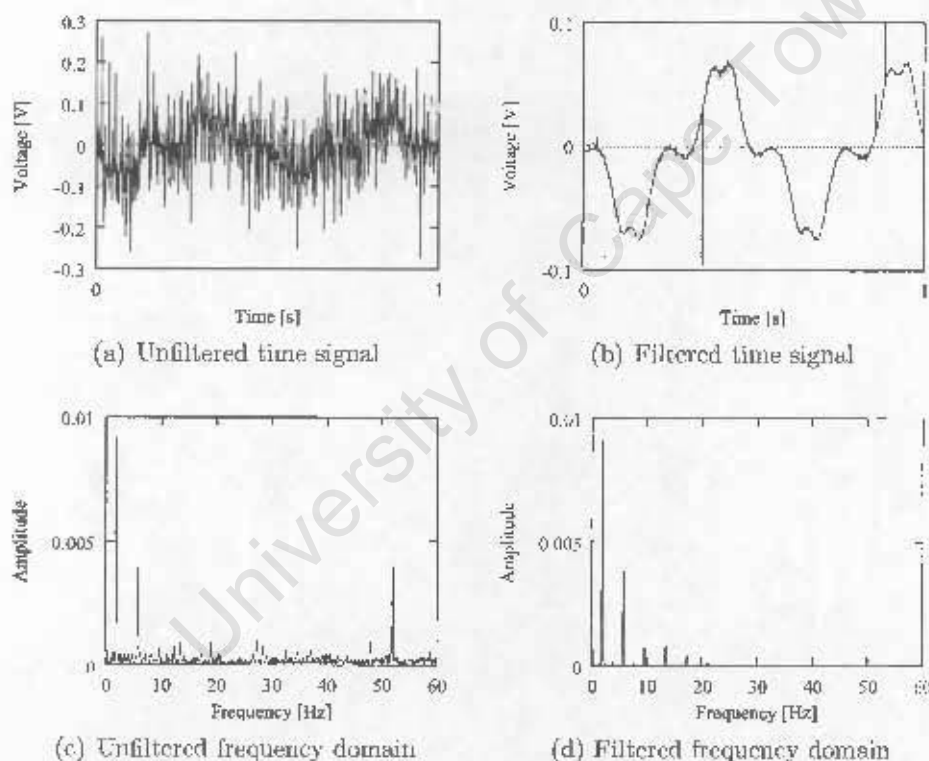
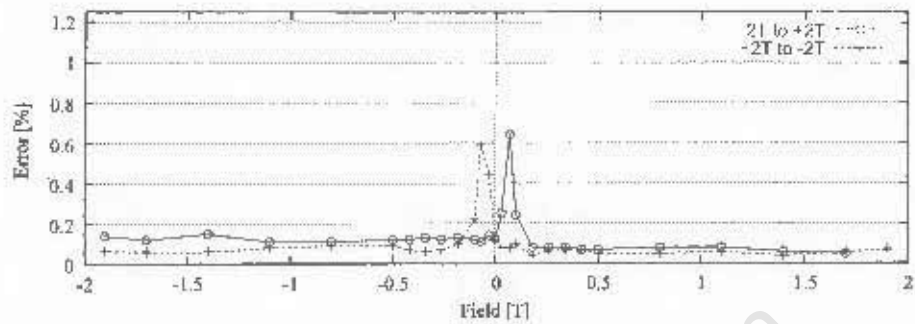


Figure 8.7: The effect of low pass filtering an un-amplified signal.

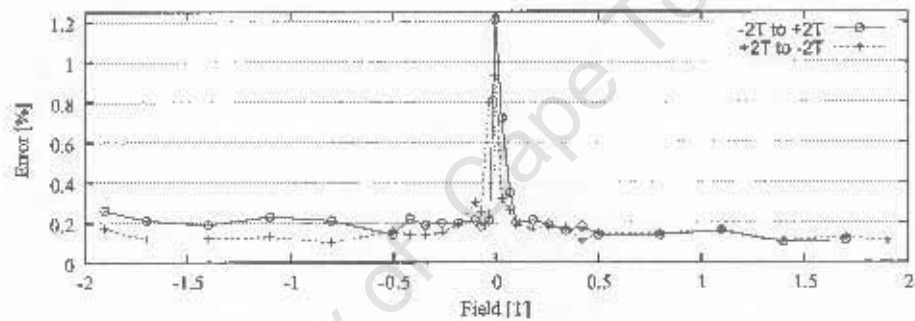
8.4 Measurement Errors

Dr. M. Clacys performed a simple error analysis for the measurements taken from two different catalyst samples. The first sample contained 19 mg of metallic Cobalt and the second sample 75mg. Ten identical, consecutive measurements were taken of each sample and the error calculated as a percentage of the mean. Fig. 8.8 shows the graphs of the error at different field values. There are two graphs for each sample since

the field is varied from -2 T to $+2\text{ T}$ and then back to -2 T . The first graph shows the error as the field is increasing and the second shows the error for a decreasing field.



(a) 75 mg Cobalt sample



(b) 19 mg Cobalt sample

Figure 8.8: Hysteresis error analysis

It is interesting to note some similarity between the graphs of the two samples. It would appear that the error is larger when the field is decreasing. The reason for this is not known but it has been suggested that the magnet may require some time to 'settle'.

Overall a very small error was obtained showing the high degree of reproducibility of the measurements.

Chapter 9

Conclusions

University of Cape Town

A decorative flourish consisting of several overlapping, curved lines that form a central, symmetrical shape, resembling a stylized infinity symbol or a calligraphic flourish.

"A good scientist is a person with original ideas. A good engineer is a person who makes a design that works with as few original ideas as possible"

- *Freeman Dyson*

1. **A unique use of a vibrating sample magnetometer has been developed.**

As stated previously, this is the first use of a VSM to perform in-situ measurements. The VSM system proved to be well suited to take measurements of this kind since no direct contact with the sample is required. This system will provide an additional measurement tool for the University of Cape Town Catalysis Research Unit and allow them to perform new research that was not previously possible.

2. **A software system providing a single point of control for the reactor rig and the VSM instrument was successfully developed.**

The product of this masters is a complete system which at the time of writing had been in use for two months without the need for major modifications or additions. The system allows the operator to control the rig and perform customizable VSM measurements.

3. **The choice of LabView as the development language for the system as well as its future use has been justified.**

LabView provided the necessary tools to enable rapid and flexible development of the system including interface design and management of the computer hardware. The main advantages of LabView in this case are that it provides ready to use toolkits enabling rapid development without the need for custom tools. LabView also decreases the amount of programming overhead by automatically handling multiple threads, error catching etc.

Chapter 10

Recommendations for Future Work

Although it was concluded that the system is complete there are some improvements and additions that may be made.

1. **Perform additional characterization of the system.**

It may prove useful in the future to have a more detailed set of data on the performance and specific characteristics of this system. In particular, calibration using a standard technique would provide information to enable the comparison of this system with other systems.

2. **Provide network access to the system.**

Currently the only way to access the system over the network is to use some virtual network computing (VNC) software such as Windows remote desktop or TightVNC¹. Although such software provides remote access to the software the option of developing a application specifically designed to provide remote access to the system should be considered.

3. **Measurement scheduling.**

Enabling the user to schedule VSM measurements would be a very

¹TightVNC: VNC-Based Free Remote Control Solution

good addition to the software. Currently the operator must start all measurements manually which makes for additional effort on the part of the operator, particularly if measurements need to be taken during the night. Three possible methods exist for developing such a feature.

- (a) The first option is to development of a separate tool that runs concurrently to the current software and passes commands to the VSM module. Such a tool would allow the user to schedule a measurement and then pass commands to load and execute the measurement to the VSM module via the module's message queue.
- (b) Secondly, the VSM module itself could be modified to include the scheduling functionality. The design of the modules described in Sec. 3.5 allows easy addition of functionality to the module. The modification would involve adding another page to the already existing tab structure on the module interface and adding the necessary functionality within new command cases to the application loop.
- (c) This method is similar to the previous method in that the functionality of the schedule execution is added to the VSM module however the creation and maintenance of the schedule is moved to a separate application. This application would allow the user to create new schedules and view and modify existing schedules. These schedules would then be saved to a file which is read by the VSM module. In this way the VSM interface is separated from the schedule interface. Some indication may be included on the VSM interface as to the status of any schedules loaded.

4. Additional hardware redundancy.

In an effort to improve safety and also minimize down time related to computer malfunction it is recommended that additional hardware be added to the system so that in the case of a computer or software failure the system may still be run manually. This hardware would mainly consist of temperature controllers which would have the primary responsibility for temperature control. The addition of these controllers would require some adjustment to the

software as the software would no longer be performing the control but simply communicating with the hardware controllers.

5. Computer control of the MFC's

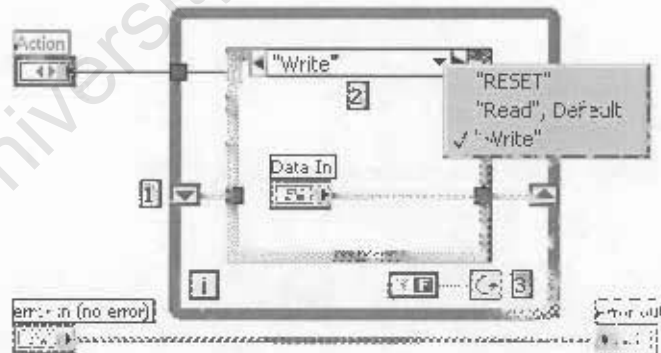
The computer control of the MFC was abolished due to the unstable nature of the Brooks Instruments MFC communication software. In the event of a new version of the MFC control software (produced by Brooks Instruments) becoming available, this functionality should be re-visited and if possible incorporated into the system.

University of Cape Town

Appendix A

Software Source Code

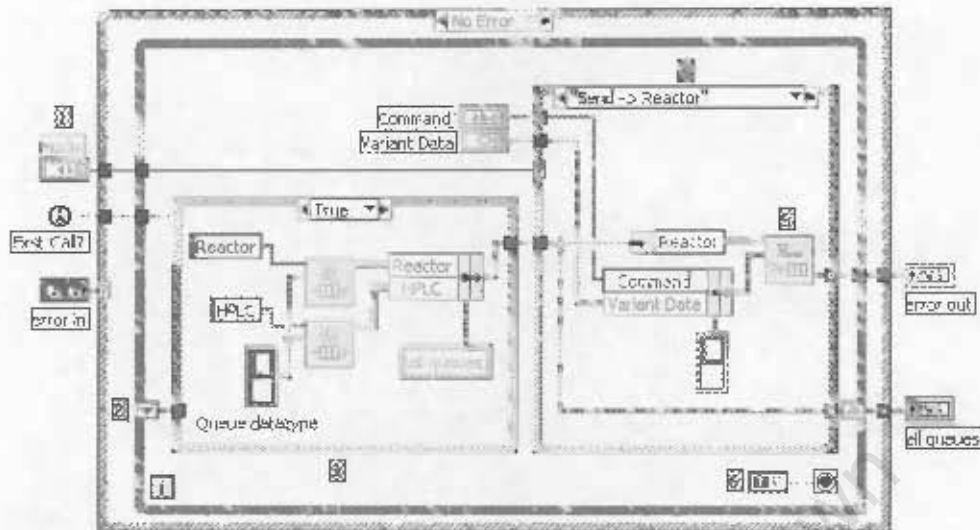
Due to the graphical nature of the source code for the software described in this document it is not practical to include the full code. Therefore these code diagrams show only specific portions of the code and in most cases have been slightly modified in for them to make sense as stand-alone code. The full code is included on the CD accompanying this document and may be viewed using LabView (an evaluation copy may be downloaded from the National Instruments website).



1. Uninitialized shift register used to store data over multiple function calls.
2. Case structure contains code for each function mode; read, write, reset.
3. Conditional statement of while loop wired to force exit after a single iteration.

(Sec. 3.4.2 on page 23)

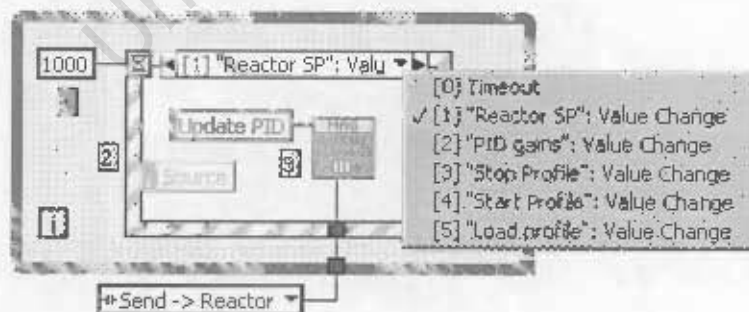
Figure A.1: Functional global implementation



1. Mode selector: 'no-op', 'destroy', 'Send -> module'.
2. Un-initialized shift register used to store queue references.
3. If first call, create queues and store references in cluster.
4. Mode case structure.
5. 'Send -> module' code: bundle data into cluster and push onto relevant queue.
6. TRUE constant wired to loop condition forces single loop execution per call.

(Sec. 3.4.3 on page 27)

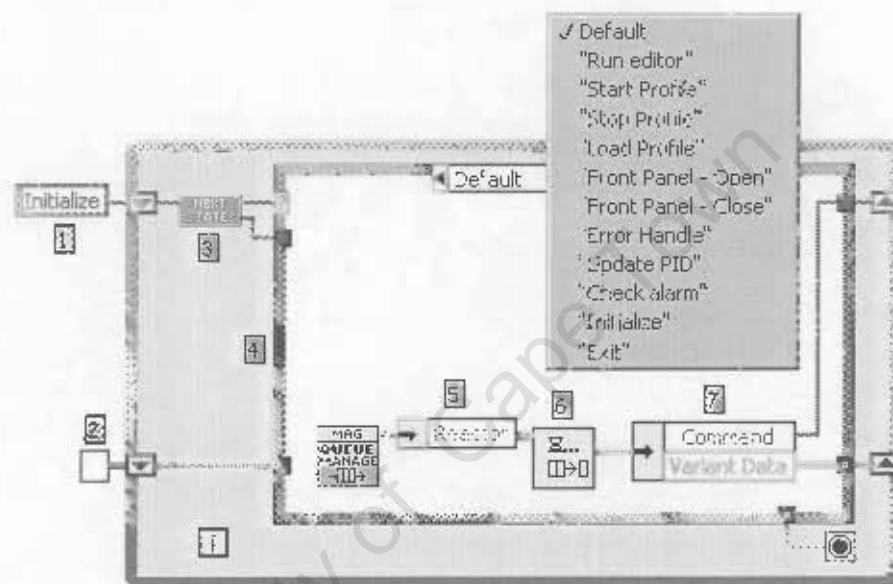
Figure A.2: Simplified queue manager code



1. Interface timeout.
2. Event structure.
3. Code executed when event triggered.

(Sec. 3.5.1 on page 30)

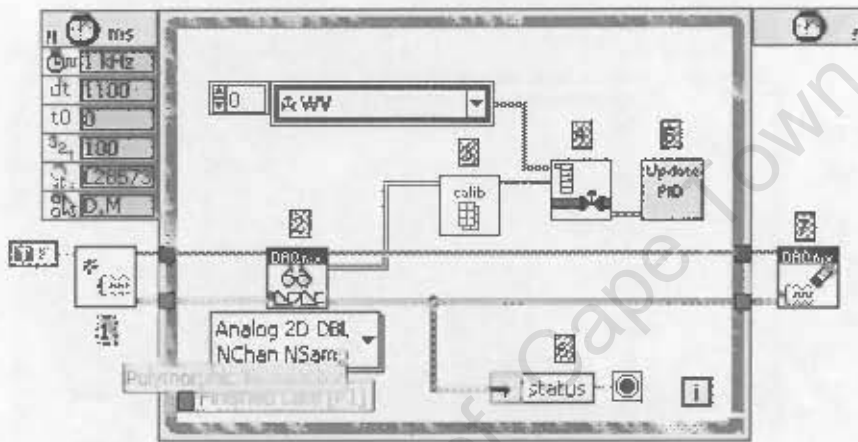
Figure A.3: Simplified interface loop code



1. States to execute on startup.
2. Shift register to pass data between states.
3. State shift register, executed in sequence.
4. Case structure contains all states.
5. Obtain queue.
6. Dequeue next element.
7. Pass command to state queue and data to data shift register.

(Sec. 3.5.1 on page 31)

Figure A.4: Simplified application loop code

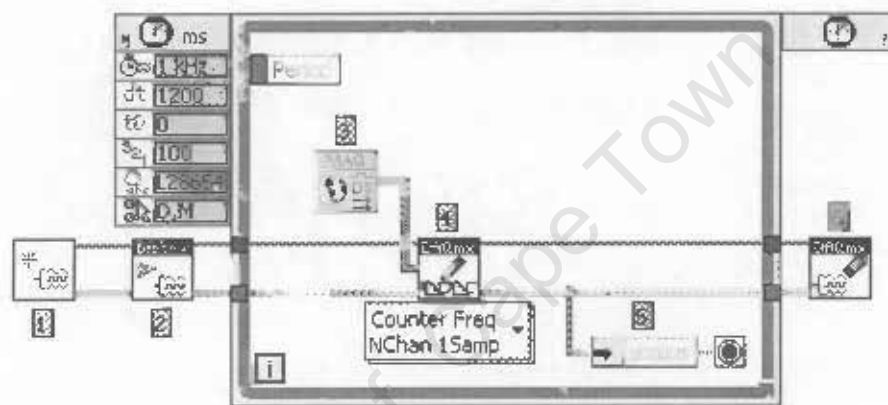


1. Initialize channels.
2. Read data.
3. Process data.
4. Write data to database.
5. Inform other modules of new data.
6. Exit loop on error.
7. Close channels on exit.

Note: Timed loop used to increase control over loop timing and priority.

(See 3.5.2 on page 31)

Figure A.5: Simplified AI loop code

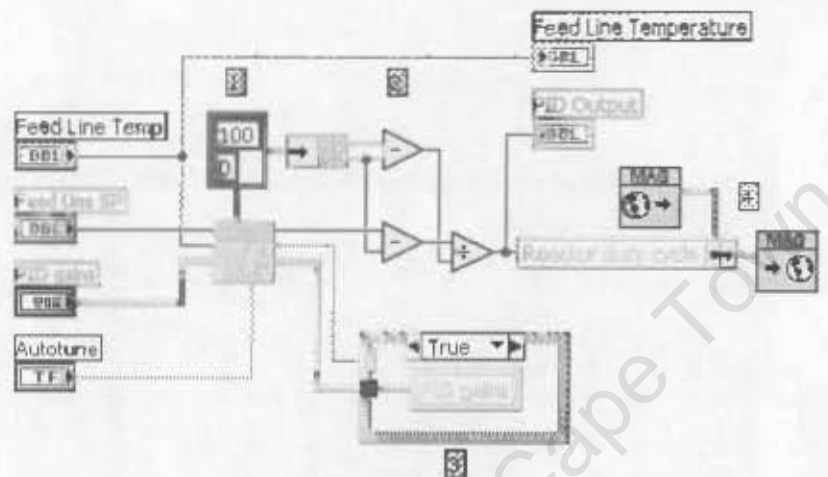


1. Initialize channels.
2. Start counters.
3. Read channel duty cycles.
4. Update duty cycle of each counter.
5. Exit on error.
6. Close channels on exit.

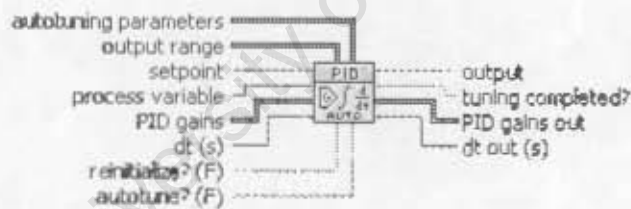
Note: Timed loop used to increase control over loop timing and priority.

(Sec. 3.5.2 on page 31)

Figure A.6: Simplified CO loop code



(a) PID code for basic control.

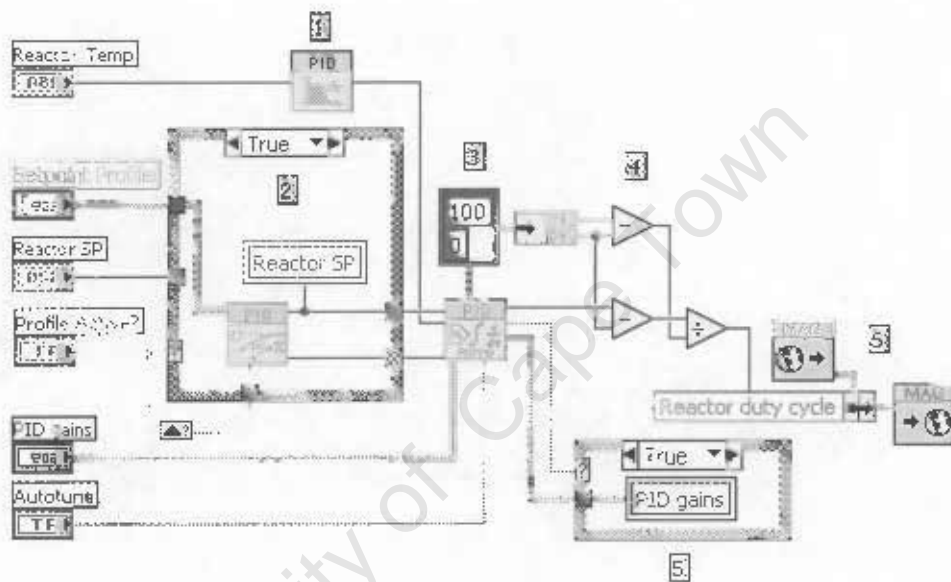


(b) Connection pane for PID function.

1. Calculate the PID output.
2. Convert the PID output to a duty cycle.
3. Update PID gains control after auto-tuning
4. Send the duty cycle to the CO module.

(Sec. 3.5.4 on page 33)

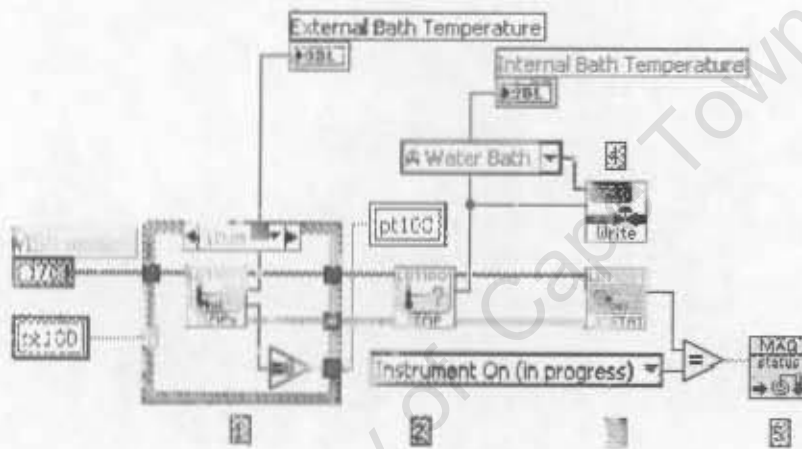
Figure A.7: Basic PID code



1. Filter the incoming temp. signal.
2. If a profile is active, update the set point based on the profile
3. Calculate the PID output.
4. Convert the PID output to a duty cycle.
5. Update PID gains control after auto-tuning
6. Send the duty cycle to the CO module.

(Sec. 3.5.5 on page 35)

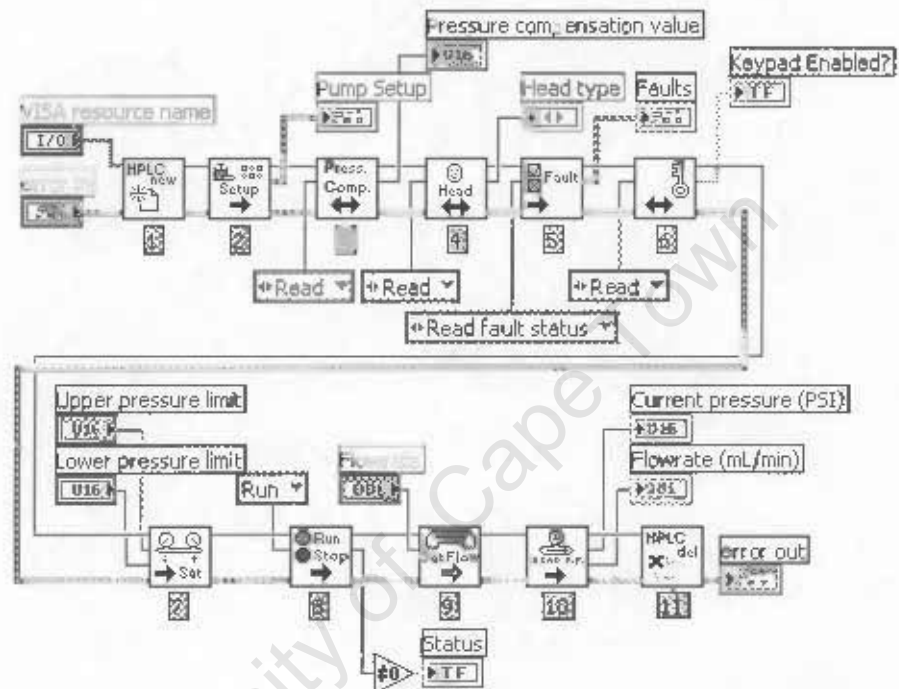
Figure A.8: Simplified code for advanced temperature control



1. If probe connected, read external temp.
2. Read bath temp.
3. Check bath status.
4. Write bath temp to database.
5. Write bath status to global variable.

(Sec. 3.5.6 on page 37)

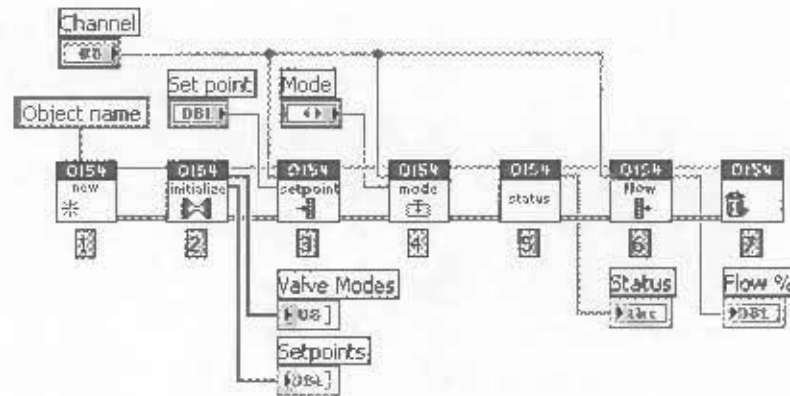
Figure A.9: Water bath code portion



1. Create new HPLC object.
2. Read pump setup.
- 3,4,5,6. Read/write additional settings from/to pump.
7. Write pressure settings to pump.
8. Change pump to run/stop mode.
9. Write flow set point to pump.
10. Read pump flow and pressure.
11. Close connection to pump and destroy HPLC object.

(Sec. 3.5.7 on page 38)

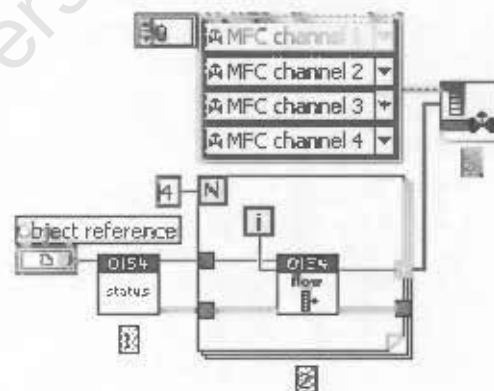
Figure A.10: HPLC functions example usage.



1. Obtain object data.
2. Check input range.
3. Format command to send to pump.
4. Write serial data.
5. Read response from pump.
6. Parse response based on command sent.
7. Update object data.

(Sec. 3.5.8 on page 39)

Figure A.13: MFC methods.



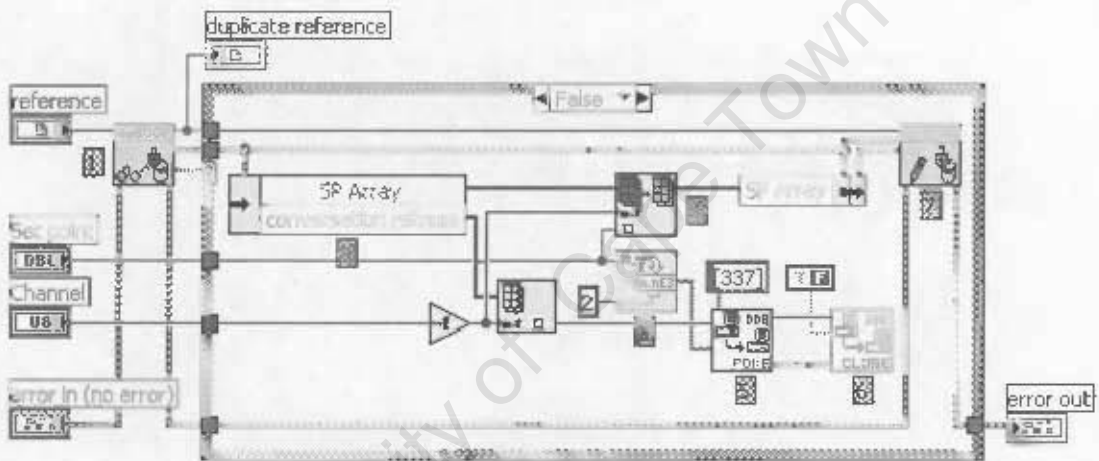
1. Check DDE status (error check).
2. Read flow for each channel.
3. Write flows to database.

(Sec. 3.5.8 on page 39)

Figure A.14: MFC module code portion: read MFC's flow.



(a) Write setpoint method connection diagram.



(b) Write setpoint method code.

1. Obtain object data.
2. Select data to use in this method.
3. Format data to write to DDE.
4. Update Set Point array.
5. Poke DDE data to location.
6. Close DDE conversation if error.
7. Update object data.

(Sec. 3.5.5 on page 39)

Figure A.15: MFC method: write set point.

Appendix B

Magnet Photographs

University of Cape Town

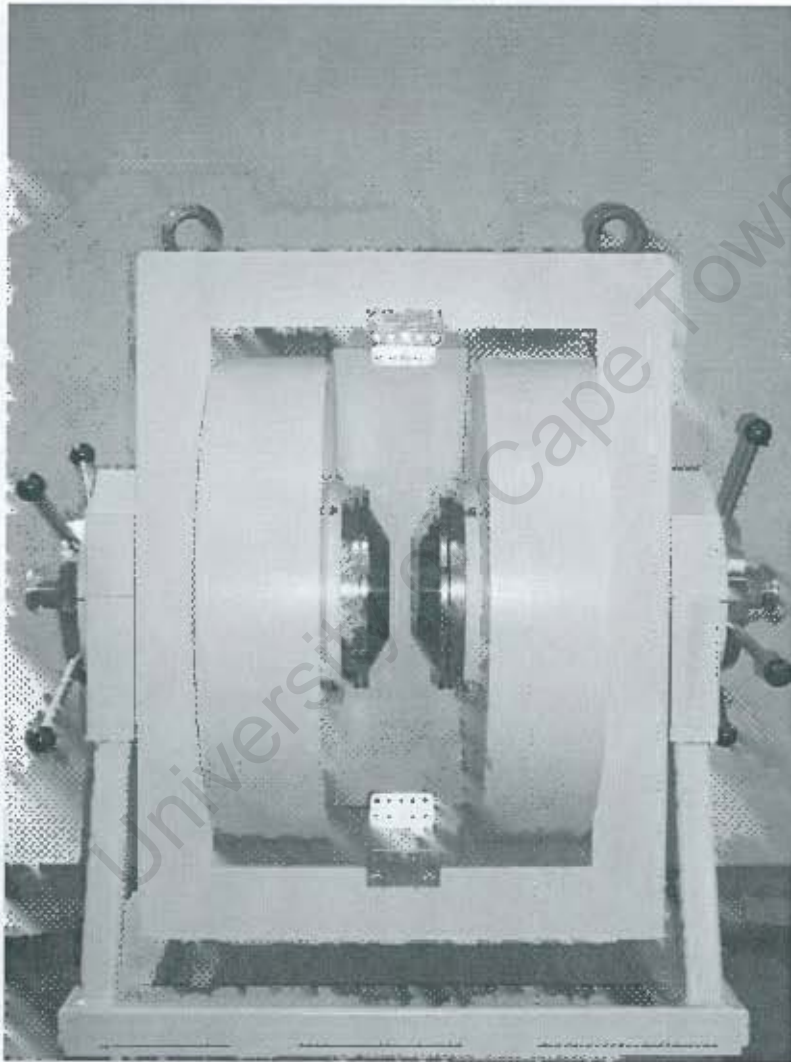


Figure B.1: Bruker B-E25V magnet, front view.

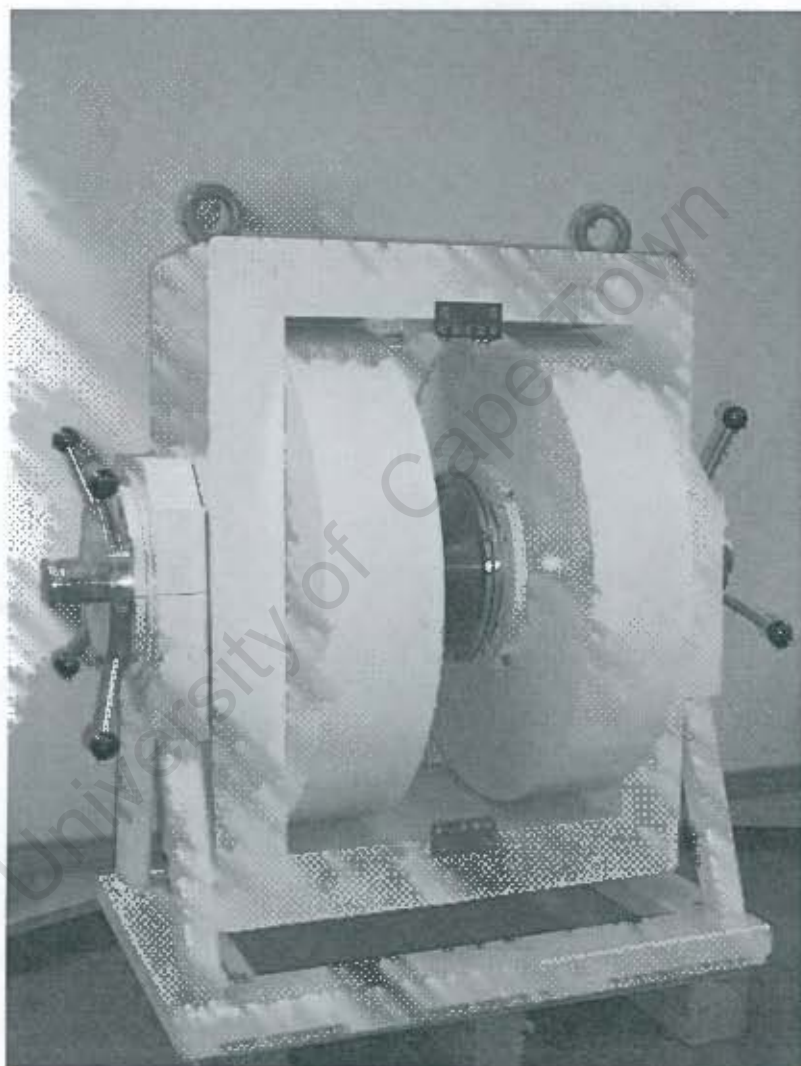


Figure B.2: Bruker B-E25V magnet, oblique view.

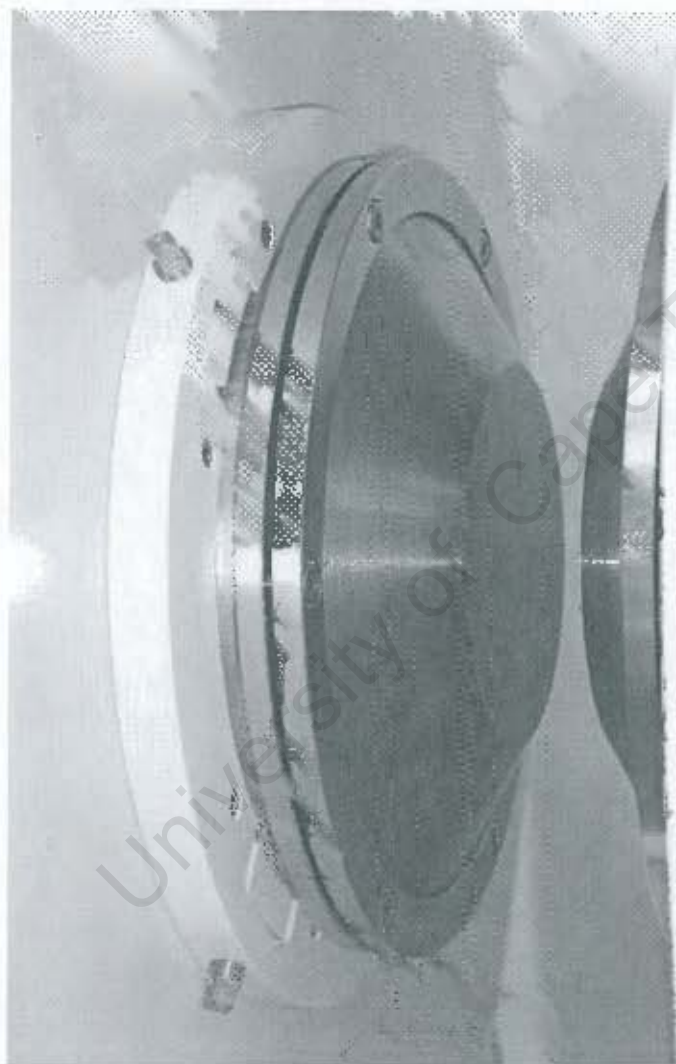


Figure B.3: Bruker B-E25V magnet showing the tapered poles. The pole face diameter is 15 cm.

Appendix C

Rig Photographs

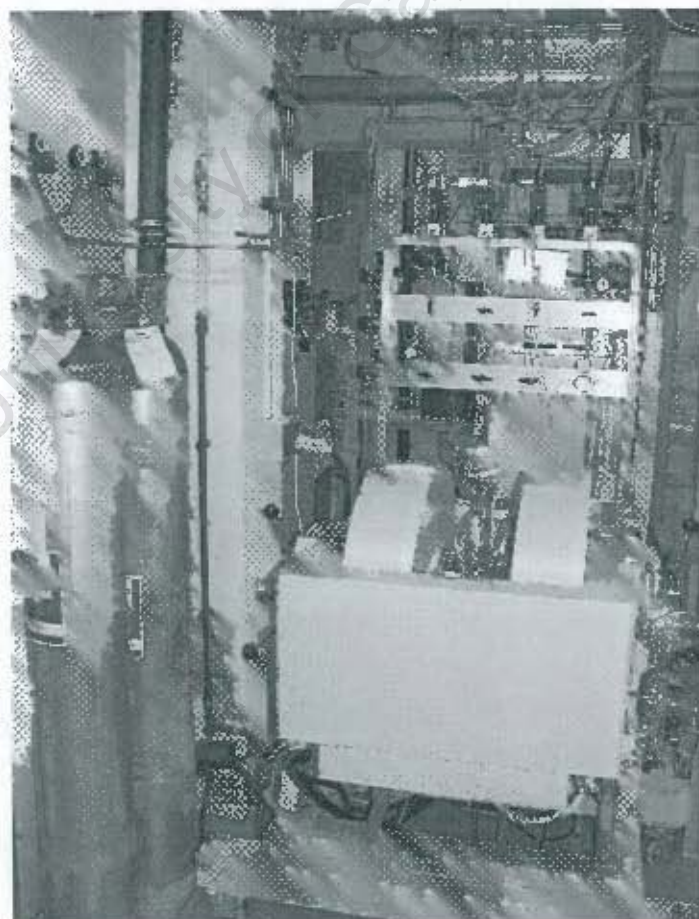


Figure C.1: Front view of the rig.



Figure C.2: Side view of the rig.

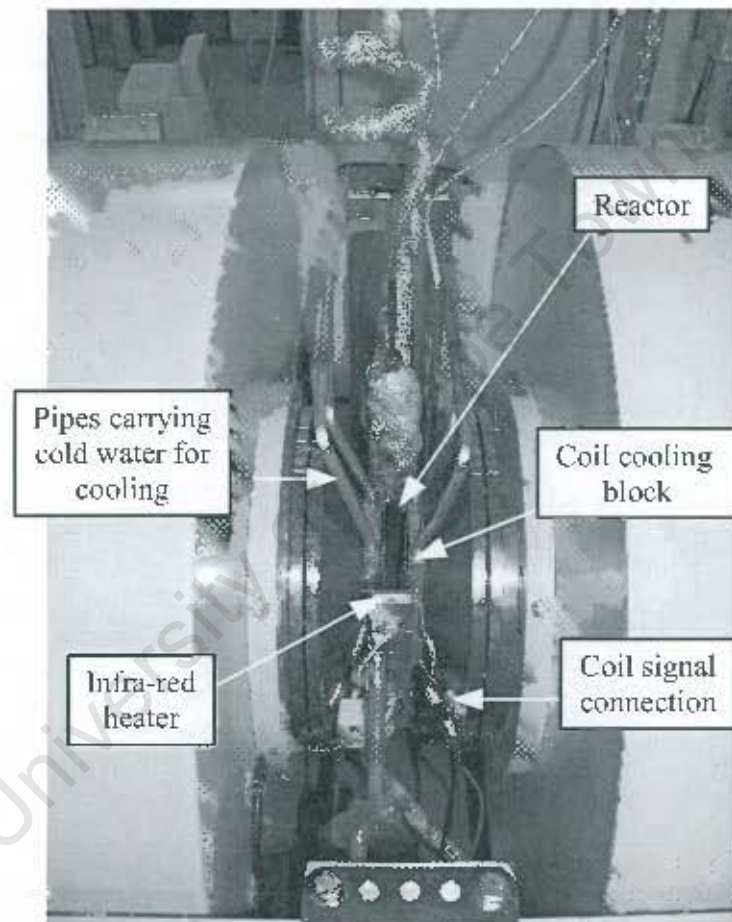


Figure C.3: Front top view of the rig showing the positioning of the reactor in the center of the magnet.

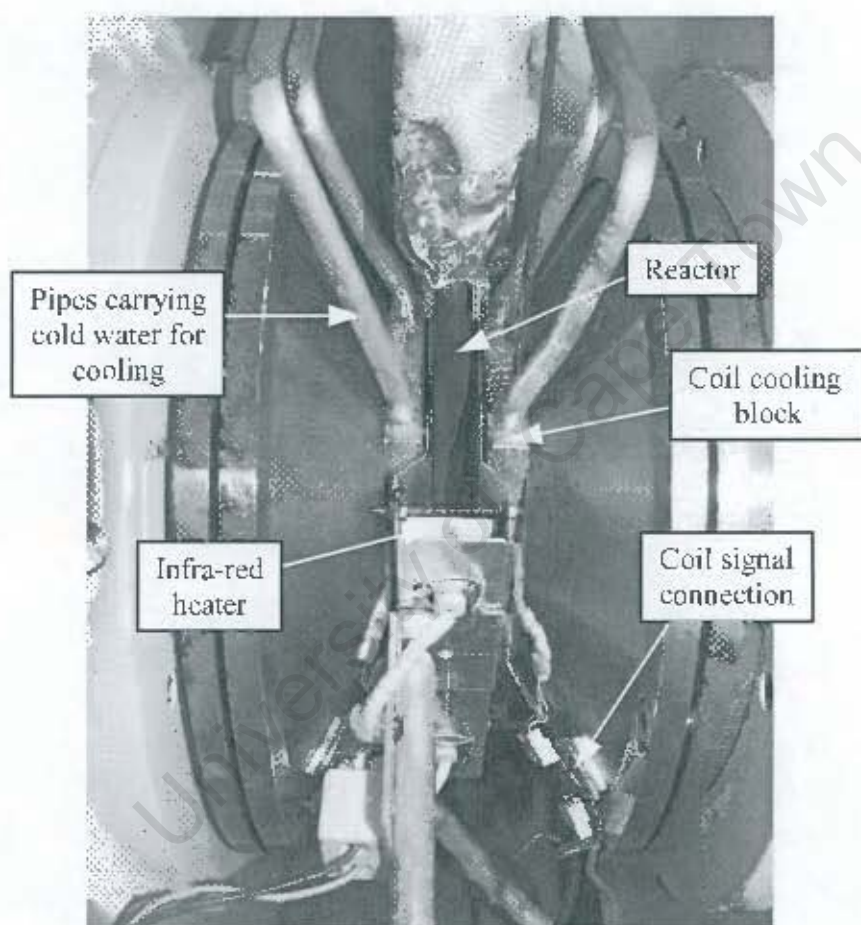


Figure C.4: Close up of the reactor showing the coil cooling, coil connections, reactor and infra-red heating.

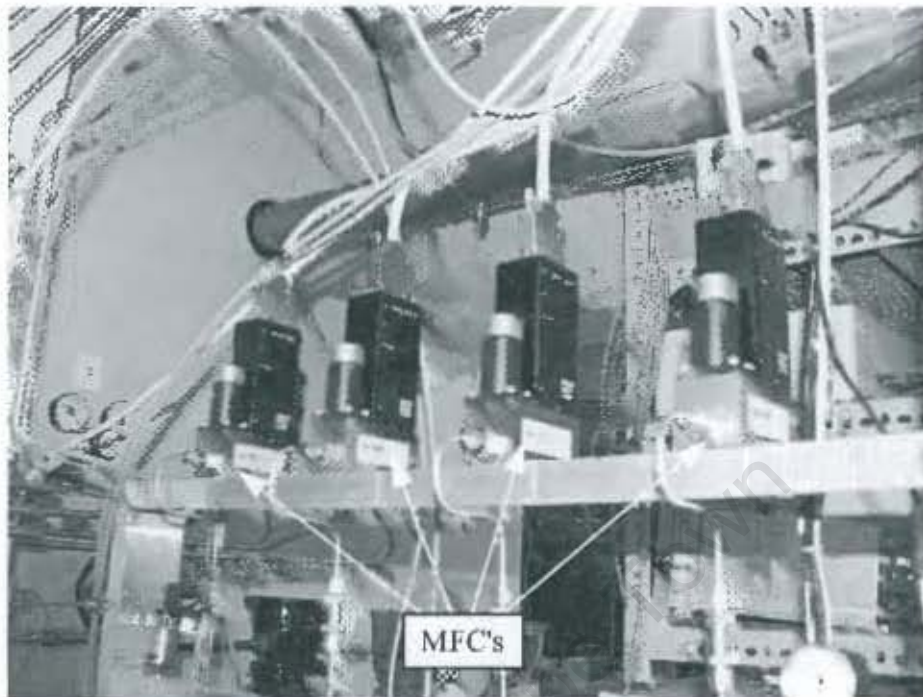


Figure C.5: The mass flow controllers mounted on an aluminum bracket which is attached to the magnet.

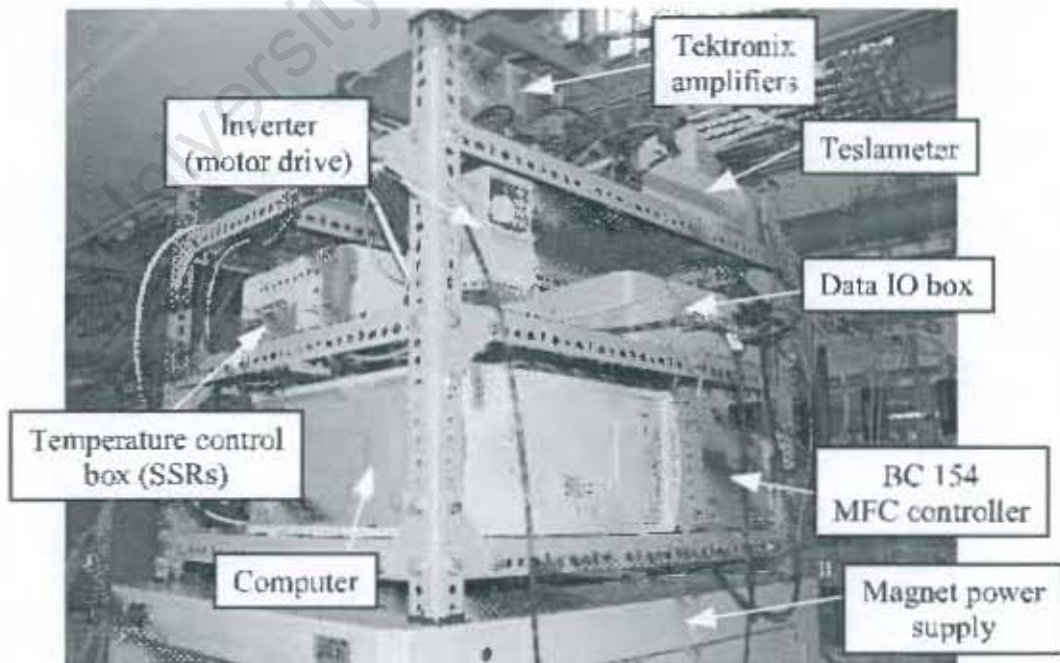


Figure C.6: The electronics for the rig are housed on top of the magnet power supply.

Appendix D

Coil Winder Photographs

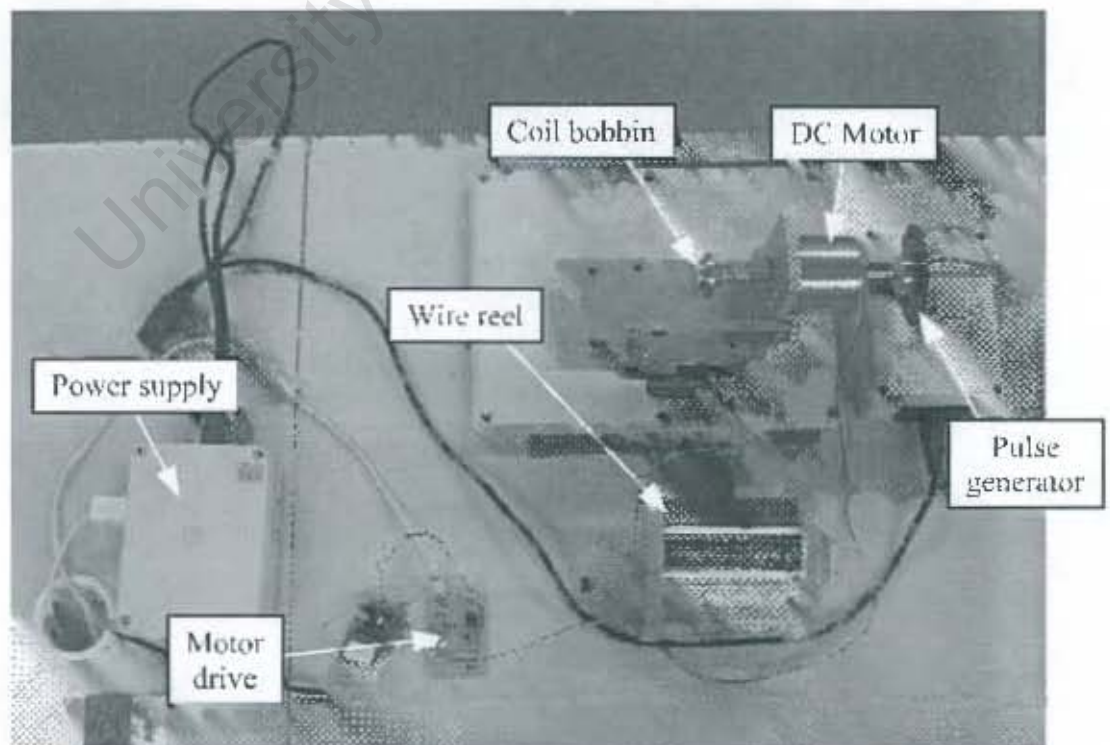


Figure D.1: Top view of coil winder.

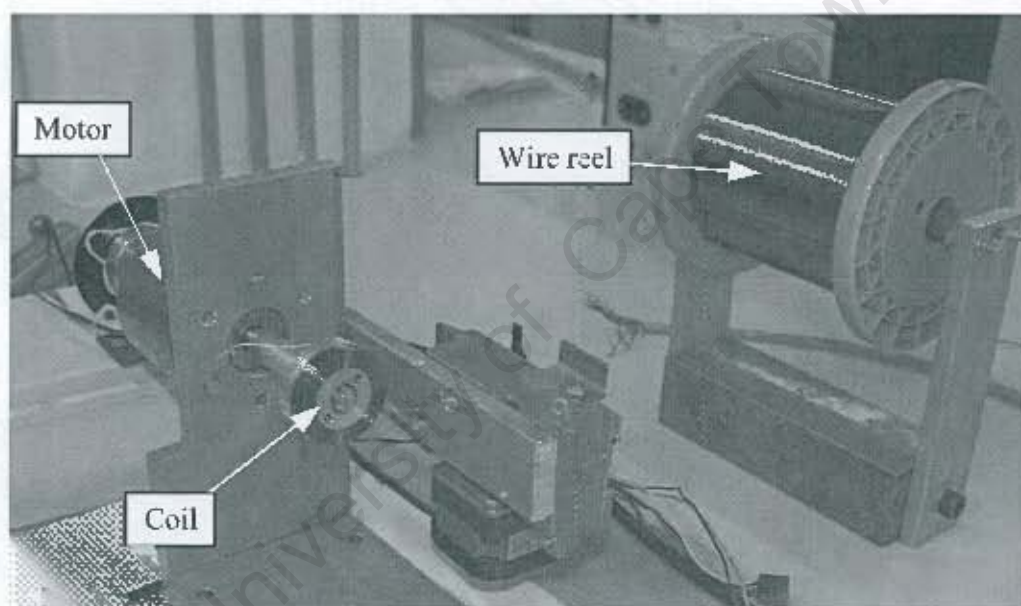


Figure D.2: Side view of coil winder.

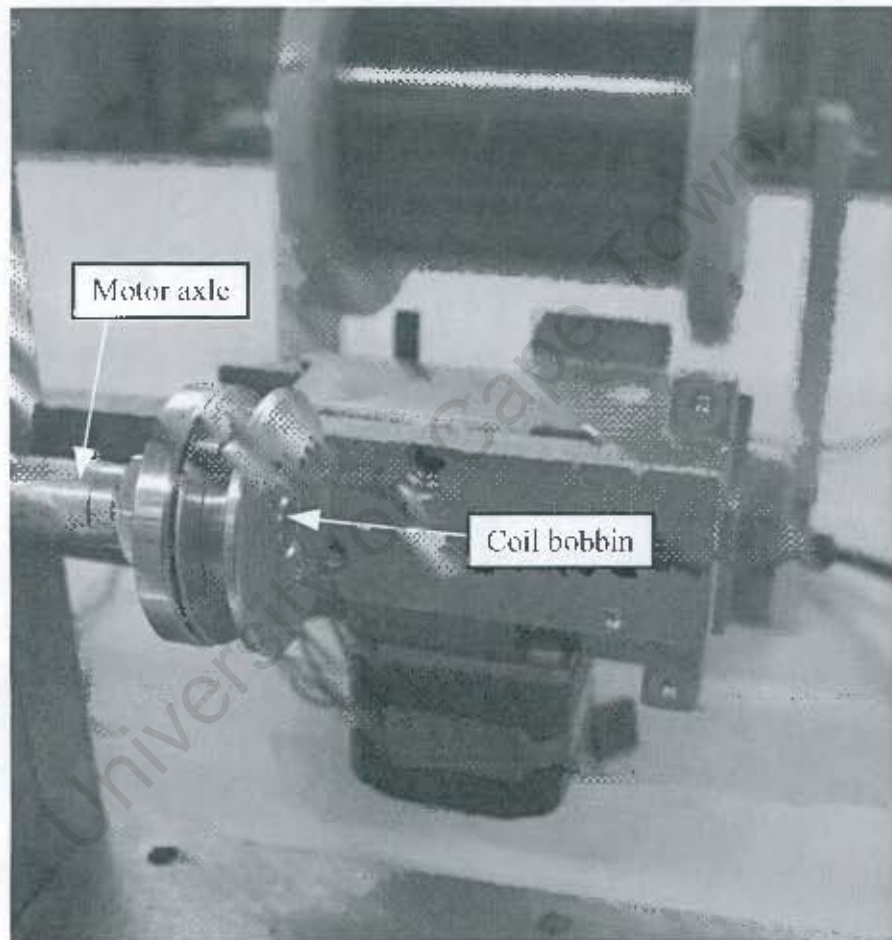


Figure D.3: Front view showing the coil bobbin attached to the motor axle.

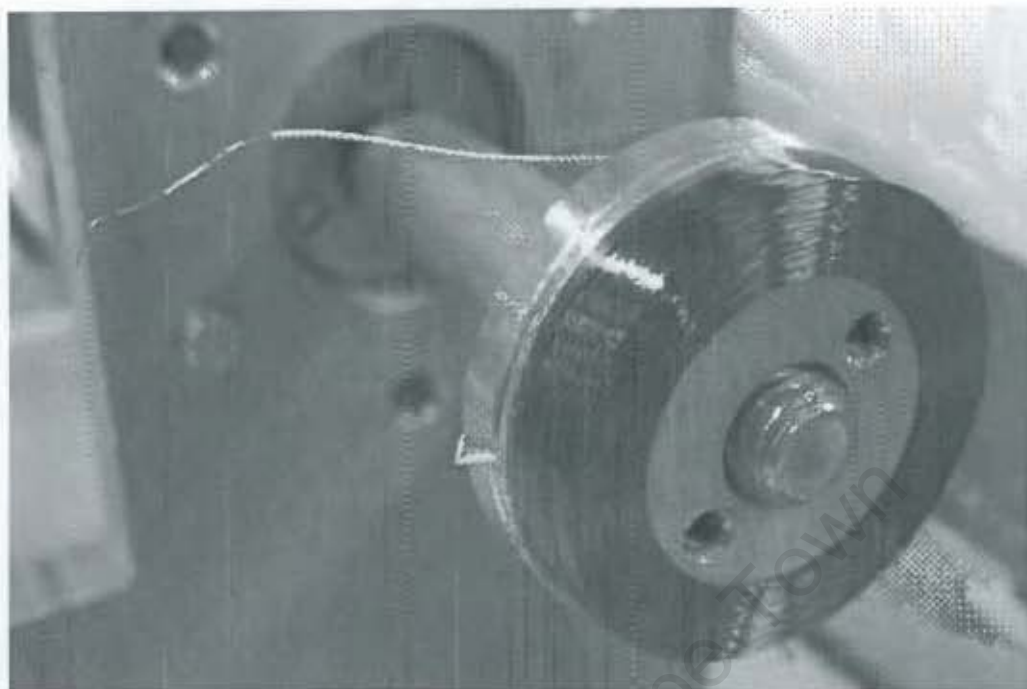


Figure D.4: Wound coil attached to the motor shaft.

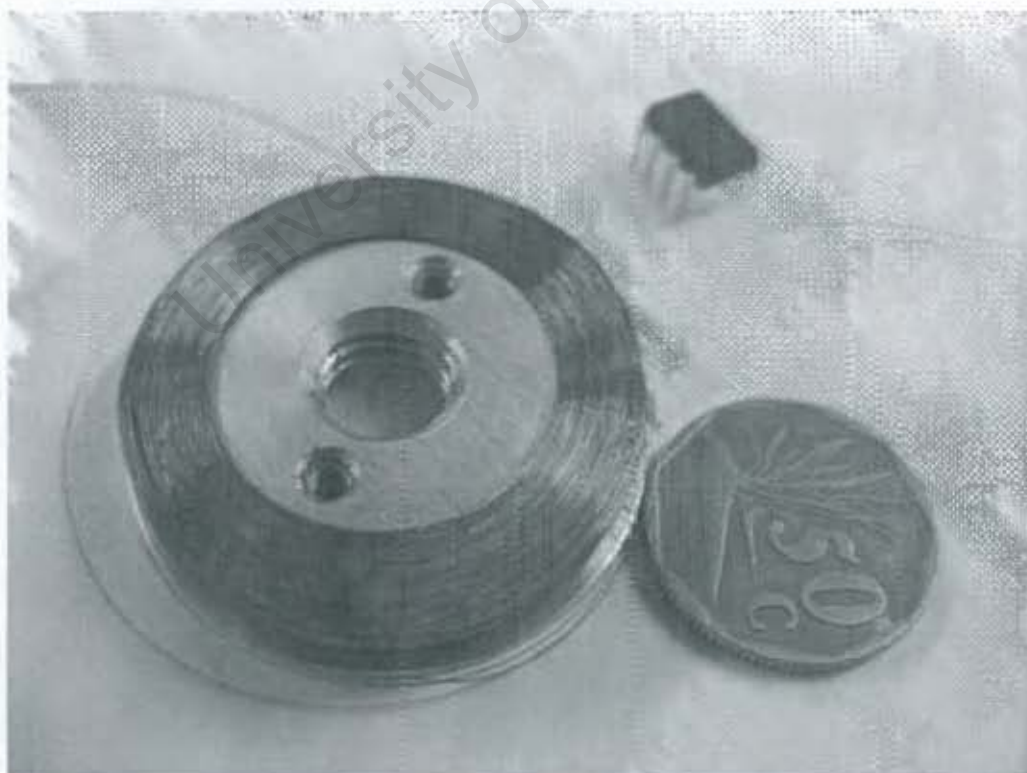


Figure D.5: A wound coil in comparison to a South African 50 cent coin and a eight pin dip chip.

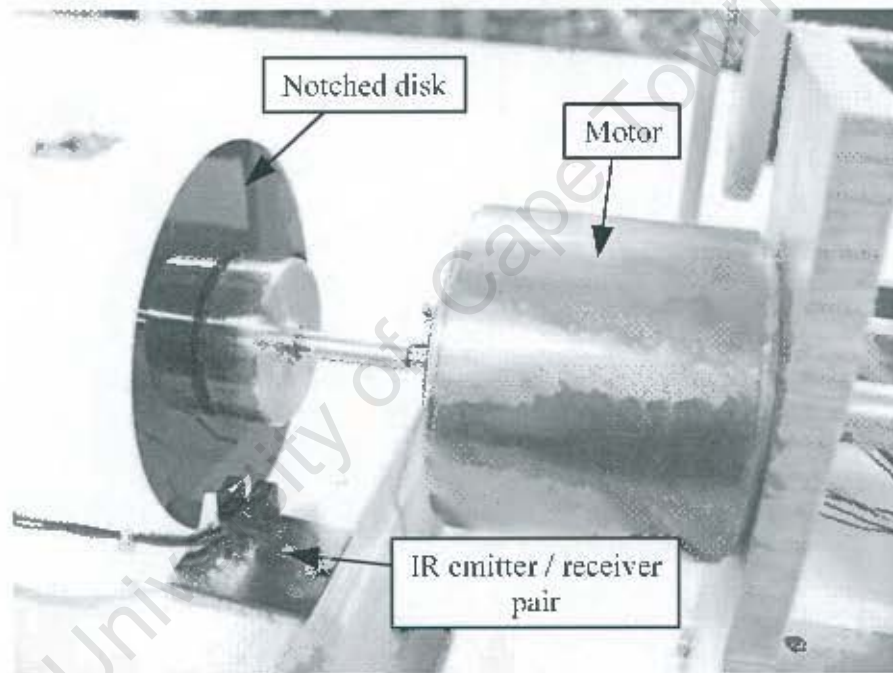


Figure D.6: Motor with notched disk attached to shaft.

Appendix E

Hardware Diagrams

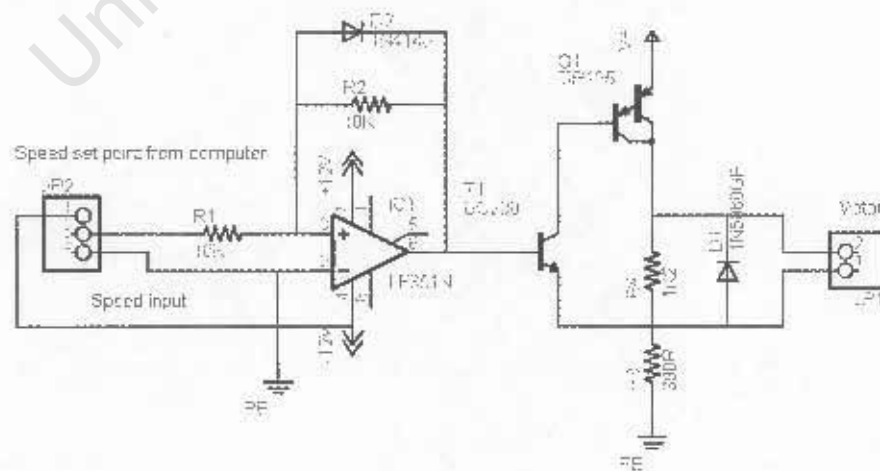
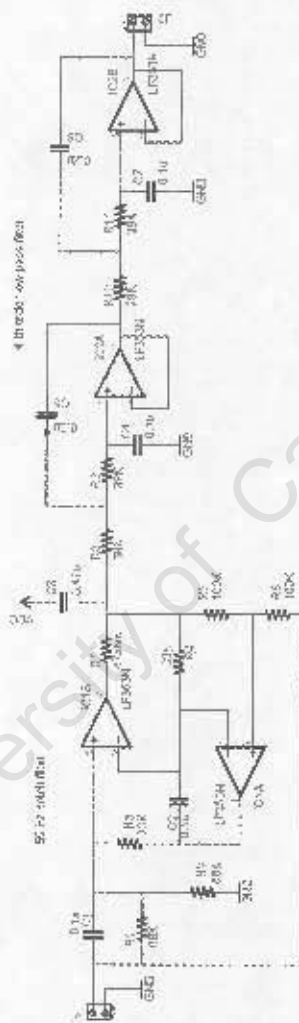


Figure E.1: Schematic of the circuit used to drive the motor from a set point generated by the computer. This circuit is part of the winder built to wind the pick-up coils.



(Sec. 6.4.2 on page 69)

Figure E.2: Filter schematic

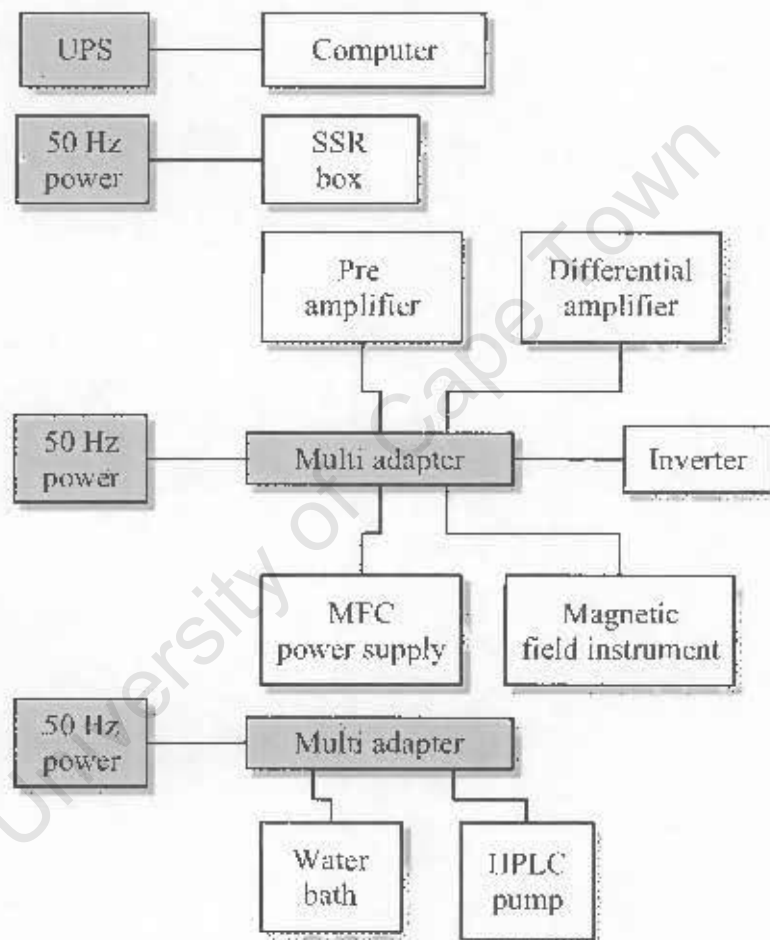


Figure E.3: Power supply connections

Bibliography

- [1] Wikipedia. Internet, October 2005. Available from: <http://en.wikipedia.org/wiki/LabVIEW>.
- [2] AUSSERLECHNER, U., KASPERKOVITZ, P., AND STEINER, W. Pick-up systems for vibrating sample magnetometers—a theoretical discussion based on magnetic multipole expansions. *Meas. Sci. Technol.* 5 (1994), 213–225.
- [3] AUSSERLECHNER, U., STEIRER, W., AND KASPERKOVITZ, P. Calibration of vibrating sample magnetometers independent of reference samples. *Meas. Sci. Technol.* 7 (1996), 1574–1578.
- [4] BOWDEN, G. J. Detection coil systems for vibrating sample magnetometers. *J. Phys. E: Sci. Instrum.* 5 (1972), 1115–1119.
- [5] BRAGG, E. E., AND SEEHRA, M. S. Analysis of induced emf in vibrating-sample magnetometers. *J. Phys. E: Sci. Instrum.* 9 (1976), 216–223.
- [6] BRAILSFORD, F. *Magnetic materials*. London : Methuen, 1960.
- [7] CAREY, R., NEWMAN, D. M., SNELLING, J., AND THOMAS, B. W. J. An optical heating stage for a vibrating sample magnetometer. *Meas. Sci. Technol.* 3 (1992), 424–425.
- [8] CRAIK, D. *Magnetism : principles and applications*. Chichester : Wiley, 1995.
- [9] DENNISON, E. Magnetic formulas. Internet, 2005. Available from: <http://www.netdenizen.com/emagnet/>.
- [10] DODRILL, B. C., LINDEMUTH, J. R., AND KRAUSE, J. K. Magnetic anisotropy: Measurements with a vector vibrating sample magnetometer. *Lake Shore Cryotronics, Inc.*

- [11] DRAKE, A. E., AND HARTLAND, A. A vibrating coil magnetometer for the determination of the magnetization coercive force of soft magnetic materials. *J. Phys. E: Sci. Instrum.* 6 (1973), 901-902.
- [12] FLANDERS, P. J., AND DOYLE, W. D. Motor drive magnetometer for thin magnetic films. *Review of Sci. Instr.* 33 (1962), 691-693.
- [13] FONER, S. Vibrating sample magnetometer. *Review of Sci. Instr.* 27 (1956), 548.
- [14] FONER, S. Versatile and sensitive vibrating-sample magnetometer. *Review of Sci. Instr.* 30, 7 (1959), 548-557.
- [15] FONER, S. The vibrating sample magnetometer: Experiences of a volunteer (invited). *J. Appl. Phys.* 79 (1996), 4140-5.
- [16] GIORDANO, J. L., AND ESPARZA, D. A. A parallel-motion vibrating-sample harmonic magnetometer. *Meas. Sci. Technol.* 5 (1994), 509-513.
- [17] GUY, C. N. Frequency doubling - a new approach to vibrating sample magnetometers. *J. Phys. E: Sci. Instrum.* 9 (1976), 433-435.
- [18] GUY, C. N. A simple approach to coil design for vibrating sample magnetometers. *J. Phys. E: Sci. Instrum.* 9 (1976), 790-1.
- [19] HOON, S. R. An inexpensive, sensitive vibrating sample magnetometer. *Eur. J. Phys* 4 (1983), 61-67.
- [20] HOON, S. R., AND WILLCOCK, S. N. M. The design and operation of an automated double-crank vibrating sample magnetometer. *J. Phys. E: Sci. Instrum.* 21 (1988), 772-785.
- [21] JOHANSSON, T., AND NIELSEN, K. G. A low frequency vibrating sample magnetometer. *J. Phys. E: Sci. Instrum.* 9 (1976), 852-854.
- [22] KRYLOV, I. Design of detector coils to measure the magnetization of super paramagnetic and ferromagnetic samples. *unpublished* (September 2004). University of the Western Cape.
- [23] LAVA FORUM. Labview advanced virtual architects forum. Internet, 2005. Available from: <http://forums.lavausergroup.org/>.

- [24] MAGNUM, B. W., AND THORNTON, D. D. Vibrating sample magnetometer for use at very low temperatures and in high magnetic fields. *Review of Sci. Instr.* 41 (1970), 1764-6.
- [25] MALLINSON, J. Magnetometer coils and reciprocity. *J. Appl. Phys.* 37 (1966), 2514-5.
- [26] NATIONAL INSTRUMENTS. National instruments online users community. Internet. 2005. Available from: <http://zone.ni.com/devzone/labviewzone.nsf/OpenPage?openagent&lvsection=labviewzone>.
- [27] NIAZI, A., PODDAR, P., AND RASTOGI, A. K. A precision, low-cost vibrating sample magnetometer. *Current Science* 79 (2000), 99-109.
- [28] NOAKES, J. E., ARROTT, A., AND HAAKANA, C. Vibrating sample magnetometers. *Review of Sci. Instr.* 39 (1968), 1436-8.
- [29] PACYNA, A. W. General theory of the signal induced in a vibrating magnetometer. *J. Phys. E: Sci. Instrum.* 15 (1982), 633-669.
- [30] PACYNA, A. W., AND RUEBONBAUER, K. General theory of a vibrating magnetometer with extended coils. *J. Phys. E: Sci. Instrum.* 17 (1984), 141-154.
- [31] REDFIELD, A. G., AND MOLESKI, C. Vibrating sample magnetometer for protein research. *Review of Sci. Instr.* 43 (1972), 760-2.
- [32] SOMMERVILLE, I. *Software Engineering*. Addison-Wesley, 1995.
- [33] SPRINGFORD, M., STOCKTON, J. R., AND WAMPLER, W. R. A vibrating sample magnetometer for use with superconducting magnet. *J. Phys. E: Sci. Instrum.* 4 (1971), 1036-40.
- [34] TEXAS INSTRUMENTS, AND MANCINI, R. Op amps for everyone, 08 2002.
- [35] WILLIAMS, C. D. H. Feedback and temperature control. Internet. 2005. Available from: <http://newton.ex.ac.uk/teaching/CDI/i/Feedback/>.

UT 621.3 KELL

BUT 621.3 Kell

low frequency, in situ vibrating simple magnetometer.

Question 5 (7)

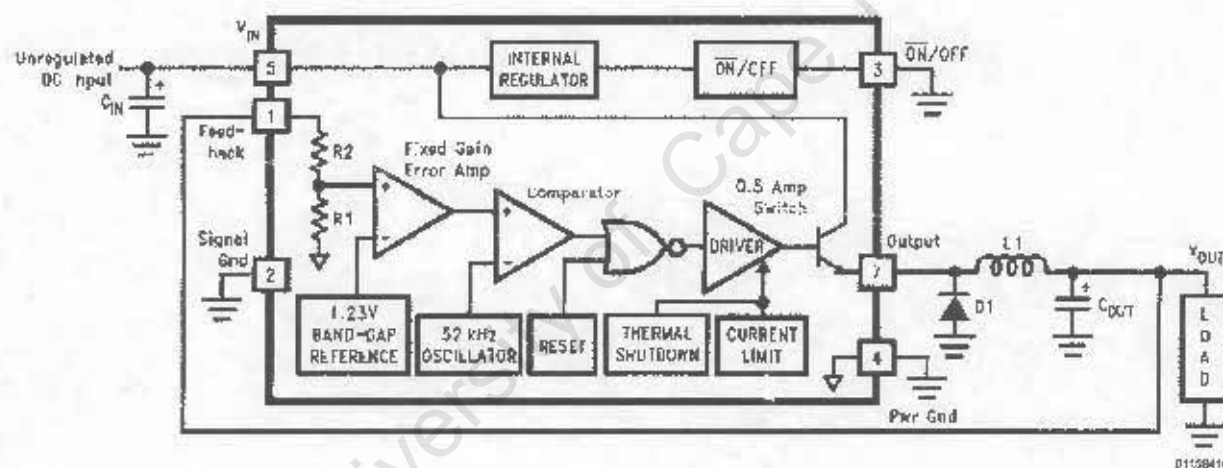
a) Draw the circuit diagram for a phase shift oscillator with two outputs. The outputs are 1kHz sine waves and must be at 90 degrees to each other. Show resistor values for the phase shift network as well as for the feedback network. Use capacitors of 1 μ F, 100nF, 10nF and 1nF for the phase shift network.

(4)

b) Draw the circuit diagram for a 10MHz oscillator which has an output frequency with a tolerance of less than 0.1%. Output voltage is not critical. (3)

Question 6 (10)

The LM2574 switched mode regulator has the following suggested circuit diagram, taken from the datasheet. The circuitry enclosed in the bold rectangle is internal to the microchip, components outside the bold rectangle are added to form a complete regulator.



a) Is this power supply galvanically isolated or not? Justify your answer. (2)

b) Modify the above circuit diagram (only the external components) showing how to double the output voltage (2)

c) Give three important characteristics of the diode D1 (3)

d) What is the function of the components L1 and COUT? (1)

e) What advantage would be gained by increasing the switching frequency of the regulator? (1)

f) Name one change that you would make to the internal circuitry of this regulator to improve its efficiency (1)

N S

① 1 ②

2 S

①

↑

University of Cape Town