MinEx CRC Limited
26 Dick Perry Avenue, Kensington, WA, 6151
PO Box 1130, Bentley, WA, 6102, Australia
admin@minexcrc.com.au

MinEx CRC
ABN: 66 625 533 913

MinEx CRC provides financial support to the value of $1K to promote Honours and Masters by Coursework projects that are aligned with the mission of MinEx CRC and to encourage young researchers toward a career in mineral exploration research. Projects are not restricted to MinEx CRC Participants and Affiliates.

Please note that the content of this thesis has not been subjected to peer-review and subsequent corrections.

Australian Government
Department of Industry, Science and Resources
Cooperative Research Centres Program

School of Electrical Engineering, Computing and Mathematical Sciences

Fluid circulation measurement for automated drilling

Digby F. Emmett

This thesis is presented for the Degree of
Bachelor of Electrical Engineering & Computer Science
of
Curtin University of Technology

January, 2022

Abstract

Initially the topic is introduced to the reader. A numebr of different technologies are discussed and evaluated in order to determine the usefullness of each technology. Further the technologies are discussed in terms of communciation with a main system.

The project idea is presented and the implementation specification and methodology is discussed while also specifying the exact functionality including the operation of each piece of code in great detail. Following this testing and results are discussed with emphasis on the reponse the system produced.

This is then used to evaluate the sucess of the paper and suggest alternative approaches to acheiving the same goal as well as possible improvements to the research process.

Digby Foley Emmett
7 Treacy Way Waterford

October 24, 2021

A/Prof Abu-Siada

Discipline Leader in Electrical and Electronic Engineering

Curtin University

Kent Street, Bentley

GPO Box U1987 Perth WA 6485

Dear A/Prof Abu-Siada,

I, Digby Emmett, hereby submit my thesis entitled "" as part of my requirements for completion of the bachelor of engineering in Computer Systems Engineering.

I declare this thesis is entirely my own work with the exception of the acknowledgements and references mentioned.

Yours Sincerly,

Digby Emmett

October 24, 2021

Declaration of published work

I acknowledge that parts of the progress report presented at the end of the first semester of this project have been used in the following chapters of this thesis and the report has been properly referenced:

- Chapter One - Introduction
- Chapter Two - Literature Review
- Chapter Three
  - Section One
  - Section Two
  - Parts of Section Three

Acknowledgements

I would like to thank my supervisor Siavash Khaksar for his support throughout the time I have spent writing this thesis.

Contents

List of Figures

Chapter 1

Introduction

Currently there is little research and analysis on the area of signal transmission through fluid mediums. This thesis aims to outline the development of a proposed method for the collection and analysis of fluid data through the use of a number of tools which will aid in the future research of this area.

This project is based around the development of a control system which will ideally enable the efficient collection and aid the analysis of fluid data run in a closed loop application. The primary objective of this project is to develop a system which is capable of varying the flow rate of a fluid through a pipe system while monitoring and recording both the input and output of the system.

It is theorised that through the utilisation of modern programming and concurrent hardware that the accurate and useful collection of fluid data this can be achieved. The hypothesis stands to demonstrate that this system which is being developed must be developed in such a way that it can be used to record and understand the effects of varying fluid pressure and the resulting effects on viscosity.

The basis of the project is to aid in research which is trying to determine if drill bore damage can be prevented via early warning signs from monitoring drilling fluids pressure, viscosity and flow rate variations. This is made possible through the use of Lab View and National Instruments hardware which provides a platform and environment which has been used through out the development of the project. The system has been developed using the Lab View environment with the primary objective being that the software remain not only usable but maintainable upon the completion of this research project.

This thesis is presented in seven chapters. The introductory chapter provides a clear overview of the thesis and the aims which are trying to be achieved through the research which is being performed. The literature review provides background information and a summary of the important components of the research project. The literature review also provides a clear picture of the project as a whole and what measurements are expected to result from the research which is being performed. Chapter 3 contains the specifications and methodology for the project and details how the project was specified and how the project was restricted in order to achieve the stated goals. Chapter 4 contains the development steps involved in the project and details how the progress was made in the project from the initial idea to the final design and solution. Chapter 5 presents the results for the project and attempts to present details as to the accuracy of those results. Finally chapter 6 offers a discussion of the content found in chapter 5 and aims to identify the success of the project as it relates to this chapter and the stated objective.

Chapter 2

State of the art

2.1 Fluid Mechanics Overview

There are two primary categories of fluids in the study of fluid dynamics. The study of Newtonian fluids is concerned with fluids maintain a consistent viscosity independent of the rate at which they are flowing. Non-Newtonian fluids viscosity varies according to the rate the fluid is being disturbed.

Viscosity is the fundamental property which differentiates the two types of fluids as described below.

2.1.1 Viscosity

Viscosity is a measure of the rate which a fluid flows relative to a boundary. It is generally modelled by the zero slip condition which assumes that the fluid in direct contact with a motionless surface will have the same velocity as that surface. Fluids such as water exhibit constant viscosity and thus may be used as a control variable for comparison. Viscosity is generally tested directly using the an experimental test bench as in Figure2.1.

There are two categories of Non-Newtonian fluids, sheer thickening and thinning. Sheer thickening fluids exhibit exponential stress increase as sheet velocity increases. Sheer thinning fluids exhibit sheer stress reduction as their sheer velocity tends to infinity.

Figure2.2 shows an example of the viscosity graph of both of these. This graph also

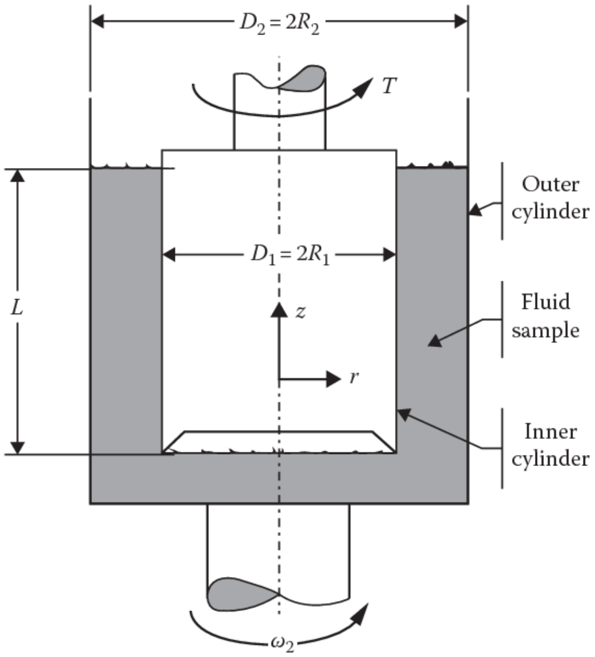

Figure 2.1: Kostic (2013) Fluid Viscosity Measurement

has a reference viscosity fluid which can be assumed to be water. This fluid exhibits no change in sheer stress when sheer velocity increases.

2.2 Key Fluid Measurements

Key fluid properties are discussed below to identify critical measurements. Discussion is necessary to ensure the sensors chosen have quantifiable means of comparison. Typically a rheological fluid would be used for experiments such as those proposed above however rheological sensors require calibration and parameter tuning [Heinisch, Voglhuber-Brunnmaier, Reichel, Dufour, and Jakoby (2014)]. As a result viscosity is not of primary importance at this stage and sensors relating to this specifically will not be discussed.

2.2.1 Pressure

One critical fluid property which must be established in order to conduct this experiment is pressure. This will be used to establish the effectiveness of variations in fluid pressure at a distance. For this it may be helpful for understanding to imagine repeatedly blowing into the end of a hose with fluids stored inside. There would be no

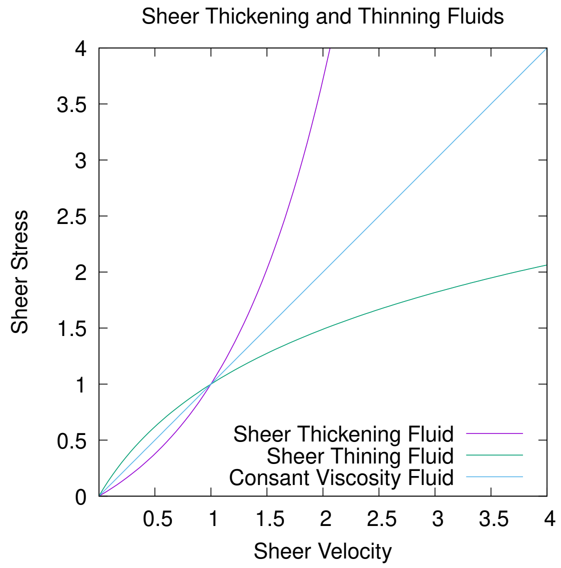

Figure 2.2: Comparison of sheer thickening and thinning fluids

apparent effect on the other end of the hose (assuming there is not enough fluid to make its way out of the hose) and instead there would be variations within the hose. This is why pressure must be captured to establish the fluids properties.

2.2.2 Flow Rate

Another aspect is the fluids flow rate. Following the analogy presented above. Consider there is enough fluid to flow out of the end of the hose. It is useful to understand the rate the fluid is exiting the hose to better understand the effective flow rate throughout the pipe. This example is more difficult to understand as a result of the non-Newtonian properties of the fluids which are of interest.

2.3 Fluid Models

Viscosity is another crucial factor which should be calculated using the equipment which is discussed below. Non constant viscosity fluids require models to predict their behaviour. For this a number of different models have been proposed to approximate non constant viscosity fluids. These models include:

- The Bingham model
- Power law

- Yield power law

The equations are less important for this section and thus have been excluded. A brief description of the purpose they serve is provided.

Bingham Model

The Bingham model describes fluids which exhibit sheet thinning, in this case these fluids are termed as visco-plastic due to their plastic deformation and gradual easing of stresses. Rehm et al. (2012). This model is simple because it predicts the sheet stress (τ) dependant on sheer velocity ($\dot{\gamma}$)Beverly and Tanner (1992).

Power Law

The power model is used to determine fluid viscosities more accurately than the Bingham model.

Yeild Power Law

Yield power law is incredibly accurate in modelling the relationship between sheer stress and sheer velocity in sheer thinning fluidsHemphill, Campos, and Pilehvari (1993).

2.4 Sensor Types

Javed, Mansoor, and Shah (2019) reviewed applications of MEMS pressure sensors for aerospace applications. It was established in this study that the sensors should have high accuracy, high reliability, an be capable of withstanding a harsh environment. This is true for the focus of this paper too. Javed et al. (2019) also compared sensors from a number of different articles on various characteristics including size, sensitivity, measurement range and mechanism.

2.4.1 Capacitive Pressure

Capacitive pressure sensors utilise two plates between which capacitance reduces with increased pressure due to the gap between the plates. Figure2.3 illustrating the function of a capacitive pressure sensor.

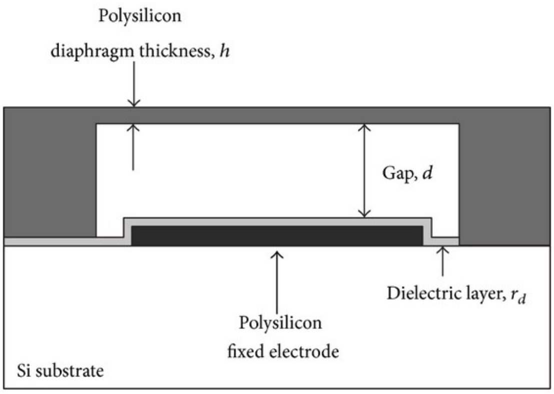

Figure 2.3: Rendon-Nava et al. (2014)Capacitve Pressure Sensor

A variations in pressure is represented by a variation in capacitance. This is transmitted through a variation in the voltage which is produced as an output through. This makes it ideal for the applications in fluid pressure. Capacitive pressure sensors also are well suited to high sensitivity and high pressure environments, they vary little with fluctuations in heat. Balavalad and Sheeparamatti (Apr 2015)

2.4.2 Strain Gauge Pressure

Strain Gage pressure sensors operate through the utilisation of a metal (often steel) bar in which the elastic deformation is measured by running a current through the stretched material which is varied according to the amount of additional resistance which is induced by the strain applied.

Strain gauges can handle extremely high pressures due to the relationship between the resistance and the length of the wire. It is also unlikely that strain gauge pressure measurements be broken as a result of their high durability.

2.4.3 Piezo Resistive Sensors

Piezo-resistive sensors vary their resistance according to the pressure applied, this is similar to strain gauge sensors however they are often much more compact.

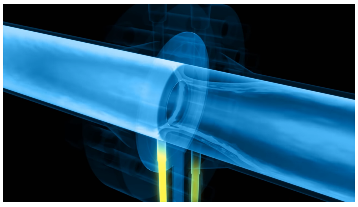

Figure 2.4: Endress+Hauser (2009a)Venturi Effect

2.5 Flow meters

Flow meters allow for the measurement of fluid volume passing through an area in a moment of time. There are a number of different types available thus it is crucial to identify characteristics for the comparison of these sensors.

Van Zeller outlined a set of requirements in trying to identify the accuracy of flow meters in medical applications. The attributes flow meters were compared against were accuracy, responsiveness and speed for fluid studies.VanZeller, Williams, and Pollock (2019)

2.5.1 Venturi Flow Meters

The Venturi flow meter makes use of the Venturi effect. As flow rate increases the pressure directly after a restriction in the pipe is dramatically decreased while the static pressure before the restriction is increased. The result is a pressure difference which can be read. The main disadvantage of the venturi flow meter is the fact that the pipe must be restricted at some point to allow for the effect to occur. This is not ideal as fluids may be made up of rocks and other substances.

2.5.2 Thermal Flow Meters

Thermal flow meters operate on the principle of heat loss over a distance. In this case the flow rate is measured by calculating the change in heat over distance which it has flowed as compared to what is expected and observed as some reference speed..

If the heat at one sensor is equal to the atmospheric temperature while the other is equal to the heating element, the flow rate is assumed to zero. Alternately if the temperature is similar at both temperature sensors two points the flow rate can be assumed to be high.

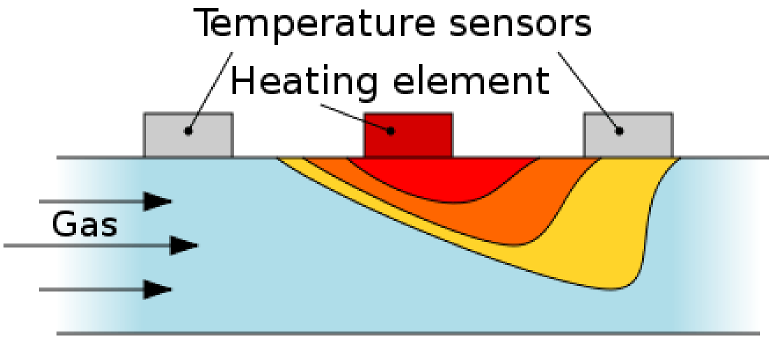

Figure 2.5: Biezl (2008) Thermal Flow Meter Principle

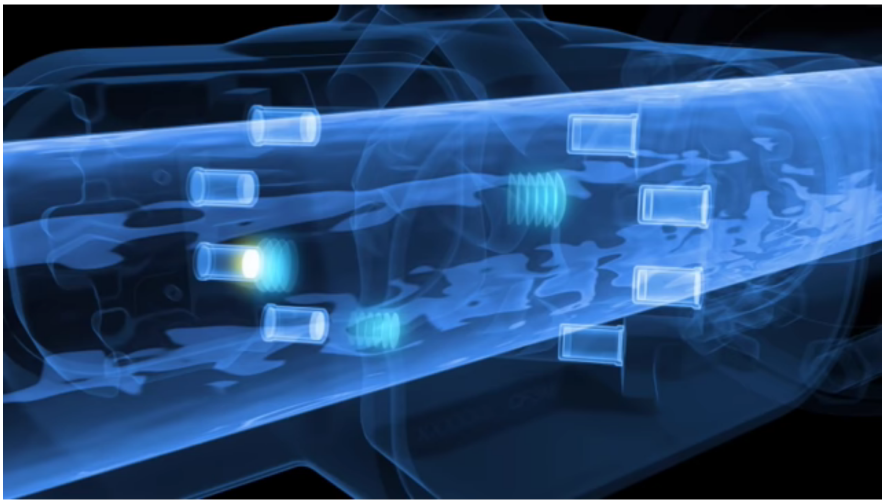

Figure 2.6: Endress+Hauser (2009b)Transit time Flow Meter Principle

Thermal flow meters are not ideal as they require knowledge of the fluid which is being measured. In this case the fluid which is being measured will have a thermal coefficient which must be observed when performing the calculations.

The operator does not know what substance is being drilled until it is pulled to the surface thus measurements could not be performed while the drill is transitioning between substrates. This introduces issues in the system and renders the flow meter less useful.

2.5.3 Ultra-Sonic Flow Meters

Ultra sonic flow meters operate by transmitting sound through the fluid medium. This is then received else ware by a high accuracy pressure sensor.

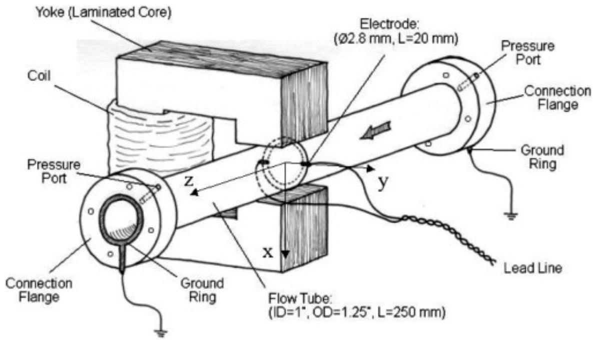

Figure 2.7: Cha et al. (2002) Electro Magnetic Flow Meter

- Doppler Effect)

  The transmission of a constant frequency through the fluid is received at the other sensor. This relies on impurities in the fluids to reflect the sounds back to the other sensor, that is, it can be used in opaque fluids Takeda (1995). The frequency shift is measured and used to calculate the rate of flow experienced by the fluid. Unfortunately this method is not ideal for fluids with small debris.

- Transit Time)

  Transit time ultrasonic sensors are easier to implement than Doppler effect. For transit time sensors a frequency is transmitted in the form of sound through the fluid directed at a receiver on the other side of the chamber where speed can be calculated. A diagram of this is shown in Figure2.6.

2.5.4 Electro Magnetic Flow meters

Electro-magnetic flow meters offer truly non invasive measurements of fluids where other types of flow meters rely on imperfections or reliance on specific properties of the fluids. This is ideal for the experimentation of this type as it ensures that damage is minimised in the flowing of fluids through the system. Electro-magnetic flow meters are fundamentally based around Faraday's laws of electro-magnetic induction.

Electro-magnetic flow meters have a very high accuracy. the diagram below shows the working principle of electro magnetic flow meters Cha et al. (2002). the basic principle of this flow meter is shown in Figure2.7

2.6 Sensor Review

2.6.1 Pressure Sensors

As outlined above pressure sensors will be evaluates on 3 criteria in order to ensure they are suitable for the task which is proposed.

- Response time
- Sensitivity range
- Temperature range
- Accuracy

Four different sensors will be summarised in order understand the limiting factors involved with the sensors. An ideal sensor will be presented with its advantages and disadvantages as well.

MRQ22 - MeasureX

The MRQ22 pressure sensor is a type of strain gauge sensor. It is one of the best pressure sensors available on the market. This sensor offers superior range in measurements as compared to other sensors. Its data shows it performs very well for a wide range of pressures and temperatures "MRQ22 Pressure Temperature Transmitter" (2011). This sensor also offers the ability to read temperature which is highly advantageous.

- $< 1ms$ Response Time
- Support for $4..20ma$
- range of $-1..1000bar$

PT5415 - IFM

IFM provide a range of different sensing devices whose functionality varies for a number of applications. The PT5415 provides less functionality. Its specifications are listed below "PT5415 Pressure Transmitter" (2015). Its main advantage is its provides lower pressure readings within a limited scale.

- $1ms$ Response Time
- Support for $4..20mA$
- range of $0..6bar$

PT5443 - IFM

PT5443 is a more advanced version of the PT5415 it provides an improved range of measurements for higher pressure environments. Aside from the improved sensitivity this sensor is practically identical to the PT5415 sensor "PT5443 Pressure Transmitter" (2011)

- $1ms$ Response Time
- Support for $4..20mA$
- range of $0..40bar$

Honeywell FP5000

The FP5000 is a series of strain gauge sensors produced by Honeywell. In this range Honeywell offer mid to high accuracy pressure sensors which are suitable for use in this research "FP5000 Series Pressure Temperature Transmitter" (2011).

- $2.85ms$ Response time
- Operating Temperature: $-40°..85°$
- Measurement range: $0..335bar$

Table 2.1: Comparison of pressure sensors

Property	**Response**	**Range**	**Temperature**	**Accuracy**
MRQ22	$< 1ms$	-1..1000 *bar*	-40..125°*C*	1%
PT5415	$1ms$	0..6 *bar*	-40..90 °*C*	0.1%
PT5443	$1ms$	0..40 *bar*	-40..90 °*C*	0.2%
FP5000	$2.85ms$	0..335 *bar*	-40..90 °*C*	2.25%

This data suggests the MRQ22 would be ideal for the application of this research. Its response time is minimal and it can sense the largest range of values as compared to the other sensors reviewed.

2.6.2 Flow Meters

SM2000

The SM2000 is a magnetic inductive flow meter. This is ideal as it is fully non-invasive and will not disturb the measurement of pressure on the fluids. It is capable of a large range of measurements which makes it ideal for variations in the speed of the fluid flow. Unfortunately the viscosity of the fluids it measures is limited $< 70cS$ "SM2000 Magnetic Inductive Flow Meter" (2021).

- $16bar$ Maximum pressure.
- $-20° < t < 80°$ Temperature range.
- Response time $0.35s$
- Measurement Range: $5..600L/min$
- Resolution: $0.5L/min$

SM2004

SM2004 is a magnetic induction flow meter and does not impede on the readings given by the pressure sensors. This flow meter has similar attributes to the SM2000 but is recommended for cooling fluid sensing by the manufacturer "SM2004 Magnetic Inductive Flow Meter" (2016).

- $16bar$ Pressure Capacity
- $-20° < t < 80°$ temperature range.
- Support for $4..20mA$
- Response time $0.35s$
- Measurement Range: $5..600L/min$
- Resolution: $0.5L/min$

G Series Pressure Turbine Meter

This product utilises a turbine to determine flow rate. The variations between the sensors discussed above and this one is the addition of a much higher temperature tolerance. This flow meter unfortunately may not work for future experiments as it utilises a turbine which may be damaged by moving debris. The G Series Turbine Sensor is not suitable for fluids of viscosities higher than $0.1Pa$.

- Pressure Capacity: *bar*
- Temperature Range: $-40° < t < 250°$
- Response Time: $100 - 1000Hz$
- Measurement Range: $3.8..760L/min$
- Viscosity: $< 100cP$

Table 2.2: Comparison of flow meters

Property	*Pressure*	*Temperature*	*Flow Rate*	*Resolution*
SM2004	16bar	-20..80 $°C$	5 ..600L/m	5L/s
SM2000	16bar	0..40 $°C$	5 ..600L/m	5L/s
G Series	340bar	-40..250$°C$	3.8..760L/m	N/A

There is little difference between the Flow meters which were discussed above. This is due to the similarities between sensors on market of the type which is required for this application. The ideal flow meter in this case appears to be the SM2004 although, SM2000 would be sufficient as the only differentiating factor is the temperature range which is sensed. The G Series sensor is not suitable despite ideal sensitivity range, this is due to the mechanism through which it operates.

2.6.3 Variable Frequency Driver

A variable frequency driver is necessary for the experiment to proceed. The variable frequency drive is one component in a number of components which decide the rate which fluids flow through the system.

OptiDrive ODP-2-52185-3KF4

The OptiDrive VFD is supplied by single phase power from a 250VAC source. It produces an output from 0Hz to 50-60Hz with 3 phase output proportional to the input signal which is fed to the component. on the inputs provide to its input terminals. "Optidrive G200" (2019)

- Frequency Range: $0..500Hz$
- Resolution: $0.1Hz$
- Output Voltage: $230VAC$
- Max Output Power: $18.75kW$

Delta VFD185B43A

Delta VFD-185B43A provides a three phase output of 230VAC given a single phase input. Its frequency can vary in the range of 0.1 to 400 Hz. Unfortunately there is no resolution specified on the data sheet, this could imply there is no limit on the accuracy of output however this is unlikely due to the design of general VFDs.

- Frequency Range: $0.1..400Hz$
- Output Voltage: $230VAC$
- Max Output Power: $18.5kW$

2.6.4 TEKDrive TEC-3-440390-3#42

TEKDrive offer a range of functionality with their variable frequency drives "TEKDrive" (2019). The *TEC-3-440390-3#42* provides support for a motor of up to $18.5kW$ with $230VAC$ 3 phase supply. TekDrive also support Bluetooth connectivity with their products. Conveniently this VFD also offers the ability to switch the voltage down to $110VAC$.

- Frequency Range: $0..500Hz$
- Resolution: $0.1Hz$

- Output Voltage: 110..230VAC
- Max Output Power: 18.75kW

Table 2.3: Comparison of Variable Frequency Drives

Property	*Freq. Range*	Resolution	# Phase	Output Voltage	Power
ODP-2-52185-3KF4	0..500 Hz	0.1Hz	3	230VAC	18.75 kW
VFD-185B43A	0.1..400 Hz	-	3	230VAC	18.5kW
TEC-3-440390-3#42	0..500 Hz	0.1Hz	3	110..230 VAC	18.75kW

2.7 Communication

There are a number of different communication means which are currently being used in control systems for analog sensing, these have been outlined below.

2.7.1 4-20mA

The older industry standard is known by its name 4-20mA where by signals are transmitted from sensors to a receiver by varying an the current flowing through the sensor back to the receiver. 4-20mA has a number of benefits over alternative communication protocols Firth et al. (2017):

- Invariance due to varying impedances.
- Reduced error when compared to voltage signals.
- Simple to test.
- Simple to setup.

4-20mA also offers safety and versatility with low currents and the ability to easily swap sensors on the go Howarth (1992).

Finally the simplicity of 4..20 allows for easy transmission of analog data which may better reflect the values of a sensor and also has an advantage in that it is less restricted by factors including an timing as found in RS-232C, RS-485 and other synchronous protocols.

2.7.2 HART

Highway Addressable Remote Transducer, is a more modern implementation of 4-20mA. HART protocol enables a sensor or device to transmit both analog and digital data simultaneously through the same line. Communication via HART requires more setup but can deliver information much faster.

Hart maintains the simplicity of 4..20 but is far newer having been introduced in 1986. It is however far less regularly implemented in sensor equipment for the reasons discussed above. Cobb (1996)

HART operates at a rate from 1200Hz to 2100Hz, requiring synchronisation hardware to enable its operation Cobb (1996). This is not feasible for smaller systems, as it space utilisation is crucial.

Some of HARTs benefits include the following Howarth (1992):

- Remote Device communication and diagnostics
- 4-20mA systems need not be rewired

2.8 Controllers

The selection of hardware for performing sensor measurements and readings is another point of contention between the research and industrial communities.

Industry currently rely on the use of PLC programming for larger industrial systems. The main reasons for this is the simplicity/atomicity and security which is implied with such a system.

Research communities prefer to utilise post processing utilities to analyse the results of their experiments and as a result choose to use faster more evolved languages to achieve results in less time making full utilisation of hardware resources. IEC61131-3 is suggested to be the new frontier for PLC development and implementation Liang and Li (2011).

Liang and Li (2011) made an explicit note that PC-based systems are lower cost, easy to use and perform better under computationally complex tasks. It was also noted that these systems are becoming the norm for industry too.

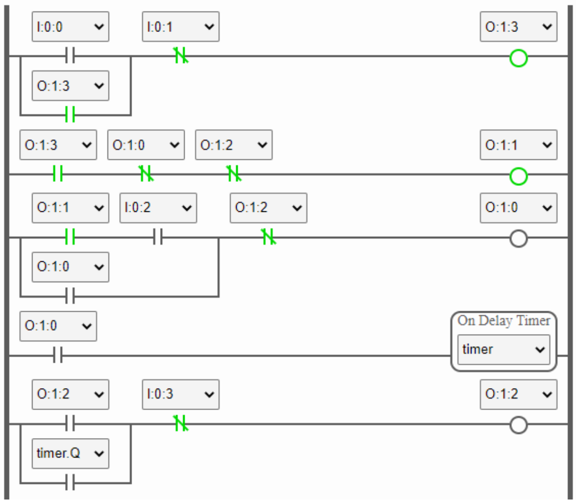

Figure 2.8: Example Of Ladder Logic

Examples of software which is used to run such software applications include Matlab, Simulink and Labview. This allows for speed of implementation with guaranteed speed and the ability to use multiple cores simultaneously.

There many advantages and disadvantages which differentiate the operation of PLC and computer systems Alphonsus and Abdullah (2016). Some of these are described below.

2.8.1 Hardware

Hardware is fundamentally different when comparing the two technologies.

PLC systems generally operate as single core solid state devices operating sequentially on one instruction at a time. Rullán (1997). Conversely modern computer systems typically operate with multiple cores and are assembled with removable media of varying speed and capacity.

A PLC system is build to support a vast array of inputs and outputs, this is different from a computer system which is generally limited to very few IO ports, typically a computer system is limited to very few external devicesRullán (1997).

Another benefit of PLC systems is their rugged versatility over which they can operate. Typically PLC systems are designed to include many redundancy measures which assist in maintaining operation indifferent to interference events such as brownouts. This provides incredible versatility and adaptability which makes it ideal for industrial applications where potential losses need to be minimised. The result of this is PLC systems are often used when safety is a requirement Rullán (1997).

2.8.2 Software

PLC systems operate according to a simple procedure which can be summarised in 3 stages Rullán (1997).

- Input
- Processing
- Output

This differs from the operations of a computer system which typically does not conform to any ordered cyclic operation. Computer systems typically perform various input and output operations concurrently or even in parallel.

There are a number of different languages which can be used to operate PLC systems. The primary language used is known as ladder or ladder logic. shows an example of a ladder logic program which could be used to control a fluid mixing plant.

Concurrency The differences between concurrency available in a PLC system and that of a regular computer are significant. Generally PLC systems are single core with a low clock rate there and only a single thread executing operation Alphonsus and Abdullah (2016). This is observed with the ladder programming logic where there can only be one state change for each scan cycle. Figure2.8 shows an example of a ladder program to be run on a PLC, it runs from top to bottom left to right.

The use of looping and time dependant tasks need to be considered here as the system runs all tasks sequentially on what is effectively the same thread. If a single task blocks for too long the PLC may fall behind, failing to perform operations correctly and potentially desynchronising.

Computer systems on the other hand operate with multiple cores on a single CPU. This allows for the computer to execute multiple tasks simultaneously.

The most interesting aspect of the use of computer systems for simulations is that the simulation can make use of the multiple cores of the system to run tasks at the same time.

Speed As mentioned above the speed of the two systems is vastly different. A PLC system typically has a scan time of between 600 Mhz and 1.2 Ghz. This is sufficient for the typical instructions, this indicates how frequently the system will execute a scan as indicated above. The scan in this case does not perform complex logic but it does the job. Another advantage of such a small scan time is the power usage of the PLC is far less than the equivalent for a computers.

A computer with a modern CPU typically has a clock rate of between 1 Ghz and 4Ghz which is vastly different from the clock speed present on a PLC system. In the case of a computer system the clock speed may even be measured by multiplying the number of cores by their respective clock speed to try to give an indication of the approximate number of instructions executed per second. In this case a clock speed of between 16Ghz and 128Ghz is easily achievable with higher end CPUs.

Memory The final area of analysis and separation between a PLC system and a computer system is the availability of random access memory. In the case of PLCs the memory availability is restrictive as the system is more focused on reliability than performance. This is generally enough for the operations which a PLC system is performing. The memory of a PLC system is not typically used to perform long term operations rather IO is performed with other systems such as hard disks or other computer systems. which handle the data for logging or other purposes. The issues with the current.

2.9 Analysis Software

2.9.1 LabView

LabView is a piece of software which is capable of interfacing with national instruments control equipment including NI Compact DAQ. This is preferential over conventional languages as less time is spent developing fundamental control systems such as user input and error checking Kirkman and Buksh (1992). Kirkman suggests a new era of programming language is being introduced which now allows programmers to state what they want to achieve instead of stating how to do it. This is an idea which LabView fundamentally supports in its design. This is referred to by Kirkman effectively

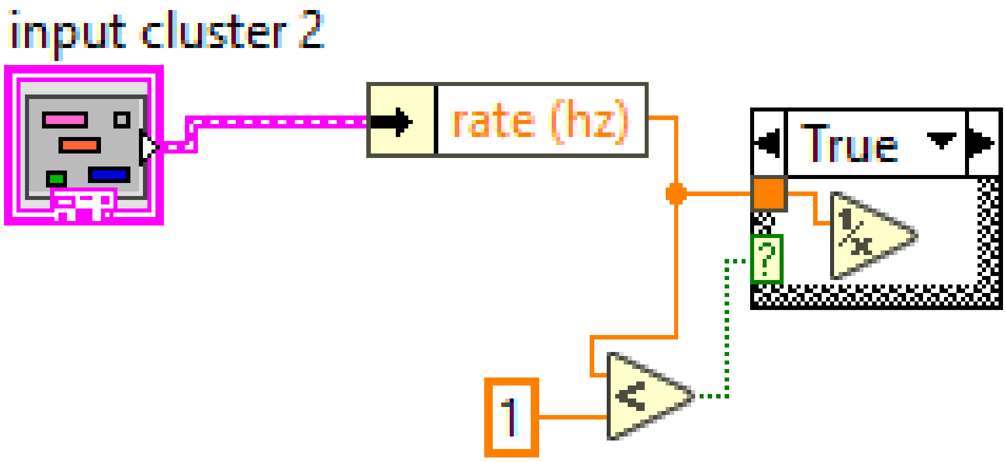

Figure 2.9: LabView Control blocks

as graphical programming and this can be seen in the interface which is presented to the developer shown in Figure2.9, this snippet implements Eqn. 2.1.

$$\begin{cases} f < 1 & \frac{1}{f} \\ f \geq 1 & f \end{cases} \tag{2.1}$$

Kirkman went on to detail a number of trials performed on LabView with comparisons to older languages like C and Ada. One similarity is the implementation of 'Virtual Instruments' which represent what would be a function in other languages Kirkman and Buksh (1992). Features noted by Kirkman included, the live compilation process of LabView where each virtual instrument is compiled before it is used making the program extremely fast, another feature was the front panels made available through lab view. Front panels in LabView allow for ease of implementation for any projectKirkman and Buksh (1992). It is quick and simple to implement a complex control which may take pages of code in other languages.

Lab view supports data logging to a binary form known as TDMS. This format allows for extremely large amounts of data to be stored. THis data can also be easily read into other software packages for subsequent analysisUrbikain and López de Lacalle (2020).

Even though LabView provides a good environment for the collection of data it does not provide a good means of data analysis after collection as it is not built for this. LabView does however support communication over the TCP layer thus suggesting data processing could be performed else ware.

2.9.2 MATLAB

MATLAB is another piece of software widely used in the research community. It provides many tools which can be used to enhance analysis without burdening the person who is writing the code. These primarily consist of pre-written mathematical analysis programs which can be used without prior knowledge of the software.

Benefits of using MATLAB include the ability to perform complex mathematical functions as well as performing operations defined in libraries which can be installed separately Dietrich, Masiero, Pollow, Scharrer, and Muller-Trapet (2010).

This is beneficial to researchers as instead of having to providing additional support libraries to each is the way Matlab implements some of its speciality functions, through packages. Such a package is the *Signal Processing Toolbox* which contains various functions which can perform. There are several toolboxes which perform similar functions related to their specific problems Berzborn, Bomhardt, Klein, Richter, and Vorländer (2017); Perraudin et al. (2014).

Lastly MATLAB supports the generation of graphs for its users. These graphs support both log and semi-log axis which are useful for modelling frequency analysis as may be necessary.

Trials have been run where communication between MATLAB and LabView was tested Ding, Sun, Huang, Lin, and Qian (2010). This was termed 'Intelligent Control' Ding et al. (2010). Communication was established and utilised to enable the real time analysis of LabView readings. This would be greatly beneficial to future analysis performed by researchers and industry alike in all disciplines.

2.10 Conclusion

To summarise the analysis in this document, a number of different sensors have been reviewed and analysed. Understanding of the limiting factors by identifying the component which will first limit the collection of data. The data collected suggests that the components will restrict performance if any one is chosen due to its superiority in one area. The key notes relating to this are summarised below.

- Combination of flow meters with pressure sensors is a crucial. Flow meters can only operate within a specified range while pressure sensors are designed to measure over a much larger range. Selecting a combination of the two will involve balancing the capabilities of both sensors in making a decision.

- The use of a personal computer in place of a PLC system is considered greatly beneficial as a computer system will allow for analysis in real time.

- The variable frequency drive which is selected should be capable of varying the rate and power which is sent into the network in which it is implemented.

Later analysis of potential signal processing limitations were discussed. There is little limitation on the ability for analysis to be performed with post processing however an IP communication between MATLAB and LabView would be highly advantageous for future research as it would allow for real time processing of data.

Chapter 3

Specifications, Methodology & Development

3.1 Overview

The project was developed incrementally with each component being formed separately then tied together into a single functional system.

The initial specifications, development and method used through this project are presented here. The main headings of hardware and software are used to separate the ideas which are presented throughout the paper. This is used to break up the development of the project into smaller and more manageable sections.

This final system presented here performs three core functions utilising all the components developed individually. The components discussed include reading, writing and logging software, also covered is the hardware which is required to incorporate this technology into the system.

3.2 Capabilities

In order to streamline development the systems capabilities and requirements must be identified. This will ensure the system is developed in such a way that it can be used for the intended purpose.

The requirements stated here have been reviewed and adapted from information which was reviewed in Chapter 2 in order to better reflect the systems functionality in relation to the operation of the system under the constraints which were realised during the systems development.

- Reading
  - The system must be capable of reading a 4..20mA signal indicating a devices measurement. This is crucial to ensure compatibility with other devices which are present in the system. This compatibility is essential to ensure hardware will be compatible in the future. This communication was used to communicate with the following components during the experimentation phase of the project.
    * NI cDAQ 9185
    * NI 9203 - Analogue Input
    * NI 9265 - Analogue Output
- It must be possible to attach at least two 4..20mA sensor for simulations sensor reading. This requirement ensures the system is capable of recording data from more than one sensor. This requirement is necessary for more complex analysis comparing pressure differentials in fluids.
- The system must be capable to processing its measurements in real time to ensure data processing does not have to be performed later. This will consist of providing the option to scale the output proportionally according to the sensor which was used enabling the user to define the sensors before hand and save time processing data in the future. Configuration options must exist for the following sensor measurements with varying scaling.
  - Pressure (bar)
  - Flow rate (l/s)
  - Amps A
- Ranges presets may exist for the following intervals:
  - 0..1

  - 0..40
  - 0..6
  - 0..10

- The system should be able to write a user defined functions. This will ensure the system is versatile enough to handle any situation.

- Data Collection

  - Data should be recorded for both the input and output signals from the system. This ensures the synchronisation of the signals which are read and written. Without this requirement the output signal would have to be recorded before being written to the output module and due to the internal buffering within the output module this is not feasible.
  - Data should be recorded to the *tmds* binary file format. This ensures not only can the data be easily read by other programs but it can also be easily written to the file for output.

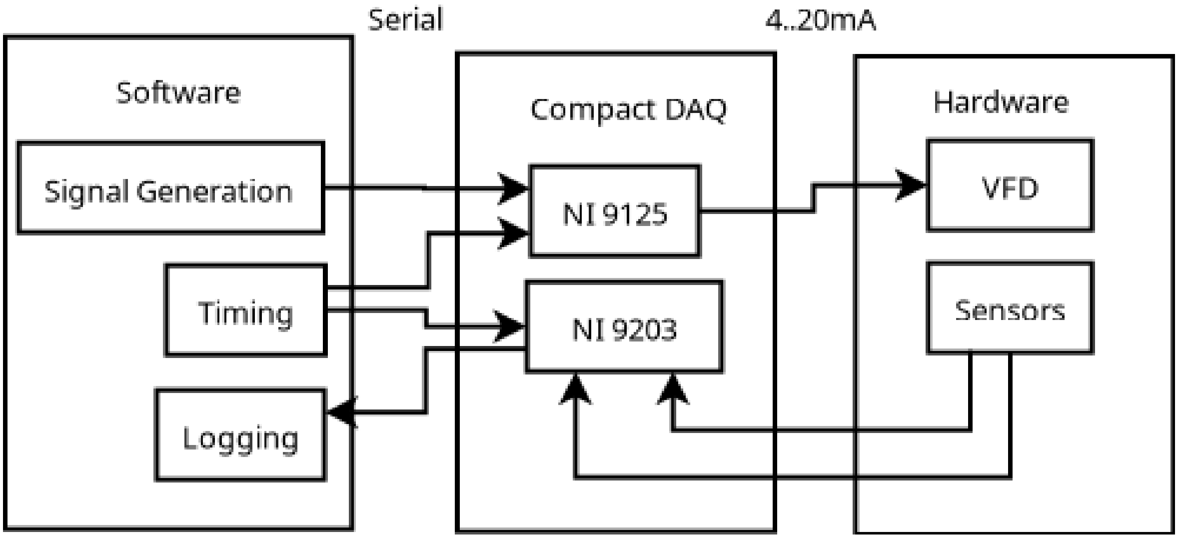

Figure 3.1: Project Block Diagram

3.3 System Diagram

Shown in figure 3.1 is the block diagram with a focus on the hardware components which are involved in the system. Key aspects of the system shown are the use of 4..20mA to send a receive signals from the sensors. Also shown in the diagram the separation between the sensors and software. It is important to understand that the software is not run on the modules which are being used. Instead the modules are used by the software which interfaces through the use of the cDAQ 9185 chassis. This is what enabled the use of the hardware.

3.4 Operational Overview

The operation of the system consists of the following steps. These steps were used in the testing phase to verify the systems functionality.

1. Sensors are connected to the system via M12 Connector.
2. Sensors are named and appropriate scaling is configured within the software interface.
3. Appropriate parameters are selected for the trials including:
    - Frequency $f = 1000Hz$ (default).

  - Appropriate log file location.
  - Signal specification for both input and output signals.

4. The user selects the `ACTIVATE` button. The system will begin to display readings on the right waveform view.
5. The user will select the `transfer` button. The system will transmit the signal specified by the user to the VFD. The response of the System with the appropriate sensors will be recorded to the files which was specified by the user.
6. This will continue until transmission of the signal to the VFD is complete.
7. During transmission the current signal observed by the sensors is shown in the waveform view on the right.
8. When the user feels the signal has appropriately settled they can end the recording process by pressing the `stop` button. This action will finalise the saved data and ensure the data file is closed properly.
9. Once this process is complete is may be re-initiated from stage 5 to record another session.

3.5 Hardware

With an understanding of the requirements and intended process which the system will be used discussion of the hardware may commence. The hardware configuration used for this project is critical to understanding the operation of the system as a whole. The following hardware components were used in the experimentation phase of the experiment to verify the design however the system has been designed for all 4..20mA sensors and should be compatible with any National Instrument hardware which performs similar functions to the hardware which is found in this specification.

- NI cDAQ-9185 with the following expansion modules:
  - NI9265 Analogue Output
  - NI9203 Analogue Input

- Windows Laptop with National Instruments Lab View software installed
- Sensors

In order to perform the experiment the following hardware will be used.

- Torque Converter
- VFD
- Motor
- Pumps
- M12 Connectors and cables.

These components will be discussed later after discussion of the hardware and software has taken place.

3.5.1 Hardware Description

The DAQmx Chassis (cDAQ-9185) accepts up to four expansion cards each connected by inserting the card into the appropriate DE-15 socket. This is shown in Figure 3.5 where both the Analogue cards are shown.

3.5.2 NI9125 Analogue Output

The NI9125 is used to produce the signal designed by the user and feed it into the VFD which is connected. The output module also enables the system to produce multiple outputs with minimal overhead cost, the advantage of this is the output module can have another output fed into the input module discussed in Section 3.5.3. This configuration is necessary as the 4..20 mA specification specifies that there should be only one input receiving the signal. If this requirement was not followed the system may have trouble either recording or transmitting the 4..20mA signal to the VFD. The current divider rule would apply to the current transmitted to both the VFD and the input module, this is shown in Equation 3.1 this shows that for the output currents I_1, I_2 there is no way for both output signals to share a similar output signal between 4..20.

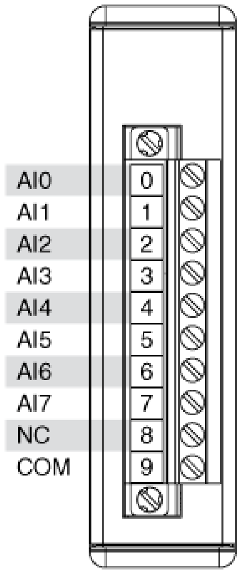

Figure 3.2: Schematic of NI9203

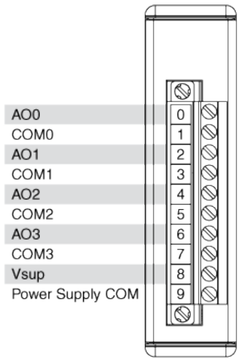

Figure 3.3: Schematic of NI9265

$$I_{in} = I_{out} = I_1 + I_2 \tag{3.1}$$

3.5.3 NI9203 Analogue Input

The input module is capable of reading 4..20mA signals from various 4..20mA capable sensors and components. The input hardware was utilised to its maximum potential as detailed in Section 3.6 by providing the ability to specify any number of input channels for data to be recorded from.

A limitation exists in the hardware due to the nature of the software, it is not currently capable of capturing more than 8 signals (via the 8 channels on the input module) at one time. This limitation was considered out of the scope of this experiment

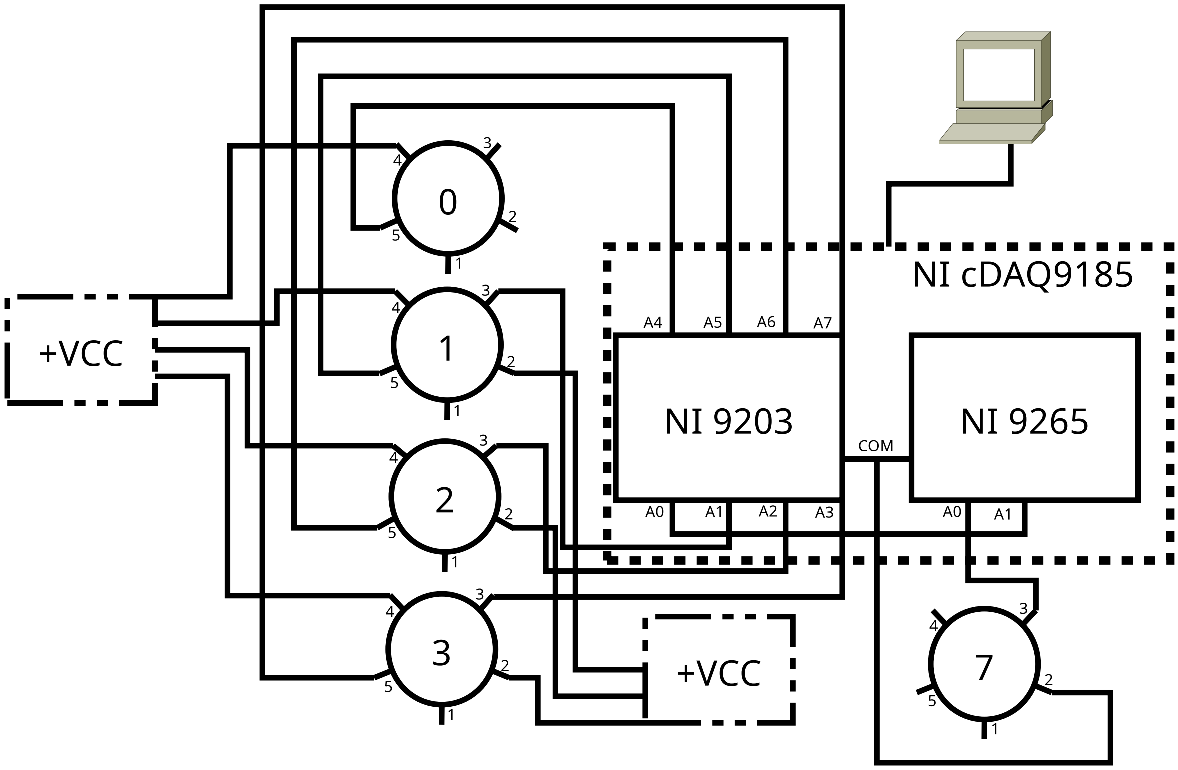

Figure 3.4: Wiring Diagram

as the requirements specified above have already been satisfied.

3.5.4 Experimental Configuration

The system was wired with the configuration shown in Figure 3.4. As mentioned earlier the key modules illustrated in this diagram are the analogue modules for input and output, the DAQmx Chassis and the connectors which are shown in Figure 3.4.

The hardware components of the experiment were added to the chassis. Wires were soldered and run from the modules to the output and input ports on the frame according to Figure 3.4. The end result of this is shown in Figure 3.5. Eight analogue wires were connected to 4 M12 connectors each with two analogue inputs connected. This allowed for the use of an additional 4 ports which were made available after these connections were made, one of which was used for the output signal.

Figure 3.5: Picture of chassis mounted with modules and wiring completed

The VFD which was used is shown in 3.6. The VFD was set to receive a $4-20mA$ signal and produce its output waveform varied linearly between $0-50Hz$ according to the signal provided.

Figure 3.6: VFD and motor used in experiment

Finally the chassis is also wired to a range of sensors which provide the 4..20mA

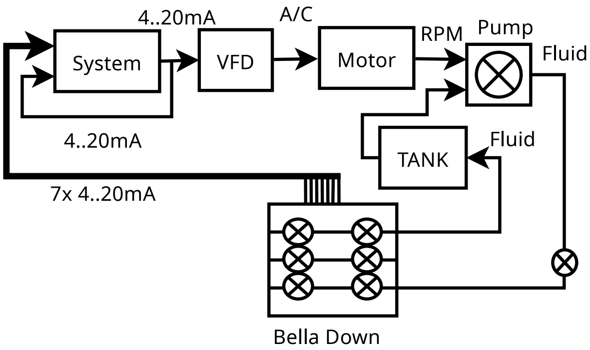

Figure 3.7: Hardware Diagram

signals expected from the selected sensors which were used on the day. The diagram illustrating this setup is shown in 3.7, the physical hardware is shown in 3.8. This image also shows the system of pipes named 'Bella Down' by the Petroleum Engineering team.

Figure 3.8: Overall operation of system

reading and writing of the signal to and from the hardware. Consideration was given to synchronising the output signal with the input signal however it was decided that it would be more efficient to read the output as another input in order to improve the accuracy and reliability of the measurements.

3.6 Software

The software component of the project consists of a number of features. Initially the use is required to connect the sensors to the hardware via M12 connectors. This also includes the connection of the VFD to allow for signal control.

Software components make up the majority of the project. A lot more consideration has gone in to the development of the software components than any other section of the project due to the projects nature.

As outlined above in Section 3.2 there are 3 tasks which are required to be run in parallel in order for the project to achieve its objective. These components are reading, writing and logging.

Each of the components discussed below have been designed to run in their own separate thread and should not block other threads while they are performing their operations. As a result of this there is a need for communication to be performed asynchronously.

Asynchronous communications are implemented through the use of a message queue as discussed earlier. The use of the message queue allows for all parts of the program to perform their operations with ease and without over utilising system resources. This is done by introducing sleep periods within the looping sections of code. Due to the sleep component there is rarely a time when all parts of the system are performing an operation at once, there is also never a time when the system suffers from full resource utilisation.

3.6.1 Main Program

The main application within the program shows a number of useful features of the program which need to be explained in order for the data flow of the rest of the program to be understood.

The first part of the program which needs to be understood is the concept of the `INPUT CLUSTER`. The input cluster serves to allow the user of the program to enter the data relating to the sensors which they intended to use in this instance of the program.

The input cluster also contains information which the user can choose to define relating to the sampling information which is used for the experiment they wish to

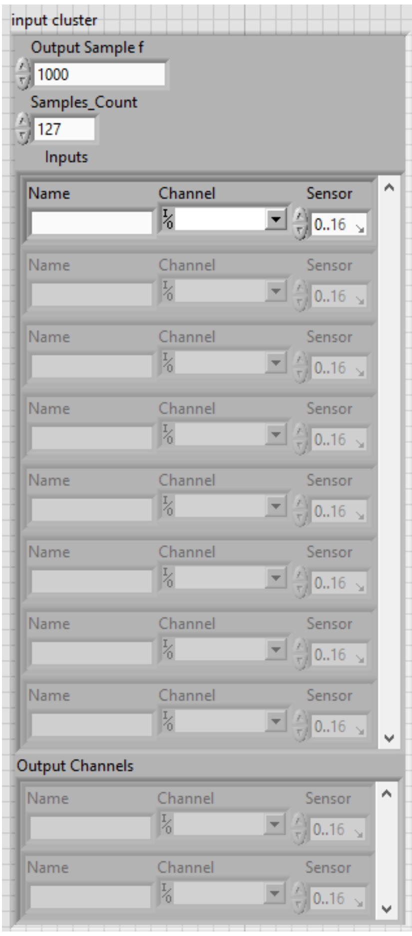

Figure 3.9: Screenshot of the input cluster implemented in Lab View

perform. This information is entered in the `Output Sample f` and `Samples Count` fields shown in Figure 3.9. The ability to vary the generated signal frequency along with the sampling frequency allows for much more control over the resolution of signal which is sent to the VFD.

It should be noted that the frequency cannot be changed during operation as the configuration is sent to the hardware components which then manage the low level synchronisation and timing, this makes it impossible to control the frequency directly as the system is operating.

Another key aspect of the main Virtual Instrument is the setup of the input signal equation and output file location. This is all done through a menu in the main virtual instrument.

The menu presents the use of a number of tabs. The first tab offers the user the ability to enter a filename which the program can load signal information from. The option is also presented to store a file discussed in the next section.

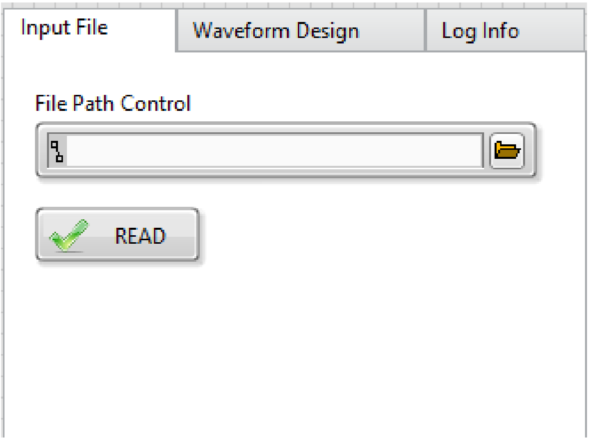

Figure 3.10: File information cluster

The next tab titled `Waveform Design` can be used to generate a signal which follows the appropriate signal. This system takes 3 parameters; the signal equation, the duration and the sampling frequency. When the user is finished entering this information they may click apply to view a preview of the equation then they may either save of continue to edit the equation to their liking. When sending the signal to the VFD it is important that the signal is shown in the preview as this shows that the signal is loaded into the system and is ready to be sent.

Another important section to discuss is the button clusters which are used to control the operation of the program as a whole. The button clusters include the `Main control cluster` and the `Transfer` and `Stop` buttons. Discussion will begin with the main control cluster shown in 3.12.

This control cluster has two important buttons, the `ACTIVATE` button and the `Run` button. These buttons together are used to start the system and to start the measurement process. Assuming the stages discussed above have been completed then the user may first press the activate button with the appropriately connected hardware. Once the system is initialised the user can press the `Run` button to begin reading information from the system. Once this state has been reached the system is ready for experimentation to begin.

3.6.2 Main VI

The main loop is made up of a number of components. The first of which is the initialisation blob shown up the top with the grid icon.

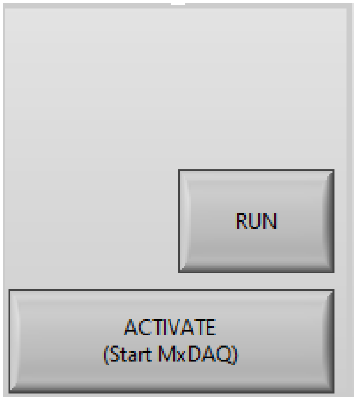

Figure 3.11: Screenshot showing the main buttons used for starting and stopping the reading process.

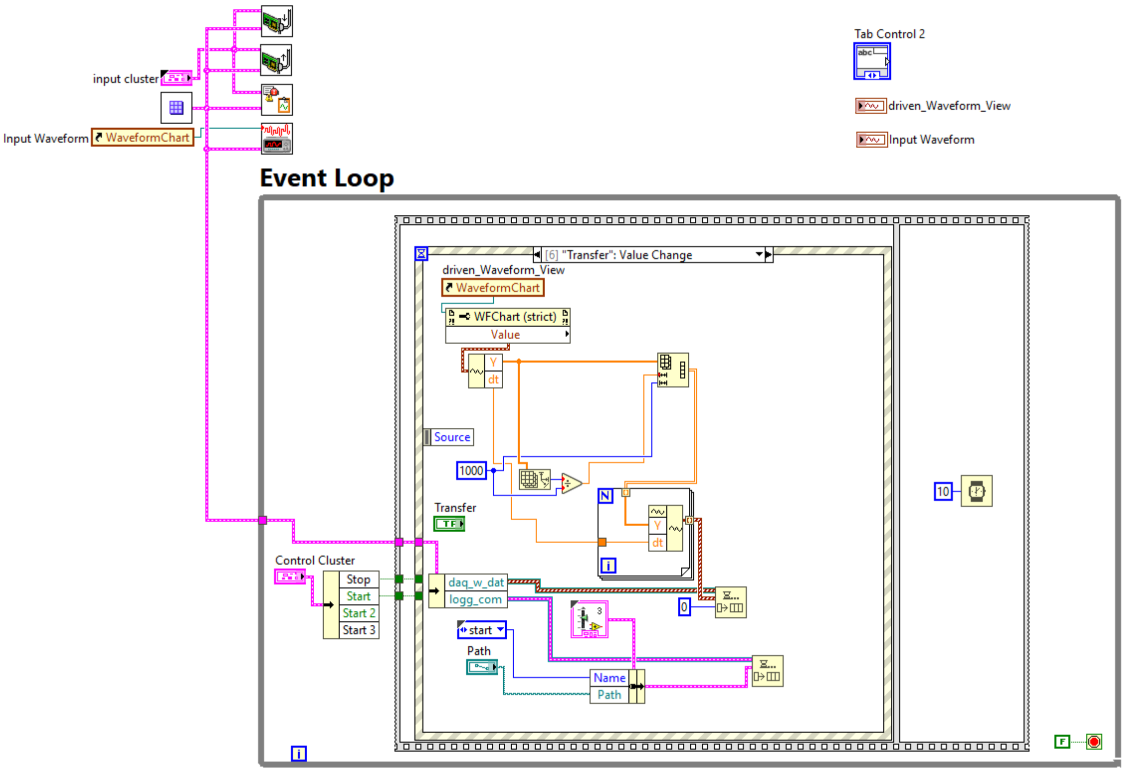

Figure 3.12: Overview of main code

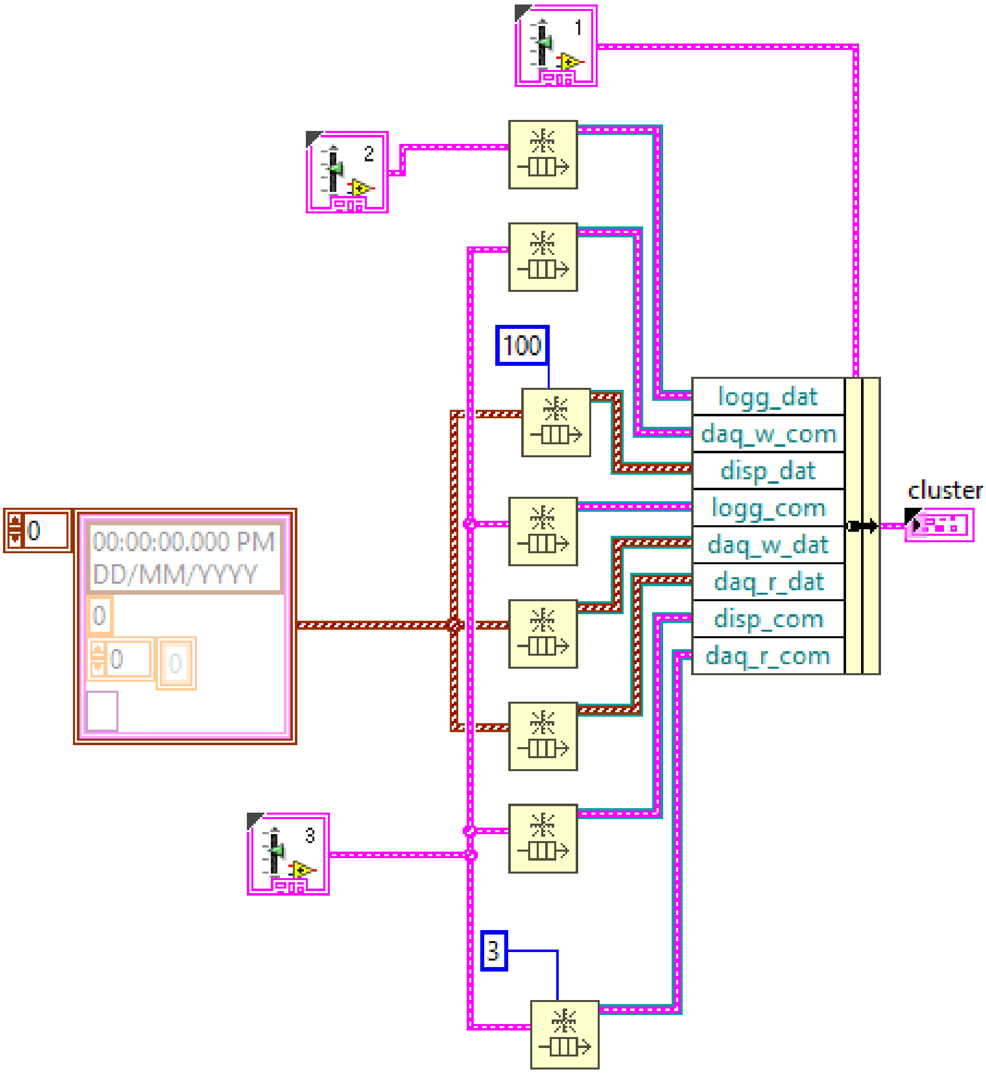

Figure 3.13: Initialisation of communication queues

The main loop is made up of a number of components. The first of which is the initialisation blob shown up the top with the grid icon. this is shown in Figure 3.13

As is shown in the figure above, the block takes no inputs and produces a single output. The output produced consists of a collection of queues which can be used to send data between threads later on in the program. The Queues are initialised here using the obtainQueue sub VIs provided by Lab View. These VI's are provided an argument which signifies the type of data which the queue should contain. The obtainQueue block can also be provided an integer input to limit the number of items within the queue. This is done for the disp_dat and daq_r_com queues. The naming convention chosen for the queues represents the process the queue is involved in being

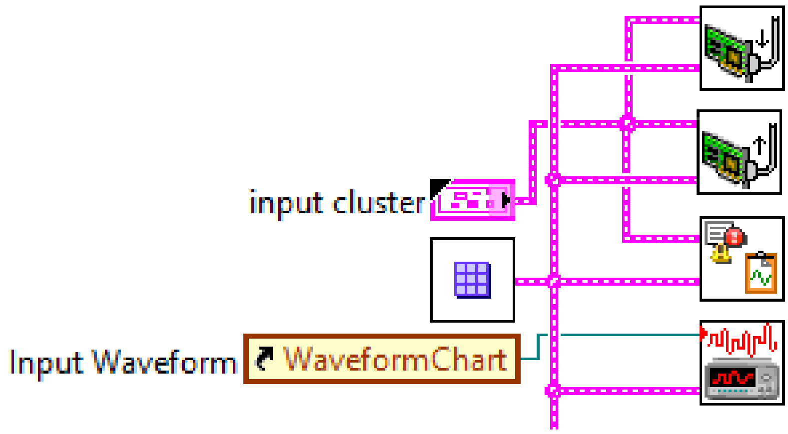

Figure 3.14: Task initialisation by the Main VI

data acquisition or logging, optionally the direction of data flow; read or write and the type of data contained being data or communications. This naming convention made it clear which queue was necessary for each part of the program.

The next section of the main program is the data initialisation VI's. These VI's represent the processes involved for either reading, writing or logging also included here is the display VI which is responsible for displaying the information which is provided by the reading VI.

The input cluster shown in Figure 3.14 has been discussed prior to this and contains all the user provided information describing the address and signal types to capture and store. Also shown in Figure 3.14 is 4 VI's (on the right) which take the data from the queue initialisation VI and the data from the input cluster to perform their duties. The Waveform Chart reference included is passed to the display VI to enable it to display the waveform in the projects main VI rather than a sub VI of the project.

The final component of the main VI is the event loop shown in Figure 3.15. The event loop is necessary to ensure the operation of the program is smooth. Rather than polling for inputs this method of input provides a responsive and fast user experience.

The event loop takes a single input, the queue collection and performs event based logic. The event loop responds to the events shown in 3.16.

The events listed in the figure above consist of button presses and text entry events. Each of these events has its own event handler handles by the event loop VI.

The start and stop buttons of the event loop are the simplest components as they require minimal processing. The figure shown below illustrates the contents of both the

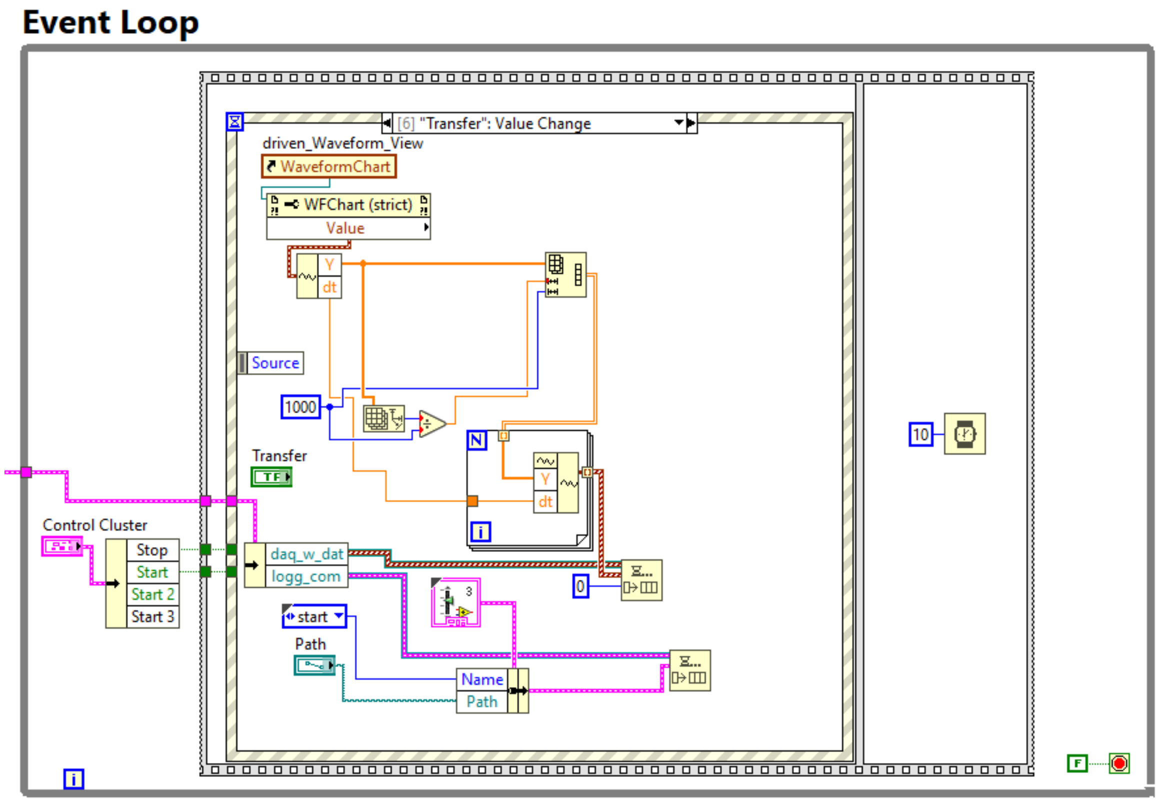

Figure 3.15: Main Event Loop

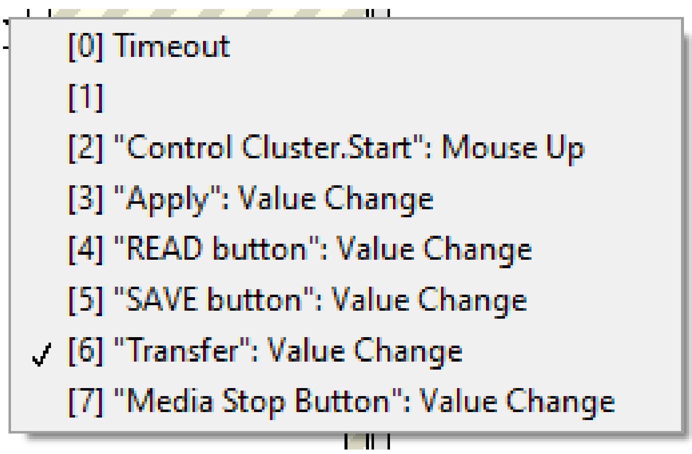

Figure 3.16: Event loop events

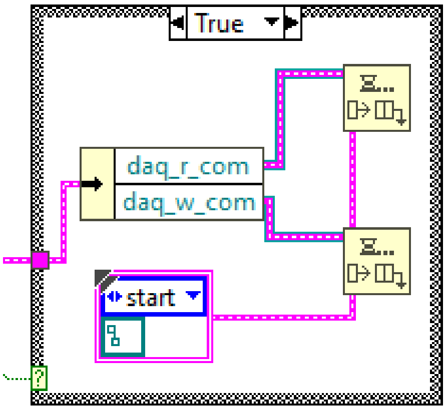

Figure 3.17: Start event

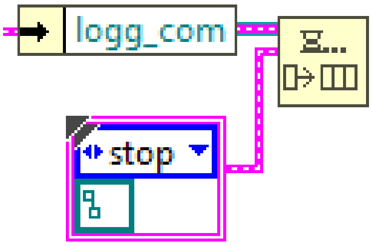

Figure 3.18: Stop event

start and stop components which are present within the VI. It is also worth noting that the stop button sends its instruction to the logger sub VI while the start instruction is sent to both the reading and writing sub VI's. The reading and writing Sub VIs have already received their initial configuration thus they just need to be instructed when to start running. The start loop accomplishes this, each event is shown in Figure 3.17 and 3.18.

When the user selects the apply button to display their equation on the right hand waveform the process shown in the figure below is followed.

This event takes the input from the users formula entered earlier along with the fields frequency, and duration and passes these into a formula Sub VI. The formula Sub VI processes the equation with the parameters specified and produces a waveform. This waveform is then stored within the waveform view on the left. This waveform view is taken from a reference as it is needed in multiple places.

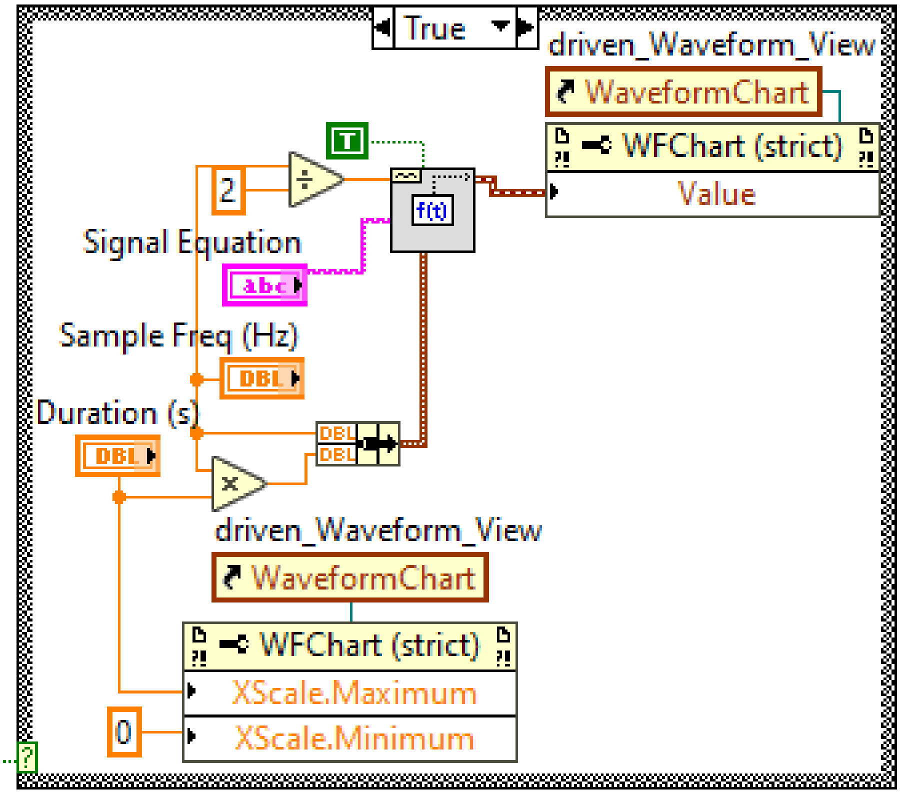

Figure 3.19: Stop event

Another event which occurs is the read event. When processing the instructions for the read event the program needs to perform a number of steps.

1. Inputs are fed from the File Path Control to the Read waveform From File VI

2. This VI reads the file and extracts the waveform from that file.

3. The waveform is then stored in the left waveform view.

4. The waveform is also analysed for their duration which is then used to set the width of the waveform view.

The save operation is another simple operation. As shown in the figure above the save button causes the current waveform stored within the waveform view component to be saved to a file using the `Write Waveform to File VI` with the file path specified by the user.

The final operation performed in the main Sub VI is the transfer event. The transfer event shown in the Figure above follows a complex but logical approach. The concept

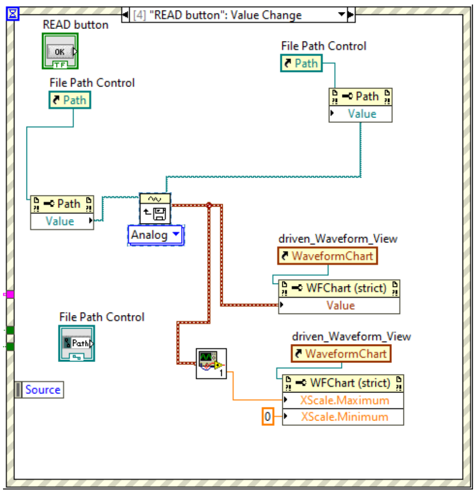

Figure 3.20: Read event

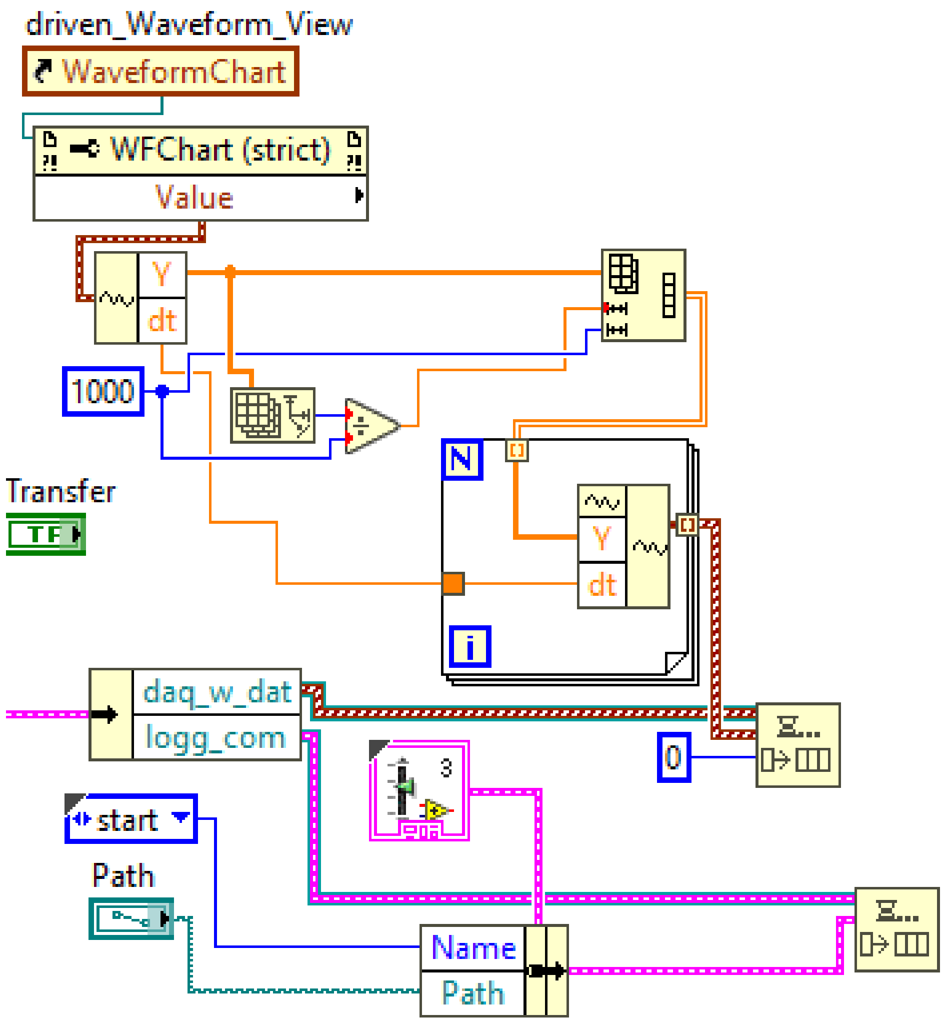

Figure 3.21: Transfer Event

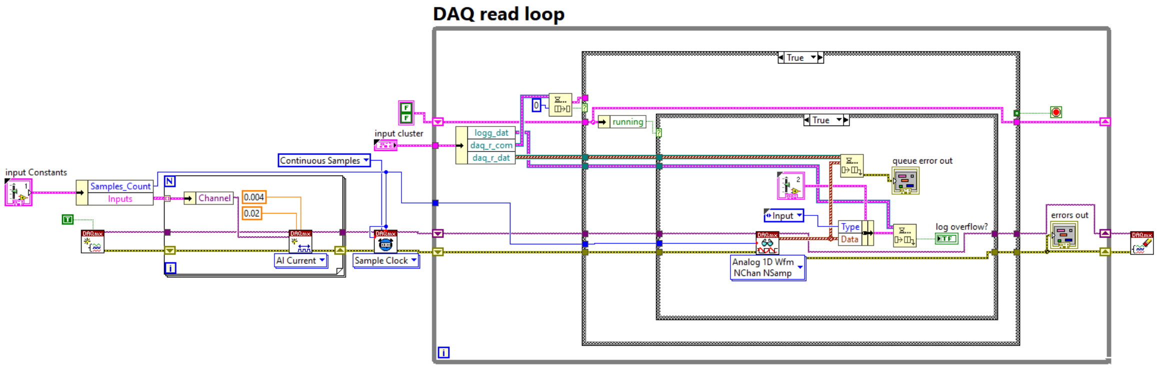

Figure 3.22: Read Overview

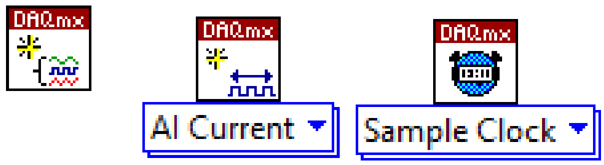

Figure 3.23: Virtual Instruments contained within Read SubVI

applied here is to break the signal down into chunks consisting of 1000 values. This ensures that the internal buffer of the NI9203 is not exceeded. Once the data is broken down into segments of length 1000 the array of segments is written to the daq_w_dat queue. Discussion of how this is received will take place later however it is important to understand the importance of breaking this signal down so that the segments can be sent individually to the hardware.

3.6.3 Reading Sub VI

The reading VI interfaces directly with the NI9203 and performs read operations from the list of sensors which was received upon initialisation. The reading VI as a whole is shown in the following figure.

The first component of the reading vi which requires discussion is the initialisation component. These instructions are run once to initialise the state of the VI to allow for the operations which are performed in the loop.

The initialisation section of the reading VI takes the input cluster which contains the information relating to the queues which were initialised in the main program. It also takes the information which was configured by the user and prepares those inputs

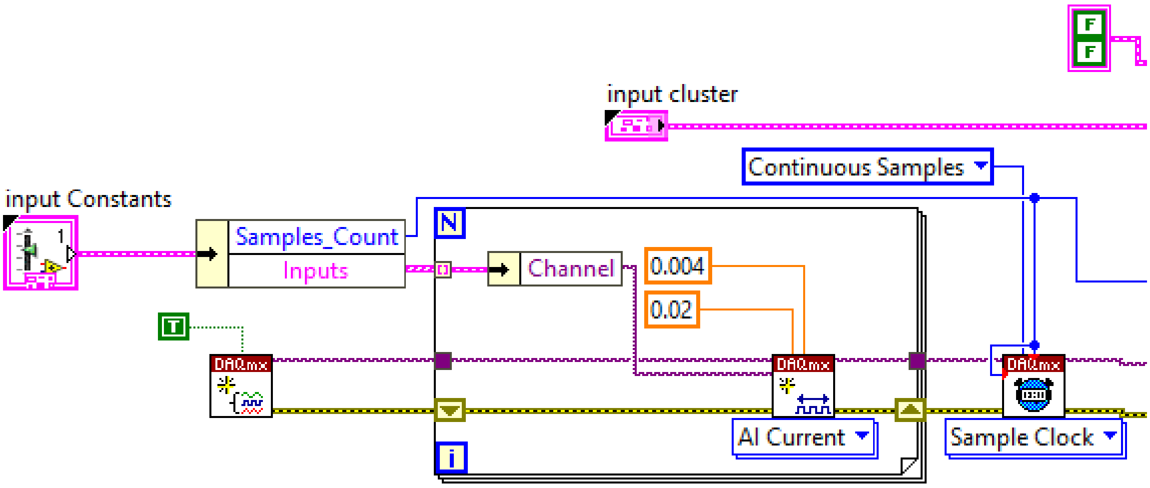

Figure 3.24: Read SubVI initialisation

in preparation for reading. This can be seen in the figure below with the use of the Create Task sub VI, the Create Virtual Channel and Sample Clock sub VI's. These are shown in the figure above. A for loop was used to allow for the user to specify more than once channel as input to the reading vi, as a result of the use of the for loop the read loop can handle an arbitrary number of sensors instead of being restricted to the number specified in the requirements. Each of the input channels specified is given its own virtual channel which can then be used to log information at a very high rate in parallel.

The next section in the reading VI which requires discussion is the loop. The loop was designed with a state based structure in mind and thus the operations performed should take very little time and should results in the change in state on the next iteration if necessary. The read loop is shown below with all components also shown within.

The read loop was designed to operate under a state based system, as a result of this the states within the read loop need to be capable of being changed. Discussion will begin with the implementation of the states then discussion of their changes will be mentioned.

The outer if statement shown in the figure above is used to ensure the VI does not run any read instructions until it is instructed by the user via a message from the main

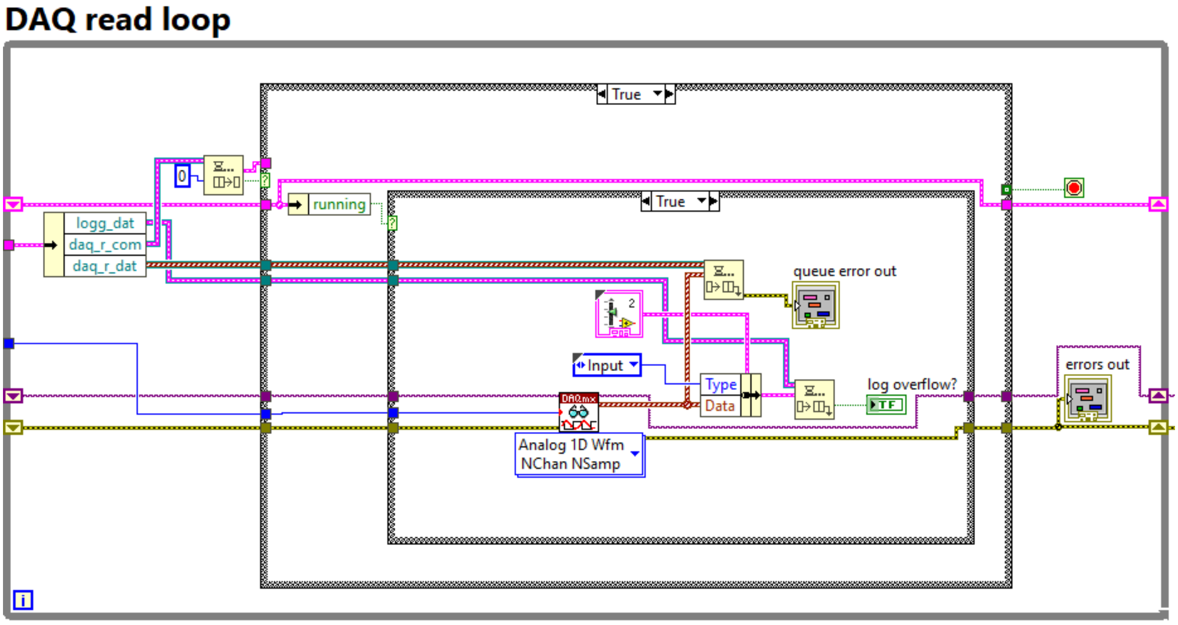

Figure 3.25: Screenshot of read loop

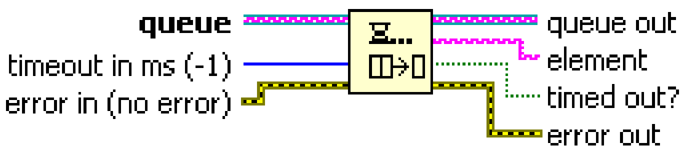

Figure 3.26:

event loop. The queue used for this is *daq_r_comm* which can be seen on the left hand side of the figure. The *daq_r_comm* is constantly dequeues using the Dequeue Element Operator, this is shown in Figure 3.26. A timeout of $0ms$ is used to ensure the loop does not wait for a message to be received at this stage instead it will check if the buffer has an item before continuing.

When an element is successfully dequeued the if statement which follows will execute under the true condition. This is because the timeout flag of the dequeue operation will be set to false as a result of there being an item within to be dequeued. When this occurs the message which was sent will be read as shown in the figure below.

The condition within the code shown in Figure 3.27 is simply testing which message was received. There are a number of cases which can be dealt with for this, they are start, stop, and end. The start condition is executed when the message received was a start message. In this case the sub VI will change its state to running if it is stopped or keep its state as it is if it is already running. This code will also modify the state of

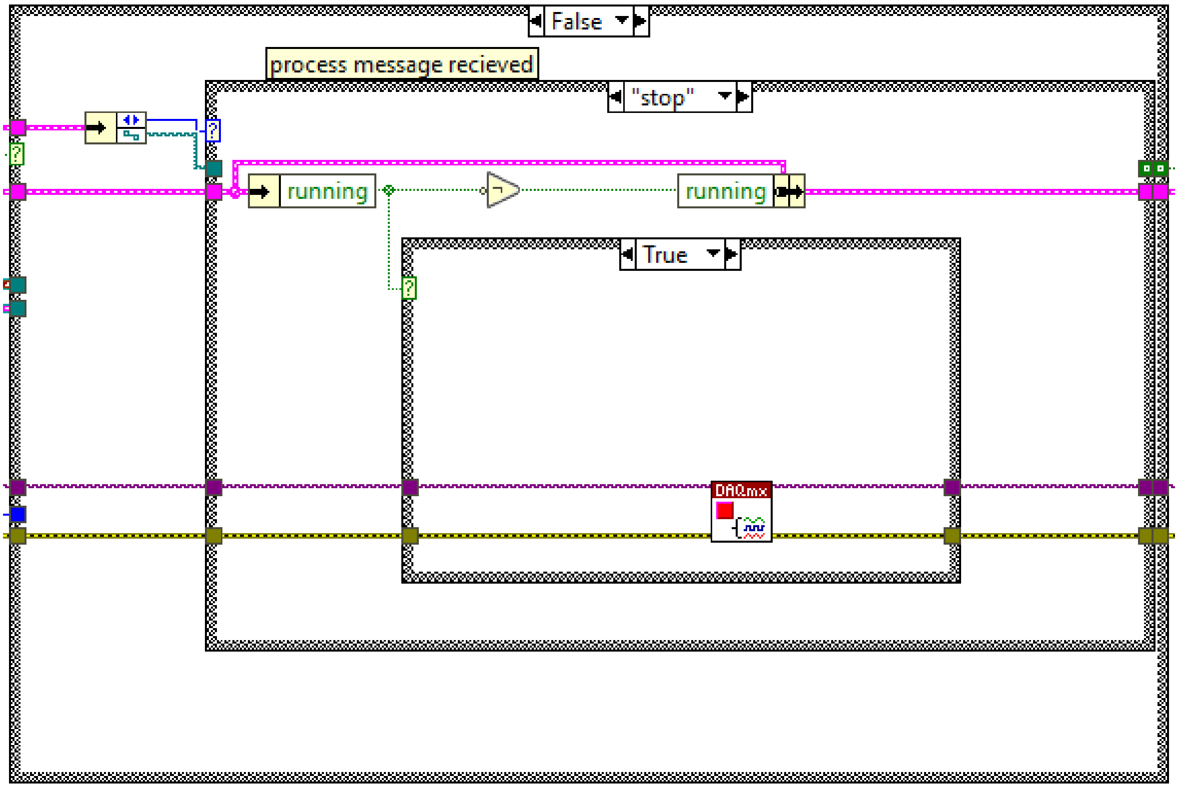

Figure 3.27: Code blocks executed when data is successfully dequeued

the DAQmx task to ensure it is running when the command to start is received. The operation of the start task is shown in Figure 3.28.

The operation of the stop and end conditions are very similar to the start case, the state is changed according to the current state and the desired state.

Figure 3.29 contains the operation which is run while the state is set to running. This state involves somewhat more complex operations than the previously discussed state machine as there is interaction with the hardware directly every time this loop is run. As shown in the figure a DAQmx Read Task is used to obtain the data from the physical channels which were configured in the initialisation phase. Once the data has been captured it is then placed in two queues. The first queue is the daq_r_data queue while the second is logg_data queue. The data is added to these queues without blocking, if the queue is full then the program will simply discard the most recent data which has become available. An notable feature of this section of code is the use of a complex data type containing the data which is being written to the logging queue. The reason the data needs a type tag added to it is to identify it when adding it to the log file. Without this tag the final data would have to be identified as input data

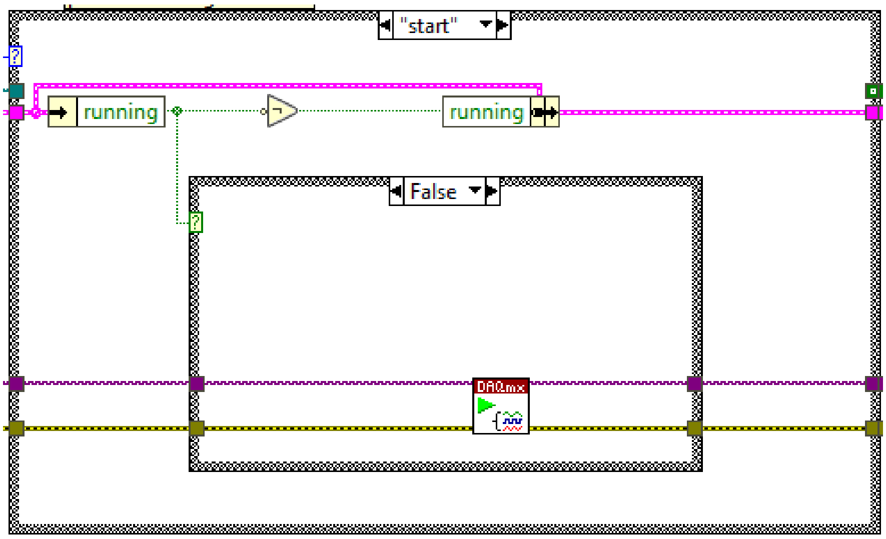

Figure 3.28: Code blocks for read start

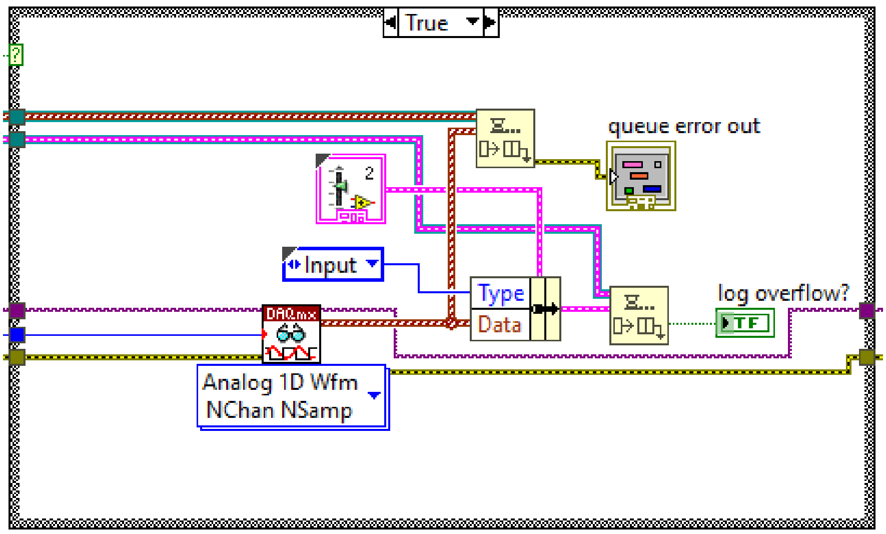

Figure 3.29: Read VI running code blocks

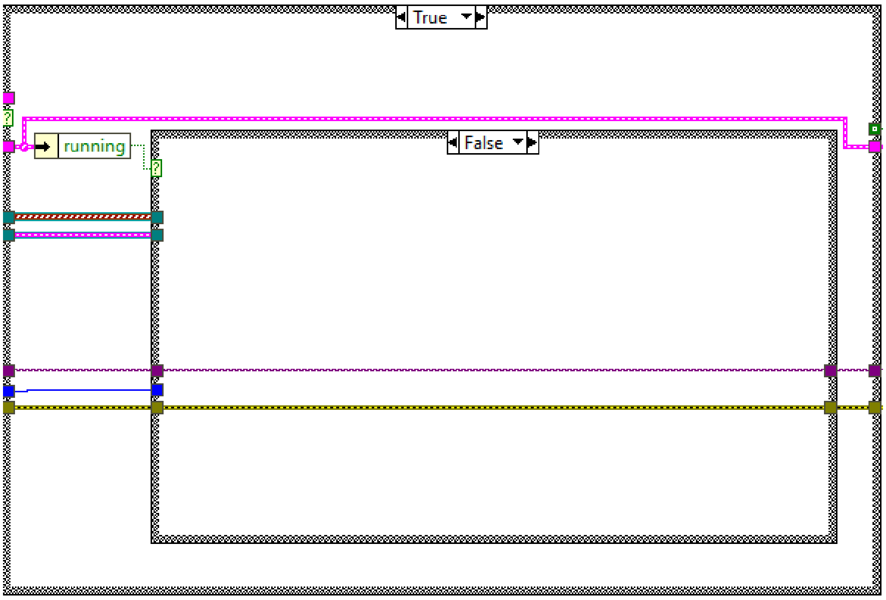

Figure 3.30: Read not Running state

via visual inspection. This would not be ideal as it should be clear which data is input data and which data is output data.

Finally the code which executes when no instructions have been received and the state is set to running is shown in Figure 3.30. As expected this involves no operations being performed as the program has been designed to wait for a message, perform actions based on that message then repeat this until the user has collected their results.

3.6.4 Writing Sub VI

The writing VI operates similar to the reading VI so similar components will not be discussed twice.

The operation of the Writing VI is substantially more complex than the reading VI. The initialisation stage is shown in Figure 3.31.

n the first stage similar operations are used to the reading VI however after that initial stage there is more configuration required. Initially the DAQmx task was configured to disable regeneration. This prevented the NI9023 from re writing the signal which it had received before to the output, instead opting to use the most recent signals which were passed in by the writing component. the next stage was also more complex as the buffer had to be filled with zeros to ensure there is data to write when the task is started. The writing task is initialised with a N channel N sample writing

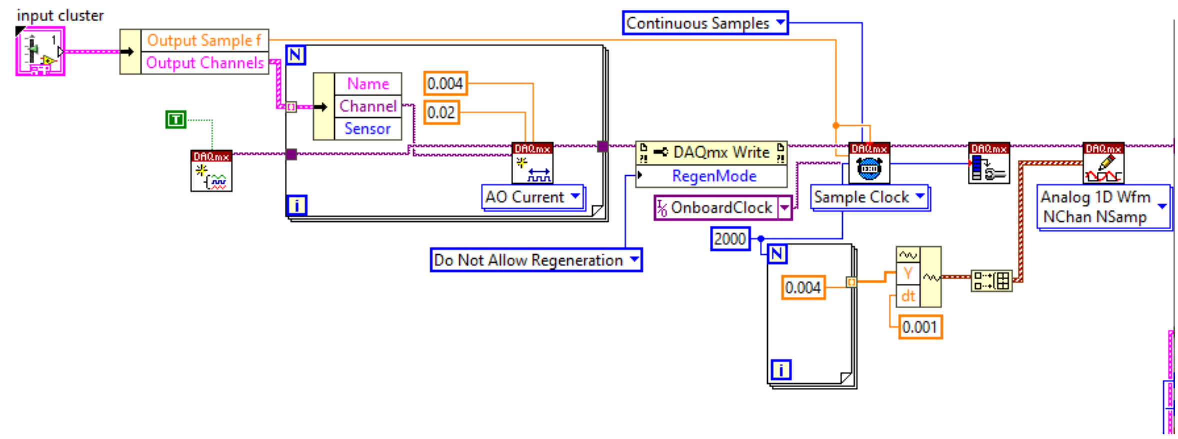

Figure 3.31: Overview of the write initialisation phase

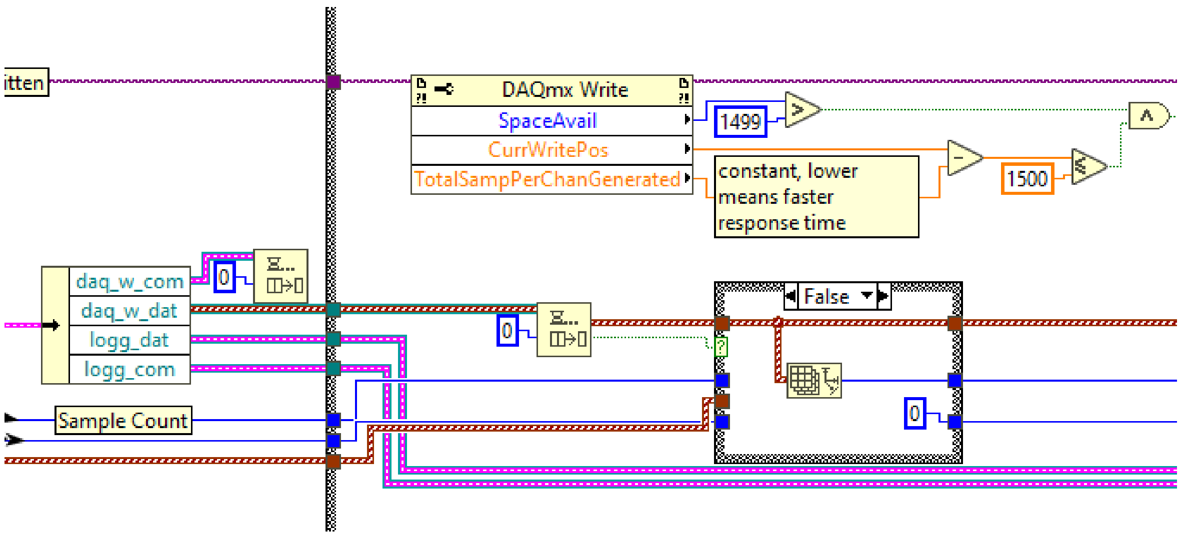

Figure 3.32: Write Checks

task to enable writing to multiple outputs. The reason for this was to enable easier synchronisation than previously possible, utilising the fact that the reading module can read multiple inputs in parallel the output can be fed to the input as well as the VFD to clearly show when the signal was sent and observed. The writing section initialises its buffer to be of length 2000 samples and setting those samples 0.

Figure 3.32 illustrates the logic behind the decision of which data to write to the system. The red wire on the left contains the waveform that are buffered but not yet written to the hardware. It is important to wait for the hardware to write the signals to ensure that the system stays responsive allowing for the user to make changes to the signal and see those changes reflected without waiting for the whole signal to be published. The if statement is used to check data has been received in the daq_w_dat

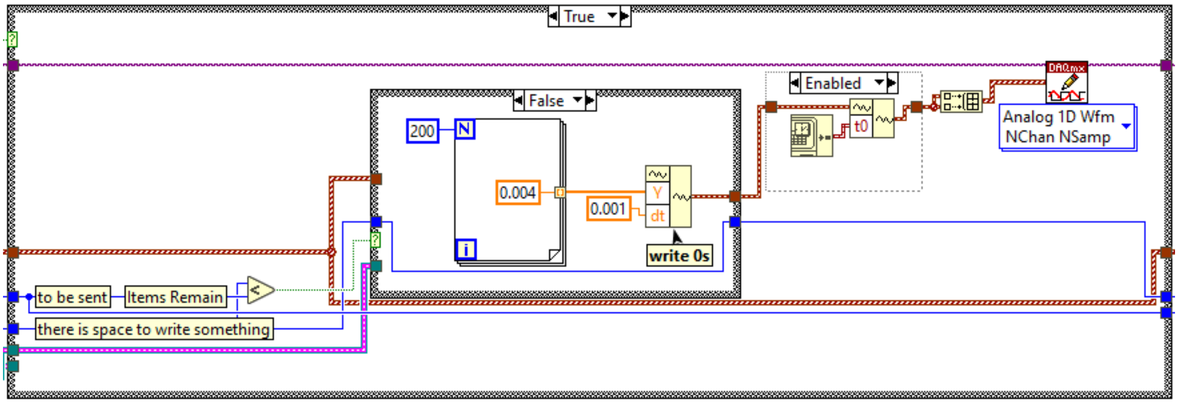

Figure 3.33: Write Tasks Should Write

queue. The if statement performs the function of storing the data passed via the queue locally and setting the sample count and counter variables so that the data can be written correctly.

The operation being performed at the top of the figure shown above is used to ensure the data stored in the buffer of the hardware does not leave the program unresponsive to user input. This is done by checking that there is enough space in the buffer and that the data written is not too far away from the current write position.. These two conditions are logically added and used to decide the operation of the next section of code.

Figure 3.33 shows the operations executed when there is no data to be written and the true case in the alternate case. The false case simply performs no-op so discussion will focus on the true case.

The true case is executed when there is space in the buffer to store more information, thus this case will result in more data being written to the hardware. The complexity in this case is the choice of which data to write to the hardware. In the case where the data which needs to be written (the data index is less than the count) the next piece of data will be retrieved then the index will be incremented. This is shown in Figure 3.34.

The alternate case is more complex than that shown in Figure 3.34. In this case the writing VI needs to generate a number of zeros to be written to the hardware. This is done using a for loop with 200 iterations. The number 200 was chosen arbitrarily however it should be noted that this number should be less than the size checked for in the checks figure, If this condition is not met the delay of the system between writes

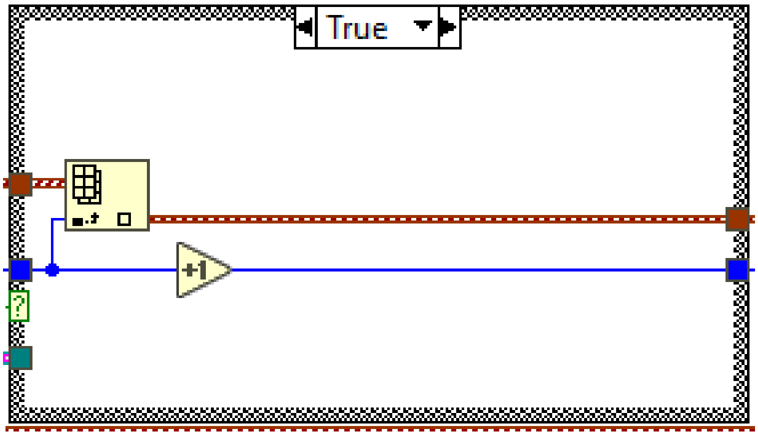

Figure 3.34: Adding a waveform to the writing sub VI

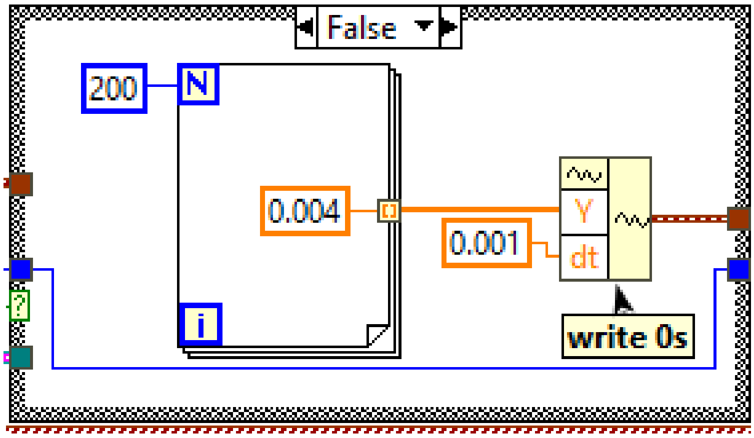

Figure 3.35: Generating zeros to indicate zero frequency

will be sluggish and unresponsive. The operations of the true case are shown in more detail in the figure below.

The code following the conditional statement discussed above simply writes the output signal to the hardware. After this the code loops to perform the same set of instructions until the program is ended.

3.6.5 Logging Sub VI

The logging VI receives data from the reading section and stores that information in a files specified by the user. The logging loop is also responsible for naming the output file appropriately and storing the output in the appropriate directory. The logging loop is shown in the image below.

The initialisation section of the logging loop is the simplest section and requires the least discussion in order to be understood. When started the logging VI creates

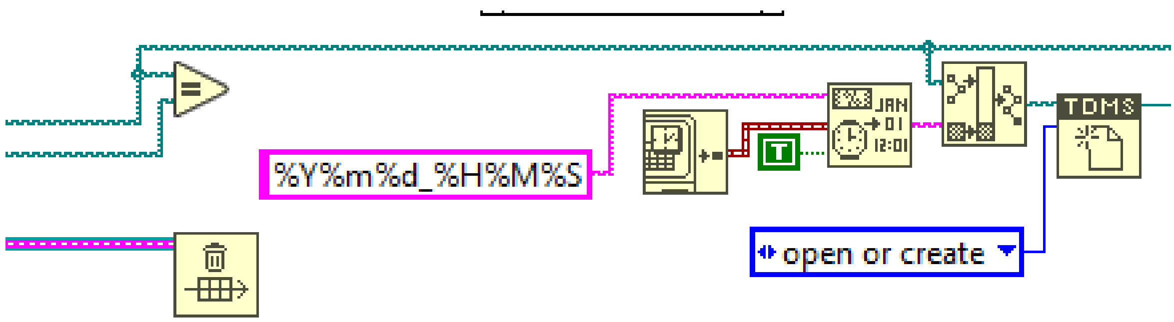

Figure 3.36: Overview of logging sub VI

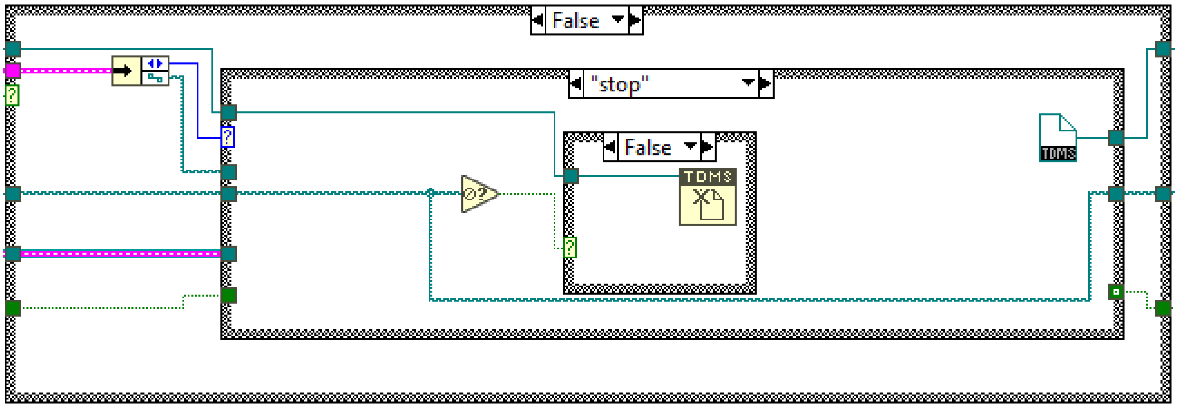

Figure 3.37: events listener within the logging sub VI

a new TDMS file reference with a default name using the TDMS Open and Generate Temporary File VIs. These are then passed into the logging loop. There are two operations performed within the logging loop, event response and logging. The event response operation is shown in the figure below.

Event response for the logging VI consists of handling events for 'stop' (Figure 3.37), and start. The stop operation simply checks if the file has already been closed and tries to close the file before creating a new reference to a null file for the program to keep working with.

The start handler is much more complex and involves picking a new name for the output file which is chosen. The user provides a path for the files to be stored. The program then uses this path and creates a new file with the current date and time as the name. The resulting file is then stored as a local variable.

The logging VI is also responsible for storage of data within the files. This is achieved with the other component. This component is shown in Figure 3.38.

The final part of the main Sub VI which requires discussion is the two graphs which

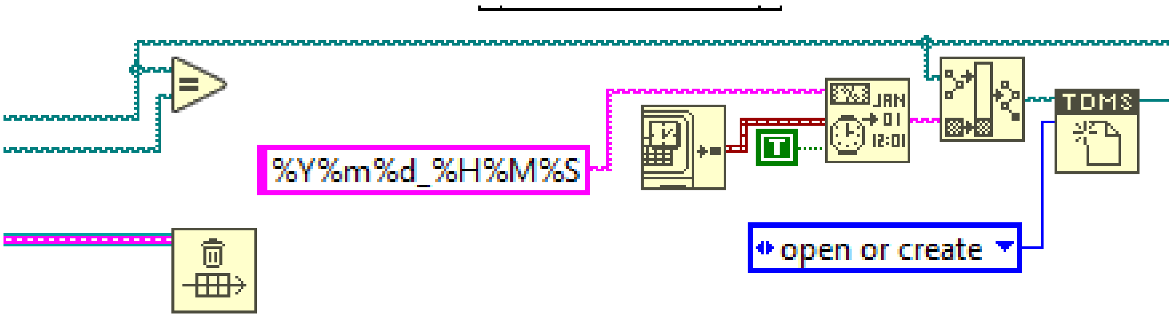

Figure 3.38: Screenshot of the logging implementation

are presented in the main UI. These graphs show the input signal and the output signal respectively. The input signal is shown on the left and will reflect the equation which is written into the equation editor box discussed earlier. The output on the right will show the recent readings taken from the system live as they are read into the system.

The main Sub VI operates on an event driven basis and after initial setup only responds to input events directly from the user. This requires much less memory than is required by other parts of the program.

In terms of optimisation as briefly discussed in Section 2 there are a number of techniques which will be used to aid in the development of the project. These include parallel programming, message passing via queues, class responsibilities and buffering.

3.6.6 Buffering

Due to the nature of the system which is being developed and the requirement of the software to allow long running experiments it is necessary to consider the case where an experiment may run for long periods of time. For this to take place buffering will be utilised. What follows is an explanation of how buffering works.

Due to the fact that devices do not have the capabilities to store infinite (or immeasurably large) sets of data at once it is necessary to compartmentalise the data so that it may be transmitted and stored in chunks. This is done through a process called buffering, the buffering which takes place can be done through a number of means.

- DMA buffering
- Message Buffering
- Hardware times writes.

Each of these methods fundamentally operate around the principle that either the data cannot all be stored on the device at once or the data which will be stored on the device cannot be sent all at once. It is important to consider both the limitation of the network and the storage when deciding this. The principle of buffering for this application applies due to the limitation of the storage which the device which is receiving has. This storage is physically limited but operates via a DMA FIFO queue.

The principle is that while the device is writing signals to the hardware from one section of the DMA buffer the software can write over the past packets which have already been written to the buffer. Generally this will not take place live as 1 buffer write is not an efficient use of space. Instead the software may wait for a certain number of packets to be written then proceed to overwrite that same number of packets to result in the same over all effect.

Buffering is implemented in the system of this experiment through a hybrid of DMA buffering and message buffering. via the use of queues data is sent through to a buffering component which is responsible for timing the writes which are sent to the hardware, This timing component monitors the space available in the device and writes a block when ever the device has enough space available to allow for a write to occur. When the device has enough space for the next block to be written the buffering controller will proceed to utilise DMA copying the data into the buffer of the hardware device.

This achieves a smooth output signal transmitted from the system to the fluid loop and allows for the system to return to an idle state in which no processing occurs.

3.6.7 Parallel Programming

The concept of parallel programming is not to difficult to understand. It consists of 2 or more parallel operations being performed at once resulting in a program completing its operations faster as it is essentially multitasking.

Parallel programming takes place in Lab View through the use of multiple while loops. Lab View code is inherently parallel without the use of loops this is hard to understand but with the implementation of loops it makes a lot more sense. When two while loops are placed inside a VI such that they are not contained within each other it becomes apparent that both will execute when the VI is executed. The VI will exit when both those while loops exit (assuming that they are the only operations in the VI).

In this system parallel programming is utilised to perform each of the essential operations without blocking the other operations. This consists of reading, writing and logging without falling behind and losing data. There are 3 processes which run in parallel and a fourth process which runs synchronously with the GUI to allow for the users input to be provided.

3.6.8 Message Passing

As mentioned above message parsing is utilised in this system to achieve communication between each of the asynchronous threads.

Queues are used to buffer the messages which are being sent between the various processes which are running in parallel. each process needs to either send or receive both data and commands through its buffer and to improve the efficiency of this process and reduce clutter this has been implemented via a unique queue for each of these data streams. This is to say each stream will have its own queue. The result of this is two buffer objects for each stream which is Incorporated.

As the queues implement buffers it is important to specify the size of each of these buffers. The messages which are stored in the queues are removed and added at different times thus there may be variation in the number of items which are stored in the queue. This variation in items may fluctuate or it may be such that data i being written much faster than it can be read. In this case the default behaviour of the queue in Lab View is to extend the length of the queue on infinitude until the computer fills its memory with unprocessed data. In order to deal with the issue of boundless buffers a maximum buffer size can be assigned to the buffer. This should be done for every occurrence of the buffers which are being used.

Chapter 4

Results & Validation

Write your chapter here.

- How did your solution work? (Simulations, field tests, graphs, schematic diagrams.)

Data for all experiments was captured in TDMS files as per the specifications defined 3.2. This data was then processed using a script written using the Python3 language to produce graphs and visualisations for the data collected. Some minor data processing and adjustments were performed in order to clean up the data before it could be visualised properly. This was done in the form of the

In order to understand the underlying frequencies imposed on the system discussion of the conversion from mA to L/m is required. The conversion from mA to L/m is due to the way the signal is passed through the system. Initially the signal is passed from the output module to the VFD at this stage the conversion is between $4 - 20mA$ to $0 - 50Hz$ the VFD then passes its output into the motor which produces its output in the form of $0 - 2880rpm$. Following this the output is taken from the motor and passed through a torque converter with a ratio of $2.35 - 1rpm$ this is then passed through to the pump which produces $0.0035L/m$ per rpm. This process is summarised in the equation below.

$$f(x) = (x-4) \cdot \frac{50}{16}\frac{Hz}{mA} \cdot \frac{2880}{50}\frac{rpm}{Hz} \cdot \frac{1}{2.35}\frac{rpm}{rpm} \cdot \frac{0.0035}{1}\frac{L/m}{rpm}$$

$$f(x) = \frac{63}{235} \cdot (x-4)(\frac{L/m}{mA}) \tag{4.1}$$

4.1 Initial Results

4.1.1 Connection Testing

A trial was run to verify that the signals could be read by the system. This test did not involve the use of any fluid equipment and simply consisted of connecting a 4..20mA sensor to each of the input ports attached to the chassis. To ensure the functionality of each of the ports which would later be used in experiments this test verified that the connections had been made correctly. This experiment is shown in Figure 4.1 and shows the times where a sensor has been connected and disconnected from the hardware. The results correlated perfectly with the output, When a sensor was attached it was expected to produce a 4 mA signal (representing 0 in the 4 to 20 mA spectrum) then when the component was disconnected the sensor detected an absolute 0 i.e. no current. This was found to work for Port 1, 2 and 4 however as can be seen from Figure 4.1 there was no output from Port 3 on the day of the test. This was taken into account and port 3 was not used for testing purposes.

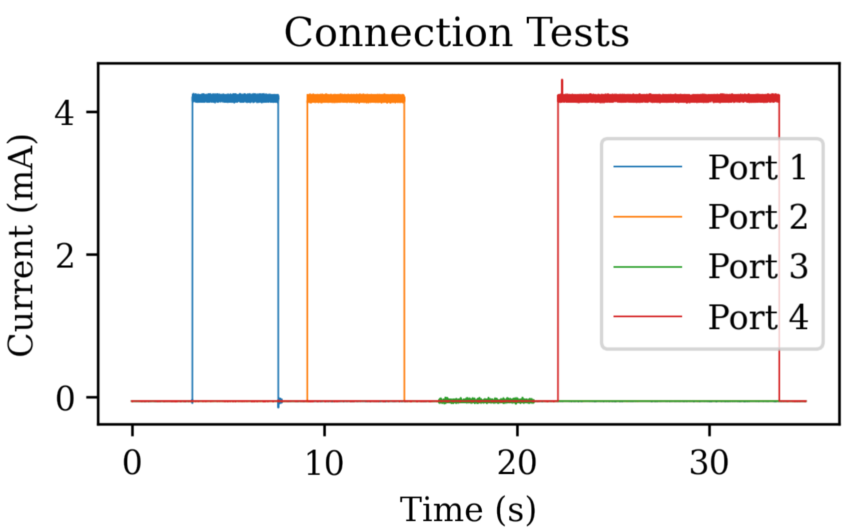

Figure 4.1: Input Testing

4.1.2 Step Response

The following test was performed by sequencing unit step functions to test the response of the system in order to better understand the rise and settling time of the fluid contained within and the speed of its response to response to the input which was

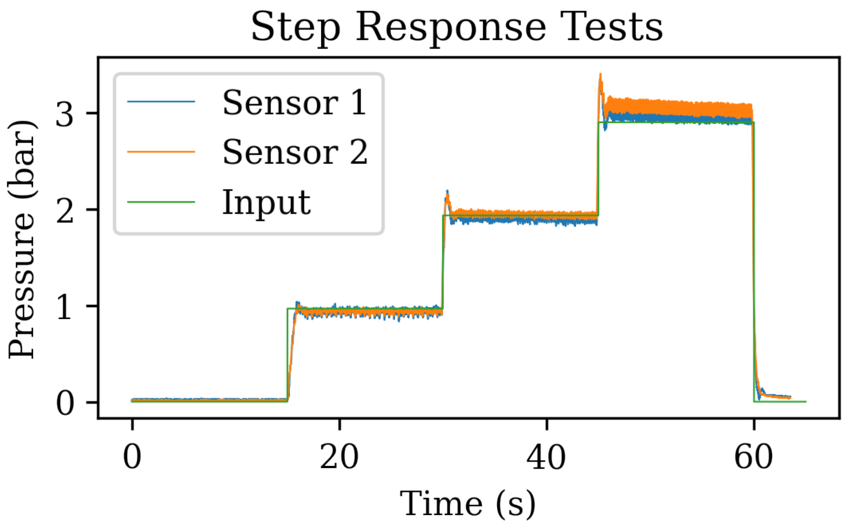

Figure 4.2: Step Response

provided. The results for this experiment are shown in Figure 4.2. The results were what was expected, with the system measuring a signal which shows direct correlation with the input signal. Features shown in Figure 4.2 include the overshoot which is seen after each of the steps after the first step. These results were as expected and can be understood by noting the underlying operations which are being performed on the fluids which are in the system. The result of the instantaneous increase in flow rate in the system, this causes the fluid to build up momentum which then causes the overshoot seen in Figure 4.2.

Although it is interesting to see the response of a non constant derivative another point of interest is the response to a signal with various other derivatives.

Figure 4.2 shows a zoomed version of Figure 4.2. This figure shows the small oscillations found when looking at the $1s$ period between $31s$ and $32s$. This period contains a small oscillation which shows a signal of $F = 4Hz$. This signal should correlate with the input signal provided to the motor from Equation 4.1. The resulting calculation when taking into account the use of all components is $9.21Hz$ as shown in Equation 4.2

$$I(q) = \frac{235}{63} \cdot q \cdot \frac{50Hz}{16mA}$$

$$I(q = 0.786) = 9.21Hz \tag{4.2}$$

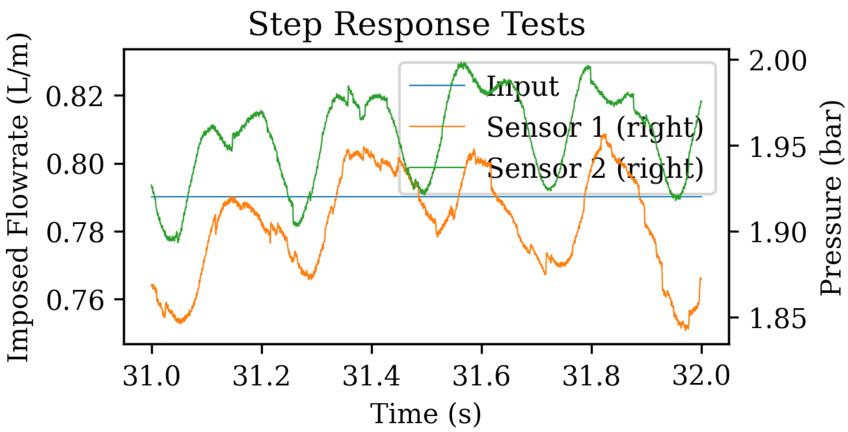

Figure 4.3: Step Response Zoomed

4.1.3 Sinusoidal Response

In order to test this further a system with a sinusoidal input was tested. In this case the signals was given a frequency of $f = 0.1Hz$. This test was successfully performed with the results shown in Figure 4.4. These results indicate the experiment was generally successful as they show the systems instantaneous response to the input provided. There are a number of interesting points to note in these results too. The flattening of the curve at the base of the sinuosity effectively represents the fluids inability to come to a complete stop. This is expected and is justified. There is also an interesting phenomenon present when the signal finishes being transmitted. This is a prolonged settling time at the completion of the transmission.

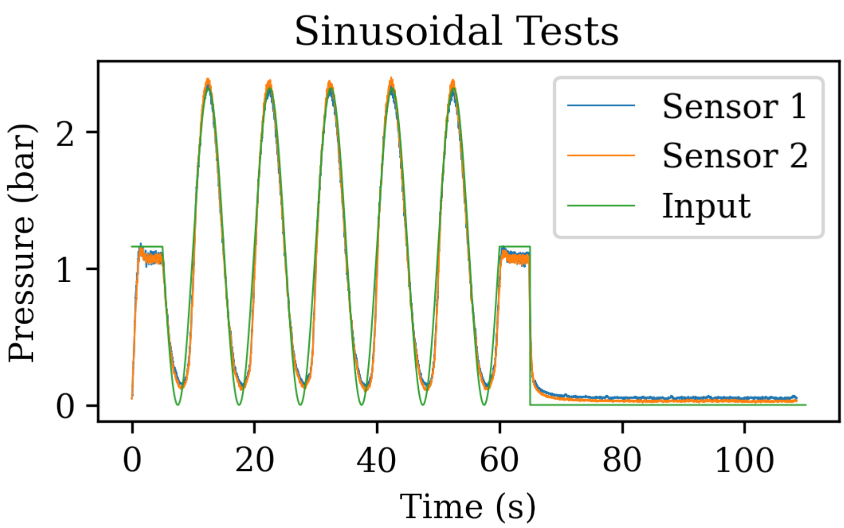

Figure 4.4: Sinusoidal Response

4.1.4 Final Analysis

The final stage of analysis to be performed on this data is to understand the correlation between the input signal and the output signal in a graphical view. This is presented in Figure 4.15 where the input signal is mapped to the x-axis while the signal read from the system is shown on the y axis.

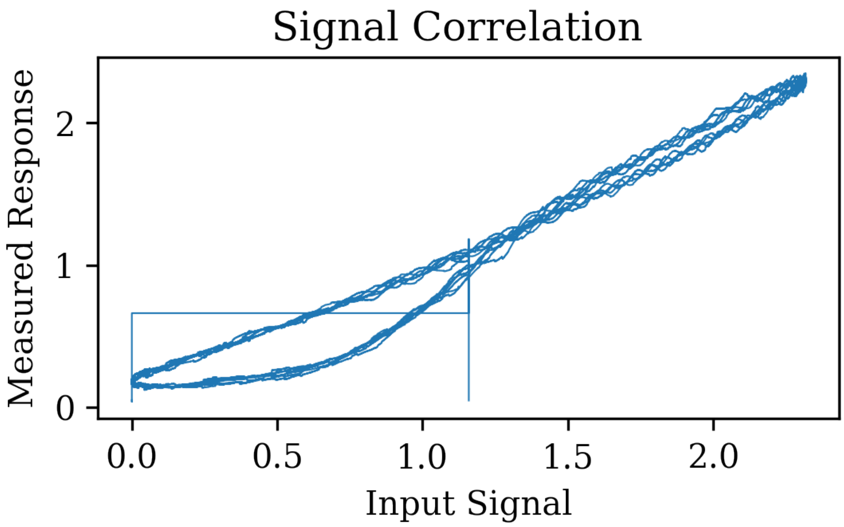

Figure 4.5: Correlation of sinusoidal signal

4.1.5 Further Testing

As the system is intended to be used for viscus fluid analysis tests were run with such fluid types. These tests consisted of a number of trials testing different properties of the fluids.

All tests consisted of a general mathematical functions. All tests involved the use of an input signal driven in the range from $4mA$ to $10mA$. The tests were run with a 16 bar pressure sensor with a sample rate of $1000s/s$ (samples per second).

The first set of experiments consisted of a number of linear interpolation trials. Each trial increased the constant coefficient representing the gradient of the line

$$y = mx + c$$

A number of gradients were used for this experiment. These gradients are shown in Table 4.1 and are approximately linear.

Duration (s)	**Gradient** $(\frac{mA}{s})$
12.5	0.48
6.5	0.92
2.5	2.40
2	3
1	6

Table 4.1: Gradients used in linear interpolation experiments

The first experiment is shown in Figure 4.6, this experiment showed interesting results compared with the initial experiments performed on water. While a direct comparison cannot be made with the unit step test conducted earlier it is clear that the fluid response is different to that which was observed in prior tests. It is worth noting that the oscillations seen above the signal is present in the water tests and thus should be ignored as they are not a feature of the fluid.

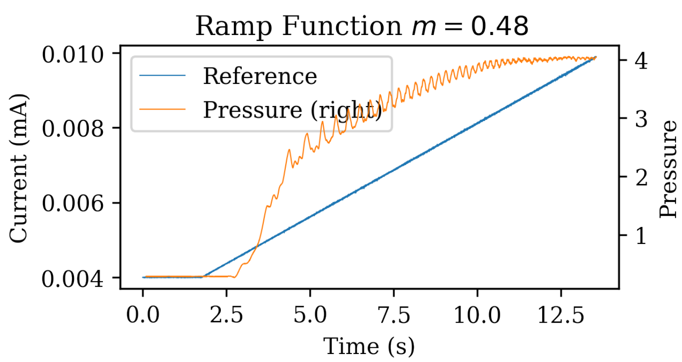

Figure 4.6: 12.5s ramp input

The test shown in Figure 4.7 is similar to that shown in Figure 4.6 however the duration has been reduced to half the time of the previous test. The results observed in this trial show a clear difference between the initial trials when compared to the trials using Citron fluid. There is a clear drop in the measured pressure in the system around the 7$^{\text{th}}$ second of operation.

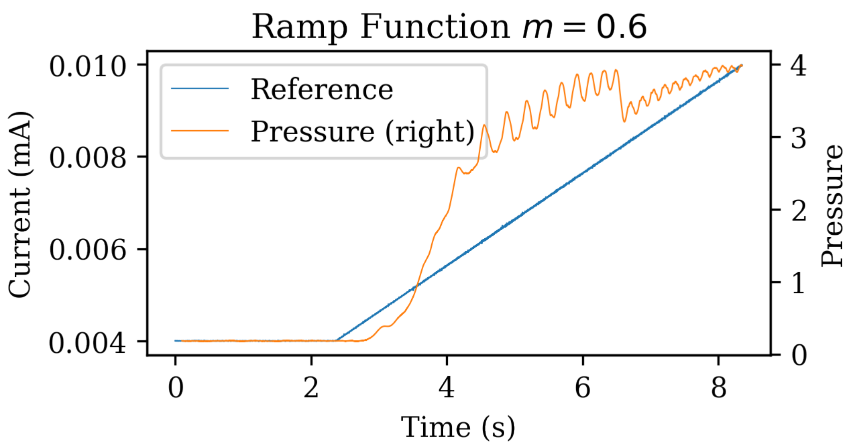

Figure 4.7: 6.5s ramp input

The results shown in Figures 4.8,4.9,4.10 all show similar behaviours to those shown in Figure 4.7 and thus further analysis is not required.

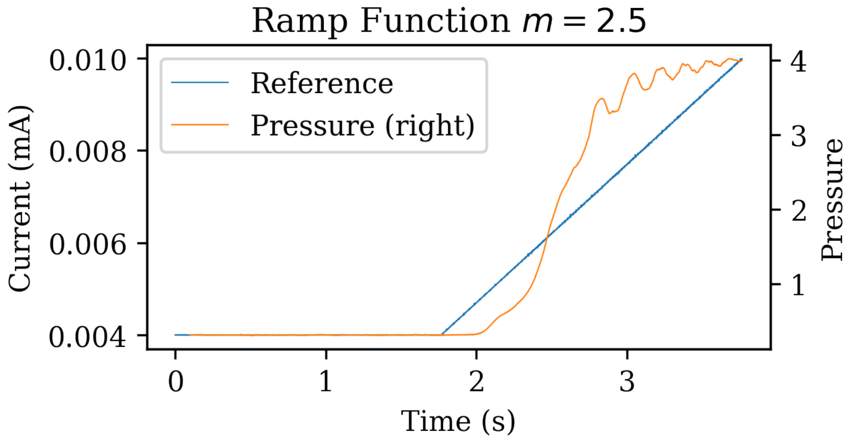

Figure 4.8: 2.5s ramp input

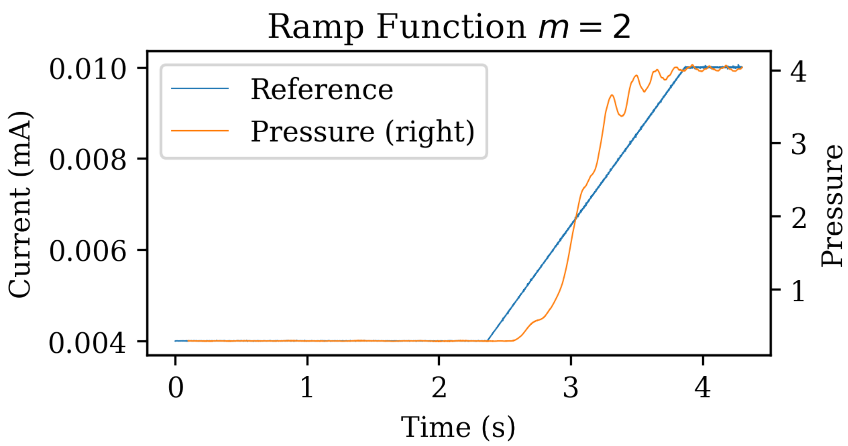

Figure 4.9: $2s$ ramp input

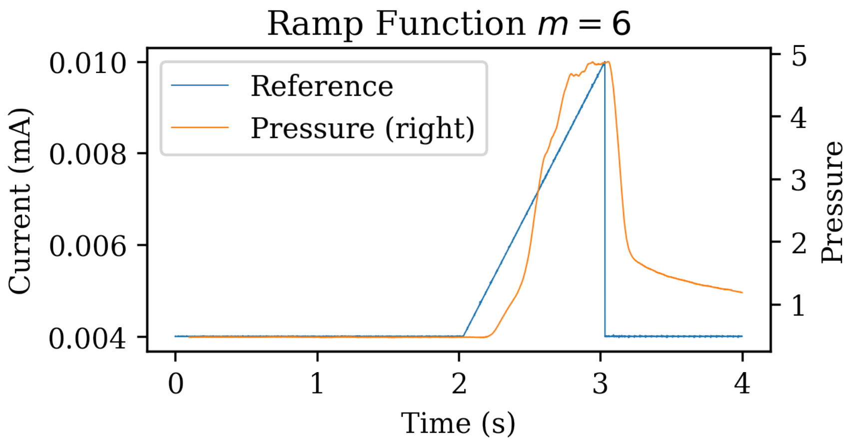

Figure 4.10: $1s$ ramp input

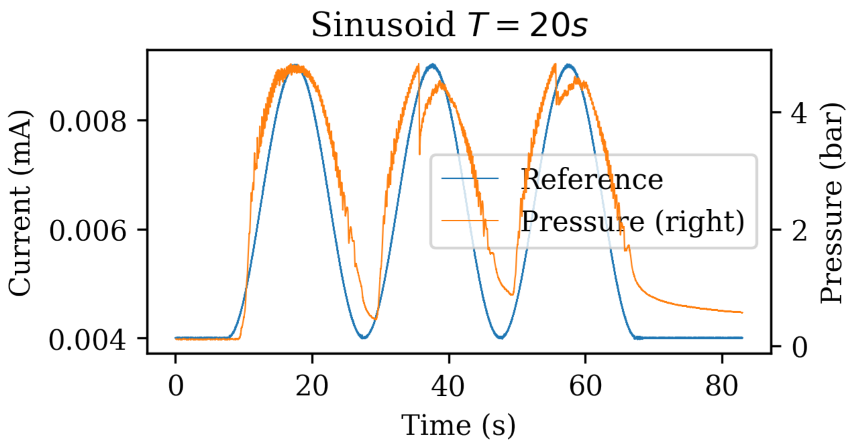

Figure 4.11: $T = 20s$ sinusoidal input

The chart shown in 4.11 shows the same behaviour as seen in Figure 4.7. This behaviour is very interesting. The behaviour can bee seen better in 4.12. This behaviours is not explainable without the use of additional knowledge of non-Newtonian fluids.

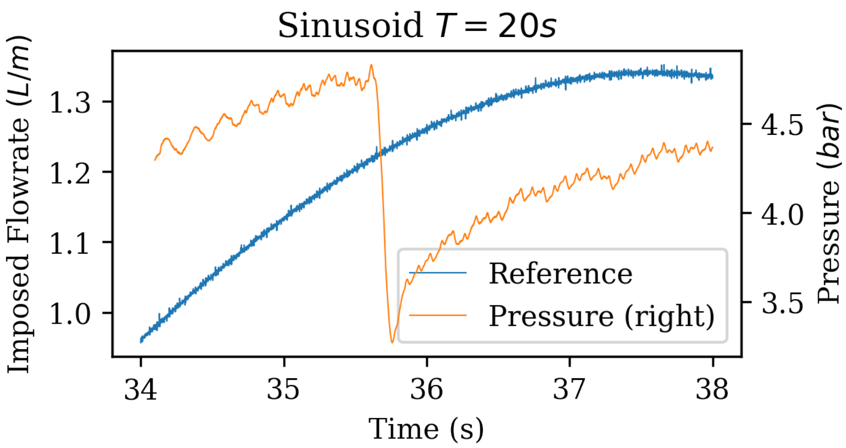

Figure 4.12: $T = 10s$ sinusoidal input

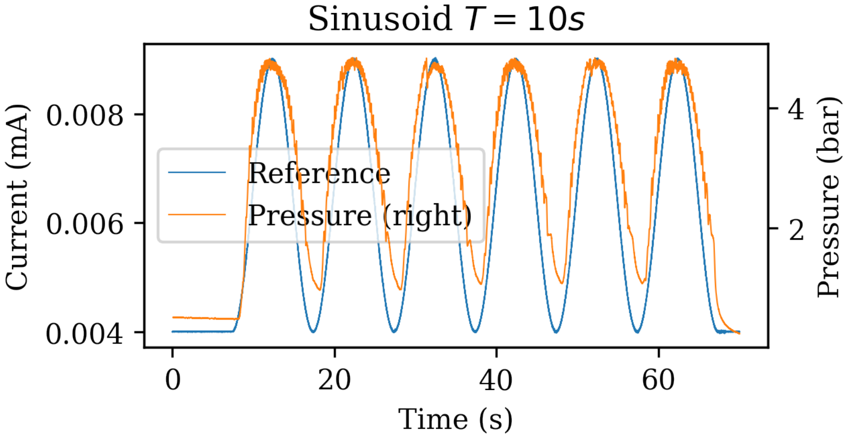

Figure 4.13: $T = 10s$ sinusoidal input

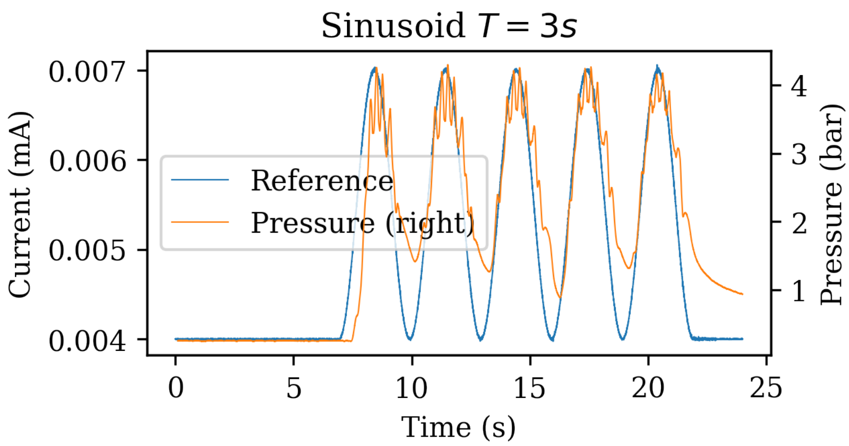

Figure 4.14: $T = 3s$ sinusoidal input

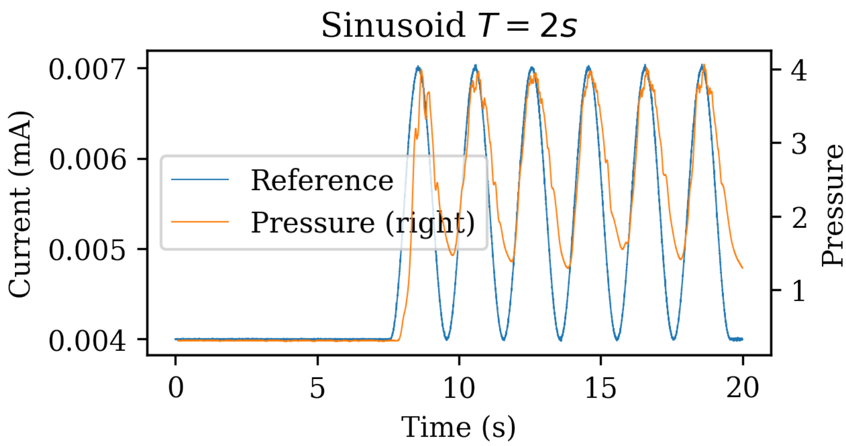

Figure 4.15: $T = 2s$ sinusoidal input

At this point experimentation was complete, advised by the petroleum team experimentation was stopped at $f = \frac{1}{2}Hz$ to prevent damage to the Bella system.

Chapter 5

Conclusion and Future work

During experimentation there were however some issue with experimentation as the results which were obtained did not conform with the results perfectly.

These errors occured mainly in the initial testing phase and were caused by unreliable measurements. The erronious measurement was found to be due to the timing of the recorded input with the output signal. This was rectified as discussed earlier. This experiment has shown that the system discussed is capable of controlling a fluid with the appropriate input parameters and recording the appropriate fluid properties for each moment of time the signal is applied. This system is useful as it enables researchers to get a better understanding of the way fluids may be used.

There are a number of benefits to the system whihc has been discussed in this paper some of which include the ability to provide a predefined signal as input. This allows for the user to record their results and reproduce experiments without the inclusion of unaccounted variables which cannot be observed through closed loop control. This system is much better than alternative systems at recording the variation from the expected system output when compared to the actual system output.

Other areas which may be explored and improved upon which the system developed here does not cover. These include the measurement of more complex fluid properties. The system discussed here did not invovle the development of complex analysis of the fluid properties which it was reading, instead it involved measuring quantities directly without feedback to the controller. Instead the system accepts a predefined input signal and sends this to the output. Future additions to this could include the option to add a form of feedback control to the signal which is beign produced. This would in effect allow for the impementation of PID control. Another useful feature which could have been implemented in the system and may improve measurements would be a kalman filter. In this case a kalman filter would have ensured that the signal being read in was not disturbed by the noise of the measurement and thus would not require filtering after the measruement had occured. This could improve the understanding which is gained from analysis. Kalman filters can also produce much more accurate results than the measurement results which were actually used. In addition to the processing which could have been implemented there could also have been additional testing to highlight issues with the system. This could include the operation of the sinusoidal signal at lower frequencies and higher periods in order to identify the drops in pressure in the system.

THe learning process undertaken during writing and experimentation covered in this paper was long and extensive. The results which were produced and the system whihc has been developed is increadibly useful for further research and the system also offers for the oppertunity to be modified in order to add further improvements.

Appendices

Appendix A

Google Drive

Links to the google drive folder containing datasheets, results and the experimental lab-view program are given here https://drive.google.com/drive/folders/1cPpGuZiWCBdyT87IMBIF
_q1pBW77H57C?usp=sharing

References

Alphonsus, E. R., & Abdullah, M. O. (2016). A review on the applications of programmable logic controllers (plcs). *Renewable and Sustainable Energy Reviews*, *60*, 1185-1205. Retrieved from https://www.sciencedirect.com/science/article/pii/S1364032116000551 doi: https://doi.org/10.1016/j.rser.2016.01.025

Balavalad, K. B., & Sheeparamatti, B. G. (Apr 2015, Apr). A critical review of mems capacitive pressure sensors. *Sensors & Transducers*, *187*(4), 120-128. Retrieved from https://link.library.curtin.edu.au/gw?url=https://www.proquest.com/scholarly-journals/critical-review-mems-capacitive-pressure-sensors/docview/1683570790/se-2?accountid=10382 (Copyright - Copyright IFSA Publishing, S.L. Apr 2015)

Berzborn, M., Bomhardt, R., Klein, J., Richter, J.-G., & Vorländer, M. (2017). The ita-toolbox: An open source matlab toolbox for acoustic measurements and signal processing. In *43th annual german congress on acoustics, kiel (germany)* (Vol. 6, pp. 222–225).

Beverly, C., & Tanner, R. (1992). Numerical analysis of three-dimensional bingham plastic flow. *Journal of Non-Newtonian Fluid Mechanics*, *42*(1), 85-115. Retrieved from https://www.sciencedirect.com/science/article/pii/037702579280006J doi: https://doi.org/10.1016/0377-0257(92)80006-J

Biezl. (2008). *Thermische massendurchflussmessung en.* Retrieved May, 06 2021, from https://commons.wikimedia.org/wiki/File:Thermische_massendurchflussmessung_en.svg

Cha, J.-E., Ahn, Y.-C., & Kim, M.-H. (2002). Flow measurement with an electromagnetic flowmeter in two-phase bubbly and slug flow regimes. *Flow Measurement and Instrumentation*, *12*(5), 329-339. Retrieved from https://

www.sciencedirect.com/science/article/pii/S0955598602000079 doi: https://doi.org/10.1016/S0955-5986(02)00007-9

Cobb, J. (1996). Control in the field with hart® communications. *ISA Transactions*, *35*(2), 165-168. Retrieved from https://www.sciencedirect.com/science/article/pii/0019057896000201 doi: https://doi.org/10.1016/0019-0578(96)00020-1

Dietrich, P., Masiero, B., Pollow, M., Scharrer, R., & Muller-Trapet, M. (2010). Matlab toolbox for the comprehension of acoustic measurement and signal processing. *Fortschritte der Akustik–DAGA, Berlin*.

Ding, Y., Sun, B., Huang, Y., Lin, W., & Qian, F. (2010). Hybrid matlab and labview to implement an intelligent foundation fieldbus control system. In *2010 8th world congress on intelligent control and automation* (p. 4539-4543). doi: 10.1109/WCICA.2010.5554100

Endress+Hauser. (2009a). *The differential pressure flow measuring principle (orifice-nozzle-venturi).* Retrieved May, 6 2021, from https://www.youtube.com/watch?v=oUd4WxjoHKY

Endress+Hauser. (2009b). *The ultrasonic flow measuring principle.* Retrieved May, 6 2021, from https://www.youtube.com/watch?v=Bx2RnrfLkQg

Firth, J., Ladouceur, F., Brodzeli, Z., Bruin, C., Wang, H., & Silvestri, L. (2017). Liquid crystal based optical telemetry applied to 4–20ma current loop networks. *Sensors and Actuators A: Physical*, *260*, 124-130. Retrieved from https://www.sciencedirect.com/science/article/pii/S0924424717305824 doi: https://doi.org/10.1016/j.sna.2017.04.005

Fp5000 series pressure temperature transmitter (Computer software manual No. 08927). (2011). Retrieved from https://au.mouser.com/new/honeywell/honeywell-fp5000-pressure-transducers/ (Issue. 2)

Heinisch, M., Voglhuber-Brunnmaier, T., Reichel, E., Dufour, I., & Jakoby, B. (2014). Reduced order models for resonant viscosity and mass density sensors. *Sensors and Actuators A: Physical*, *220*, 76-84. Retrieved from https://www.sciencedirect.com/science/article/pii/S0924424714004014 doi: https://doi.org/10.1016/j.sna.2014.09.006

Hemphill, T., Campos, W., & Pilehvari, A. (1993, 8). Yield-power law model more accurately predicts mud rheology. *Oil and Gas Journal; (United States)*, *91*(0). Retrieved from https://www.osti.gov/biblio/5452894

Howarth, M. (1992). You've got to have hart. *Assembly automation*.

Javed, Y., Mansoor, M., & Shah, S. (2019, 07). A review of principles of mems pressure sensing with its aerospace applications. *Sensor Review*. doi: 10.1108/SR-06-2018-0135]

Kirkman, I. W., & Buksh, P. A. (1992). Data acquisition and control using national instruments' "labview" software. *Review of Scientific Instruments*, *63*(1), 869-872. Retrieved from https://doi.org/10.1063/1.1142631 doi: 10.1063/1.1142631

Kostic, M. (2013). 46.1 shear viscosity 46 fluid viscosity measurement dimensions and units of viscosity rheometer @bullet rotational methods/concentric cylinders @bullet cone-and-plate viscometers @bullet parallel disks falling sphere @bullet falling cylinder @bullet falling methods in opaque liquids @bullet rising b..

Liang, Q., & Li, L. (2011). The study of soft plc running system. *Procedia Engineering*, *15*, 1234-1238. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877705811017292 (CEIS 2011) doi: https://doi.org/10.1016/j.proeng.2011.08.228

Mrq22 pressure temperature transmitter (Computer software manual No. MRQ22). (2011). Retrieved from https://www.measurex.com.au/au/mrq22.html (Ver. 5.3-17/05)

Optidrive g200 (Computer software manual No. 145506-01). (2019). Retrieved from https://isource.invertekdrives.com/client-uploads/download-manager/brochures/85-ODP2B-IN%20V3.22%20Optidrive%20P2%20Brochure.pdf

Perraudin, N., Paratte, J., Shuman, D., Martin, L., Kalofolias, V., Vandergheynst, P., & Hammond, D. K. (2014). Gspbox: A toolbox for signal processing on graphs. *arXiv preprint arXiv:1408.5781*.

Pt5415 pressure transmitter (Computer software manual No. PT-040-SEG14-A-ZVG/US/ /W). (2015, 09). Retrieved from https://www.ifm.com/au/en/product/PT5415

Pt5443 pressure transmitter (Computer software manual No. PT5443). (2011). Retrieved from https://www.ifm.com/au/en/product/PT5443

Rehm, B., Haghshenas, A., Paknejad, A., Al-Yami, A., Hughes, J., & Schubert, J. (Eds.). (2012). Chapter 2 - flow drilling: Underbalance drilling with liquid single-phase systems. , 39-108. Retrieved from https://www.sciencedirect.com/science/article/pii/B9781933762050500097 doi: https://doi.org/10.1016/B978-1-933762-05-0.50009-7

Rendon-Nava, A. E., Díaz-Méndez, J. A., Nino-de Rivera, L., Calleja-Arriaga, W., Gil-Carrasco, F., & Díaz-Alonso, D. (2014). Study of the effect of distance and misalignment between magnetically coupled coils for wireless power transfer in intraocular pressure measurement. *The Scientific World Journal*, *2014*. doi: 10.1155/2014/692434

Rullán, A. (1997). Programmable logic controllers versus personal computers for process control. *Computers and Industrial Engineering*, *33*(1), 421-424. Retrieved from https://www.sciencedirect.com/science/article/pii/S0360835297001277 (Proceedings of the 21st International Conference on Computers and Industrial Engineering) doi: https://doi.org/10.1016/S0360-8352(97)00127-7

Sm2000 magnetic inductive flow meter (Computer software manual No. SMR21XGXFRKG/US). (2021). Retrieved from https://www.ifm.com/au/en/product/SM2000

Sm2004 magnetic inductive flow meter (Computer software manual No. SMR21XGX50KG/US). (2016). Retrieved from https://www.ifm.com/au/en/product/SM2004

Takeda, Y. (1995). Velocity profile measurement by ultrasonic doppler method. *Experimental Thermal and Fluid Science*, *10*(4), 444-453. Retrieved from https://www.sciencedirect.com/science/article/pii/089417779400124Q (Experimental methods in Thermal and Fluid Science) doi: https://doi.org/10.1016/0894-1777(94)00124-Q

Tekdrive (Computer software manual No. 145506-01). (2019). Retrieved from https://www.aimsindustrial.com.au/assets/files/TECDrive-AC-Variable-Speed-Drive-Brochure.pdf

Urbikain, G., & López de Lacalle, L. (2020). Monithor: A complete monitoring tool for machining data acquisition based on fpga programming. *SoftwareX*, *11*, 100387. Retrieved from https://www.sciencedirect.com/science/article/pii/S2352711018300645 doi: https://doi.org/10.1016/j.softx.2019.100387

VanZeller, C., Williams, A., & Pollock, I. (2019, Apr 08). Comparison of bench test results measuring the accuracy of peak flow meters. *BMC Pulmonary Medicine*, *19*(1), 74. Retrieved from https://doi.org/10.1186/s12890-019-0837-3 doi: 10.1186/s12890-019-0837-3