

Master's thesis

Jonas Peter Hyatt

Experimental AC Susceptometry

Developing a control system for AC susceptometry measurements

Advisor: Kim Lefmann

Handed in: November 12, 2021

Abstract

Magnetic materials impact us every day, from the rare-earth magnets that are part of electric engines powering trains, to high temperature superconductors used for low-loss filtering at telecommunications base stations. Common for all cases is a need for understanding before implementation. The need for understanding has driven science to explore materials in more extreme environments such as under large magnetic fields, or at very low temperatures, where we can make novel discoveries.

This thesis extends an AC susceptometer, which can reach these extreme sample conditions, developed for use in research at the University of Copenhagen. The theory behind susceptibility, susceptometry, and cryogenic refrigeration is briefly explained. A new software package for conducting susceptometry experiments is then presented, along with necessary hardware changes. The setup and operation of the susceptometry laboratory is explained in detail in a lab manual which covers both manual physical processes, as well as the integration of the susceptometry software package.

Experiments showed the software and susceptometer worked, but unfortunately also showed issues with the dilution refrigerator that ensures a cold sample environment for the susceptometer, specifically a leaking valve connecting the still and the inner vacuum chamber, preventing further progress with experimental work.

Resumé

Magnetiske materialer påvirker os hver eneste dag, fra de magneter af sjældne jordarter der udgør del af elmotoren der driver toget fremad, til høj temperaturs superledere som bliver brugt til filtrering med lavt tab af telekommunikation udbydere. Fælles for dem er et behov for forståelse før implementering. Behovet for forståelse har skubbet videnskaben til at udforske materialer under mere ekstreme miljøer, så som ved meget lave temperaturer, eller i store magnetfelter, det er her vi kan opnå ny forståelse.

Dette speciale bygger videre på et AC susceptometer, som kan opnå de ekstreme prøve miljøer vi søger, der er blevet udviklet til forskning på Københavns Universitet. Teorien bag susceptibilitet, susceptometri, og kryo-teknisk køling bliver kort beskrevet. En ny software pakke til at gennemføre susceptometri eksperimenter bliver præsenteret, sammen med nødvendige hardware ændringer. Opsætningen og operationen af laboratoriet bliver nøje gennemgået i en laboratorie manual der dækker både de fysiske processer og integrationen til susceptometri software pakken.

Eksperimenter viste at software og ac-susceptometer fungerede, men det viste desværre også problemer med blandings kryostaten, der sikrer det kolde prøvemiljø til susceptometeret. Konkret har vi en utæt ventil ind til det indre vakuumkammer, dette stoppede yderligere eksperimentelt arbejde.

Preface and acknowledgements

This is the thesis that concludes my Master of Science degree in Quantum Physics at the University of Copenhagen. The thesis has been written in the X-Ray and Neutron Science group of the Niels Bohr Institute under the guidance of Kim Lefmann.

The project builds on the works of many previous experimenters, most recently Mathias Mikkelsen [5] and Celeste Qvotrup [11]. For a more complete overview of the history of the project see[5].

I took on the project at the end of 2019, and I immediately reached out to Mathias Mikkelsen who gave me an informal introduction to the lab and has helped me several times throughout the project, for that I am very grateful, additionally Claus Birger Sørensen has helped me by ensuring that all the equipment of the lab was functional, and they both answered my questions very patiently. I would also like to thank Thomas Sand Jespersen for helping me understand the dilution refrigerator and the associated control board. I would like to thank Finn Berg Rasmussen who helped identify a critical defect in one of the valves of the dilution refrigerator.

Finally I would like to thank the X-Ray and Neutron science group for company and help during my thesis.

Contents

1	Introduction	1
2	Background	2
2.1	Magnetism	2
2.1.1	Susceptibility	2
2.1.2	Atomic theory of magnetism	5
2.1.3	Susceptibility of a simple system	6
2.1.4	Damped harmonic oscillator	8
2.2	Susceptometry	9
2.2.1	Determining the susceptibility from measurements	10
2.3	Dilution refrigeration	11
2.3.1	The working principle behind dilution refrigeration	11
3	Software developed	14
3.1	Overview of software developed	14
3.2	Socket based communication	14
3.3	Susceptometry control panel	16
3.4	Instrument drivers	16
3.4.1	Picowatt AVS-47B Resistance bridge	16
3.4.2	Keysight Technologies 2700 Digital Multimeter	17
3.4.3	Keysight Technologies N9310A Signal Generator	18
3.4.4	Stanford Research Systems SR830 Lock-in amplifier	18
3.4.5	Leiden Cryogenics MCK50-100 Dilution Refrigerator	18
3.4.6	Leiden Cryogenics Triple current source	18
3.4.7	Pfeiffer Vacuum MaxiGauge Pressure controller	18
3.4.8	Cryogenics Limited 16T Magnet controller	18
3.4.9	Agilent Analog Discovery 2	19
3.4.10	Tektronix TBS1072B	19
3.5	Integrating the driver into the stations	19
3.6	Socket clients	19
3.6.1	Collecting data using the socket client (process_next_step)	19
3.7	The webserver	20
3.7.1	The HTTP server	20
3.7.2	The socket server	20
3.8	Extra software developed during the project	21
4	Hardware	22
4.1	Automated coil winder	22
4.2	New cryogenics PC	24
4.3	New oscilloscope	24

5	Lab manual	25
5.1	Experimental setup	25
5.1.1	Magnetism station	25
5.1.2	Cryogenics station	25
5.1.3	Data management station	25
5.2	Winding a set of coils	31
5.2.1	Determining the optimal number of turns	31
5.2.2	Programming the number of turns	33
5.2.3	Running the coil winder	33
5.3	Preparing a sample	34
5.4	Using the cryostat	35
5.4.1	Understanding the controls	35
5.4.2	Preparing the cryostat	38
5.4.3	Cooling to approximately 4K	43
5.4.4	Cooling below 4K	45
5.4.5	Reheating the system	48
5.5	Running the software	49
5.5.1	Starting the system	49
5.5.2	The status page	49
5.5.3	The experiment configuration page	52
5.5.4	The cryogenics configuration page	52
5.5.5	Extracting data and running analysis	54
6	Experiments	56
6.1	The first cooldown	56
6.2	Subsequent experiments	56
6.3	Finding the leaking valve	59
7	Conclusion and outlook	62
7.1	Future work	62
	Bibliography	63
A	An overview of the control system	64
A.1	Controlling instruments using Python	64
A.1.1	Connecting the instrument to the PC	64
A.1.2	Connecting the instrument to Python	65
A.1.3	Determining which address corresponds to which instrument	65
A.1.4	Finding appropriate drives	65
A.2	Taking measurements	66
A.2.1	Adding the instrument to the station	66
A.2.2	Calling the instrument from the socket clients	66
A.2.3	Process next step	67
A.3	Background jobs	68
A.4	Using the lab without the complete susceptometry setup	69
A.5	Common issues	69
A.5.1	Wrong IP-Address	69
A.5.2	Wrong instrument addresses	69

List of Figures

2.1	Illustration of the use of AC susceptibility to probe the gradient of the magnetization by applying a large DC field and an AC probing field simultaneously. Here the DC field is illustrated by the blue dotted line, the AC field is illustrated by the black sinusoidal line, the magnetization as a function of external field is illustrated by the black curve, and the gradient of the magnetization is illustrated by the red dotted line. Figure source: [12]	4
2.2	Example of the complex AC susceptibility split into χ' and χ'' according to equation (2.11) using the damped harmonic oscillator as the example system. We see three separate phases, first the isothermal regime where ω is much smaller than the systems characteristic time, next we have the intermediate regime where ω is on the order of the systems characteristic time, and finally we have the adiabatic regime where ω is much larger than the characteristic time.	4
2.3	3d models of the susceptometer, produced using OpenSCAD.	9
2.4	A simplified overview of the principle behind dilution refrigeration. The blue gradient indicates the relative concentration of 3He to 4He, it is exaggerated for illustrative purposes. In the figure we see the helium mix settle out as two separate phases (due to the difference in mass), the phases mix spontaneously in the mixing chamber (due to the Van der Waals force) leading to increased entropy inside the mix, and a decreased entropy to anything thermally coupled to the mix. The 3He moves through the mixing phase until it reaches the still, where 3He is selectively removed using a heater (due to a difference in boiling points).	11
2.5	A more realistic illustration of an implementation of dilution refrigeration. Liquid 3He enters from the 1.5K condenser line and flows down to the mixing chamber exchanging heat underway. In the mixing chamber it replenishes the lighter 3He phase, and can possibly dissolve into the dilute 3He-4He phase. The dilute phase is connected to the still where the 3He is distilled off to recirculate in the loop. The percentages given er concentrations of 3He. Figure from Pobell[9].	12
3.1	Sketch to illustrate how the different experiment stations are connected, and which instruments are connected to each. The cog illustrations are meant to illustrate a generic instrument, while the lines into the instrument shows what type of data connection is currently in place.	15
3.2	Calibration data extracted from Leiden Cryogenics LabView modules fitted to a cubic spline interpolation. The calibration shows the effective range of the DALE1500 is approximately 1-1000 millikelvin while the 10k RuO_2 has an effective range of approximately 10-2000 millikelvin.	17
4.1	The coil winder before and after modification.	23
4.2	Fabrication process of the automated coil winder.	23
5.1	A complete overview of how the instruments and sensors are interconnected, the connections are a mixture of analog and digital (see figure 3.1 for more details on the connection types). . . .	26
5.2	The Cryogenic Limited magnet power supply controls the superconducting magnet made from Nb_3Sn and NiTi. The magnet should be kept under 5K when it is in use. The power supple can be controlled using the panel on the front, or via a serial connection to the magnetism station. . .	27

5.3 The Agilent N9310A function generator is used to control the magnetic field created in the excitation coil, it can be controlled via USB connection to the magnetism station. In the current experimental setup we only use the low frequency (LF) mode. 27

5.4 The Tektronix TBS 1072B Digital Oscilloscope is used to measure the current flowing through the excitation coil, this is done by measuring the voltage drop over a well known resistor (inside the blue box displayed on the image), it is controlled through a USB connection to the magnetism station. 28

5.5 The Stanford Research Systems SR830 DSP Lock-in amplifier is used to measure the complex (ac) voltage induced in the measurement coils, it is controlled through a GPIB interface to the magnetism station. 28

5.6 Schematic overview of the wired connections of the magnetism station. The LOCKINAMP component represents the Stanford Research Systems SR830 Lock-in amplifier, where the ext. trig. represents the reference signal, and the A input represents the lock-in amplifier input. The SIGNALGENERATOR component represents the Keysight N9310A Signal Generator where the output (+ amd COM) represents the low frequency output of the N9310A. The MULTIMETER component represents the Tektronix TBS1072B oscilloscope where the input is channel 1 on the oscilloscope. The resistor R1 is the total resistance in the circuit before the coil, and the mutual inductor T1 is the combined excitation and measurement coils. 29

5.7 The Pfeiffer Vacuum MaxiGauge controls the vacuum gauges mounted inside the dilution refrigerator. It is connected through a serial port to the cryogenics station. It allows us to measure the IVC and Still pressure in the interval $< 5 \cdot 10^{-4}$ to approximately 800 millibars, the sensitivity around 1000 millibars is very high, which is why it shows > 1500 millibars. 29


5.8 The Leiden Cryogenics Triple Current Source controls the heating of a set of resistors mounted inside the dilution refrigerator to enable efficient cooling, currently only one resistor is connected (the resistor in the still, which is the most important one). It is connected to the cryogenics station through a serial port. 30

5.9 The Picowatt AVS-47B Resistance bridge allows us to monitor the temperature of a set of resistors mounted inside the dilution refrigerator by using a very small amount of power. It is controlled by the cryogenics station using a serial port. 30

5.10 The Keithley 2700 Digital Multimeter is used to probe a set of 9 thermocouples placed throughout the superconducting magnet and pulse tube compressor, the data allows us to keep track of the cooling process. It is connected to the cryogenics station using a serial port. 30

5.11 A image of the coils mounted to the dilution insert with dc lines attached to measure a signal from a hex key used as a control sample. 34

5.12 A image of the dilution refrigerator insert with important components labeled. Source: Leiden Cryogenics[2]. 35

5.13 An overview of the dilution refrigerator frontpanel. This graphical representation differs from the physical presentation as I have excluded lines, valves and sensors which are not connected, and I have included the actual position of the pressure sensor P2 as well as the readout. This representation should be more accurate than the representation seen on the physical panel (which can be seen on figure 5.14). The yellow circuit is where atmospheric gasses circulate, while the green circuit is where the helium mix is circulated, the graphical split between the two visualizes when we need to be extra cautious (that is when we open the valves that connect the two circuits A9 and A10). The circles marked A0 to A10 are electronic valves that operate on the atmospheric circuit, while the circles marked 0 to 14 are electronic valves that operate in the helium circuit, and the circles with crosses over marked M1-M3 are manual valves that need to be turned physically. The pumps are visualized by this icon  which shows the pumping direction (gas goes from the wide part (the backend), to the narrow part (the frontend) of the converging lines). 36

5.14	An image showing the frontpanel of the Gas Handling System (the GHS), which is illustrated in figure 5.13. The GHS is controlled by a serial connection to the cryogenics station, here we can open and close valves, we can read off pressures, and we can activate the connected pumps, there are also a number of manual valves which prevent us from automating this completely, but also reduces the likelihood of errors leading to loss of gas.	37
5.15	An image showing the three manual valves, M1, M2, and M3, that are attached inside the GHS. M1 and M2 control the mix dumps, and M3 is connected to an external hose that we use to create a vacuum where we need it, and to measure pressures (using P3) where we need it.	37
5.16	An image showing the manual valves that control flow through the trap, which filters impurities out of the helium mix. These need to be turned to open, and must be opened for helium mix to flow through the system.	39
5.17	An image showing the valves used for managing the atmosphere inside the cryostat sleeve. There is one more manual valve at the end of the hose marked "To cryostat sleeve". The SV valve is connected to M3, allowing us to create vacuum or measure the pressure of the exchange gas.	40
5.18	An image showing the crane used for moving the insert in and out of the cryostat.	41
5.19	An image showing the valve sealing the sleeve. We can use this valve to seal off the sleeve when the insert has been taken out.	41
5.20	An image showing the valve sealing the superconducting magnet. We use this valve to ensure a vacuum in the chamber containing the superconducting magnets to prevent moisture (and ice) damaging the coils.	42
5.21	An image showing the trap in full size (though rotated), encircled is the area containing the activated charcoal, which can be cleaned of impurities by heating with a heat gun.	43
5.22	An image showing the valve controlling waterflow to the turbopump, it is open in this case. When the turbopump is running it needs watercooling, but when the setup is idle we can turn off the waterflow to minimize the likelihood of contamination or a leak.	44
5.23	45
5.24	The status page. This is the first page the user sees when they navigate to the GUI. We see basic information about what is going on right now. The top plot shows the current temperature of the cryogenic setup, and the plot below shows the current status of the magnets (both the one attached to the function generator, and the superconducting DC magnet. Lastly we have a status over the datapoints in the latest experiment run, so we can see how far along the experiment is.	50
5.25	The configuration page. Here the user can setup specify the experimental parameters to use. When the configuration is saved, the system launches a new experiment.	51
5.26	The cryogenics configuration page. This page shows more detailed information about the status of system, including a plot of the temperatures with a longer timescale to see trends.	53
5.27	The data management page. Here the user sees an overview of the experiments that have been run, they can export data related to a specific experiment, and see an example of how to read the data.	55
6.1	A plot of temperatures in the cryogenic system during a cooldown where I was trying to figure out why the system was not cooling sufficiently. Initially the dilution refrigerator was not cooling, I checked the pressure of the exchange gas and found the none so I added some exchange gas (which is seen as the peak in temperature at approximately 330000 seconds). After the exchange gas was added it starts cooling correctly, and we reach approximately 0.36K. Once we reached 0.36K the system seemed to be saturated and I could not get it to condense any more mix.	57

6.2 Plots of the pressure right after the compressor (probe 2) and right after the scroll pump by the dump valves (probe 5), see figure 5.13 for a detailed view. The top plot shows the mix condensing after it has been added to the dilution refrigerator loop. The bottom plot shows the pressure before and after the compressor when new mix is added to the loop. We see the pressure before the compressor drop rapidly as the mix is moved into the dilution refrigerator, while later the pressure rebuilds once the uncondensed gas is pumped away from the still, this is a good example of the expected behavior. 58

6.3 Plot of the IVC pressure during the followup experiment to check if A0 was leaking. 59

6.4 Plots of the IVC pressure as a function of time in the experiment where we found the leaking valve. The top plot shows the IVC being pumped out, while the two remaining plots show the pressure problematically responding to new mix being added to the dilution refrigerator. 60

6.5 Plot of temperatures throughout the cryostat during the cooldown where we found the leaking valve. We see that the system cools down normally using the pulse tube compressor (the top figure), and we see the dilution refrigerator coming into action and cooling further beyond the approximately 4K the compressor can achieve. 61

6.6 A plot of the temperatures in the dilution refrigerator during the last part of cooling seen on figure 6.5, we see the temperatures drop to approximately 0.25K, which proves that dilution refrigeration was taking place, but unfortunately we were unable to cool further. 61

Chapter 1

Introduction

Research into magnetism and superconductivity currently requires a combination of cryogenic temperatures, and large magnetic fields, and it is generally under these extreme conditions we can make novel discoveries.

One experimental technique that thrives under these conditions is *AC Susceptometry*, where a sample responds magnetically to an external sinusoidal magnetic field, which we can utilize to determine frequency dependent behavior of the material.

The X-Ray and Neutron Science group has for a long time had a dilution refrigerator with superconducting magnets capable of producing the extreme environment we seek, but the apparatus is not ready to use for experimental work, and has historically had a large barrier to entry due to many undocumented facets of the instruments, as well as malfunctioning equipment.

In this thesis I address the issues by building an automated coil winder to ensure we can easily create new coils whenever they break, I write a standalone manual to document the processes required to obtain the extreme conditions we are interested in, and finally I develop a complete software suite that integrates all the necessary instruments to run and monitor the experiments from start to finish.

The thesis assumes a basic understanding of condensed matter physics, and targets an undergraduate student who has taken the course *Condensed Matter Physics 1* at the University of Copenhagen.

In the background chapter I briefly describe the necessary theory for understanding the link between magnetism and susceptibility, first from a bulk perspective, then from a microscopic perspective, I move onto susceptometry and describe how to determine susceptibility from measurement, and finally I describe in broad terms how a dilution refrigerator works.

Chapter 3 presents the newly developed software package for conducting susceptometry experiments using all the instruments available in the lab in an automated fashion, making the lab more accessible to new personal. Chapter 4 explains the hardware changes I have implemented in the course of my thesis.

Next we have a lab manual which defines three experimental stations that make up the experimental setup, here I describe which experimental apparatus we have in the lab, then I go into winding a set of coils for the susceptometer, how to setup a sample in the susceptometer. In the section "Using the cryostat" I go into details on how to operate the dilution refrigerator with a description on preparing the system, cooling down, and finally reheating and disassembling the experiment. The last section of the lab manual explains how to use the software to run an experiment and extract data for offline processing.

In the experiments chapter I discuss a few key experiments which illustrate the current state of the lab. In the conclusion and outlook I summarize the progress I made in the lab, the problems we currently face, and the future of the lab.

Finally in the appendix I go into more depth on how the software works from a perspective of extending it to include more functionality, as well as solutions to some of the most common problems.

Chapter 2

Background

In this chapter I will briefly cover the necessary theory to understand the experimental procedure and the resulting data treatment. First I cover magnetism where we start by defining the magnetic susceptibility by considering a magnetic bulk material, then we move down to the atomic level where we define the magnetic behavior of atoms, to understand this we apply the behavior to a two level system, which we use to derive the Curie-Weiss law, and then we consider a damped harmonic oscillator as a model for magnetic materials. After getting a better understanding of magnetism we move on to susceptometry where I sketch out the theory behind the experimental setup we use, and finally we consider dilution refrigeration that allows us to cool the system to very low temperatures.

2.1 Magnetism

The magnetic properties of a given material depend on the microscopic properties and arrangements of the atoms that the material is made up of, and these microscopic properties can in turn lead to macroscopic effects [1]. When we describe a magnetic system, we can either focus on the macroscopic effects in terms of the resulting electromagnetic fields, or derive the effects using a microscopic quantum-mechanical description of how the system's state depends on the applied external fields. Using AC-susceptometry we can extract the bulk characteristics which limits any microscopic model we may derive, thus simplifying the process of getting a complete model of a system [12]. In this manner AC-susceptometry allows us to gain deeper insights into the properties of a material.

2.1.1 Susceptibility

We begin by defining the magnetic susceptibility χ :

$$\chi = \lim_{H \rightarrow 0} \frac{\mathbf{M}}{\mathbf{H}} \quad (2.1)$$

where \mathbf{M} is the magnetization of the sample in the applied magnetic vector field \mathbf{H} . In a real experiment we need a finite magnetic field to get a valid signal, which means the vanishing limit cannot be achieved, to account for this we define the differential susceptibility χ_{exp} :

$$\chi_{exp} = \frac{\delta \mathbf{M}}{\delta \mathbf{H}} \quad (2.2)$$

In the (common) case where $\chi \ll 1$ we can approximate \mathbf{H} using the following relation:

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}) \approx \mu_0 \mathbf{H} \quad (2.3)$$

Leading to the approximate differential susceptibility [1], which applies to both AC and DC susceptibility, and is a very useful approximation used for modelling experimental data:

$$\chi_{\text{exp}} \approx \mu_0 \frac{\delta \mathbf{M}}{\delta \mathbf{B}} \quad (2.4)$$

When the system in question is isotropic, meaning the magnetization does not depend on the direction of the applied field, we may consider the susceptibility as a scaling constant where:

$$\mathbf{M} = \chi \mathbf{H} \quad (2.5)$$

Substituting eq. (2.3) and using the relation $\mathbf{B} = \mu \mathbf{H}$ we get[5]:

$$\mathbf{B} = \mu_0(1 + \chi)\mathbf{H} \quad (2.6)$$

$$\mu = (1 + \chi)\mu_0 \quad (2.7)$$

Which means given an expression of the magnetization of a system, we can compute the expected susceptibility which can be directly compared with the experimental results.

DC Susceptibility

For a DC susceptibility measurement we apply a small constant external magnetic field and measure the response, often by using a separate measurement coil and physically moving the sample and coil relative to each other and recording the induced current, or by measuring the torque the magnetic field gives rise to. We may describe the DC susceptibility using equation (2.4) [5].

AC Susceptibility

If we change the magnetic field as a function of time and measure the resulting magnetization, we may probe anywhere along the magnetization as a function of field by using a DC field as an offset and an AC field as a probe, we can probe $\frac{\partial \mathbf{M}}{\partial \mathbf{B}}$ directly[5], which is illustrated on figure 2.1. We can describe the applied magnetic field in terms of the DC and AC fields separately:

$$\mathbf{B}(t) = \mathbf{B}_{\text{DC}} + \mathbf{B}_{\text{AC}} \cos(\omega t) \quad (2.8)$$

Where ω is the angular frequency of the applied oscillating field, \mathbf{B}_{AC} is the amplitude of the AC magnetic field, and \mathbf{B}_{DC} is the DC magnetic field. Using this we can derive the AC susceptibility to be:

$$\chi_{AC} = \lim_{B_{AC} \rightarrow 0} \frac{\mu_0 M_{AC}}{B_{AC}} \quad (2.9)$$

$$= \mu_0 \left. \frac{\partial M}{\partial B} \right|_{\mathbf{B}_{DC}} \quad (2.10)$$

Using equation (2.10) we can probe the dynamics of a system by varying the frequency of the applied magnetic field, which is illustrated in figure 2.1.

In general this model is not quite sufficient to describe all systems and frequencies, to expand it we need to consider the susceptibility as a complex function dependent on the excitation frequency [12]:

$$\chi_{AC}(\omega) = \chi'_{AC}(\omega) + i\chi''_{AC}(\omega) \quad (2.11)$$

This splits the signal response into an in-phase component $\chi'_{AC}(\omega)$ and an out-of-phase component $\chi''_{AC}(\omega)$, where the in-phase component represents an interaction which conserve energy, and the out-of-phase component represents an interaction which dissipates energy through the material (in the form of heat)[12].

To gain a deeper understanding of the implications of the frequency dependence, we can consider the extremes of ω (see figure 2.2 for an illustration of the dependence for the harmonic oscillator). If we have a very

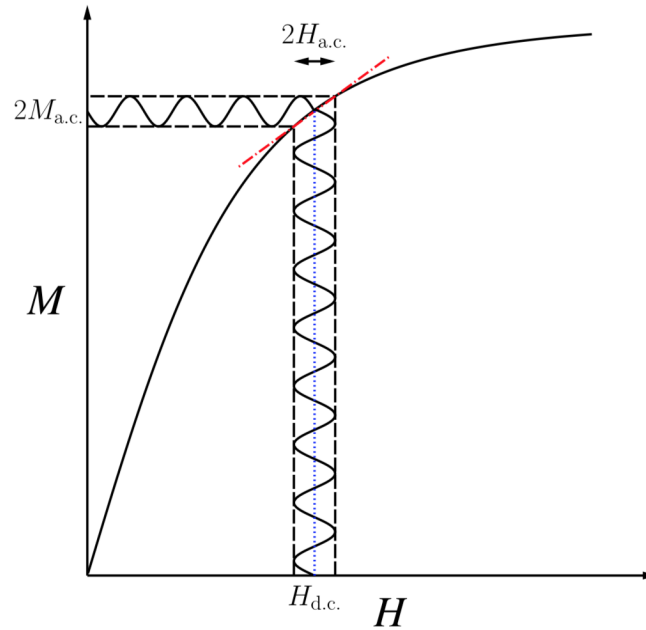


Figure 2.1: Illustration of the use of AC susceptibility to probe the gradient of the magnetization by applying a large DC field and an AC probing field simultaneously. Here the DC field is illustrated by the blue dotted line, the AC field is illustrated by the black sinusoidal line, the magnetization as a function of external field is illustrated by the black curve, and the gradient of the magnetization is illustrated by the red dotted line. Figure source: [12]

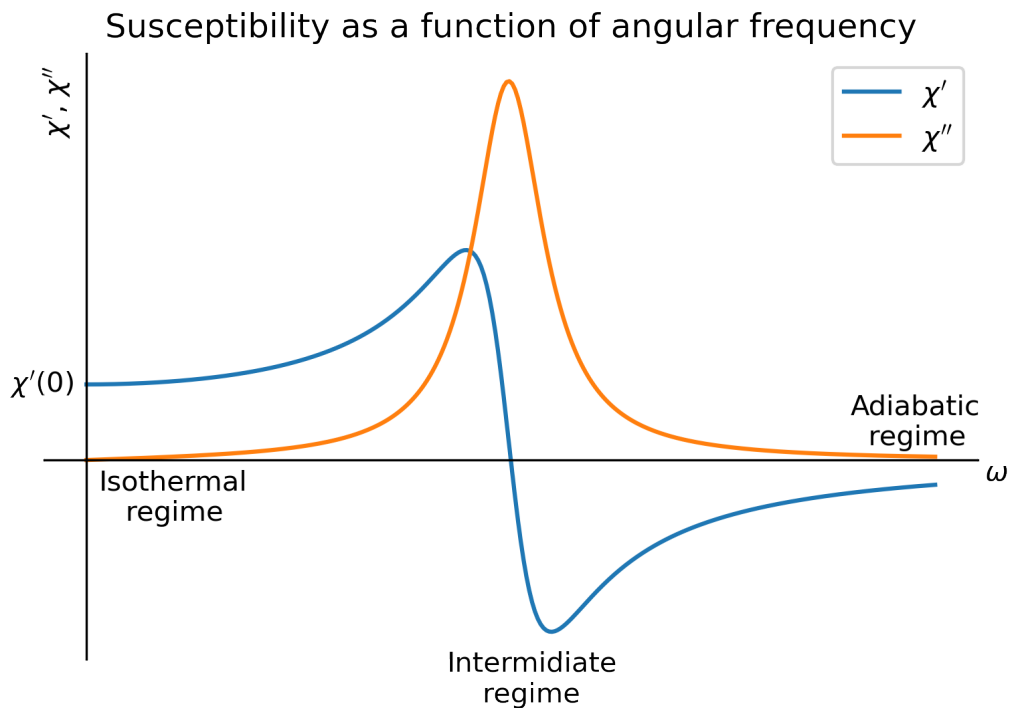


Figure 2.2: Example of the complex AC susceptibility split into χ' and χ'' according to equation (2.11) using the damped harmonic oscillator as the example system. We see three separate phases, first the isothermal regime where ω is much smaller than the systems characteristic time, next we have the intermediate regime where ω is on the order of the systems characteristic time, and finally we have the adiabatic regime where ω is much larger than the characteristic time.

low excitation frequency the magnetic moments are able to follow the driving frequency while staying in thermal equilibrium with the lattice of the material, in this case there is no energy transfer dissipation since the excitation is fully recovered and only the in-phase component remains, taking a value known as the isothermal susceptibility (χ_T)[12]. On the other end of the scale, at very high frequencies, the excitation is so fast that the process is adiabatic and similarly there is no energy dissipated. In this case there is still no out-of-phase component and the in-phase component takes a value known as the adiabatic susceptibility (χ_S)[12]. In the intermediate regime we have energy dissipation into the lattice, and the amount of energy deposited is dependent on the excitation frequency leading to a peak in the out-of-phase component[12].

2.1.2 Atomic theory of magnetism

Now that we have an understanding of how to measure the bulk magnetic properties, we would like to know what drives magnetism to occur in the first place to understand what the susceptibility actually measures, and to do this we delve into the realm of atoms.

Magnetism stems from the fundamental property of electron spin, which has a magnetic moment. We start by looking at a single electron, then we move on to an entire atom which we describe as a collection of electrons. We start with the electrons intrinsic magnetic moment[12]:

$$\mu_e = -g\mu_B \mathbf{s} \quad (2.12)$$

where g is the gyromagnetic ratio, μ_B is the Bohr magneton, and \mathbf{s} is spin of the electron[12]. Next we apply an external magnetic field, $\mathbf{B} = \nabla \times \mathbf{A}$ where \mathbf{A} is the magnetic vector potential, this shifts the energy by:

$$E = -g\mu_B \mathbf{B} \cdot \mathbf{s} \quad (2.13)$$

$$= -g\mu_B m_s B \quad (2.14)$$

$$\approx \mp \mu_B B \quad (2.15)$$

where $m_s = \pm \frac{1}{2}$ is the spin component of the electron which is parallel with the magnetic field[12], here I have assumed $g = |g_e| \approx 2$ to simplify the expression.

Next we consider a collection of electrons in a magnetic field, first we have the total Hamiltonian of the electron without a magnetic field:

$$\hat{H}_0 = \sum_{i=1}^Z \left(\frac{p_i^2}{2m_e} + V_i \right) \quad (2.16)$$

where p_i is the momentum of the individual electron, V_i is the potential of the individual electron, m_e is the mass of the electron, and Z is the atomic number which corresponds to the number of electrons[12].

The next step is to apply the magnetic field to our collection of electrons. We add the energy shift from eq. (2.13) to each electron, and replace the kinetic momentum with the canonical momentum[12]:

$$\hat{H} = \sum_{i=1}^Z \left(\frac{(\mathbf{p}_i + e\mathbf{A})^2}{2m_e} + V_i + g\mu_B \mathbf{B} \cdot \mathbf{s}_i \right) \quad (2.17)$$

where e is the elemental charge. Next we expand:

$$(\mathbf{p}_i + e\mathbf{A})(\mathbf{p}_i + e\mathbf{A}) = p_i^2 + e^2 \mathbf{A}^2 + 2e\mathbf{p}_i \cdot \mathbf{A} \quad (2.18)$$

Now we pick the symmetric gauge $\mathbf{A} = \frac{1}{2} \mathbf{B} \times \mathbf{r}$ where \mathbf{r} is the position in space and use the triple product identity ($\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a})$):

$$p_i^2 + \frac{e^2}{4} (\mathbf{B} \times \mathbf{r}_i)^2 + e\mathbf{p}_i \cdot (\mathbf{B} \times \mathbf{r}_i) = p_i^2 + \frac{e^2}{4} (\mathbf{B} \times \mathbf{r}_i)^2 + e\mathbf{B} \cdot (\mathbf{r}_i \times \mathbf{p}_i) \quad (2.19)$$

$$= p_i^2 + \frac{e^2}{4} (\mathbf{B} \times \mathbf{r}_i)^2 + e\hbar \mathbf{B} \cdot \mathbf{L}_i \quad (2.20)$$

where $\hbar\mathbf{L}_i = \mathbf{r}_i \times \mathbf{p}_i$ is recognized as the angular momentum, we insert eq. (2.20) into our Hamiltonian and get[1]:

$$\hat{H} = \sum_{i=1}^Z \left(\frac{p_i^2 + \frac{e^2}{4}(\mathbf{B} \times \mathbf{r}_i)^2 + e\hbar\mathbf{B} \cdot \mathbf{L}_i}{2m_e} + V_i + g\mu_B\mathbf{B} \cdot \mathbf{s}_i \right) \quad (2.21)$$

$$= \sum_{i=1}^Z \left(\frac{p_i^2}{2m_e} + V_i + \frac{e^2}{8m_e}(\mathbf{B} \times \mathbf{r}_i)^2 + \frac{e\hbar\mathbf{B} \cdot \mathbf{L}_i}{2m_e} + g\mu_B\mathbf{B} \cdot \mathbf{s}_i \right) \quad (2.22)$$

$$= \hat{H}_0 + \sum_{i=1}^Z \left(\frac{e^2}{8m_e}(\mathbf{B} \times \mathbf{r}_i)^2 + \mu_B\mathbf{B} \cdot \mathbf{L}_i + g\mu_B\mathbf{B} \cdot \mathbf{s}_i \right) \quad (2.23)$$

$$= \hat{H}_0 + \sum_{i=1}^Z \left(\frac{e^2}{8m_e}(\mathbf{B} \times \mathbf{r}_i)^2 + \mu_B\mathbf{B} \cdot (\mathbf{L}_i + g\mathbf{s}_i) \right) \quad (2.24)$$

we define the total orbital angular momentum of the atoms $\mathbf{L} = \sum_{i=1}^Z \mathbf{L}_i$ and $\mathbf{S} = \sum_{i=1}^Z \mathbf{s}_i$ the total intrinsic spin angular momentum[1]:

$$\hat{H} = \hat{H}_0 + \mu_B\mathbf{B} \cdot (\mathbf{L} + g\mathbf{S}) + \frac{e^2}{8m_e} \sum_{i=1}^Z (\mathbf{B} \times \mathbf{r}_i)^2 \quad (2.25)$$

Let's take a close look at eq. (2.25): The first term is the unperturbed Hamiltonian which describes the basic interactions of the collection of electrons. The second term in eq. (2.25) is the paramagnetic term which gives rise to magnetism parallel to the applied magnetic field ($\chi > 0$). The last term in eq. (2.25) is the diamagnetic term which gives rise to magnetism anti-parallel to the applied field ($\chi < 0$).

In most cases the paramagnetic term dominates, since it only vanishes when both the orbital and spin angular momenta are zero while the diamagnetic term is typically small.

2.1.3 Susceptibility of a simple system

Now that we have developed a theory of magnetism on the atomic level, we can consider consider a simple system in an external magnetic field, and since the diamagnetic term is typically small we will focus on the paramagnetic term. Given an atom with the total magnetic moment $J = |L + S| = \frac{1}{2}$ in a magnetic field B , the spin component in the direction of the magnetic field may take the values $m_J = \pm\frac{1}{2}$, if the magnetic moment of the atom is given by μ , then the energy shift relative the the atom without a magnetic field becomes[1]:

$$E = \pm\mu B \quad (2.26)$$

Now we can compute the partition function of this two level system:

$$Z = \sum_i \exp\left(\frac{-E_i}{k_B T}\right) \quad (2.27)$$

$$= \exp\left(\frac{-\mu B}{k_B T}\right) + \exp\left(\frac{\mu B}{k_B T}\right) \quad (2.28)$$

$$= 2 \cosh\left(\frac{\mu B}{k_B T}\right) \quad (2.29)$$

Given the partition function our simple two level system, we can create a partition function for a collection of N independent atoms, which is simply:

$$Z_N = Z^N \quad (2.30)$$

And the total free energy of the system is:

$$F = -k_B T N \ln Z \quad (2.31)$$

From the free energy we can compute the magnetization by first computing the total magnetic moment (m), and then dividing by the volume of our system[1]:

$$MV = m = - \left(\frac{\partial F}{\partial B} \right)_T \quad (2.32)$$

$$= k_B T N \frac{\partial}{\partial B} \ln \left(2 \cosh \left(\frac{\mu B}{k_B T} \right) \right) \quad (2.33)$$

$$\Rightarrow M = \frac{\mu N}{V} \tanh \left(\frac{\mu B}{k_B T} \right) \quad (2.34)$$

This magnetization is plotted as a function of H (instead of B) on figure 2.1.

The Curie law

When we apply a large magnetic field the magnetization will saturate and become $M = \mu N/V$, but if the ratio between B and T is approximately 0 we get that the magnetization and susceptibility becomes approximately inversely proportional to the temperature[12]:

$$M(B/T \approx 0) \approx \frac{\mu^2 N B}{V k_B T} \quad (2.35)$$

$$\Rightarrow \chi \approx \frac{\lambda}{T} \quad (2.36)$$

where $\lambda = \frac{\mu_0 \mu^2 N}{V k_B}$ is the Curie constant. The Curie law of paramagnetism is valid for systems where the spins do not interact, and a good approximation for systems where the interaction is weak[12].

The Curie-Weiss law

To expand the validity of the Curie law, we can consider the Curie-Weiss law for paramagnetic materials, which includes sample demagnetization and crystal structure[9]. To do this we start by considering the local magnetic field experienced by an atom:

$$B_{local} = B + B_{demag} + B_{Weiss} \quad (2.37)$$

Here the demagnetization is represented by B_{demag} which comes from magnetization of the sample, and the Weiss field is represented by B_{Weiss} which comes from the effects of surrounding dipoles in the crystal structure.

Let's consider an ellipsoidal sample, which is magnetized along its longest axis, has volume V and geometry factor D (with $D = \frac{1}{3}$ for a sphere), we can write the demagnetization field[9]:

$$B_{demag} = \frac{-D \mu_0 M}{V} \quad (2.38)$$

And the Weiss field:

$$B_{Weiss} = \frac{\alpha \mu_0 M}{V} \quad (2.39)$$

where α depends on the crystal structure ($\alpha = \frac{1}{3}$ for cubic lattices [9, Equation 12.35 on p. 321]). This leads to the local susceptibility and local field:

$$\chi_{local} = \frac{\mu_0 M}{B_{local}} = \frac{\lambda}{T} \quad (2.40)$$

$$\Rightarrow B_{local} = \frac{\mu_0 M T}{\lambda} \quad (2.41)$$

We can now solve for the external field to find a correction:

$$B_{local} = B + \frac{-D\mu_0 M}{V} + \frac{\alpha\mu_0 M}{V} \quad (2.42)$$

$$\Rightarrow B = B_{local} - \frac{-D\mu_0 M}{V} - \frac{\alpha\mu_0 M}{V} \quad (2.43)$$

$$= B_{local} - (\alpha - D) \frac{\mu_0 M}{V} \quad (2.44)$$

$$= \frac{\mu_0 M}{\lambda} \left(T - (\alpha - D) \frac{\lambda}{V} \right) \quad (2.45)$$

$$= \frac{\mu_0 M}{\lambda} (T - \Delta) \quad (2.46)$$

where Δ is the Weiss constant. Let's now consider the definition of DC susceptibility and insert the corrected B field we just found:

$$\chi_{DC} = \lim_{B \rightarrow 0} \frac{\mu_0 M}{B} \quad (2.47)$$

$$= \lim_{B \rightarrow 0} \frac{\mu_0 M}{\frac{\mu_0 M}{\lambda} (T - \Delta)} \quad (2.48)$$

$$= \lim_{B \rightarrow 0} \frac{\lambda}{T - \Delta} \quad (2.49)$$

which is the Curie-Weiss law[12].

2.1.4 Damped harmonic oscillator

To expand our understanding further, let us discuss an example. The damped harmonic oscillator without inertia shows us how we can predict the frequency-dependent susceptibility of a simple theoretical system.

The magnetization is driven by an external magnetic field H , which oscillates with the frequency ω , it has a spring constant k and a damping term α :

$$\alpha \dot{M} + kM = H \quad (2.50)$$

We insert $H(\omega)$ as a purely complex oscillation, and from this we solve for the susceptibility[12]:

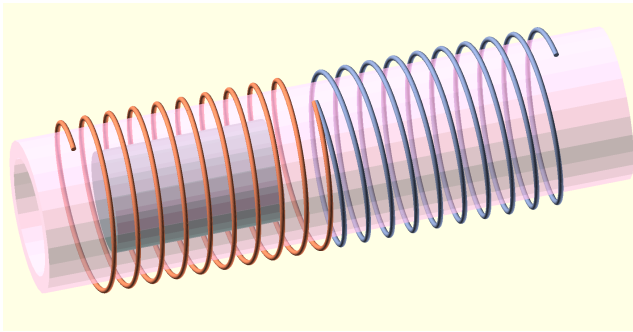
$$\chi(\omega) = \frac{M(\omega)}{H(\omega)} = \frac{k^{-1}}{1 - i\omega\tau} \quad (2.51)$$

where $\tau = \frac{\alpha}{k}$ is the relaxation time of the system. We can now separate the system into a real and an imaginary part (which can be seen illustrated on figure 2.2):

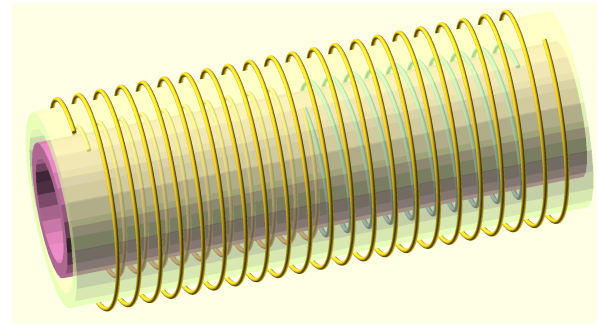
$$\chi'(\omega) = \frac{k^{-1}}{1 + \omega^2\tau^2} \quad \chi''(\omega) = \frac{\omega\tau k^{-1}}{1 + \omega^2\tau^2} \quad (2.52)$$

We can see that both susceptibilities go to 0 when ω goes to infinity, which means the adiabatic susceptibility is 0. If the system in question has an adiabatic susceptibility, we can add a term (χ_S) to describe it:

$$\chi'(\omega) = \chi_S + \frac{\chi_T - \chi_S}{1 + \omega^2\tau^2} \quad \chi''(\omega) = \frac{\omega\tau(\chi_T - \chi_S)}{1 + \omega^2\tau^2} \quad (2.53)$$



(a) 3d model of the measurement coils wound in opposite directions, with a sample inside the left chamber. The coil pair measure the difference in susceptibility between the two coils.



(b) 3d model of the excitation coil containing the measurement coil as illustrated in figure (a). The excitation coil creates a magnetic field around both of the coils in the measurement pair, the field is theoretically identical between the two (approximately identical in real life due to fabrication errors), meaning we can use the two coils to measure a difference in susceptibility.

Figure 2.3: 3d models of the susceptometer, produced using OpenSCAD.

This model, and its generalizations, are useful for providing a phenomenological description of measured data, as well as extracting relaxation times of systems from frequency dependent measurements of susceptibilities[12].

2.2 Susceptometry

Now that we have a basic understanding of the theory behind susceptibility, we can delve into the experimental counterpart: Susceptometry. In this section we will explore the working theory behind a susceptometer, which will lay the groundwork for building our own, as well as understanding the data it outputs.

To measure the susceptibility, we place a sample inside a well known magnetic field, and measure the influence of the sample on the field[5].

On figure 2.3 we see an example of a system of coils which is setup to detect the difference in magnetization between a sample and empty space. The excitation coil (seen on figure 2.3b) is driven by an alternating current creating an AC magnetic field we can control, inside the excitation coil sits two identical (but oppositely wound) detection coils wired in series (seen on figure 2.3a). If we put a sample inside one of the chambers a difference in susceptibility between the two detection coils will lead to an induced current in the detection coils[5].

The design with two detection coils inside a common excitation coils has a number of advantages. The differential measurement allows measuring the difference in susceptibility with a high sensitivity, we only need to wind three coils, and the compact design of the susceptometer means we can place it inside a complex sample environment (in our case we place it inside a cryostat surrounded by a superconducting magnet allowing us to control the DC magnetic field). The design naturally has downsides as well, the two detecting coils need to be as identical as possible, the system is sensitive to alignment issues of both the detecting coils inside the excitation coil (if they experience a difference in magnetic field, the difference will show in the resulting data), and alignment issues of the sample (it needs to be placed as close to the center of one of the detecting coils as possible). All these issues may be addressed by trial and error, either by winding many coils to find the best set, and by aligning the coils to minimize the error when we have no sample in the chamber[5].

2.2.1 Determining the susceptibility from measurements

In this section we want to derive how the susceptibility gives rise to a voltage signal using the susceptometer we described in the previous section.

The time-dependent magnetic field \mathbf{B} induces an electromotive force (EMF) in each turn of the measurement coils[5]:

$$\varepsilon = -\frac{\partial\Phi}{\partial t} = -\iint \frac{\partial\mathbf{B}}{\partial t} \cdot d\mathbf{a} \quad (2.54)$$

where Φ is the total magnetic flux through each turn and $d\mathbf{a}$ is an infinitesimal element of the cross sectional area of the turn. Using $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$ the EMF becomes:

$$\varepsilon = -\mu_0 \iint \frac{\partial\mathbf{H}}{\partial t} \cdot d\mathbf{a} - \mu_0 \iint \frac{\partial\mathbf{M}}{\partial t} \cdot d\mathbf{a} \quad (2.55)$$

but due to the differential nature of the measurement coils, the \mathbf{H} term goes to 0 due to an equal and opposite contribution from the coil without a sample leaving only the \mathbf{M} term[5]:

$$\varepsilon = -\mu_0 \iint \frac{\partial\mathbf{M}}{\partial t} \cdot d\mathbf{a} \quad (2.56)$$

The \mathbf{M} term only appears when there is a sample present in one of the measurement coils. The voltage is induced in each of the N turns of the measurement coil, and thus only depends on the AC part of the magnetization[5]:

$$V = -\mu_0 N \iint \frac{\partial\mathbf{M}}{\partial t} \cdot d\mathbf{a} \quad (2.57)$$

We seek to find the magnetization, in order to do so we need to evaluate the integral. In the ideal case the magnetization is constant throughout the cross sectional area, here we can replace the integral by the product of the magnetization and the area. In the real world this is not the case, but we assume any other effects may be accounted for using a scalar constant c :

$$V = -c\mu_0 N A \frac{\partial\mathbf{M}}{\partial t} \quad (2.58)$$

We define a constant $C = c\mu_0 N A$ which depends on the specific susceptometer as well as the sample size and shape. If we assume our material to be linear, we can write the magnetization as time-dependent function using first order terms[5]:

$$M(t) = H_{0,AC}(\chi' \cos(\omega t) + \chi'' \sin(\omega t)) \quad (2.59)$$

where $\chi = \chi' + i\chi''$ is the complex susceptibility, $H_{0,AC}$ and ω is the amplitude and frequency of the excitation field. We now insert this into equation (2.58) and get:

$$V = -C\omega H_{0,AC}(\chi' \cos(\omega t) + \chi'' \sin(\omega t)) \quad (2.60)$$

$$= V' \sin(\omega t) - V'' \cos(\omega t) \quad (2.61)$$

We can now solve for the susceptibility assuming the calibration constant C is known, and we have measured V' and V'' :

$$\chi' = \frac{V'}{C\omega H_{0,AC}} \quad \chi'' = \frac{V''}{C\omega H_{0,AC}} \quad (2.62)$$

With this theoretical background we can now understand the output of our susceptometer.

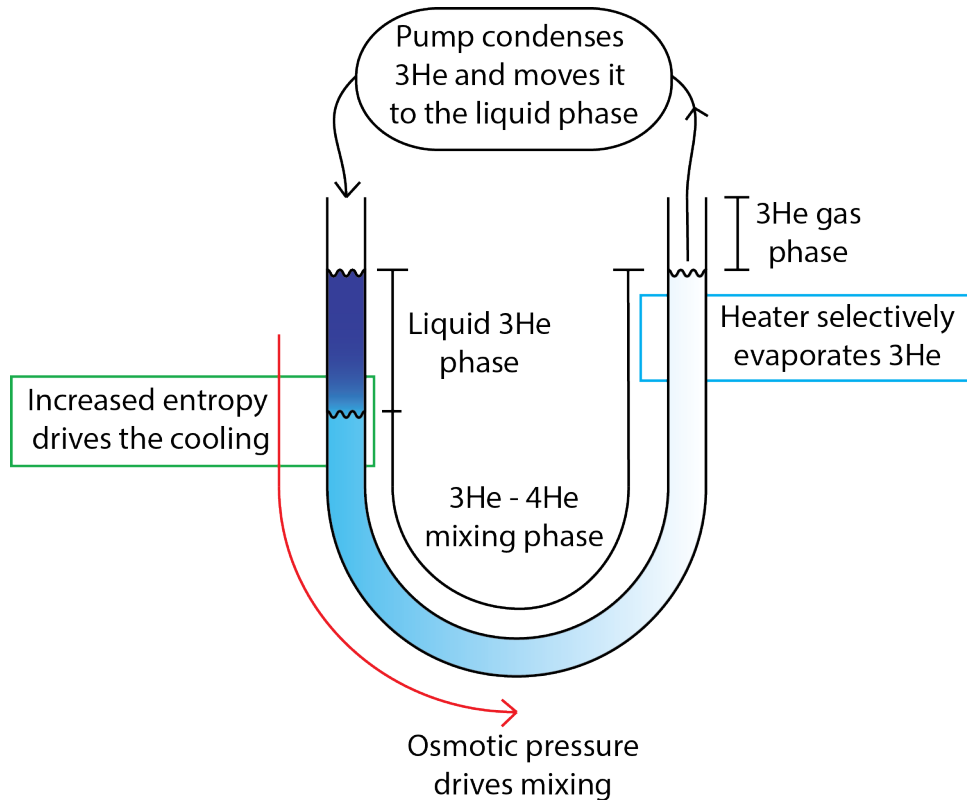


Figure 2.4: A simplified overview of the principle behind dilution refrigeration. The blue gradient indicates the relative concentration of ^3He to ^4He , it is exaggerated for illustrative purposes. In the figure we see the helium mix settle out as two separate phases (due to the difference in mass), the phases mix spontaneously in the mixing chamber (due to the Van der Waals force) leading to increased entropy inside the mix, and a decreased entropy to anything thermally coupled to the mix. The ^3He moves through the mixing phase until it reaches the still, where ^3He is selectively removed using a heater (due to a difference in boiling points).

2.3 Dilution refrigeration

In this section I will discuss the background behind the dilution refrigerator, which is the most important cooling technique for achieving temperatures between about 5mK and 1K[9]. The theory behind the concepts and illustrations is largely based on the textbook by Pobell[9]. I refer to helium-3 as ^3He and helium-4 as ^4He throughout this section.

2.3.1 The working principle behind dilution refrigeration

The principle behind dilution refrigeration is an entirely quantum mechanical concept, as it utilizes the idea of zero-point motion which stems from Heisenberg's uncertainty principle. The principle tells us that an atom with a very well defined position cannot have a very well defined momentum which we interpret as movement, even at absolute zero. In the end we have a somewhat undefined position and a somewhat undefined momentum, and that is essentially vibration. The amount of vibration depends on the mass of the atom, a smaller mass leads to a larger vibration.

Next we consider the attractive forces between two helium atoms, since helium is a noble element there is very little attraction, so the only attractive force between the atoms is the Van der Waals force, which depends very strongly on the distance. If we consider the attraction between two ^3He atoms versus a ^3He and a ^4He atom we can conclude that the Van der Waals attraction is greater between the ^3He and ^4He atoms when compared to two ^3He atoms, as the mean distance due the zero point motion is smaller because of the larger mass of the ^4He

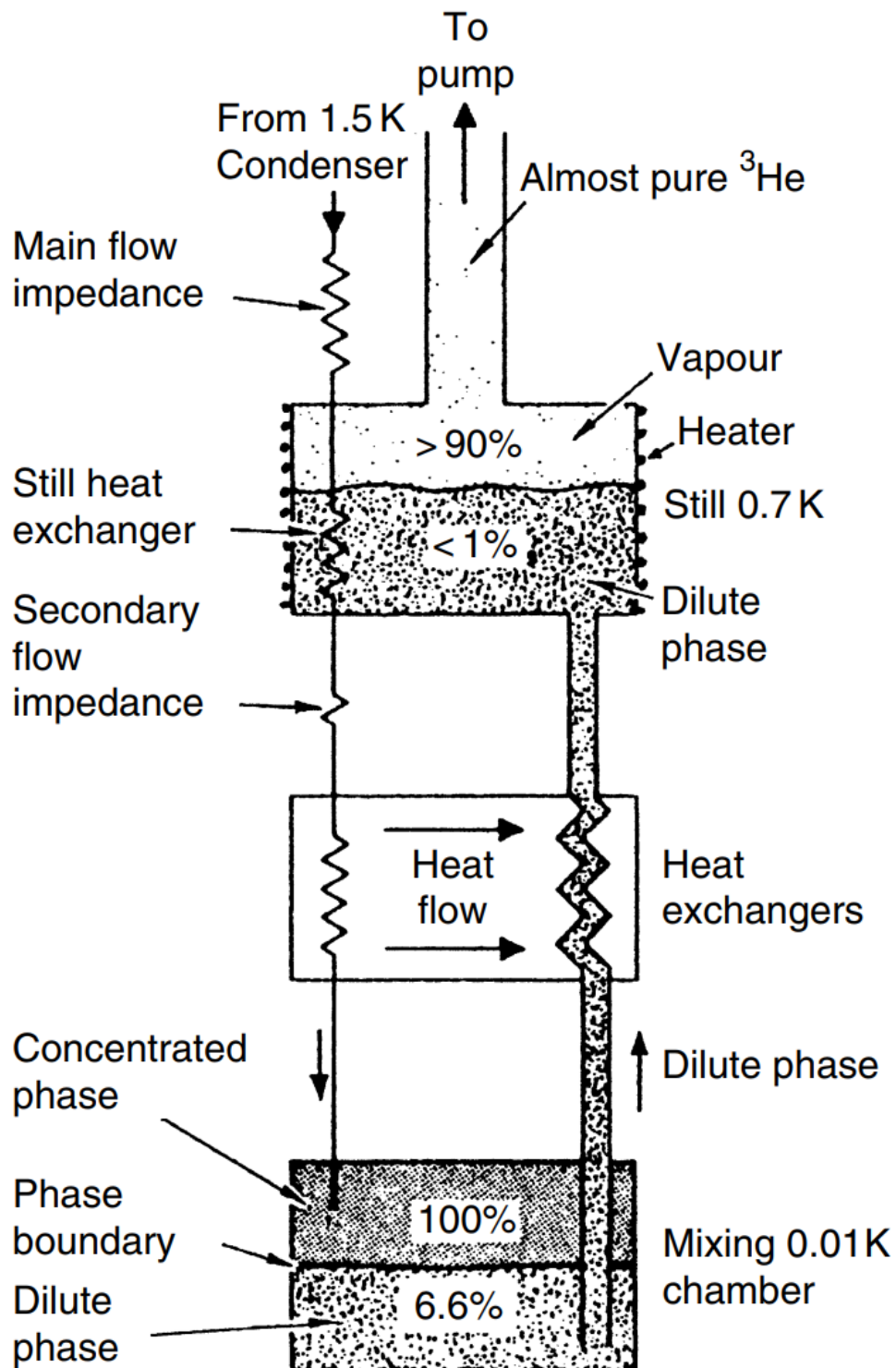


Figure 2.5: A more realistic illustration of an implementation of dilution refrigeration. Liquid ^3He enters from the 1.5K condenser line and flows down to the mixing chamber exchanging heat underway. In the mixing chamber it replenishes the lighter ^3He phase, and can possibly dissolve into the dilute ^3He - ^4He phase. The dilute phase is connected to the still where the ^3He is distilled off to recirculate in the loop. The percentages given are concentrations of ^3He . Figure from Pobell[9].

atom.

Now we consider the U-shaped tube illustrated on figure 2.4, when we fill this tube with a mix of liquid ^3He and ^4He , the two liquids should separate into a liquid ^3He phase, and a liquid ^4He phase due to the difference in weight, but the two phases should also mix at the interface between them due to the Van der Waals attraction between ^3He and ^4He . The mixing means our system becomes less ordered, leading to an increased entropy. We can supply the entropy in the form of thermal energy by thermally coupling our sample to the mixing interface.

The ^3He that is mixed into the ^4He phase will diffuse through the ^4He phase by osmotic pressure to balance out the concentration, this means the system will cool until we reach an equilibrium concentration of ^3He in the ^4He . To keep the cooling going we need to selectively remove ^3He from the ^4He phase (on the right side of the tube illustrated on figure 2.4), we can achieve this by heating the mixed ^3He - ^4He phase which will preferentially boil off ^3He as it has a lower boiling point. The evaporated ^3He gas can be pumped away, condensed, and dropped into the liquid ^3He phase, restarting the cycle to further cool the system.

A more realistic illustration can be found on figure 2.5 where we can see how the heat exchangers are connected to the different phases of the dilution refrigeration, optimizing the cooling effect while circulating the ^3He in a closed loop.

Chapter 3

Software developed

3.1 Overview of software developed

Previously the instruments have been controlled either by manually dialing in the parameters on the physical device, or by a collection of LabView¹ scripts. LabView is expensive proprietary software, and since most of the original software was created, open source alternatives have been developed. The new software developed for instrument control and data collection is based around QCodes², which is a python package developed partly by QDev at UCPH, it creates consistent interfaces to the instruments by using object oriented programming, and has an integration to PyVisa³ which makes connection and automation of most instruments much easier. Physically the experimental setup is distributed between two stations (two separate computers), which, if we want to automate the data collection, means we need communication between the stations, this is achieved using WebSocket⁴ connections between each of the two stations and a central data management station. A sketch of the network can be seen on figure 3.1.

3.2 Socket based communication

The new control system is implemented in 4 separate components, we have a web server which the user can access for a GUI, here the user can setup experiment parameters and monitor results in real time, they can also download data from previous runs to do offline data analysis. Along with the webserver is a socket server which provides the access point for the measurement stations, as well as the real time data served to the GUI. The server is currently run on the magnetism station, but can be run on any other computer by changing the IP-address the clients connect to. We have two socket clients connected to the socket server, one running on each computer, together they control all of the instruments to set the experiment state, and measure all relevant parameters. Instructions are sent to the two clients through the websockets (using SocketIO⁵), and measurements are returned to the socket server where they are saved and sent to any GUI that is connected. One of the major advantages of this setup is the real time data access, another is the ability of one station waiting for the other to achieve a set state forming a closed control loop. Technical details on the implementation of the system can be found in appendix A.

¹<https://www.ni.com/da-dk/shop/labview.html>

²<https://github.com/QCoDeS/Qcodes/>

³<https://pypi.org/project/PyVISA/>

⁴<https://en.wikipedia.org/wiki/WebSocket>

⁵SocketIO can be found here: <https://socket.io/>

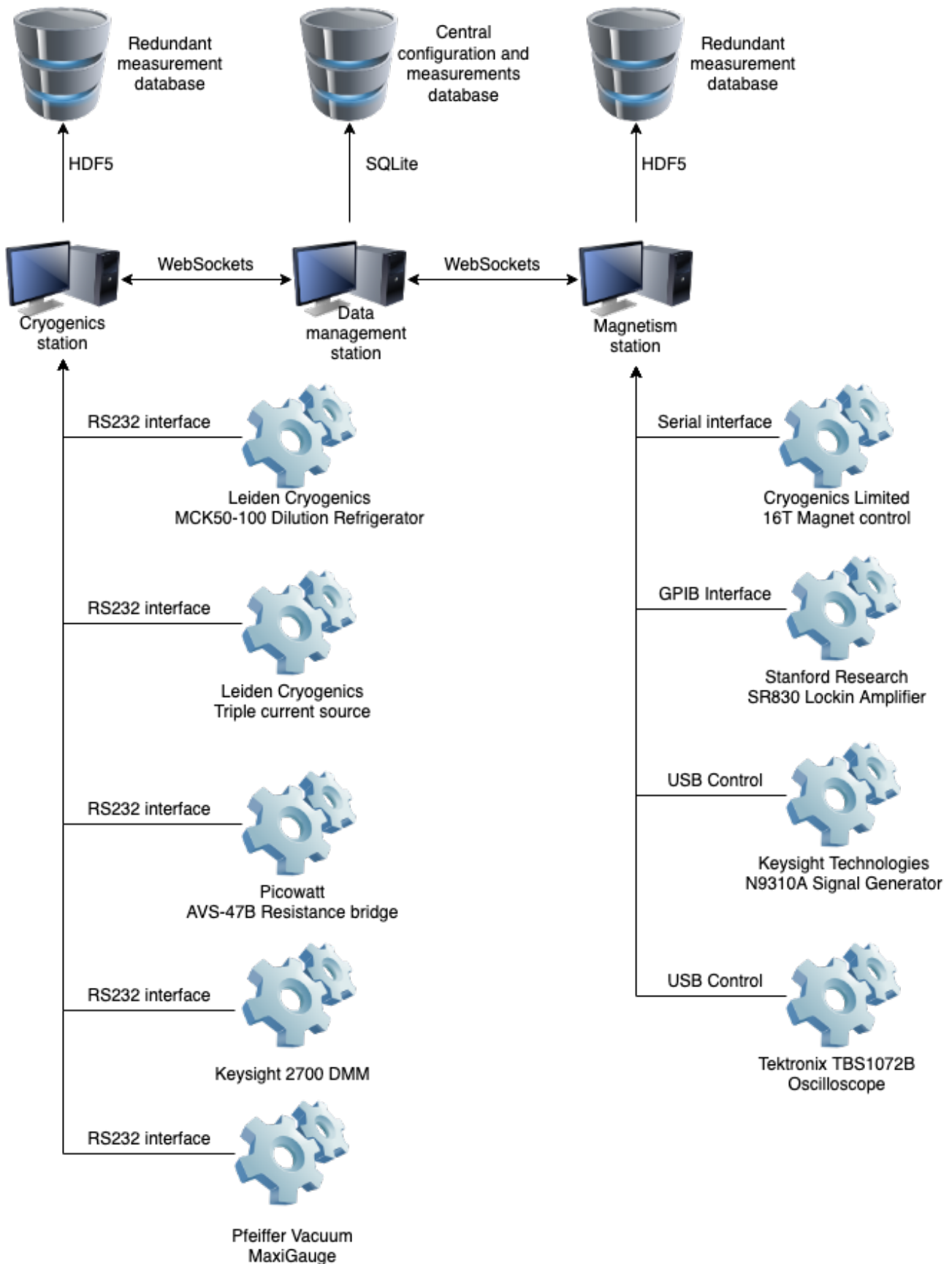


Figure 3.1: Sketch to illustrate how the different experiment stations are connected, and which instruments are connected to each. The cog illustrations are meant to illustrate a generic instrument, while the lines into the instrument shows what type of data connection is currently in place.

3.3 Susceptometry control panel

The control panel is implemented as a web application, this way we can ensure easy access using all types of devices (we can even use this to monitor an experiment using a smartphone), additionally it is very easy to find libraries to ease the implementation and available documentation is also excellent. The implementation uses a mixture of HTML (for creating the structure of the site), CSS (for improving the user experience and aesthetic), and JavaScript (to make the application behave dynamically and respond to updates from the server).

3.4 Instrument drivers

The instruments are controlled using QCodes, and one of the requirements for controlling instruments is a QCodes driver, which is an interface that defines how the hardware is to be controlled. There are two types of QCodes driver: VISA drivers and DLL-based drivers. VISA drivers control instruments that implement a standardized communication protocol, while DLL-based drivers are for instruments that are controlled by other means (usually proprietary protocols). Before the project began only two instruments already had drivers available, and I have developed the 8 remaining drivers.

3.4.1 Picowatt AVS-47B Resistance bridge

Two versions of this driver was developed, first I assumed the resistance bridge had a signal converter box which maps VISA commands to a synchronous RS232 protocol, this assumption was made from impressions given by the programming reference manual [8]. The first version was a simple VISA driver where we simply mapped each instrument command to a parameter defined in code, but since we do not have a Picowatt signal converter box, this driver is ultimately not used.

The second driver is a DLL based driver where we hook into the individual datalines of the RS232 connection interface to communicate with the device directly. In this version each interaction between the controlling computer and the resistance bridge consists of the following 9 steps:

1. Construct the transmission string by encoding the instrument parameters in binary.
2. Allocate a binary string to contain the results of the interaction.
3. Encode the address of the resistance bridge in binary.
4. Send the address to the resistance bridge and simultaneously read the old address using the datalines.
5. Strobe the datalines to commit the new address.
6. Send the encoded transmission string and read the received string to/from the resistance bridge using the datalines.
7. Decode the results, these contain the old settings, the ADC value, and status parameters.
8. Convert the ADC value to a resistance using the range settings.
9. Convert the resistance to a temperature using calibration data (seen on figure 3.2).

More details on the communication protocol can be found in the Picowatt manual [8].

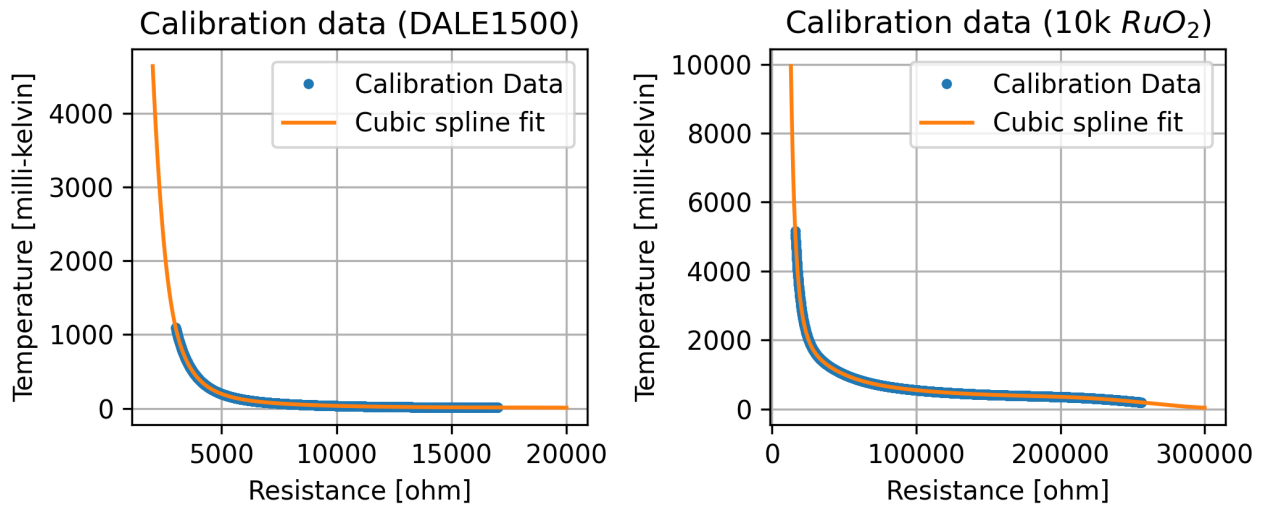


Figure 3.2: Calibration data extracted from Leiden Cryogenics LabView modules fitted to a cubic spline interpolation. The calibration shows the effective range of the DALE1500 is approximately 1-1000 millikelvin while the 10k RuO_2 has an effective range of approximately 10-2000 millikelvin.

Calibration of the resistance bridge

The calibration of the resistance bridge is hidden inside the software provided by Leiden Cryogenics as a LabView module, making it difficult to integrate with the Python module. To extract the calibration I first experimented by manually inputting some resistances into the LabView program, and seeing the converted temperatures. The manual probing told me that the calibration is approximated by some smooth function with a minimum (about $12.2mK$ for DALE1500 sensor), and it gave me a starting point to write a LabView program to automatically extract and save the calibration data.

The LabView program created to extract the calibrations is very simple, it scans over resistances in the calibration range of the two sensors[2] and saves the two resistances and corresponding temperatures to a text file.

After obtaining the calibration data from the LabView module, I wrote a python script to convert the LabView dataformat to a plain CSV. The CSV is imported into the resistance bridge driver where it is approximated by a cubic spline (To obtain a smooth function very similar to the one available in the LabView module), the cubic spline is then used to convert between resistance and temperature when the user queries the driver. The resulting data and cubic splines can be seen on figure 3.2, they behave as expected within the calibration range.

3.4.2 Keysight Technologies 2700 Digital Multimeter

The digital multimeter is used to measure temperatures outside the dilution refrigerator to determine how the Cryomech PT415 pulse tube compressor is behaving. The driver was also simple to implement as the device has a standard VISA interface which is well documented in the manual. It is simplified further by only implementing a scan over the 9 resistance channels we're interested in.

The device outputs a resistance in Ohms, which we must convert to a temperature. The conversion is done using linear interpolation (this is how it was implemented on the original system) of some calibration data covering a wide temperature range. The original calibration data was stored as a collection of CSV files (one for each sensor), making it very easy to import and use in our new system.

3.4.3 Keysight Technologies N9310A Signal Generator

The signal generator implements VISA so the python driver is relatively simple, in addition we only need the low frequency (LF) module as the experiment is optimized to run at 1 kHz, further simplifying the driver. The driver can control the frequency and amplitude of the LF output, and it can turn it on and off. These are all implemented as simple QCodes parameters.

3.4.4 Stanford Research Systems SR830 Lock-in amplifier

The lock-in amplifier used in the experimental setup is extremely popular in the scientific community, so drivers were already available to control it. To integrate it into the system I used the driver bundled with QCodes⁶.

3.4.5 Leiden Cryogenics MCK50-100 Dilution Refrigerator

The gas handling system (GHS) of the dilution refrigerator also implements a VISA interface, although it has a nonstandard identifier command and format. With this driver we can retrieve the status of every LED on the GHS, we can read the pressure sensors, we can read the pressure settings (for alarm values), and we can read the key status of every button on the GHS. Additionally we can also send a signal which is equivalent to pressing a button, and through this we can control all aspects of the device (except the manual valves). Currently automated cooldown is not implemented in this driver (to ensure we do not lose expensive helium), but it is possible to implement such a system to complete the integration between all the systems.

3.4.6 Leiden Cryogenics Triple current source

The triple current source implements a communication protocol similar to the GHS, so with some configuration we can still use a VISA driver for this. The driver implements control of the 3 channels independently, where the user can specify the on/off state of each channel, whether any channel needs to be gated, and what current to output to each channel (including auto-ranging the channels).

3.4.7 Pfeiffer Vacuum MaxiGauge Pressure controller

The pressure controller can measure the pressure in up to 6 analog pressure gauges (though only two are currently installed), the controller implements a protocol that resembles VISA, but is not compatible, it communicates using the following steps:

1. Host sends a command mnemonic (for example PR6 which measures the pressure on gauge number 6)
2. Controller sends ACK og NAK signal
3. Host sends ENQ signal
4. Controller sends results

This means that the driver was implemented as a DLL driver which calls the controller directly using the PySerial module, it has a parameter for each of the six channels, and can get the pressure using the parameter.

3.4.8 Cryogenics Limited 16T Magnet controller

The magnet controller implements a standard VISA control where we can control the current running through the magnet (and through that we can control the magnetic field), the driver is implemented so we automatically switch to the Tesla unit.

⁶The documentation for the driver can be found here: https://qcodes.github.io/Qcodes/examples/driver_examples/Qcodes%20example%20with%20Stanford%20SR830.html

3.4.9 Agilent Analog Discovery 2

This mixed signal oscilloscope is controlled by a proprietary library that is distributed in DLL form, so to interface with it we need to write a DLL-based driver. It is simplified by using a python wrapper⁷ for the DLL. The driver implements control over acquisition frequency, buffer size, ranging, trigger, and enabling of each channel, it can get a trace and it can compute the RMS-value of said trace. Using the RMS value we can use it as a voltmeter in the experimental setup. As this was my personal oscilloscope we replaced it with a Tektronix TBS1072B, so ultimately this driver was not used.

3.4.10 Tektronix TBS1072B

The replacement oscilloscope for the agilent Analog Discovery 2 is a Tektronix TBS1072B which has the same programming interface (although slightly less features) as the Tektronix TPS2012B. A driver for the TPS2012B is bundled with QCodes⁸, so to control it we simply import the existing driver and begin sending requests.

3.5 Integrating the driver into the stations

To create a bridge between the instrument and the remaining software suite, we create two QCodes stations (one for the cryogenics station, and one for the magnetism station), this is a fairly simple module that collects the instrument addresses and uses them to instantiate the drivers that are relevant to each station, this ensures that they can connect on startup and that they are available to the rest of the software suite.

3.6 Socket clients

The next step in the software stack is the socket clients, which establish a connection between the instrument stations, and the data collection station. The socket clients contain two main components, firstly we have a WebSocket interface that can send and receive data to and from the central web server, it has methods that represent each of the function calls we wish to make. When a message is received by the WebSocket interface, it places a request in a queue, which then executes the function we wish to call and sends a response back using the WebSocket interface. In addition to the the messages sent back in response to a call, some monitoring data is sent every 5 seconds from the queue through the WebSocket interface back to the central web server. The monitoring data is displayed in tables and graphs on the website, and is updated in real time.

3.6.1 Collecting data using the socket client (process_next_step)

When an experiment is configured the central web server creates a number of steps corresponding to each point in a scan over the parameters selected, each step contains all the information necessary to configure the entire experimental setup to collect a single datapoint.

When an experiment starts the web server select the first datapoint which hasn't been executed and it sends it to both of the socket clients. In the socket clients a call to the method "process_next_step" is placed in the queue along with the experiment configuration for that step. The client then waits for any previous steps to be completed, then it configures the instruments and sends a ready signal to the other socket client. Once both clients are ready to take the measurement, they both collect all the data that is available to them for the specific configuration.

The data is saved locally in a HDF5 format to ensure no data is lost, and then the data is transmitted to the web server where it is saved an extra time to the central database, and where it is made available to the user (in the data management interface of the website).

⁷It can be found here: <https://pypi.org/project/dwf/>

⁸The documentation for the oscilloscope driver can be found here: https://qcodes.github.io/Qcodes/examples/driver_examples/Qcodes%20example%20with%20Tektronix%20TPS2012.html

3.7 The webserver

The webserver acts as the central broker for all communication between the stations and all the users of the application. The server is built in two parts, the first part is a traditional HTTP server built using aiohttp⁹, and the second part is the WebSockets server built using SocketIO¹⁰.

3.7.1 The HTTP server

The HTTP server has a few roles to play in the bigger picture, it provides the endpoint for the users interaction with the system. This happens by serving a HTML file when the browser navigates to the IP-address of the system, the browser then requests some JavaScript and CSS files based on the contents of the HTML file, these additional files are also served by the HTTP server, finally the browser requests a WebSocket connection which is brokered by the HTTP server.

In addition to this basic functionality the HTTP server also sends some dynamically generated files when they are required. Currently two are implemented, firstly we have a data export which queries the central database for all data related to an experiment and generates a file containing a large JSON object which is downloaded by the user requesting the data export. The second file is a plot of the temperatures over a long period of time (this period is configurable, but defaults to 120 hours).

The long time range plot differs from the realtime plots that are otherwise present in the system in the amount of data that can be present. The JavaScript program that runs the realtime plots is quite slow, and is not great at handling large amounts of data, so for the realtime plots on the web interface we limit the number of plotting points to 50, whereas the long time range plot cannot have such limitations if we wish to see a decent picture of how the temperature behaves over time. The plot is implemented as part of the HTTP server, where it uses matplotlib to generate a plot based on the latest data and send it to the user.

3.7.2 The socket server

The socket server handles the vast majority of communication, it is divided into 3 namespaces, one for the browser, one for the magnetism client, and one for the cryogenics client. The user connecting through the website can only interact with the browser namespace, which then in turn queries the two other namespaces for the information required, the two other namespaces respond by querying the browser namespace to send the results back to the user. Each namespace maintains a list of connected clients, as well as some basic state information which is used to keep track of what each component of the system is doing. The stateful nature of the namespaces allow us to provide the user with information about what parts of the system are operational, it also allows us to keep track of when the clients are ready to take a measurement.

⁹Documentation for the module can be found here: <https://docs.aiohttp.org/en/stable/>

¹⁰The documentation for SocketIO can be found here: <https://python-socketio.readthedocs.io/en/latest/index.html>

3.8 Extra software developed during the project

In addition to the susceptometry system, I also created a number of small programs during the course of my thesis.

Most significantly I wrote the firmware required to run the automatic coil winder¹¹, this is written in C using the Arduino standard library and compiled with the help of Platform IO¹². When I exported the temperature calibrations from the existing cryogenics setup, I wrote a program to convert the file from a proprietary labview input format to a normal CSV. I also wrote a small program to use with the labs smaller cryostat, the program measures the temperature inside the cryostat and the input of the lock-in amplifier and saves the data to a CSV. This was also developed using QCodes. And finally I wrote a program to monitor the temperature and pressure of the cryostat to find out which valve was leaking.

¹¹You can find the source code here: https://github.com/hyatt03/coil_winder

¹²Which can be found here: <https://docs.platformio.org/en/latest/core/installation.html>

Chapter 4

Hardware

In this chapter I will describe what hardware has changed during this thesis, and why it needed to be upgraded. The upgrades were funded by the Lighthouse Quantum Materials project.

4.1 Automated coil winder

The lab already had a manual coil winder (see figure 4.1a), which was used to wind the coils for previous projects [5] [11]. The version seen on figure 4.1a was designed and produced by Jan H. Fagerlund and Dennis W. Wistisen of the NBI mechanical workshop, and was initiated by Mathias Mikkelsen and Kim Lefmann of the X-Ray and Neutron science group. To wind a coil using the manual coil winder, the operator needs quite a bit of finesse and patience. To alleviate this I designed and modified the coil winder to be easier and more automated.

It is a simple modification of the existing machinery to add two stepper motors, one to control the rotation of the coil, and one to aid in the placement of the wire relative to the longitudinal axis of the coil. First I sketched the modifications to the machine in 3d using OpenSCAD (see figure 4.2a), and then I 3d printer the new components and mounted them to the coil winder (see figure 4.2b).

Once all the components were successfully installed, I started working on the controller, which consists of an Arduino Uno¹, a CNC-shield V3², and two DRV8825 stepper motor drivers³. The software for the controller is written in C++ and can be found here⁴, it is controlled via a serial connection to the computer, where a coil winding can be started by typing "go" followed by a configuration number, the following 3 configurations are available:

1. Winds a large coil, it is 25mm tall and has 990 turns over 9 layers.
2. Winds the first half of a small coil in the clockwise direction, it is 3.9mm tall has 187 turns over 11 layers.
3. Winds the second half of a small coil in the counter-clockwise direction, it is also 3.9mm tall and has 187 turns over 11 layers.

Additionally it has a "calib" command which is used to calibrate the linear distance from the number of steps taken by the stepper motor used to control the wire placement relative to the longitudinal axis of the coil.

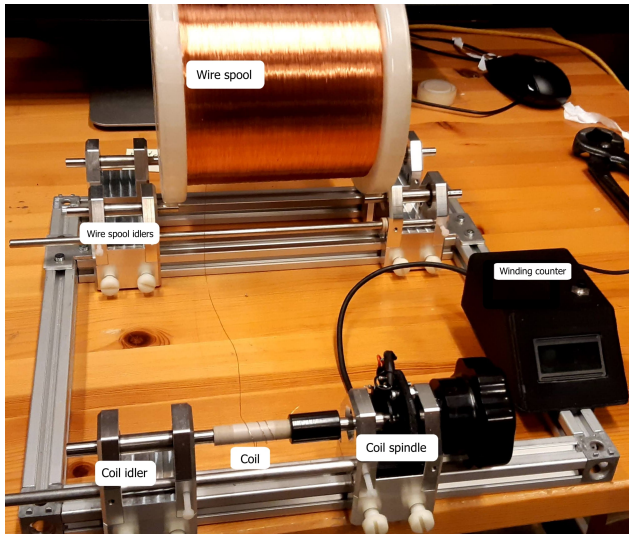
The configurations can be altered by changing some simple variables in the coil winder firmware, and uploading it again using PlatformIO, which is installed on the data management PC.

¹<https://store.arduino.cc/arduino-uno-rev3>

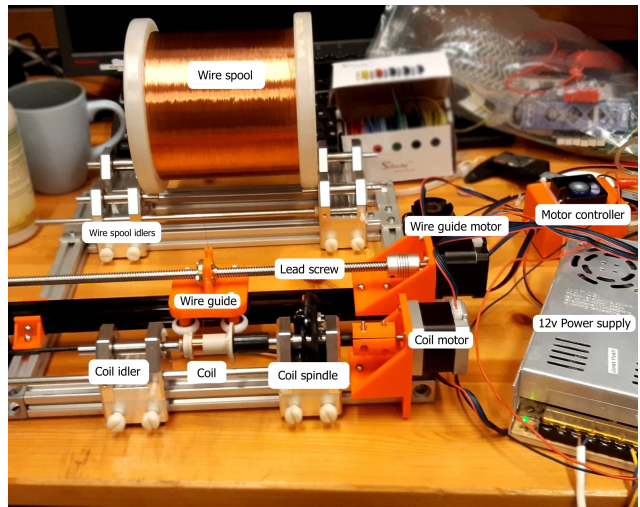
²<https://www.aliexpress.com/item/32829639929.html>

³<https://www.aliexpress.com/item/4000210614636.html>

⁴https://github.com/hyatt03/coil_winder

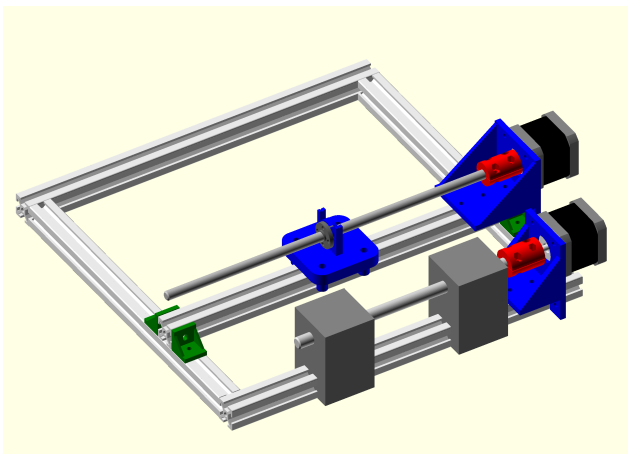


(a) The manual coil winder as it was before modification. In this version the coil would be mounted to the spindle, and the wire would be turned onto the coil manually while the winding counter would keep track of how many turn had been wound.

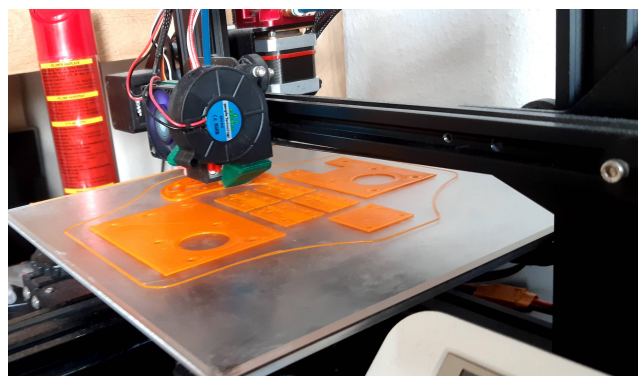


(b) The coil winder after modification. In this version the wire is wound around the coil by a stepper motor while the wire is guided by a second stepper motor. The spindle was modified to accept a motor mount, while the spool- and coil idlers are unmodified.

Figure 4.1: The coil winder before and after modification.



(a) CAD model of the automated coil winder, made using OpenSCAD. Blue, red, and green parts were designed and 3d printed (using orange filament).



(b) 3d printing parts for the automated coil winder using my personal 3d printer.

Figure 4.2: Fabrication process of the automated coil winder.

4.2 New cryogenics PC

When the project started the cryogenics station was running on Windows XP, to secure it from virus and hackers it was disconnected from the network. To resolve this issue and ensure the future of the lab, I selected a new computer that was purchased by the IT department of NBI. I setup the new computer to run the cryogenics instruments by installing the relevant drivers and the newly developed susceptometry software package. With the new computer we can now connect to the network and further automate the lab.

4.3 New oscilloscope

Previously we used the lock-in amplifier to determine the current running through the excitation coil to determine the magnetic field, which meant we either had to determine the current before actual measurements, or we needed two separate lock-in amplifiers (with only one available to the lab), to resolve this I found a new oscilloscope (Tektronix TBS1072B) which was purchased by Gitte Michelsen. Since the system runs at relatively low frequency the primary search goals were that it would be easy to integrate with QCodes, and it was relatively inexpensive. The oscilloscope we settled on had a USB connection so we could control it from the computer, and there was a compatible QCodes driver available in the standard library, making the integration very easy, it is also flexible for future experiment variations, and it solved the issue of measuring current while the experiment is underway, without the need of an extra lock-in amplifier.

Chapter 5

Lab manual

In this chapter I will provide a manual for using the laboratory, first I will give description of each station and the instruments it controls, I will describe how to wind a set of coils for use with the susceptometer, then I describe how to prepare a sample as well as sample loading, I will go through preparing and using the cryostat, and finally I will describe how to use the software I have developed for the lab.

5.1 Experimental setup

The lab is divided into 3 stations, a station is an isolated part of the experiment that communicates with the other stations. Two stations that handle interactions with the lab equipment, and one that handles interactions with the user and the data. All the stations are connected through the network so they can work together efficiently. A detailed diagram over the experimental setup can be found on figure 5.1, and a more thorough explanation of the underlying software can be found in appendix A along with solutions to some common issues.

5.1.1 Magnetism station

The magnetism station handles all instruments which relate to the control and measurements of magnetism. It controls the magnetic field by controlling the superconducting magnets powersupply (seen on figure 5.2), which sets the DC magnetic field, and it controls the function generator (seen on figure 5.3) which sets the AC magnetic field. It measures the current running through the excitation coil using an oscilloscope (seen on figure 5.4), and it measures the results from the signal coil by controlling a lock-in amplifier (seen on figure 5.5). A simplified schematic overview of the measurement circuit can be seen on figure 5.6

5.1.2 Cryogenics station

The cryogenics station controls all things related to temperature, it has control of the gas handling system (seen on figures 5.13 and 5.14) of the dilution refrigerator and monitors the pressures (seen on figure 5.7) of the system through the GHS and a pressure controller. It controls the triple current source (seen on figure 5.8) that heats the resistors inside the dilution refrigerator, and it monitors the temperature of the system by controlling a resistance bridge (seen on figure 5.9) and a digital multimeter (seen on figure 5.10).

5.1.3 Data management station

Finally we have the data management station, here we connect the two other stations together, we send data to the stations to setup new experiment configurations, we relay communications between the stations to indicate the experimental state, and we receive the recorded data during experiment runs. The data management station also runs the servers which the user uses to interact with the system through a GUI, that is explained at the end of this chapter.

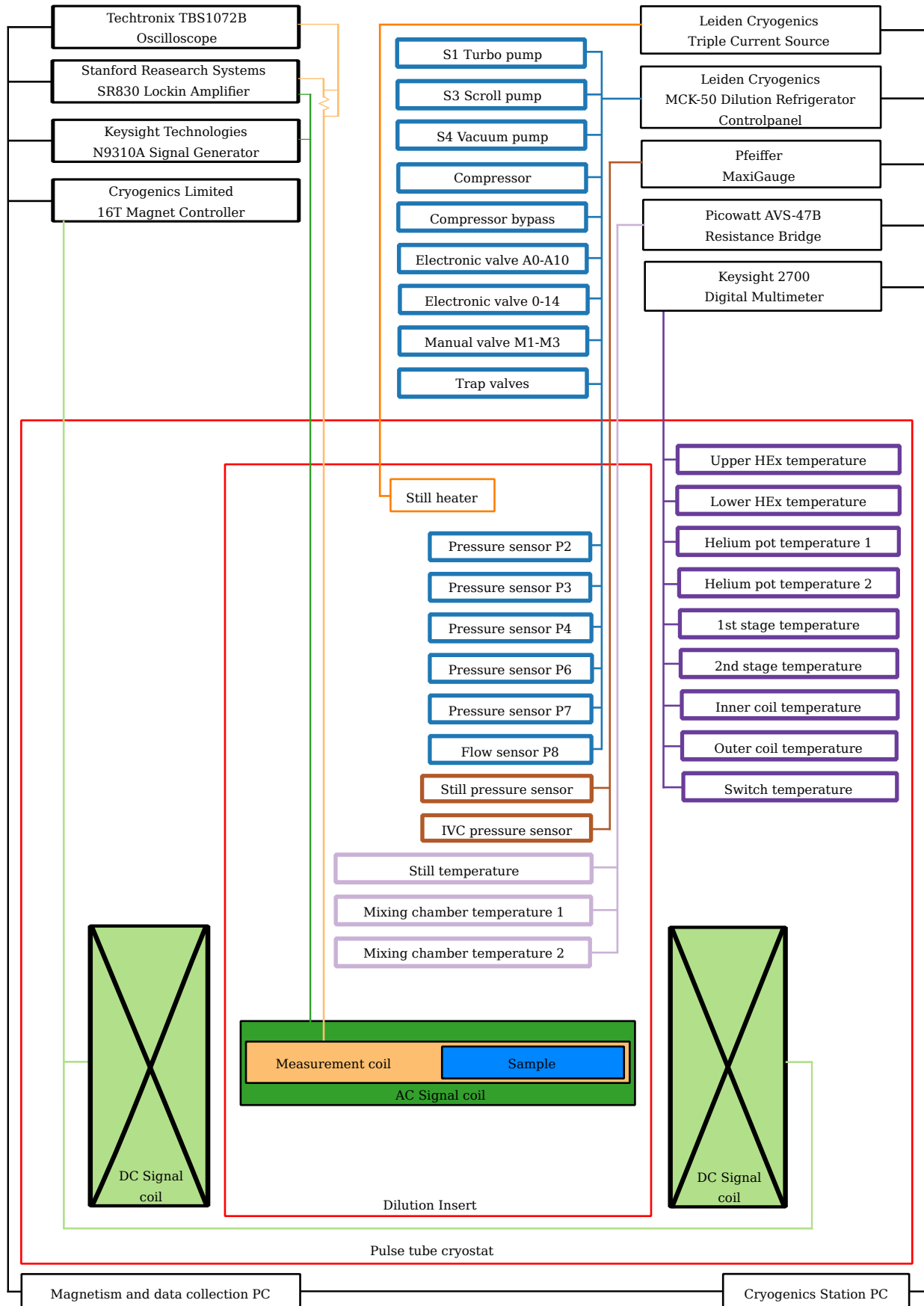


Figure 5.1: A complete overview of how the instruments and sensors are interconnected, the connections are a mixture of analog and digital (see figure 3.1 for more details on the connection types).

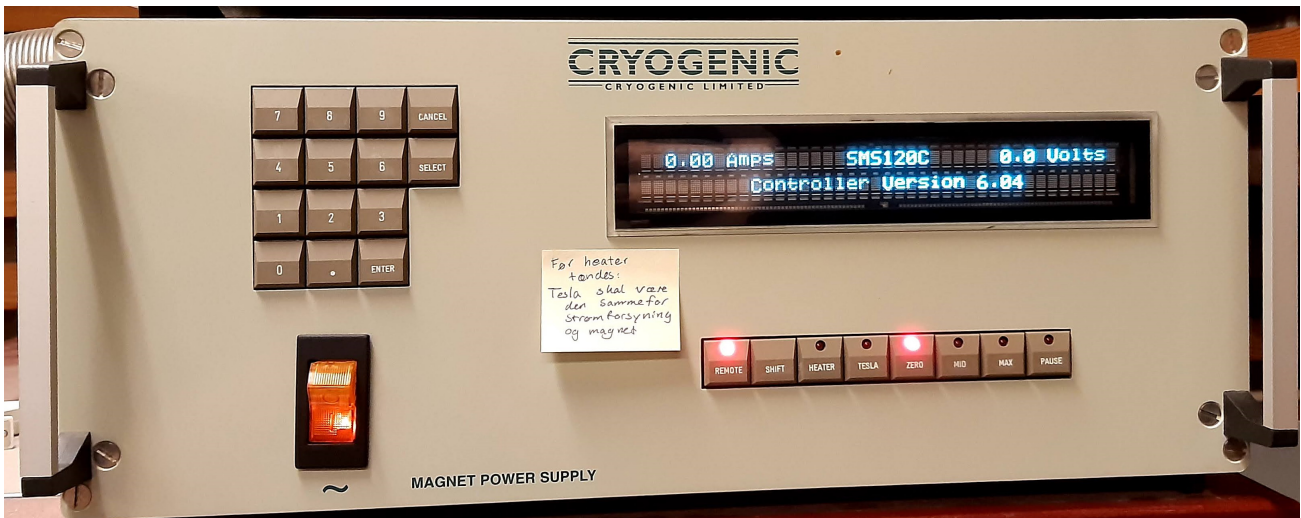


Figure 5.2: The Cryogenic Limited magnet power supply controls the superconducting magnet made from Nb_3Sn and NiTi . The magnet should be kept under 5K when it is in use. The power supply can be controlled using the panel on the front, or via a serial connection to the magnetism station.

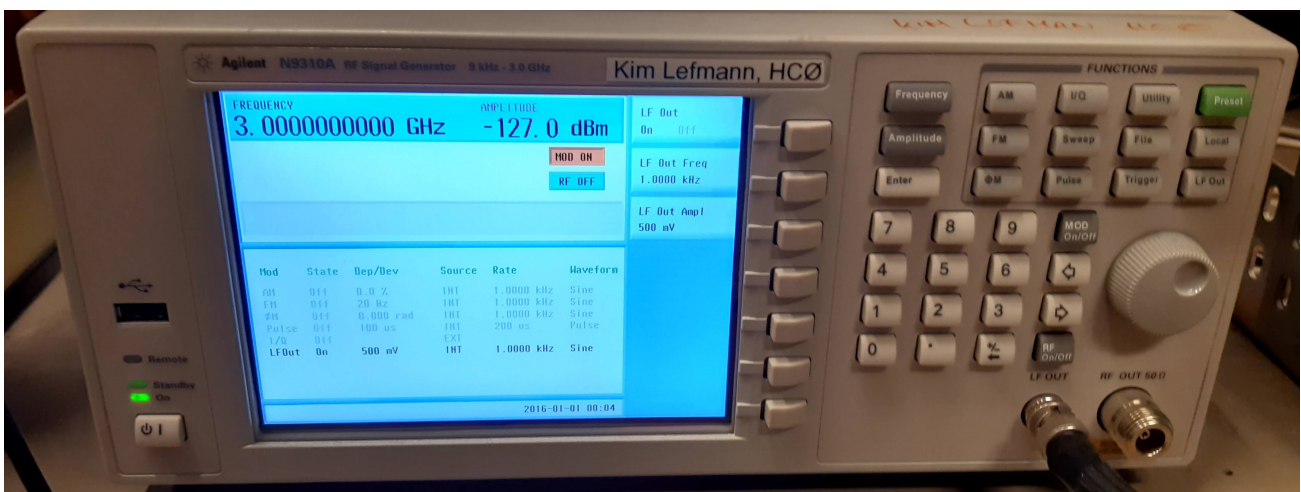


Figure 5.3: The Agilent N9310A function generator is used to control the magnetic field created in the excitation coil, it can be controlled via USB connection to the magnetism station. In the current experimental setup we only use the low frequency (LF) mode.

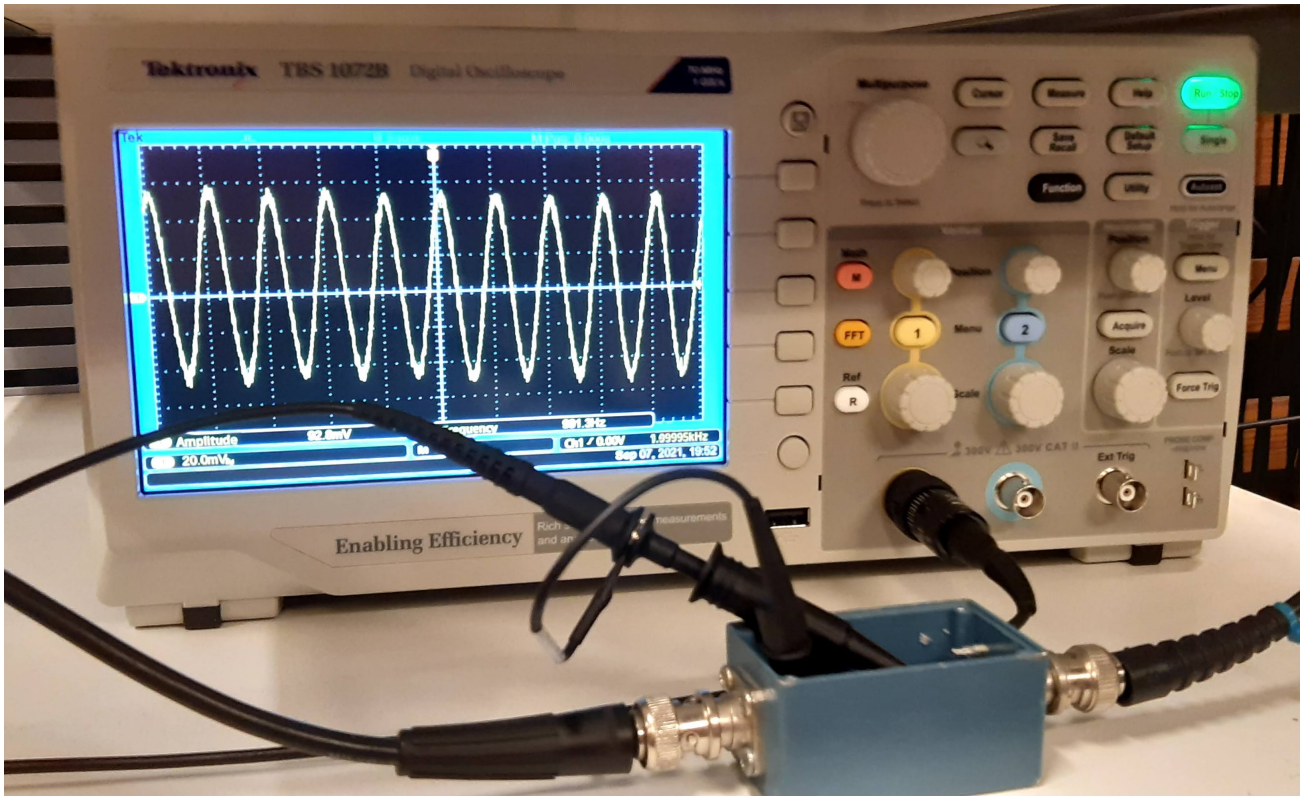


Figure 5.4: The Tektronix TBS 1072B Digital Oscilloscope is used to measure the current flowing through the excitation coil, this is done by measuring the voltage drop over a well known resistor (inside the blue box displayed on the image), it is controlled through a USB connection to the magnetism station.



Figure 5.5: The Stanford Research Systems SR830 DSP Lock-in amplifier is used to measure the complex (ac) voltage induced in the measurement coils, it is controlled through a GPIB interface to the magnetism station.

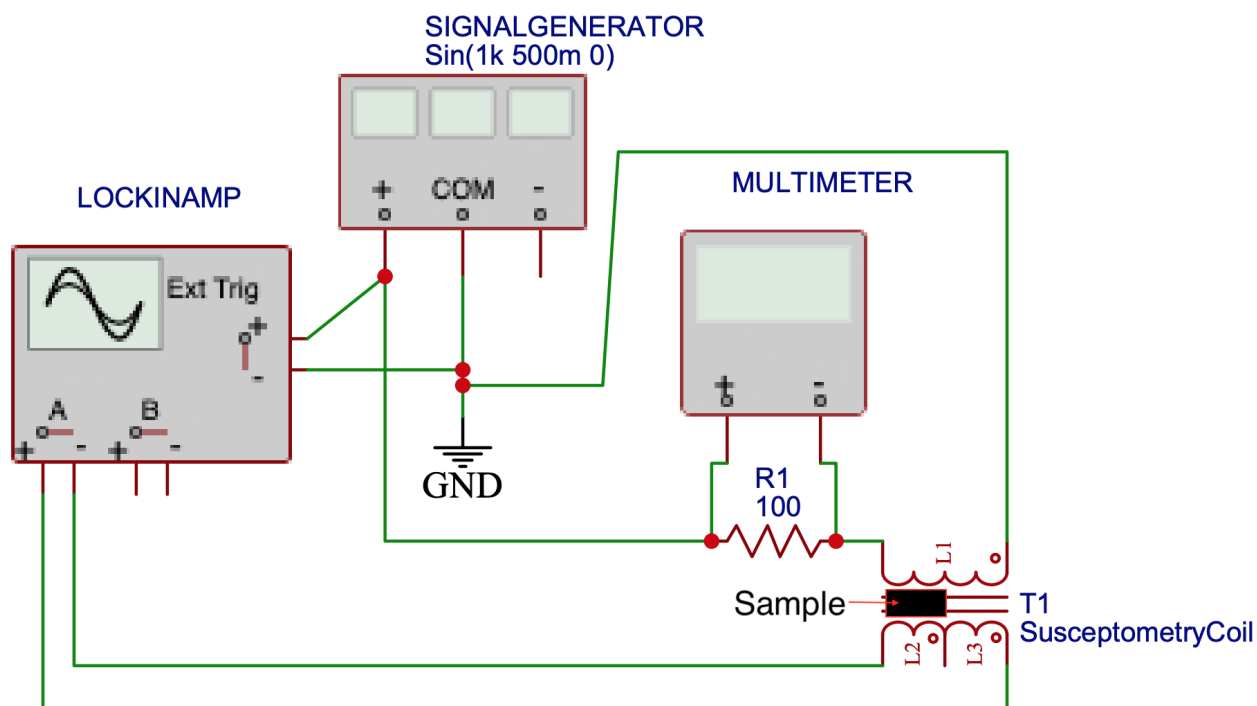


Figure 5.6: Schematic overview of the wired connections of the magnetism station. The LOCKINAMP component represents the Stanford Research Systems SR830 Lock-in amplifier, where the ext. trig. represents the reference signal, and the A input represents the lock-in amplifier input. The SIGNALGENERATOR component represents the Keysight N9310A Signal Generator where the output (+ and COM) represents the low frequency output of the N9310A. The MULTIMETER component represents the Tektronix TBS1072B oscilloscope where the input is channel 1 on the oscilloscope. The resistor R1 is the total resistance in the circuit before the coil, and the mutual inductor T1 is the combined excitation and measurement coils.

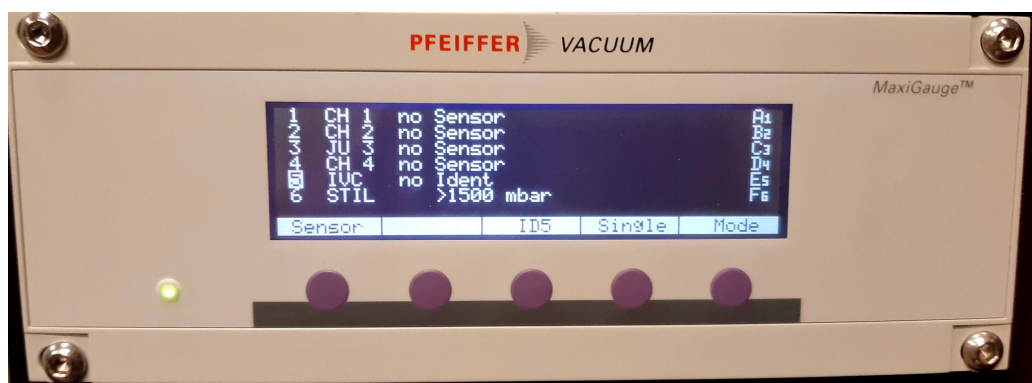


Figure 5.7: The Pfeiffer Vacuum MaxiGauge controls the vacuum gauges mounted inside the dilution refrigerator. It is connected through a serial port to the cryogenics station. It allows us to measure the IVC and Still pressure in the interval $< 5 \cdot 10^{-4}$ to approximately 800 millibars, the sensitivity around 1000 millibars is very high, which is why it shows > 1500 millibars.

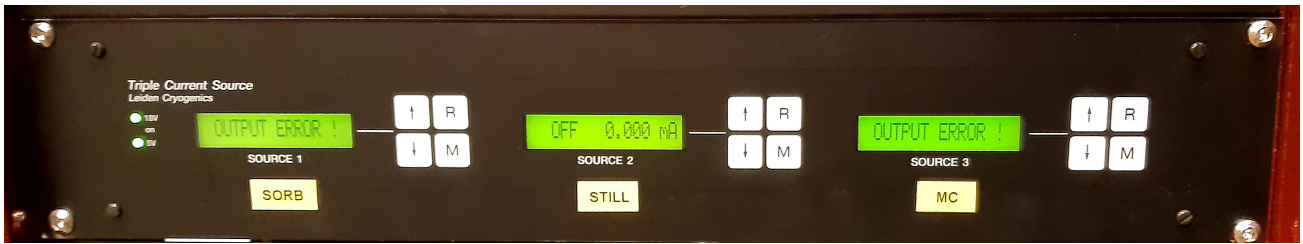


Figure 5.8: The Leiden Cryogenics Triple Current Source controls the heating of a set of resistors mounted inside the dilution refrigerator to enable efficient cooling, currently only one resistor is connected (the resistor in the still, which is the most important one). It is connected to the cryogenics station through a serial port.

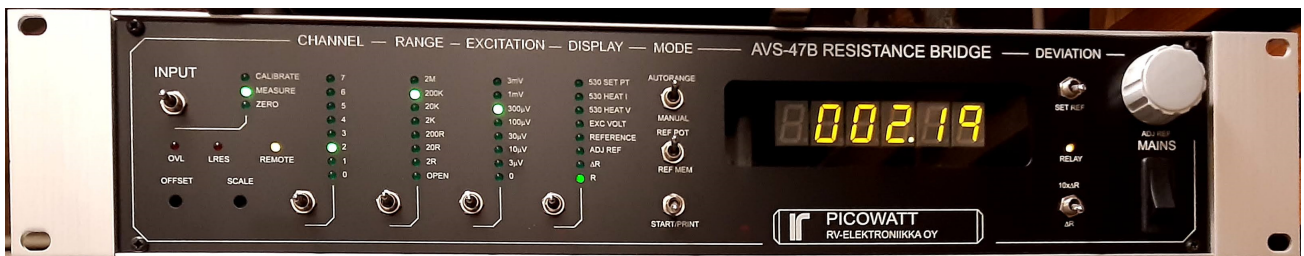


Figure 5.9: The Picowatt AVS-47B Resistance bridge allows us to monitor the temperature of a set of resistors mounted inside the dilution refrigerator by using a very small amount of power. It is controlled by the cryogenics station using a serial port.



Figure 5.10: The Keithley 2700 Digital Multimeter is used to probe a set of 9 thermocouples placed throughout the superconducting magnet and pulse tube compressor, the data allows us to keep track of the cooling process. It is connected to the cryogenics station using a serial port.

5.2 Winding a set of coils

In section 4.1 I described the automated coil winder, in this section I will describe how to use the coil winder. We will use a coaxial susceptometer where the secondary coils are placed end-to-end inside a primary excitation coil based on the work by Piatek[7], Nikolo[6], and Porter[10]. The physical design of the coils was made for a previous thesis by Mathias Mikkelsen[5] in association with the NBI workshop.

Before we can wind a set of coils, we must first decide on the excitation frequency we want to measure around (this depends on the material of interest), and from this we can derive the other characteristics of the coils, in this case we will use 1 kHz as an example.

5.2.1 Determining the optimal number of turns

At the centre of a solenoid, the magnitude of the magnetic field along the coils axis is[4][5]:

$$B = \frac{\mu_0 NI}{l} \quad (5.1)$$

Where μ_0 is the vacuum permeability, N is the number of coil turns, I is the current passing through the coil, and l is the length of the coil. The self inductance of the coil is:

$$L = \frac{\mu_0 N^2 A}{l} = \frac{\mu_0 \pi N^2 a^2}{l} \quad (5.2)$$

Where $A = \pi a^2$ is the cross sectional area of the coil. We use the self inductance to compute the impedance (ignoring the capacitance between the wires):

$$Z = \sqrt{R^2 + \omega^2 L^2} \quad (5.3)$$

Where R is the resistance of the wire material in the coil, and $\omega = 2\pi f$ is the angular excitation frequency. Looking at equation (5.1) we see that the magnetic field is proportional to the current, so if we drive the coil using a constant current source, or if we measure the current driving the coil we can specify the magnetic field independently of the wire thickness and the diameter of the coil, however if we need many turns to achieve the desired magnetic field we may need to reduce the wire thickness to stay within the physical constraints of the system, this increases the resistance of the coil, which in turn increases the required voltage to drive the coil for a given current. The function generators we use to drive the coil have a limited range of both voltage and current, so we need to take the impedance into account to realize the largest possible excitation field[5].

Let's assume a constant driving voltage using Ohms law:

$$I = \frac{U}{Z} \quad (5.4)$$

Where I and U are RMS values of current and voltage respectively, we insert the impedance from equation (5.3):

$$I = \frac{U}{\sqrt{R^2 + \omega^2 L^2}} \quad (5.5)$$

Next we compute the resistance of the wire:

$$R = \frac{l_{wire} \rho}{A_{wire}} = \frac{2\pi a N \rho}{\frac{\pi d^2}{4}} = \frac{8\rho a N}{d^2} \quad (5.6)$$

Where l_{wire} is the length of the wire, A_{wire} is the cross sectional area of the wire, ρ is the resistivity of the wire, a is the radius of the solenoid, and d is the diameter of the wire. We insert equation (5.6) and equation (5.2) into equation (5.5):

$$I = \frac{U}{\sqrt{\left(\frac{8\rho a N}{d^2}\right)^2 + \omega^2 \left(\frac{\mu_0 \pi N^2 a^2}{l}\right)^2}} \quad (5.7)$$

Wire diameter [m]	Optimal number of turns
1	979
$0.2 \cdot 10^{-3}$	979
$0.108 \cdot 10^{-3}$	972
$0.088 \cdot 10^{-3}$	964
$0.072 \cdot 10^{-3}$	946
$0.058 \cdot 10^{-3}$	902
$0.048 \cdot 10^{-3}$	824
$0.039 \cdot 10^{-3}$	677
$0.032 \cdot 10^{-3}$	501

Table 5.1: Optimal number of turns as a function of wire thickness given a current of 60mA RMS, a voltage of 3V RMS, and a frequency of 1kHz. The results are consistent with Mikkelsen[5] and Pobell[9]

Now we insert this expression for the current into equation (5.1) and simplify:

$$B = \frac{\mu_0 N}{l} \frac{U}{\sqrt{\left(\frac{8\rho a N}{d^2}\right)^2 + \omega^2 \left(\frac{\mu_0 \pi N^2 a^2}{l}\right)^2}} \quad (5.8)$$

$$= \frac{U}{\sqrt{\left(\frac{8la\rho}{\mu_0 d^2}\right)^2 + (\pi\omega N a^2)^2}} \quad (5.9)$$

We combine this expression with equation (5.1) to eliminate the magnetic field and setup the equation as a quartic equation in N [5], which we can later solve for a given geometry, voltage, frequency, and current:

$$\frac{\mu_0 N I}{l} = \frac{U}{\sqrt{\left(\frac{8la\rho}{\mu_0 d^2}\right)^2 + (\pi\omega N a^2)^2}} \quad (5.10)$$

$$\Rightarrow 0 = (\pi\omega a^2)^2 N^4 + \left(\frac{8la\rho}{\mu_0 d^2}\right)^2 N^2 - \left(\frac{U l}{\mu_0 I}\right)^2 \quad (5.11)$$

Next we need to settle on a specific geometry, as the coil depends on physical dimensions determined in previous projects[5][11], we know that the inner radius of the primary coil is 7.25mm, and the length is 25mm, these numbers were chosen to lower the variation in field that the measurement coils experience. With the coil dimensions in place, we only need to pick the wire diameter. We can find the diameter by numerically solving equation (5.11) for different possible wire diameters. The solutions are listed in table 5.1, where we see a wire diameter of 0.2mm works with as many turns as the wire diameter of 1 meter, giving us a reasonable choice.

Now we have defined the geometry of the primary, we need to define the geometry of the secondary, as with the primary we can rely on experience from previous projects, where they found the secondaries work well with approximately 220-225 turns each[5][11].

5.2.2 Programming the number of turns

Now that we know the parameters of our coils, we can begin programming the coil winder to use those parameters¹. In the file "src/main.cpp" you can set the variables `wirewidth`, `n_layers`, `n_turns_per_layer`, and `coil_height` under each configuration section to define the parameters used for the coils. If you have problems with the wire drifting between layers you can adjust the variable `mm_per_step` to adjust how far the wire moves when the motor takes a step. Once the parameters are set, use platform io to write the firmware to the arduino and open a serial console (opening the arduino IDE also works well for this purpose). You should see a welcome message, if not try typing help.

5.2.3 Running the coil winder

To mount a secondary, screw the thread at the end of a secondary in to the thread mounted to the motor. To mount a primary, place it over a secondary you have screwed into the coil winder, and thread a bolt over the other end of the secondary to secure the primary into place.

Once the coil is mounted plug in the power supply, and open a serial console to connect to the coil winder.

Place some wire over the wire guide onto a coil, and hold it in place, write "go 1" to wind a primary coil, write "go 2" to wind the first half of a secondary coil, and write "go 3" to wind the second half of a secondary coil. Using those commands should automatically wind a coil after a delay of 3 seconds.

Once the coil is done winding guide the wire through a wire channel (straight channels for the secondaries, and a hole for the primaries), wrap the coil in teflon tape to secure the wire to the spool.

Cut some pins (I used some 26 AWG solid core wire) and solder them to the end of the coil wires, once you are done soldering the coils are ready for calibration.

¹This assumes you are using the firmware developed for the project, which can be found on github: https://github.com/hyatt03/coil_winder. It is a basic platform io project, a guide to setting up an environment (which is already available on the magnetism station) can be found here: <https://docs.platformio.org/en/latest/core/quickstart.html>

5.3 Preparing a sample

In this section I will describe how to prepare and load a sample into the coils we just created. I assume you have a powder sample which we can cast into a pill form that fits into the sample chamber.

To cast the sample we need an epoxy "Stycast W19" combined with "Catalyst 9" as it has good properties when cooled to cryogenic temperatures, and we need mold. Presently we have molds made from polytetrafluoroetylen (Also known as PTFE or Teflon) that create a cylinder with the correct diameter.

The first step is mixing the epoxy, it is important that it isn't too viscous so it mixes well with the powder sample, then we mix in as much sample as possible², then we fill the mixture into the mold, add a wire through the pill for thermal conductivity³, and wait for the epoxy to set. Once the cast is set we can press out the pill from the mold. If the pill breaks when you press out the sample from the mold, you can try to use less catalyst (resulting in a slightly more viscous epoxy) or you can mix in less sample powder.

Assemble the coils using a set of excitation and measurement coils mounted inside the holder, mount the assembly to a fork, and mount the fork to the dilution insert (see figure 5.11). Now we want to connect the coils to the electrical lines, use a multimeter set to continuity mode to figure out where the DC lines are connected to breakout box mounted to the insert⁴.

Test the electrical connection by connecting the coils to the signal generator and the lock-in amplifier, run a test signal and adjust the hex key to check if the lock-in amplifier responds to the changes, remember to only push the hex key half way through the coil, otherwise the two measurement coils cancel the generated signal.

Once you are satisfied that the signal is available, you can remove the hex key and insert the actual sample. Position the sample with a set of calipers using the depth gauge and wrap the thermal conductivity wire around the cold finger. Fix any loose wiring to the sample area using teflon tape (regular tape may result in degassing under low pressure which can make it difficult to achieve a vacuum).

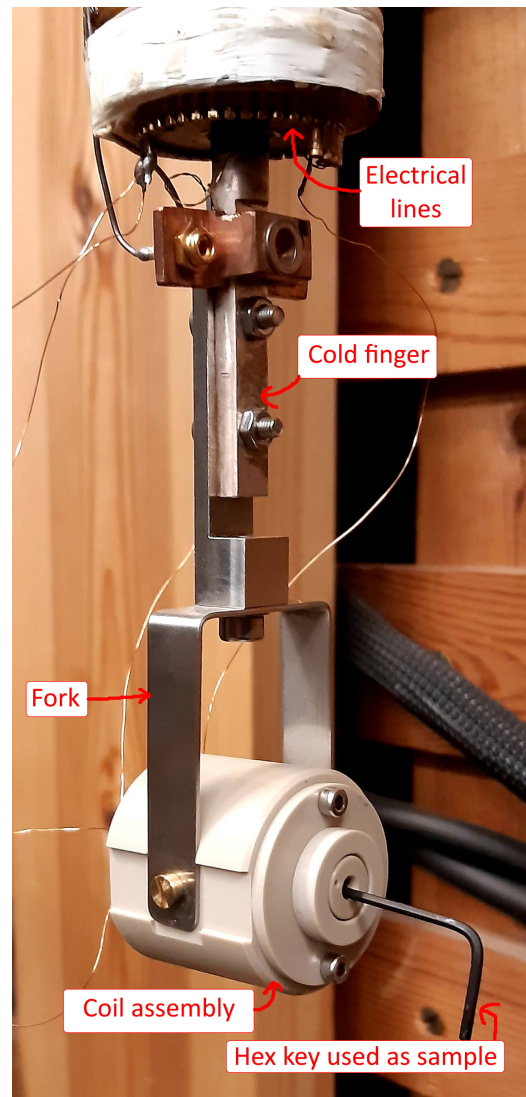


Figure 5.11: A image of the coils mounted to the dilution insert with dc lines attached to measure a signal from a hex key used as a control sample.

²This may require a few attempts, as the cast sample may break when it is released from the mold

³Make sure the wire is long enough to reach the cold finger from both sides, the wire needs to run the entire length of the coils to ensure the added susceptibility is cancelled by the measurement coils.

⁴Only the center conductor is connected to the DC lines, the outer conductors all share ground.

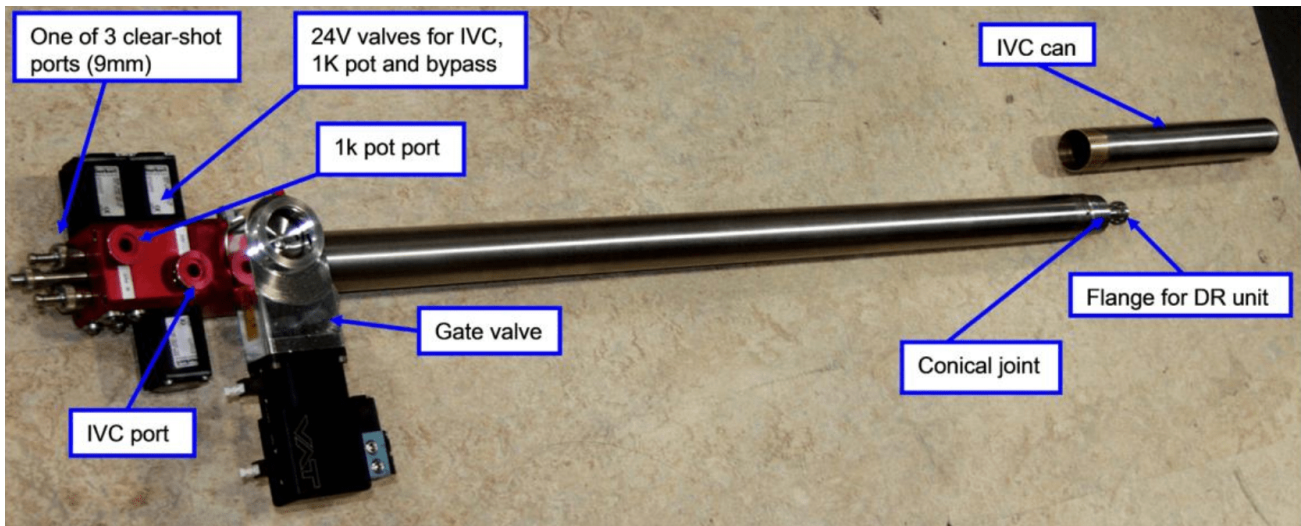


Figure 5.12: A image of the dilution refrigerator insert with important components labeled. Source: Leiden Cryogenics[2].

5.4 Using the cryostat

In this section I will walk through the process of cooling the experiment using a combination of the Cryomech PT415 pulse tube compressor[3] and the Leiden Cryogenics MCK50-100 Dilution refrigerator[2] (seen on figure 5.12). It will be largely based on oral introductions I received from Mathias Mikkelsen combined with the instructions given in the operating guide. This walk through has the notable caveat that I was not able to cool down to the target 15mK during the experiment runs (likely due to a leaky valve), so the last part of the cooldown guide is solely based on the operating guide.

5.4.1 Understanding the controls

The experimental setup used to achieve cryogenic temperatures is complex and before we begin using it we must understand how everything is connected, so we know what to do if something goes wrong. An illustration of the Gas Handling Systems (the GHS) frontend can be seen on figure 5.13.

The GHS is split into two separate circuits, the yellow circuit controls atmospheric gas, and the green circuit controls the helium mixture (containing both He^3 and He^4).

The buttons labeled A0 through A10 represent valves controlling the atmospheric circuit, the buttons labeled with numbers 0 to 14 represent valves controlling the helium circuit, the markings M1, M2, and M3 represent manual valves which can be seen on figure 5.15. The valves A9 and A10 control the flow between the two circuits, so when we activate these valves we must be extra careful to avoid losing the helium mix. The valve marked "Gate valve" is a large valve on the cryostat end of the large vacuum hose, the last valve in the circuit is called "Bypass compressor"⁵ and is used to open a line between P2 and P8, both of these valves are activated by the labeled button in the cluster of buttons in the top right corner.

S4 represents the vacuum pump which is connected to the atmospheric circuit on the frontend, and vents to atmosphere on the backend, it is used for pumping a vacuum where a normal vacuum is needed, and it is used as a backing pump, it can be routed to the manual valve M3 which leads to a hose that can be attached at many points, we use this pump to ensure a vacuum in the superconductors vacuum chamber, in the cryostat sleeve, in the 1K pot, and when we need to evacuate atmospheric gasses and impurities from the helium circuit. S3 represents the scroll pump which is used to move helium throughout the helium circuit, and as a backing pump. S1 represents the turbopump, this is used to pump gas at very low pressure where traditional pumps no longer

⁵This valve has previously been sticky, meaning it wouldn't open or close as expected.

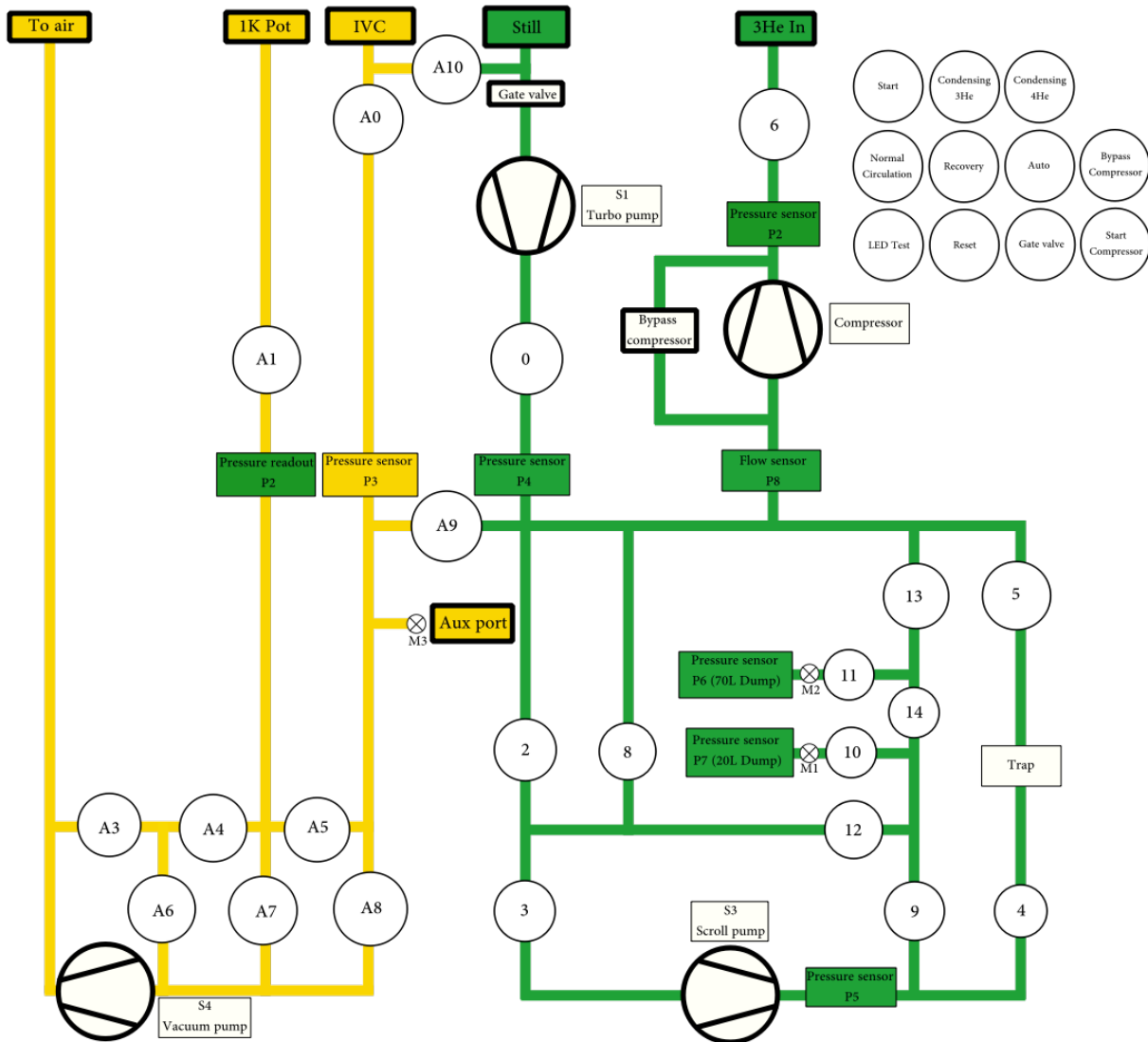



Figure 5.13: An overview of the dilution refrigerator frontpanel. This graphical representation differs from the physical presentation as I have excluded lines, valves and sensors which are not connected, and I have included the actual position of the pressure sensor P2 as well as the readout. This representation should be more accurate than the representation seen on the physical panel (which can be seen on figure 5.14). The yellow circuit is where atmospheric gasses circulate, while the green circuit is where the helium mix is circulated, the graphical split between the two visualizes when we need to be extra cautious (that is when we open the valves that connect the two circuits A9 and A10). The circles marked A0 to A10 are electronic valves that operate on the atmospheric circuit, while the circles marked 0 to 14 are electronic valves that operate in the helium circuit, and the circles with crosses over marked M1-M3 are manual valves that need to be turned physically. The pumps are visualized by this icon  which shows the pumping direction (gas goes from the wide part (the backend), to the narrow part (the frontend) of the converging lines).

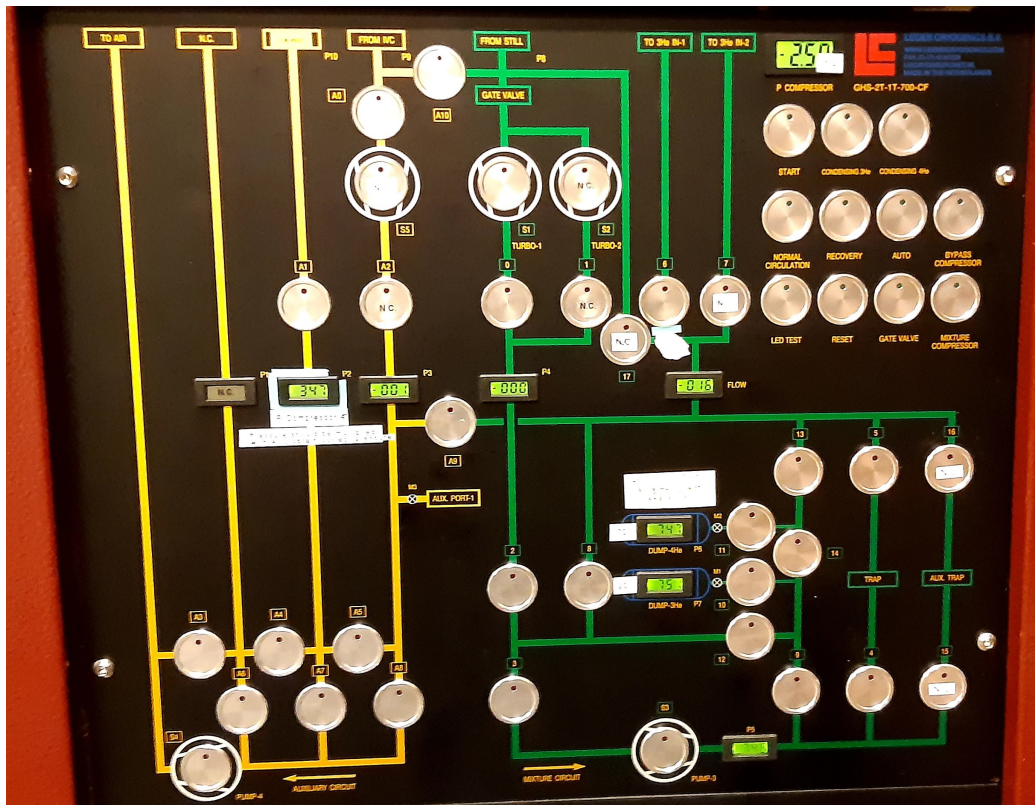


Figure 5.14: An image showing the frontpanel of the Gas Handling System (the GHS), which is illustrated in figure 5.13. The GHS is controlled by a serial connection to the cryogenics station, here we can open and close valves, we can read off pressures, and we can activate the connected pumps, there are also a number of manual valves which prevent us from automating this completely, but also reduces the likelihood of errors leading to loss of gas.

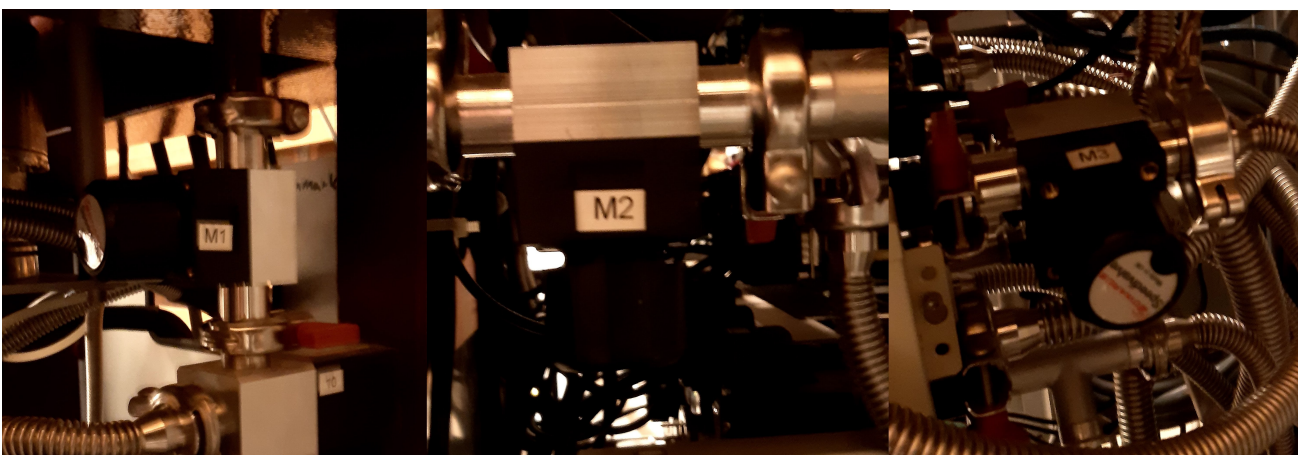


Figure 5.15: An image showing the three manual valves, M1, M2, and M3, that are attached inside the GHS. M1 and M2 control the mix dumps, and M3 is connected to an external hose that we use to create a vacuum where we need it, and to measure pressures (using P3) where we need it.

works, it requires a backing pump that ensures the pressure on the backend remains low (either S4 when we want to discard the gas, or S3 when running in normal operation).

The gauges marked P2 to P8⁶ represent pressure sensors, on the front panel they are read out in millibar⁷, these are vital to keeping track of the pressures in the system, they are also plotted over time in the GUI. It is generally a good idea to keep the pressures below 1000 millibar (approximately one atmosphere) this ensures any leaks lead to gas entering the system rather than leaking helium mix to the atmosphere. The pressure sensors have a built-in alarm which alerts when the pressure exceeds approximately 1600 millibars (the maximum range of the sensors), at this point a loud beeping starts and a lot of LEDs start blinking, if this happens you need to let the gas go somewhere (preferably somewhere with a large volume, such as one of the dumps), once the pressure drops to an acceptable level you can press the reset button to disable the alarm, you may have to press it more than once.

In the button cluster there are a number of options for controlling the cryostat automatically, they should not be used as the automatic profiles have never worked properly with this cryostat.

The GHS is externally connected at a few points, as discussed it is connected to the atmosphere through S4, and can be connected to many points using M3. It is connected to the 1K Pot (the chamber around the sample) through A1, it is connected to the inner vacuum chamber (IVC) through A0 and A10, it is connected to the still (using a large hose) through A10 and the Gate Valve, it is connected to the helium port through valve 6, and finally is connected to the trap through valves 4 and 5⁸.

5.4.2 Preparing the cryostat

In this section I will walk through the steps required to prepare the cryostat for a cooldown, I assume the system has been out of operation for a while and is at room temperature.

1. First we check if the mix is in the dumps, and the dumps are closed.
 - (a) Check the pressures of P6 and P7, they can vary a little depending on how the mix was channeled to the dumps. If the pressures of both dumps are between 500 and 1200 it should be good⁹. If the mix is somewhere else in the system, you can follow the guide in section 5.4.5 on recovering the mix.
 - (b) Check the valves to the dumps are closed. M1 and M2 should be tight, though not overtight, with no red band visible, and valves 10 and 11 should not be lit on the frontpanel.
2. We now need to remove the insert, this is a two person job.
 - (a) Close valve M3.
 - (b) On the end of the hose connected to M3 are two other valves (see figure 5.17), ensure both are closed.
 - (c) Remove the hose connected to the valve marked SV (see figure 5.17).
 - (d) Slowly open the valve marked SV.
 - (e) Ensure the valve at the end of the hose marked "To cryostat sleeve" is also open.
 - (f) The sleeve is now connected to atmosphere, which means we can pull in air to displace the volume removed by the insert.
 - (g) Now we want to ensure the 1K pot does not fall off, it is held on by a vacuum, so we just have to ensure the vacuum is strong enough (less than 20 millibars). Start the vacuum pump by pressing S4 and wait for the noise to become consistent.

⁶Where P8 represents a flow sensor.

⁷Except P2 which displays $\frac{1}{4}$ of the pressure in millibar on the backend of the compressor

⁸There are also manual valves on the trap, they can be seen on figure 5.16.

⁹The system can have both dumps at a similar level (perhaps around 700-750), or have one dump containing more than the other (perhaps one has around 1000 and one has around 500)



Figure 5.16: An image showing the manual valves that control flow through the trap, which filters impurities out of the helium mix. These need to be turned to open, and must be opened for helium mix to flow through the system.

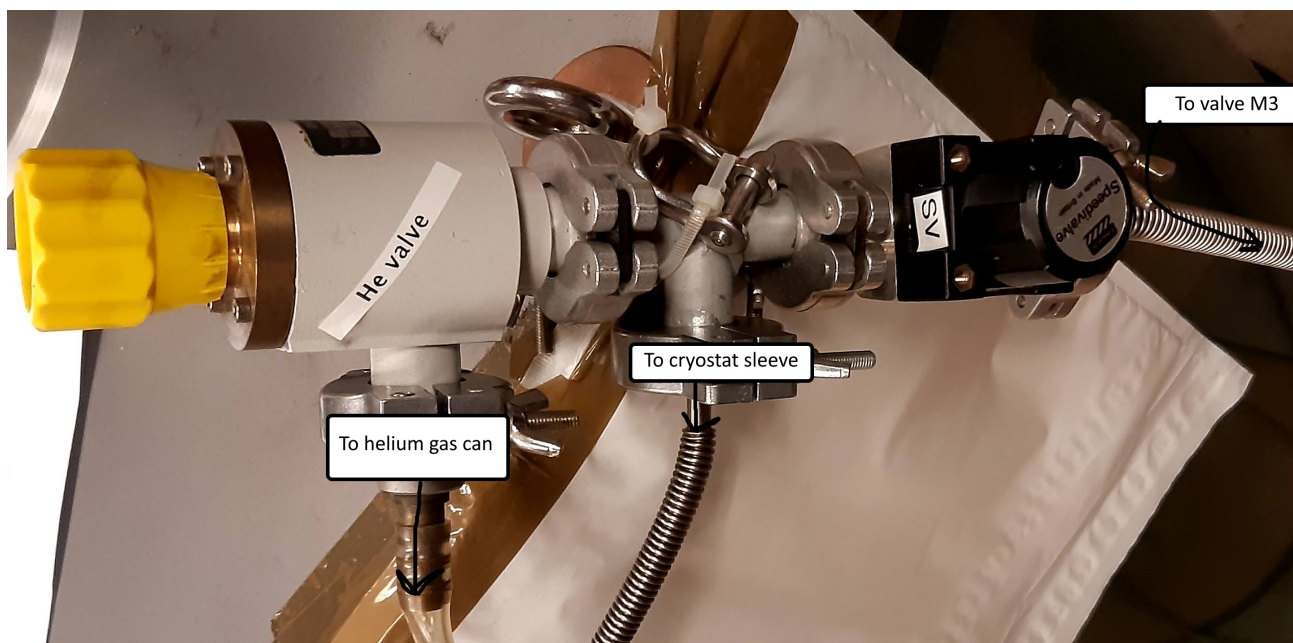


Figure 5.17: An image showing the valves used for managing the atmosphere inside the cryostat sleeve. There is one more manual valve at the end of the hose marked "To cryostat sleeve". The SV valve is connected to M3, allowing us to create vacuum or measure the pressure of the exchange gas.

- (h) Open valves A7 and A1 and wait for a vacuum to form¹⁰.
 - (i) Open the insert holder mounted to the wall at the end of the crane tracks (see figure 5.18).
 - (j) With one person stabilizing the insert, and one person operating the crane, remove the insert from the cryostat, and be careful not to rub the cranes chains against any of the cables or hoses as they are very easy to damage.
 - (k) Secure the insert to the wall.
3. Close the valve sealing the chamber (see figure 5.19).
 4. Close all the other open valves (including A7 and A1), and turn off S4 if it is running.
 5. Now remove the IVC can (see figure 5.12), this is a two person job.
 - (a) You must be very careful with the IVC can, as the conical joint at the end is very precisely ground and even a small bump can result in leaks.
 - (b) One person should hold the IVC can, while the other opens A3, A4, and A1 to let atmosphere into the 1K Pot.
 - (c) The IVC can should drop off, or be easily removed now. Place it in the foam holder to avoid damaging it while setting up the rest of the system.
 6. Mount and wire your experiment as described in section 5.3.
 7. If you wish to leave the insert mounted to the wall for a while, it can be a good idea to pump out both vacuum chambers in the cryostat to avoid any issues with condensation.
 8. Now that we have mounted our sample, we can begin re-inserting the insert, this is also a two person job.

¹⁰If you are uncertain, you can also open A5 and check the sensor P3 for a low pressure.



Figure 5.18: An image showing the crane used for moving the insert in and out of the cryostat.

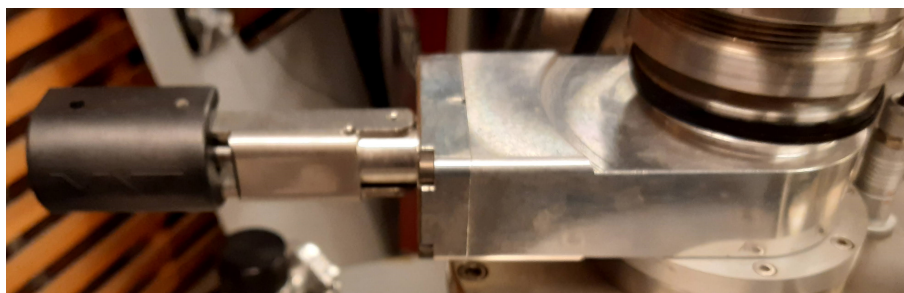


Figure 5.19: An image showing the valve sealing the sleeve. We can use this valve to seal off the sleeve when the insert has been taken out.

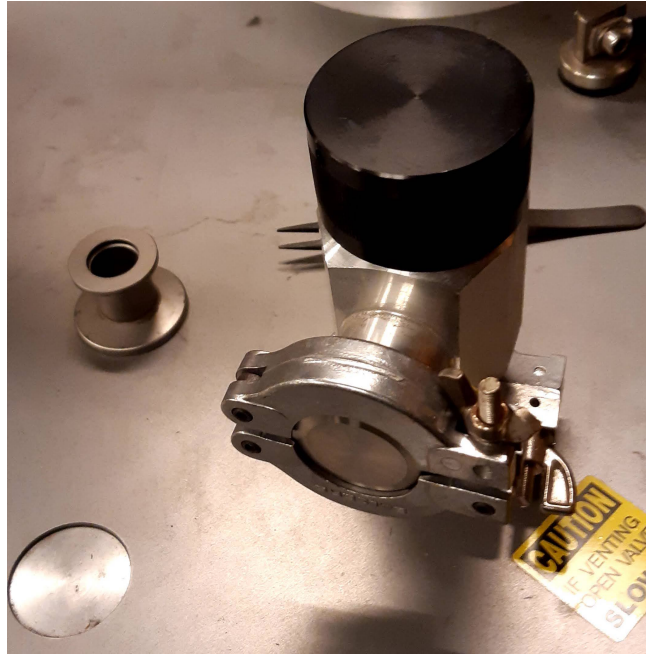


Figure 5.20: An image showing the valve sealing the superconducting magnet. We use this valve to ensure a vacuum in the chamber containing the superconducting magnets to prevent moisture (and ice) damaging the coils.

- (a) We begin by mounting the IVC can to the insert, first we clean the conical joints of any dust, debris, and old vacuum grease using ethanol, then we grease the conical joints with fresh vacuum grease (a thin even layer is optimal), one person carefully slides the IVC can over the sample area until the two conical joints mate, then the other person begins pumping a vacuum by starting S4 and opening A7 and A1, you should feel the joint getting sucked tight, allow the vacuum to build until you can confirm a constant tone from the vacuum pump.
 - (b) Open the manual valves to the sleeve as we did just previously, and open the valve sealing the sleeve.
 - (c) One person holds the insert while the other person unhooks it from the wall mount.
 - (d) One person operates the crane, while the other person guides the insert over to the sleeve, where it is slowly lowered into until the strings holding the insert become a bit slack.
9. Now we want to pump out the sleeve until we have a vacuum, ensure the manual valve M3 is connected through a vacuum hose to the manual valve marked SV (see figure 5.19). Close any open ports, start pump S4 and open A8, M3, SV, and the small valve connected directly to the sleeve, once P3 shows only a few millibar we can close A8, SV, M3, and turn off S4.
 10. Connect M3 to the manual valve of the superconducting magnet (see figure 5.20). Pump out any gas that might be by starting S4 and opening A8, M3, and the manual valve to the superconducting magnet. The pressure should read 0 or -1 millibars on P3. Close A8 and the manual valve to the superconducting magnet. Connect the hose to the valve marked SV.
 11. Next we want to clean out the trap (see figure 5.21). This is not necessary every time you operate the cryostat, but it does improve the lifetime and quality of the cooling.
 - (a) Check that the dumps contain the material you expect, the manual valves M1 and M2 are closed, and valves 9, 10, 11, 13, and 14 are closed.
 - (b) Start pump S4, open valves A8, A9, 4, and 5. Open the manual valves on the trap (see figure 5.16).



Figure 5.21: An image showing the trap in full size (though rotated), encircled is the area containing the activated charcoal, which can be cleaned of impurities by heating with a heat gun.

- (c) Begin heating the area indicated on figure 5.21 with a heat gun, ensure the entire filter area is heated evenly. The filter should be hot to the touch before continuing.
 - (d) Let the trap cool down while the pump is running, once the trap is at room temperature you can close the valves A8, A9, 4, and 5 again, and turn off S4.
12. Ensure the watercooling for the turbopump is running by opening the valve behind the large red cabinet containing the pumps and control equipment (see figure 5.22).

5.4.3 Cooling to approximately 4K

In this section we go through the procedure required to cool to approximately 4K, where helium condenses to a liquid.

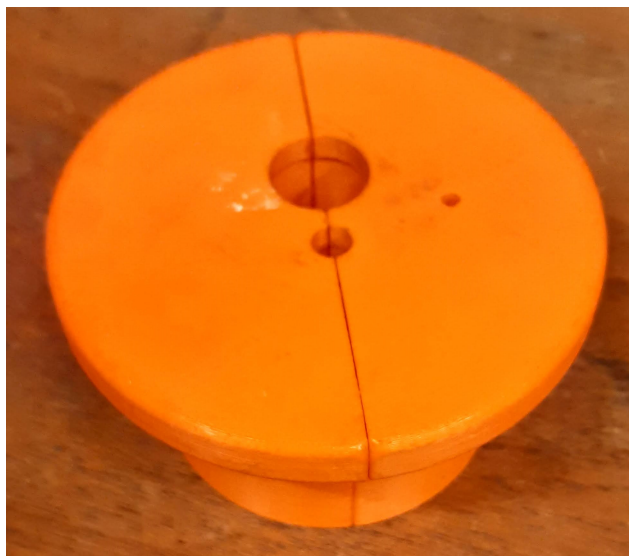
1. First we need to add exchange gas.
 - (a) With all valves on the control panel closed, open manual valves M3 and the valve marked SV (see figure 5.17), and ensure the small valve leading to the sleeve is also open.
 - (b) If there is any pressure showing on P3, remove the gas with S4 through A8.
 - (c) Add a small amount of exchange gas, it is important to be careful here as the helium cylinder is at a high pressure. Close the helium valve shown on figure 5.17, close all the valves on the helium cylinder. Open the first valve on the helium cylinder, wait for pressure to build (it should be very quick, less than a second), and close it again. open the next valve to pressurize the hose to the helium valve, close the valve once you see the hose is pressurized, slowly open the final helium valve until you see P3 change and be ready to close it quickly (to avoid a large pressure), pressurize to about 100 millibars, if you let too much exchange gas through you can remove the gas with S4.
 - (d) Close M3 and remove any remaining gas for the P3 using S4.
2. Start pumping out the inner vacuum chamber (IVC) and still (this takes several hours, and can happen while the system does the initial cooldown).
 - (a) If you have used the cryostat recently, you should use S3 as a backing pump, otherwise use S4 to clean out any contaminants.
 - (b) If using S4, start S4, open valves A8 and A9.
 - (c) If using S3, open valves M1, 10, 9, start S3, open valves 3, 2.
 - (d) Open valves 0, Gate valve, and A10.
 - (e) Once the IVC and Still pressures drop to about 80 millibars it is safe start the turbopump S1 (with a backing pump running).
3. Turn on the CP1000 pulse tube compressor. The initial cooldown takes about 48 hours.



Figure 5.22: An image showing the valve controlling waterflow to the turbopump, it is open in this case. When the turbopump is running it needs watercooling, but when the setup is idle we can turn off the waterflow to minimize the likelihood of contamination or a leak.



(a) An image showing the dewar that holds the liquid nitrogen used to cool the trap.



(b) The 3d printed cap that holds the trap in the dewar and prevents excessive evaporation by thermal insulation.

Figure 5.23

4. While the system is cooling, you can check if there is enough exchange gas by opening from the sleeve through to P3, if it goes below 50 millibars you should add more until you have about 100 millibars.

You can follow the progress of the cooling by opening the web interface and selecting "Save temperatures continually" (see section 5.5). The cooldown should look similar to the temperature graph on figure 5.26.

5.4.4 Cooling below 4K

Once the system is below 4K we can begin running the dilution refrigerator.

1. The first step is to cool the trap, fill the dewar with liquid nitrogen (see figure 5.23a), and lower the trap into the dewar until it hangs just above the bottom, hold it in place using the dewar cap (see figure 5.23b).
2. While the system is running you need to refill it with liquid nitrogen every 2-3 days, there is a smaller red dewar you can use to move the nitrogen from the filling station to the lab.
3. Pump out the IVC and still using S1 until the pressure reads $< 5 \cdot 10^{-4}$ millibars. If the IVC pressure increases during the cooldown, it will cause problems.
4. Open valve 6. The pressure on P2 should be about 0, if it isn't you can either pump it into the dump using S3, or vent it to atmosphere using S4.
 - (a) To vent it to atmosphere open the valve marked "bypass compressor", turn on S4, open valves A9 and A8, and wait for the pressure to drop to about 0.
 - (b) Close any open valves and turn off S4.
 - (c) To move the atmosphere to the dump open the valve marked "bypass compressor", open valves 8, 3, 9, 10, and M1, and turn on S3.

- (d) Close any open valves and turn off S3.
5. The next steps should be done with supervision from someone experienced with the cryostat.
 6. Close all valves and turn off any running pumps.
 7. Now we prepare the compressor to run by increasing the pressure behind it to between 900 and 1000 millibars (as read on sensor P5).
 - (a) Open valves 4, 5, and the manual valves on the trap (see figure 5.16).
 - (b) Now it is important to monitor pressure sensor P5, if the pressure becomes too large (over 1200 millibars) you should close valve 5 and 12, and open valve 8, 3, 9, 10 and M1, be prepared to react quickly.
 - (c) Open valves M1 and 3.
 - (d) Alternate between opening 10 and 12 (with only one open at a time) to add the mix in small batches until the pressure behind the compressor is large enough¹¹.
 8. Now we turn on the compressor¹².
 - (a) Close valve 9, open the gate valve and valves 0, 2, 3, 6¹³. Ensure S1 and S3 are running, wait until P5 is stable.
 - (b) Now we are ready to start the compressor, you should be ready to remove some of the mix as soon as you hear the compressor start (it has a deep hum). You can remove the mix by opening the valve "bypass compressor" (this is the most time-critical step), opening valve 8, and 9, and closing valve 4, the mix should go into the 20L dump.
 - (c) Once P5 is around 100 millibars you can reopen the flow by closing valve 8 and 9, and opening valve 4 (before the pressure on P5 gets too high).
 9. Now the compressor should be running, "bypass compressor" should be open, "gate valve" should be open, valves 0, 2, 3, 4, 5, and 6 should be open, valve 8 and 9 should be closed.
 10. Close the valve "bypass compressor".
 11. Helium is now circulating, and, given sufficient exchange gas, should be condensing slowly. You can follow the progress on pressure sensor P2 (keep in mind that it shows one quarter of the actual pressure).
 12. When P2 has been stable for a while, you can add more mix by alternating between valves 9 and 10.
 13. Keep adding mix until the 20L dump is empty¹⁴.
 14. The temperatures inside the dilution refrigerator should be falling to below 1K, and cooling your sample with it, but in order to optimize the cooling power, we need to add more mix.
 15. Close manual valve M1 and valve 10. Open manual valve M2, and valves 11 and 14.
 16. Keep adding mix in the same way as for the small dump (be careful to begin with as the pressure is higher) until the mixture is saturated¹⁵

¹¹If the pressure on P2 raises, then the "bypass compressor" valve is still open, this has been an issue before.

¹²You should not leave the system while the compressor is running, if the pressure spikes you need to react and redirect the gas into the dump (using S3 and the bypass compressor valve).

¹³Now all valves marked with A should be closed. Valves 8, 9, 11, 13, 14 should be closed. Valves "gate valve", 0, 2, 3, 4, 5, 6, 10, and M1 should be open.

¹⁴When the pressure in the 20L dump is low enough (on the order of 10 millibars), you will need to start alternating between valves 10 and 12 instead of 9 and 10 to use the scroll pump to remove the remaining mix from the dump.

¹⁵The mixture is saturated when an increase in pressure does not result in mixture condensing.

17. When the enough mix is condensed it is time to open the bypass valve and switch off the compressor, wait for the system to reach a steady state, you may need to remove some of the mix due to the lowered pressure while the cycle is running. Once it is running well, close the valves leading into the dumps.
18. Switch on the triple current source (TCS, see figure 5.8). Set source 2 to 10 mA for the fastest cooling, or ≈ 4 mA for the lowest temperature¹⁶.

The cryostat should now be actively cooling, and with sufficient thermal conductivity your sample should be too. Wait for the system to reach a steady state, be sure to check the exchange gas pressure, and the level of liquid nitrogen regularly throughout your experiment.

It is now time to run your experiment, and collect some data.

¹⁶The pressure and current need tuning to reach the lowest temperature.

5.4.5 Reheating the system

Once the experiment is done, we need to reheat the system while capturing the helium gasses in the dumps.

1. Open valves 9, 10, 11, 14, and manual valves M1 and M2.
2. Close valve 4 and open valve 8.
3. Ensure pumps S1 and S3 are running.
4. Turn TCS source 2 and 3 to 20 mA.
5. Wait for the still pressure to reach $< 5 \cdot 10^{-4}$ millibars.
6. Close M1, M2. The helium is now recovered.
7. Close all the remaining valves and switch off all the pumps.
8. Turn off the pulse tube compressor, and wait for the system to reach room temperature.
9. Once the system has reached room temperature, add in exchange gas until you have approximately 1000 millibars (use P3 to monitor the level), be sure to add the gas slowly.
10. The insert can now be removed in the same manner as in section 5.4.2.

5.5 Running the software

In this section I will describe how to use the lab software to run an experiment from start to finish. A more thorough description of the software architecture and debugging procedures can be found in appendix A.

5.5.1 Starting the system

For the magnetism station:

- Cryogenics Limited 16T Magnet controller (flip the switch on the front)
- Stanford Research Systems SR830 Lock-in amplifier (flip the switch on the back right side)
- Keysight N9310A Signal Generator (press the button on the front)
- Tektronix TBS1072B Oscilloscope (click the button on the top left side)

For the cryogenics station:

- Picowatt AVS-47B Resistance bridge (flip the switch on the front)
- Keysight 2700 DMM (press the button on the front)
- Leiden Cryogenics Tipple Current Source
- Leiden Cryogenics Gas Handling System
- Pfeiffer Vacuum MaxiGauge

The next step is starting up the webserver, to help with this a batch script called start server is distributed along with the source files, it is linked on the desktop of the magnetism computer and is called "Start susceptibility server".

Once the server is running you can start the clients. To start the magnetism client there is another batch script, also distributed with the source files, the link on the desktop is called "Start magnetism client", the client should connect to the different instruments and then connect to the server.

On the cryogenics computer we can start the cryogenics client by opening a batch script linked to the desktop called "Start cryogenics client", similarly to the magnetism client it should connect to some instruments, and then to the server, if it fails then it is most likely the IP-address that has changed see appendix A on how to fix this issue.

Now the two socket clients are running and connected to the server, then we only need to launch the GUI, you can do this on any computer connected to the NBI network by opening the browser and navigating to the IP-address of the system running the server on port 3000¹⁷.

5.5.2 The status page

The first page you see is the status page (see figure 5.24 for an example), here you can at a glance see a snapshot of how the system is behaving right now. In the menu on the left you can select which page you wish to see, below that you can see the connection status of the clients, as well as whether the system is saving temperature datapoints. In the main window are two plots (both update automatically) showing the current temperatures and magnetic fields, and below that you can see how many datapoints that are in the latest experiment run (which also updates as the data ticks in).

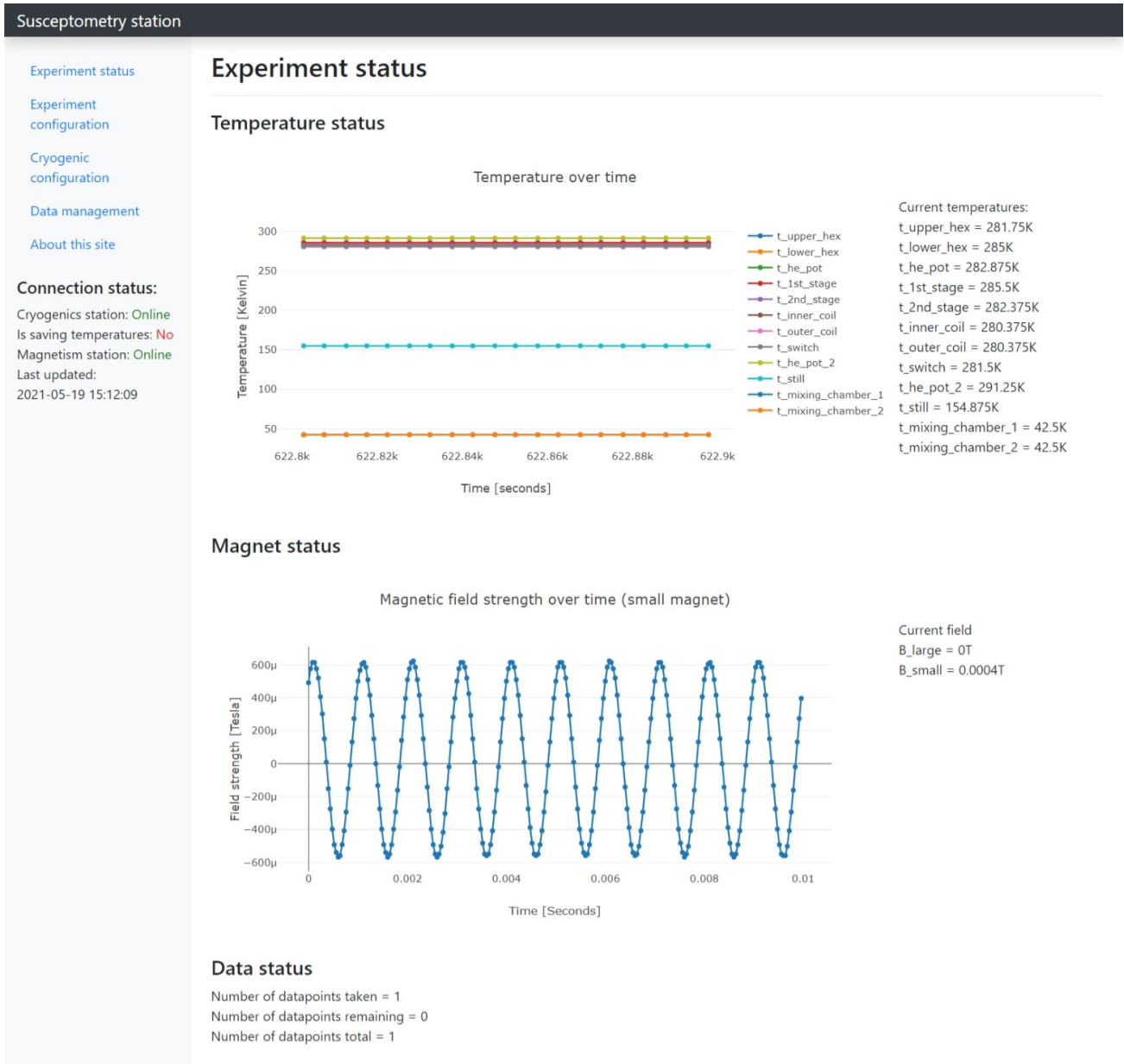


Figure 5.24: The status page. This is the first page the user sees when they navigate to the GUI. We see basic information about what is going on right now. The top plot shows the current temperature of the cryogenic setup, and the plot below shows the current status of the magnets (both the one attached to the function generator, and the superconducting DC magnet). Lastly we have a status over the datapoints in the latest experiment run, so we can see how far along the experiment is.

Susceptometry station

Experiment status

Experiment configuration

Cryogenic configuration

Data management

About this site

Connection status:
Cryogenics station: Online
Is saving temperatures: No
Magnetism station: Online
Last updated:
2021-05-19 15:12:34

Experiment configuration

Data collection options

Select wait time before measuring [s]:

Select number of points per measurement:

Lock-in amplifier (SR830) configuration

Select sensitivity:

Frequency:

Buffer size:

Signal generator (N9310A) configuration

Select minimum frequency [Hz]:

Select maximum frequency [Hz]:

Select minimum Amplitude [V]:

Select maximum Amplitude [V]:

Select number of sweep points:

Cryonics magnet configuration

Select minimum fieldstrength [T]:

Select maximum fieldstrength [T]:

Select number of sweep points:

Oscilloscope (Analog discovery 2) configuration

Save experiment configuration

Warning: Pressing this button will stop any current run, overwrite the experiment configuration, and start a new run.

Figure 5.25: The configuration page. Here the user can setup specify the experimental parameters to use. When the configuration is saved, the system launches a new experiment.

5.5.3 The experiment configuration page

The experiment configuration page allows you to start an experiment (see figure 5.25). You can configure how long to wait between each measurement (to ensure stability and statistical independence of each measurement), you can select the number of datapoints to collect per experimental setting (if you want to average to decrease uncertainty for example), you can set the lock-in amplifiers sensitivity and collection parameters to optimize them for the sample and experiment parameters you have, you can control the AC-field by sweeping through a set of frequencies and amplitudes, and finally you can sweep through a DC-field range to scan the entire magnetic range of your sample. When you save an experiment configuration, only the current datapoint of any running experiments will be saved and the rest of the experiment will be cancelled, a new set of experiment steps is generated and the next step is sent to the socket clients immediately.

5.5.4 The cryogenics configuration page

On the cryogenics configuration page (see figure 5.26) you can see the status of the dilution refrigerator, you can monitor the current temperatures and pressures, and if you press the button to save the temperatures you can see a long timescale plot of the temperatures (this plot is the only one that does not live update, a button is available to update it).

¹⁷at the time of writing the url is <http://172.20.2.237:3000>

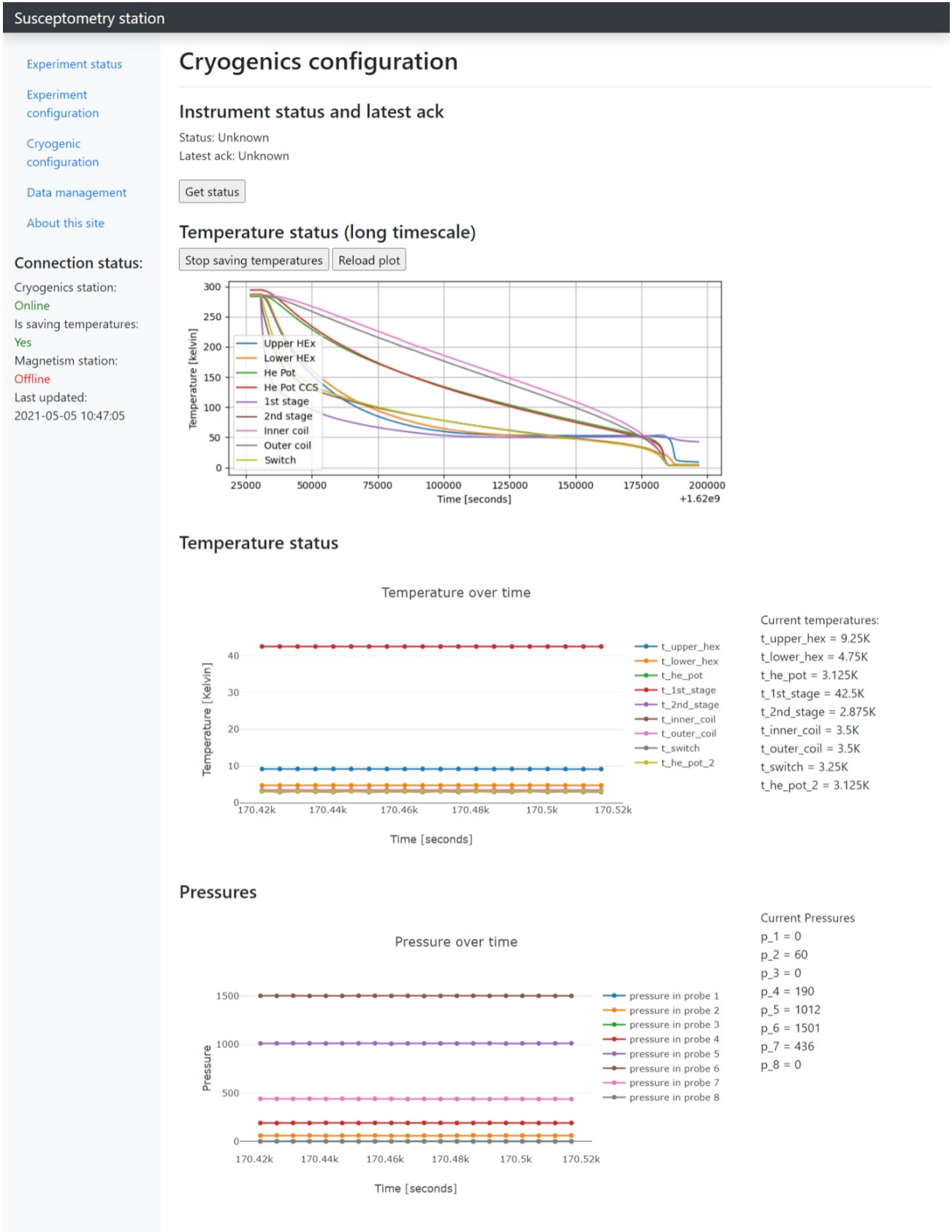


Figure 5.26: The cryogenics configuration page. This page shows more detailed information about the status of system, including a plot of the temperatures with a longer timescale to see trends.

5.5.5 Extracting data and running analysis

When we have created an experiment, we can immediately see the configuration on the data management page, this allows us to check if the experiment is configured correctly, and especially how many steps were generated. It is possible to export the data that has been measured by clicking the export button, this can be done at any point if you want to compute or plot something that isn't readily available in the GUI. The export contains all the generated steps, even if they don't have any associated measurements, so you need to check for data.

Once you have clicked the export button, the browser should download a JSON file containing all the data, and if you need help with opening the datafile you can copy the python script from the data management page and change the filename to the file you want to open, make sure the script and the data file are in the same directory. The script has an example of how you can find the mean lock-in amplifier amplitudes by iterating over each step and extracting the steps data.

The datafile contains an object which has the experiment configuration at the outermost level, as well as a list of all the steps that were generated from that experiment configuration. Each step contains the settings which the instruments had at that point as well as a list of datapoints. Each datapoint has a list of magnetism datapoints, pressure datapoints, and temperature datapoints, which are the actual values of the measurements the system made at that point. The helper script shows how to iterate through each step, and each magnetism datapoint to grab the mean lock-in amplifier amplitude for each step.

Susceptometry station

Experiment status

Experiment configuration

Cryogenic configuration

Data management

About this site

Connection status:

Cryogenics station: Online
 Is saving temperatures: No
 Magnetism station: Online
 Last updated:
 2021-05-19 15:13:54

Data management

#	Created	Datapoints	Delay	Magnet	Excitation amplitude	Excitation frequency	Lockin sensitivity	Export
30	2021-05-07T08:34:22	1 / 1	1 s	0 to 0 T in 1 steps	0.5 to 0.5 V in 1 steps	1000 to 1000 Hz in 1 steps	0.000001 V	Export
29	2021-05-05T12:55:57	1000 / 1000	1 s	0 to 1 T in 10 steps	0.4 to 0.6 V in 10 steps	950 to 1050 Hz in 10 steps	0.0005 V	Export
28	2021-05-05T12:18:52	10 / 10	1 s	0 to 1 T in 10 steps	0.5 to 0.5 V in 1 steps	1000 to 1000 Hz in 1 steps	0.0005 V	Export
27	2021-05-05T11:51:58	10 / 10	1 s	0 to 1 T in 10 steps	0.5 to 0.5 V in 1 steps	1000 to 1000 Hz in 1 steps	0.0005 V	Export
26	2021-04-20T09:56:22	1 / 1	1 s	0 to 0 T in 0 steps	0.5 to 0.5 V in 0 steps	1000 to 1000 Hz in 0 steps	0.001 V	Export
25	2021-03-15T13:50:04	400 / 400	1 s	0 to 0 T in 0 steps	0.5 to 1 V in 20 steps	900 to 1000 Hz in 20 steps	0.000001 V	Export
24	2021-03-15T13:46:23	400 / 400	1 s	0 to 0 T in 0 steps	0.5 to 1 V in 20 steps	900 to 1000 Hz in 20 steps	0.000001 V	Export
23	2021-03-15T13:45:57	400 / 400	1 s	0 to 0 T in 0 steps	0.5 to 1 V in 20 steps	900 to 1000 Hz in 20 steps	0.000001 V	Export
22	2021-03-15T11:32:15	100 / 100	1 s	0 to 0 T in 1 steps	0.5 to 0.8 V in 10 steps	900 to 1000 Hz in 10 steps	0.000001 V	Export
21	2021-03-15T11:25:39	200 / 200	1 s	0 to 0 T in 1 steps	0.5 to 0.8 V in 200 steps	1000 to 1000 Hz in 200 steps	0.000001 V	Export

How to import and treat the data

The data is saved to a database when the experiment is running, and is sent to the user in a JSON format. The JSON format is a nested object that can be read by builtin methods of most programming languages. In Python it is read into a dictionary by default, but pandas for example supports reading it into a series for processing.

The outermost layer of the format contain the most generic options, here you can find the experiment configuration used to generate the steps, as well as a list of all the steps generated. Each step contains the specific parameters used in that step and it contains a list of datapoints. Each datapoint has a time of when it was saved, and it has some magnetism datapoints, where the ac rms field, the dc field, the lockin amplitude, and the lockin phase are saved. It also has pressure datapoints where each pressure of the GHS is saved, and finally it has temperature datapoints where all the temperatures are saved.

The following is an example Python script that opens a datafile (set by the variable "data_filename"), it prints the entire export dictionary, then it grabs the max function generator frequency used, and it computes the mean lockin amplitudes and prints those out.

```

1 # We need a couple of modules to import the data and print it
2 import json
3 from pprint import pprint
4
5 # We import numpy to process data, but we don't actually need it to import the data
6 import numpy as np
7
8 # We set the filename here
9 data_filename = 'data_export_id17.json'
10
11 # We open the file
12 with open(data_filename, 'r') as df:
13     # And read the contents
14     file_contents = ''.join(df.readlines())
15
16 # Next we decode the file contents using the json module
17 experiment_configuration = json.loads(file_contents)
18
19 # Now we have all the data available
20
21 # And we print it to see what is going on
22 print('the entire experiment')
23 pprint(experiment_configuration)
24
25 # After this we create a new line to separate the content
26 print('\n')
27
28 # It is a deep dictionary where the outermost layer contains the experiment configuration
29 # To see a specific experiment configuration parameter (for example the max frequency of
30 # the function generator) we can run:
31 print('the max frequency:', experiment_configuration['n9310a_max_frequency'])
32
33 # After this we create a new line to separate the content
34 print('\n')
35
36 # Inside the configuration we have a parameter called steps, which contains all the steps
37 # generated by this specific configuration.
38 # We can either iterate through each step, or we can select a specific step.
39 # Let's iterate through the steps, and find the mean lockin amplitude for each step
40
41 # We start with an empty list
42 mean_lockin_amplitudes = []
43
44 # Then we run through each step
45 for step in experiment_configuration['steps']:
46     # We create a second, temporary, list to hold the the amplitudes of this step (the ones we take the mean of)
47     lockin_amplitude_datapoints = []
48
49     # We iterate through the datapoints
50     for datapoint in step['datapoints']:
51         # And we iterate through the magnetism datapoints
52         for mdp in datapoint['magnetism_datapoints']:
53             # We add the lockin amplitude to the list we created inside the for loop
54             lockin_amplitude_datapoints.append(mdp['lockin_amplitude'])
55
56     # Now we find a mean of that list and add it to the list of mean amplitudes
57     mean_lockin_amplitudes.append(np.mean(lockin_amplitude_datapoints))
58
59 # Finally we print out the list of mean amplitudes
60 print('the mean lockin amplitudes')
61 pprint(mean_lockin_amplitudes)

```

Figure 5.27: The data management page. Here the user sees an overview of the experiments that have been run, they can export data related to a specific experiment, and see an example of how to read the data.

Chapter 6

Experiments

In this chapter I will describe some of the experiments I conducted. The experiment schedule was unfortunately impacted heavily by restrictions due to the COVID-19 pandemic.

6.1 The first cooldown

The first experiment was run to see if we could cool down the system properly, and was meant to serve as an instruction in using the equipment. I contacted Mathias Mikkelsen (MM), and he agreed to instruct Kim Lefmann and I in the operation of the cryostat. In preparation I had pulled a vacuum in the required places using a scroll pump and cooled the system below 4K.

When MM arrived we started pulling a high vacuum on the IVC using the turbo pump. In the mean time we flushed the system of contaminants by heating the activated charcoal filter and pumping out any contaminants. Pulling a high vacuum is a slow process so when we got to a pressure of approximately 10^{-3} millibar we decided to start the circulation (despite the advice that we should reach approximately $5 \cdot 10^{-4}$ millibar). MM instructed us in how we started normal circulation, and how we should restore the mix to the dumps, then we tried starting the compressor but without success. In previous experiments[5] they had similar issues where the bypass valve was stuck but moving some mix and opening the valve clearly showed that this was not the issue.

Instead we tried loading the backside of the compressor with increasing amounts of mix until it started at approximately 1000 millibar. We let the circulation run with the compressor for a while, this condensed some of the mix and lead to a drop in temperature.

The condensing stagnated, and we concluded it was due to a too high starting pressure the the IVC, so we decided to stop the cooldown, which I could retry at a later stage. We restored the mix to the dumps and later I began reheating the system so I could remove the insert and develop the lab software further.

During the first experiment it also became clear that we needed to log the temperatures for a longer timescale than was currently possible, this was later implemented.

6.2 Subsequent experiments

Once the software reached a maturity where I felt confident in the functionality I mounted a set of coils loaded with a powder sample of cerium(3+) magnesium nitrate hydrate (CMN) and went through the process of preparing for a cooldown described in chapter 5.

The next few cooldowns I made sure the starting pressure was sufficiently low and I attempted to cool down both with and without help from the compressor but in all cases the condensing stagnated before a sufficient amount of mix was in circulation to reach our temperature targets. In the experiments I conducted I varied the pressure of the exchange gas (to ensure the insert was cooled to a point where we could condense helium, but without adding too much heat from the surroundings), next I varied how fast the mix was added into circulation,

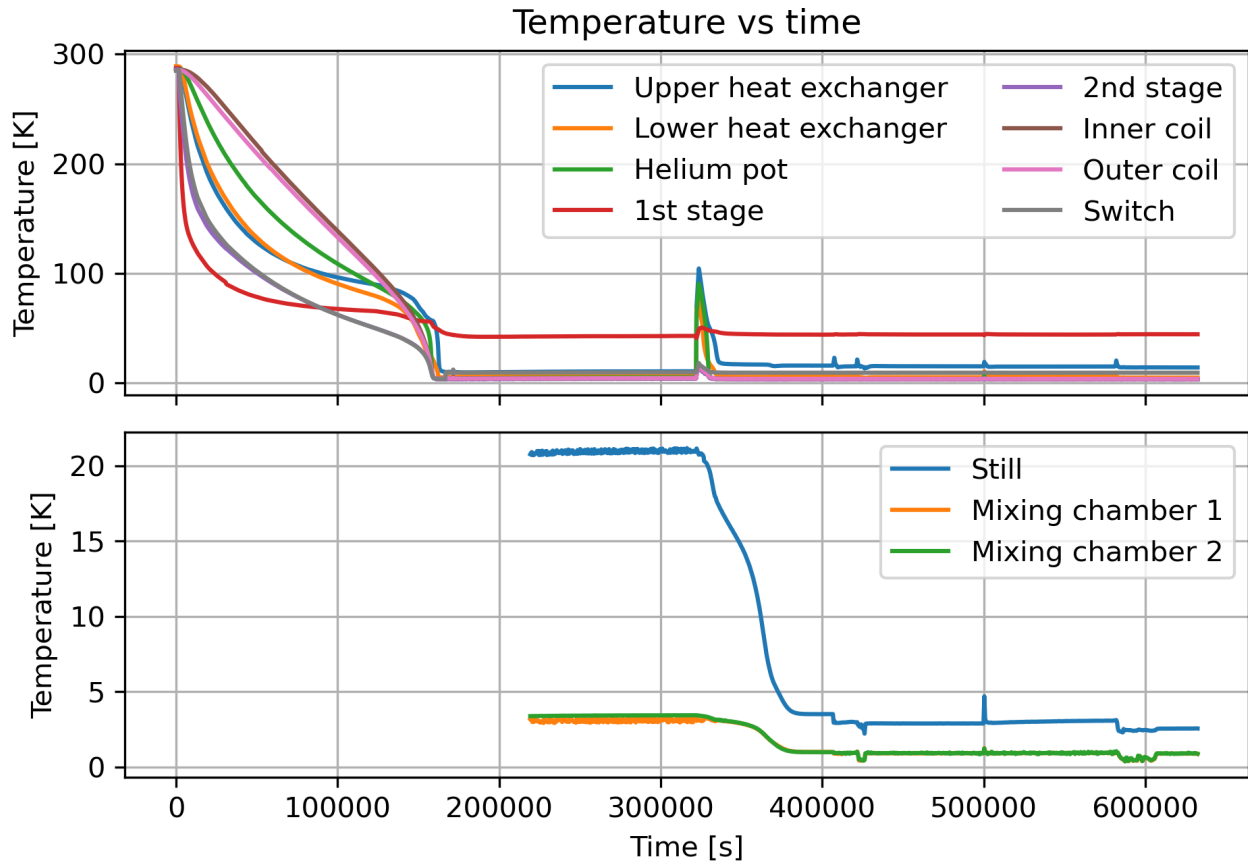


Figure 6.1: A plot of temperatures in the cryogenic system during a cooldown where I was trying to figure out why the system was not cooling sufficiently. Initially the dilution refrigerator was not cooling, I checked the pressure of the exchange gas and found the none so I added some exchange gas (which is seen as the peak in temperature at approximately 330000 seconds). After the exchange gas was added it starts cooling correctly, and we reach approximately 0.36K. Once we reached 0.36K the system seemed to be saturated and I could not get it to condense any more mix.

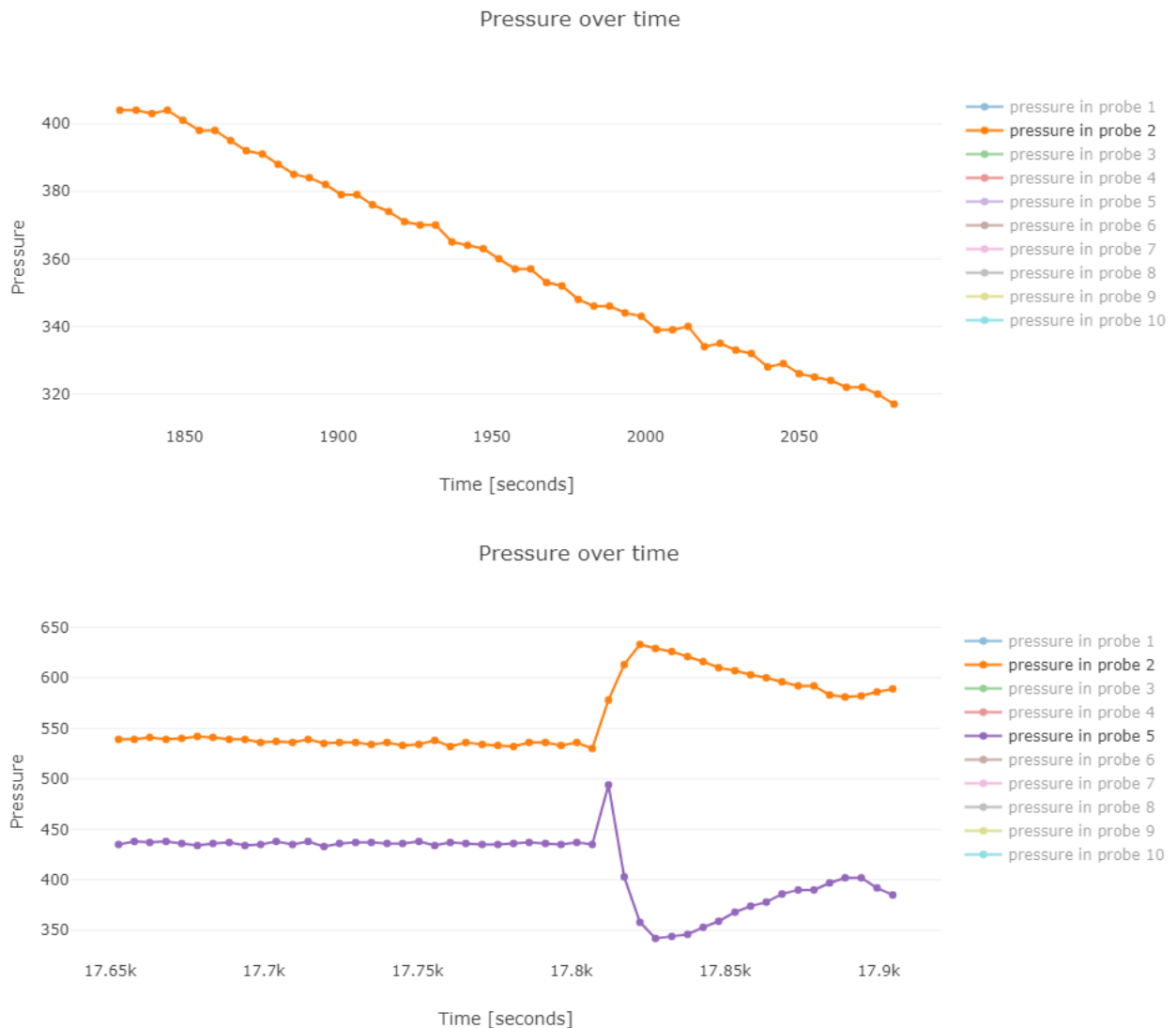


Figure 6.2: Plots of the pressure right after the compressor (probe 2) and right after the scroll pump by the dump valves (probe 5), see figure 5.13 for a detailed view. The top plot shows the mix condensing after it has been added to the dilution refrigerator loop. The bottom plot shows the pressure before and after the compressor when new mix is added to the loop. We see the pressure before the compressor drop rapidly as the mix is moved into the dilution refrigerator, while later the pressure rebuilds once the uncondensed gas is pumped away from the still, this is a good example of the expected behavior.

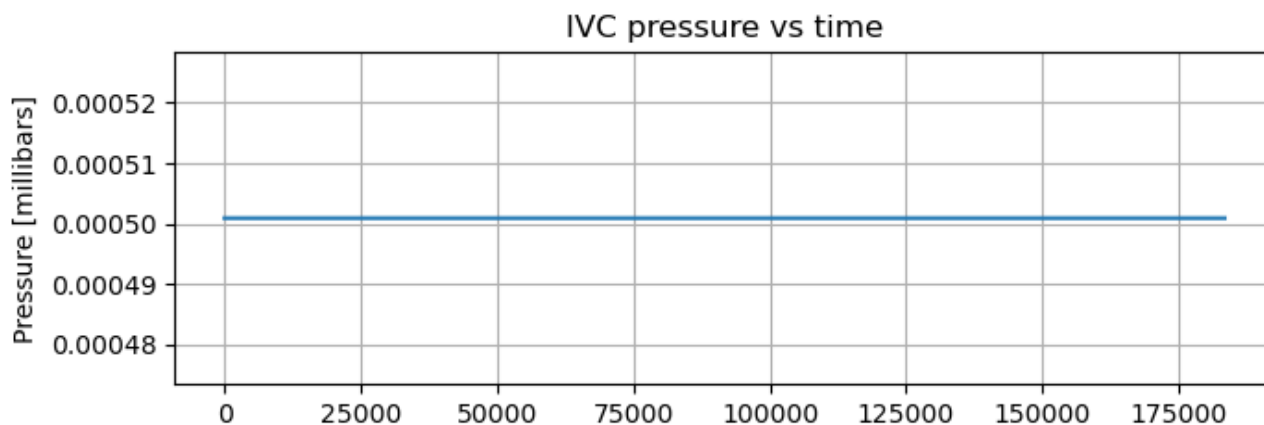


Figure 6.3: Plot of the IVC pressure during the followup experiment to check if A0 was leaking.

I tried both small additions with a long time to condense, and larger additions more frequently but to no avail. An example of such a cooldown can be seen on figure 6.1, where the system became saturated with mix before we could reach the target of 15 mK.

6.3 Finding the leaking valve

Finally I enlisted the help of Finn Berg Rasmussen and Kim Lefmann with Kristine Krighaar observing. I started by preparing the system, ensuring the pressures were correct with high vacuum in the still and IVC, and I cooled the system down below 4K to ensure we could condense helium, then I topped off the cold trap with LN₂.

We started condensing helium using the compressor as done previously until we reached the equilibrium point, where we tried adding and removing mix from the still to see if we could keep condensing when we noticed the IVC pressure responding when mix was added (see figure 6.4) indicating either a response to the warmer gas, or a leak, in either case it prevents us from cooling down completely as the cryostat lacks sufficient insulation. We reached approximately 0.25K, the temperatures of the system throughout the experiment can be seen on figures 6.5 and 6.6.

From this experiment we have three theories:

1. A10 is leaking (the valve between the still and the IVC)
2. A0 is leaking (the valve between the IVC and the low vacuum side of the circuit)
3. The increased pressure is caused by heating condensed gasses inside the IVC

I let the system sit idle at low temperature overnight, if the pressure inside the IVC is caused by heating condensed gasses in the IVC, then we should see the pressure drop off completely by leaving it cold, however I found no significant change. Next I pumped out the still and the IVC to a high vacuum, and then I added exchange gas through the aux port to the volume behind A0, and saw no change in pressure inside the IVC (see figure 6.3 for the IVC pressure data), from this it seems likely that A10 is leaking and need to be serviced or replaced to continue.

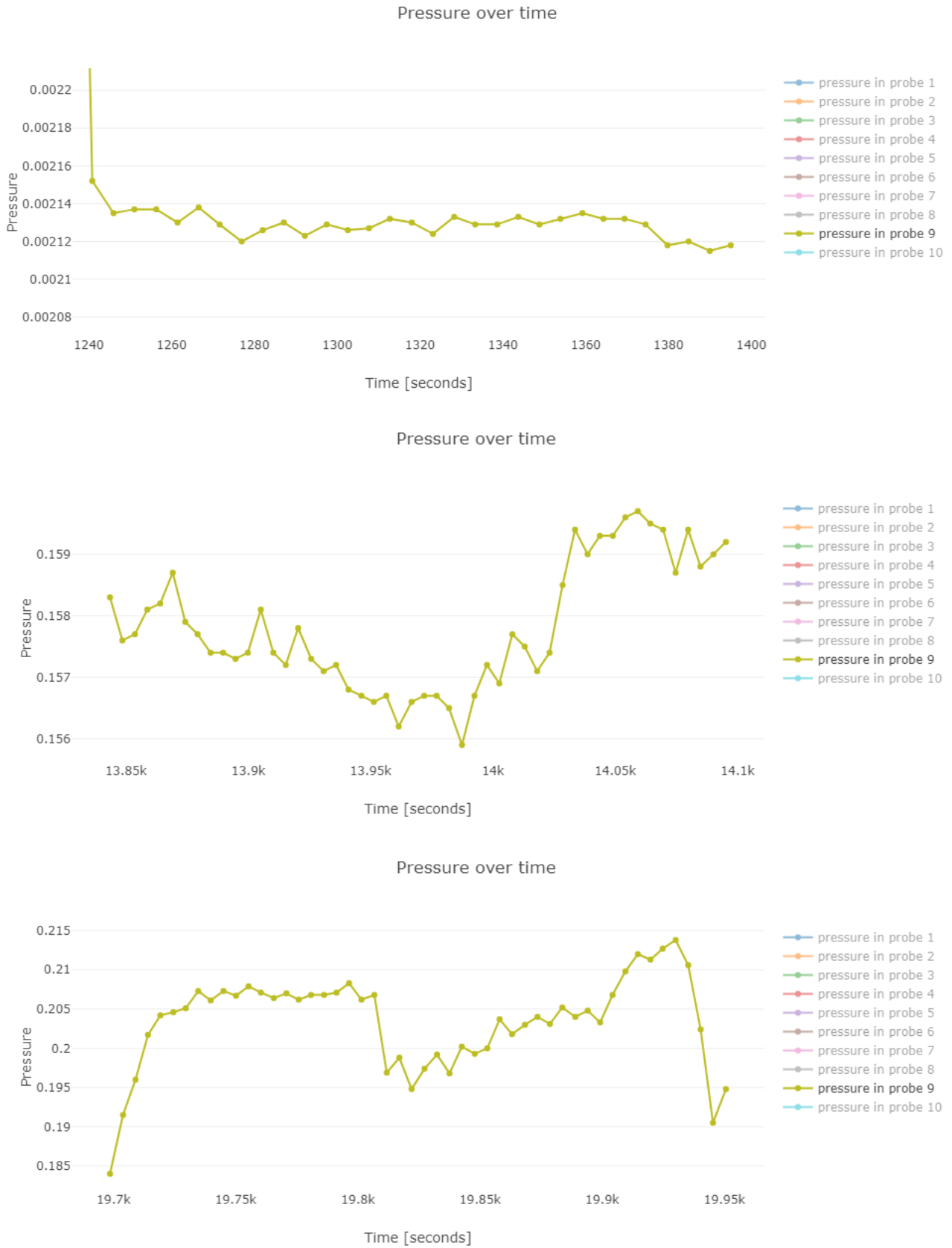


Figure 6.4: Plots of the IVC pressure as a function of time in the experiment where we found the leaking valve. The top plot shows the IVC being pumped out, while the two remaining plots show the pressure problematically responding to new mix being added to the dilution refrigerator.

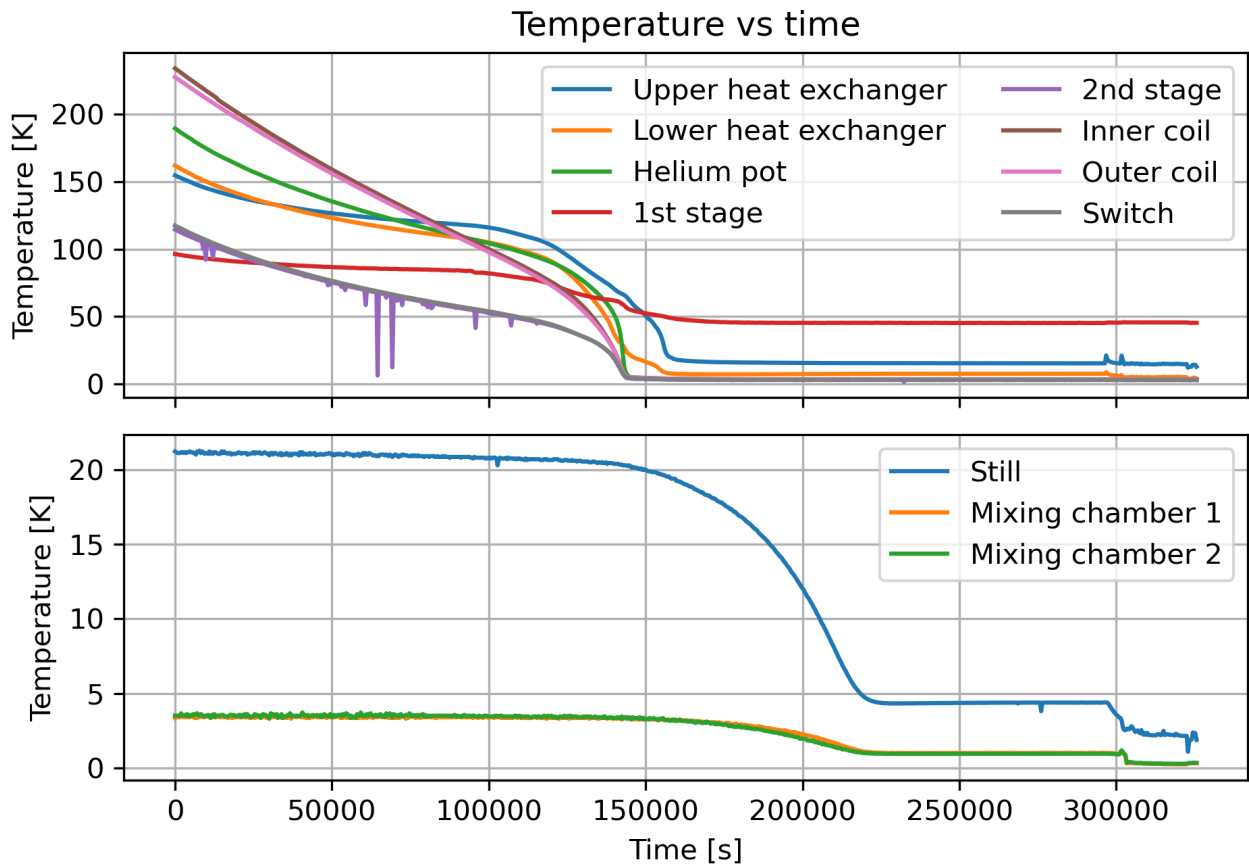


Figure 6.5: Plot of temperatures throughout the cryostat during the cooldown where we found the leaking valve. We see that the system cools down normally using the pulse tube compressor (the top figure), and we see the dilution refrigerator coming into action and cooling further beyond the approximately 4K the compressor can achieve.

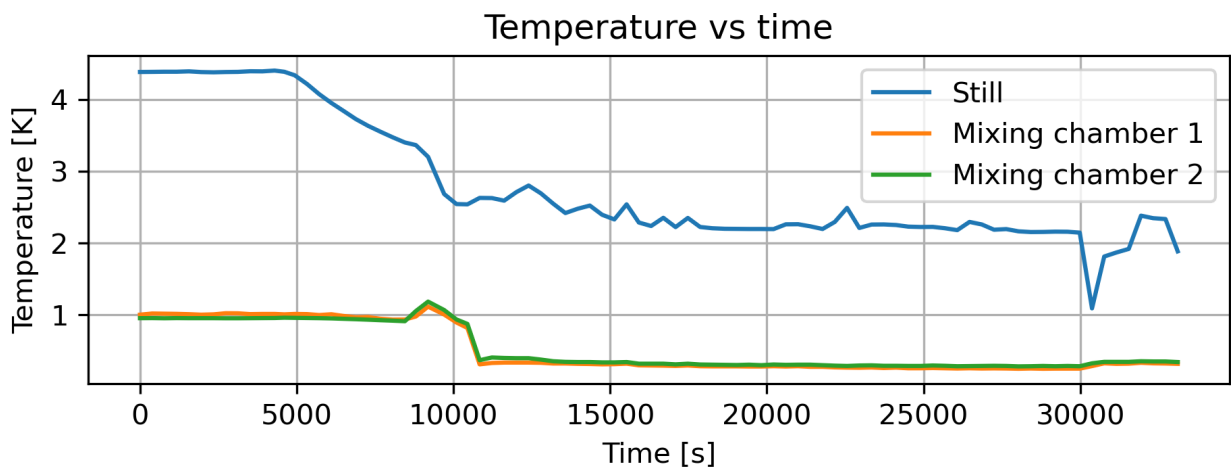


Figure 6.6: A plot of the temperatures in the dilution refrigerator during the last part of cooling seen on figure 6.5, we see the temperatures drop to approximately 0.25K, which proves that dilution refrigeration was taking place, but unfortunately we were unable to cool further.

Chapter 7

Conclusion and outlook

My aim was to lower the barrier to entry into the lab, and to the field of susceptometry in general, by developing a complete and integrated software solution we have achieved that. In addition we can now deal with the recurring issue of broken coils more effectively by using the automated coil winder we developed, and finally we have addressed the lack of documentation by creating an in depth laboratory manual.

The manual covers how the lab is connected, the role of each instrument in the measurement of the susceptibility, how to practically wind a set of coils and mount a sample inside them, how to operate the cryogenic system with specific goals in mind (preparing the system, cooling down, and reheating), and finally how to use the newly developed software to setup an experiment, run the experiment, extract the data we have gathered, and analyse the data to form future conclusions.

It was necessary to update some of the equipment in the lab to allow further automatization using the software suite, we specified, ordered, and setup a new lab computer and oscilloscope.

By running experiments in the form of cooldowns I have tested the software suite and diagnosed a leaking valve, which unfortunately prevented us from achieving 15 millikelvin temperatures the apparatus should be capable of.

7.1 Future work

The next step of the project is moving the lab to the Niels Bohr Building, this will be a large undertaking, and will likely require extensive work to setup, moving the cryostat will likely need external professional help, and the remainder needs to be disassembled, moved and reassembled. Once the lab has been moved, we will need to fix the leaking valve, this is most likely an off the shelf part that can be managed in-house, but it is a larger undertaking non the less.

Once the lab is setup and running, we will hopefully have a fully functioning susceptometer, allowing us to do in-house characterization of magnetic materials such as superconductors and single-molecule magnets. Using the automated coil winder, we can wind new measurement coils to increase the sensitivity by reducing the wire diameter allowing more turns, we will be able to automate the cooling process by programming the processes described in chapter 5 into the susceptometry control system, and we will (hopefully) be able to develop our understanding of magnetism in condensed matter physics in general.

Bibliography

- [1] Stephen Blundell. *Magnetism in Condensed Matter*. 1st. Oxford, New York: Oxford University Press, 2001. ISBN: 0-19-850591.
- [2] Leiden Cryogenics. *Microkelvin 50-100/400 tof dilution refrigerator instruction manual*. 2020.
- [3] Leiden Cryogenics. *Temperature Control, Automation and Probe control software manual*. 2016.
- [4] David J. Griffiths. *Introduction to Electrodynamics*. Prentice-Hall Inc., 1999. ISBN: 013805326X.
- [5] Mathias Mikkelsen. “Measuring Magnetism”. Master thesis. University of Copenhagen, 2019.
- [6] Martin Nikolo. “Superconductivity: A guide to alternating current susceptibility measurements and alternating current susceptometer design”. In: *American Journal of Physics* 63.1 (1995), pp. 57–65. ISSN: 0002-9505. DOI: 10.1119/1.17770.
- [7] Julian Piatek. “Ultra Low Temperature Susceptometer and Magnetometer - Study of the Spin Glass Series $LiHo_xEr_{1-x}F_4$ ”. PhD Thesis. École Polytechnique Fédérale de Lausanne, 2012.
- [8] Picowatt. *Avs-47B Ac Resistance Bridge operators manual*. 2006.
- [9] F. Pobell. *Matter and methods at low temperatures*. Springer, 2007, p. 563. ISBN: 9783540463566. DOI: 10.1016/0011-2275(93)90254-1.
- [10] D Porter. “Susceptibility Study and Measurements”. In: *Stagier Report, Institute Laue-Langevin* (2007).
- [11] Clara Celeste Qvotrup. “Design and testing of an AC susceptometer for low temperature susceptibility measurements”. Bachelor’s thesis. University of Copenhagen, 2019.
- [12] C. V. Topping and S. J. Blundell. “A.C. susceptibility as a probe of low-frequency magnetic dynamics”. In: *Journal of Physics Condensed Matter* 31.1 (2019). ISSN: 1361648X. DOI: 10.1088/1361-648X/aaed96.

Appendix A

An overview of the control system

In this chapter I will attempt to give an overview of the technical aspects of the entire control system, I will start with controlling the instruments using Python 3 and move along the stack to the website which the user interacts with.

A.1 Controlling instruments using Python

First I recommend going through the tutorial "15 minutes to QCodes"¹ to better understand the foundation on which the control is based.

A.1.1 Connecting the instrument to the PC

We start by considering which instruments we want to connect to, and we must find a way to connect all the instruments to a computer. Most of the instruments either have a dedicated USB interface, or they have an RS232 interface (which is a serial connection).

Once we have the instruments connected to their stations, we can start trying to connect to them, to do this we typically use a protocol called VISA which is an abstraction layer made to unify communication protocols with scientific instruments. VISA (which is implemented in QCodes) provides a common interface to the instruments regardless of whether they are connected via USB (such as the oscilloscope), GPIB (such as the lockin amplifier), or RS232 (such as the cryogenics gas handling system). It can be very useful to connect to the instruments using a serial program such as NI MAX or PuTTY and trying to send them commands such as "*IDN?" which is an identity command.

Once we know that we can communicate with the instrument we can begin setting up our station in software. The next step is finding a QCodes driver for the device, some drivers are implemented directly by the QCodes team² (such as the Stanford instruments SR830 lockin amplifier which we use), some are implemented by the community³, and finally some must be implemented by us⁴.

¹Which can be found here: https://qcodes.github.io/Qcodes/examples/15_minutes_to_QCoDeS.html

²The list of available drivers can be found here: https://qcodes.github.io/Qcodes/api/generated/qcodes.instrument_drivers.html

³They can be found in a separate GitHub repository here: https://github.com/QCoDeS/Qcodes_contrib_drivers

⁴The instrument drivers that I have implemented for this project can be found here: https://github.com/hyatt03/Susceptometry-measurements/tree/master/instrument_drivers

A.1.2 Connecting the instrument to Python

Once we have found the drivers we need the VISA address of the instrument, I used NI Max (which is bundled with LabView) to find the address but other methods work equally well, for example using PyVISA:

```
import pyvisa
print(pyvisa.ResourceManager().list_resources())
```

This example produces a list of the available VISA addresses.

A.1.3 Determining which address corresponds to which instrument

In some cases the addresses are self explanatory (which is mostly the case with USB devices where you can see a change after turning it on), but since all com ports are listed at once, we may need to determine which instruments reside at which addresses. Most VISA compatible instruments support an identity command `"*IDN?"` which you can use to help identify what components are connected to certain com ports, here is an example which sends that command to all available com ports:

```
# We import pyvisa
import pyvisa

# We open a resource manager
rm = pyvisa.ResourceManager()

# We iterate through the connected resources
for res in rm.list_resources():
    # We open a connection to a resource
    inst = rm.open_resource(res)

    # We query the resource for an identity
    print(res, inst.query("*IDN?"))

    # We close the connection to the instrument
    inst.close()

# And finally we close the resource manager
rm.close()
```

In some cases this code may fail (if the device does not respond), then you can just iterate through the ones that do respond manually, and try to identify the device through other means (typically I would consult the manual in this case).

A.1.4 Finding appropriate drives

Now that we know which devices are connected and what the corresponding addresses are, we can begin integrating the device into the measurement system. The first step is locating an appropriate QCodes driver to run the device. There are two main libraries of QCodes drivers, one which is maintained along with the QCodes core (which can be found here⁵), and one which is maintained by the community (which can be found here⁶). If the driver you need is unavailable there, your best bet will be to write it yourself. In my experience most devices have documentation about automation included in the manual, or available by emailing the manufacturer. The QCodes documentation has a guide on how to write drivers (which can be found here⁷).

⁵https://qcodes.github.io/Qcodes/api/generated/qcodes.instrument_drivers.html

⁶https://github.com/QCoDeS/Qcodes_contrib_drivers

⁷<https://qcodes.github.io/Qcodes/examples/index.html#writing-drivers>

A.2 Taking measurements

In this section I will describe the necessary steps to take a measurement and sending it to the server.

A.2.1 Adding the instrument to the station

The first step in taking a measurement is opening a connection to the instrument. In the susceptometry framework this is done at the station level, meaning you should select the station you need (for example: if the instrument is connected to the magnetism station you should open the file "stations/magnetism_station.py"). Next you need to add four lines of code, firstly you need to import the driver (typically done along with the other imports), then you add a variable for the address of the driver you found previously (do this along with the remaining addresses), then you initialize the instrument driver in the "setup_instruments" function, and finally you add the initialized instrument driver to the array at the bottom of the function in the return statement. This way the instrument gets connected at the start of the program, and can be used in the subsequent files.

A.2.2 Calling the instrument from the socket clients

The next step in the technology stack is the socket clients, they are responsible for opening a network connection to the socket server (explained in the next section). We have two socket clients, one for the magnetism station, and one for the cryogenics station.

Each socket client is divided into three components, the first is the startup script which ensures that the clients are configured correctly, that they connect to the server, they start background jobs, and that they don't close unless we want them to. The second component is a queue, which ensures each event is processed in the correct order, and handles slow interactions with both the disk and the instrument interfaces. The third component is the actual socket client which sends and receives events and data between the stations and the central web server.

All of the components have generic parts which are inherited through base classes which can be found in the file "socket_clients/baseclient.py".

Let's imagine we want to add an instrument to the file "socket_clients/cryogenics_client.py" which contains the cryogenics socket client. At the top of the file we have the imports, since we already added the new instrument to the cryogenics station (see previous section), we don't need to import anything new.

Below the imports we have the experiment state, this is used to (temporarily) save data about the current state of the system (used to save the last 20 temperature measurements for easy querying), if we need to remember the state of a system we add a parameter to the dictionary (just start by adding the initial state, such as an empty array).

After setting up the state, we can add the instrument to the queue instance, this is done by add a variable in the init method of the queue class in the format "self.your_var_name = your_var", adding the instrument here can make sense to make it easier to query it later (but it's not strictly necessary).

Next we add a new method to the queue class which queries what we want to know from the instrument, when the method is called you have access to 4 different variables, the first is self which references the queue instance (it is here you can query the instrument using the variable you just assigned), the second is the queue backend which is not useful in most cases, the third is the name variable which contains the name of the working serving the request, and finally we have the task, which is a user configured object that contains all the parameters for this specific execution. The method should now query the instrument for the measurement, then it should send the results back to the socket client (which can be accessed through "self.socket_client").

Once we have written at least a partial method, we should add another line to the init method where we add the new method to the list of queue methods, this is done through the method "register_queue_processor", it takes two arguments, the first is the name used to call the queue method (which must be unique), and the second is a reference to the method.

Now we add two new methods to the socket client, one to call the new queue method, and one to send the results back to the server. The first method should respond to a websocket event, so it has to be named according to a specific convention "on_your_event_name", which ensures that anytime we send a websocket request with

the correct name ("your_event_name") the method is called. In this method add a line which appends the event to the queue:

```
await self.append_to_queue({
    'function_name': 'queue_method_name',
    'options': 'additional_parameters_here'
})
```

Here you have the option to append additional parameters to the queue call (which will be available in the task variable). The next method we need to add is something to transmit the results back to the server, this can have any name you wish, as you call it from the queue processor method, in this method you have a method available called "self.emit" which can send the results to the server, it takes two parameters, the first parameter is the event name, and the second parameter is the data you wish to transmit (these should be simple datatypes, such as python lists, objects, numbers, or strings). Once the last method has been called, we have sent the results to the server.

A.2.3 Process next step

There is a special method in the queue called "process_next_step" which is called as part of the experiment run, so if you need to add more data to the experiment run, this is the method you must modify.

There is a separate implementation of this method in both of the socket clients, as each client has a different flow for taking a measurement.

In both cases we the first task is waiting, we don't want to start a measurement before the previous measurement has taken place, the socket server emits events to signal when a measurement is completed, and the socket clients listen for these events. Once the go-ahead is received we ensure the correct files are open (these files function as a backup to a specific experiment run, and try to ensure no data is lost), then the client signals to the other client that it is ready to take a measurement and waits for the other client to emit the same signal. When the signal is received by both clients we start iterating over the number of measurements to be taken, the first step is always to wait for a configured amount of time (this lets the system we're probing stabilize in the new experiment conditions) then we take the specific measurements and emit the results back to the server, which saves them and send a copy to any connected web browsers.

Once the measurement loop is concluded we save the results to the backup files and finally emit a "step done" signal.

A cryogenics measurement

The cryogenics client probes the GHS for the pressures, and the two temperature controllers for all temperatures in the system.

A magnetism measurement

Before the loop start, a magnetism measurement first configures the experiment to reflect the desired state of the system, this happens by first changing the DC magnetic field being applied, then it changes the AC amplitude and frequency, then it sets the sensitivity of the lock-in amplifier, and finally it autoscales the oscilloscope to measure the AC current being applied.

In the loop it probes the rms current from the oscilloscope, the actual dc field being applied, and it grabs a trace from the lock-in amplifier, this also happens concurrently.

A.3 Background jobs

In addition to "process_next_step" there is a background job running which periodically calls a few measurement methods, the results are typically used to display graphs in the gui of the system on the various status pages.

Generally there is a method called "background_job" defined on each of the socket clients, this method is run in a separate background thread and periodically appends to the relevant queue to enable "live" monitoring of the system status remotely.

In the case of the magnetism client it probes the oscilloscope for a trace every 5 seconds and sends that trace, along with an rms value, to any browsers connected to the system and the trace is displayed on the experiment status page along with the RMS value of the trace, as well as the value of the current DC magnetic field.

The cryogenics client first starts a separate queue job that probes the temperature (this is required to reduce the latency of the system), and then it begins probing for the pressures and the GHS status every 5 seconds. The temperature probe is quite slow as the resistance bridge has to stabilize each time it change measurement channel (The time it takes to stabilize depends on the resistance it's measuring, therefore it is adjustable through the gui), this is also why it is necessary to split the pressure and temperature measurements into separate jobs.

A.4 Using the lab without the complete susceptometry setup

Sometimes it is necessary to write some custom software to run one-off experiments, when this is the case we can still reuse a large portion of the code by integrating directly with the QCodes drivers. One example of such an integration can be found along with the remaining code here: https://github.com/hyatt03/Susceptometry-measurements/blob/master/record_pressure_and_temperature.py, it is a script that records the temperatures and pressures to a file every 15 seconds until the user interrupts the script.

The script works by opening a connection to four instruments, the maxigauge, the GHS, the resistance bridge, and the DMM, it sets up some basic parameters for each of the instruments, and finally it loops infinitely where it records the a line to a file with each of the measurements made by the instruments. The file can then be processed like any other.

In this case I made the script such that only the cryogenics station had to run, and without needing to specify how many datapoints I wanted ahead of time.

A.5 Common issues

Before starting the stations, ensure the instruments are turned on and are done booting up. If issues persist, turn everything off (all instruments and computers) and restart them, then once they are running try opening the socket clients again. Also be sure to start the webserver before the socket clients.

A.5.1 Wrong IP-Address

If you are having trouble connecting the stations together it is most likely a change in IP-address that is causing the problems. Open the network settings of the magnetism station and find the IP-address, write down the IP-address.

On the cryogenics PC find the source folder by right clicking the "Start cryogenics client" link on the desktop and selecting "Open file location", open the folder "socket_clients" and open the file "baseclient.py".

On line 88 the old IP-address is typed out, exchange the existing IP-address for the one you found on the magnetism PC, be sure to keep "http://" and ":3000" which is the protocol and port.

Try starting the client now, the network issues should be resolved.

A.5.2 Wrong instrument addresses

If the problem is not network connectivity, then it might be that an instrument has changed address. First verify the problem by restarting the instruments, if it still doesn't work find the source code like we did for a wrong IP-address and the source code of the station you are sitting at inside the stations folder.

The addresses are close to the top of the script, so they should be easy to find. Check if the address is correct by comparing the address to what you find using NI Max (which should be installed on the computers, otherwise it is free for students). If any of the addresses are wrong, correct them and see if the problem is resolved⁸.

⁸On the cryogenics station, we connect directly to the resistance bridge and the maxigauge, so you should just change the COM number, you can find this using the device manager setting in windows.