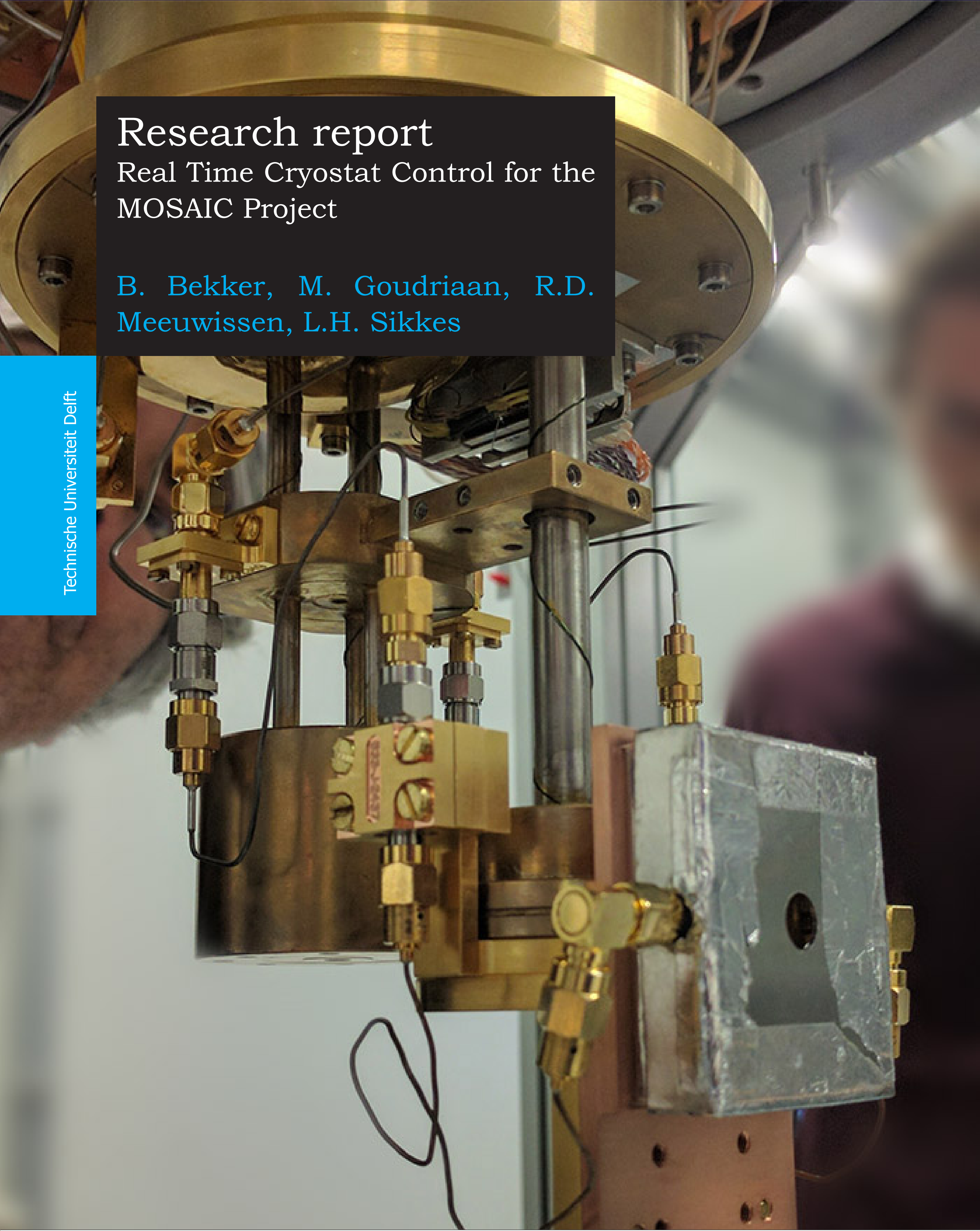
Research report

Real Time Cryostat Control for the MOSAIC Project

B. Bekker, M. Goudriaan, R.D. Meeuwissen, L.H. Sikkes

Technische Universiteit Delft



TU Delft Delft University of Technology

Challenge the future

Research report

Real Time Cryostat Control for the MOSAIC Project

by

B. Bekker, M. Goudriaan, R.D. Meeuwissen, L.H. Sikkes

in partial fulfillment of the requirements for the degree of

Bachelor of Science
in Computer Science

at the Delft University of Technology.

Supervisor:	Dr. R. Krebbers,	TU Delft
Project Owner:	J. Bueno,	SRON
	S. Haehnle,	SRON

Delft University of Technology

Contents

1 Introduction

The Astronomical Instrumentation group of SRON, the Netherlands Institute for Space Research, is working on the design of a new Multi-Object Spectrometer named MOSAIC that can measure the spectrum of radio emissions from galaxies and other objects in deep space. The spectrometer has an array of 25 pixels that give it the ability to measure the radio spectrum for 25 objects at the same time. Additionally the beam of each pixel can be steered electrically to lock onto an individual astronomical object. MOSAIC will be installed on the 10m Japanese ASTE observatory, a world-class astronomical facility based at the Atacama desert in Chile at 5000m altitude.

MOSAIC is based on novel superconducting circuits, which need to be cooled down to cryogenic temperatures to function. Development and testing of the new instrument takes place at the TU Delft's Else Kooi laboratory. Part of the test setup is a cryostat able to reach a temperature of around 0,25 Kelvin.

The cryostat consists of three active elements: A pulse tube refrigerator, a compressor and a He-7 cooler. All of these need to be monitored and operated remotely. In the current system, the three parts of the system are controlled separately using different programs or using manual controls on the components. The programs are able to control the cryostat, but do not have all functionality the client desires. Our program will integrate the different programs into a single application that is able to communicate to all three active elements and offer increased control and monitoring functionality. This will make it easier and faster to set up experiments with MOSAIC at sub-kelvin temperatures and incorporate the cryostat temperature into experiment results.

First of the working of the cryostat and its components are explained. Next the problem statement is formulated along with the problem and the new situation. Then the requirements and four concepts are discussed to illustrate the program that will be created. Finally it will be discussed what architecture will be implemented and which language, architectural software pattern and tools will be used. The paper is finished with a conclusion.

2

Cryostat

This chapter will explain the basic workings of the the cryostat and how it is controlled. The cryostat consists of separate components including multiple active components and control electronics. Figure 2.1 shows a schematic overview of the components, connections and insulation layers. Table 2.1 contains a short definition of the components referenced throughout this paper. Figure 2.2 shows the control electronics.

To achieve a temperature under 300mK (mili Kelvin), the temperature is lowered from ambient temperature ($\sim$ 300K) in multiple steps. The Bluefors casing insulates the experiment from the ambient temperature by holding a vacuum to prevent convection, and has multiple barriers to shield the experiment from heat radiation. The first layer of the cryostat is cooled by the pulse tube refrigerator to 50K, the inner layer is cooled to 3K by a second stage of the pulse tube. The pulse tube refrigerator is driven by a compressor that compresses and expands the gas in the pulse tube to. A separate vacuum pump is used to remove the air from the cryostat.

Mounted inside the cryostat is a He-7 cooler, consisting of a ^{3}He stage and a ^{4}He stage. The cooler cools its cold head to less than 300mK by evaporative cooling of ^{3}He (see appendix A). It also has a ^{4}He "buffer" cooler that pre-cools to 850mK, and is able to sustain a higher cooling load. The experiment is mounted on the He-3 cold head.

The evaporation of He isotopes cools the head by adsorbing latent heat during its liquid to gas transition. The gas particles are "pumped" from the reservoir by adsorption to reduce the vapour pressure and promote the evaporation process. The pumping works because gas particles form a film over a surface of an adsorbent in the pump, in this case activated charcoal, creating a lower pressure in the pump. When all Helium is pumped from the reservoir, the cooler needs to be recycled. In the recycle process, the charcoal is heated to over 20K to make it release the gas. During the initial cool down, the charcoal must be kept heated above 20K, to prevent gas being directly adsorbed by the charcoal. [1] Pumping is activated using gas-gap heat switches, which activate or prohibit heat conductance away from the charcoal. The heat switches are activated using resistance heaters.

The ^{4}He has an higher liquidification point and is used to cool the ^{3}He to its liquidification point of 3.32K[1]. Because of the greater cooling capacity and greater availability of ^{4}He compared to ^{3}He, it is used to cool the experiment as low as possible before activating the He-3 pump, and to buffer all (conductive) connections to the experiment [2].

The compressor is able to be controlled over a remote MODBUS Ethernet connection. A Lakeshore temperature controller [3] monitors several temperature sensors inside the cryostat and controls heater elements. Temperatures can be read using an USB connection. The He-7 cooler is controlled using custom electronics build around an Agilent 34972A data logger / switch unit [4]. The 34972A has an USB and Ethernet interface. It measures temperatures on the different heads and pumps of the cooler, and controls the switches of the pumps and heater elements on the pumps.

Additionally the pressure inside the cryostat is measured using pressure sensors. There is currently no method to read the output of these sensors.

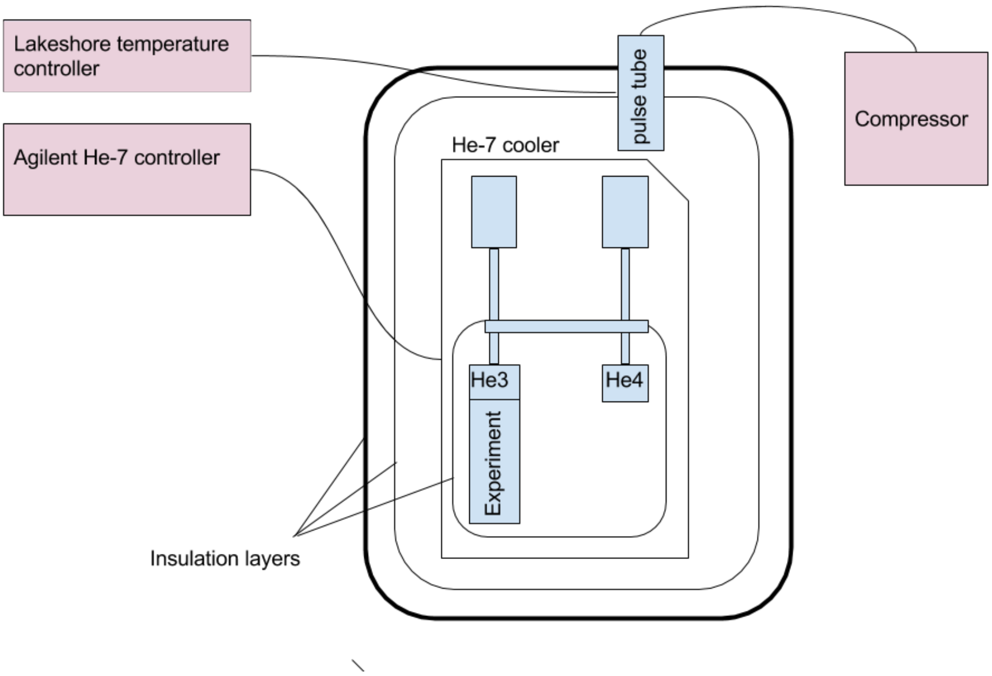


Figure 2.1: Overview of the cryostat and control systems. The components are described in table 2.1

Cryostat	Complete cooling system including casing, pulse tube refrigerator and He-7 cooler
He-7 cooler	Helium cooler consisting of a He-3 and He-4 stage that is able to reach temperatures <300mK
Pulse Tube Refrigerator (PTR)	Compression/Expansion based cooler that is able to cool from 300K down to 3K
Bluefors cryostat	Device that holds the experiment inside in a (partial) vacuum. It insulates the experiment from outside heat.
Compressor	Device that drives the pulse tube cooler by compressing and expanding gas.
Vacuum pump	External pump that creates a vacuum inside the Bluefors cryostat.
Lakeshore temperature controller	Controller connected to several temperature sensors and valves controlling the cool down of the cryostat.
Agilent controller	Input and output device monitoring and controlling the He-7 cooler.

Table 2.1: System components

Figure 2.2: Control devices for the cryostat inside the laboratory. In the left: Agilent (green display) and Lakeshore (blue display) devices front panel. In the right: Rear side showing connections

3
Problem Description

In this chapter the problem is described in further detail. First the problem statement is given. Secondly the current situation is described, followed by a description of the encountered problems and the new features of the client. In the last section the new situation is described, which solves the problems and implements the new features of the client.

3.1. Problem Statement

The ability to perform experiments with the cryostat is limited with the current set-up. The user needs to control the compressor manually. The He-7 and PTR can be controlled with a separate computer program for each of the components, which are not suitable for the current experiments. This way the user has a limited overview and cannot retrieve the desired information, which decreases the efficiency of the experiments.

3.2. Current Situation

In the current situation the user of the cryostat needs to perform multiple actions to control the cryostat. As explained in chapter 2, the cryostat consist of multiple components which are mostly working on their own. This means that those components are connected to the computer, but need to be controlled with separate programs. In the following paragraph it is explained what is needed to run a cool down session, which is globally visualized in figure 3.1.

First a vacuum pump needs to be connected to the Bluefors. Connecting the vacuum pump is only needed once at the start of the cool down process. Once the pressure inside the Bluefors is down to the required level the compressor and the He-7 cooler can be turned on. By turning on the compressor the PTR is automatically turned on. The compressor needs to be turned on manually at the compressor itself which is placed in a wooden box which is not easy accessible. This is only needed once during a cool down session. Since the He-7 cooler is connected to a computer it can be turned on with the computer.

The program used for controlling the He-7 cooler is outdated and does not fulfill the requirements to perform experiments efficiently. The program has a function which is not working: there is an option to set the voltages for heating the helium pumps, but it is either on or off. Controlling the heaters in volts gives the advantage of precisely controlling the temperatures, which enhances the experiments.

The temperature graph inside the program of the He-7 cooler lacks functionality regarding the experiments. It is not possible to zoom in and only the latest part of the session is shown. The Lakeshore temperature controller has its own computer program which monitors the two temperatures of the insulation layers, which are cooled down by the PTR. The graph of those two temperatures has the same downsides as the graph of the He-7 cooler, no zooming and only the latest part of the session is shown. Once the Bluefors is vacuum: the vacuum pump can be removed. This requires manual intervention as valves needs to be closed, which is done once during the cool down session.

Once the cryostat is fully cooled and has run for about 24 hours the He-7 cooler has warmed up too much, all the helium is used, and needs to be recycled. This recycling process takes about two hours. According to the client this a problem as someone needs to manually recycle the He-7 cooler two hours before an experiment can start.

When the cooling session is ended, the following actions are needed to turn the cryostat off. First of all the compressor has to be turned of manually. Secondly the computer programs need to be stopped. To increase the warm up process a heating system in the He-7 can be started. Finally the pressure needs to be restored inside the Bluefors by manually opening a valve.

3.3. Problems and New Features

In this section the problems derived form the current situation and the new features are discussed.

Problems

In the previous section the current situation is explained, the following problems were encountered:

- Direct control on the compressor itself, which has difficult access.
- Multiple programs are needed to control different components instead of one program.
- The used programs do not fulfill the requirements.
- Graphs in the programs do not give clear information for experiments.

The need for setting up the vacuum pump is not experienced as annoying by the client and therefore not is the scope of this project. Even if this process was automated it would still require manual input to connect the pipes and opening the valves, which cannot be automated. The compressor can be placed in an easier accessible place, but still requires the need to manual turn the compressor on. Controlling the compressor via the computer is desired by the client as it solves both problems: single control and no access needed to the compressor. For the need of simplicity and efficiency the client desires one computer which runs a single program which can control all components. Finally the graphs cannot fulfill the needs of the client and therefore need to be updated.

New Features

Besides solving the problems the client encounters in the current situation, the client wants extra features which increase the efficiency of the experiments. The first new feature allows the usage of Python scripts to give simple commands and retrieve data from the program. Another feature is a timer function on the cool down and recycle session, as this increases efficiency and requires less personal presence. To increase the graph functionality the graphs should: be zoomable, show the full session and be able to select which sensors to display.

3.4. New Situation

In this section the new situation is described which solves the problems listed in the previous section and has the new features implemented. A global overview of the new situation can be seen in figure 3.1.

When starting the cool down session the vacuum pump still needs to be controlled manually, as described this is out of the scope for this project. This also holds for disconnecting the vacuum pump. Once the vacuum pump is connected and running the user can start the cool down process in the GUI. The program starts the compressor, the He-7 cooler and the logging of data. The start time of the cooling session can be set, this way the vacuum pump can be used on Friday and the actual cooling session can start on a Sunday morning so that the cryostat is cold on Monday morning. This also holds for the recycle session of the He-7 cooler, which can be either done directly or on a timed moment. The temperatures which are logged are displayed in a graph, which is zoomable and also shows the full session, or at least the desired interval. To stop the session the user can press a button in the GUI then the program turns the compressor off and the heaters of the He-7 off and starts the warm-up mode of the He-7 cooler. Finally the valve needs to be opened to restore the pressure inside the Bluefors.

In the new situation the ability to perform experiments with the cryostat is increased by automating the cooling and recycling of the cryostat. This is achieved by creating a single program which controls all components of the cryostat, besides the vacuum pump. The graphs have the ability to zoom in, display the full time line and filter out sensors. The program can be accessed with Python scripts to give simple commands or retrieve the data needed for the experiments.

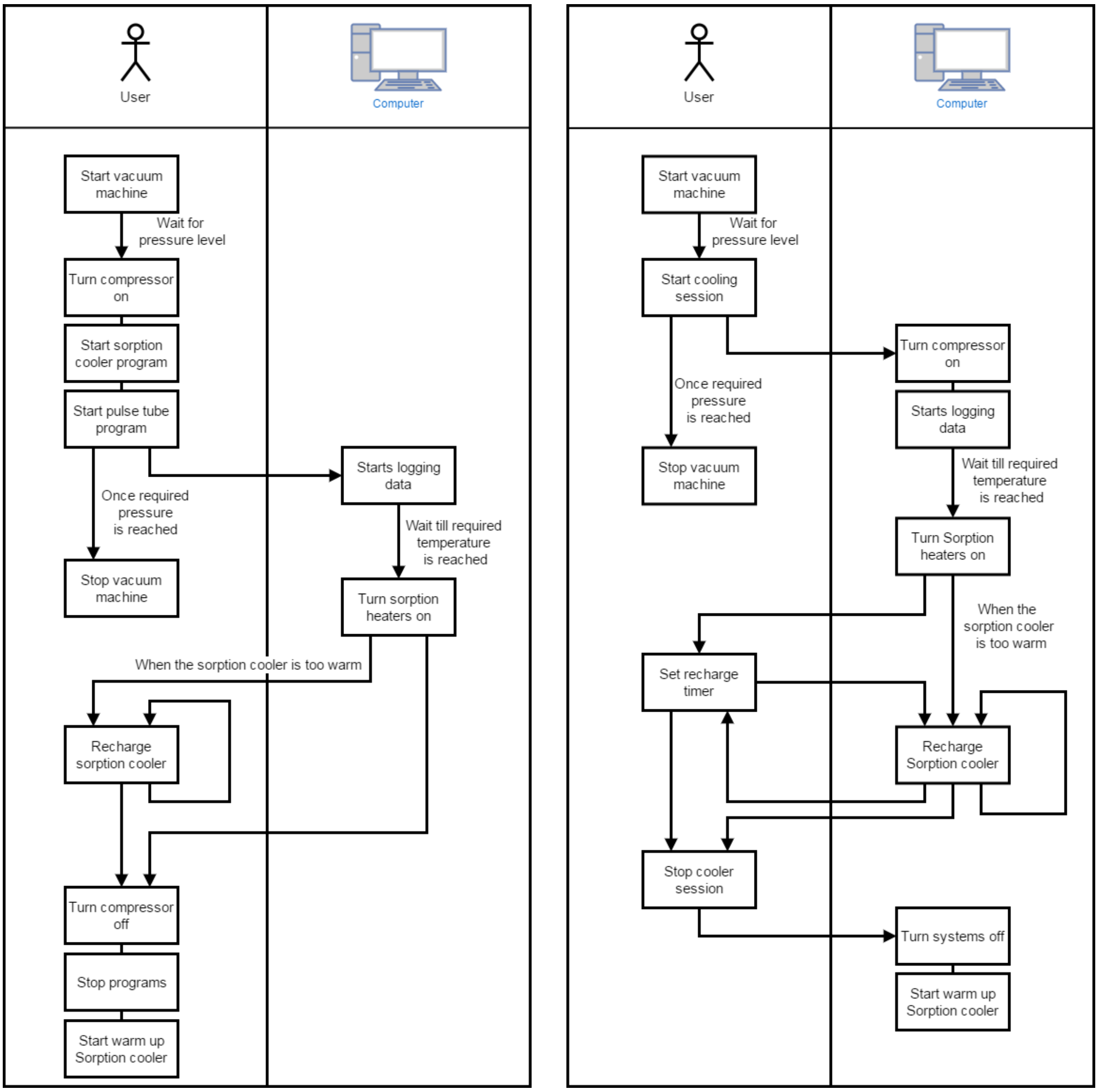


Figure 3.1: This figure displays the global steps needed to execute a cooling session. This is divided in the work what the user and what the computer has to do. In the left: the current situation, which is explained in section 3.2. In the right: the new situation, which is explained in section 3.4

4
Requirement Analysis

In this chapter the requirements of the program are analyzed. First off the requirements are stated and given priorities. Next the design concepts for the program will be discussed.

4.1. MoSCoW analysis

The requirements of our program will be analyzed using the MoSCoW model [5]. The MoSCoW method divides each requirement in one of four categories. These four categories are in descending order of priority: Must have, Should have, Could have and Won't have. The must have's are the bare minimum that the program should fulfill and make the minimum viable product. Should have's are the requirements that are really wanted. Both have a very high priority and should be in the program unless a good reason is given. Could have's will be implemented if the time allows it. Finally won't have's are the problems that have very low priority or are marked as non-appropriate for now, this list can grow in time but is minimal at this point of time in this case. To further clarify our requirements the division between certain components of the program is made.

First off there will be a GUI that allows the user to see what is going on in the cryostat as well as controlling its main functions. In the GUI there will be a graph that shows the temperature of certain components over time. The automatic control mode is the mode that will automatically get the cryostat ready for use. Whenever other settings are needed the manual control mode can be used. To be able to document what is going on in the cryostat certain readings will be logged in a file. Finally there is a possibility to use Python scripts to control certain parts of the cryostat.

Must have

The GUI will contain:

- A button to start one of the automatic control modes:
  - Start the cooling down process.
  - Start the recycling process for the He-7 cooler.
  - Start the warm up process.
- A graph that measures the temperatures of all thermometers over time. This graph will have:
  - A visualization that shows changes in the state of the equipment, for example when the He-7 cooler is turned on.
  - The possibility to zoom in. This zooming option is usable on a manually set period or on a certain state of the equipment.
  - The option to hide certain thermometers in the graph.
- The option to set certain attributes manually:
  - Turn the heater in the He-7 cooler on or off.
  - Turn the compressor on or off.

  - Change the voltages of the heater in the He-7 cooler.

- The option to enable or disable logging for certain sensors.
  - The option to set the frequency of logging for certain sensors.

There will be a graph showing the temperature over time of:

- The status and pressure readings of the compressor.
- The water temperature of the compressor.
- The temperature of the compressor.
- The temperature of the He-7 cooler.
- The temperature of the Bluefors cryostat.
- The temperature of the heater in the He-7 cooler.

The automatic control mode can:

- Start the cooling down process:
  - Start the compressor.
  - Start retrieving the temperature every x seconds, where x can be set manually.
  - Start a graph that measure the temperature over time.
  - Start data logging as explained in the logging section.
  - Start the He-7 cooler, after reaching a temperature of 50K.
- Start the recycling process for the He-7 cooler.
- Start the warm up process:
  - Stop the compressor.
  - Stop the He-7 cooler.
  - Enhance the heating process of the He-7 cooler by warming up the heater in the 3K plate.

The manual control mode can:

- Turn the heaters on or off.
- Turn the compressor on or off.
- Change the voltages of the heaters.
- Interrupt the automatic or manual mode and switch to the other.

The following information will be logged:

- The status and pressure readings of the compressor.
- The water temperature of the compressor.
- The temperature of the compressor.
- The temperature of the He-7 cooler.
- The temperature of the Bluefors cryostat.
- The heater output reading of the He-7 cooler.

There will be the possibility to use Python scripts to:

- Give simple commands to the cryostat: set the temperature or set the voltage.
- Retrieve sensor values: get the temperature or get the voltage.

Non-functional:

- The program must be reliable. Since the cryostat can be active up to a month, the program must be stable during this session and must not crash or give any complications meanwhile.
- The core of the program must run on a single computer that uses Windows.
- This computer is connected to the cryostat which means that it must communicate with three devices.
- Since the cryostat is only a tool to perform experiments, the operating program must be simple. The program must therefore be able to perform actions with a few clicks.

Should have

The GUI will contain:

- Tabs which zoom in on the information about a certain component of the cryostat:
  - A main overview of the cryostat as described in the Must have section.
  - A detailed overview of the He-7 cooler.
  - A detailed overview of the compressor.
  - A detailed overview of the Bluefors cryostat.

The automatic control mode can:

- Have a delay built in for the cooling down process of the whole cryostat.
- Have a delay built in for the recycling process of the He-7 cooler.

Could have

The GUI will contain:

- The option to drag the tabs of the GUI to create a new window.
  - There could also be the option to place these windows back in the main window.
- Notifications when to start the cooler.
- An overview of the pressure per area in the compressor.
- Active heater voltage control to reach a set temperature point.
- Errors messages to indicate when an error occurs either in the program or with the cryostat.

Won't have

The automatic control mode cannot:

- Control and monitor the vacuum pump.

4.2. Concepts

Before developing the program, four different design concepts are made, which all fulfill the requirements. The first three concepts make use of a server/client model as network architecture with different interpretations of the client. The fourth concept is a program on one computer without a network architecture. The decision which concept to use is discussed with SRON.

Server/Client model

Concept 1: Python client and a GUI client

A server runs on a computer which retrieves all data from the different devices. A client can access this data real time by means of a network layer and control this in a GUI or with a Python script.

Advantages:

- Program has an API which can be used for multiple applications and programming languages.
- Easy control of data for Python.
- The program itself can be used from an external computer.

- The server could be accessed from different devices on the same time.

Disadvantages:

- There is an additional network layer between the data retrieved and the GUI.

Concept 2: Python client and a web based GUI Client

This concept is the same as concept 1 with the difference that the GUI is not a native application, but a web client instead.

The advantages and disadvantages are basically the same as concept 1, though instead of developing a program for every operating system, a single web client can run on any operating system because it can run in a browser.

Concept 3: Python client and an integrated GUI attached to the server.

This concept is the same as the previous concepts with the difference that the GUI client is integrated into the server, which means that the GUI and the server are embedded in one program.

The advantages of this concept to the other two is that there is one program (server and client as one) but can still be extended with other clients. The disadvantage is that the client, which will be developed, always has to be on the same device as the server. If the client is wanted on a different device as where the server is running, a new client should be created.

No network architecture

Concept 4: GUI with an extension for Python.

One program with a GUI retrieves and interprets all data retrieved from the different devices. The program has the option to run a Python script which can access data from the program. This could be done with the .NET Python integration IronPython [6].

Advantages:

- No complication of making the program work over a network.

Disadvantages:

- IronPython is an open source project, difficult to estimate if it will fulfill the requirements.
- Program can run only on one operating system.
- Program can only run on the computer connected to the measurement devices.
- There is no API for creating other clients.

Conclusion

After discussing the concepts with SRON, it is concluded that a server based program is preferred. This is mainly because SRON likes the idea of both concept 1 and 2 to have the option to create a client for multiple applications and programming languages. A combination of these two concepts would be nice, though for time considerations the focus is on one concept. If SRON has to choose between a web based and a native GUI, a native GUI would get preference because they are used to work with that. Therefore concept 1, a server with an API for both a Python and a native GUI client, will be a requirement and will be implemented. The advantage of this concept is that SRON will be able to extend the program, so they are able to make an additional web based client or a client for another operating system later on.

5

Software Design

In this chapter the software design is discussed. First the architecture of the program will be explained. Secondly the choice of programming language and GUI will be substantiated. Finally the tools, which are used for developing,are mentioned.

5.1. Architecture

The core of the program that will be developed consist of a server that retrieves and sets data from multiple devices. This server contains an API that can be used for setting and getting information. An overview of the architecture can be seen in (figure 5.1). This server client model is the most commonly used form of network architecture [7]. An advantage of this architecture is that the server handles all data and the clients only need to give commands to set or retrieve certain data. As a result of this the logic of the program will be in the server part.

The server runs on the main computer and handles the received information from the three devices. The compressor will be connected with an Ethernet cable, the temperature controller of the PTR will be connected with a USB connection and the He-7 cooler controller is connected with a Ethernet cable as well.

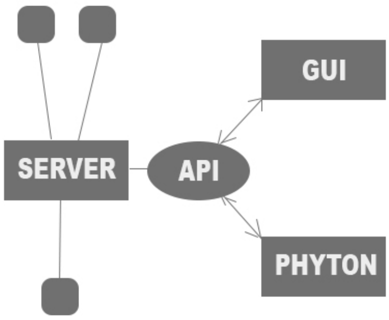


Figure 5.1: Overview of the use of the server

5.2. Programming language

To choose a programming language for this project several commonly used languages, which allow the implementation of the desired architecture, were considered. These languages are compared against four important functions, explained below.

First off we evaluate the languages capability to create a GUI. Most of the considered languages have the option to choose between several GUI frameworks. For comparing reasons the most applicable option(s) are considered. The program also has to communicate with three devices that use different communication methods. Therefore the language should be able to handle Ethernet connections as well as connections over a USB port, we evaluate this in the column networking. Since the time of the project is limited, efficiency is important. To evaluate this, the amount of code required to reach our goals is compared. Finally reliability is important, to guarantee this the program will have to be tested. The language will have to support this as well. The trade-off can be seen in table 5.1.

After evaluating all languages we decided to use C# to write the server and the client. C# is overall a good programming language and does not score negative on any of the functions. The GUI works well on a Windows PC, the communication with the devices is native supported, it is efficient and is a good option for testing. Java also scores good on most points, but from previous experience the choice for C# is made since it has proven to work well.

	C#	**Java**	**C++**	**Python**
GUI	Can use .NET which does work great on windows	Works fine	Can use .NET, works less than with c#	Not very suitable for building a GUI
Networking	Native support (.NET)	External libraries for COM port	Native support (.NET)	Simple library
Amount of code required	Low amount of code	Average amount of code	Large amount of code	Very low amount of code
Testing	Average testability	Good testability	Bad testability	Good testability

Table 5.1: To choose a programming language for this project, we rate 4 common languages based on the main functions we will use in our project: The ability to create a GUI interface, the ability to communicate with an external device over Ethernet or a USB port, the amount of code required to create a feature and the ability to test the code for bugs and confirmation to specification.

5.3. Architectural Software Pattern

The GUI will be based on the Windows Presentation Foundation (WPF) framework. This framework works well with C# and the development tools that will be used. It allows applications to use the MVVM pattern [8], which separates the data and business logic (Model) from the GUI (View), and creates bindings between them (View Model). Their state is kept consistent by using observers and notifications to change the state between these components. This combination cannot be done with any other GUI framework.

5.4. Tools

To guarantee that the product fulfills our expected quality quality and programming standards, multiple tools will be used. These will increase important aspects like the readability, understandability and consistency. The tools that are going to be used are:

- GitHub: A management tool that allows the user to control the written code. GitHub also allows its users to communicate and store documentation.
- Travis [9]: A continuous integration tool that verifies each checkin by an automated build.
- Visual Studio 2017 [10]: An integrated development environment that helps writing code efficiently and accurately.
- Microsoft Unit Testing Framework: A unit testing framework, supported by Visual Studio 2017.
- StyleCop [11]: A code analysis tool that runs against C# source code. It aims to improve readability, consistency and maintainability of the code.
- GhostDoc [12]: A generation tool that is allows its user to create documentation in XML format.
- CodeMaid [13]: An open source Visual Studio extension to cleanup and simplify C# coding. It's features include cleaning, reorganizing and formatting.
- Resharper [14]: A Visual Studio extension that helps refactoring, analyzes code quality, eliminates errors and more.

6
Conclusion

In this document it was researched how to increase the ability to perform experiments with the cryostat by creating a program to operate it. The current programs do not suffice anymore and therefore a new one is desired. The main problems that arise are: The user of the program has a limited overview of the cryostat and they cannot retrieve the desired information from it. This decreases the efficiency of performing experiments.

To solve the described problem a new situation is presented. By making a program to control the cryostat from a single GUI, instead of operating several devices individually, the ability to perform experiments is greatly increased.

Lastly it was discussed that a server client architecture is the best option to implement the program. This will allow SRON to communicate with the cryostat through both a GUI and through Python scripts. All of this will be implemented in C# since it has no downsides and has proven well to work when creating a server client architecture with a GUI. Finally the WPF framework will be used for creating the GUI since it allows the implementation of the MVVM design pattern.

Bibliography

[1] R. Bhatia, S. Chase, S. Edgington, J. Glenn, W. Jones, A. Lange, B. Maffei, A. Mainzer, P. Mauskopf, B. Philhour, and B. Rownd, *A three-stage helium sorption refrigerator for cooling of infrared detectors to 280 mk,* Cryogenics **40**, 685 (2000).

[2] Chase Research Cryogenics, *Two-stage sub-kelvin ^{3}He cooler instructions,* Included in appendix A.

[3] *lakeshore 335 datasheet,* http://www.lakeshore.com/Documents/335.pdf, accessed May 4, 2017.

[4] *Agilent 34972a user manual,* literature.cdn.keysight.com/litweb/pdf/34972-90010.pdf, accessed May 4, 2017.

[5] Q. Ma, *The effectiveness of requirements prioritization techniques for a medium to large number of requirements: A systematic literature review,* (2009).

[6] *Ironpython,* Http://ironpython.net.

[7] M. D. Hanson, *The client/server architecture,* Server Management , 3 (2000).

[8] E. Sorensen and M. Mikailesc, *Model-view-viewmodel (mvvm) design pattern using windows presentation foundation (wpf) technology,* MegaByte Journal **9**, 1 (2010).

[9] *Travis,* Https://docs.travis-ci.com/.

[10] *Visual studio 2017,* Https://www.visualstudio.com/.

[11] *Stylecop,* Https://stylecop.codeplex.com/.

[12] *Ghostdoc,* Http://submain.com/products/ghostdoc.aspx.

[13] *Codemaid,* Http://www.codemaid.net/.

[14] *Resharper,* Https://www.jetbrains.com/resharper/.

Appendices

A

He-7 Cooler Manual

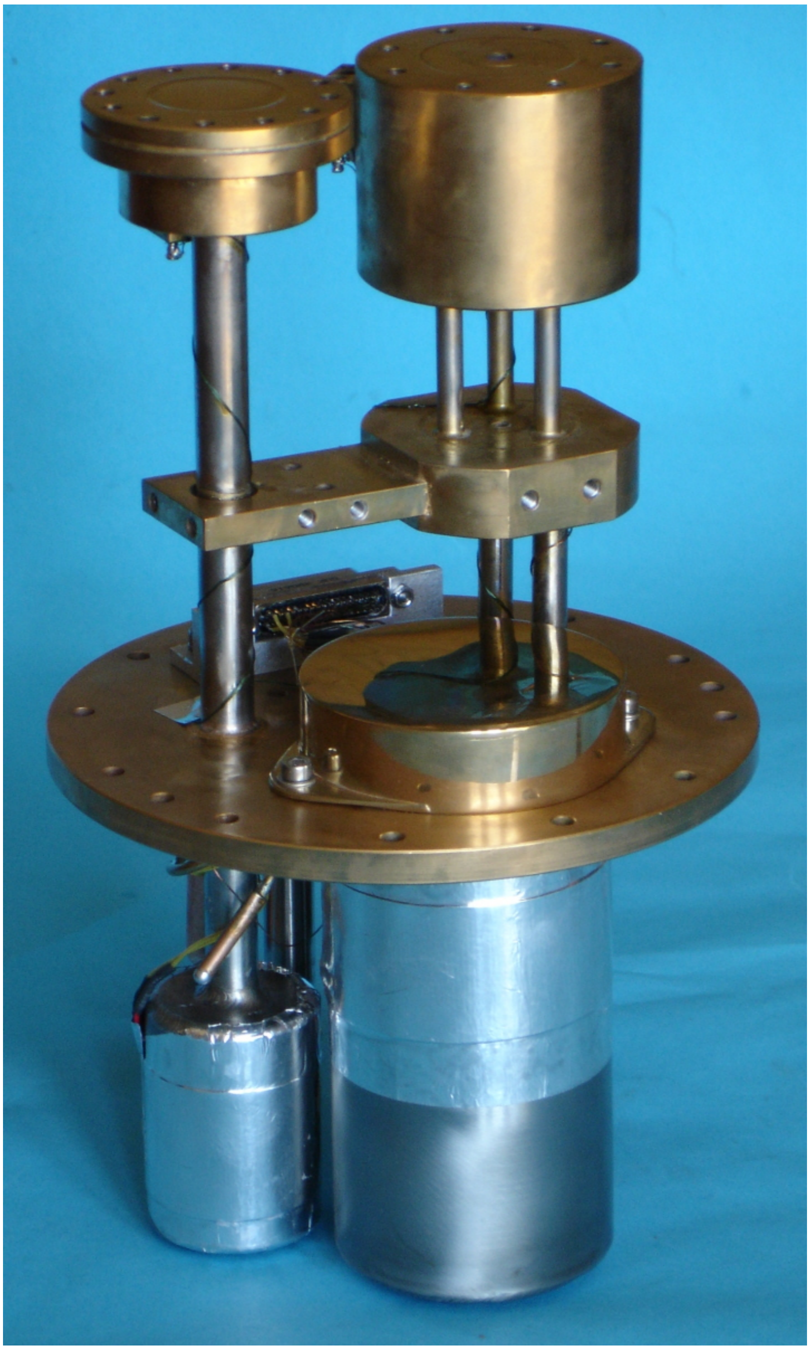
CHASE RESEARCH CRYOGENICS LTD.
WORLD LEADERS IN SUB-KELVIN CRYOGENICS
'UPLANDS', 140 Manchester Road, Sheffield S10 5DL
Tel. & Fax: +44 (0) 114 268 0672
Director: Dr. S.T. Chase. Secretary: Dr. L.C. Kenny
Registered in England & Wales, No. 4643351
VAT registration No. GB 763 8558 84

Date: 4th July 2010

TWO-STAGE SUB-KELVIN ^{3}He COOLER

No. CRC-7B-002.

INSTALLATION AND OPERATING INSTRUCTIONS.



GENERAL HANDLING.

WARNING!

THIS UNIT CONTAINS HELIUM GAS AT HIGH PRESSURE.

Do not crush, twist or bend the unit. Avoid applying mechanical stresses. Do not heat the unit above room temperature. Keep in a sealed cryostat, or in the shipping box and brace in which it came.

Do not hold or lift the unit by means of the cold heads.

Avoid the use of acid fluxes when soldering in the vicinity of the cooler. Chloride based fluxed will corrode stainless steel.

After unpacking the unit according to the instructions supplied, the unit should be immediately transferred into the host cryostat. The shipping brace doubles as a stand for the unit, though when used as a stand, the three screws through the aluminium plate into the cold heads should NOT be in place. When picking the unit up, it should be firmly grasped by the cryopump radiation shield or the main plate/angle bracket.

SAFETY OF CHASE RESEARCH CRYOGENICS PRODUCTS

1. Pressure Equipment Directive 97/23/EC (Pressure Equipment Regulations 1999)
This CRC cooler unit is manufactured in accordance with Sound Engineering Practice. The volume and gas pressure within the cooler are such that the equipment falls below the lower classification limit in Annex II of the Pressure Equipment Directive. Hence the requirements for Conformity Assessment do not apply and no Declaration of Conformity can be made, or CE marking applied.

The cooler is covered by Article 3 Paragraph 3 of the Pressure Equipment Directive, which states: "Pressure equipment and/or assemblies below or equal to the limits in sections 1.1, 1.2 and 1.3 and section 2 respectively must be designed and manufactured in accordance with the sound engineering practice of a Member State in order to ensure safe use. Pressure equipment and/or assemblies must be accompanied by adequate instructions for use and must bear markings to permit identification of the manufacturer or of his authorized representative established within the Community. Such equipment and/or assemblies must not bear the CE marking referred to in Article 15."

2. Pressure Systems Safety Regulations 2000

This cooler unit does not contain a pressure x volume product exceeding 250 bar-litres hence PSSR regulations 5(4), 8-10 and 14 do not apply. This means that the system does not require a written scheme of examination. The cooler is not 'mobile' in the sense intended in the PSSR hence *the owner* has duties under these regulations to ensure that a) the safe operating limits are not exceeded; b) the unit is operated in accordance with these instructions; c) the unit is returned to Chase Research Cryogenics Ltd in the event that any maintenance is required. The cooler contains no user-serviceable parts.

3. Safe Operation
The safe operating temperature range of this cooler is 0 to 320 K.

4. Risk Assessment

CRC coolers contain Helium gas under pressure. The stored energy of the system is between 26 and 28 STP litres at 1 bar. All system components are integrity tested during manufacture; the slightest leak will make the cooler lose its stored gas and cease to function. A unit that has leaked presents no risks whatever to the user; the following risk assessment applies therefore only to functional units.

4.1. Hazards and consequences

Accidental damage to the cooler unit could result in the sudden release of pressurised gases, causing mechanical failure of the unit and potential injury (or damage to surrounding instruments) from ejected debris.

Possible events leading to failure are: overheating of the unit, for example in a fire; dropping or crushing of the unit; twisting or bending of the gas tubes. Mechanical damage to the unit is most likely to occur during assembly of the instrument of which the cooler forms part.

4.2. Risks without controls in place

It is extremely unlikely that the above events will lead to danger. Chase Research Cryogenics Ltd has produced more than one hundred cooler units of various designs, which are in use for a range of applications worldwide. To date there has never been a sudden failure of a cooler unit – indicating that with normal use (including inevitable handling mishaps) the units have an excellent safety record. User experience to date shows that accidental mechanical damage to cooler units is likely to result in slow leaks, not sudden failures.

4.3. Controls in place

The controls that are in place to eliminate (as far as reasonably practicable) the risks arising from mechanical damage to a cooler unit are:

- This written instruction manual, containing warnings about the potential risks arising from damage to the unit and alerting the user to more risky operations;
- Instructions that the unit should not be used if it has been subjected to overheating, dropping, crushing, bending or twisting;
- A warning label on the transit box that the instructions should be read prior to handling the unit.

The applications for which cooler units are intended make it impossible to place warning labels on the unit itself. However if the cooler is incorporated into another instrument, this instrument should carry a warning label to alert the user that the cooler contains no user-serviceable parts and should not be disassembled.

4.4. Risks with controls in place

Providing users read and follow this instruction manual the risks are negligible.

INSTALLATION.

This unit is designed to work in either ‘wet’ dewar using liquid ^{4}He, or in a ‘dry’ dewar, i.e. from a mechanical cooler head such as a pulse tube. Test runs at Chase Research were performed in a wet dewar. During the test runs at Chase Research, the fridge was mounted vertically from the ^{4}He cold plate on a set of three legs, two of which were stainless steel, and the other was a 1/4” by 2” OFHC copper plate, all of length 112mm. The copper leg provided adequate thermal contact to the cold plate.

Mounting holes are provided on the fridge mainplate, for attaching the fridge to your dewar cold plate. There are twelve 4.1mm diameter (M4 clearance) holes symmetrically distributed upon a 114.3 mm (4.5”) pitch circle around the periphery of the circular main plate. In addition to these, there are also two M4 clearance holes on the mainplate periphery, and a further row of 4 clearance holes, at ½” centres, in front of the electrical connector.

Always use spring washers, or suitable low expansion washers (e.g. Invar or Tungsten), under every bolt head. These will take out differential thermal contraction that might otherwise cause loosening of the bolts, and thus compromise thermal contact.

All electrical connections are brought out to a 37-pin MDM-SSP connector mounted onto the main plate. Pin-outs are listed in the appendix. Wiring for the RuO_2 sensors on the cold heads is carried directly to the 37-pin connector. The RuO_2 resistance thermometers should be excited by AC currents no greater than 1μA.

To begin the cooling cycle, once the mainplate and cold heads are at around 4.2K, it is necessary to warm the ^{4}He cryopump to around 60K and the ^{3}He cryopump to 40 to 50K. A heater current of up to 80 to 100mA or so is necessary to heat the ^{4}He cryopump rapidly, and around 50 to 60mA for the ^{3}He cryopump. The pump heater impedances are about 200Ω for the ^{4}He cryopump and about 400Ω for the ^{3}He cryopump. Try to ensure that the lead-in wiring to these heaters is not unduly dissipative.

The diode thermometers require excitation with currents of 10μA DC. All together there are 6 diode thermometers installed on this unit.

Both cryopumps are provided with small gas-gap heat switches that are activated by 10 kΩ heater resistors, and each carries a diode thermometer. As supplied, these heat switches require about 3 or 4 V to keep them on at around 20 to 25 K, and cool to the OFF state ($T < 8$ or 10 K) in ten to fifteen minutes. These switches can be partially turned on ('soft' turn-on) with 2 to 2.5V). There should be no need to touch the heat switches or heat straps during installation or normal operation of the fridge. The heat switches can be easily damaged, and if bent or twisted are likely to fail.

ATTACHING YOUR EXPERIMENT TO THE FRIDGE.

This model of cooler provides three points at which heat may be extracted from an experiment mounted on a separate cold table. They are the ^{3}He cold head, the ^{4}He 'buffer' head, and the ^{4}He film burner (the copper platform that sits between the heads and the main plate). In fact, to achieve optimum performance, only a very small load should be applied directly to the ^{3}He cold head. The main source of cooling power is the film burner, which can sustain a thermal load of at least 250mW at a temperature of less than 1K. The unit is supplied with an OFHC copper heat strap attached between the ^{4}He 'buffer' head and the ^{4}He film burner. This is installed to ensure that the film burner temperature is kept as low as possible, to reduce the parasitic load on the ^{3}He cold head as much as possible, and thereby allow the lowest temperature to be reached. It may be removed or replaced/relocated if desired, but an increase in the ^{3}He cold head base temperature may result.

The top surface of the ^{4}He 'buffer' head has 8 holes tapped M4 on a 40mm P.C.D., and a further axial hole tapped UNC #6. The ^{3}He cold head has 12 tapped holes, again M4, on a 38mm P.C.D. The film burner has 6 M4 tapped holes on the main body, in pairs on each of the three free sides, and 6 more on the extension bracket that connects to the ^{3}He gas pipe.

Both of the cold heads, the film burner, and any cold table/experimental equipment/detector assembly must be properly radiation shielded at around 4K, in order to achieve sub-Kelvin operation. Any ancillary support structure (cold table) and experimental wiring looms must be thermally sunk to the film burner at some point between the ^{3}He cold head and the 4K plate, if a satisfactory operating temperature is to be reached. Temperatures in the range below 300mK are only achievable if the total thermal load on the ultra-cold head is kept to a minimum (below about 8 or 10 μW). The film burner is designed to buffer the parasitic loads due to wiring and mechanical support structures. No other mechanical attachments to the fridge unit are necessary, in order to achieve satisfactory operation.

While fixing experimental equipment to the cold head, extreme care should be taken, not to torque or bend the gas pipes. Always support the cold heads against the applied torque.

Under no load, the ^{3}He head will run at about 240mK, the ^{4}He ‘buffer’ head at about 850mK, and the film burner also at about 850mK. Run time is limited by the buffer head, which will last about 19 hours under no load. The ^{3}He head will continue to run at about 300mK for some time after the buffer head has expired. When loads are applied, the heads and film burner naturally run warmer. Load data for the Buffer head is included in the accompanying Excel data file. Note that the test run was performed with a thermal short (OFHC copper heat strap) inserted between the buffer head and the film burner. This arrangement is likely to result in the lowest attainable base temperature at the ^{3}He cold head.

OPERATION.

PRE-COOL TO L^{4}He.

In a wet dewar, the bare unit will cool to around 80 K in 4 to 5 hours, once LN_2 is introduced into the cryostat. During L^{4}He transfer, the cold heads cool rapidly while the cryopumps are hotter than about 25 or 30K, but will cool very slowly after the cryopumps cool to below 25 K, and the gas is adsorbed into them. If left like this, the heads will take two or three days to cool. The key to a rapid cooldown is to keep both of the cryopumps warmer than 25 K until the heads are cold. If L^{4}He is transferred at a normal rate until the main bath is full, the cryopump heat switches may not turn off until after the cryopumps are cold. In this case it will be necessary to apply heat to both cryopumps at the rate of about 50 mW each, in order to keep them warm while the heads cool. An alternate strategy is to do an initial partial transfer at a placid pace, stopping the transfer when the cryopumps reach about 20 K, and restarting an hour or two later. The optimum strategy and timings will naturally depend upon the thermal loads and masses connected to the cooler unit. Figure 2 shows the cooldown from LN_2 to L^{4}He. In the first test run, only the ^{3}He pump cooled down below 25K before the corresponding heat switch had turned off, but I re-heated both pumps to around 40 or 50K while the heads cooled down, so that I could commence the cycle immediately.

When pre-cooling with a mechanical cooler cooling timescales will be similar to the above, unless limited by the cooling rate of the host device. The cryopumps should be warmed back up once their corresponding heat switches have turned off.

Because the cooling down of the heads depends upon gas convection, the fridge should be kept close to vertical during the cooldown process.

RUNNING THE COOLER.

It is rather easy to get this unit to run, but it takes practice and some experimentation to achieve the best possible performance. Your particular experimental configuration will affect the thermal loadings on, and conductances between, the various parts of the fridge, and may consequently alter the optimum mode of operation. I shall describe the method of operation by presenting a log of my most satisfactory test run, full data from which may be found on the accompanying disc. I recommend that variations on this scheme be tried out, once some familiarity with the successful operation of the unit has been gained.

The scheme, broadly speaking, is first to heat both pumps up to between 40 and 60 K, and to maintain them at this temperature while ensuring that the ^{4}He liquefaction point (in this model, the main plate) cools to below the critical point of ^{4}He (5.2K). In order to get the unit to operate satisfactorily, it is crucial that high liquefaction efficiency is achieved with the ^{4}He stage. The colder the liquefaction point gets while the pumps are warm (particularly the ^{4}He pump), the higher will be the liquefaction efficiency, and hence the longer the fridge will run before needing to be recycled. You should aim for a ^{4}He liquefaction temperature of below 4.6K. You should try variations on the pump temperatures to find a regime giving the best performance for your particular experimental set-up. I have used different regimes for other fridges. Stabilising power for the ^{3}He pump should be around 60 to 90 mW (12 to 15mA heater current) at 40 or 50K

Once the ^{4}He charge is liquefied, the ^{4}He pump is allowed to cool by turning off the heater power and turning on the heat switch. This can be done more or less abruptly, depending upon the voltage applied to the switch heater. For operation in a dry dewar, I suggest a slow turn-on, so that heat is not dumped too rapidly from the hot pump. The only danger is that, if the main plate temperature should rise above about 12 or 15K, the other heat switch might turn on prematurely. The ^{3}He does not liquefy at the main plate, but at the film burner. The temperature of the main plate after ^{4}He liquefaction is not of direct relevance to the liquefaction efficiency of the ^{3}He charge.

Once the ^{4}He has been liquefied, and the ^{4}He pump is cooling, the cold head temperatures and film burner temperature will start to fall rapidly. The ^{4}He pump cooling curve will flatten below 25 K as the charcoal begins to pump on the ^{4}He vapour, and the head temperature will continue to drop. One can turn off the ^{3}He pump heater power once the ^{4}He pump switch is ON, and achieve satisfactory liquefaction efficiency by waiting to turn the ^{3}He pump heat switch ON until the head temperature has dropped to below 2K.

Once the ^{3}He pump has begun to cool, the ^{3}He cold head will also cool rapidly. Final stabilisation at the operating temperature will take some time, though how long will depend upon the thermal loads that are applied to the heads by your experimental arrangement. In general, lower loads result in lower running temperatures and these require longer to achieve stabilisation. The ultrahead in particular can take some while to stabilise, particularly with applied loads of less than 1μW or so. This is because the liquid ^{3}He has a high specific heat capacity compared to the rate at which gas evaporation at very low vapour pressure can extract latent heat. The lower the final temperature, the lower will be the corresponding saturated vapour pressure, and thus the rate at which gas evaporates.

When the fridge is running, the heat switches for both pumps should be kept on, or the pumps will warm up a bit and this will result in higher head temperatures.

Annotated logs of two CRC-7B-002 test cycles.

Begin at t = 0, all timings in minutes. This test run commences with the L^{4}He transfer, and everything at about 77K. The ^{4}He pump switch turned OFF while the ^{4}He pump was still quite warm, but the ^{3}He pump was quite cold before its switch turned OFF. This is why more heater power was required to warm the ^{3}He pump than the ^{4}He pump, on this occasion. See Figure 2.

t = 2. Start L^{4}He transfer.

t = 29. ^{4}He pump switch turns OFF.

t = 35. ^{3}He pump switch turns OFF. Apply 60mA to ^{3}He pump heater.

t = 36. Apply 50mA to ^{4}He pump heater.

t = 37. Reduce ^{3}He pump heater current to 15mA.

t = 42. Reduce ^{3}He pump heater current to 12mA.

t = 53. Increase ^{4}He pump heater current to 70mA.

t = 59. Reduce ^{4}He pump heater current to 20mA, and ^{3}He pump heater current to 10mA.