Actualization of a Valve Sub-system for Localized Heart Cooling

Volume I

Team Heart: Benjamin Anderson, Christopher Fultz, Andrew Penning, Benjamin Simmer, and Joel Scheumann
Advisor: Pramote Hochareon
Client: Nirva Medical
May 7, 2013
1. Executive Summary
In order to treat heart attacks, Nirva Medical is developing a localized heart cooling system, meant to cool the heart while allowing the rest of the body to remain at temperature which will not induce shivering. As part of this process, Team Heart was tasked with building a valve sub-system to control the timing of the flow into and out of the heart. Our prototype will be used in an upcoming animal study, in conjunction with the rest of Nirva Medical’s system.

The valve sub-system reads in the timing of the heartbeat from Nirva Medical’s system, along with values selectable by a user to choose the amount of perfusion into the heart, drainage out of the heart, and how long to pause between having either valve open. The heartbeat is read in as a square wave from an aortic balloon pump. The user inputs the desired flow settings using a dial, with an LCD displaying the settings. There are two switches that will hold open the valves to facilitate tubing changes and as emergency safeguards.

**Figure 1.1: User Inputs**

In addition to controlling the flow, the sub-system houses the valves, controller, circuitry, and method of user input. This is done in a way that protects the components during use and transportation, as well as provides safety to the user. Cables between the aortic balloon pump, user interface housing, and valves were made to be common detachable connectors. This allows for easy system assembly and disassembly, easy system transportation, easy cable replacement, and selectable lengths of cable for different situations. Cost was a significant factor in the building of the prototype, as well as feasibility of design and implementation over a period of one semester.

**Figure 1.2: Connectors**

The resulting prototype is fully functional and able to be used in the animal study. The controller inside the housing has a permanent access cable, so the software can be easily updated if required in the future. Both the valve housing and user interface housing have cooling fans to ensure that all components will remain cool and operate normally even under sub-optimal conditions.
# Table of Contents

1. EXECUTIVE SUMMARY ........................................................................................................... 0

2. MEMBER CONTRIBUTIONS ................................................................................................... 3

3. PROBLEM DEFINITION ............................................................................................................ 3
   3.1 PROBLEM SCOPE .................................................................................................................. 4
   3.2 TECHNICAL REVIEW ............................................................................................................. 4
   3.3 DESIGN REQUIREMENTS ..................................................................................................... 6

4. DESIGN DESCRIPTION .............................................................................................................. 7
   4.1 SUMMARY OF THE DESIGN ................................................................................................. 7
   4.2 DETAILED DESCRIPTION ..................................................................................................... 8
   4.3 ADDITIONAL USES .............................................................................................................. 13

5. EVALUATION .......................................................................................................................... 14
   5.1 EVALUATION PLAN .............................................................................................................. 14
   5.2 EVALUATION RESULTS ....................................................................................................... 14
   5.3 DISCUSSION ......................................................................................................................... 20
2. Member Contributions

For this project, Benjamin Simmer was the project manager. He was responsible for communicating the team progress to our advisor, as well as communicating any general requests from our advisor to the team. Ben ran meetings, providing the agenda as well as meeting summaries after meetings were done. He, along with our advisor, oversaw interaction between team members. Having a valve housing, user interface housing, and wiring components to work on, made it key that all members were making adequate progress, and Ben did this. He made significant contributions to the team’s assignments, including outlines, diagrams, and writing. He routinely coordinated with team members to compose progress updates which he conveyed to the advisor. Many times he met one-on-one with the advisor to discuss the project progress. He also maintained the team website and the budget.

Benjamin Anderson worked with Andrew Penning to complete the controller and wiring part of the project. He completed most of the soldering, circuit assembly, and final assembly of the valve housing and user interface components and the electrical system. He assisted Andrew with programming troubleshooting. Ben also worked with testing the valves and controller to ensure that they were connected and running properly. This included initial valve functionality tests and flow tests requested by the project’s advisor. He was also the one who contacted the valve manufacturer through the semester with the different questions that arose. Ben performed a patent search looking for designs that were similar to the scope of this project and made the initial WBS and Gantt chart.

Andrew Penning worked with Benjamin Anderson to wire and program the electrical components of our project. Andrew worked with the Arduino Mega to design the software necessary to control the valves as well as read in user input. Andrew worked on testing the valves to make sure they operated as desired for all inputs. He also ran ANSYS simulations on the valves and user interface housing to check for potential overheating issues.

Joel Scheumann was responsible for designing and building the valve housing. He picked the final valve housing concept and created a CAD model for it. Joel machined the plates and bars in the machine shop and assembled the components together. He was responsible for purchasing the screws and bolts needed for assembly. In addition to working on the valve housing, Joel contributed to Volume II in the sections of Bill of Materials, part drawings for user interface, assembly instructions for valve housing, and P2D flow settings. Finally, Joel located an IR camera for testing overheating of valve housing.

Christopher Fultz designed and machined the user interface housing. He chose the final design and built a CAD model of the housing, along with simplified models of the components to be housed. Chris then machined the aluminum for the user interface housing in the student machine shop. He helped assemble the user interface housing. Also, Chris researched the design for the user interface housing. Additionally, Chris did extensive editing to both volumes of the design report.
3. Problem Definition
3.1 Problem Scope
Nirva Medical is developing a localized heart cooling system, which will be used to treat patients following heart attacks. Before this project, Nirva Medical’s localized heart cooling system lacked a valve sub-system—including flow loop, control, and user interface—which is the problem that Team Heart solved. This was a problem because the valve sub-system controls the flow entering and exiting the heart, which is used to cool the heart. Without such a valve sub-system, the localized heart cooling system would be ineffective and, possibly worse, harmful to the patients who would be treated. The key customer is the CEO of Nirva Medical, Pramote Hochareon.

3.2 Technical Review
According to the American Heart Association, deaths linked to heart attacks claimed approximately 125,000 lives in the United States in 2009. In addition, the number of individuals who suffered heart attacks in 2009 totaled about 715,000 [1].

According to the National Heart, Lung, and Blood Institute, a heart attack occurs when blood flow to the heart is blocked [2]. If this blockage of blood is not removed quickly, part of the heart muscle dies because of the lack of oxygen. Heart attacks commonly have one of two causes: buildup of plaque in coronary arteries—arteries that supply blood directly to the heart—or a severe spasm of a coronary artery. Heart attacks are primarily caused by the blockage of coronary arteries. According to the A.D.A.M. Medical Encyclopedia, this blockage is usually caused by a substance called plaque, which slowly blocks the blood flow in coronary arteries [3]. The blockage of these arteries increases when the plaque splits open, causing blood platelets to form a clot, obstructing the blood flow. The blockage in blood flow results in a heart attack. In addition to contributing to the formation of clots, plaque may also cause a heart attack by obstructing the oxygen supply to the heart muscle. However, a pure buildup of plaque is less common than formation of clots. Below, Figure 3.2.1 shows a diagram of a heart attack resulting from the clogging of coronary arteries.

![Figure 3.2.1: Heart Attack Diagram](image-url)
In Figure 3.2.1, one can see an overview of the heart with the heart muscle and coronary arteries. Also, Figure 3.2.1 shows a more detailed view of a coronary artery describing how a buildup of plaque, coupled with the formation of a clot, blocks oxygen-rich blood flow to the rest of the heart muscle, causing the heart muscle to die.

Heart attacks are also caused by a spasm of the coronary artery. Spasms of the coronary artery can happen in arteries where no atherosclerosis—blockage with plaque—occurs. These spasms constrict blood flow by a tightening of the coronary artery.

Currently, methods of heart attack treatment include thrombolysis, coronary angioplasty, coronary artery bypass graft surgery, atherectomy, cardiomyoplasty, heart transplant, and implantation of a stent. In many cases, a combination of these methods can be used [5].

Thrombolysis is the use of drugs to break down blood clots. These drugs are administered after arriving at a hospital. A possible complication from these drugs is excessive bleeding, resulting from the fact that the drugs thin the blood [6].

Coronary angioplasty involves using a small balloon to widen the arteries and improve blood flow. This is a temporary procedure, and according to the Mayo Clinic, it usually includes the use of a stent to hold the artery open after the balloon is removed. A stent is a small, stiff tube inserted in the arteries [7].

Coronary artery bypass surgery is a procedure in which a vein or artery from another part of the body is used to bypass blocked coronary arteries by grafting. The new vein or artery is grafted to form a link between the aorta and the coronary artery, bypassing the blockage [8].

Coronary atherectomy is a procedure where a catheter with a small cutting tool at the end is inserted into the coronary artery and any blockage is cut away and removed from the artery. This procedure can be used in combination with a stent [9].

Cardiomyoplasty is the process of taking muscle from another part of the body and wrapping it around the heart. This muscle is then stimulated to help the heart pump. The extra muscle helps in cases of heart failure [10].

Recently, external cooling of patients after a heart attack has emerged as a possible treatment. This is done to prevent brain damage from lack of oxygen by lowering the metabolism of the cells [11].

Another promising method of treatment is the use of progenitor cells from bone marrow to improve heart function after a heart attack. In a study in the New England Journal of Medicine, the cells were injected into the bloodstream at the patients’ hearts. The patients who received the progenitor cells showed significantly more function in their left ventricles than the patients who received a placebo [12]. A progenitor cell, like a stem cell, is able to form more than one type of
cell. However, progenitor cells are more limited in what types of cells they can become and how many times they can divide [13].

Currently, localized heart cooling is not used as a treatment for heart attacks. Nirva Medical is in the process of making localized heart a viable treatment. Localized heart cooling involves sending cool blood to the heart, while warming blood flowing to the rest of the body. In this way, the heart can be cooled while the rest of the patient’s body remains at normal temperature.

The following is terminology used in the project: localized heart cooling refers to the nature of the cooling method in that the effects of the cooling are limited to the heart, perfusion refers to blood delivery to the body, drainage refers to the removal of blood from the body, systole refers to the contraction of the heart, and diastole refers to the expansion of the heart.

### 3.3 Design Requirements

1. **Cardiac Synchronization and Quantified Flow Response**
   - **Requirement**—Synchronization of the valves with the patient’s cardiac rhythm will be obtained and the response delay will be quantified
   - **Source**—Interaction with customers
   - **Importance**—Nirva Medical has a specific goal to control flow based on the patient’s cardiac rhythm
   - **Measurement**—Synchronization and response delay will be measured with a flow test

2. **P2D Ratio**—Ratio of perfusion to drainage into and out of heart
   - **Requirement**—System will have a minimum of 10 separate, selectable P2D Ratios
   - **Source**—Interaction with customers
   - **Importance**—It is vital that different ratios can be tested during the animal study so that proper flow ratios can be determined and the desired rate of cooling can be achieved
   - **Measurement**—Counting the number of function P2D ratios

3. **Valve Heating**
   - **Requirement**—Maximum surface temperature of valves will not exceed 50°C
   - **Source**—Mitigation of possible system failure and possible harm to operator
   - **Importance**—If the valves become too hot, they could potentially harm users as well as heat the blood
   - **Measurement**—A picture will be taken with an infrared camera to determine the maximum temperature after one hour of use

4. **Temperature of Controllers**
   - **Requirement**—Maximum surface temperature of the controllers will not exceed 50°C
   - **Source**—Mitigation of possible system failure and possible harm to operator
   - **Importance**—It is important that the controllers not overheat, and that operators will not burn themselves if they come in contact with equipment
   - **Measurement**—Temperature measurement will be obtained with an ANSYS simulation
4. Design Description

4.1 Summary of the Design

The valve sub-system we have designed is integrated with the existing flow loop of Nirva Medical’s localized heart-cooling device. The device function is to control the flow of blood into and out of the heart. This design incorporates a user interface, a controller, two valves, and housing for each component.

The controller is an Arduino microcontroller, and there are two main inputs into the controller. These inputs are the user-selected P2D ratio, which controls the flow intervals, and the square wave that is based upon the patient’s heartbeat. The P2D ratio describes the flow of the blood into and out of the heart. There are three numbers in the P2D ratio. The first number is how many heartbeats the valve controlling perfusion is open. The second number is how many beats both valves are closed. The third number is the number of beats the drainage valve is open. Each of these conditions occur sequentially, so there is never a situation where both valves are open at the same time. From these inputs, the Arduino then controls when and how long the valves are open and closed.

The user interface, which is mounted to the front face of the housing, consists of two variable knobs—which are potentiometers—four switches, and an LCD display. One switch controls which of the two knobs is being referenced for the preset P2D flow rate. A second switch is a system on/off switch. The other two switches will hold open the valves when they are flipped. The LCD displays the selected P2D setting.
4.2 Detailed Description

4.2.1 Functional Block Diagram

4.2.2 Functional Description

4.2.2.1 User Interface

The function of the user interface is to allow the user to select the operating settings of the valve sub-system. The user interface housing contains the power supply, transformer, trigger circuits, Arduino, and breadboards, and prevents unwanted contact between the user and circuitry. The housing also positions the dials, switches, and LCD for the user’s convenience. The power switch controls all power flowing into the circuits contained in the housing. In the current design, only one of the two dials is programmed. The resolution on the dials was higher than originally planned, so one dial was enough to satisfy our requirement. The second dial was still included in the design along with the toggle switch so that in future use, more P2D ratios can be added if needed. In this case, Dials 1 and 2 can be turned to adjust the P2D ratio. The toggle switch would select which dial is actively choosing a preset ratio. In the current design, one dial is used and can be turned to switch between the ten preset P2D ratios. The LCD displays the selected ratio, with a numerical value given for each the perfusion, drainage, and holding—elsewhere called delay—values. The hold open switches allow the user to open the valve electronically for ease of tubing changes or in an emergency. One switch is for the perfusion valve and the other controls the drainage valve.
4.2.2.2 Controller

For the controller program, the functional block diagram is shown in section 4.2.3.2. The first step is to power on the system. This means supplying power to the microcontroller, the valves, the cooling fan, and the components that are connected through the circuit. When all components are powered on, the program initializes all the variables. This includes the variables for user input, the LCD display, the heartbeat signal, and the output to the valves. As the software is run, the first step is reading the selected P2D ratio from the user interface into the microcontroller to determine what setting to use. Once the P2D setting is read in, the program jumps to the selected P2D ratio section of the algorithm and reads in the patient’s heartbeat. The heartbeat is read in from an aortic balloon pump. The aortic balloon pump outputs a square wave that corresponds to the heartbeat. There is a then a check to make sure that the heartbeat is low, meaning 0, or the bottom of the square wave. The reason for this is to accurately count the heartbeat for the P2D setting. Based on the P2D setting, the program counts the number of beats for perfusion into the heart. Once the perfusion heartbeat count is met, the program moves into the delay portion of the P2D setting. Some of the settings will not include a delay, and the program will switch the valves directly into drainage. Again, the program counts the number of beats for the delay once the desired number of beats is met and proceeds to the next part of the code. Once this happens, the program repeats again for the drainage following the similar system. The program then repeats to check if the P2D setting has been changed by the user. If the setting has not changed, the program repeats with the same P2D setting. If the setting has changed, another section of the program is run that has the counts for perfusion, delay, and drainage that correspond to the selected ratio.

In respect to the square wave, there are certain requirements that dictate the P2D ratio timing. The perfusion valve always opens when the square wave goes from low to high. The drainage valve is always opened when the square wave goes from high to low. The opposite is true for when the valves close. The perfusion valve closes when the square wave goes from high to low and the drainage valve closes when the wave goes from low to high. The P2D settings were programmed based on these requirements.

4.2.2.3 Valves

The purpose of the valves is to regulate the flow by pinching the tubing. Details of the specific valves used are in Volume II Section 2.1.3. When the tubing is pinched, the flow is unable to surpass the valve in the flow direction. When the valve is open, the flow proceeds though the valve unimpeded. Section 4.2.3.3 below shows the valve in the open and closed positions. The nominal position of the valves is closed, so they actuate only when an input voltage is recieved. For this reason, no additional power is needed when the valve is closed. The actuation is determined by the controller algorithm. Included in the valves are manual override buttons, which give the opportunity to open the valves if an emergency occurs.
4.2.3 Overview Drawings

4.2.3.1 User Interface

Figure 4.2.3.1: User Interface Overview Drawing

Figure 4.2.3.1 shows the front, top, and one side of the user interface housing. There is room for four different switches, two dials, and an LCD screen. The functions of these different components was discussed in Section 4.2.2.1.

4.2.3.2 Controller

Figure 4.2.3.2.1: Controller Overview Drawing
Figure 4.2.3.1.2 is a functional block diagram of the controller. This diagram was discussed in Section 4.2.2.2.

Figure 4.2.3.2: Algorithm Overview Drawing
The Algorithm Overview drawing gives a summary of the Arduino code that is used to control the system. This was discussed in Section 4.2.2.2 also.
The Controller Circuit Overview Drawing shows all of the physical circuit components of the system. There are two main parts of the circuit. The first is the power supply and valve part of the circuit. This can be seen on the left side of Figure 4.2.3.2.3. This part of the circuit contains a wall power input, a transformer, a 24V power supply, two fans, two switches, two control circuits and two Arduino inputs. The Arduino inputs provide a signal to the control circuits when each valve is supposed to open. The control circuits are from Acro Associates and are designed for the pinch valves being used. These control circuits output a high voltage (in this case 24V) to the valves in order to cause actuation. Once the valve is actuated, the control circuit steps the voltage down to a lower value when the valve is held open. The benefit of this is that it reduces the heat generated by the valves over time.

The second part of the Controller Circuit Overview Drawing is the Arduino user inputs. This can be seen on the right side of Figure 4.2.3.2.3. This part of the circuit contains an Arduino pin to connect all of the ground connections together, the BNC input from the Aortic Balloon pump, the two dials (potentiometers), a toggle switch, and the two valve switches. The purposes of these components were discussed in Section 4.2.2.1.
4.2.3.3 Valves

Figure 4.2.3.3 shows the two valve positions that were discussed in Section 4.2.2.3. It also depicts the manual override button. This button can be pushed and held down to keep the valves in an open state.

4.3 Additional Uses

Possible additional uses of this design include applications where it would be necessary to cool parts of the body based off patient cardiac rhythm. This design could also be used when it is necessary to control flow based on any type of input signal.
5. Evaluation
5.1 Evaluation Plan
5.1.1 Cardiac Synchronization and Time Response
One critical design requirement was that our system had the capability of synchronizing the flow with the patient’s cardiac rhythm. The valve system has the ability to read in the heartbeat from a patient in the form of a square wave and control the flow according to this input. Two flow meters along with data acquisition software was used to check for synchronization and to measure response time from the heartbeat input to the point of peak flow. More detail is included in Volume II, Section 3.1.1.

5.1.2 Variable P2D Setting
Another design requirement was that our system would have a variable P2D ratio. This is important because varying the P2D ratio is our main objective, which the animal study will seek to explore. To test this requirement, the P2D ratio was varied while the valve trigger and patient input were compared. More detail is included in Volume II, Section 3.1.2.

5.1.3 Heating of Valves
Ensuring that valves do not become too hot is important for three reasons. The first is we do not want to damage the valves by overheating them. The second is we do not want have a risk of hurting the user. The final reason that valve temperature is important is so that heat is not transferred from the valves to the fluid flowing in the tube connected to the valves. A test was conducted in which the system ran for over an hour at a setting of 2 Hz and the temperature of the valves with respect to time was measured with an IR camera. More detail are included in Volume II, Section 3.1.3.

5.1.4 Heating of User Interface Assembly
Inside of the user interface are the Arduino microcontroller, power supply, circuit board, and transformer. Many of these components generate heat, and it must be ensured that this heat generation does not damage the components after an extended period operation. A test was conducted in which the system ran for 2 hours at a setting of 2 Hz and the temperature of the user interface components was examined by touch. More detail is included in Volume II, Section 3.1.3.

5.1.5 Testing Continuation
More testing was done to meet our client’s other needs. These tests are discussed in Volume II.

5.2 Evaluation Results
5.2.1 Cardiac Synchronization and Quantified Flow Response
It is critical that the valve system is able to synchronize the flow with the cardiac rhythm. It is also important to quantify the response time of the flow in order to determine how well the flow synchronizes with the patient’s cardiac rhythm. A closed loop flow system was connected to the valve control system. Also connected to the flow system was a Transonic T206 flow meter, by means of an inline flow probe. This setup is depicted in Figure 5.2.2.1
Figure 5.2.1.1: Synchronization and Flow Test Setup

Figure 5.2.1.2 shows the relation of the perfusion and drainage flows to the square wave output from the aortic balloon pump for a 1Hz or 60 bpm rate.

The figure above clearly shows that the valve control system facilitates synchronization of the flow with the cardiac rhythm at a typical heart rate of 60 bpm.

The following table shows the measured flow response times of five selected response times. For longer delay settings, the opening flow responses times were found to be lower than those at shorter delay settings, which was expected because there were larger back pressure buildups. Also, the closing flow response times were all very near to 150ms, meaning that the flow shutoff time is independent of flow setting.
Table 5.2.1: Response Time At Selected P2D Settings

<table>
<thead>
<tr>
<th>P2D Setting</th>
<th>Average Time Open [ms]</th>
<th>95% CI Time Open [ms]</th>
<th>Average Time Close [ms]</th>
<th>95% CI Time Close [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-0-1</td>
<td>151</td>
<td>16</td>
<td>135</td>
<td>4</td>
</tr>
<tr>
<td>2-0-2</td>
<td>190</td>
<td>21</td>
<td>146</td>
<td>11</td>
</tr>
<tr>
<td>1-1-1</td>
<td>93</td>
<td>23</td>
<td>149</td>
<td>6</td>
</tr>
<tr>
<td>1-1-0</td>
<td>127</td>
<td>7</td>
<td>138</td>
<td>7</td>
</tr>
<tr>
<td>2-1-0</td>
<td>184</td>
<td>27</td>
<td>135</td>
<td>10</td>
</tr>
</tbody>
</table>

5.2.2 P2D Setting
The P2D ratio describes the flow into and out of the heart. In order for our client to test the localized heart cooling system in an animal study, our device must be able to change settings to find the best flow rate for heart cooling.

For this test, the controllers were hooked up to the valves, and the unit powered on. The LCD display was read as the dial was turned through all the settings. At each setting, the dial was released and the valves observed and compared to the square wave input using the same setup as the Cardiac Synchronization and Quantified Flow Response test. If the LCD output and valve timing matched the desired P2D setting, that setting was declared functional.

Table 5.2.2 P2D Settings

<table>
<thead>
<tr>
<th>P2D Setting</th>
<th>Test Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:0:1</td>
<td>Functional</td>
</tr>
<tr>
<td>2:0:2</td>
<td>Functional</td>
</tr>
<tr>
<td>1:0:2</td>
<td>Functional</td>
</tr>
<tr>
<td>1:1:1</td>
<td>Functional</td>
</tr>
<tr>
<td>2:0:1</td>
<td>Functional</td>
</tr>
<tr>
<td>2:1:1</td>
<td>Functional</td>
</tr>
<tr>
<td>2:1:2</td>
<td>Functional</td>
</tr>
<tr>
<td>2:0:3</td>
<td>Functional</td>
</tr>
<tr>
<td>1:1:0</td>
<td>Functional</td>
</tr>
<tr>
<td>2:1:0</td>
<td>Functional</td>
</tr>
</tbody>
</table>

As the above table shows, the ten settings that we tested were functional. These settings display a wide range of possibilities for flow. Our design requirement of having 10 selectable and functional P2D ratio settings is met.

5.2.3 Valve Overheating
One of the design requirements of this project was to ensure that the valve temperatures remain in a reasonable range. The valves are capable of becoming too hot to comfortably touch after extended operation. It was also desired to make sure that the valves will not overheat and cease operation. With a cooling fan attached, the valves were run for 90 minutes at an average
frequency of 2 Hz. A picture was then taken with an infrared camera that was provided through our advisor.

**Figure 5.2.4.1: Valve Temperature**

The maximum valve temperature according to this picture is 27.1°C. This is well below our target of 50°C. With the fan attached, high valve temperatures will not be an issue with our system.

An ANSYS simulation was also run to calculate the valve temperature at steady state. The results can be seen below in Figures 5.2.4.2 and 5.2.4.3.
From the fan specifications, it was possible to estimate a mass flow of 0.016 kg/s through the housing and a heat flux of 2500 W/m² was assumed for this simulation. The estimates used were conservative, and should provide a worst-case scenario temperature.

The temperature found from the IR Camera was lower than the value predicted by an ANSYS simulation. The ANSYS results show that our surface temperature is 38ºC, significantly higher than the value given by the infrared camera. This could be due to the assumptions we made about the amount of heat going into the valves, as well as the flow profile.
5.2.4 User Interface Overheating
One concern for the user interface housing was that placing electrical components inside the housing would restrict airflow and cause overheating. We placed a fan inside the housing, and made large vent slots in the ends of the housing.

Ideally, we would have ascertained the internal temperature using the infrared camera. However, the view into the housing was obstructed by the fan and plastic placed in the housing to insulate the aluminum from the electricity in the controllers. We used an ANSYS simulation to determine the internal temperature of the power supply, which was observed to be the hottest component when the circuits were not in the housing. The maximum temperature from ANSYS was 29°C.

![Figure 5.2.4.1: User Interface Housing Temperature](image1)

The maximum temperature from ANSYS is significantly lower than our maximum acceptable temperature. Even if there is significant error, we are confident that overheating will not be an issue for the components in the user interface housing.

![Figure 5.2.4.2: User Interface Housing Flow Profile](image2)
The user interface flow profile picture shows the simulated flow through the user interface housing. From the fan specifications, it was possible to estimate a mass flow of 0.016 kg/s through the housing. It was assumed that the heat flux was 410 W/m² and the total heat input was 4 W. The estimates used were conservative, and should provide a worst-case scenario temperature, similar to that of the valve housing simulations.

5.3 Discussion

5.3.1 Strengths and Weaknesses

One strength of our system is its sturdiness. The user interface housing and valve housing are made out of aluminum, and will not break under any foreseeable operating or transportation conditions. This solidness of construction also dampens vibrations for the valves, in the case of the valve housing, and for the controllers in the case of the user interface housing. However, this solid construction does come with a weight penalty. The user interface housing is roughly 4 kg without any of the internal components installed. While this is not a problem during use, it does make transportation more difficult than it would be if a material such as plastic were used.

Another strength of our system is its relative simplicity. Since the components are all connected with wires, it is not difficult to disassemble the system if, say, a controller should need to be replaced. However, this means that there are many wires inside the user interface housing.

The code in the Arduino Mega microcontroller is easily modified, due to the use of a USB cable to power the controller. This is desirable, since the code was designed by mechanical engineers, rather than software engineers. There may be situations where the code that we have written is not adequate. The external USB port allows this to be remedied easily.

We discovered that the resolution of the Arduino was high enough that we could read all ten settings off one potentiometer. This allowed us to simplify our circuit and not include a second potentiometer and a toggle switch. We do now have two holes in our user interface housing that are not being utilized by working components, but this is an aesthetic rather than functional issue.

5.3.2 Next Steps

For future development, several steps should be taken before use with an actual human patient. Currently, the prototype is adequate for an animal study. The prototype has one dial with ten settings. There is potential for twenty settings if a second dial is added, and the toggle switch made functional.

The user interface housing and valve housing should both be redesigned with lighter materials. It is possible that a good solution would be to buy a factory-made box to put components in. This was not done for this project since we were able to get aluminum scrap free.
The Arduino code should be examined by a software engineer and possibly updated for better functionality and robustness. Currently the program checks to see if the valves are held open periodically, so there is sometimes a delay when changes are made.