Toronto Metropolitan University
Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Low Cost Pulsatile Flow Pump To Mimic Arterial Line Pressure Reading For Interventional Neuroradiology Training Simulation

by

Aiden Van Greuning

Biomedical Engineering Capstone Design Project

Toronto Metropolitan University, 2025

Acknowledgements

The author of this report would like to thank Kirill Pustovetov and Dr. Victor Yang for the guidance, support and supervision in the design process, creation and implementation of this capstone project. Additionally, I would like to thank the lab members at Northern Vascular Systems for advice given on this project, all the professors who have inadvertently contributed to this final project through many educational courses and all the peers who gave support and advice as well.

Certification of Authorship

The author certifies that this report is a result of months of research work, design, prototyping and implementation. The design of the system and preparation of this report are done completely by the author. Proper acknowledgement has been given to all sources and individuals whose contributions were utilized in the creation and presentation of this design process.

Table of Contents

Abstract

This report details the methodology, approach, synthesis, analysis and implementation of designing a low cost pulsatile pump to mimic arterial line (Art-line) pressure reading for interventional neuroradiology training simulation. This system is utilized to improve physician training by providing a simulated environment for biplane fluoroscopy through realistic pulsatile flow that mimics hemodynamics in normal vascular conditions. Replication of pathological flow in vascular lesions such as aneurysms and arteriovenous malformations were considered quite early in the design process but is unrealistic to simulate at a low cost. The final pump system was designed such that water-glycerol solution resembling blood would flow in a closed loop with a provided 3D printed model of the Circle of Willis and pressure sensors would be utilized to receive and generate cardiac cycle signals. To achieve these objectives, research was conducted on cerebral hemodynamics, both healthy and pathological in addition to pulsatile pump simulations even though they are primarily utilized for non-cerebrovascular flow. A cost analysis for components was also conducted in addition to advantages and disadvantages being noted. The greatest challenges were implementation of a pulsatile flow (peak systolic flow, end diastolic flow) and measuring pressure signals while keeping the cost low. Additionally, there were limited academic resources on this very specific topic (i.e. simulation of cerebrovascular flow/conditions via pump/pulsatile pump). In turn, the final design featured an Arduino Due activating a Kamoer stepper motor peristaltic pump using a TB6600 motor driver to create different types of cerebral flow with silicone tubing and a water-glycerol solution. To verify flow, a Digiten G1/4” Flow Sensor and Keyees Logic Level Shifter were implemented with a MB104 Breadboard. To measure pressure, a Honeywell MPRLS0025PA00001A pressure sensor was implemented into the flow path through a T junction connector and protective silicone tubing of decreasing diameters such that the pressure sensor would not be compromised by the flow. Typical fluid dynamics laws such as Poiseuille Law were used to calibrate the system so cerebrovascular flow could be properly replicated and processing speed from the microcontroller was kept in mind as well. While prototyping was costly, the promise of low cost has been achieved. The overall design process, prototyping phase and final implementation shall be discussed in this report.

Introduction & Background

In this chapter, there will be a short summary of interventional neuroradiology and the specific field in which this project is based on. This will be followed by background information on the pathological conditions that the project was to initially follow and the specific arterial behaviours the project is actually built to follow. The advantages of a low cost simulation system for interventional neuroradiology will also be discussed.

Interventional neuroradiology is a subfield of radiology with specific concentration on diagnosis and treatment of neurological conditions via minimally invasive image-guided surgical procedures. There can be further specialization as there are different types of neuroradiologists with the most well known type being a vascular interventional neuroradiologist. In that specific field there is a focus on diagnosing and treating cerebrovascular conditions such as aneurysms and arteriovenous malformations [1]. To work in such a prestigious field, substantial training is required. The most popular type of training for vascular interventional neuroradiologists are different types of simulations. These can range from virtual reality simulators, high tech catheters integrated in an angiography sweet to silicone vascular models that are combined with a circulation pump for fluid flow. Each of these simulation types have advantages and disadvantages, especially because of the difficulty of simulating the human body. However, all are still useful in some form when utilized for training purposes [2][3]. The initial objective for the capstone project was to design a pulsatile pump similar to other common vascular models that replicates normal cerebrovascular flow and pathological flow.

The three pathological conditions highlighted: aneurysms, arteriovenous malformations and dural arteriovenous fistula each have different flow behaviours and physical properties that were considered for implementation. Aneurysms are characterized by a pathological wall structure with internal elastic lamina and endothelial dysfunction caused by local flow dynamic dysregulation in an artery which leads to weakened focal pouches of the arterial wall. In turn an inflammatory process occurs with positive feedback regulation, leading to one of these pouches swelling into what is called a “bleb” [4][5][6]. Flow dynamics can be affected by many different components and there is a critical risk of vessel rupture, hence the importance of diagnosing aneurysms in time [4][6].

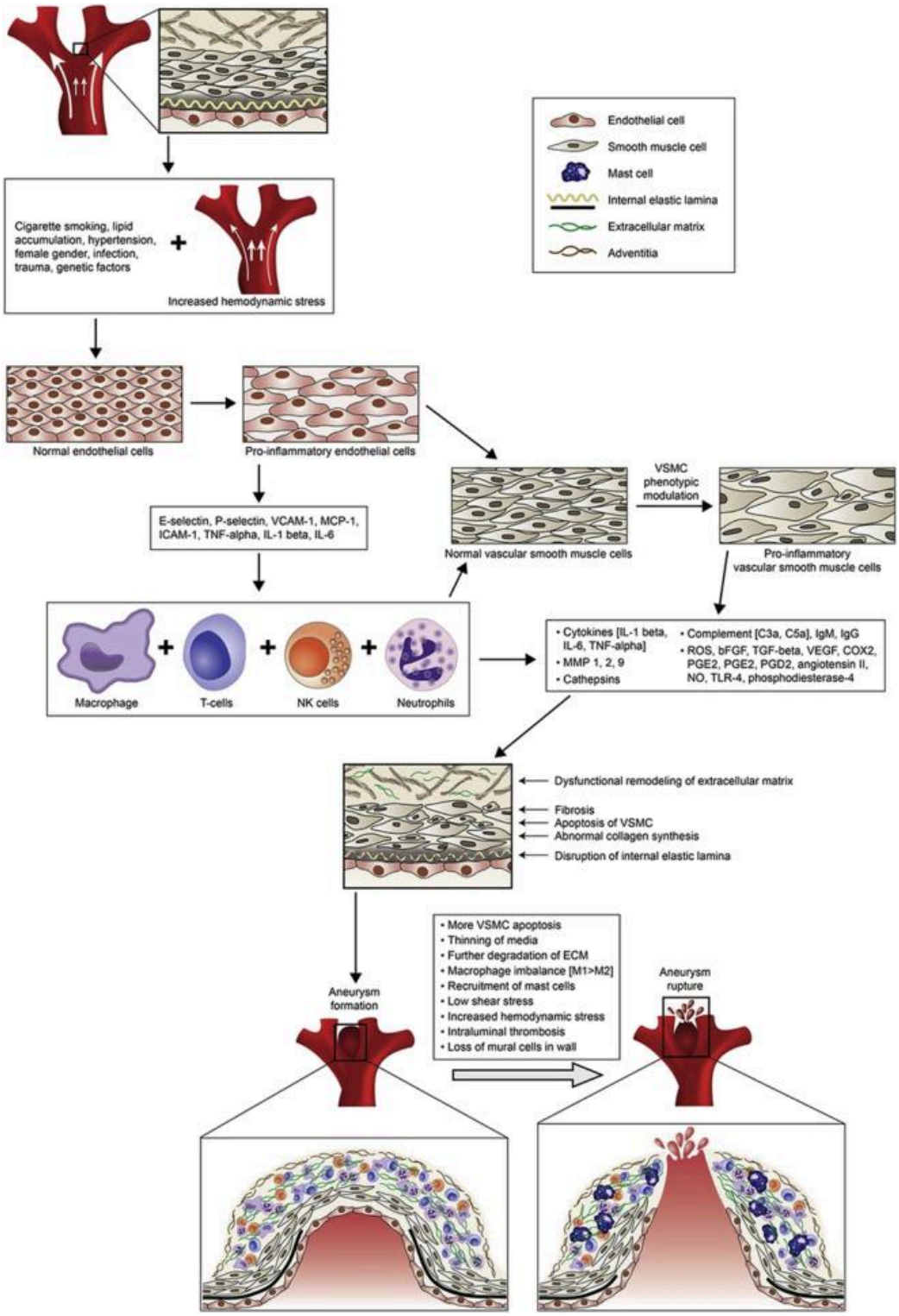

Figure 4.1. Biological process of aneurysm formation [4]

Arteriovenous malformations (AVMs) are an abnormal tangle of blood vessels in which there are no capillaries to connect arteries to veins, deliver oxygen to cells and regulate blood flow rate. Thus blood passes directly from arteries to veins, resulting in tissue that needs

oxygenated blood being damaged in addition to a dangerously high blood rate and blood pressure. In turn, arteries that have blood flowing into the AVM swell while the veins undergo thinning known as stenosis. Aneurysms can also be caused by AVMs through vein and arterial walls weakening in addition to there being the risk of AVMs growing with blood flow increasing, another cause for rupture [7] [8]. Finally, dural arteriovenous fistula is a type of arteriovenous malformation that typically occurs in the membrane that surrounds the brain and spinal cord due to membrane injury. In turn, a fistula is created between an artery and vein, essentially an abnormal connection in which blood is drained under high pressure between nearby veins or the spinal cord, causing swelling [9][10]. A common location of the brain where all three or at least two of the pathological conditions occur was initially investigated in order to design the pump at a low cost efficiently.

The specific location found was the middle cerebral artery (MCA), in which both aneurysms and arteriovenous malformations occur [11][12][13]. Phylogenetically the youngest of all cerebral vessels, the MCA is the formation of internal carotid arteries branching out to receive blood from the vertebral arteries [13]. It accounts for approximately 21% of total cerebral blood flow [14] [15]. On average its length is found to be 22.5 $\pm$ 8.1 mm with an average diameter at 3 mm [16][17]. The relationship between blood flow and pressure in the MCA based on a traumatic brain injury study is found to be inversely proportional; that being average pressure increased from 29 to 53 mmHg (3.866 to 7.066 kPa) while flow velocity decreased from 53 cm/s to 40 cm/s. Pulsatile flow velocity meanwhile increases from 77 cm/s to 98 cm/s [18]. When undergoing an aneurysm, the following velocities occur at each of the following locations of the MCA in the table below:

Table 4.1. Flow velocity at different locations of MCA while undergoing aneurysm [11][19].

Location at MCA while undergoing aneurysm	Flow velocity (cm/s)
M1 (Trunk)	25-75
Bleb (End of Trunk)	$\simeq 0$
M2 (Left bifurcation)	$\leq$ 25
M2 (Right bifurcation)	$\leq$100

For AVMs, the mean blood flow and mean cerebral blood flow of the MCA when undergoing this condition were 170 mL/min and 317 mL/min respectively [20]. For another simulation of MCA flow, the flow rate utilized for normal flow was between 250-260 mL/min [21]. Thus the middle cerebral artery was the initial artery that was to be replicated in the pump design with its flow conditions until pathological flow was proven too difficult to replicate. While basic flow rates and physical models of pathological conditions can be simulated, no model can directly influence the flow according to members of Northern Vascular Systems, so a flow distribution such as the one represented in Table 4.1 would be nearly impossible to create especially at a low cost.

In turn the priority shifted to normal cerebral flow and another artery took priority once the flow path for the 3D printed Circle of Willis was known: the Internal Carotid Artery (ICA). This artery in the Circle of Willis is one of the primary arteries that blood flows into from the rest of the body before splitting into many different pathways, the most important being directly the MCA, thus simplifying the potential flow path [22]. On average its diameter is slightly larger than the MCA due to the need for splitting pathways, calculated to be about 4.92 mm for simulation purposes [17] [23]. There are many ranges of flow rates for the ICA, for normal flow, pulsatile flow and simulated flows. It can be similar to the MCA normal flow ranging from 250-260 mL/min to having a peak systolic flow of 1041 mL/min and end diastolic flow of 394.2 mL/min to simulated standard pulsatile ranges of 1200-1800 mL/min or simulated normal flow ranges of 400-500 mL/min [21][15][24][25][26][27].

With a low cost pump such as the one being designed, it will allow for greater training of interventional neuroradiologists at a lower cost. The flow patterns will be made quite clear with the pressure readings and flow rate in addition to a provided 3D printed Circle of Willis model. In turn, cerebral flow is easier to interpret when imaging a specific part of the brain.

This chapter covered the potential need for a pulsatile pump and its advantages in training interventional neuroradiologists. In addition to that, interventional neuroradiology and the specific field this project is based on was discussed. Finally, the three pathological flow conditions originally intended to be replicated were discussed anatomically in addition to the arterial flow conditions that would actually be replicated.

Objectives

The objectives of this design project are many as a low cost simulation tool for interventional neuroradiology. All were analyzed to a varying degree and were the basis of the design choices made and goals that needed to be accomplished, listed in order of most to least importance.

A. Objective 1

The system must be low cost.

B. Objective 2

The pulsatile pump must replicate both normal and pulsatile cerebrovascular flow conditions as accurately as possible.

C. Objective 3

Potentially 20 repeated cardiac cycles must be performed when one cycle is the input.

D. Objective 4

The pressure sensor/transducer must accurately generate signals from the flow conditions.

In summary, the objectives from most to least importance were covered in this chapter, outlining the key aspects to creating a proper training simulation device for interventional neuroradiologists.

Theory and Design

This chapter demonstrates the relevant information from the first term and new information leading up to the design. This includes an updated literature review of pulsatile pump designs along with rejected designs and components. Finally, the final basic design shall be discussed in addition to challenges already faced and challenges likely to come.

A. Literature review

As previously established, an arterial pulsatile pump is typically utilized where there is no concern for invasiveness while prioritizing greater accuracy. Known as intra-arterial blood pressure (IABP), it is typically measured through a fluid column directly connecting the arterial system to a pressure transducer through hydraulic coupling. Thus a pressure waveform of an arterial pulse is transmitted through the column of fluid travelling to the pressure transducer. In turn, the pulse is converted into an electrical signal that is processed, amplified and converted into a visual display by a microprocessor. This signal can be further analyzed from observing beat to beat blood pressure and the waveform characteristic shape or utilize more complex systems to calculate other cardiovascular parameters [28]. The most important physical principles analyzed were fluid mechanics and dynamics, specifically cerebral hemodynamics with respect to the Circle of Willis [29][30][31][32][24][33][34][35][25][26]. Control systems principles were also considered as most pulsatile pumps, including cerebrovascular flow, were closed-loop systems [21][36][37]. As previously mentioned, due to a lack of industry standards to these specific types of simulations, a range for hemodynamics was set.

B. Design Approach

The key objectives did not change with the design, the approach to complete the objectives was what changed. The first and second objectives of a low cost and replicating normal cerebrovascular flow conditions were of the utmost importance while finding the pressure would be the final objective once the first and second objectives were completed.

C. Summary of Rejected Designs Before Prototyping

In the previous term, a total of three microcontroller-led designs were conceptualized and rejected for mostly the same reasons. The first design consisted of a pump system with an artificial diaphragm capable of “swelling” and returning to normal form to replicate normal and pathological flow demonstrated in different pathological cerebral conditions such as aneurysms and arteriovenous malformations [4][5][6][7][8][9][10] [38]. The second design consisted of a pump system with tubes replicating normal and pathological cerebral conditions. The third design consisted of a hybrid model of the first two designs in which there was one T-shaped 3D model of the middle cerebral artery (MCA) undergoing both normal and pathological flow as it

was undergoing a bifurcation aneurysm [21][11][12][13][14][15][16][17][18][19][20]. Below are block diagrams constructed of each of the rejected designs.

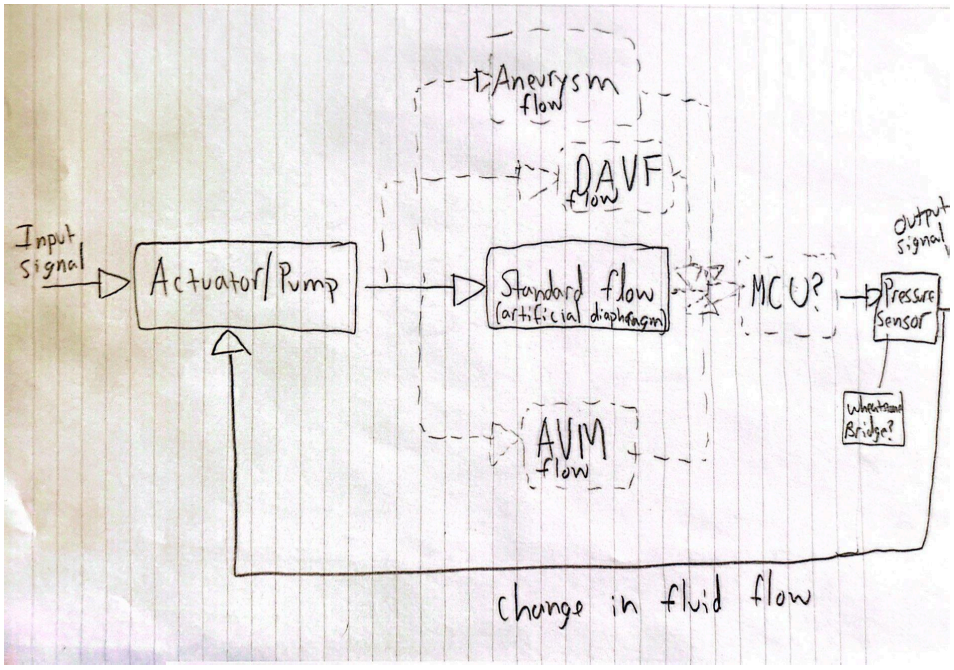

Figure 6.1. Block diagram of the flow pump considering change in flow and the "swelling" of the artificial diaphragm.

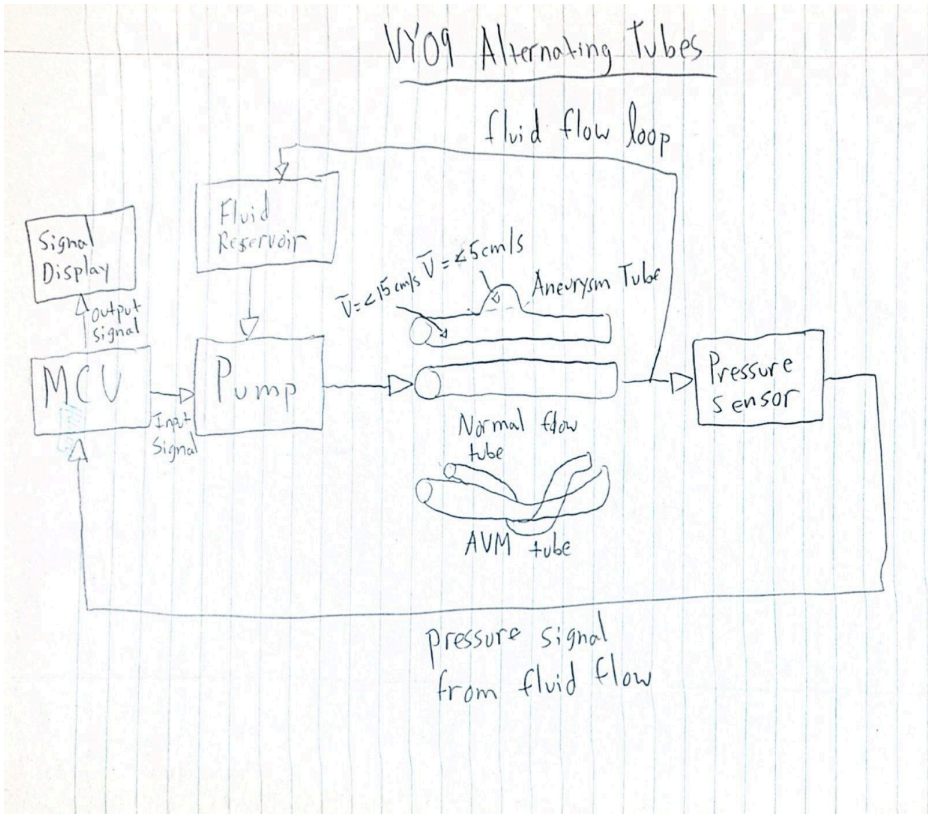

Figure 6.2. Block diagram of the flow pump considering alternate tubes that can be switched out to replicate each condition.

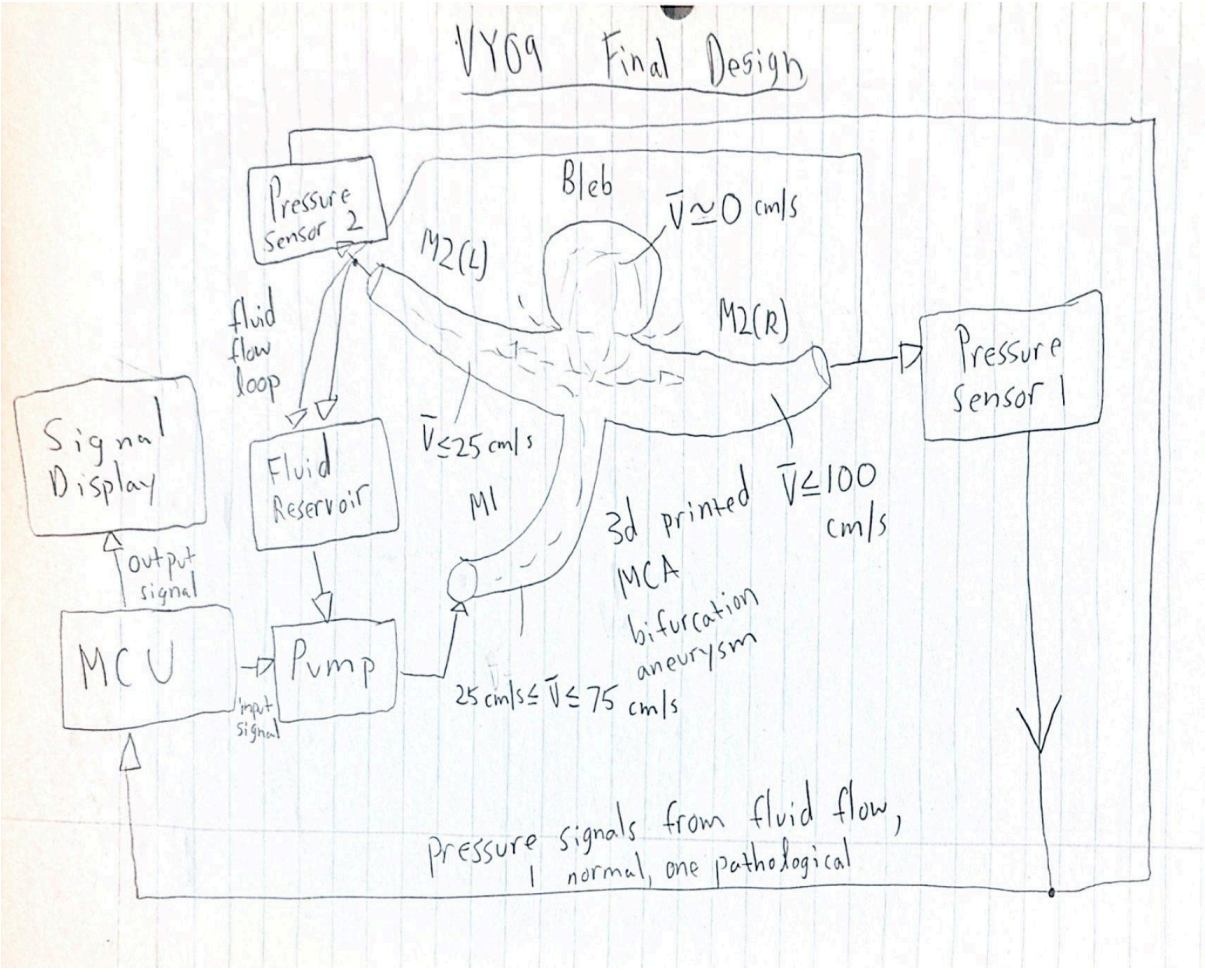

Figure 6.3. Block diagram of the final rejected design with a "T" shaped MCA undergoing bifurcation aneurysm.

All in all, each preceding design was rejected due to time constraints and other difficulties such as high cost, impracticality in terms of maintenance and designs far too complicated to complete that featured too many variables. The two greatest reasons however were a fundamental misunderstanding of cerebral hemodynamics in terms of simulation in addition to a misunderstanding of the role of the pump system by itself and in the grander scheme of the three interventional neuroradiology EDPs.

The pump system, while still a closed loop system, was to flow through a separately designed 3D printed model of the entire Circle of Willis that the microcatheter project would interact with as well. As for the former fundamental misunderstanding of cerebral hemodynamics; cerebral blood flow is more unique compared to other parts of the circulatory system as the brain undergoes a process known as autoregulation to keep flow stable [30]. When taking into account the three preliminary designs from the first term and the current design for the larger project in the second term, it was quite obvious that even if certain physical conditions were recreated, the blood flow characteristics would not be anatomically accurate. All 3D printed structures would be rigid bodies that cannot accurately replicate the biological processes the brain undergoes regularly. Thus the final general design was conceptualized for the second term, but not before one key component was rejected.

For each pump system conceptualized and designed, all were to be microcontroller led as the pump would be more precise overall. Arduino compatible microcontrollers were considered due to previous success found in other simulations [39][40]. Multiple arduino microcontrollers were considered with the final two candidates being the Arduino Uno R3 and the Arduino Due. The Uno R3 was chosen as first due to evidence of its functionality for pump designs, with compatible code started in preparation for it. However, upon further research, it was found the Uno was not as widely available as first thought and the Due was more highly capable of potential complex signal processing required for the pressure signals returned in the system, specifically with pulse-width modulation (PWM). Thus the Uno was rejected as well.

D. Current Final Design

With the grand design of the interventional neuroradiology groups kept in mind, the final microcontroller-led design was created. Only normal cerebrovascular flow was kept in mind while pulsatile flow was the next step with enough time and adjustments. Using code from the Arduino Due, a peristaltic pump was to be activated through an external power supply to supply the 3D printed Circle of Willis with a water-glycerol mix to act as blood. A pressure sensor would take measurements before entry into the Circle of Willis and potentially afterward. Below is the ideal combination of all three interventional neuroradiology EDPs in addition to the pump system design.

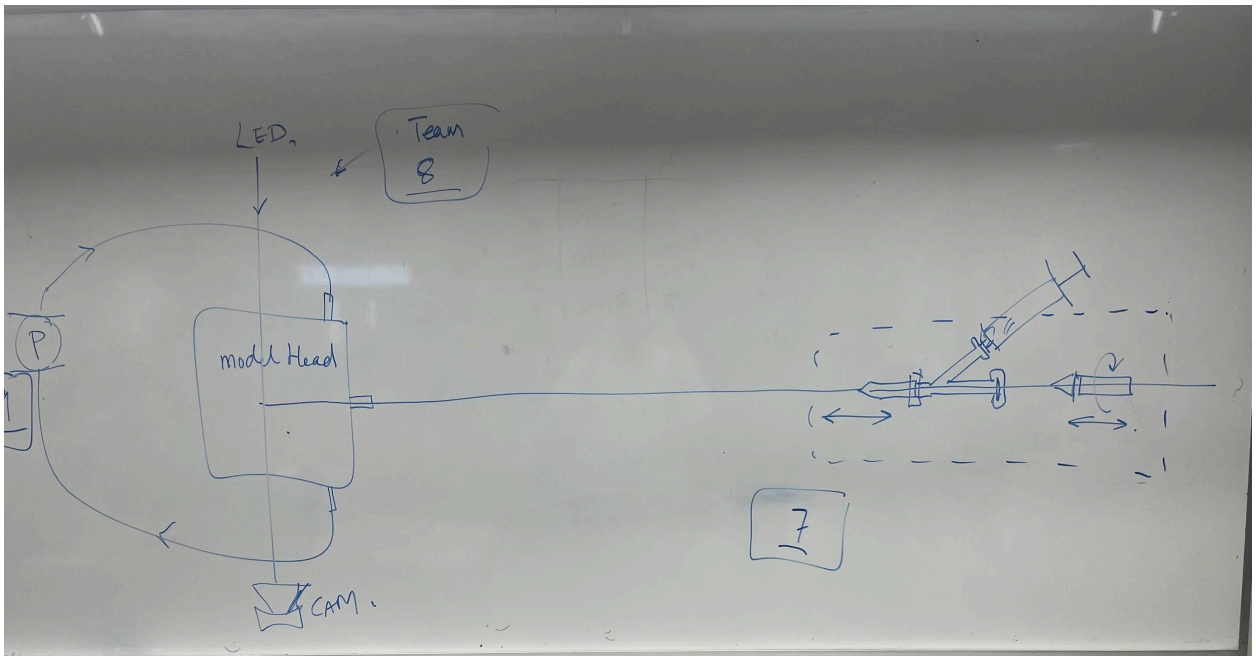

Figure 6.4. Diagram of the three combined projects and their roles.

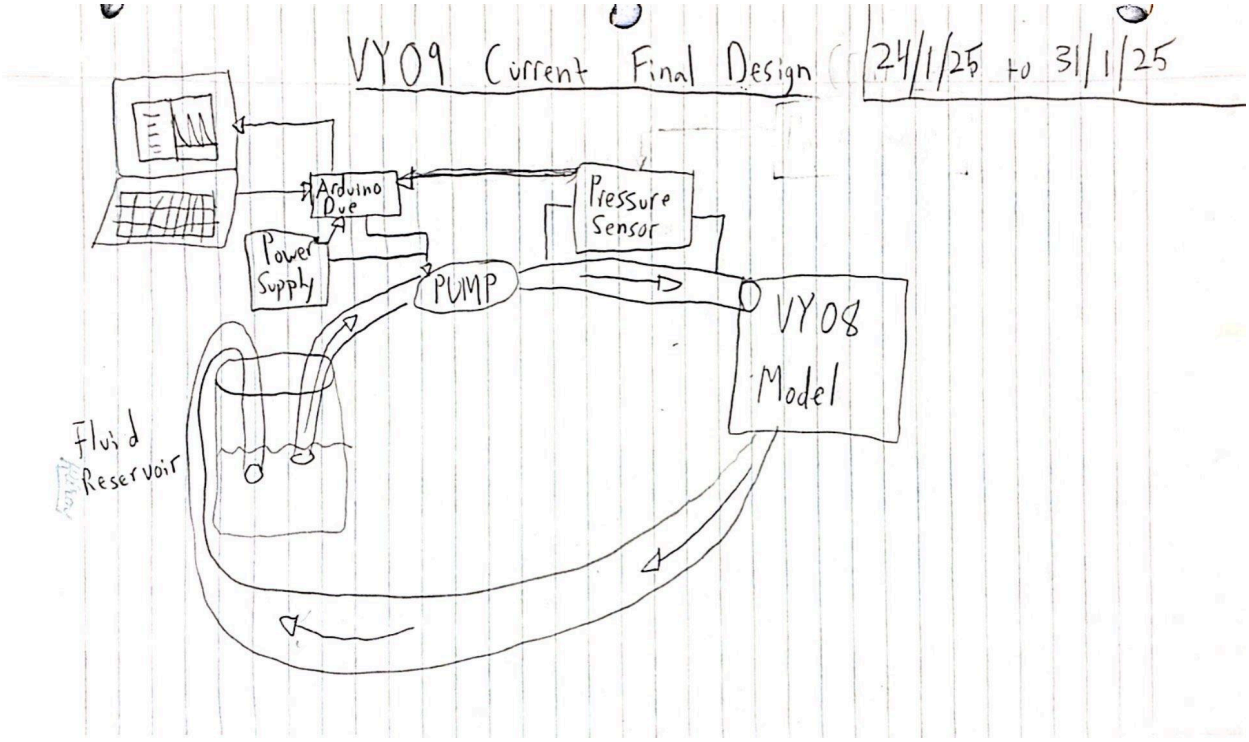

Figure 6.5. Block diagram of the current final design.

Previously a peristaltic pump was assumed to be too expensive but further research both through online and in-person sources proved they are essential for low-cost medical simulations [41][42][29]. As stated, the Arduino Due would trigger the peristaltic pump through an external power supply and a pressure sensor would in turn send back signals to be processed further. As the model being utilized is the entire Circle of Willis; the complete normal cerebral flow rate, flow rate distribution and relationships were found. In turn, the dimensions of the Circle of Willis model would be accounted for through the simple flow rate formula, potentially Navier Stokes and Poiseuille Law as shown below:

$$(1)\ \rho_{blood}\left(\frac{\partial v}{\partial t} + (v \cdot \nabla)v\right) = -\nabla p + \mu\nabla^2 v + F$$

where v is velocity field, p is pressure, μ is dynamic viscosity and F is the force of the body and

$$(2)\ Q = V/t = Ad/t = A\bar{v}$$

where Q is flow rate, V is volume, t is time, v is velocity, d is distance and A is cross-section.

$$(3)\ \Delta P = \frac{8\mu LQ}{Ar^4}$$

where P is pressure, u is viscosity, L is length, Q is flow rate, A is cross-section and r is radius

The absolute maximum flow rate has been found to be across several sources approximately 700 mL/min with some outliers of lower and higher flow rate in addition to distribution to arteries such as the MCA and ICA [29][30][31][32][24][33][34][35][25][26]. Depending on the inlet location on the 3D model, in this case the ICA, the flow rate may be

greatly reduced as without autoregulation and with knowledge of the structure being rigid; the flow rate would not distribute properly and would likely be governed by Poiseuille Flow Equations. For signal processing from the pressure signals, it has been found that cerebral blood flow and pressure have a relationship similar to a high pass filter. Essentially, the cerebrovascular system regulates blood flow by buffering slower blood pressure oscillations in the frequency range between 0.02 to 0.07 Hz while the faster oscillations greater than 0.2 Hz are transmitted with less buffering [43]. While this cannot be replicated physically, there are applications within the code to produce pressure signals similar to cerebral blood pressure ones.

E. Challenges

The greatest challenges were overall conceptualization as both a single project and a component of a much larger project due to a lack of sufficient academic resources as pump simulations are not common, especially for cerebrovascular training. In addition to that, the few articles that mention such mechanisms never describe the proper materials utilized. As previously stated, a range was set with a proper balance on in-vivo and in-vitro cerebral hemodynamics. A proper relatively low budget has been set but the greatest challenge after setting up the code with the pump will be the pressure sensor and relaying proper signals back to be processed.

In conclusion, this chapter discussed both the relevant information from the first term and new information leading up to the current final design. There was an updated literature review in addition to rejected designs and components. Finally, the final basic design was demonstrated along with challenges already faced and the challenges likely to appear in the testing process.

Alternative Designs

In this chapter, there shall be discussion of rejected physical components and the brief prototyping phase with justifications for component decisions. Proof of prototyping shall be demonstrated as well.

During the prototyping phase, not too many components were removed or changed from the design as it was kept basic for both simplicity of implementation and low budget. As stated, the peristaltic pump was chosen due to its ability to reach cerebrovascular flow rates and the simplicity in controlling it [42][29]. The MEGA2560 was the first microcontroller considered due to current possession of it but according to required parameters, it would be too slow [40]. The Arduino Uno was next considered as it is one of the more common Arduino boards and used for a range of projects. Code for it was started with no purchase made but upon further research the Arduino Due was proven more capable of the pressure signal processing capabilities desired. Thus the Due was purchased instead. Below is a portion of the Uno code.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#if defined(__AVR_ATmega328P__) // Arduino Uno
  #define BUTTON_A 2
  #define BUTTON_B 3
  #define BUTTON_C 4
  #define PUMP_PIN 9
```

Figure 7.1. Portion of Arduino Uno code prepared before switching to Arduino Due.

Once the microcontroller was decided, the rest of the component choices were centred around achieving flow through the peristaltic pump and finding pressure signals through a sensor. A greater idea of the combined EDP projects was found and considered as well, specifically in regards to fluid flow. [44] [24][30][43]. The main alternative design that was considered was utilizing a Kamoer 400 mL/min stepper motor peristaltic pump as it was capable of reaching the normal flows of the ICA and MCA. It was primarily utilized for proof of concept with the Due and other components such as the TB6600 motor driver. Below is the first prototype of the pump system.

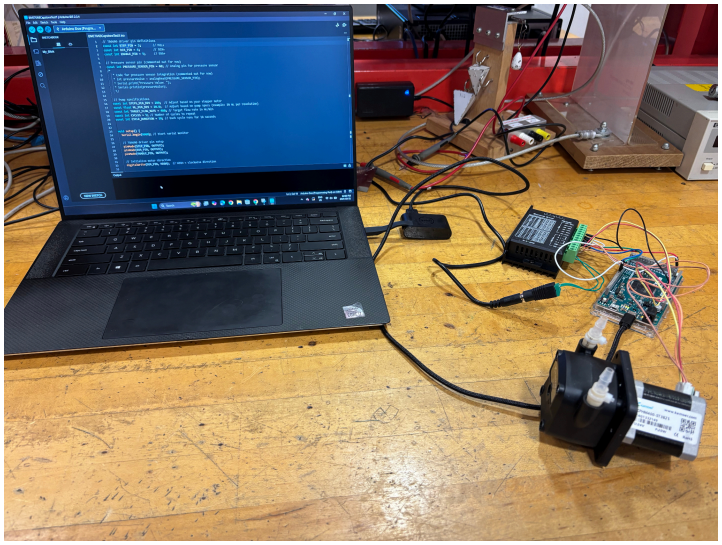

Figure 7.2. First pump system prototype with Arduino Due, TB6600 Motor Driver, 24 V power supply and Kamoer 400 mL/min stepper motor peristaltic pump.

Afterwards, there were no alternative designs. There was only the final design that had a Kamoer pump with a much higher flow rate range due to the intention of reaching peak systolic flow in the arteries and the scaling up of the Circle of Willis model. Other components such as the flow sensor and pressure sensor were only considered for the final pump and were intended to be adjusted according to testing.

In summary, the rejected physical components and the brief prototyping phase were discussed. Justifications for component decisions/rejections were provided. Proof of prototyping was demonstrated as well.

Material/Component list

Table 8.1. List of all components acquired for testing including the alternative pump.

Item	Units bought	Units Utilized	Cost (incl. tax)
Peristaltic Pump 24V Stepper Motor high Flow 400ml/min Kamoer KPHM400 lab Liquid dosing Pump	1.00	1.00	55.37
24V 2A Power Supply	1	1	19.2
5.5 * 2.1 mm DC screw terminal	10	1	11.85
TB6600 Stepper motor driver	1	1	20.33
Kamoer KK2000 high Flow peristaltic Pump 24V high Precision Stepper Motor Quick Tube Change Liquid dosing Pump RS485 Speed Control 700-2000 ml/min	1	1	173
4.8mm ID x 8mm OD 10ft Flexible Silicone Rubber Tube, Clear Silicone Tubing	1	1	28.35
1/4"(6.4mm) ID 3/8"(9.6mm) OD 10ft, Flexible Silicone Tube	1	1	33.55
Feelers 5/8" ID x 3/4" OD Silicone tube	1	1	21.46
Keeyee 4 channel I2C logic level converter 3.3 V to 5 V	10	1	14.45
Patikil 1/4" to 5/8" brass hose reducer	1	1	22.14
DIGITEN G1/4" Water Flow Sensor	1	1	15.81
2 mm ID x 4 mm OD 1.5 m silicone rubber tube	1	1	19.98
Adafruit TCA9548A I2C multiplexer	1	1	24.8
Honeywell MPRLS0025PA00001A pressure sensor (x2)	2.00	1.00	66.17
1/4" T junction	2	1	7.89
Glycerin	3	3	39.85
24V 4A 100W AC/DC Power Adapter	1	1	29.37
Total			$603.57

Measurement and Testing Procedures

In this chapter, measurement and testing procedures shall be discussed, including specific tests performed and additional tests that needed to be performed due to specific results. The potential of mass production shall also be analyzed. Performance results and analysis shall be discussed in other chapters.

To run the pulsatile pump design, simply utilize Arduino code to have the Arduino Due activate the pump through proper connections to the TB6600 and 24 V power supply. Stripping of wires will be required for the pump connections to the TB6600. All Arduino connections from the TB6600 must be either to digital pins or ground pins. The silicone tubing compatible with the Kamoer KK2000 is the ¼" inner diameter in which it can simply be rested on the rollers and secured. Ensure there is space for the reservoir of water or water-glycerol to be withdrawn and returned in a closed loop. To add the flow sensor, its wires must be stripped as well. Due to the sensor outputting flow rate values at 5 V, the logic level shifter must be utilized as well to shift down from 5 V to 3.3 V logic. Cut some of the silicone tubing to implement the flow sensor and utilize the MB104 breadboard so multiple components can use the same pins on the Arduino board. Use female to male wires to facilitate connections. Finally, to add the Honeywell pressure sensor, simply cut the tubing to partially cover the perpendicular end of the T junction and to connect the T junction to the rest of the system. Then cut the 4.3 mm inner diameter and 2 mm inner diameter tubings to appropriate sizes and stack them inside the original tubing on the perpendicular end such that the pressure sensor nozzle can sit in the 2mm tubing without being compromised by liquid flow. Utilize appropriate stripped wires to solder to the pins on the pressure sensor and use female to male wires to facilitate connection to the Arduino SDA and SCL pins in addition to the 3.3 V and ground pins through the breadboard.

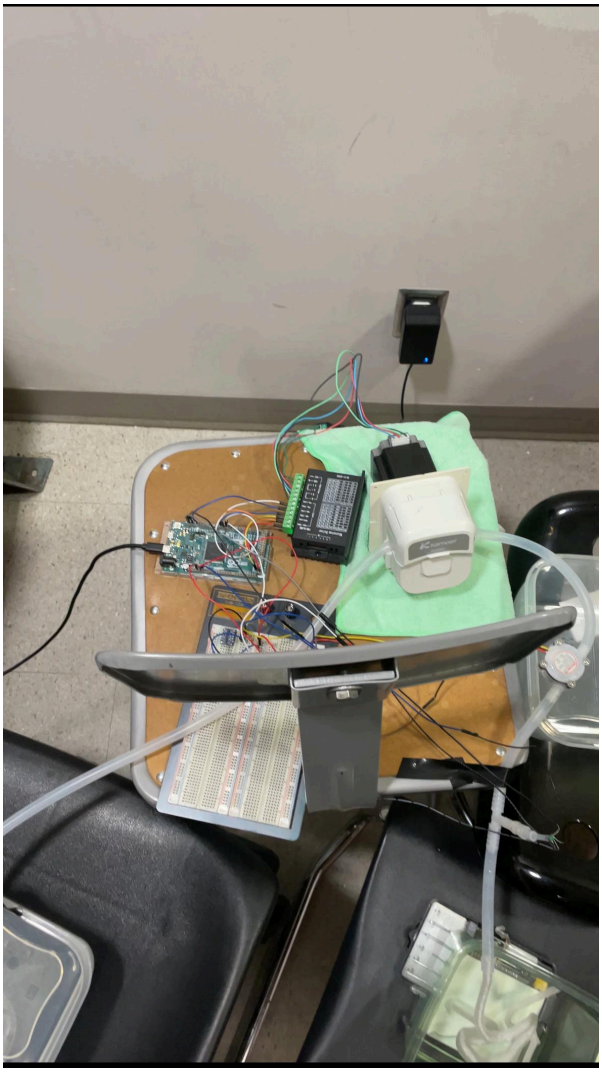

Figure 9.1. Final design implementation with the pump, flow sensor, pressure sensor and arterial model.

```
// --- Motor Control ---
const int STEP_PIN = 3;
const int DIR_PIN = 4;
const int ENABLE_PIN = 5;
const int STEPS_PER_REV = 1600;          // Updated: microstepping resolution
const float ML_PER_REV = 2.12;           // Calibrated volume per rev

// --- Flow Sensor ---
const int FLOW_SENSOR_PIN = 2;
volatile unsigned long flowPulseCount = 0;
float flowRate = 0;

// --- Flow Constants ---
enum Artery { ICA, MCA };
Artery arteryMode = ICA;

float SCALE_FACTOR = 1.5; // Use 1.2–1.5 for ICA, 3.333 for MCA
unsigned int cardiacCycles = 20; // Number of cardiac cycles to simulate

// Example average ICA flow pattern (mL/min)
float ICA_PEAK = 802;
float ICA_DIASTOLIC = 267;
```

Figure 9.2. Snippet of pulsatile flow code utilized for testing.

First simple activation of the pump was tested, followed by liquid flow both with water and water-glycerol mix. The flow sensor and logic level shifter were next introduced to ensure the pump is running at the desired flow rate. Afterwards the pressure sensor was added to generate pressure signals. The flow sensor was also tested on its own using a disposable code to compare flow rate in mL/min to actual volume deposited in one minute. Further results-based testing was conducted to ensure key objectives were met and shall be discussed in the analysis section.

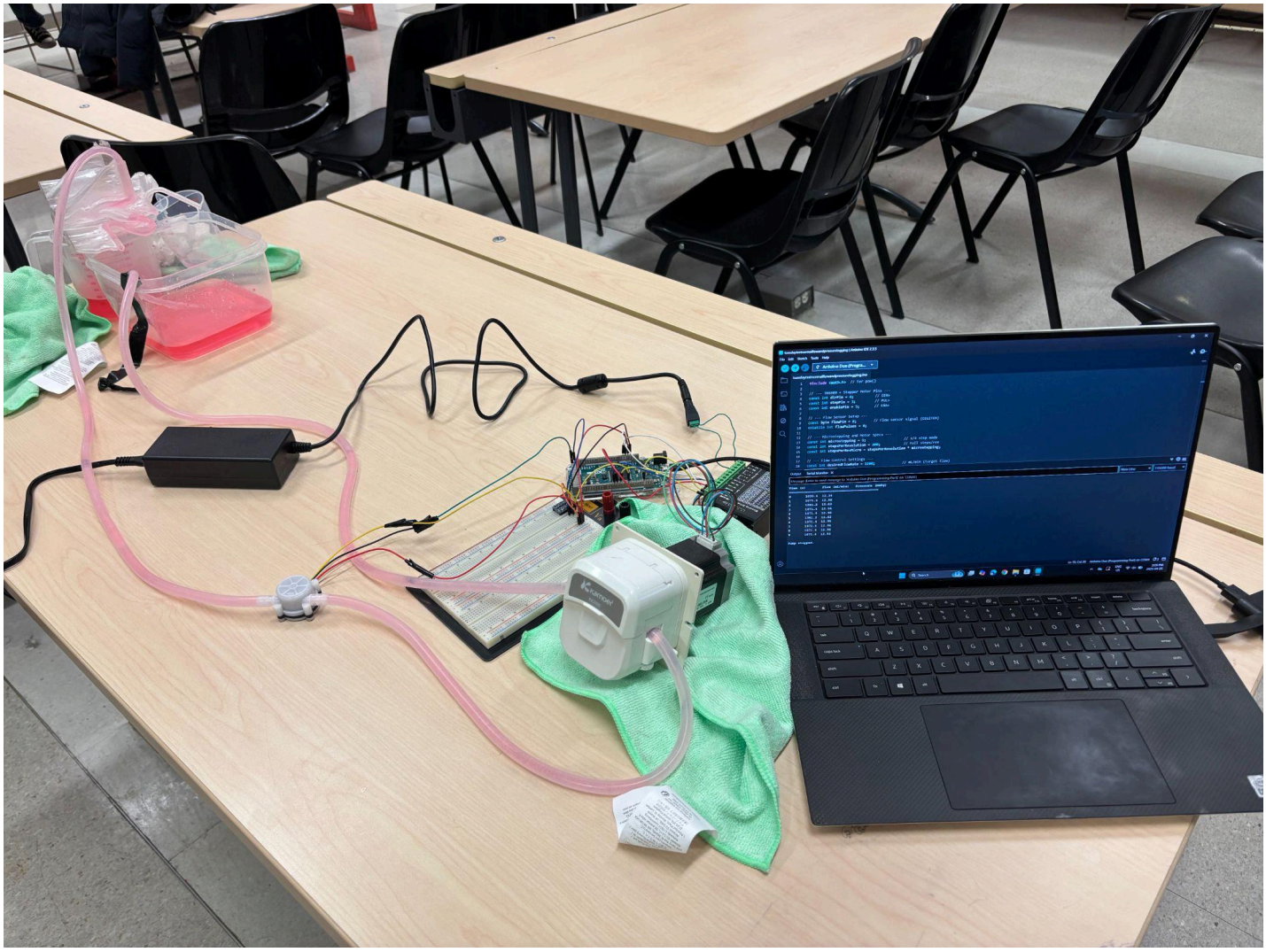

Figure 9.3. Final design implementation for Engineering Day with the pump, flow sensor and arterial model.

To potentially mass produce this system, components would have to be separated such that the system can be manually built. If left together, some wires can degrade physically especially with the weight of the peristaltic pump being utilized and the fragility of the pressure sensor with its soldered wires. The silicone tubes could also be precut to establish the different connections.

In conclusion, measurement and testing procedures were discussed, including specific tests performed and additional tests that were needed to be performed due to specific results. The potential of mass production was also analyzed.

Performance Measurement Results

This chapter shall briefly cover results obtained from testing, with analysis conducted in another chapter.

The pump can activate simply enough with the provided code and conduct fluid flow. However, once attempting to replicate ICA pulsatile flow for 20 cardiac cycles with the flow sensor taking the flow rate, the projected flow rate was never the same as the actual flow rate. First there was underperformance by 200 mL/min, then that underperformance increased to 300-700 mL/min with repeated tests and attempts to change microstepping on the motor driver. Attempted tests with the pressure sensor did not yield desired results as even basic code to detect I2C devices failed to detect the honeywell sensor, let alone the initial code to detect flow and pressure. Further adjustments were needed as the pressure sensor at one point interfered with pump activation entirely. Below are screenshots of the unsatisfactory pulsatile flow tests.

```
Setup complete. Running ICA flow simulation...
Cycle 1
Target Flow (end): 951.5 mL/min | Actual Flow: 360.1 mL/min
Cycle 2
Target Flow (end): 951.5 mL/min | Actual Flow: 491.8 mL/min
Cycle 3
Target Flow (end): 951.5 mL/min | Actual Flow: 728.9 mL/min
Cycle 4
Target Flow (end): 951.5 mL/min | Actual Flow: 614.7 mL/min
Cycle 5
Target Flow (end): 951.5 mL/min | Actual Flow: 728.9 mL/min
Cycle 6
Target Flow (end): 951.5 mL/min | Actual Flow: 737.6 mL/min
Cycle 7
Target Flow (end): 951.5 mL/min | Actual Flow: 728.9 mL/min
Cycle 8
Target Flow (end): 951.5 mL/min | Actual Flow: 737.6 mL/min
Cycle 9
Target Flow (end): 951.5 mL/min | Actual Flow: 737.6 mL/min
Cycle 10
Target Flow (end): 951.5 mL/min | Actual Flow: 728.9 mL/min
Simulation complete.
```

Figure 10.1. Measured flow rate underperformance by ~200 mL/min when attempting to replicate ICA pulsatile flow

```
Setup complete. Running ICA flow simulation...
Cycle 1
  Target: 951.5 mL/min | Actual: 10.2 mL/min
Cycle 2
Setup complete. Running ICA flow simulation...
Cycle 1
  Target: 951.5 mL/min | Actual: 112.0 mL/min
Cycle 2
Cycle 3
  Target: 934.0 mL/min | Actual: 408.2 mL/min
Cycle 4
Cycle 5
Cycle 6
  Target: 1066.5 mL/min | Actual: 571.4 mL/min
Cycle 7
Cycle 8
  Target: 876.3 mL/min | Actual: 612.2 mL/min
Cycle 9
Cycle 10
Cycle 11
  Target: 1120.7 mL/min | Actual: 612.2 mL/min
Cycle 12
Cycle 13
  Target: 920.8 mL/min | Actual: 612.2 mL/min
Cycle 14
Cycle 15
Cycle 16
  Target: 953.2 mL/min | Actual: 612.2 mL/min
Cycle 17
Cycle 18
Cycle 19
  Target: 958.9 mL/min | Actual: 612.2 mL/min
Cycle 20
Simulation complete.
```

Figure 10.2. Measured flow rate underperformance by ~300 mL/min when attempting to replicate ICA pulsatile flow plus clear evidence of glitching in code output or speed underperformance with missing cycles.

```
  Target Flow (end): 1203.0 mL/min | Actual n | Actual Flow: 529.6 mL/min
Cycle 4
): 1203.0 mL/min | Actual Flow: 459.0 mL/min
Cycle 3
  Target Flow (end): 1203.0 mL/min | Actual Flow: 529.6 mL/min
Cycle 4
Setup complete. Running ICA flow simulation...
Cycle 1
  Target Flow (end): 1203.0 mL/min | Actual Flow: 496.0 mL/min
Cycle 2
  Target Flow (end): 1203.0 mL/min | Actual Flow: 459.0 mL/min
Cycle 3
  Target Flow (end): 1203.0 mL/min | Actual Flow: 564.9 mL/min
Cycle 4
  Target Flow (end): 1203.0 mL/min | Actual Flow: 459.0 mL/min
Cycle 5
  Target Flow (end): 1203.0 mL/min | Actual Flow: 564.9 mL/min
Cycle 6
  Target Flow (end): 1203.0 mL/min | Actual Flow: 459.0 mL/min
Cycle 7
  Target Flow (end): 1203.0 mL/min | Actual Flow: 529.6 mL/min
Cycle 8
  Target Flow (end): 1203.0 mL/min | Actual Flow: 494.3 mL/min
Cycle 9
  Target Flow (end): 1203.0 mL/min | Actual Flow: 564.9 mL/min
Cycle 10
  Target Flow (end): 1203.0 mL/min | Actual Flow: 459.0 mL/min
Cycle 11
  Target Flow (end): 1203.0 mL/min | Actual Flow: 564.9 mL/min
Cycle 12
  Target Flow (end): 1203.0 mL/min | Actual Flow: 494.3 mL/min
Cycle 13
  Target Flow (end): 1203.0 mL/min | Actual Flow: 529.6 mL/min
Cycle 14
  Target Flow (end): 1203.0 mL/min | Actual Flow: 494.3 mL/min
Cycle 15
  Target Flow (end): 1203.0 mL/min | Actual Flow: 529.6 mL/min
```

Figure 10.3. Measured flow rate underperformance by ~600-700 mL/min when attempting to replicate ICA pulsatile flow

In summary, basic results were achieved with the current design. However, as more complicated factors and components were added, deviations occurred with pulsatile flow not reaching the minimum capabilities of the pump after several tests and the pressure sensor not being detected at all. Analysis of these results shall be conducted in the next chapter.

Analysis of Performance

This chapter shall analyze the results obtained from testing of the final design and potential improvements. These can range from simple changes to fundamental flaws of the design.

The Honeywell sensor yielded the most unsatisfactory results as the Arduino Due could not even detect it. This could be due to unstable wiring and soldering as the sensor is quite tiny. As previously stated, code adjustment was required while testing the sensor as flow activation was interfered with. This could've likely been due to multiple wires and devices needing to share specific ports in the Arduino Due through the breadboard among other reasons. Further work could be done to improve the sensor connection but it was eventually abandoned entirely due to time constraints.

The disadvantages of silicone tubing were found quickly as the tubes get filled with air bubbles, distorting the flow rates and preventing exact startups each time a simulation is run. Furthermore, the peristaltic pump does not seem to reach required flow rates especially for pulsatile flow, meaning the stepper motor may not be capable of the specific flow replication or further adjustments are needed with the TB6600 and potentially even the power supply. The flow sensor is proven to deliver flow measurements properly especially after a volume test. However, it is dependent on the performance of the pump which is quite inconsistent. This was especially proven true when there was not much difference in behaviour found between water and the water-glycerol solution. All in all, it is likely simple adjustments for the problems with the pump within the code or the TB6600 but the pressure sensor integration may be fundamentally flawed.

Eventually further testing was conducted on ensuring normal flow for the pump and Posieuille Law was implemented in the Arduino Code to simply convert flow rate in mL/min to pressure in mmHg. While there is a clear loss of accuracy and no pulsatile flow, there at least is evidence of what the typical pressure should be based on flow rate and normal conditions of the arteries. The power source was upgraded as well to allow for more current flow, from 2 A to 4 A. Pulsatile flow was found to be too difficult to accurately replicate with the low cost pump as the typical cardiac waveform with the peak systolic flow and end diastolic flow both occurring in seconds which only more advanced pumps are capable of.

In conclusion, the results were not as expected but the likely reasons for these unsatisfactory results are few, ranging from simple changes to fundamental flaws, especially regarding the pressure sensor. Appropriate adjustments were made to meet the objectives.

Conclusions

In conclusion, the methodology, approach, synthesis, analysis and implementation of designing a low cost pulsatile pump to mimic arterial line (Art-line) pressure reading for interventional neuroradiology training simulation was reported. Replication of pathological flow in vascular lesions such as aneurysms and arteriovenous malformations were considered quite early in the design process but was found to be unrealistic to simulate at a low cost. The final pump system was designed such that water-glycerol solution resembling blood would flow in a closed loop with a provided 3D printed model of the Circle of Willis and pressure sensors would be utilized to receive and generate cardiac cycle signals. To achieve these objectives, research was conducted on cerebral hemodynamics, both healthy and pathological in addition to pulsatile pump simulations even though they are primarily utilized for non-cerebrovascular flow. A cost analysis for components was also conducted in addition to advantages and disadvantages being noted. The budget ballooned to over 500 dollars due to the uniqueness of the materials but it was still lower than the typical lab tested pump systems such as the one demonstrated at Northern Vascular Systems.The greatest challenges were implementation of a pulsatile flow and measuring pressure signals while keeping the cost low. The pressure sensor was not detected in the system and may need calibration or could be fundamentally flawed. The pump may not be capable of pulsatile flow but can achieve normal flow if correct preparations are made. This was proven later with further testing for normal flow and Poiseuille Law calculations can compensate for the lack of a pressure sensor. Additionally, there were limited academic resources on this very specific topic (i.e. simulation of cerebrovascular flow/conditions via pump/pulsatile pump). Access to a printed 3D model was only given in the last two weeks as there were difficulties printing it among other issues, thus delaying final design implementation. Regardless, the final design featured an Arduino Due activating a Kamoer stepper motor peristaltic pump using a TB6600 motor driver to create different types of cerebral flow with silicone tubing and a water-glycerol solution. To verify flow, a Digiten G1/4” Flow Sensor and Keyees Logic Level Shifter were implemented with a MB104 Breadboard. To measure pressure, a Honeywell MPRLS0025PA00001A pressure sensor was implemented into the flow path through a T junction connector and protective silicone tubing of decreasing diameters such that the pressure sensor would not be compromised by the flow. While prototyping was costly, the promise of low cost has been achieved. However, there is a clear loss in accuracy both when choosing materials and attempting to measure pressure signals.

References

[1] Mya Care, "INTERVENTIONAL NEURORADIOLOGY: APPLICATIONS, RISKS, LATEST ADVANCEMENTS, AND MORE | MyA Care," *Mya Care*. https://myacare.com/blog/interventional-neuroradiology-applications-risks-latest-advancements-and-more?form=MG0AV3

[2] K. Kreiser, N. Sollmann, and M. Renz, "Importance and potential of simulation training in interventional radiology," RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren, vol. 195, no. 10, pp. 883–889, May 2023, doi: https://doi.org/10.1055/a-2066-8009.

[3] M. Ernst *et al.*, "Quantitative Evaluation of Performance in Interventional Neuroradiology: An Integrated Curriculum Featuring Theoretical and Practical Challenges," *PLOS ONE*, vol. 11, no. 2, p. e0148694, Feb. 2016, doi: https://doi.org/10.1371/journal.pone.0148694.

[4] Texakalidis, P., Sweid, A., Mouchtouris, N., Peterson, E. C., Sioka, C., Rangel-Castilla, L., Reavey-Cantwell, J., & Jabbour, P. (2019). Aneurysm Formation, Growth, and Rupture: The Biology and Physics of Cerebral Aneurysms. *World Neurosurgery*, *130*, 277–284. https://doi.org/10.1016/j.wneu.2019.07.093

[5] Isoda, H., Hirano, M., Takeda, H., Kosugi, T., Alley, M. T., Markl, M., Pelc, N. J., & Sakahara, H. (2006). Visualization of Hemodynamics in a Silicon Aneurysm Model Using Time-Resolved, 3D, Phase-Contrast MRI. *American Journal of Neuroradiology*, *27*(5), 1119–1122.

[6] "Brain Aneurysm | neuroangio.org," *Neuroangio.org*, 2022. https://neuroangio.org/patient-information/patient-information-brain-aneurysm/.

[7] U.S. Department of Health and Human Services. (n.d.). *Arteriovenous malformations (avms)*. National Institute of Neurological Disorders and Stroke. https://www.ninds.nih.gov/health-information/disorders/arteriovenous-malformations-avms#:~:text=What%20are%20arteriovenous%20malformations%3F,develop%20elsewhere%20in%20the%20body.

[8] "Brain Arteriovenous Malformation | neuroangio.org," *Neuroangio.org*, 2022. https://neuroangio.org/patient-information/patient-information-brain-arteriovenous-malformation/

[9] "Brain Dural Fistula | neuroangio.org," *Neuroangio.org*, 2022. https://neuroangio.org/patient-information/patient-information-brain-dural-fistula/

[10] Gailloud, P., Muster, M., Piotin, M., Mottu, F., Murphy, K. J., Fasel, J. H. D., & Rufenacht, D. A. (1999). In Vitro Models of Intracranial Arteriovenous Fistulas for the Evaluation of New Endovascular Treatment Materials. *American Journal of Neuroradiology : AJNR*, *20*(2), 291–295.

[11] P. M. Munarriz, P. A. Gómez, I. Paredes, A. M. Castaño-Leon, S. Cepeda, and A. Lagares, "Basic Principles of Hemodynamics and Cerebral Aneurysms," *World Neurosurgery*, vol. 88, pp. 311–319, Apr. 2016, doi: https://doi.org/10.1016/j.wneu.2016.01.031.

[12] H. Su and J. Yu, "Brain arteriovenous malformations of the middle cerebral artery region: image characteristics and endovascular treatment based on a new classification system," *BMC Neurology*, vol. 23, no. 1, Jan. 2023, doi: https://doi.org/10.1186/s12883-023-03084-y.

[13] D. Purves *et al.*, "The Blood Supply of the Brain and Spinal Cord," *Nih.gov*, 2015. https://www.ncbi.nlm.nih.gov/books/NBK11042/

[14] "Middle Cerebral Artery | neuroangio.org." https://neuroangio.org/anatomy-and-variants/middle-cerebral-artery/

[15] L. Zarrinkoob, K. Ambarki, A. Wåhlin, R. Birgander, A. Eklund, and J. Malm, "Blood flow distribution in cerebral arteries," *Journal of Cerebral Blood Flow & Metabolism*, vol. 35, no. 4, pp. 648–654, Apr. 2015, doi: https://doi.org/10.1038/jcbfm.2014.241.

[16] S. Gunnal, M. Farooqui, and R. Wabale, "Study of middle cerebral artery in human cadaveric brain," *Annals of Indian Academy of Neurology*, vol. 0, no. 0, p. 0, 2019, doi: https://doi.org/10.4103/0972-2327.144289.

[17] A. T. Rai, J. P. Hogg, B. Cline, and G. Hobbs, "Cerebrovascular geometry in the anterior circulation: an analysis of diameter, length and the vessel taper," *Journal of NeuroInterventional Surgery*, vol. 5, no. 4, pp. 371–375, Apr. 2012, doi: https://doi.org/10.1136/neurintsurg-2012-010314.

[18] M. O. Kim *et al.*, "Principles of cerebral hemodynamics when intracranial pressure is raised," *Journal of Hypertension*, vol. 33, no. 6, pp. 1233–1241, Jun. 2015, doi: https://doi.org/10.1097/hjh.0000000000000539.

[19] S. Wang *et al.*, "Middle cerebral artery bifurcation aneurysms are associated with patient age, sex, bifurcation angle, and vascular diameters," *Scientific Reports*, vol. 13, no. 1, Dec. 2023, doi: https://doi.org/10.1038/s41598-023-50380-1.

[20] Y. Takeda *et al.*, "Hemodynamic Analysis of Cerebral AVMs with 3D Phase-Contrast MR Imaging," *American Journal of Neuroradiology*, vol. 42, no. 12, pp. 2138–2145, Oct. 2021, doi: https://doi.org/10.3174/ajnr.a7314.

[21] Falk, K. L., Harvey, E. C., Schafer, S., Speidel, M. A., & Strother, C. M. (2019). Optimizing the Quality of 4D-DSA Temporal Information. *American Journal of Neuroradiology*, *40*(12), 2124–2129. https://doi.org/10.3174/ajnr.A6290

[22] "Anatomy and Variants | neuroangio.org," *Neuroangio.org*, 2022. https://neuroangio.org/anatomy-and-variants/

[23] J. Krejza *et al.*, "Carotid Artery Diameter in Men and Women and the Relation to Body and Neck Size," *Stroke*, vol. 37, no. 4, pp. 1103–1105, Apr. 2006, doi: https://doi.org/10.1161/01.str.0000206440.48756.f7.

[24] Yanina Zócalo and D. Bia, "Sex- and Age-Related Physiological Profiles for Brachial, Vertebral, Carotid, and Femoral Arteries Blood Flow Velocity Parameters During Growth and Aging (4–76 Years): Comparison With Clinical Cut-Off Levels," *Frontiers in Physiology*, vol. 12, Aug. 2021, doi: https://doi.org/10.3389/fphys.2021.729309.

[25] S. E. Razavi and R. Sahebjam, "Numerical Simulation of the blood flow behavior in the circle of Willis.," *PubMed*, Jan. 2014, doi: https://doi.org/10.5681/bi.2014.008.

[26]"Blood Flow Through the Circle of Willis — Simulation Example - ANSYS Innovation Courses," *ANSYS Innovation Courses - Free, online, physics and engineering courses*, Aug. 24, 2020. https://innovationspace.ansys.com/courses/courses/viscous-laminar-flows/lessons/homework-quizzes-simulation-examples-lesson-3/topic/blood-flow-through-the-circle-of-willis-simulation-example/

[27] Y. Nagai, M. K. Kemper, C. J. Earley, and E. Jeffrey. Metter, "Blood-flow velocities and their relationships in carotid and middle cerebral arteries," *Ultrasound in medicine & biology*, vol. 24, no. 8, pp. 1131–1136, Oct. 1998, doi: https://doi.org/10.1016/s0301-5629(98)00092-1.

[28] "Physical Principles Of Intra-arterial Blood Pressure Measurement," *WFSA Resource Library*. https://resources.wfsahq.org/atotw/physical-principles-of-intra-arterial-blood-pressure-measurement/

[30] S. Fantini, A. Sassaroli, K. T. Tgavalekos, and J. Kornbluth, "Cerebral blood flow and autoregulation: current measurement techniques and prospects for noninvasive optical methods," *Neurophotonics*, vol. 3, no. 3, p. 031411, Jun. 2016, doi: https://doi.org/10.1117/1.nph.3.3.031411.

[31] F. C. Gallardo *et al.*, "Novel Simulation Model with Pulsatile Flow System for Microvascular Training, Research, and Improving Patient Surgical Outcomes," *World Neurosurgery*, vol. 143, pp. 11–16, Nov. 2020, doi: https://doi.org/10.1016/j.wneu.2020.07.116.

[32] U. Cikla, B. Sahin, S. Hanalioglu, A. S. Ahmed, D. Niemann, and M. K. Baskaya, "A novel, low-cost, reusable, high-fidelity neurosurgical training simulator for cerebrovascular bypass surgery," *Journal of Neurosurgery*, vol. 130, no. 5, pp. 1663–1671, May 2019, doi: https://doi.org/10.3171/2017.11.jns17318.

[33] M. J. Cipolla, *Control of Cerebral Blood Flow*. Morgan & Claypool Life Sciences, 2009. Available: https://www.ncbi.nlm.nih.gov/books/NBK53082/

[34] M. S. Alnæs, J. Isaksen, MardalK.-A., B. Romner, M. K. Morgan, and T. Ingebrigtsen, "Computation of Hemodynamics in the Circle of Willis," *Stroke*, vol. 38, no. 9, pp. 2500–2505, Sep. 2007, doi: https://doi.org/10.1161/strokeaha.107.482471.

[35] K. DEVAULT, P. A. GREMAUD, V. NOVAK, M. S. OLUFSEN, G. VERNIÈRES, and P. ZHAO, "BLOOD FLOW IN THE CIRCLE OF WILLIS: MODELING AND CALIBRATION," *Multiscale modeling & simulation : a SIAM interdisciplinary journal*, vol. 7, no. 2, pp. 888–909, Jan. 2008, doi: https://doi.org/10.1137/07070231X.

[36] Isoda, H., Hirano, M., Takeda, H., Kosugi, T., Alley, M. T., Markl, M., Pelc, N. J., & Sakahara, H. (2006). Visualization of Hemodynamics in a Silicon Aneurysm Model Using Time-Resolved, 3D, Phase-Contrast MRI. *American Journal of Neuroradiology*, *27*(5), 1119–1122.

[37] W. Tsai and O. Savas, "Flow pumping system for physiological waveforms," *Medical & Biological Engineering & Computing*, vol. 48, no. 2, pp. 197–201, Jan. 2010, doi: https://doi.org/10.1007/s11517-009-0573-6.

[38] N. Kaneko *et al.*, "In Vitro Modeling of Human Brain Arteriovenous Malformation for Endovascular Simulation and Flow Analysis," *World Neurosurgery*, vol. 141, pp. e873–e879, Sep. 2020, doi: https://doi.org/10.1016/j.wneu.2020.06.084.

[39] S. E. Herwald, D. Y. Sze, D. B. Ennis, and A. M. Vezeridis, "Design and implementation of a cost-effective, open-source, and programmable pulsatile flow system," *HardwareX*, vol. 19, pp. e00561–e00561, Jul. 2024, doi: https://doi.org/10.1016/j.ohx.2024.e00561.

[40] S. Drost, B. J. de Kruif, and D. Newport, "Arduino control of a pulsatile flow rig," *Medical Engineering & Physics*, vol. 51, pp. 67–71, Jan. 2018, doi: https://doi.org/10.1016/j.medengphy.2017.10.006.

[41] A. Menkara, A. Faryami, D. Viar, and C. Harris, "Applications of a novel reciprocating positive displacement pump in the simulation of pulsatile arterial blood flow," *PLoS ONE*, vol. 17, no. 12, pp. e0270780–e0270780, Dec. 2022, doi: https://doi.org/10.1371/journal.pone.0270780.

[42] A. McCulloch *et al*, "Evaluation of vessel injury after simulated catheter use in an endothelialized silicone model of the intracranial arteries," *Neuroradiology,* vol. 65, *(10),* pp. 1507-1515, 2023. Available: http://ezproxy.lib.torontomu.ca/login?url=https://www.proquest.com/scholarly-journals/evaluatio

n-vessel-injury-after-simulated-catheter/docview/2864005159/se-2. DOI: https://doi.org/10.1007/s00234-023-03197-8.

[43] J. D. Smirl, K. Hoffman, Y.-C. Tzeng, A. Hansen, and P. N. Ainslie, “Relationship between blood pressure and cerebral blood flow during supine cycling: influence of aging,” *Journal of Applied Physiology*, vol. 120, no. 5, pp. 552–563, Mar. 2016, doi: https://doi.org/10.1152/japplphysiol.00667.2015.

Appendices

Arduino Due Pinout sheet: https://docs.arduino.cc/resources/pinouts/A000056-full-pinout.pdf

Arduino Due datasheet: https://docs.arduino.cc/resources/datasheets/A000062-datasheet.pdf

Honeywell Datasheet: https://prod-edam.honeywell.com/content/dam/honeywell-edam/sps/siot/es-mx/products/sensors/pressure-sensors/board-mount-pressure-sensors/micropressure-mpr-series/documents/sps-siot-mpr-series-datasheet-32332628-ciid-172626.pdf

Normal flow that converts measured flow rate into pressure Arduino code (Used for Engineering Day):

```
#include <math.h>  // For pow()



// --- TB6600 + Stepper Motor Pins ---

const int dirPin = 4;          // DIR+

const int stepPin = 3;         // PUL+

const int enablePin = 5;       // ENA+



// --- Flow Sensor Setup ---

const byte flowPin = 8;        // Flow sensor

volatile int flowPulses = 0;



// --- Microstepping and Motor Specs ---

const int microstepping = 1;                  // 1/4 step mode

const int stepsPerRevolution = 200;           // Full steps/rev

const int stepsPerRevMicro = stepsPerRevolution * microstepping;
```

```
// --- Flow Control Settings ---
const int desiredFlowRate = 1100;              // mL/min (target flow)
const int runTimeSeconds = 10;                 // Pump runtime

// --- Timing Variables ---
unsigned long previousMillis = 0;
unsigned long intervalMillis = 1500;           // Read every 1.5 sec
int currentSecond = 0;

float delayMicrosecondsPerStep = 0;

// --- Poiseuille Variables ---
const float viscosity = 0.004;                    // Pa·s (viscosity of 60% water and 40% glycerol)
const float tubingLength = 0.965;                    // meters (length of tubing between sensor and output)
const float tubingRadius = 0.003175;            // meters (1/4" ID tubing radius)

void flowSensorISR() {
  flowPulses++;
}

void setup() {
```

```
  Serial.begin(115200);


  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(flowPin, INPUT_PULLUP);


  attachInterrupt(digitalPinToInterrupt(flowPin), flowSensorISR, RISING);


  // Enable motor and set direction
  digitalWrite(enablePin, LOW);    // LOW = enabled
  digitalWrite(dirPin, HIGH);      // Direction set


// --- Calculate RPM from desired flow rate ---
// KK2000 Peristaltic Pump: flow ≈ 13.3 * RPM - 633
  float rpm = (desiredFlowRate + 633.0) / 13.3;
  rpm = constrain(rpm, 250.0, 350.0);  // Prevent overheating
  float stepsPerSec = (rpm * stepsPerRevMicro) / 60.0;
  delayMicrosecondsPerStep = 1000000.0 / stepsPerSec;


  Serial.println("Time (s)\tFlow (mL/min)\tPressure (mmHg)");
  Serial.println("=========================================");
}
```

```
void loop() {
  // Stop after target time
  if (currentSecond >= runTimeSeconds) {
    digitalWrite(enablePin, HIGH); // Disable motor
    Serial.println("\nPump stopped.");
    while (1); // Stop further loop execution
  }

  // --- Stepper Pulse Generation ---
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(delayMicrosecondsPerStep / 2);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(delayMicrosecondsPerStep / 2);

  // --- Flow Rate Calculation Every 1s ---
  if (millis() - previousMillis >= intervalMillis) {
    previousMillis = millis();

    int pulses = flowPulses;
    flowPulses = 0;

      // Flow (mL/min) = Frequency * (1000 / 98) = Frequency * 10.204;
DIGITEN has 98*Q in L/min for frequency
```

```
    float flowRate = pulses * 10.204;



    // Convert mL/min → m³/s

    float Q = (flowRate / 1000000.0) / 60.0;



      // Poiseuille's Law: ΔP (mmHg) = (8 * μ * L * Q) / (π * r^4) *
0.00750062

     float pressureDrop = (8.0 * viscosity * tubingLength * Q) / (PI *
pow(tubingRadius, 4)) * 0.00750062;



    // --- Serial Plotter Output ---

    Serial.print(currentSecond);

    Serial.print("\t");

    Serial.print(flowRate, 1);        // Flow in mL/min

    Serial.print("\t");

    Serial.println(pressureDrop, 2); // Pressure in Pascals



    currentSecond++;

  }

}
```

First failed pulsatile flow Arduino code:

```
// --- Motor Control ---

const int DIR_PIN = 4; //DIR+
```

```
const int STEP_PIN = 3; //PUL+

const int ENABLE_PIN = 5; //ENA+

const int STEPS_PER_REV = 1600;                // Updated: microstepping resolution

const float ML_PER_REV = 2.12;            // Calibrated volume per rev



// --- Flow Sensor Setup ---

const int FLOW_SENSOR_PIN = 8;

volatile unsigned long flowPulseCount = 0;

float flowRate = 0;



// --- Flow Constants ---

enum Artery { ICA, MCA };

Artery arteryMode = ICA;



float SCALE_FACTOR = 1.5; // Use 1.2-1.5 for ICA, 3.333 for MCA for initial scaling up with model from VY08

unsigned int cardiacCycles = 20; // Number of cardiac cycles to simulate



// Example average ICA flow pattern (mL/min)

float ICA_PEAK = 802;

float ICA_DIASTOLIC = 267;



// Example average MCA flow pattern (mL/min)
```

```
float MCA_PEAK = 265;

float MCA_DIASTOLIC = 113;



float pulseFrequency = 1.0; // 1 Hz = 60 BPM

float shapedSine(float t);



// --- Setup ---

void setup() {

  Serial.begin(115200);

  pinMode(STEP_PIN, OUTPUT);

  pinMode(DIR_PIN, OUTPUT);

  pinMode(ENABLE_PIN, OUTPUT);

  digitalWrite(DIR_PIN, HIGH);

  digitalWrite(ENABLE_PIN, HIGH);  // Disable motor initially



  pinMode(FLOW_SENSOR_PIN, INPUT_PULLUP);

   attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN), countFlowPulses,
RISING);



  delay(1000);  // Allows everything to settle

  Serial.println("Setup complete. Running ICA flow simulation...");



  digitalWrite(ENABLE_PIN, LOW); // Enable motor before starting

  runPulsatileSimulation();
```

```
}



void loop() {

  // simulation runs once

}



// --- Main Full Pulsatile Simulation ---

void runPulsatileSimulation() {

  float peak = (arteryMode == ICA ? ICA_PEAK : MCA_PEAK) * SCALE_FACTOR;

  float diastolic = (arteryMode == ICA ? ICA_DIASTOLIC : MCA_DIASTOLIC) *
SCALE_FACTOR;



  // Safety flow rate limits

  diastolic = max(diastolic, 250.0);

  peak = min(peak, 340.0 * ML_PER_REV); // 340 RPM upper bound as 350
causes overheating



  float amplitude = (peak - diastolic) / 2.0;

  float baseline = (peak + diastolic) / 2.0;

  float cycleDuration = 1.0 / pulseFrequency;

  float stepInterval = 0.005; // Higher resolution



  for (int cycle = 1; cycle <= cardiacCycles; cycle++) {

    Serial.print("Cycle ");
```

```
    Serial.println(cycle);



    flowPulseCount = 0; // Reset pulse count for this cycle

    unsigned long cycleStart = millis();



    for (float t = 0; t < cycleDuration; t += stepInterval) {

      float shaped = shapedSine(t);

      float targetFlow = baseline + amplitude * shaped;

      targetFlow = constrain(targetFlow, diastolic, peak);



      float rpm = targetFlow / ML_PER_REV;

      rpm = constrain(rpm, 250, 340);  // Safe RPM range



      float stepsPerSecond = rpm * STEPS_PER_REV / 60.0;

      float stepDelay = 1000000.0 / (stepsPerSecond * 2.0);  // Half cycle
for HIGH/LOW



      digitalWrite(STEP_PIN, HIGH);

      delayMicroseconds((int)stepDelay);

      digitalWrite(STEP_PIN, LOW);

      delayMicroseconds((int)stepDelay);

    }



    // Calculate flow after 1 cardiac cycle
```

```
    unsigned long durationMs = millis() - cycleStart;
    calculateFlowRate(durationMs);

    Serial.print("  Target Flow (end): ");
    Serial.print(peak, 1);
    Serial.print(" mL/min | Actual Flow: ");
    Serial.print(flowRate, 1);
    Serial.println(" mL/min");
  }

  digitalWrite(ENABLE_PIN, HIGH); // Disable the motor
  Serial.println("Simulation complete.");
  while (true); // Stop the program
}

// --- Systolic-Diastolic Pulse Waveform ---
float shapedSine(float t) {
  float sine = sin(2 * PI * pulseFrequency * t);
  float systolic = pow(max(sine, 0.0), 3); // Sharper peak
  float diastolic = 0.3 * pow(min(sine, 0.0), 2); // Gentle trough
  return systolic - diastolic;
}
```

```
// --- Flow Sensor ISR ---

void countFlowPulses() {

  flowPulseCount++;

}



// --- Flow Calculation ---

void calculateFlowRate(unsigned long durationMs) {

  const float SENSOR_CONSTANT = 98.0; // Hz per L/min for DIGITEN

  noInterrupts();

  unsigned long pulses = flowPulseCount;

  flowPulseCount = 0;

  interrupts();



  float freq = (float)pulses * 1000.0 / durationMs;    // Hz

  float L_per_min = freq / SENSOR_CONSTANT;

  flowRate = L_per_min * 1000.0;  // Convert to mL/min

}
```

Second failed pulsatile flow Arduino code:

```
// --- TB6600 + Stepper Motor Pins ---

const int dirPin = 4; //DIR+
```

```
const int stepPin = 3; //PUL+

const int enaPin = 5; //ENA+



const int microstepping = 1; // 1/4 step mode

const int stepsPerRevolution = 200; //Full steps/rev

const int stepsPerRevMicro = stepsPerRevolution * microstepping;



const char vesselType[] = "ICA"; // "ICA" or "MCA"

const int cardiacCycles = 20;     // Number of cardiac cycles

const int bpm = 80;



const int cycleSamples = 50; // Sample number of waveform samples per cardiac cycle

unsigned long cycleDurationMs = 60000 / bpm; // One cardiac cycle duration (ms)

unsigned long sampleIntervalMs = cycleDurationMs / cycleSamples;



// ICA waveform (normalized 0-1, 50 samples)

float icaWaveform[50] = {

  0.35, 0.42, 0.50, 0.60, 0.72, 0.84, 0.92, 1.00, 0.94, 0.88,

  0.81, 0.75, 0.70, 0.65, 0.62, 0.60, 0.58, 0.57, 0.55, 0.53,

  0.52, 0.51, 0.50, 0.49, 0.47, 0.45, 0.44, 0.42, 0.41, 0.40,

  0.39, 0.38, 0.38, 0.38, 0.39, 0.40, 0.42, 0.44, 0.47, 0.50,

  0.53, 0.56, 0.60, 0.65, 0.70, 0.76, 0.83, 0.90, 0.97, 1.00
```

```
};


// MCA waveform (normalized 0-1, 50 samples)
float mcaWaveform[50] = {
  0.40, 0.48, 0.56, 0.66, 0.78, 0.89, 1.00, 0.95, 0.90, 0.85,
  0.80, 0.76, 0.72, 0.69, 0.66, 0.64, 0.62, 0.60, 0.59, 0.58,
  0.57, 0.56, 0.55, 0.54, 0.52, 0.50, 0.48, 0.46, 0.45, 0.43,
  0.42, 0.41, 0.40, 0.40, 0.41, 0.43, 0.46, 0.50, 0.55, 0.60,
  0.65, 0.70, 0.76, 0.82, 0.88, 0.93, 0.97, 0.99, 1.00, 0.98
};


// ICA and MCA flow boundaries (mL/min)
float systolicFlowICA = 2000.0;
float diastolicFlowICA = 800.0;


float systolicFlowMCA = 1248.0;
float diastolicFlowMCA = 700.0;


void setup() {
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  digitalWrite(enablePin, LOW); // LOW = enabled
```

```
  digitalWrite(dirPin, HIGH); // Direction set

  Serial.begin(115200);

  delay(1000);

}



void loop() {

  float* waveform;

  float systolicFlow, diastolicFlow;



  if (strcmp(vesselType, "ICA") == 0) {

    waveform = icaWaveform;

    systolicFlow = systolicFlowICA;

    diastolicFlow = diastolicFlowICA;

  } else {

    waveform = mcaWaveform;

    systolicFlow = systolicFlowMCA;

    diastolicFlow = diastolicFlowMCA;

  }



  int totalSamples = cardiacCycles * cycleSamples;



  for (int i = 0; i < totalSamples; i++) {

    int sampleIndex = i % cycleSamples;
```

```
    float normalized = waveform[sampleIndex];

      float currentFlow = diastolicFlow + normalized * (systolicFlow - diastolicFlow);



     float rps = (currentFlow / 1000.0) / 0.074; // Approx flow/rev (74 mL/rev)

    float delayMicros = (1.0 / (rps * stepsPerRevMicro)) * 1000000.0;



    digitalWrite(stepPin, HIGH);

    delayMicroseconds(delayMicros / 2);

    digitalWrite(stepPin, LOW);

    delayMicroseconds(delayMicros / 2);



    delay(sampleIntervalMs);

  }



  // Stop motor

  digitalWrite(enablePin, HIGH);

  while (1);

}
```