

University of Arkansas, Fayetteville

ScholarWorks@UARK

---

Mechanical Engineering Undergraduate Honors  
Theses

Mechanical Engineering

---

5-2021

## Design and Control of a Peristaltic Pump to Simulate Left Atrial Pressure in a Conductive Silicone Model

Jeremy Collins

Follow this and additional works at: <https://scholarworks.uark.edu/meeguht>



Part of the [Biology and Biomimetic Materials Commons](#), [Biomaterials Commons](#), [Biomedical Devices and Instrumentation Commons](#), [Cardiovascular System Commons](#), [Controls and Control Theory Commons](#), [Dynamics and Dynamical Systems Commons](#), [Electro-Mechanical Systems Commons](#), [Fluid Dynamics Commons](#), [Service Learning Commons](#), [Signal Processing Commons](#), and the [Statistical, Nonlinear, and Soft Matter Physics Commons](#)

---

### Citation

Collins, J. (2021). Design and Control of a Peristaltic Pump to Simulate Left Atrial Pressure in a Conductive Silicone Model. *Mechanical Engineering Undergraduate Honors Theses* Retrieved from <https://scholarworks.uark.edu/meeguht/107>

This Thesis is brought to you for free and open access by the Mechanical Engineering at ScholarWorks@UARK. It has been accepted for inclusion in Mechanical Engineering Undergraduate Honors Theses by an authorized administrator of ScholarWorks@UARK. For more information, please contact [ccmiddle@uark.edu](mailto:ccmiddle@uark.edu).

# **Design and Control of a Peristaltic Pump to Simulate Left Atrial Pressure in a Conductive Silicone Model**

An Honors Thesis submitted in partial fulfillment of the requirements for Honors Studies in Mechanical Engineering

by Jeremy Collins

Spring 2021

Department of Mechanical Engineering

College of Engineering

University of Arkansas

## **ABSTRACT**

According to the CDC, atrial fibrillation is responsible for more than 454,000 hospitalizations and approximately 158,000 deaths per year. A common treatment for atrial fibrillation is catheter ablation, a process in which a long flexible tube is guided through the femoral artery and to the source of arrhythmia in the heart, where it measures the electrical potential at various locations and converts problematic heart tissue to scar tissue via ablation. This paper details the design and control of a low-cost (\$400) peristaltic pump system using repetitive control to replicate blood pressure in the left atrium in a conductive silicone model for use in modeling catheter ablation. Using repetitive control, an average root-mean-square error of 0.038 psi was achieved.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	4
INTRODUCTION.....	5
METHODS.....	8
A. Design and Fabrication.....	8
B. Control System Design.....	12
RESULTS.....	17
CONCLUSIONS AND DISCUSSION.....	20
REFERENCES.....	22
APPENDICES.....	25
Appendix A: Additional Figures.....	25
Appendix B: Arduino Code.....	27

## **ACKNOWLEDGEMENTS**

I would like to give my sincere thanks to Dr. Yue Chen for providing me the opportunities and resources to perform robotics research. His guidance allowed my growth as an engineer, and ultimately led to my decision to pursue robotics as a career. I would also like to express my thanks to Dr. Uche Wejinya for serving on my thesis defense committee. Dr. Wejinya solidified my interest in robotics and controls with daily discussions of fundamental principles and how concepts relate to one another. Finally, I would like to thank Anthony Gunderman for his continual support as my direct supervisor in the lab and his consistently insightful assistance in the ideation process.

## INTRODUCTION

Atrial fibrillation is a form of heart arrhythmia characterized by rapid or irregular electrical activity in the atria [1]. According to the CDC, atrial fibrillation is responsible for more than 454,000 hospitalizations and approximately 158,000 deaths per year [2]. A common treatment for atrial fibrillation is catheter ablation, a process in which a long flexible tube is guided through the femoral artery and to the source of arrhythmia in the heart. The catheter measures the electrical potential at various locations throughout the heart, and this information is processed to locate the heart tissue causing the arrhythmia. The catheter then converts this tissue to scar tissue via ablation at very high or low temperatures to render it electrically inactive [1].

A common biological mechanism for transporting fluids is peristalsis. Peristalsis occurs with the wavelike propagation of contraction within a flexible tube, thus pushing a fluid inside the tube in the direction of the contraction. This is typically a strictly biological process occurring in the esophagus and intestines of vertebrate animals. A peristaltic pump utilizes this principle by using two or more rollers to squeeze a flexible tube along a semicircular housing for the purpose of transporting a fluid. Peristaltic pumps are a type of positive displacement pump used in a wide variety of applications such as transporting bodily fluids for medical applications, pumping concrete, and dispensing beverages [3]. Reasons for using a peristaltic pump include preserving the sterility and physical integrity of a fluid and avoiding pump maintenance associated with the wear and tear caused by harsh fluids.

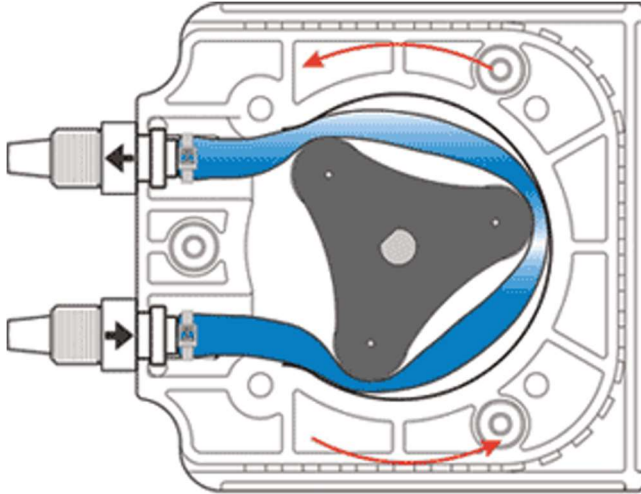


Figure 1. Peristaltic pump operation [4].

Flow in a peristaltic pump may be approximated by the following equation:

$$Q = an$$

where  $Q$  is flow,  $a$  is the flow coefficient, and  $n$  is the rotor speed [5].

The flow coefficient  $a$  is typically an experimentally determined value that is dependent on the tube diameter, roller diameter, number of rollers, and tube material. The deformation of the flexible tube also changes  $a$ , but it tends to settle at low rotor speeds after extensive use. At high rotor speeds, however, flow is not proportional to rotor speed due to increased backflow and increased tube deformation. For this reason, rotor speed and acceleration limits will be imposed on the controller in order to limit the time spent operating in the nonlinear zone.

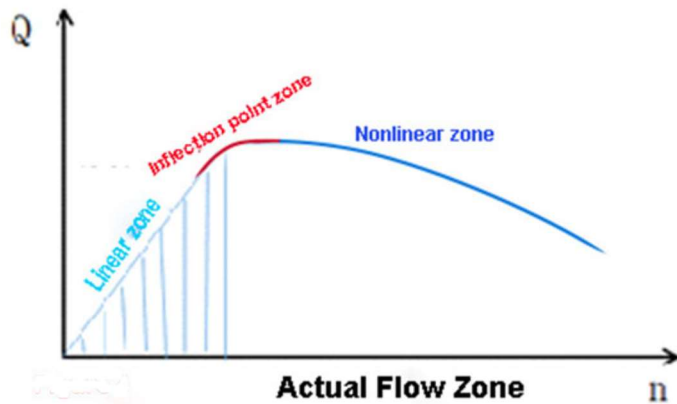


Figure 2. Peristaltic pump flow vs. rotor speed [5].

Analytical solutions for the outlet pressure in a peristaltic pump are not well defined due to pressure oscillations at the output. Pressure is often instead analyzed with finite element analysis software such as Abaqus and COMSOL [19].

The objective of this project is to design and control a peristaltic pump to simulate blood pressure in a conductive silicone left atrium for use in modeling catheter ablation. Current methods of modeling the human heart exist on the market [6, 7], but cost thousands of dollars and are not capable of accurately providing voltage measurements associated with catheter ablation due to the electrically insulative properties of silicone. Additionally, flow control [8] and speed control [9] are most commonly used in peristaltic pumps simulating blood flow, but pressure control is less often implemented, leading to inaccurate modeling of forces exerted on the catheter.

There are a variety of established methods for the control of peristaltic pumps. Flow control is most commonly used, often implemented as classical PI/PID control [10], fuzzy control [8], or repetitive control [11, 16, 17]. In this paper, similar methods will be attempted in order to track a periodic setpoint via pressure control.



## METHODS

### *A. Design and Fabrication*

A 3D model of a left atrium was first created using a volumetric model of a human left atrium rendered by OsiriX, an image processing application, in an MRI scanner. The model was then manually edited to create smooth and continuous geometry.

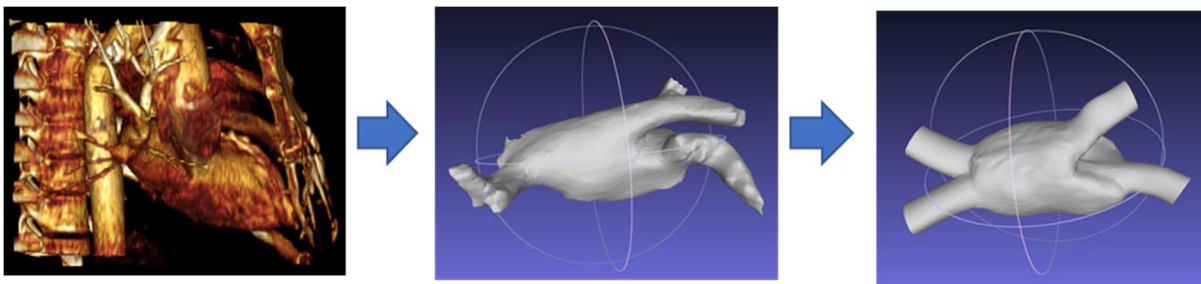


Figure 3. Left atrium 3D model.

Based on this 3D model, four outer molds and one inner mold were designed and 3D printed via stereolithography with a 100-micron resolution. Using these molds, an atrium model was manufactured out of conductive silicone. To replicate the heart's electrical resistivity of approximately  $3.5 \Omega \cdot \text{m}$  [12], acetylene black carbon nanoparticles were added to a two-part silicone (Dragon Skin FX Pro, Smooth-On) at a concentration of 10 parts per hundred rubber (PHR). This concentration was arrived on with the sequential manufacturing and resistivity measurement of conductive silicone cylinders, beginning with values per [13]. Once the correct concentration was determined, a 10 PHR conductive silicone mixture was placed between the inner and outer molds and was allowed to cure. A small cylindrical sample was also extracted from this mixture and its resistivity was determined to be 4.16

$\Omega^*$ m. Once released from the outer mold, the inner mold was cut out of the atrium and the incision was sealed with a silicone adhesive (DOWSIL 732 Multi-Purpose Sealant).



Figure 4. Atrium molds.



Figure 5. Conductive silicone atrium.

A simple and cost-effective peristaltic pump was then designed and additively manufactured out of polylactic acid (PLA) with three rollers composed of 8mm bore, 22mm diameter ball bearings. The pump was designed for 3/8 inch inner diameter, 1/2 inch outer diameter silicone tubing, and minimized backflow by compressing the tube such that the

inner walls formed a watertight seal. The pump's rotor was driven by a NEMA 34 closed-loop stepper motor driven by a STEPPERONLINE CL86Y motor driver.

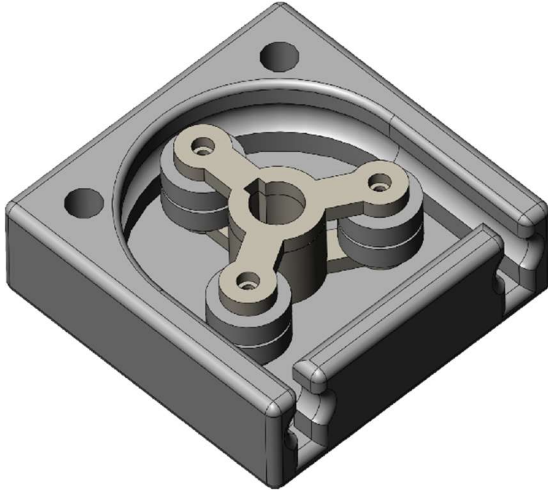


Figure 6. Peristaltic pump design.

Silicone tubing was placed between the rotor and housing of the pump and one end of the tube was placed in a 1000 mL water source. The output of the peristaltic pump was then fixed to a high-precision pressure transducer (WIKA A-10) and fed into the inlet of the conductive silicone atrium. Four smaller silicone tubes (1/4 inch ID, 3/8 inch OD) were then placed between the outlets of the atrium and the water source. The 10V analog output of the pressure transducer was stepped down to 5V with a voltage divider and the input impedance of the microcontroller was matched with a unity gain buffer. The pressure transducer input and step pin were powered with a 16V source to prevent signal interference, and the step pin was controlled with a 5V signal via an N-type MOSFET (P60NF06L).

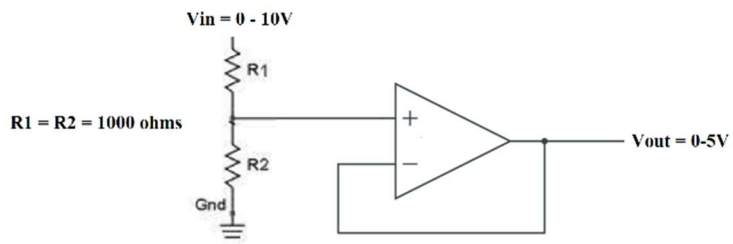


Figure 7. Unity buffer op-amp circuit.



Figure 8. Experimental setup.

## B. Control System Design

Many different controllers were designed and tested, each of which adjusted the stepper motor's rotational speed to control the outlet pressure of the peristaltic pump. The objective was to achieve a periodic setpoint corresponding to pressure in the human left atrium. A pressure curve from Braunwald et. al (1961) [14] was scanned with the MATLAB GRABIT tool to transcribe this data. The data was then approximated with an 8<sup>th</sup> order Fourier approximation to enable discretization at arbitrary intervals.

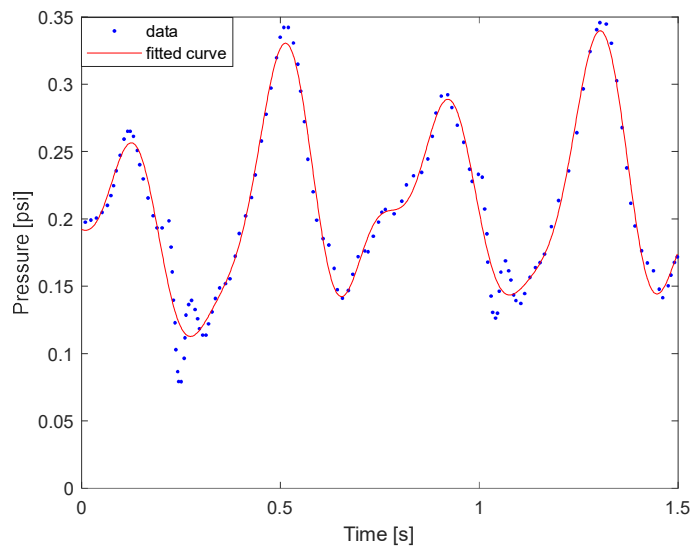


Figure 9. 8<sup>th</sup> order Fourier approximation of setpoint.

The system was first controlled with a classical PD controller with an unfiltered signal. This caused a significant amount of high-frequency disturbance, and it was apparent that the system was not controllable without filtering the pressure. A moving average filter was applied to the pressure transducer output, and a root-mean-square (RMS) error of 0.055 psi was achieved.

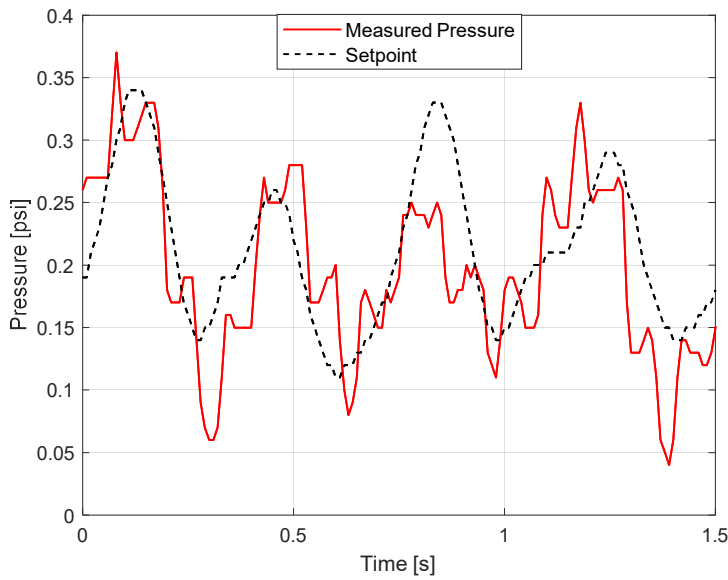


Figure 10. Rolling average filter, classical PD control

To filter the high-frequency noise more effectively, a finite impulse response (FIR) filter was constructed with unity gain from 0 to 3 Hz and a -40 dB gain above 8 Hz. These parameters were determined by observing the fast Fourier transform (FFT) of the setpoint function as well as the unfiltered response at a constant rotor velocity of 5 rad/s in order to differentiate frequencies associated with achieving the setpoint and frequencies which can be attributed to measurement noise.

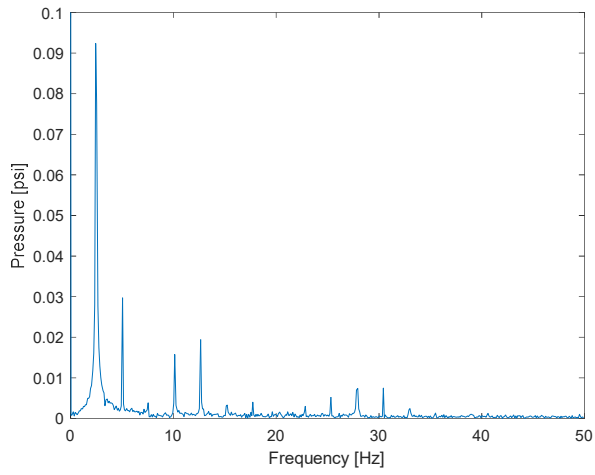


Figure 11. Fast Fourier transform of unfiltered response to a constant rotor velocity.

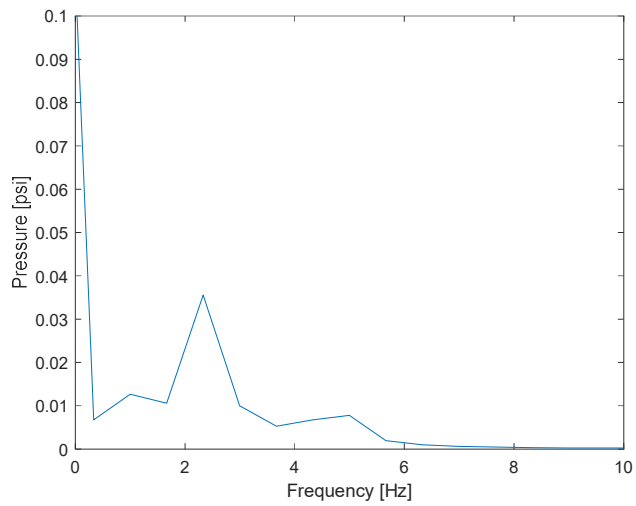


Figure 12. Fast Fourier transform of discretized setpoint function.

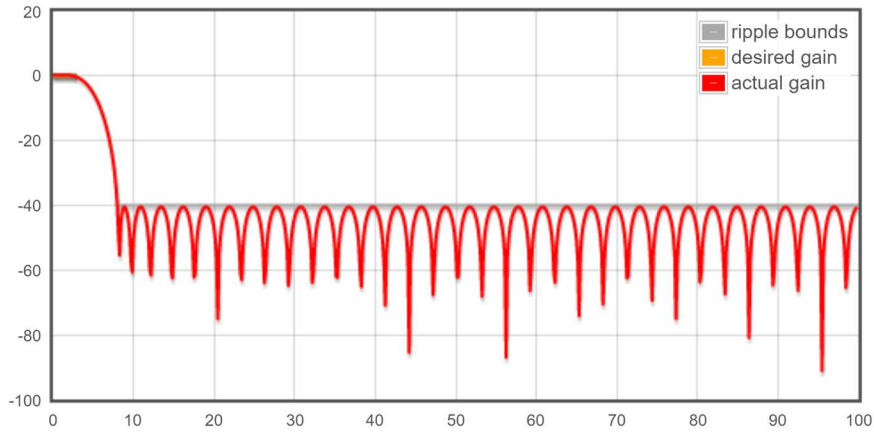


Figure 13. FIR filter gain (dB) vs. frequency (Hz) [15]

An integral term was then incorporated to create a classical PID controller, which provided a much smoother pressure response and a root-mean-square error of 0.101 psi.

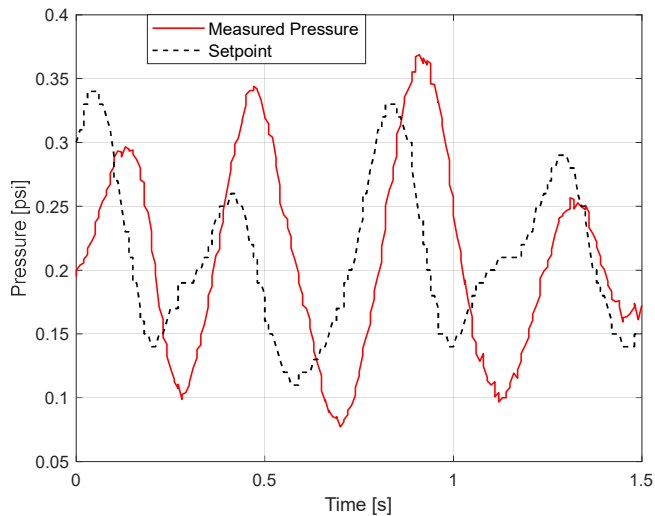


Figure 14. Typical performance for controller with FIR filter and classical PID control (RMS error of 0.101 psi).

This control strategy continued to have significant pitfalls. Although the controller was capable of tracking the setpoint with relative accuracy for small intervals of time, a phase shifting occurred which prevented the controller from maintaining this accuracy, causing the



error to drastically increase and decrease at regular intervals. Additionally, large high-frequency pressure spikes were observed in the unfiltered data that were a result of aggressive controller compensation rather than measurement noise, which created a misleading representation of the output pressure.

To mitigate these issues, a repetitive controller similar to controllers used for sinusoidal setpoint tracking [11] and the rejection of sinusoidal disturbances [16], was implemented. This new control strategy was a PI controller with a gain proportional to the setpoint function before the plant, using the current value of the setpoint rather than a typical time-delay model. This gain eliminated phase shifting and stabilized the steady-state error by providing a model of the input signal which biased the output to be in phase with the setpoint.

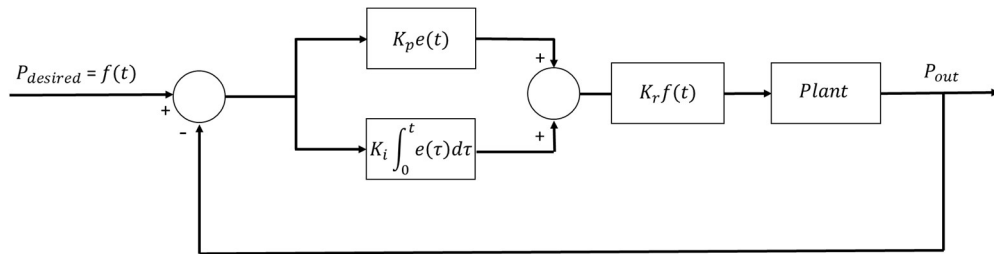


Figure 15. Repetitive controller block diagram

## RESULTS

A repetitive controller with a proportional gain of 500000, an integral gain of 1250, and a control signal gain of  $1f(t)$  was successfully implemented to allow the peristaltic pump to track a periodic pressure setpoint. A rotor speed limit of 25 rad/s and a rotor acceleration limit of 20 rad/s<sup>2</sup> were imposed on the system to maximize linearity, and the stepper motor was actuated at a resolution of 51200 steps per revolution to minimize oscillation. Over a testing period of 60 seconds, the repetitive controller successfully tracked the setpoint with a root-mean-square error of 0.038 psi. Because a filter was not applied during measurement, the data was smoothed after collection by taking the moving average of the output pressure with a window of 0.1 seconds. The maximum root-mean-square error within one period of the setpoint function (1.5 seconds) was 0.053 psi, and the minimum root-mean-square error was 0.024 psi.

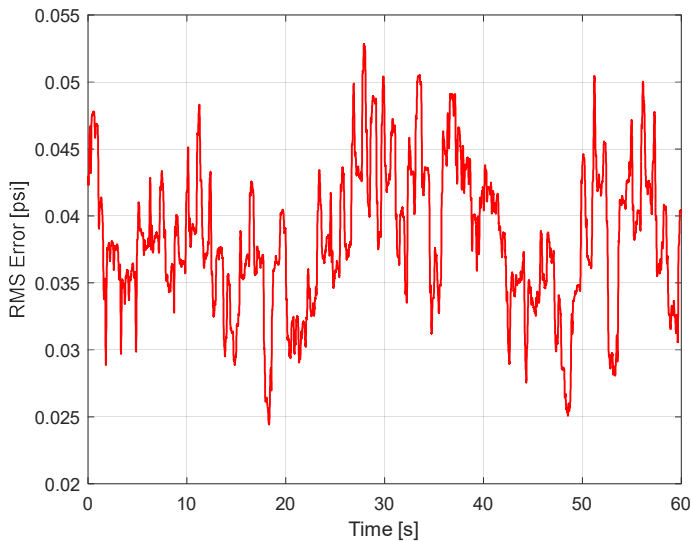


Figure 16. Repetitive controller RMS error vs. time over measurement window  $T = 1.5$  seconds.

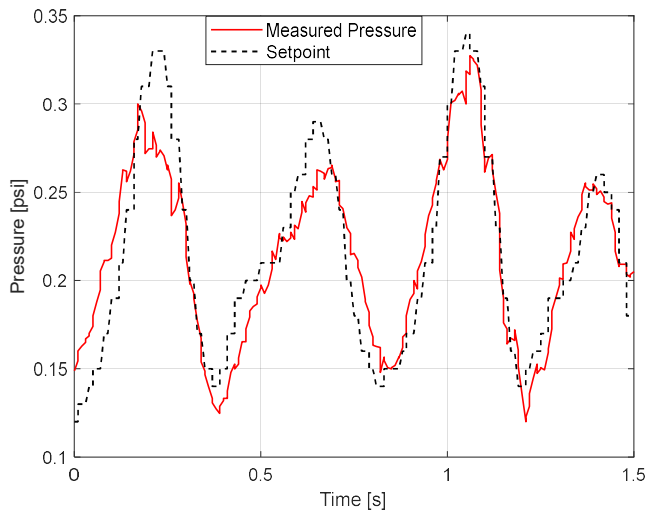


Figure 17. Repetitive controller, minimum root-mean-square error (0.024 psi).

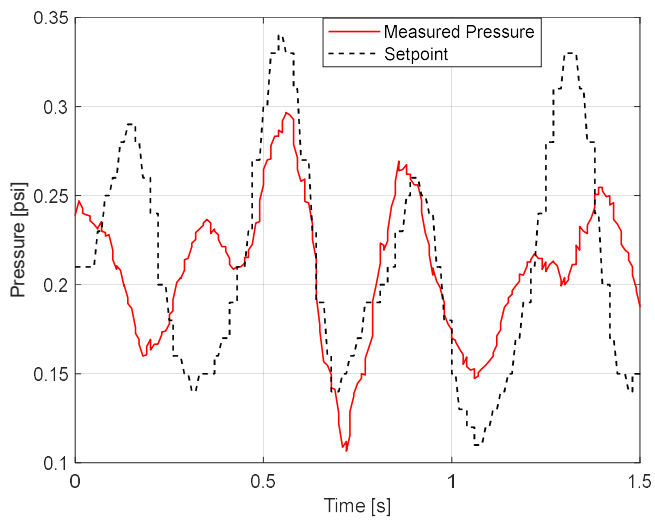


Figure 18. Repetitive controller, maximum root-mean-square error (0.053 psi).

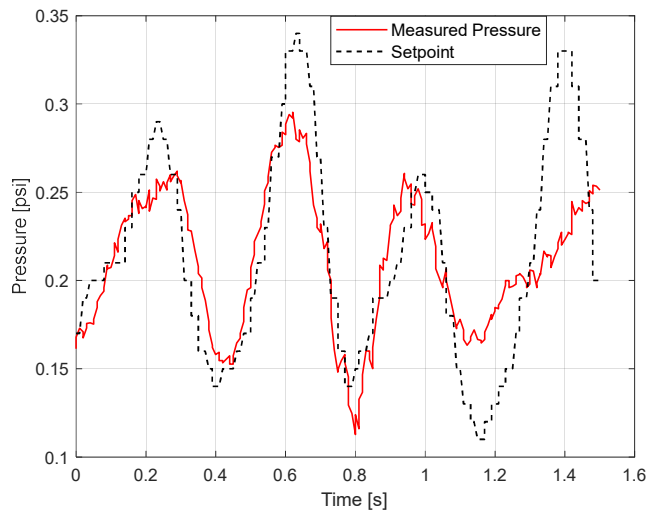


Figure 19. Typical controller performance. Root-mean-square error of 0.038 psi.

## CONCLUSIONS AND DISCUSSION

Repetitive control was found to be an effective method of controlling pressure in a conductive silicone model with a peristaltic pump, and an average root-mean-square error of 0.038 psi was achieved. A novel conductive silicone model of the human left atrium was designed and manufactured using a two-part silicone embedded with 10 parts per hundred rubber of acetylene black carbon nanoparticles. The peristaltic pump and conductive atrium system presented in this paper is a practical, cost-effective (\$400) solution for modeling catheter ablation.

The repetitive controller performed acceptably at tracking pressure in the human left atrium; however, there are many ways in which the system may be improved. Firstly, the system is not consistent enough for medical use. It is important that the behavior of the pump is predictable with consistent and minimal error values. It is also important that the local minima and maxima of the pressure output are accurate, whereas the current system often undershoots these values. Additionally, the performance of the controller is highly dependent on the average pressure being equal to that of the average setpoint value. This system might be made more robust with the addition of a method of controlling the pressure within the water reservoir to regulate the hydrostatic pressure in the atrium model.

The necessity to make an incision in the conductive atrium to release it from the inner mold is a major flaw in the manufacturing process, as it has a significant impact on resistivity. This issue may be mitigated with the implementation of a dissolvable inner mold. The use of a 3D-printable material such as polyvinyl alcohol (PVA) for the inner mold would allow it to dissolve in water, leaving the conductive atrium unscathed. Additionally, the lack of an automated process to simplify the geometry of volumetric models from MRI scans makes generating

patient-specific models a laborious task. Simplification of the mesh using an automated script in 3D graphics software such as Blender may be a suitable solution.

Although flow control was not implemented, flow rate must be considered, and more thought should be put toward aspects of the mechanical design, such as tube diameter and material, rotor and housing geometry, etc., to ensure that the flow rate is consistent with that of the human left atrium. Further tuning of the mechanical design should also be performed to minimize the nonlinear aspects of the pump's mechanics such as backflow, hysteresis and fatigue of the tubing, and disturbances generated by the impact on and release of the rollers from the tube. The latter two issues may possibly be mitigated with the use of spring-loaded rollers to supply a more constant pressure within the tubing and minimize the impact of the rollers.

Future work includes the tuning of the peristaltic pump's mechanical design to minimize nonlinear behavior, as well as the implementation of more complex control schemes. In particular, it would be useful to obtain the transfer function of the plant by observing its response to step and ramp inputs, and then creating a controller using a simplified version of the plant. A controller made in the frequency domain may be converted to a  $Z$  transform and implemented as an infinite impulse response filter in a microcontroller or may more simply be implemented as an RLC circuit. Lastly, because peristaltic pumps exhibit some nonlinear properties, it may follow to implement nonlinear control methods such as sliding mode control or gain scheduling.

## REFERENCES

- [1] Taylor, A. J., Chen, Y., Fok, M., Berman, A., Nilsson, K., and Ho Tse, Z. T. (June 27, 2017). "Cardiovascular Catheter With an Expandable Origami Structure." ASME. J. Med. Devices. September 2017; 11(3): 034505. <https://doi.org/10.1115/1.4036581>
- [2] Atrial fibrillation. (2020, September 08). Retrieved from [https://www.cdc.gov/heartdisease/atrial\\_fibrillation.htm](https://www.cdc.gov/heartdisease/atrial_fibrillation.htm)
- [3] J. Klespitz and L. Kovács, "Peristaltic pumps — A review on working and control possibilities," 2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2014, pp. 191-194, doi: 10.1109/SAMI.2014.6822404.
- [4] McDowell, B. (2008, March 1). Peristaltic or Diaphragm Metering Pump. Retrieved from <https://www.watertechnonline.com/home/article/14171450/peristaltic-or-diaphragm-metering-pump>
- [5] How to calculate the flow rate of LEADFLUID peristaltic pump. (2020, March 17). Retrieved from <https://leadfluid.com/how-to-calculate-the-flow-rate-of-leadfluid-peristaltic-pump/>
- [6] Elastrat. (2020). Venous Transeptal model for Atrial Fibrillation (AF). Retrieved from [https://www.elastrat.com/documents/fichiers/278\\_54bh\\_v-t-s-n-020v-a-s-n-020-ler-atrial-fibrillation-pour-site-internet-a.pdf](https://www.elastrat.com/documents/fichiers/278_54bh_v-t-s-n-020v-a-s-n-020-ler-atrial-fibrillation-pour-site-internet-a.pdf)
- [7] Elastrat. (2020). Heart venous model reference V-T-S-N-001+ V-A-S-N-001+. Retrieved from [https://www.elastrat.com/documents/fichiers/275\\_ryi3\\_v-t-s-n-001v-a-s-n-001pour-site-internet-a.pdf](https://www.elastrat.com/documents/fichiers/275_ryi3_v-t-s-n-001v-a-s-n-001pour-site-internet-a.pdf)

- [8] J. Klespitz and L. Kovács, "Identification and control of peristaltic pumps in hemodialysis machines," 2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI), 2013, pp. 83-87, doi: 10.1109/CINTI.2013.6705169.
- [9] G. Hillerstrom and J. Sternby, "Application of Repetitive Control to a Peristaltic Pump," 1993 American Control Conference, 1993, pp. 136-141, doi: 10.23919/ACC.1993.4792823.
- [10] Ingesson, G., & Sandberg, H. (2013). Speed control of a peristaltic blood pump.
- [11] Gunnar Hillerström, Kirthi Walgama, Repetitive Control Theory and Applications - A Survey, IFAC Proceedings Volumes, Volume 29, Issue 1, 1996, Pages 1446-1451, ISSN 1474-6670, [https://doi.org/10.1016/S1474-6670\(17\)57870-2](https://doi.org/10.1016/S1474-6670(17)57870-2).  
(<https://www.sciencedirect.com/science/article/pii/S1474667017578702>)
- [12] Steendijk, P., Van Dijk, A., Karsdon, J. et al. 128 ELECTRICAL RESISTIVITY OF CARDIAC MUSCLE RELATED TO PERFUSION. *Pediatr Res* 20, 1055 (1986).  
<https://doi.org/10.1203/00006450-198610000-00183>
- [13] Princy, K.G., Joseph, R. and Kartha, C.S. (1998), Studies on conductive silicone rubber compounds. *J. Appl. Polym. Sci.*, 69: 1043-1050. [https://doi.org/10.1002/\(SICI\)1097-4628\(19980801\)69:5<1043::AID-APP24>3.0.CO;2-0](https://doi.org/10.1002/(SICI)1097-4628(19980801)69:5<1043::AID-APP24>3.0.CO;2-0)
- [14] Braunwald E, Brockenbrough EC, Frahm CJ, Ross J Jr. Left atrial and left ventricular pressures in subjects without cardiovascular disease: observations in eighteen patients studied by transeptal left heart catheterization. *Circulation*. 1961 Aug;24:267-9. doi: 10.1161/01.cir.24.2.267. PMID: 17533672.
- [15] Isza, P. (2011). TFilter - free ONLINE FIR filter design. Retrieved from <http://t-filter.engineerjs.com/>



- [16] S. A. Zulkifli, M. N. Hussin and A. S. Saad, "MATLAB-Arduino as a low cost microcontroller for 3 phase inverter," 2014 IEEE Student Conference on Research and Development, 2014, pp. 1-5, doi: 10.1109/SCORED.2014.7072980.
- [17] G. Hillerstrom and J. Sternby, "Application of Repetitive Control to a Peristaltic Pump," 1993 American Control Conference, 1993, pp. 136-141, doi: 10.23919/ACC.1993.4792823.
- [18] Latham, T.W. (1966). Fluid motions in a peristaltic pump.
- [19] El-Abbasi, N., Bergström, J., & Brown, S. (2011). Fluid-Structure Interaction Analysis of a Peristaltic Pump.

## APPENDICES

### Appendix A: Additional Figures



Figure A1. Cylindrical conductive silicone sample for resistivity measurement

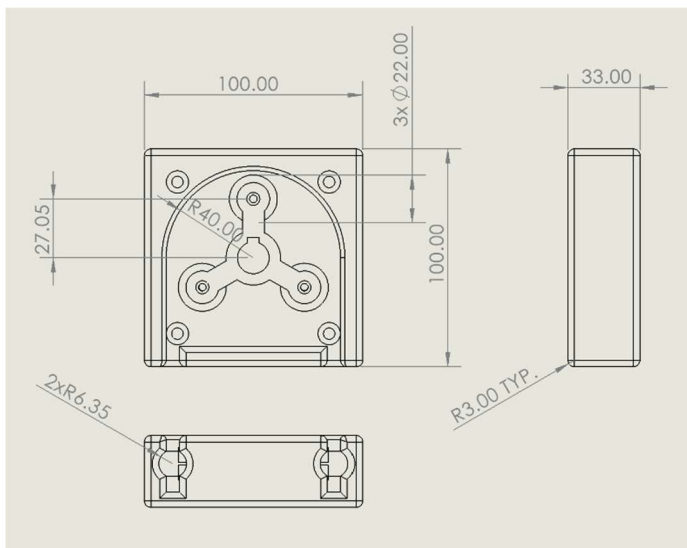


Figure A2. Peristaltic pump dimensions.

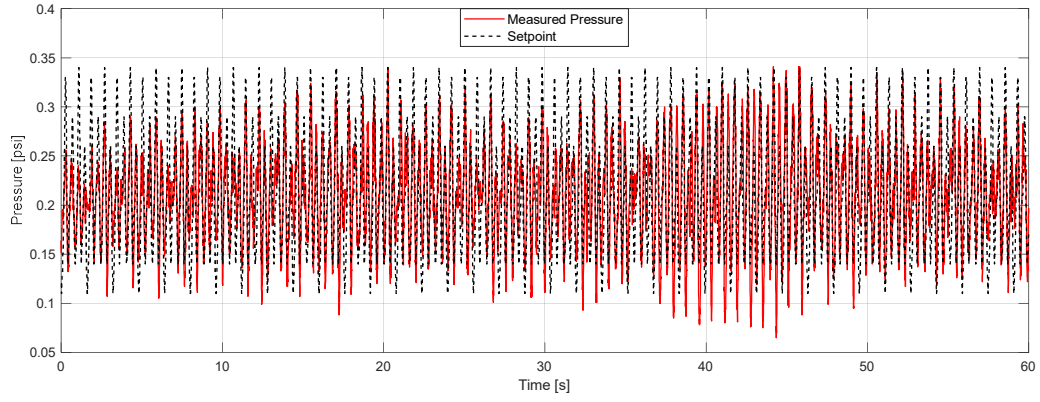


Figure A3. Repetitive controller response.

## Appendix B: Arduino Code

```
#include <AccelStepper.h>
#include <FIR.h>

int stepPin = 2;
int dirPin = 3;
int pressure1Pin = A0;
int stepsPerRev = 51200;
float pressure1 = 0;
float filteredPressure1 = 0;
float Speed = 0;
int i = 0;
float e = 0;
float smoothe = 0;
float eint = 0;
float edot = 0;
float prevPressure1 = 0;
int timeDiff1 = 0;
int lastTime1 = 0;
int timeDiff2 = 0;
int lastTime2 = 0;
float dSpeed = 0;
float lastSpeed = 0;

AccelStepper stepper1(1,stepPin,dirPin);
unsigned long time;
unsigned volatile long milliTime = millis();

float sensorSampleTime = 20; //milliseconds
float pidSampleTime = 5; //milliseconds
```



```

    lastTime1 = now;
}

timeDiff2 = (now-lastTime2);
if(timeDiff2 > sensorSampleTime)
{
    i = i + sensorSampleTime;
    lastTime2 = now;
}

    RunStepper();
}

void RunStepper(){
    stepper1.runSpeed();
}

void PidControl(){

    if(i>1500) i=0;

    pressure1 = analogRead(pressure1Pin)*3.3/pow(2,resolution);
    filteredPressure1 = fir.processReading(pressure1);
    milliTime = millis();
    Serial.print(milliTime/1000.0);
    Serial.print("\t");
    Serial.print(pressure1);
    Serial.print("\t");
    Serial.print(setpoint[i]);
    Serial.print("\t");
    Serial.println(pressure1 - setpoint[i]);

    e = setpoint[i] - pressure1;

```

```
smoothe = setpoint[i] - filteredPressure1;

eint = eint + e;

edot = (pressure1 - prevPressure1)/pidSampleTime;
prevPressure1 = pressure1;
Speed = setpoint[i]*(kp*e + ki*eint*pidSampleTime + kd*edot);
//Speed = setpoint[i]*(kp*smoothe + ki*eint*pidSampleTime + kd*edot);

dSpeed = Speed - lastSpeed;
if(dSpeed > maxAcceleration) dSpeed = maxAcceleration;
if(dSpeed < -maxAcceleration) dSpeed = -maxAcceleration;
Speed = lastSpeed + dSpeed;

if(Speed < 0) Speed = 0;
stepper1.setSpeed(Speed);
lastSpeed = Speed;
}
```