Discrete-Time Controlled Closure of the Aortic Valve Using Blood Vortices

SUPERVISOR:

Andrew Lewis

AUTHORS:

Ian Hogeboom-Burr
Hollis Holmes

Queen's University
Kingston

2020

Abstract

In this paper, a blood vortex is modelled using the Navier-Stokes equations, and its effects on the motion of the aortic valve are analyzed using a simple drag model. The vorticity of this vortex is then used as a control input to close the aortic valve for specific goal times representing the full range of human heart rates. The controller design is presented sequentially, building from a basic linear state feedback controller to an event-triggered discrete-time linear state feedback controller which uses a polynomial function of valve angle to produce control gains. This work appears to be unique in fluids research literature, as it focuses on the control of fluid flow to move an unactuated body immersed in the flow, whereas much of the existing related literature focuses on either the control of articulated bodies immersed in uncontrolled fluid flow or the control of fluid flow around stationary bodies.

Acknowledgements

The authors would like to acknowledge the outstanding faculty support received throughout their four years of study in the Mathematics and Engineering Program, as well as the supervision of Professor Andrew Lewis. The authors would also like to thank all of those at Queen's University who continued to work during the COVID-19 pandemic; their continued efforts made it possible for this paper to be completed. We also extend our gratitude to all of the healthcare workers, retail employees, truck drivers, and others who provided necessary services to our country during the pandemic.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

To pump blood effectively throughout the body, the human heart maintains a complex network of fluid flow within its four chambers. The aortic valve, which separates the left ventricle from the aorta, is a crucial element of a fully functioning heart. This valve effectively retains blood in the left ventricle until the pumping heart forms a pressure gradient between the left ventricle and the ascending aorta, causing the valve to open [24]. After this, blood flows into the aorta, where it is then dispatched to the rest of the body. When this ejection is complete, the aortic valve must close until the heart is ready to pump again. The closing of this valve is a complex and widely disputed process, but research has shown that the formation of vortices in the left ventricle during pumping plays a role in optimizing the closure of the valve (see: [24; 34]).

In recent years, extensive developments have been made in the field of totally artificial hearts. These are ventricular assistance devices that are designed to fully replace the human heart. Based on the observed links between blood vortices and the closure of the aortic valve in the human heart, this paper will assess the feasibility of a vortex-driven closure mechanism for the aortic valve in an artificial heart. Such mechanisms have been proposed before, most notably in a patent from 1975 [9], but no publicly available research on this topic can be found.

1.2 Problem Definition

The goal is to i) model the closure of the aortic valve under the forces of a blood vortex and ii) control this closure through discrete-time modification of the vortex parameters.

The aortic valve is composed of three, equally-sized leaflets which move independently. Only one leaflet of the valve is considered, with rotational motion about a fixed end. The leaflet is immersed in the blood flow.

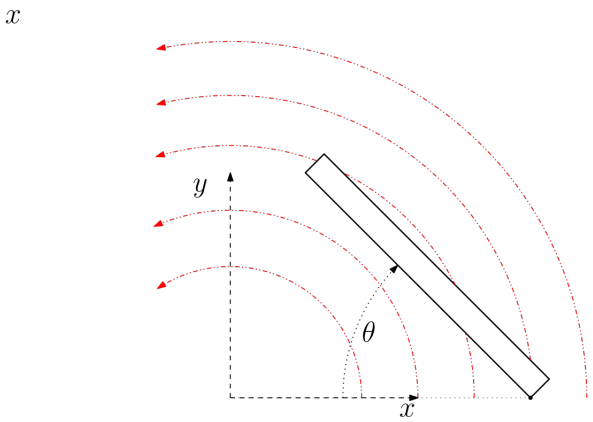


Figure 1: Valve in vortex flow

The valve starts in an upright position $\theta_0 = \pi/2$, and the goal is to steer the valve to $\theta_{T_{\text{goal}}} = 0$ at time $t = T_{\text{goal}}$ through controlling the circulation Γ of the vortex flow. Controlling Γ alters the rotational velocity of the vortex according to the Navier-Stokes equations.

Because real-world control systems operate in discrete-time, the controller must function in discrete-time and be robust to measurement noise in order to be feasible to implement. Computational limitations of on-board hardware must also be considered.

1.3 Impacts of Solution

The outcomes of the project are highly relevant to the design of totally artificial hearts. With confirmation that a controller can achieve success in closing the valve, those designing artificial hearts can begin exploring designs which utilize this method. Closing the aortic valve using blood vortices has benefits over possible mechanical closure mechanisms because i) the vortices will likely occur naturally due to cavity geometry and pressure gradients, as they do in the human heart, so existing energy may be available to be harnessed for this purpose and ii) the vortices will aid in cleansing the left ventricle of calcium buildup and plaque [9]. To fully evaluate the solution from an engineering perspective, a triple bottom line analysis will be performed in Section 6.

1.4 Project Constraints

Due to the nature of the problem, several constraints are imposed on possible project solutions. In many novel controls applications, controller properties are discussed in continuous time. Because the controller in this setting will be working inside an artificial heart, likely using an embedded CPU, the sample rate at which the controller operates will likely be too slow to make the assumption that continuous time solutions will apply. For this reason, this controller will be analyzed in the context of discrete-time control, with the goal being to evaluate the ability of the control system to be controlled in an environment in which computational constraints are imposed. Further constraints imposed by the physical system are discussed in Section 4.1.

2 Background and Literature Review

In order to both approach and contextualize the research presented in this paper, a variety of sources from academic literature were consulted. The problem that will be tackled involves modelling the human heart, controlling fluid flow, and controlling a system in discrete-time. Resources on each of these topics were explored and used to inspire and inform the model and controller design.

2.1 Fluid Dynamics in Left Ventricle

Because of the importance of the aortic valve and left ventricle in the heart, much research has been done on the flow of fluid through these components, both through simulations and through imaging of blood flow in actual patients. Research in this field has lead to the successful development of ventricular assistance devices and existing total artificial hearts. Existing research has evaluated the vortices present within the left ventricle, as found by [28] and [10]. Further, in [41] the fluid flow in the left ventricle was found to affect the motion of the the aortic valve. The work of Ming and Zhen–Huang in 1986 [34] showed that the vortices formed in the quasi-steady phase of blood ejection controlled the motion of the aortic valve. The combination of these works substantiated the project proposal and problem definition by showing that blood vortices may be used to optimize aortic valve closure.

The team was careful to review existing models when making design decisions. In the work of Wong et al., MRI images of the left atrium were used to develop a statistical analysis of the vorticity distribution during distinct cardiac events [25]. A visualization of the velocity field from this model can be found in Figure 2. Wong et al. determined the most effective tool to model vortices present in the heart was the Lamb-Oseen vortex.

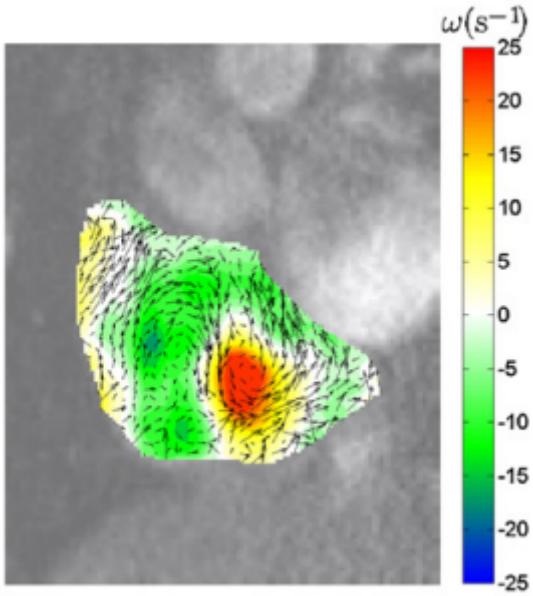


Figure 2: Lamb-Oseen vortices used for modelling blood flow in the heart [25]

2.2 Control of Fluids Systems

A wealth of literature exists on controlling mechanical bodies with dynamic motion in an uncontrolled fluid flow. For instance, see the work of Roenby and Aref, who studied the chaotic motion of bodies in idealized two-dimensional point vortices [41]. Also refer to [46], which provides a constructive method for approaching problems with articulation in fluid flow.

There has also been extensive research on the control of fluid flows around bodies. For instance, the topic of controlling fluid flow around a cylinder with the goal of reducing drag has been explored in much detail (see [8] or [33], among others). In cases like these, mechanisms that actuate using processes such as blowing or sucking around the surface of the cylinder are used to control the Navier-Stokes equations. This is done with the goal of altering the characteristics of the fluid flow, particularly in the wake of the body, which contributes to the drag force. In some cases, the bodies may be rotated as part of the drag reduction processes.

Research on the control of fluid flow around bodies is highly important for the broader problem of valve closure, as we make the assumption that the fluid flow can be perfectly controlled. This assumption would need to be justified by research on the control of fluid flow around the valve in future work. Given the prevalence of blowing/sucking mechanisms in the existing literature, these present a promising method of altering the vortex flow.

These previously explored problems in the literature fundamentally differ from the one that will be explored here, as they do not focus on using the motion of the fluid as the control input to affect the motion of an unactuated body. Research on this topic was not found during the course of this project.

2.3 Discrete-Time and Event-Triggered Control

In applied mathematics, there is much interest in discrete-time and event-triggered control, due to the limitations of real-world control systems. Event-triggered control is a control method in which a controller only actuates at certain times, based on a determination of when actuation is necessary.

Ideally, event-triggered control reduces the quantity of executed control tasks while maintaining the desired closed loop performance [21]. These principles are generally used in networked control systems, where computational loads can become intensive. The team used principles outlined by Heemels, Johansson and Tabuada found in [21] as a basis for the implementation. In their research, a Lyaponov function is used to define the performance of a controller. This performance metric allowed for the times at which the control should be applied to be determined separately from the control calculation. In our paper, this concept is simplified to implement a performance condition in an event-triggered control algorithm.

3 Model of Valve Closure

Prior to implementing a controller, a model had to be developed for the motion of the valve under vortex forces. To do this, several simplifications and assumptions had to be made regarding the behaviour of the fluid vortex, the dimensions of the valve, and the forces affecting the valve's motion.

3.1 Methodology

The goal of this project is to answer the following question: is it feasible to close a valve in an artificial heart using only vortex forces? To attempt to answer this question, we will be exploring a best-case scenario simulation, since, if such a mechanism is not feasible in a best-case scenario, it certainly will not work in reality. Defining this best-case scenario means:

- *Placing the valve directly in the vortex flow.* In the real heart, the interaction of the vortex and valve can be highly complex due to the time-evolving geometry of the heart, but here we will simplify it into the most direct interaction possible, with the fixed end of the valve always sitting 1.4 cm from the vortex core, fully immersing the valve.
- *Choosing valve parameters that offer maximum benefit.* In human hearts, there can be large variations in the properties of the aortic valve; here, the optimal values of those variations will be used to provide maximal force/acceleration on the valve (e.g. minimal weight, maximum surface area, etc.). In an artificial heart, these parameters will be decided upon during the design process, and so the freedom to make these parameter decisions will depend on other design constraints, such as available material, machining capabilities, and interfacing requirements with other components.
- *Allowing the circulation of the vortex to be controlled freely and exactly.* In real hearts, the circulation of the vortex is determined by many factors, such as the pressure gradient entering the aorta, the geometry of the heart, and the heart rate of the patient. Here, we assume that a controller in the artificial heart has full control over this circulation.
- *Ignoring the impact of the valve on the vortex.* In reality, the vortex flow will have complex interactions with the valve, being slowed down upon impact and creating cascading effects on the overall fluid motion in the system; here, we ignore these effects, allowing the vortex to push the valve without being impeded by it. This idealizes the forces being extracted from the vortex on the valve.

This last point will be addressed further, due to the major implications of this assumption. Firstly, due to computational limitations, this project had to be approached strictly using the Navier--Stokes equations, rather than applying Computational Fluid Dynamics (CFD), which uses finite element analysis to predict fluid motion. While the Navier-Stokes equations can be coupled to rigid body dynamics, this process extends beyond the abilities of the members of this project, and would

limit the chance that the problem at hand could be solved with the given time-frame and resources. Even with a Navier-Stokes approach, many papers still complete their simulations through a CFD computation (e.g. [18], and related literature). Therefore, based on these limitations in approach, simplifications had to be made to ensure the problem could be solved given the logistical project constraints.

By analyzing the vortex motion in other studies on the heart, such as is found in [44] and [38], it can be seen that the rotational motion of the blood vortices in the left ventricle persists throughout the valve closure phase. While there are local impacts on the flow at the valve surface, the overall flow region of interest generally displays vortex characteristics. Furthermore, while the effects of the valve on the local vortex flow will result in cumulative effects over the course of the closure, the time interval in which this process occurs is very small (around 0.25 seconds) and the speeds quite slow (less than 0.5 m/s), and so these cascading effects will have less time to accumulate than in more general, unbounded-time scenarios. Lastly, we can observe that the most important interaction between the vortex and the valve will be when the vortex is strongest, which is at the initial time stage. Much of the valve's motion will be determined by the force of the initial impacts with the vortex as it forms, and subsequent effects of local flow abnormalities should be small compared to the force of this impact.

Prior to the valve fully closing, at each time step this situation can be thought of as being similar to a fixed airfoil with a large angle of attack in a fluid flow; this setting has been examined extensively in the literature, and results show that the airfoil has limited upstream effects on the flow [48]. Therefore, the majority of the fluid effects of the flow interaction should occur in the wake of the valve. General wake effects are accounted for as part of the experimentally-derived drag coefficient method which will be used to calculate friction drag, and so they are partially incorporated in the force calculations, although the pressure effects of the wake will not be considered in this project.

Because of these limitations and justifications, the effects on the fluid flow of interacting with the valve are not modelled. The fluid model is left as only evolving as a function of time and position. While this turns out, admittedly, to give a very simple control system, this was not known at the outset of the project, as existing research on this topic was not available. If the authors were to approach this problem again, focusing exclusively on modelling the coupling of the motion between the valve and the vortex would have likely provided more fruitful research, although less meaningful to evaluating the overall problem of valve closure in artificial hearts, which requires controller analysis. Ultimately, the authors were tasked with deciding between making broad simplifications in order to tackle the engineering problem from start to finish, or reducing the simplifications but only focusing on one mathematical facet of the problem statement; in order to provide engineering insight in addition to mathematical exploration, it was decided that the entire problem should be investigated, albeit in a simplified form.

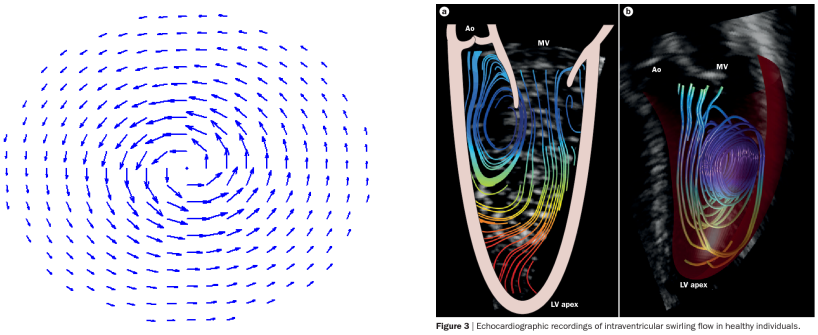
An additional limitation of this method is that, by focusing on this ideal scenario, we will only be able to comment on whether it is generally *possible* to control the valve closure using vortices, and not if it is *probable*. This is because the effects of the assumptions are difficult to quantify in magnitude.

While not possible for this project, an alternative approach to this problem would be to use CFD methods to model the flow, and then attempt to simplify the observed dynamics into a closed-form expression which can be used to develop a controller. Such a method would be superior in terms of model accuracy, but it is unclear whether developing a real-time controller using these methods would be possible, as CFD simulations are extremely computationally expensive and the valve closure is a very fast process.

3.2 Vortex Model

The Lamb-Oseen vortex was chosen to model the blood vortex in the left ventricle. The Lamb--Oseen vortex is a solution to the Navier-Stokes equations that portrays a vortex with purely rotational motion that decays due to viscosity as one moves out from the vortex core. The Lamb-Oseen vortex was chosen to model this scenario due to the resemblance of the Lamb-Oseen flow field to digital imaging of fluid flow in the left ventricle, as well as because blood has a non-negligible viscosity whose effects should be considered. This decision is further supported by the work of [25], who used Lamb-Oseen vortices to model blood motion in the left atrium.

Figure 3 shows a comparison between a vector field plot of a Lamb-Oseen vortex and an echocardiographic image of blood flow in a human heart.



(a) Lamb-Oseen vortex [26] (b) Fluid flow in left ventricle [38]

Figure 3: Selection of Lamb-Oseen vortex

The Lamb-Oseen vortex is defined by the fluid velocities in the flow field, which in polar coordinates are:

$$v_r = 0, \quad v_\theta = \frac{\Gamma}{2\pi r} g(r,t), \quad v_z = 0,$$

where Γ is the circulation of the vortex, and $g(r,t)$ is given by:

$$g(r,t) = 1 - e^{-r^2/4\nu t},$$

where r is radius, t is time, and ν is kinematic viscosity.

In this vortex, only the z-axis has non-zero vorticity (i.e. the spinning of the fluid only occurs around the z-axis), while the pressure field provides the centripetal force necessary to spin the fluid around the vortex core.

For this vortex, it is assumed that the fluid is Newtonian (i.e. the viscosity ν is constant), the flow is incompressible (i.e. the density ρ is constant), and the flow is completely orthogonal to the z-axis.

3.2.1 Verifying Satisfaction of Navier-Stokes Equations by Lamb-Oseen Vortex

For completeness and in order to better understand the fluid behaviour, the Navier-Stokes equations are evaluated using the Lamb-Oseen vortex velocities to determine the pressure distribution in the flow and to ensure the continuity condition is satisfied. While the Lamb-Oseen vortex is not an uncommon solution in modelling vortices, many papers simply refer to the original German paper from 1912 by C.W. Oseen [37], and full, explicit verifications of the results are not readily available

online.

In two-dimensional flow orthogonal to the z-axis, the unforced Navier-Stokes equations in polar coordinates are given by:

$$\frac{\partial v_r}{\partial t} + v_r \frac{\partial v_r}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_r}{\partial \theta} - \frac{v_\theta^2}{r} = -\frac{1}{\rho}\frac{\partial P}{\partial r} + \nu\left(\nabla^2 v_r - \frac{v_r}{r^2} - \frac{2}{r^2}\frac{\partial v_\theta}{\partial \theta}\right) \tag{1}$$

$$\frac{\partial v_\theta}{\partial t} + v_r \frac{\partial v_\theta}{\partial r} + \frac{v_\theta}{r}\frac{\partial v_\theta}{\partial \theta} - \frac{v_r v_\theta}{r} = -\frac{1}{\rho}\frac{1}{r}\frac{\partial P}{\partial \theta} + \nu\left(\nabla^2 v_\theta - \frac{v_\theta}{r^2} + \frac{2}{r^2}\frac{\partial v_r}{\partial \theta}\right) \tag{2}$$

$$\frac{1}{r}\frac{\partial}{\partial r}(r v_r) + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} = 0 \tag{3}$$

In these equations, P is pressure, ρ is density, and ∇^2 denotes the Laplace operator.

The components of vorticity are given by:

$$\omega_r = 0,\ \omega_\theta = 0,\ \omega_z = \frac{1}{r}\frac{\partial}{\partial r}(r v_\theta) - \frac{1}{r}\frac{\partial v_r}{\partial \theta}$$

Here, ω_i is the vorticity of the fluid in the i direction. Vorticity describes the local spinning motion of the fluid near a point.

Equation (1) can be solved using the defining velocities for the Lamb-Oseen vortex:

$$-\frac{v_\theta^2}{r} = \frac{-1}{\rho}\frac{\partial P}{\partial r}$$

This gives the necessary pressure gradient with radius in the Lamb-Oseen vortex: $\frac{\partial P}{\partial r} = \rho\frac{v_\theta^2}{r}$.

Equation (2) reduces to:

$$\begin{aligned}
\frac{\partial v_\theta}{\partial t} &= \frac{-1}{\rho}\frac{1}{r}\frac{\partial P}{\partial \theta} + \nu\left(\frac{\partial^2 v_\theta}{\partial r^2} - \frac{v_\theta}{r^2}\right) \\
\frac{\partial g(r,t)}{\partial t} &= \frac{-2\pi}{\rho\Gamma}\frac{\partial P}{\partial \theta} + \nu\left(\frac{\partial^2 g(r,t)}{\partial r^2} - \frac{1}{r}\frac{\partial g(r,t)}{\partial r}\right) \\
\frac{-r^2}{4\nu t^2}e^{-r^2/4\nu t} &= \frac{-2\pi}{\rho\Gamma}\frac{\partial P}{\partial \theta} + \nu\left(\left(\frac{-2r}{4\nu t}\right)^2 e^{-r^2/4\nu t} - \frac{2}{4\nu t}e^{-r^2/4\nu t} + \frac{2}{4\nu t}e^{-r^2/4\nu t}\right) \\
0 &= \frac{\partial P}{\partial \theta}
\end{aligned}$$

Thus giving the pressure gradient with angle in the Lamb-Oseen vortex as zero.

The third equation, known as the continuity equation, is satisfied. Both partial derivatives are zero, giving $0 = 0$. Therefore, the Lamb-Oseen vortex satisfies the Navier-Stokes equations when the pressure distribution is as derived above.

Applying the vorticity equations, the vorticity about the z-axis becomes:

$$\omega_z(r,t) = \frac{\Gamma}{4\pi\nu t}e^{-r^2/4\nu t}$$

The vorticity vector for the flow is then: $\omega = (0, 0, \frac{\Gamma}{4\pi\nu t}e^{-r^2/4\nu t})$

The last step of this validation is to confirm that Γ, the parameter used in defining v_θ, is in fact the circulation of the vortex. To do this, the relationship between circulation and vorticity can be used. Circulation is the total amount of vorticity orthogonal to a given area. If this area is

taken to be A, defined by the boundaries $r_1, r_2, \theta_1, \theta_2$, and a normal vector $n = (0, 0, 1)$ which is orthogonal to the x-y plane, then the circulation through A, $\bar{\Gamma}$ is:

$$\begin{aligned}
\bar{\Gamma} &= \int \omega \cdot n dA \\
&= \int_{r_1}^{r_2} \int_{\theta_1}^{\theta_2} r\omega_z(r,t) dr d\theta \\
&= \int_{r_1}^{r_2} (\theta_2 - \theta_1) r\omega_z(r,t) dr \\
&= \int_{r_1}^{r_2} \frac{\Gamma r(\theta_2 - \theta_1)}{4\pi\nu t} e^{-r^2/4\nu t} dr \\
&= \Gamma \frac{(\theta_2 - \theta_1)}{2\pi} \int_{r_1}^{r_2} \frac{r}{2\nu t} e^{-r^2/4\nu t} dr \\
&= \Gamma \frac{(\theta_2 - \theta_1)}{2\pi} \left[-e^{-r^2/4\nu t} \right]_{r_1}^{r_2}
\end{aligned}$$

If we expand our area A to encompass the entire vortex flow, with $r_1 \to 0$ and $r_2 \to \infty$, as well as $\theta_1 = 0$, $\theta_2 = 2\pi$, then $\bar{\Gamma} \to \Gamma$. Therefore, the circulation of the Lamb-Oseen vortex is given by Γ, as expected.

Finally, we note that to find the pressure difference between two points of distance r_1 and r_2 from the vortex core, we can use the pressure gradients derived here. To do this, we compute the integral:

$$\begin{aligned}
P_2 - P_1 &= \int_{r_1}^{r_2} \rho \frac{v_\theta^2}{r} \, dr \\
&= \int_{r_1}^{r_2} \rho \frac{\Gamma^2/(4\pi^2 r^2)}{r} (1 - e^{-r^2/4\nu t}) \, dr \\
&= \left[-\rho \frac{\Gamma^2}{8\pi^2 r^2} \right]_{r_1}^{r_2} - \int_{r_1}^{r_2} \rho \frac{\Gamma^2}{4\pi^2 r^3} e^{-r^2/4\nu t} \, dr \\
&= \left[-\rho \frac{\Gamma^2}{8\pi^2 r^2} \right]_{r_1}^{r_2} + \left[\frac{e^{-r^2/4\nu t}}{2r^2} + \frac{\int_{-r^2}^{\infty} e^{-z}/z}{8\nu t} \right]_{r_1}^{r_2}
\end{aligned}$$

3.3 Parameters of Valve and Blood

The aortic valve is a tricuspid valve, meaning it is composed of three flat leaflets (see Figure 4). Each of these leaflets can be individually modelled as a rotating body with a fixed trailing edge. In this project, the tricuspid nature of the valve will be simplified and a single leaflet will be used to model one third of the valve. In the rest of this paper, 'valve' will be used to mean the leaflet of the aortic valve being simulated.

To keep the analysis in a feasible realm given the available resources, the vortex flow in the left ventricle was simplified to be uniform in the z-direction (using the Lamb-Oseen vortex), and so by orienting the leaflet with the rotational flow, the problem simplifies to a two-dimensional one.

One may note that in the case of a tricuspid valve, all of the valve leaflets will line up differently with the vortex flow. For this reason, one might consider using a bicuspid valve in the design of an artificial heart to simplify the problem. One could also design a variety of combined-closure mechanisms that use the vortex forces on the valve leaflet placed most directly in the flow to aid in closing the rest of the valve. Since this project is simply to evaluate the general feasibility of a vortex-closure mechanism, these complications are left as open problems, and a single leaflet oriented perfectly in the flow is evaluated.

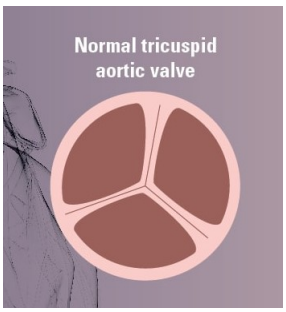


Figure 4: Tricuspid nature of aortic valve [32]

To further simplify the problem, rather than the valve leaflet having a side-face that is a third of a circle, it will be simplified as having a triangular face that varies from a depth of $\frac{2\pi}{3}L$ at its fixed end to a depth of 0 at its far end, where L is the radius of the aortic valve.

Data regarding the physical parameters of the aortic valve was collected, and is summarized in the table below. Data regarding the relevant fluid properties of blood is also included.

Model Parameter	Value	Units	Source
Length of Leaflet	0.012	m	[30]
Thickness of Valve	0.0014	m	[49]
Left Ventricle Diameter	0.052	m	[36]
Valve Weight	1.6	g	[23]
Density of Blood	1060	kg/m^3	[2]
Dynamic Viscosity of Blood	2.4×10^{-6}	m^2/s	[2]

Table 1: Physical parameters of aortic valve and blood

One of the most notable fluid properties of blood is that it is a non-Newtonian fluid [6], meaning the viscosity of blood is a function of the stress acting on it, and as a result blood does not conform to Newton's Law of Viscosity. The Navier-Stokes equations are still valid for non-Newtonian fluids, although the computations become more complex; in this analysis, we do not explicitly account for the non-Newtonian effects of blood, as the Lamb-Oseen vortex assumes constant viscocity.

It should also be mentioned that blood has visco-elastic properties due to the presence of red blood cells, meaning it may store and release energy when acted upon by a force [7]. This property is ignored due to the complexity of these effects.

3.4 Forces on Valve and Motion Update

The two directions of force exerted by a two-dimensional vortex on a body are lift (y direction) and drag (x direction). We will refer to both of these forces as 'drag forces'. The two causes of these forces are:

- Forces caused by the pressure distribution around the valve (pressure drag).
- Forces caused by shear friction between the fluid and the valve (friction drag).

We neglect compressibility in this project, since the maximum flow speeds in this region of the heart are 4.7 ± 0.9 m/s [13]. Likewise, there will be no forces from shock waves to consider.

The forces acting over some surface area on the valve will then be:

$$F_x = \int \left(\int P \cos(\phi) + \tau_W \sin(\phi) dx \right) dz$$

$$F_y = -\int \left(\int P \sin(\phi) + \tau_W \cos(\phi) dy \right) dz$$

Where F is force, P is pressure, ϕ is the angle of action of the fluid on the body, and τ_W is the shear force between the fluid and the body.

The first terms in the force integrals correspond to the pressure forces, while the second terms correspond to the friction forces. Both forces will be evaluated separately. Given the fixed position of the valve's trailing edge and its purely rotational movement, the moment induced by the forces on the valve will be of interest rather than the forces themselves.

3.4.1 Pressure Drag

To calculate the moment exerted by the pressure field on the valve, a numerical integration is performed. Starting at the fixed end of the valve, and moving up in steps of size l, the pressure force on each section of the valve is calculated. This force then produces a moment on the valve. The moment on each section is given by the following equation:

$$M_{\text{pressure}}(d) = (P_{\text{top}}(d) - P_{\text{bottom}}(d))(l * h * d)$$

Where $P_{\text{top}}(d) - P_{\text{bottom}}(d)$ is the pressure difference between the top and bottom points r_T and r_B, which are the distances from the top and bottom of the valve at point d to the vortex core. d is the distance from the fixed end of the valve to the centre of the segment the calculations are being done on. h is the depth of the valve at point d. For further clarification, see Figure 5.

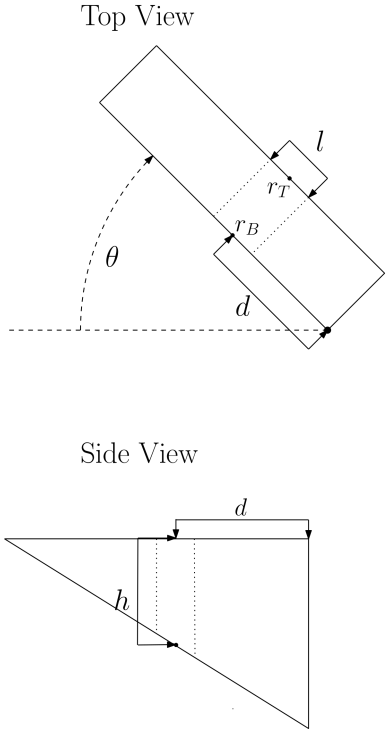


Figure 5: Diagram of valve during numerical integration

When a numerical integration over the length of the valve is completed, summing the moments from each segment gives the total moment on the valve from pressure forces.

3.4.2 Friction Drag

In addition to the pressure drag, the friction drag must also be considered. Since the shear force of the fluid, particularly in non-Newtonian fluids such as blood, can be complicated to calculate, the friction drag will be calculated using drag coefficients rather than from the integral definition. Drag coefficients are experimentally-determined values for various shapes that indicate what the expected friction drag force on the shape will be. This force is predicted by the equation:

$$D = \frac{1}{2} c_{\mathrm{D}} \rho A V^2.$$

Where c_D is the drag coefficient of the body, A is the effective area of the body (generally the front-facing area), and V is the relative velocity of the body to the fluid flow. Note that this equation is not valid for particles moving at very low speeds; in this case, the exponent on V must be 1. In this project, the valve has significant dimensions and will be in flow moving at speeds up to 0.5 m/s, so the standard version of the equation is used.

Each segment of the valve moves in both the x and y directions, in a fluid flow that is also moving in both the x and y directions. Thus, the relative velocity of the body to the fluid must be calculated in both directions, and the drag forces in x and y can then be calculated and converted into moments.

Since the relative velocity of the valve to the fluid varies as one moves along the valve, a numerical integration is completed starting from the fixed end of the valve. Small, approximately rectangular pieces of the valve are considered individually. At each piece, the velocity of the vortex flow in the x and y directions is calculated. In addition, the velocity of the segment in the x and y directions is also calculated using the angular velocity of the valve multiplied by the radial distance from the fixed edge. The relative velocity is then computed.

Given the shape of the valve, each small, approximately rectangular segment has $A = lh$ where l and h are the length and height of the segment, respectively. Given the dimensions of these segments, a drag coefficient of 1.9 is used based on reference tables [35].

For these simulations, the drag force is calculated using the fluid velocities at the bottom face of the valve, since it is the face that is directly interacting with oncoming fluid given the direction of motion for closure. The fluid velocities are essentially the same at both the top and bottom faces due to the small width of the valve.

The drag in the x-direction on a segment of the valve is calculated as:

$$D_{\mathrm{x}} = \frac{1.9}{2} \rho \, (l * h)(V_{\mathrm{x}} - U_{\mathrm{x}})(-V_{\mathrm{x}} + U_{\mathrm{x}})$$

Where V_x is the velocity of the fluid in the x-direction and U_x is the velocity of the valve in the x-direction. The equation is identical for the y-direction. Note that the sign difference between the last two terms is used to account for the direction that the drag force acts in.

The total moment from drag on the segment is then calculated by:

$$M_{\mathrm{friction}}(d) = (-D_{\mathrm{x}} \sin(\theta) - D_{\mathrm{y}} \cos(\theta)) \times d$$

Here the variable θ is oriented as in Figure 1.

3.4.3 Motion Update

To model the rotation of the valve, the total moment about the fixed base resulting from the pressure and friction forces is calculated by $M_{\text{tot}} = M_{\text{pressure}} + M_{\text{friction}}$. The dynamics of the system are then updated using an iterative position-update process applied in discrete-time:

$$\ddot{\theta}(k) = \frac{M_{\text{tot}}}{I}$$
$$\dot{\theta}(k) = \dot{\theta}(k-1) + \ddot{\theta}(k)dt$$
$$\theta(k) = \theta(k-1) + \dot{\theta}(k)dt$$

Where θ is the angle of the airfoil, M_{tot} is the total applied moment, dt is the time step of the simulation, and I is the rotational moment of inertia of the airfoil.

Given the triangular shape of the valve section, the rotational moment of inertia of the valve about its base is $I = \frac{1}{8}mL^2$.

To test that the drag model was functioning as desired, different settings were simulated. To match the theoretical model, the simulations should result in the following properties: if the body is moving in still fluid, the drag should be quadratic with velocity; if the body and the fluid are moving in the same direction with the same speed, the drag should be zero.

First the valve was given an initial angular velocity of 10 rad/s in a stationary flow field. This produced the trajectory shown in Figure 6. The angular position of the valve is quadratic with respect to time, as expected.

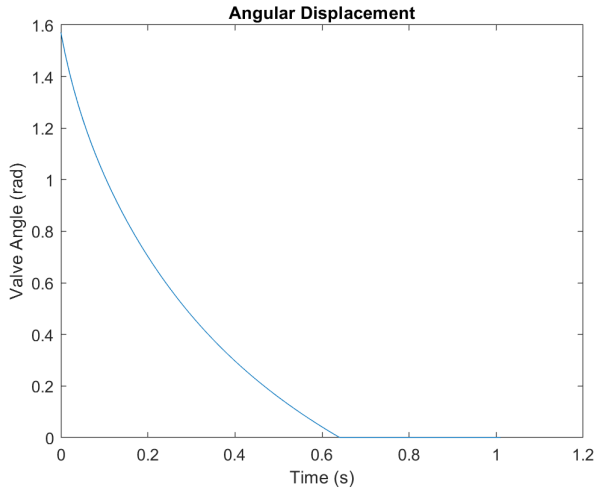


Figure 6: Diagram of valve moving in motionless blood

In the second simulation, shown in Figure 7, the valve was given an initial angular velocity of 10 rad/s in a flow field moving at 10 rad/s 0.01 m from the vortex core (i.e. $\Gamma \approx 0.0628$). In this case, the valve closes along a nearly-linear trajectory in approximately 0.158 seconds. This matches expectations, given the valve and fluid were moving nearly in unison and started at 1.57 radians. The slight nonlinear deviations occur due to fact that the Lamb-Oseen vortex has angular velocity that varies with radius due to viscosity, and so the relative velocity is not zero along the entire body.

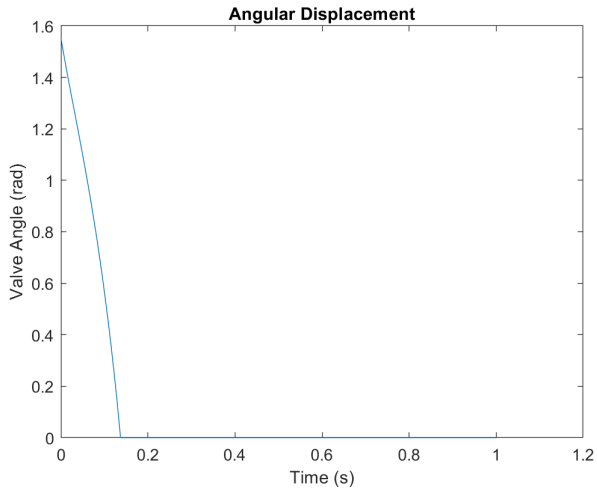


Figure 7: Diagram of valve moving in near-unison with blood

3.5 Model Validation

Designing this simulation model required several simplifications. Furthermore, physical testing scenarios, which are often used for validation in fluids dynamics problems, were not accessible. Because of this, it was crucial to compare the simulation results to real-world results to ensure the model developed here was sufficiently accurate to continue to the control phase of the project.

A previous study done using magnetic resonance imaging on human patients found the average angular velocity around the vortex core in the left ventricle was 30.08 ± 9.98 rad/s during valve closure [28]. In initial simulations using our model with a constant circulation value for the vortex flow, to close the valve in a time frame of 0.25 seconds (which corresponds to a heart rate of approximately 120 beats per minute, representing a heart rate between resting and strained), the maximum angular velocity in the flow field is 13 rad/s. This slight underestimate from our model is likely caused by the idealized construction used in developing the simulation, with the valve sitting directly in the vortex flow and not affecting the vortex through physical interactions. With that said, while the model does not perfectly represent the behaviour of the heart, the values are in the correct order of magnitude.

A study using CFD methods to simulate the vortices during valve closure in the left ventricle found that the velocity vectors in the vortex ranged in magnitude from approximately 0 to 0.4 m/s given a closing time of 0.25 seconds [44]. In our simulation, velocity values given a closing time of 0.25 seconds ranged from 0 to 0.28 m/s. Once again, this slight discrepancy is likely due to the idealized model being simulated.

Overall the simulated model, which was designed to be an idealized scenario, narrowly underestimated the required motion inside the vortex to close the valve. Because of this success, the model was determined to be sufficiently accurate to continue to the control aspect of this design problem.

3.6 Model Assumptions and Potential Failures

This model makes many simplifications so that the problem can be tackled using the available resources and to create the idealized scenario desired for this project. For this reason, the model may deviate significantly from real behaviour in the left ventricle. Potential major causes of inaccuracy are listed here:

- *The Experimentally-Derived Approach to Drag.* The approach to determining the drag force in this simulation relies on the accuracy of the drag coefficient method. While this method is accurate on large scales, it is unclear whether this model breaks down on smaller scales, such as during the numerical integration over the valve. One would expect the model to break down at the scale where individual interactions between fluid molecules become significant. For this reason, a larger numerical integration step size was used to limit these effects, but it is not guaranteed that the magnitude of the drag force is entirely accurate.

- *Neglecting Interaction of Fluid Particles and the Valve.* This approach neglects the effects of the interaction between the fluid and the valve on the vortex flow field. It is assumed that the vortex flow hits the valve and then passes through it immediately. This assumption was made to limit the complexity of the problem, since a full model of each fluid particle's motion would require CFD, as well as to contribute to the idealized scenario being simulated.

- *Location and Orientation of the Valve in the Flow.* It is assumed here that the portion of the tricuspid valve being simulated lies directly in the vortex flow. In a real heart, this is more complicated, with the geometry of the heart playing a greater factor in where fluid interacts with the valve. This assumption was made as part of creating an idealized scenario to test whether it is feasible to utilize vortices to close the valve.

- *Vortex Shape.* It is assumed that the vortex flow can be modelled as a Lamb-Oseen vortex. While this assumption was made based on echocardiographic images, it must be noted that the vortices are not perfectly circular in actual hearts, and so the radial velocity will not simplify to zero. Accounting for this would create a far more complex model, and creating a perfect model of the vortex would require extensive study of solutions to the Navier-Stokes equations and data collection of fluid motion in numerous human hearts. This is beyond the scope of this project, but we refer readers to [25] for an example of how this might be done.

- *Unique Properties of Blood:* As mentioned earlier, blood has visco-elastic properties. These are neglected here due to the complexity of the problem, but these effects may play a role in blood storing and releasing energy when force is applied and relaxed. Furthermore, blood is a non-Newtonian fluid, which may cause further deviations since the Lamb-Oseen vortex does not account for these effects.

3.7 Conclusions on Model

While the model does not perfectly match existing data regarding vortex flow in the left ventricle, it provides results which are similar to reality. The goal of the model was to simulate an idealized scenario to evaluate the overall feasibility of a vortex-closure mechanism for the aortic valve. The model appears to achieve this goal, providing slight underestimates on the required vortex motion to close the aortic valve when compared to existing results in the literature.

The model could be improved in the future by accounting for any of the assumptions listed in Section 3.6, or by designing a physical testing scenario to compare the results against.

4 Implementing a Controller

The first stage of this project involved developing a model for the motion of the aortic valve under the forces of naturally occurring blood vortices. Since one could complete a simulation of this motion more accurately using CFD, and the computational resources required to do this were not available, it was decided that the unique capabilities of this simpler model would be explored.

While a CFD model would be more accurate in its predictive capabilities, the intensity of the simulations makes it very difficult to apply traditional control algorithms in these settings due to

the computational speed at which the simulations occur (see, for instance, the problem explored in [50]). Whereas, while the Navier-Stokes approach provided non-linear time-varying dynamics for our system, these can be linearized and controlled using basic control-theoretic principles.

Additionally, as part of determining the feasibility of a vortex-closure mechanism for an artificial heart, real-world constraints on the controller capabilities should be applied. These constraints should reflect actual limitations on system behaviour, sensor accuracy, and controller ability. The iterative design process for developing such a controller is described here.

4.1 Necessary Criteria for Controller

One of the necessary design criteria for the controller is that the control input Γ should decay to zero as the valve angle approaches zero. This is a constraint of the physical system, since the vortex must come to rest when the valve has closed so that incoming blood from the left atrium can effectively enter the left ventricle.

Secondly, the controller should minimize input perturbations so that the flow in the heart is as stable and predictable as possible. This means the controller should not see large spikes or highly noisy signals, since these will cause major disturbances to the flow and may begin to affect other functions of the left ventricle. The magnitude of these signals should also not result in flow speeds which exceed 4.7 ± 0.9 m/s, since this is the expected maximum speed of blood travelling through the aorta in real patients, and higher blood speeds could result in unforeseen consequences [13].

Lastly, the controller should be able to close the valve over a range of speeds that covers the entire spectrum of heart rates. Given that a patient's heart rate should fall between 60 and 220 beats per minute, this corresponds to valve closure times between 0.1 and 0.5 seconds. A heart rate of 120 beats per minute was chosen as the mean heart rate for evaluation purposes, which corresponds to a mean valve closure time of 0.25 seconds. This was done in part to match the data from [44] for comparative purposes. Furthermore, for any given closure, the closure time should not deviate from the goal time by more than 0.01 seconds; this number is based on natural deviations seen in the heart [1].

4.2 Controller Design

Let the state of the system be $[\theta(t),\ \dot{\theta}(t)]^T$, where θ is the angle of the valve as measured in Figure 1, and $\dot{\theta}$ is its time derivative. The control input for the system will be the circulation Γ of the vortex, since Γ fully determines the behaviour of the Lamb-Oseen vortex flow (outside of fixed constants such as ρ and ν).

The goal of the controller will be to bring the valve from an opened position $\theta_0 = \pi/2$ at $t = 0$ to a closed position $\theta_{T_{\text{goal}}} = 0$ at a given goal time T_{goal}. To do this, two different reference trajectory-generating methods will applied and compared. These methods will generate trajectories for the controller to track, with the goal being to generate trajectories that, when tracked by the controller, result in the necessary criteria from Section 4.1 being satisfied.

4.2.1 Linearizing the System Dynamics

The state space equation for the system can be found by linearizing the system dynamics.

Since the dynamics of $\ddot{\theta}$ with respect to Γ are highly complex and nonlinear, several assumptions

need to be made. These are outlined below:

$$\ddot{\theta}(t) = \frac{M_{\text{tot}}}{I} \qquad \text{(Step 1)}$$

$$\approx \frac{M_{\text{pressure}}}{I} \qquad \text{(Step 2)}$$

$$\approx \left[-\rho \frac{\Gamma^2}{8\pi^2 r^2}\right]_{r=\text{midpoint,bottom}}^{r=\text{midpoint,top}} A_{\text{tot}} \frac{L}{3} I^{-1} \qquad \text{(Step 3)}$$

Step 2 comes from neglecting the forces of friction drag, which were determined through simulations to generally be significantly smaller than the pressure forces. Step 3 comes from neglecting the second term in M_{pressure}, which is typically quite small, and by simplifying the numerical integration of the pressure force into calculating the pressures at the midpoint on the top and bottom surfaces, and then applying these pressures along the entire valve, resulting in a moment which is applied at $L/3$ away from the fixed end of the triangular-faced valve.

This equation can then be linearized in terms of Γ. Doing this yields the following state space representation of the system, linearized about its current position. Note that r is a function of $\theta(t)$.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \left[-\rho \frac{1}{4\pi^2 r^2}\right]_{r=\text{m,b}}^{r=\text{m,t}} A_{\text{tot}} \frac{L}{3} I^{-1} \end{bmatrix} \Gamma$$

4.2.2 Implementing the Control Algorithm

To implement a linear state feedback control algorithm in MATLAB, an error vector is defined:

$$e(t) = [\theta(t) - \theta_d(t),\ \dot{\theta}(t) - \dot{\theta}_d(t)]^T$$

Where θ is the current angle of the valve, and $(\theta_d(t), \dot{\theta}_d(t))$ is a desired trajectory. The linear state feedback controller will then be some control law of the form $\Gamma(t) = K(\theta, t)e(t)$, where $K(\theta, t)$ is a 1×2 gain matrix.

At each time stage, the control algorithm computes the gain matrix K. It does this by linearizing about the current position and time to obtain the A and B matrices from the state space representation, and then using the MATLAB 'place' command to find a gain matrix which results in the poles of the system $A + BK$ being in specified locations in the left-half complex plane. An alternative method to this is discussed in Section 4.6. It should also be noted that in each iteration of the controller design presented in this section, the locations of these poles were re-adjusted in an attempt to optimally tune the controller.

Once K is determined, Γ is computed. This control is then applied as the circulation in the valve closure model, which updates the position of the valve using the dynamics determined in Section 3. The control algorithm is then re-applied until the valve has fully closed.

To prevent the controller from exhibiting undesirable behaviour, a small check is inserted: since the vortex should only rotate in one direction when closing the valve, Γ is prohibited from being negative. If a negative Γ is returned by the controller, it is turned to zero. This prevents the controller from attempting to compensate for pushing too hard by reversing the flow direction.

4.2.3 Tracking a Linear Trajectory

First, the controller was given the task of tracking a linear trajectory. That is, a trajectory in which the valve starts at $t = 0$ in an open position with $\theta_d = \frac{\pi}{2}, \dot{\theta}_d = 0, \ddot{\theta}_d = 0$, and then proceeds with:

$$\theta_d(t) = \begin{cases} \frac{\pi}{2} - \left(\frac{\pi}{2}\right)\left(\frac{t}{T_{\text{goal}}}\right), & 0 \leq t \leq T_{\text{goal}} \\ 0, & t > T_{\text{goal}} \end{cases}$$

In the equation above, T_{goal} is the specified desired closure time for the valve.

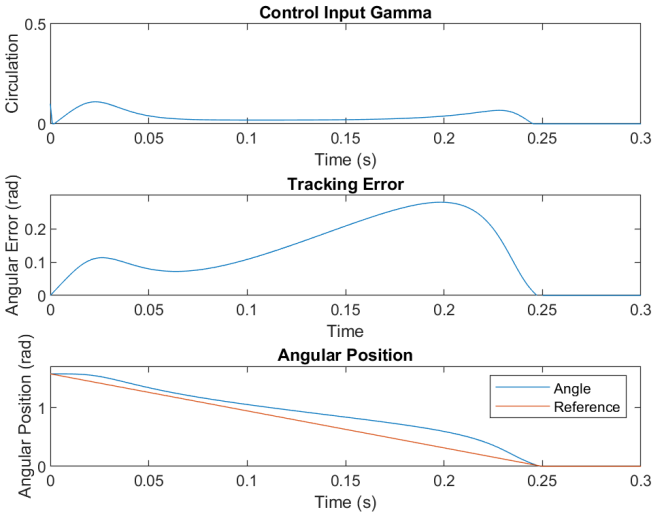


Figure 8: Tracking a linear reference trajectory with goal time of 0.25 seconds

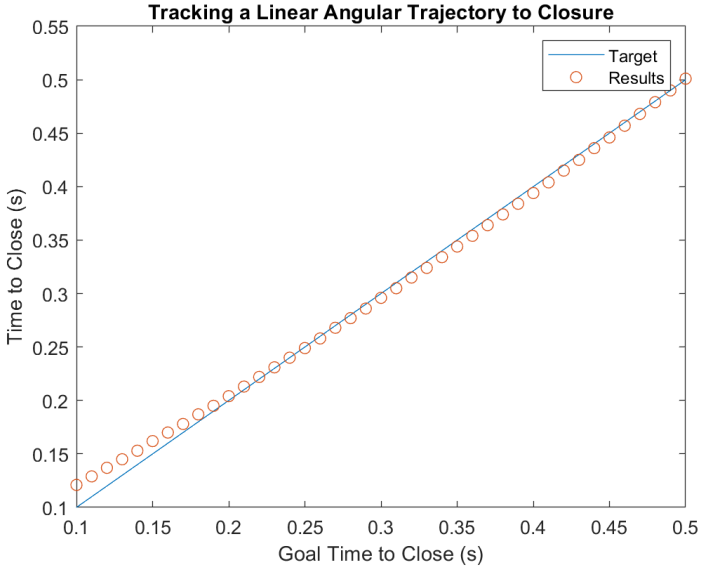


Figure 9: Tracking a linear reference trajectory over all goal times

The pole locations for the control algorithm were tuned to provide optimal response at the mean

goal time $T_{\text{goal}} = 0.25\,s$. The results for the trajectory of $\theta(t)$ for $T_{\text{goal}} = 0.25\,s$ are plotted in Figure 8. The simulation was run using a time step of 0.001 seconds.

In this case, the valve is able to close in 0.249 seconds. The rise in tracking error that occurs around 0.065 seconds is due to the effects of drag (which the controller does not account for) becoming significant as the valve speeds up. The valve then stabilizes near the angular velocity of the vortex, and the controller applies a last-second adjustment to close the valve in time.

The ability for the controller to track linear trajectories was then evaluated over a range of T_{goal} from 0.1 to 0.5 s. This is plotted in Figure 9.

The controller is quite accurate at tracking a range of trajectories, despite being tuned specifically for performance at $T_{\text{goal}} = 0.25\,s$. The root mean squared error for the closure time over this range is 0.0077 seconds, with the maximum deviation between goal time and closure time being 0.021 seconds when the goal time is 0.1 seconds. The controller is least accurate at very fast closure times.

4.2.4 Tracking a Step Trajectory

In search of better performance of the controller and more desirable input behaviour, a step trajectory was also analyzed.

With the step-trajectory, the error is defined as $e(t) = [\theta(t) - \theta_d(t),\ 0]^T$. Defining the error in this manner gives the added advantage that the controller does not need to observe the angular velocity of the valve to function, and is done since the reference trajectory has a constant angular velocity of zero.

A variety of pole locations were tested to determine the fastest the valve could close while exhibiting stable input behaviour. It was found experimentally that this limit was 0.144 s.

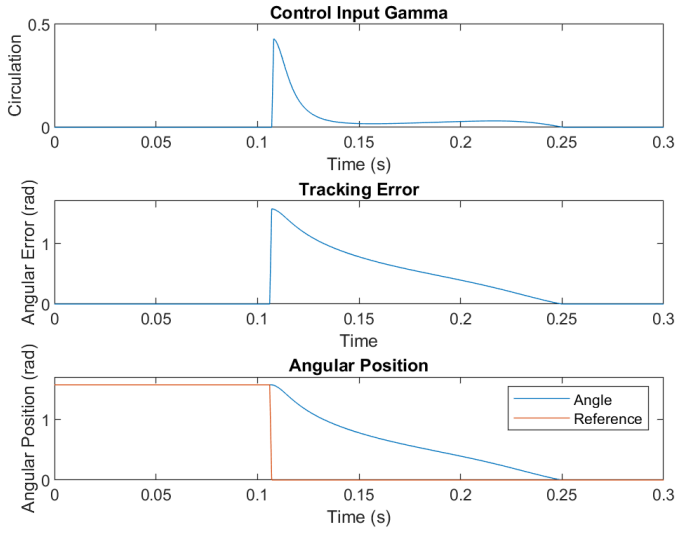


Figure 10: Tracking a step trajectory with goal time of 0.25 seconds

Given this limit, a step trajectory is generated using the following equation, given some desired

closing time T_{goal}:

$$\theta_d(t) = \begin{cases} \frac{\pi}{2}, & 0 \leq t < T_{\text{goal}} - 0.144, \\ 0, & t \geq T_{\text{goal}} - 0.144 \end{cases}$$

I.e. the controller waits until it is 0.144 seconds away from needing to be closed, and then closes in 0.144 seconds. The behaviour is identical across all goal times, and the controller can perfectly track all goal times that are greater than or equal to 0.144 seconds.

Figure 10 shows a visualization of the controller perfectly tracking a goal time of 0.25 seconds.

The closure time results for goal times from 0.1 to 0.5 seconds are plotted in Figure 11.

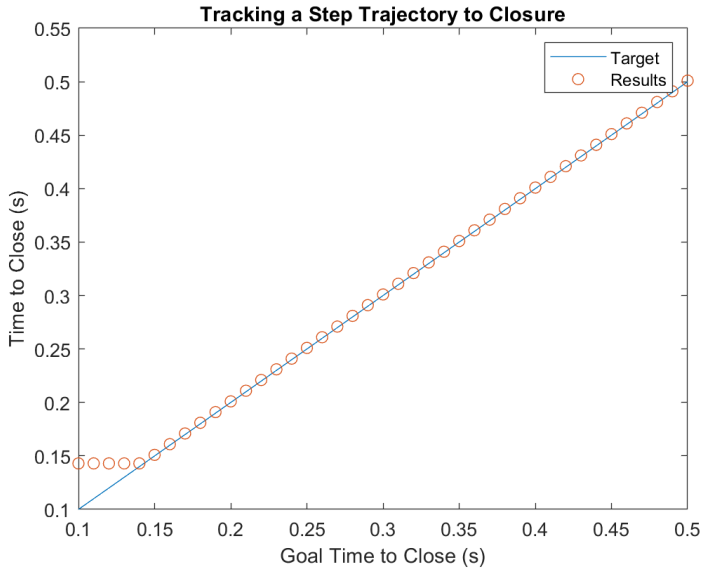


Figure 11: Tracking a step trajectory over all goal times

4.2.5 Comparing Control Methods

There are trade-offs between the two reference trajectory-generating methods described above. To continue with the project, one of these methods must be selected.

The linear trajectory method is characterized by:

- Strong ability to meet all goal times
- Lower magnitude control inputs
- Non-zero circulation near goal time
- Requires angular position and velocity of valve to be known

The step trajectory method is characterized by:

- Perfect ability to meet all goal times greater than 0.144 s
- Higher magnitude control inputs
- Circulation decays steadily to zero near goal time

- Only requires angular position of valve to be known

To select a method, each point must be addressed individually.

The linear trajectory can track all goal times to within a small error. The step trajectory is unable to track the quickest goal times near 0.10 s with good accuracy, but has better accuracy over all goal times greater than 0.144 seconds. A maximum valve closure time of 0.144 seconds corresponds to an approximate maximum heart rate of 208 beats per minute. For any patient over the age of 20, 208 bpm should be higher than the maximum bpm achievable by their natural heart (approximately 200 bpm) [43]. Thus, only at extreme heart rates is the step trajectory worse than the linear trajectory. These heart rates are well outside the scope of care for a general artificial heart patient, therefore the step trajectory is superior in terms of accuracy.

The linear trajectory method uses significantly smaller circulation inputs to close the valve than the step trajectory method. This will require less force to impel, and thus may be easier to implement in an artificial heart.

The step trajectory satisfies the goal of having the control input decay smoothly to zero, while the linear trajectory does not. This makes the step trajectory method more prepared for the incoming blood flow from the left atrium when the valve has fully closed.

To track the linear trajectory, the controller requires knowledge of the angular velocity of the valve. To track the angular trajectory, since error in angular velocity is not incorporated, only an angular position sensor is needed. This means fewer electronic components needs to be incorporated into the artificial heart design.

Therefore, considering the outlined controller goals, the step-trajectory method appears superior, since it is better than the linear method in 3 of the 4 points discussed above.

It should be noted that a variety of different reference trajectory-generating techniques could be tested in an attempt to improve performance. The two methods explored here were selected initially due to their simplicity. Since the step-trajectory method performed near-perfectly, more complex solutions were not explored.

4.3 Adding Measurement Noise

The controller requires knowledge of the angular position of the valve to function. In an artificial heart, this angle can be measured using a variety of sensors, all of which will involve sensor noise.

To capture the effect of using real-world sensors on the controller, normally distributed noise was added to the angular measurement θ. Since the sensor implemented in the artificial heart will need to have a long life with minimal maintenance, will be in constant contact with fluid, and will be small enough to fit on the valve, it may have errors greater than general standards in the industry. For this reason, a standard deviation of ± 0.02 radians was used. For reference, the smallest angular measurement device sold by AMS has a standard deviation of approximately ± 0.002 radians if the sensor has no bias and is used with standard gain settings [3]. The errors added here are chosen to be one order of magnitude larger to reflect a worst case scenario for error, to check if the controller is robust enough to achieve an acceptable degree of accuracy in an extremely noisy setting.

The noise effects were first added to the mean scenario, where $T_{\text{goal}} = 0.25\,s$. In the simulation with noise shown in Figure 12, the controller closes the valve in 0.253 s. Over 100 simulations with a goal time of 0.25 seconds, the greatest observed deviation from the goal time was 0.004 seconds and the root mean square deviation was 0.001 seconds. These are well within the specified controller criteria.

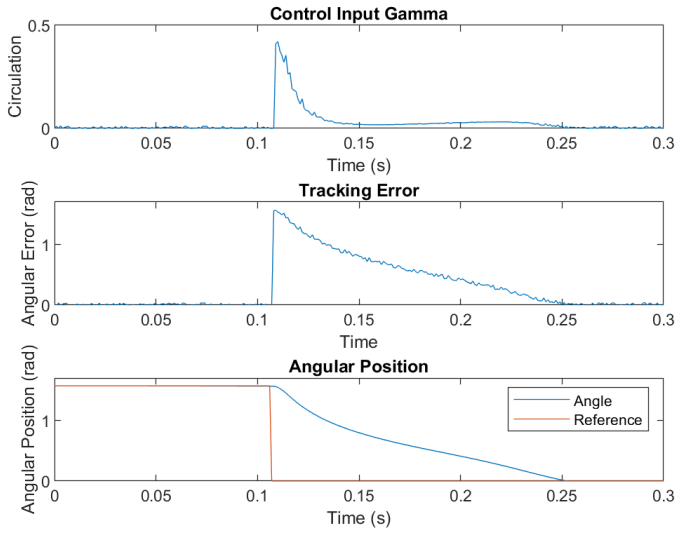


Figure 12: Tracking a step trajectory with measurement noise for $T_{\text{goal}} = 0.25\,s$

Simulations over all goal times from 0.1 to 0.5 are presented in Figure 13.

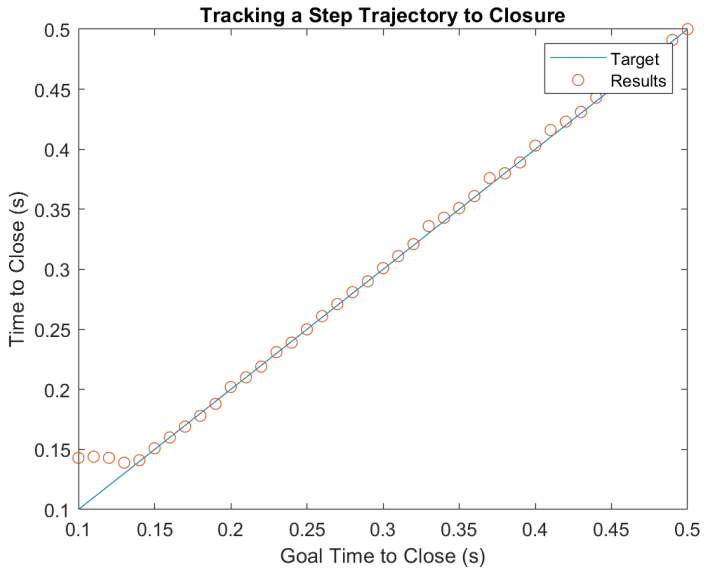


Figure 13: Tracking a step trajectory with measurement noise for all goal times

In the round of simulations pictured in Figure 13, the greatest deviation from the goal time (for goal times of 0.15 s and above) was 0.006 s, which occurred at $T_{\text{goal}} = 0.30\,s$.

Therefore, the controller appears sufficiently robust to noise given the controller criteria.

4.4 Implementing Controller Sample Rate

Another effect of implementing controllers in the real-world is that they can not function in continuous time. Controllers will actuate based on a sample rate, only updating control actions in discrete time.

Because this system will be implemented using small electronic components, and the artificial heart should function without external hardware, the control computations will need to be done on an on-board computer. For this reason, the sample rate may be quite slow, and the slowest sample rate that results in acceptable performance will be evaluated here to provide a lower limit for necessary computational speed.

Holding the goal time steady at 0.25 s, a range of sample frequencies were tested, from 1000 Hz (the simulation frequency) to 1 Hz, with no measurement noise in the system. The closure times are plotted in Figure 14. This figure gives a preliminary indication that the controller does not experience major deviations from the goal time at a sample rate of 10 Hz.

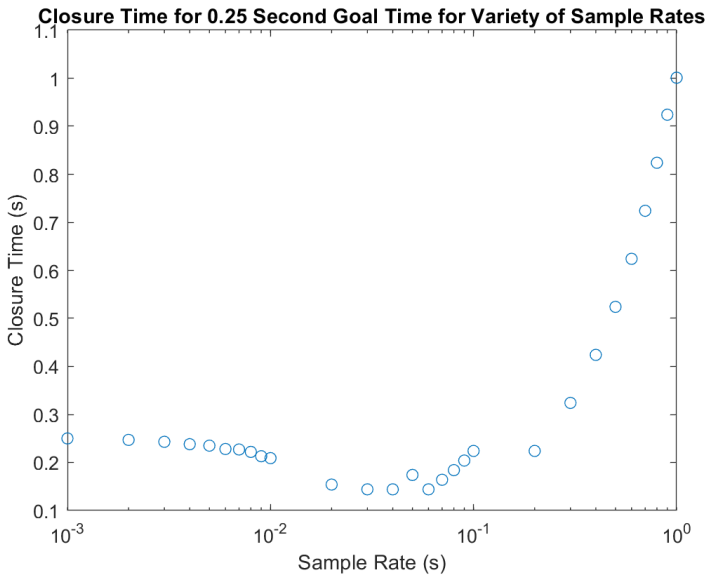


Figure 14: Tracking a goal time of 0.25 s with a sampled controller at various frequencies

In Figure 15, the full results for a simulation with a sample rate of 10 Hz is plotted.

By visualizing the full trajectory, one can see the problems that arise with sample frequencies as slow as 10 Hz. The controller only applies one very strong control, which it applies for 0.1 seconds. This results in the vortex lasting 0.05 seconds longer than desired, despite the valve fully closing, which could have many unintended consequences on the functioning of the heart. The input also does not decay smoothly to zero as the valve closes. Therefore the behaviour of the controller is unacceptable at a sample frequency this slow.

Now, consider a sample rate of 100 Hz. The results for a simulation for $T_{\text{goal}} = 0.25\,s$ are shown in Figure 16. In this simulation, the controller closes the valve in 0.209 seconds.

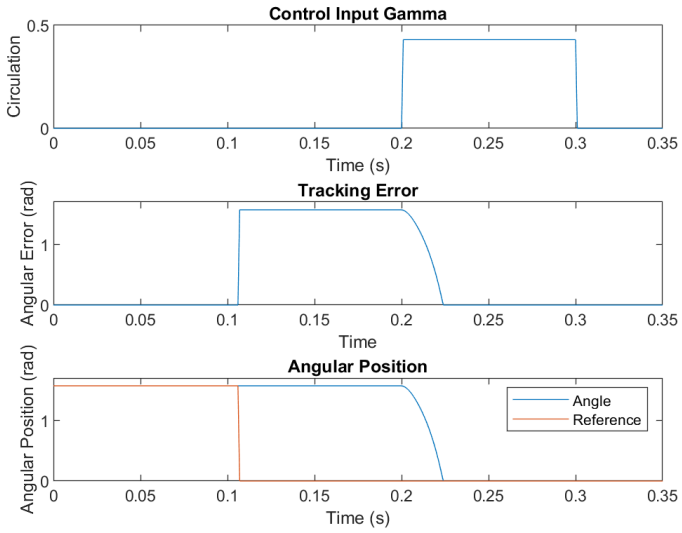


Figure 15: Tracking a goal time of 0.25 s with a sampled controller at 10 Hz

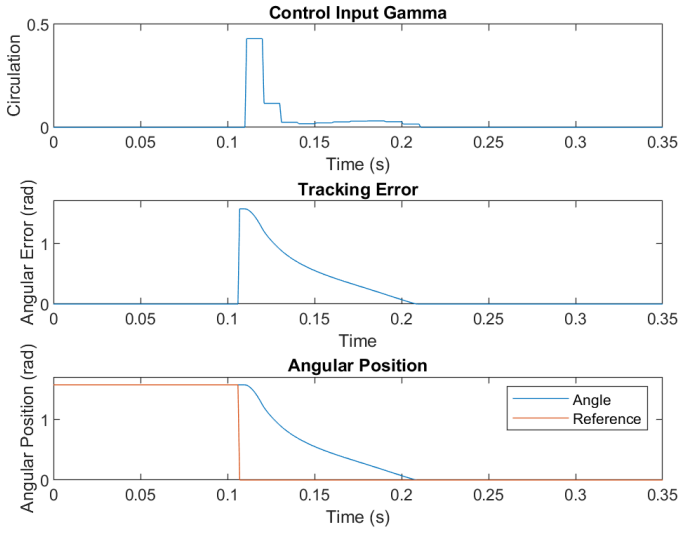


Figure 16: Tracking a goal time of 0.25 s with a sampled controller at 100 Hz

By adjusting the reference step trajectory-generating algorithm for the 100 Hz sample rate, the controller can be designed to close the valve perfectly over all goal times. With a sample rate of 100 Hz, the step trajectory-generating algorithm from Section 4.2.3 only needs to be adjusted to be offset by 0.109 s, rather than 0.144 s.

When this adjustment to the trajectory is completed, the controller is able to perfectly track all goal times in a noiseless environment, as is shown in Figure 17.

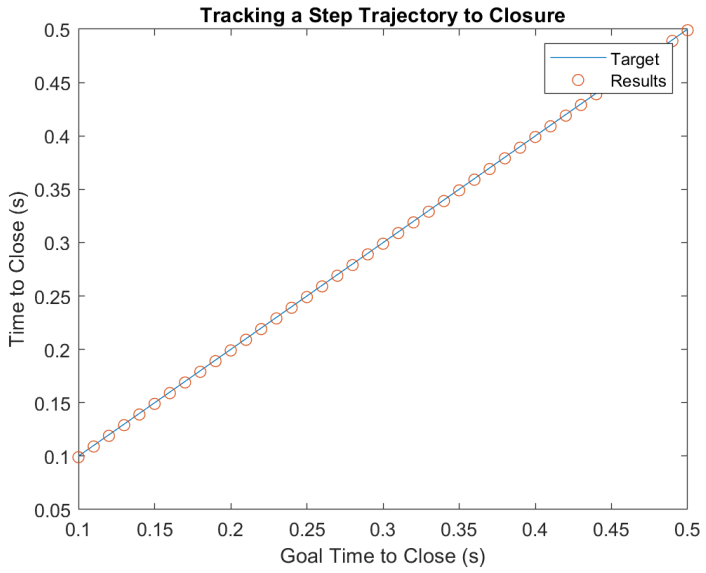


Figure 17: Tracking a step trajectory with a sampled controller at 100 Hz for all goal times

When noise is incorporated into this simulation, the deviations become more severe than in Section 2.3. One example simulation over all goal times is shown in Figure 18.

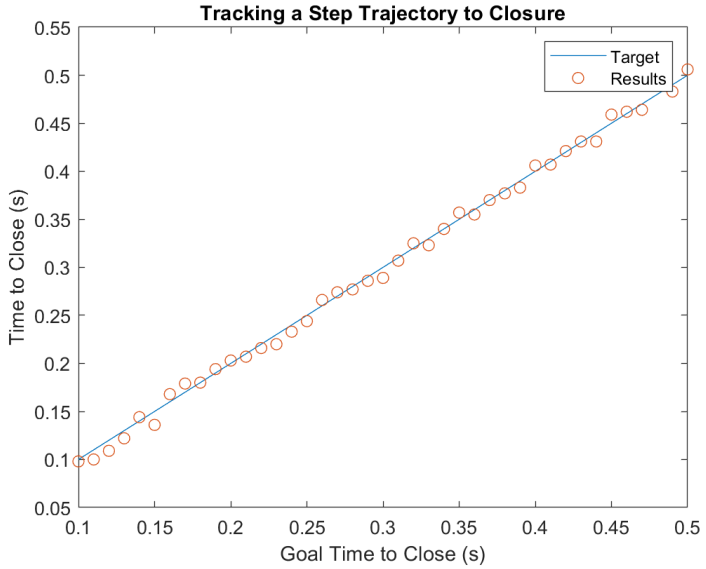


Figure 18: Tracking a step trajectory with a sampled controller at 100 Hz for all goal times with noisy measurements

Therefore, while the sampled controller initially improved on the original controller by broadening the viable goal times down to 0.1 seconds, the sampled controller is significantly less robust to noise. The maximum deviation from the goal time in this round of simulations over all goal times was 0.014 seconds, over double the maximum deviation over all goal times seen in the original controller.

For a goal time of 0.25 seconds, sample results appear in Figure 19. In the simulation pictured, the valve closes in 0.244 seconds. Over 100 simulations for a goal time of 0.25 seconds, the maximum

deviation from the goal time was 0.014 seconds. The root mean square deviation from the goal time was 0.0019 seconds.

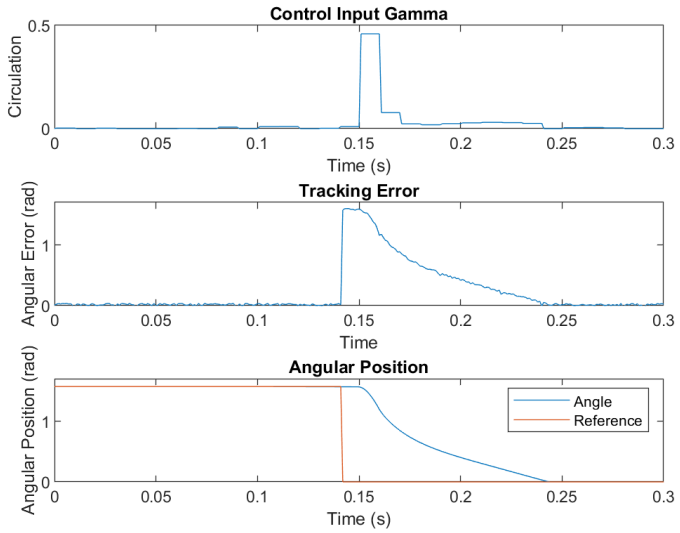


Figure 19: Tracking a step trajectory with a sampled controller at 100 Hz for a 0.25 second goal time with noisy measurements

The sampled controller at 100 Hz results in maximum deviations that are beyond the acceptable limit. While faster sampling would have resulted in lower error, 100 Hz was chosen as an upper limit due to the speed at which the MATLAB controller computation occurs (approximately 55 Hz) and due to possible limitations of communication speed between the sensors, actuator, and controller. These limitations of controller computation are further discussed in Section 4.6.

Therefore, 100 Hz is selected as the sampling frequency, with faster frequencies requiring controller and actuator capabilities which may be unrealistic, and slower frequencies resulting in unacceptable sensitivity to measurement noise and undesirable input behaviour.

4.5 Designing an Event-Triggered Control Algorithm

In reality, we may wish to implement controls less frequently than at each sample time. Because the control can not be applied continuously, it may be advantageous to apply as few unique control actions as possible. This is because the mechanism that actuates the control inputs in an artificial heart will disturb the fluid. Every time the controller adjusts how it pushes the blood, this will cause perturbations, potentially making the real fluid behaviour deviate further and further from the model, which assumes the fluid flows in a perfect vortex. If the controller were to make a new decision at every sample time, particularly given the measurement noise in the system, this could lead to erratic fluid behaviour, making the model become invalid and rendering the controller entirely ineffective. For this reason, a controller which only applies a few unique controls is desirable, since this will result in the model predicting the system more accurately, and thus the controller working more effectively.

To account for this, an event-triggered control algorithm is applied. The algorithm works by running the controller at each sample time. The controller then evaluates whether the control it wants to apply at the current sample time is sufficiently different from the control it applied at the previous sample time, where being sufficiently different is determined by evaluating whether the magnitude of the difference is greater than a preset "sensitivity". In this case, we also implement a

threshold of 0.01 above zero where all desired inputs below this get mapped to zero; this prevents situations in which the controller wishes to turn the actuator off, but the current control is smaller than the sensitivity. This threshold was chosen experimentally based on optimal behaviour.

In Figure 20, the closure time given $T_{\text{goal}} = 0.25$ seconds (and no measurement noise) is plotted for a variety of sensitivities. Note that the simulation window ended at 1 second, and so the tests which failed to converge within 1 second are clustered at this time.

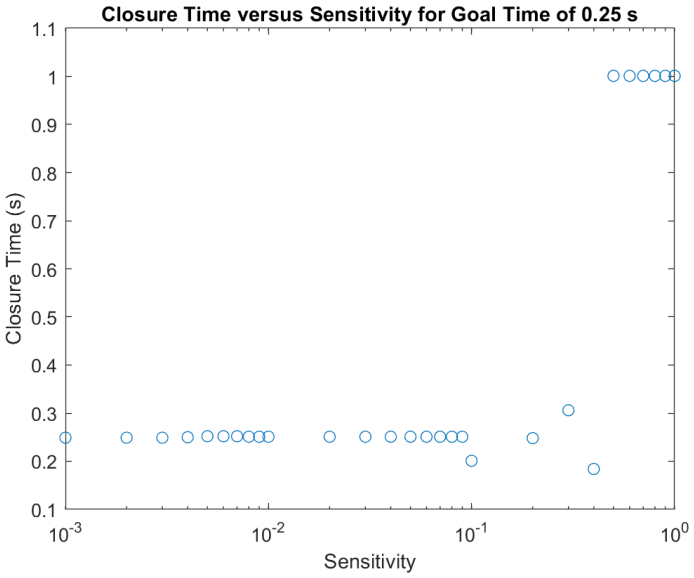


Figure 20: Closure time of valve at a variety of sensitivities for $T_{\text{goal}} = 0.25\,s$ for an event-triggered controller

Figure 21 shows the number of times the controller changes the input for a variety of sensitivities. In these simulations, at a sensitivity of 0.075 the controller closes the valve in 0.251 seconds and only applies 3 different control actions.

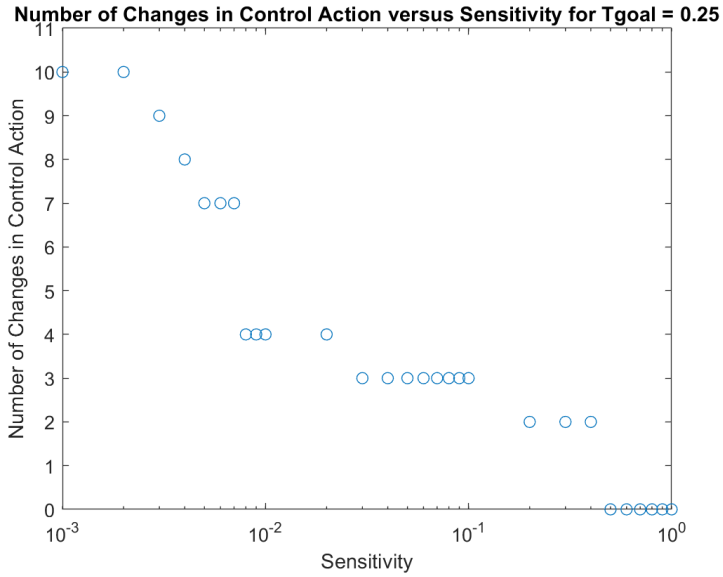


Figure 21: Number of changes in control action for $T_{\text{goal}} = 0.25\,s$ at a variety of sensitivities for an event-triggered controller

The 0.075 sensitivity appears to minimize control changes while maintaining accuracy, and so it was investigated further. When one includes measurement noise, the deviations become more severe, as displayed in Figure 22. In this case, the furthest deviation from the goal time was 0.048 seconds which occurred when the goal time was 0.47 seconds.

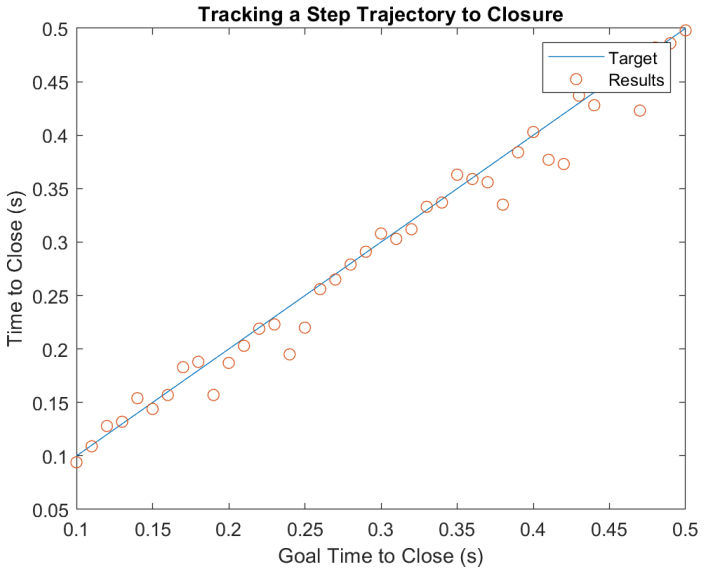


Figure 22: Tracking a step-trajectory with an event-triggered controller with a sensitivity of 0.075 for all goal times

Figure 23 shows a sample result for a goal time of 0.25 seconds. In the pictured simulation, the closure time was 0.245 seconds. Over the course of 100 simulations, the maximum deviation from the goal time was 0.043 seconds, the root mean square deviation was 0.0116 seconds, and the controller always applied 3 different controls. These deviations are well beyond the 0.01 second maximum deviation limit. Therefore, while the 0.075 sensitivity appeared promising in contrast to other sensitivities in a noiseless setting, it was too sensitive to noise to meet the controller requirements.

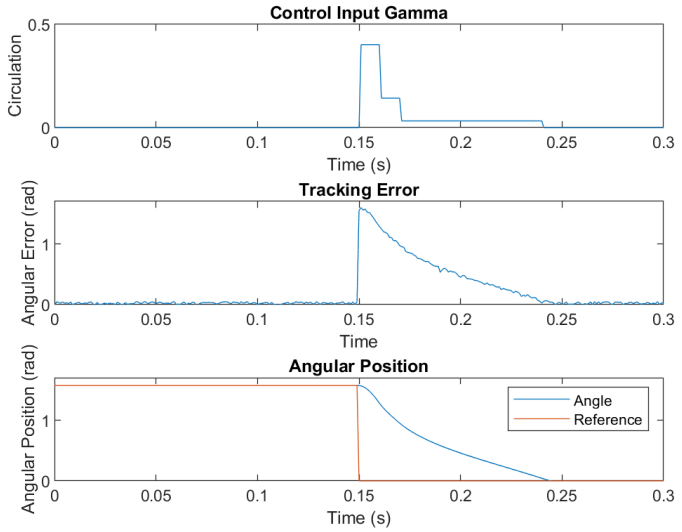


Figure 23: Tracking a step-trajectory with an event-triggered controller with a sensitivity of 0.075 for $T_{\text{goal}} = 0.25\,s$

In contrast, using a finer sensitivity of 0.025 for a goal time of 0.25 seconds gives better results. As with the sampled controller, to achieve these results the reference trajectory generating method was altered, with the offset changing from 0.109 s to 0.101 s. Over 100 simulations, this controller displayed a maximum deviation of 0.012 seconds, with a root mean square deviation of 0.0021 seconds. These deviations are, essentially, identical to those seen in the standard sampled controller. In fact, the maximum deviation is even lower, likely due to the addition of the threshold which eliminates control inputs which are caused strictly by noise. The controller always applies either 3 or 4 different controls, with a mean number of 3.73.

Overall, applying a coarser sensitivity results in the controller being less robust to noise. The trade-off is that a coarser sensitivity controller disturbs the fluid less by applying fewer distinct spinning rates for the vortex. Ideally, a continuous time controller could be applied and the vortex could smoothly decay to zero, but this is not possible in the real world.

To determine which event-triggered control algorithm sensitivity is optimal, one must compare two potential causes of failing to close at T_{goal}: i) failure caused by the model breaking down when the fluid is overly perturbed and ii) failure caused by limited robustness to measurement noise. To quantify the effects of i) is beyond the scope of this project, and would likely require CFD. However, given the analysis performed here, it is recommended that the event-triggered control system be used with a sensitivity no smaller than 0.025. The event triggered controller with this sensitivity is as robust to error as the sampled controller and reduces the mean number of changes in control input from 10 to 3.7; thus, there appears to be no benefit to applying a finer sensitivity than this. To apply a coarser sensitivity would require justification that the benefits of applying fewer control inputs outweighs the deviations in closure times caused by measurement noise.

4.6 Approximating Control Input

Due to the complexity of the operations completed in the MATLAB 'place' command, it is possible that an embedded controller in an artificial heart would be unable to compute the gain matrix quickly enough given the required sample rate of 100 Hz found in Section 4.4. On an Intel Core i5-6300U CPU with 4 GB of RAM, to run one iteration of the controller that uses the 'place' method took MATLAB 0.018185 seconds, which corresponds to a rate of approximately 55 Hz. This is too slow to allow the controller to execute at the sample rate. Therefore, it is necessary to find a faster method to compute the gain matrix.

To achieve this faster speed, a polynomial function will be used to approximate the output of the 'place' command as a function of θ, given fixed pole locations that were used for tuning the controller in Section 4.5.

Since K is a function of θ and t, simulations were computed to determine K given a variety of θ values at constant $t = 0.125$. This is plotted in Figure 24.

Further testing revealed that for the range of times in which the controller acts (0 to 0.5 seconds), K is approximately independent of t.

A regression analysis was then performed to determine how K varied with θ. Note that because of the way error is defined for the step trajectory, only the first entry of K affects the control input, and so it is the only entry that must be evaluated. Let k refer to the first entry in the gain matrix K.

Using the polynomial regression tool in Excel, a degree 6 polynomial was used to fit the curve shown. The equation of this polynomial is:

$$k(\theta) = -0.09\theta^6 + 0.4241\theta^5 - 0.589\theta^4 + 0.1061\theta^3 + 0.6788\theta^2 - 0.7667\theta + 0.2736$$

The residuals for this approximation are plotted in Figure 25.

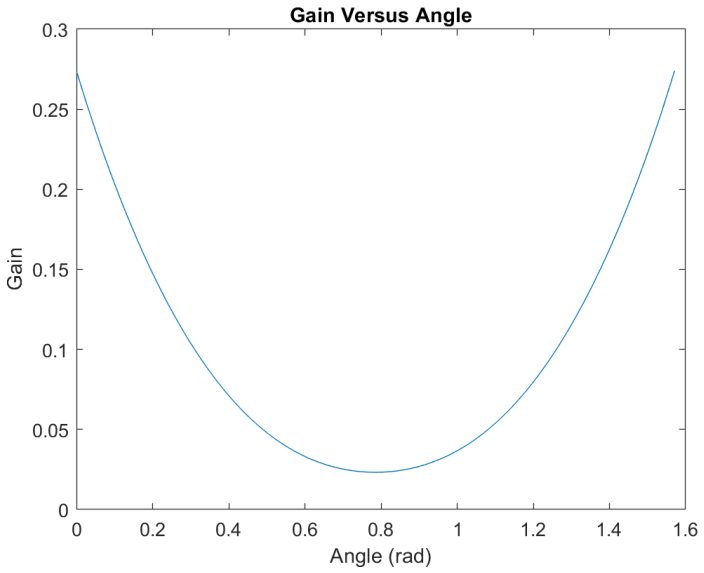


Figure 24: Gain as a function of θ in MATLAB's 'place' command

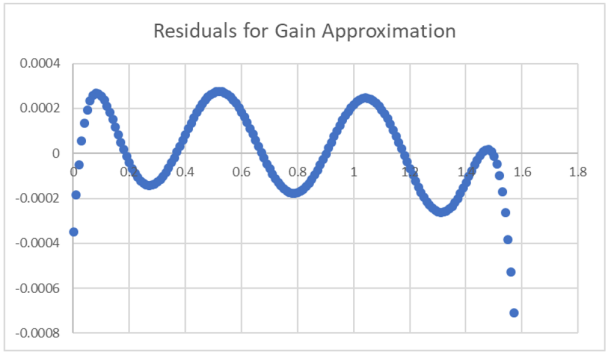


Figure 25: Residuals of polynomial fit for $k(\theta)$

The pattern of the residuals indicates that the data would be better fit by a different function; perhaps a higher order polynomial or a more complicated combination of functions. Lower order polynomial fits exhibited similar residual patterns.

The goal of this approximation is not to discover the true relationship between θ and k inside the 'place' function, but rather to speed up the process of computing k without sacrificing controller performance. Applying this approximating polynomial to generate the gain (with no measurement noise) results in closure times that are identical to those of the 'place' method for the event-triggered controller. There is no detectable change in tracking performance in a noisy setting either. Therefore, while the polynomial is not a perfect representation of the relationship between θ and k, it is sufficiently accurate to not affect controller performance.

To run one iteration of the approximate controller takes, on average, 0.0002 seconds, which is a rate of 5000 Hz. While the embedded CPU in the artificial heart may be slower at computing these values than the CPU used in these simulations, the approximate controller offers enough latitude to have slower computations while still remaining faster than 100 Hz. Given the unknown (and likely significant) limitations of the actuating mechanism and embedded CPU, limiting the sample rate to 100 Hz provides room for additional time constraints without compromising controller feasibility.

The event-triggered controller which uses this approximation method to generate gain will be referred to as the Approximate Event-Triggered Controller (AETC), and is our final recommended design solution for the problem of valve closure.

4.7 A Possible Self-Triggered Control Algorithm

Self-triggered control is an offshoot of event-triggered control which further reduces computational load. In a self-triggered controller, after actuating, the controller *predicts* when it will next need to actuate, and then does not communicate with the sensors or actuate until this time. This is different than the event-triggered controller, which evaluates whether to actuate at each time step. The event-triggered controller removes the need to update actuation at every time step, whereas the self-triggered controller removes the need to both actuate and communicate with the sensors at every time step.

A self-triggered controller can be implemented on this system quite easily due to the repetitive behavioural pattern the controller displays. In Figure 26, the average distribution of actuation times for the AETC over 100 simulations with $T_{\text{goal}} = 0.25\,s$ is displayed.

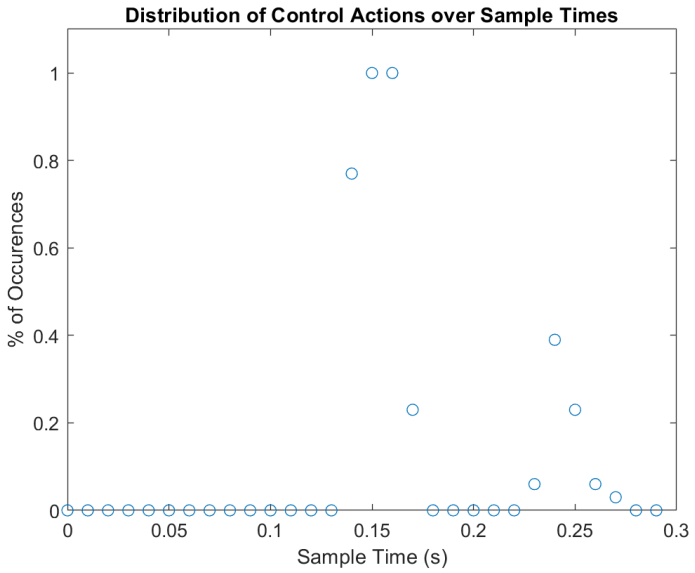


Figure 26: Distribution of control actuation times

There were a total of 377 control actions over the 100 simulations. In 100% of simulations, the controller actuated at 0.15 and 0.16 seconds. In 77% of simulations, the controller actuated between 0.24 and 0.28 seconds. It should be noted that the control actions which occurred after the goal time were all instances where the controller turned off. In 100% of simulations, the control actuated at either 0.14 or 0.17 seconds.

Following this distribution, we can select $t \in \{0.14, 0.15, 0.16, 0.17, 0.23, 0.24, 0.25, 0.26, 0.27\}$ in the $T_{\text{goal}} = 0.25\,s$ case and $t \in \{T_{\text{goal}} + \{-0.11, -0.10, -0.09, -0.08, -0.02, -0.01, 0, 0.01, 0.02\}\}$ in the general case, as the set of times in which the controller should communicate with the sensor. This should result in identical performance to the AETC, with the exception of potential rare deviations in behaviour where the AETC would wish to actuate at times not seen in the 100 simulation test.

Ultimately, this algorithm was chosen to not be recommended for final implementation. This is because the advantages of implementing self-triggered control are not worth the risks of not receiving sensor information. While theoretically this algorithm reduces computational load, this

is a simple system with only one sensor and one controller, and so this load should not be overly intensive to begin with. Furthermore, if an abnormality were to occur in the heart, such as an unexpected blockage or delay in the valve movement, the controller must be able to sense this as soon as possible to react. Reducing controller communication with the sensor opens the pathway for the controller to miss crucial information. With a system as sensitive as the heart, missing this information for several time steps could have fatal effects.

4.8 A Note on the Controllability of the System

Here, a specific control goal has been explored, one which did not require a full analysis of the controllability of the system. To speak more broadly of the control capabilities of the system requires an analysis of controllability. For completeness, we make note of this here.

For this system, the non-linear time-varying dynamics are linearized to create a linear control system which is evaluated at each time step. The controllability matrix of this linear system at a given time t and given valve configuration is always full rank. The controllability matrix is given by:

$$\mathcal{C}(A,B) = \left(-\rho\frac{1}{4\pi^2 r^2}\right)_{r=m,b}^{r=m,t} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Therefore, we have controllability of the linearization of the system for any state and time combination.

It should be noted that our linearization of the system was derived using simplification steps prior to taking the Jacobian. Therefore, it is unclear whether the controllability of this linearization implies local controllability of the full system for all configurations and times, since standard local controllability theorems do not apply.

4.9 Ability to Meet Constraints

The final controller design is the event-triggered controller which uses the approximating polynomial to compute control gain (referred to as the AETC), and which uses the step-trajectory method to generate reference trajectories. Final results for goal tracking for all of the controllers using the step trajectory reference are presented below:

Controller	Max Deviation, $T_{\text{goal}} = 0.25s$	# of Control Inputs	Calculation Speed
Standard	0.004 s	1000+	55 Hz
Sampled	0.014 s	10+	55 Hz
Event-Triggered	0.012 s	≈3.7	55 Hz
AETC	0.012 s	≈3.7	5000 Hz

Table 2: Controller results

To ensure the controller is viable, the velocities in the vortex must remain within the limits of maximum blood velocities in the left ventricle and aorta. Over all simulations using the AETC with measurement noise, the largest circulation was approximately 0.40, corresponding to a maximum blood velocity around the valve of 0.5 m/s. This is well within the limit of $4.7 \pm 0.9\, m/s$.

In the preceding sections, most analysis on controller error was done using the mean goal time of 0.25 seconds. Because the effects of error may differ based on goal time, a full analysis was performed over all goal times to estimate the maximum possible deviation using the AETC. The maximum observed deviation was 0.018 seconds, which occurred for $T_{\text{goal}} = 0.45\, s$.

These maximum observed deviations are beyond the limit of 0.01 seconds set out in the controller design criteria. This is in part caused by that fact that the measurement noise being added to the system was chosen to be an order of magnitude larger than that seen in common industry sensors. In a more moderate noise environment, with measurement standard deviation on the scale of 0.008 radians, which is still larger than that seen in industry, these maximum deviations fall to the acceptable level of 0.01 seconds. Therefore, despite failing in the very high noise setting simulated here, the controller is still robust to measurement noise that is significantly greater than industry standards, and it should be possible to incorporate sensors into the system which allow for acceptable levels of noise.

When it comes to input behaviour, in the final controller design it is not possible for the input signal to smoothly decay to zero because of the discrete nature of the control. It can be observed though that the signal does decay, with the greatest signal magnitude appearing when the controller first actuates, and that signal either remaining constant or dropping in magnitude at each subsequent sample time. This matches the desired input behaviour.

4.10 Conclusions on Controller Design

By taking the simplified dynamics of a valve in a blood vortex, it was found that a controller could be designed which used the circulation of the vortex to effectively close the valve for a range of times corresponding to the range of heart rates seen in a patient.

The final controller narrowly failed to meet the error specifications in a very high noise setting, but showed acceptable performance when measurement noise was reduced to levels closer to those seen in industry sensors. The final controller met the other two design criteria, providing reasonable maximum vortex speeds and using control signals which approached zero as the valve closed.

5 Discussion of Results

The results presented here are those of a very specific problem – closing a heart valve in a vortex in a specific amount of time. Despite this, the conclusions can be discussed more broadly.

At the core of this problem is the question of whether fluid can be manipulated to achieve desired motion for an unactuated object placed in the fluid. This question has two prongs: i) is this possible from a controls perspective, given the dynamics of the system, the nature of the goals, the physical limitations of the controller, and the available modes of control of the motion, and ii) is this possible from a mechanical perspective, to manipulate fluid in a predictable way. Here, we have addressed i). To fully understand this problem requires both questions to be answered.

We make no claim that the results here are applicable to any problem other than the specific one studied. But, the results here do indicate that i) can in fact be possible in limited, simplified fluids environments. The main takeaway of this project is then this: if one assumes the motion of a fluid can be manipulated, then there exist circumstances in which one can achieve desired motion of objects immersed in the fluid using a discrete-time controller which manipulates the Navier-Stokes equations.

6 Engineering Impact of Solution

While the controller discussed in this paper will have minimal direct societal impact, it may contribute to the broader impact on society caused by the development and increased utilization of totally artificial hearts.

6.1 Social

The popularization of artificial hearts will result in many people afflicted with heart disease living longer, more active lives. A study from 1991 focused on the quality of life of patients with mechanical circulation systems, and ultimately predicted that their lives would be extended and their quality of life would improve, so long as there is little societal stigma attached to their new medical situation [22]. To prevent societal stigma attached to medical conditions, we can learn from two classic cases where stigma has fundamentally affected the lives of people with a condition: AIDS and mental health. Research has shown that the societal stigma around AIDS has negatively affected how patients are able to manage the disease [29]. Mental health has had a similar stigmatization in our society [42]. To combat these issues, some previously effective methods have focused on educating the general population, peer support for patients, and awareness campaigns [5; 11]. These lessons can be applied to artificial hearts to ensure patients do not feel stigma associated with their condition and can achieve the highest quality of life possible.

Moreover, vortices formed in the heart provide benefits include softer closing of valve leaflets and a cleansing effect which may help to prevent thrombosis formation [31]. Strokes and peripheral thrombotic events affect 12% and 14% of patients, respectively, within thirty days of receiving an artificial heart [12]. Control of the vortices could help strengthen the cleansing effect, resulting in fewer post-operation complications for patients and fewer return visits to hospitals. This lessens the physical and emotional strain on the patients, as well as the strain on the healthcare system.

6.2 Environmental

Hospitals are major producers of greenhouse gas emissions and massive electricity consumers; in the United States, they are responsible for 10% of greenhouse gas emissions and 9% of criteria air pollutants [15]. Hospitals also consume 4.3% of all commercial electricity despite accounting for less than 2% of all commercial floor space [47]. By limiting the need for extensive in-hospital patient care, artificial hearts would ideally reduce the resource expenditure of hospitals, resulting in lower emissions and a lightened strain on the electrical grid.

Possibly hazardous substances entering the environment must also be considered. Many medical inventions, such as birth control, end up leeching into water supplies and poisoning ecosystems [14]. To ensure this product does not contribute to hazardous medical runoff or waste, the artificial heart should require minimal artificial fluids or hormones to function, since the body may naturally secrete these into water supplies.

Lastly, when a patient dies, the artificial heart will become a biohazard [16]. It will not decompose naturally, and contains electronic components, so it should be removed from the patient. This must be dealt with in a safe way to ensure biological particles do not enter the general environment. Ideally, the product can be directly recycled by being installed in a new patient, much in the same way that biological heart transplants can be done, as this will reduce waste. This will depend on how palatable such a procedure is to prospective patients, since they may be put-off by the idea, and whether the heart can be safely cleaned to ensure there is no host rejection [10].

6.3 Economic

The most important economic consideration for this project is that the final product is financially feasible to implement for patients requiring artificial hearts. This means component cost must be low and the installation procedure must be as quick and simple as possible. Financial feasibility will also be linked to reliability of the design, to ensure the device does not require costly maintenance or replacement [22].

A Syncardia totally artificial heart, one of the leading artificial heart models on the market, costs approximately $166,000 CAD initially, and about $24,000 CAD per year to maintain [4]. It is currently the only total artificial heart to be approved in Canada and the United States [17]. This is clearly very expensive, given that the median household income in Canada is $35,000 CAD [45]. Currently, OHIP, the Ontario provincial healthcare plan, covers all surgeries involving artificial valve replacement [39]. Thus, the most relevant stakeholder for taking on the burden of cost will be the provincial government. Work from Ohio State University has shown the costs of a left-ventricular assistance device insertion and artificial heart transplant procedure to be $686,289 and $881,586, respectively [27]. For Canada as a whole, the current costs directly associated with heart transplants amount to $2.8 billion CAD [20]; if this product can reduce those costs, it will benefit all taxpayers as well as provincial governments.

6.4 Ethics and Equity

Many ethical problems arise from the design of medical devices. Most notably, by designing such a system, the engineer who signs off on the design assumes an ethical responsibility for possible malfunctions resulting in casualties [40]. To ensure that such instances do not occur, trials and tests must be performed in a rigorous clinical setting prior to implementation. Approval from Health Canada is a necessary prerequisite for beginning human trials. The following criteria are used to determine whether such approval is granted [19]:

- Sterilization of the device
- Execution of successful non-human studies
- Proper manufacturing and product control standards
- Evidence of product effectiveness
- Design and manufacturing standards used in the product

For this project, the design explored here would need to be validated by a larger team of experts and through experimental testing prior to being implemented in human studies.

From an equity perspective, the design explored has been chosen to work for a wide range of heart rates, meaning the heart should be viable for implementation in patients of all ages and health statuses. The main equity concern revolves around the economics of the total artificial heart, since the cost may prove a barrier to this potentially life-saving procedure if it is not covered by provincial healthcare providers. This is ultimately a provincial government policy decision, and can only be altered by engineers through lobbying once the product is ready for market.

7 Recommendations

7.1 Recommendations for Artificial Heart

While the initial exploration in this report has suggested that it may be possible to use a controlled vortex mechanism to close the aortic valve in an artificial heart, more work must be done to confirm this. The work from this project could be improved upon and built on in the following ways:

- *Complete a simulation of the full valve in three-dimensions.* The analysis provided here only focused on one third of the tricuspid aortic valve in a two-dimensional flow. Future work should analyze either bicuspid or tricuspid valve designs, which interact with a three-dimensional vortex flow.

- *Utilize the capabilities of CFD to validate and improve the model.* This project focused on using the Navier-Stokes equations to design the controller. While this approach has advantages when designing a controller, the model used here makes many simplifications. In order to improve the model, computational fluid dynamics methods could be used to validate the model, and potentially add correcting factors to account for the effects of valve contact on the vortex flow, the geometric constraints around the valve, and the complex fluid properties of blood.

- *Evaluate alternative valve closure methods.* While it may end up being feasible to close the aortic valve using blood vortices, this may not be the most efficient method of closing the valve. Ultimately, in a device as crucial as an artificial heart, reliability is the most important factor in design. Alternative closure mechanisms should also be explored to ensure artificial hearts are built using the safest design practices available.

- *Evaluate actuation mechanisms.* The work done in this project presupposes that there is an effective method of controlling the circulation of the vortices in an artificial heart. In real human hearts, these vortices form naturally due to the pressure gradients and geometry in the heart. The question of how this circulation can not only be achieved, but can be controlled inside an artificial heart, is crucial to answer. The solution may lie in blowing/sucking mechanisms on the valve surface, and this option should be closely evaluated.

- *Improve accuracy of vortex model.* While the Lamb-Oseen vortex is a reasonable flow solution given the digital imaging available, further improvements could be made by exploring the complex flow in the heart in greater detail, with focused research on solutions to Navier-Stokes equations that serve this purpose.

7.2 Recommendations for Fluids Research

This report presented a controller design for manipulating the Navier-Stokes equations to achieve desired motion of an unactuated body. This research appears unique in the literature, but has taken a very narrow approach due to the specific goals of the problem being tackled. Future research may explore this topic in the following ways:

- *General Relationship Between Navier-Stokes Equations and Body Motion.* This paper only utilized a particular solution to the Navier-Stokes equations, the Lamb-Oseen vortex, to manipulate the body motion. By applying a more general Navier-Stokes equation setup, a wider array of situations can be analyzed.

- *Free Motion of Bodies.* This paper focused on controlling the rotational motion of a body with a fixed edge. Further research could be done on manipulating bodies in $\mathbb{R}^2$ or $\mathbb{R}^3$.

- *Application Areas.* To justify further research on this topic, additional application areas in which fluid may be controlled to manipulate object motion should be explored. Other such examples may include manipulating blood flow in the body to distribute pharmaceuticals or nanomedical devices, altering HVAC systems in buildings to better manipulate motion of suspended particles in air, or changing plumbing infrastructure to better carry solid waste.

8 Conclusion

In this paper, a controller was presented which uses an event-triggered discrete-time control algorithm to close the aortic valve for a variety of heart rates. This work suggests that it may be possible to implement a mechanism in an artificial heart which uses vortex forces to close the valve. Further research on this problem should explore how the control of this fluid flow could be

achieved, as well as how to design more complex models which better simulate the flow behaviour in the left ventricle.

References

[1] S. A. Aase, H. Torp, and A. Støylen, *Aortic valve closure: relation to tissue velocities by Doppler and speckle tracking in normal subjects*, European Journal of Echocardiography, 9 (2008), pp. 555–559.

[2] H. Abdel Baieth, *Physical parameters of blood as a non-newtonian fluid*, International journal of biomedical science, 4 (2008), pp. 323–329.

[3] AMS, *As5050a low power 10-bit magnetic position sensor.* Product Specification Sheet., October 2014.

[4] J. Ashton, *Man goes home with "total artificial heart"*, CBS News, (2010).

[5] Avert, *HIV stigma and discrimination.* https://www.avert.org/professionals/hiv-social-issues/stigma-discrimination., April 2018. Accessed: 03/10/2019.

[6] H. Baieth, *Physical parameters of blood as a non-newtonian fluid*, International Journal of Biomedical Science, 4 (2008), pp. 323–329.

[7] L. Campo-Deaño, R. P. Dullens, D. G. Aarts, F. T. Pinho, and M. S. Oliveira, *Viscoelasticity of blood and viscoelastic blood analogues for use in polydymethylsiloxane in vitro models of the circulatory system*, Biomicrofluidics, 7 (2013).

[8] H. Choi, W.-P. Jeon, and J. Kim, *Control of flow over a bluff body*, Annual Review of Fluid Mechanics, 40 (2008), pp. 113–139.

[9] M. Clune and J. Shanebrook, *Prosthetic heart valve.* Patent: US4011601A, October 1975.

[10] Columbia University Irving Medical Center, *Ventricular assist devices (vads) and mechanical circulatory support.* https://columbiasurgery.org/conditions-and-treatments/ventricular-assist-devices., 2019. Accessed: 03/10/2019.

[11] Committee on the Science of Changing Behavioral Health Social Norms; Board on Behavioral, Cognitive, and Sensory Sciences; Division of Behavioral and Social Sciences and Education; National Academies of Sciences, Engineering, and Medicine, *Ending Discrimination Against People with Mental and Substance Use Disorders: The Evidence for Stigma Change*, National Academies Press, 2016.

[12] J. Cook, K. Shah, M. Quader, R. Cooke, V. Kasirajan, K. K. Rao, M. Smallfield, I. Tchoukina, and D. G. Tang, *The total artificial heart*, Journal Thorasic Disease, 7 (2015), pp. 196–201.

[13] P. Das, H. Rimington, N. Smeeton, and J. Chambers, *Determinants of symptoms and exercise capacity in aortic stenosis: a comparison of resting haemodynamics and valve compliance during dobutamine stress*, European Heart Journal, 24 (2003), pp. 1254–1263.

[14] C. Dean, *Thousands of chemical traces found in drinking water*, New York Times, (2007).

[15] M. Eckelman and J. Sherman, *Environmental impacts of the u.s. health care system and effects on public health*, PLoS One, 11 (2016).

[16] Engineering and Technology History, *Biohazards in engineering*, August 2014. Accessed: 04/06/2020.

[17] Globe Newswire, *New record of 161 syncardia total artificial heart implants set in 2013.* https://www.globenewswire.com/news-release/2014/01/07/600557/10063207/en/New-Record-of-161-SynCardia-Total-Artificial-Heart-Implants-Set-in-2013.html ., 2014. Accessed: 03/10/2019.

[18] I. Hadzic, J. Hennig, M. Peric, and Y. Zing-Kaeding, *Computation of flow-induced motion of floating bodies*, Applied Mathematical Modelling, 29 (2005), pp. 1196–1210.

[19] Health Canada, *Reader's guide to the summary basis of decision - medical devices.* https://https://www.canada.ca/en/health-canada/services/drugs-health-products/drug-products/, 2019. Accessed: 03/10/2019.

[20] Heart and Stroke Foundation, *2016 report on the health of Canadians.* https://www.heartandstroke.ca/-/media/pdf-files/canada/2017-heart-month/heartandstroke-reportonhealth-2016.ashx?la=en ., 2016. Accessed: 03/10/2019.

[21] W. Heemels, K. Johansson, and P. Tabuada, *Chaos in body–vortex interactions*, IEEE, (2013), pp. 3270–3285.

[22] J. Hogness and M. VanAntwerp, *The Artificial Heart: Prototypes, Policies, and Patients*, Institute of Medicine (US) Committee to Evaluate the Artificial Heart Program of the National Heart, Lung, and Blood Institute, 1991.

[23] B. Iung and A. Vahanian, *Valve weight in aortic stenosis: back to the basics*, European Heart Journal, 37 (2015), pp. 700–702.

[24] L. Kadem, R. Rieu, D. Garcia, L. Durand, and P. Pibarot, *Vortex dynamics and aortic valve closure*, vol. Supplement, September 2005, pp. 157–158.

[25] K. Kelvin, J. Wong, R. Kelsoc, S. Worthley, P. Sanders, J. Mazumdar, and D. Abbott, *Cardiac flow component analysis*, Science Direct, 32 (2010), pp. 174–188.

[26] H. Khan, *Characterization of Flow Patterns in MRI Phase Contrast Data*, PhD thesis, November 2011.

[27] A. Kilic, G. Phillips, N. Chimanji, B. Sai-Sudhakar, A. Hasan, R. Higgins, and B. Whitson, *Cost Comparison between Heart Transplantation and Left Ventricular Assist Device Implantation*, Journal of Cardiac Failure, 20 (2014), pp. 212–213.

[28] W. Kim, P. Walker, E. Pedersen, J. Poulson, S. Oyre, K. Houlind, and A. Yoganathan, *Left ventricular blood flow patterns in normal subjects: a quantitative analysis by three-dimensional magnetic resonance velocity mapping.*, Journal of the American College of Cardiology, 26 (1995).

[29] E. Kontomanolis, S. Michalopoulos, G. Gkasdaris, and Z. Fasoulakis, *The social stigma of hiv-aids: society's role*, HIV/AIDS, (Auckland, N.Z.), 9 (2017), pp. 111–118.

[30] P. Lancellotti, *Grading aortic stenosis severity when the flow modifies the gradient valve area correlation*, Cardiovascular Diagnosis and Therapy, 2 (2012), pp. 6–9.

[31] P.-C. Lu, J.-S. Liu, R.-H. Huang, C.-W. Lo, H.-C. Lai, and N. Hwang, *The Closing Behavior of Mechanical Aortic Heart Valve Prostheses*, ASAIO Journal, 50 (2004), pp. 294–300.

[32] Michigan Health, *5 basics to know about bicuspid aortic valve disease.* https://healthblog.uofmhealth.org/heart-health/5-basics-to-know-about-bicuspid-aortic-valve-disease., May 2017. Accessed: 03/10/2019.

[33] C. Min and H. Choi, *Suboptimal feedback control of vortex shedding at low reynolds numbers*, Journal of Fluid Mechanics, 401 (1999), pp. 123–156.

[34] L. Ming and K. Zhen-huang, *Study of the closing mechanism of natural heart valves*, Applied Mathematics and Mechanics, 7 (1986), pp. 955–964.

[35] B. Munson, D. Young, and T. Okiishi, *Fundamentals of Fluid Mechanics*, Wiley, 8th ed., 2007.

[36] K. Narayanan, K. Reinier, C. Teodorescu, A. Uy-Evanado, R. Aleong, H. Chugh, G. A. Nichols, K. Gunson, B. London, J. Jui, and S. S. Chugh, *Left ventricular diameter and risk stratification for sudden cardiac death*, Journal of the American Heart Association, 3 (2014).

[37] C. Oseen, *Uber die wirbelbewegung in einer reibenden flussigkeit*, Arkiv för matematik, astronomi och fysik, 7 (1912), pp. 14–26.

[38] G. Pedrizzetti, G. La Canna, O. Alfieri, and G. Tonti, *The vortex—an early predictor of cardiovascular outcome?*, Nature Reviews Cardiology, 11 (2014), pp. 545–553.

[39] L. Priest, *What heart valve operations are covered.* http://health.sunnybrook.ca/navigator/what-heart-valve-operations-are-covered/., March 2013. Accessed: 03/10/2019.

[40] Professional Engineers Ontario, *Use of the professional engineer's seal.* http://www.peo.on.ca/index.php/ciid/22148/laid/1.htm. Accessed: 03/10/2019.

[41] J. Roenby and H. Aref, *Chaos in body–vortex interactions*, The Royal Society Publishing, 466 (2010), pp. 125–143.

[42] W. Rössler, *The stigma of mental disorders: A millennia-long history of social exclusion and prejudices*, EMBO Rep, 17 (2016), pp. 1250–1253.

[43] M. A. Sarzynski, T. Rankinen, C. P. Earnest, A. S. Leon, D. C. Rao, J. S. Skinner, and C. Bouchard, *Measured Maximal Heart Rates Compared to Commonly Used Age-Based Prediction Equations in the Heritage Family Study*, American Journal of Human Biology, 25 (2013), pp. 695–701.

[44] J. Spuhler, J. Jansson, N. Jansson, and J. Hoffman, *3d fluid-structure interaction simulation of aortic valves using a unified continuum ale fem model*, Frontiers in Physiology, 16 (2018).

[45] Statistics Canada, *Income of individuals by age group, sex and income source, canada, provinces and selected census metropolitan areas.* https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1110023901., 2017. Accessed: 03/10/2019.

[46] G. Turk, P. Mucha, I. Essa, N. Kwatra, C. Wojtan, and M. Carlson, *Fluid simulation with articulated bodies*, IEEE Transactions on Visualization and Computer Graphics, 16 (2010), pp. 70–80.

[47] U.S. Energy Information Administration, *Commercial buildings energy consumption survey.* https://www.eia.gov/consumption/commercial/reports/2007/large-hospital.php., August 2012. Accessed: 03/10/2019.

[48] L. Wang, S. Cao, Y. Li, and J. Zhang, *On synchronization in flow over airfoil with local oscillating flexible surface at high angle of attack using lagrangian coherent structures*, The European Physical Journal on Special Topics, 228 (2019), pp. 1515–1525.

[49] R. Webb, N. Culliford-Semmens, K. Sidhu, and N. Wilson, *Normal echocardiographic mitral and aortic valve thickness in children*, Heart Asia, 9 (2017), pp. 70–75.

[50] W. Zuo and Q. Chen, *Real time or faster-than-real-time simulation of airflow in buildings*, Indoor Air, 19 (2009), pp. 33–34.

9 Appendix: MATLAB Code

For completeness, key portions of the MATLAB code used in simulation are presented here.

9.1 Main Code

Contents


```
clear all
nm = 0;
```

Simulate over a variety of goal times

```
UniqueCount = zeros(20,1);
controlnumbers = zeros(30,100);
for Tgoal = 0.25


    nm = nm+ 1;
```

Physical Parameters of Fluid

```
rho  = 1060; %Density of blood
nu = (2.4*10^(-6)); %viscocity of blood
```

Physical Parameters of Valve

```
x2 = 0.01; % X Location of fixed point of valve
y2 = 0.01; % Y Location of fixed point of valve
length = 0.02; %Length of valve (maximum seen in patients)
mass = 0.0016; %mass of valve
initialAngle = pi/2; %initial angle of valve
thick = 0.001; %Thickness of valve
```

Lamb-Oseen Vortex Parameters and Intializing State

```
Gamma(1) = 0; %initial circulation of Lamb-Oseen vortex
omega = 0; %intial angular velocity
```

```
alpha = 0; %initial angular acceleration
dt = 0.001; %time step for simulation
sample = 0.01; %Controller sample time (100 Hz)
Sensitivity = 0.025; %event-triggered sensitivity
x1 = x2 - length*cos(initialAngle); %location of rotating bottom surface of valve
y1 = y2 + length*sin(initialAngle);
thetaV2 = pi - atan2(y1 - y2,x2 - x1); %Getting angle using 4 quadrant inverse tangent
count = 0;
%This count variable is used in place of time in the motion variables
%since time steps aren't integers
Moment = 0; %Initial moment on valve
Tend = 1; %End of simuation time
```

Define Desired Trajectory

```
%The desired trajectory is a function theta(t)

t = 0:dt:Tgoal;
t2 = Tgoal+dt:dt:2*Tgoal-dt;
path1 = (pi/2)*ones(1,round((Tgoal - 0.144)/(dt)+1)); %ORIGINAL METHOD
path1 = (pi/2)*ones(1,round((Tgoal - 0.109)/(dt)+1)); %SAMPLED METHOD
path1 = (pi/2)*ones(1,round((Tgoal - 0.101)/(dt)+1)); %EVENT METHOD
path2 = zeros(1,round(Tend/dt + 1)+1); %zeros after end
path = [path1 path2];
```

Additional Variables

```
stoptime = 0; %Counting Variable
angle = 0; %Plotting Variable
```

Simulation

```
for t = 0:dt:Tend
```

$\text{count} = \text{count} + 1;$

Controller

```
if mod(t, sample) == 0 %Check if it is a sample time

y_n = thetaV2(count)+ randn(1)*(0.02);
%measurement of angle with additive measurement noiseˆ

K = LinearFeedbackVortex2(t, y_n, x2, y2, 'EventTriggered');
%Call controller to get gain matrixˆ
K = [0 ApproximateK(y_n) ]; %note:backwards from doc
e(count) = y_n - path(count);
%Compute error
```

```
%This statement checks to make sure our error knows that we can't go beyond
%the physical limits of the valve
if count > 1
   if (e(count) - e(count-1)) > pi/2
       e(count) = e(count) - pi/2;

   elseif e(count - 1) - e(count) > pi/2
       e(count) = e(count) + pi/2;
   end
end

input = K*(e(count));
%Controller gives proportional gain based on Tgoal

Gamma(count+1) = input(2);
%Note: this is opposite ordering from in document

%This statement is for standard setups, to prevent our vortex from trying
%to push backwards.
 if Gamma(count+1) < 0
     Gamma(count+1) = 0; %We don't want our vortex to spin backwards
 end

%Check if we're allowed to apply control, if not, reset control to before
if EventTriggeredControl(Gamma(count), input(2), Sensitivity) == 0
    Gamma(count+1) = Gamma(count);
elseif EventTriggeredControl(Gamma(count), input(2), Sensitivity) == 2
    Gamma(count+1) = 0;
else
     UniqueCount(nm) = UniqueCount(nm) + 1;
end

else %if it isn't a sample time, don't update control
    Gamma(count+1) = Gamma(count);
    y_n = thetaV2(count)+ randn(1)*(0.02);
    e(count) = y_n - path(count);
end
```

Motion Update

```
%NOTE x2, y2 is FIXED POINT OF VALVE

%Call motion update to compute new angle and angular velocity

[thetaV2(count+1), omega(count+1)] = ...
MotionUpdateVortex(thetaV2(count), omega(count),t, Gamma(count+1), dt, x2, y2);

%This is to prevent our valve from dipping slightly below the closed point
%between two time steps
if thetaV2(count+1)<=0
    thetaV2(count+1) = 0;
    omega(count+1) = 0;
end
```

```
if thetaV2(count+1)>=pi/2
    thetaV2(count+1) = pi/2;
    omega(count+1) = 0;
end


if thetaV2(count) == 0
   stoptime = stoptime + 1;
 %Count how long we've been waiting since we closed (used to calculate EndTime)
end


%Update positions of the four corners of the valve
x1 = x2 - length*cos(thetaV2(count+1));
y1 = y2 + length*sin(thetaV2(count+1));
x3 = x1 + thick*sin(thetaV2(count+1));
y3 = y1 + thick*cos(thetaV2(count+1));
x4 = x2 + thick*sin(thetaV2(count+1));
y4 = y2 + thick*cos(thetaV2(count+1));



end

controltimes(nm,:) = Gamma;

EndTime(nm) = Tend - (stoptime-1)*dt; %When did the valve close

fprintf('It took %.3f seconds for the valve to close \n', EndTime);
fprintf('The valve wanted to close in %.3f seconds', Tgoal);


It took 0.250 seconds for the valve to close
The valve wanted to close in 0.250 seconds


end
```

9.2 Functions

9.2.1 Updating Motion

```
function [angle, omega] = ...
MotionUpdateVortex(angle, omega, t, input, dt, x2, y2)

rho  = 1060;
nu = (2.4*10^(-6));

thetaV = angle;
length = 0.02; %Length of valve
mass = 0.0016; %mass of valve
thick = 0.001; %Thickness of valve
I = (1/6)*mass*length^2; %Moment of inertia of valve about base
int = 0.001; %Interval to integrate pressures over
```

```
limitx = x2 - length; %limit conditions for when valve is closed
limity = y2; %limit conditions for when valve is closed

limitopenx = x2;
limitopeny = y2 + length;


x1 = x2 - length*cos(angle);
y1 = y2 + length*sin(angle);
x3 = x1 + thick*sin(angle); %Points on upper surface of valve
y3 = y1 + thick*cos(angle);
x4 = x2 + thick*sin(angle);
y4 = y2 + thick*cos(angle);

Moment = 0;

    for step = 0:int:(length-int)
    depth = ((step + int/2)/length)*(2*pi/3)*0.02; %Depth of valve varies
    depth = (2*pi/3)*length - (2*pi/3)*(step + int/2);
    r2 = sqrt((x2 - step*cos(thetaV) - (int*cos(thetaV))/2)^2...
    + (y2 + step*sin(thetaV) + (int*sin(thetaV)/2))^2);
    r4 = sqrt((x4 - step*cos(thetaV) - (int*cos(thetaV)/2))^2...
    + (y4 + step*sin(thetaV) + (int*sin(thetaV)/2))^2);
    F = int*depth*PressureDiff(input, r2, r4, t);
    ubottom = -(input./(2.*pi.*r2)).*vortex(r2,t,nu)*sin(thetaV);
    vbottom = (input./(2.*pi.*r2)).*vortex(r2,t,nu)*cos(thetaV);
    Dbottomx = 1.9*depth*rho*0.5*int*-1*sign(-ubottom ...
    +(step+int/2)*omega*sin(thetaV))*(-ubottom +(step+int/2)*omega*sin(thetaV))^2;
    Dbottomy = 1.9*depth*rho*0.5*int*-1*sign(vbottom...
    +(step+int/2)*omega*cos(thetaV))*(vbottom +(step+int/2)*omega*cos(thetaV))^2;
    Moment = Moment + (-F + Dbottomx*sin(thetaV) + ...
    Dbottomy*cos(thetaV))*(step + (int/2));
    end

alpha = (Moment)/I;
omega = omega + alpha*dt;
angle = angle + omega*dt;
x1 = x2 - length*cos(angle);
y1 = y2 + length*sin(angle);

if angle<0
y1 = limity;
x1 = limitx;
angle = 0;
omega = 0;
elseif angle>pi/2
y1 = limitopeny;
x1 = limitopenx;
angle = pi/2;
omega = 0;
end

if y1 < limity
    y1 = limity;
    x1 = limitx;
    END = 1;
```

```
end


end
```

9.2.2 Calling Controller

```
function [Gain] = LinearFeedbackVortex2(t, angle, x2, y2, type)
rho  = 1060; %Density of blood
nu = (2.4*10^(-6)); %Viscocity of blood
length = 0.02; %Length of valve
mass = 0.0016; %mass of valve
thick = 0.001; %Thickness of valve

x4 = x2 + thick*sin(angle);
y4 = y2 + thick*cos(angle);

I = (1/6)*mass*length^2; %Moment of inertia of valve about base
r2 = sqrt((x2 - 0.5*length*cos(angle))^2 + (y2 - 0.5*length*sin(angle))^2);
%Distance to center point of valve from origin^
r4 = sqrt((x4 - 0.5*length*cos(angle))^2 + (y4 - 0.5*length*sin(angle))^2);
Area = (1/2)*2*(pi/3)*length^2; %Surface area of valve that pressure acts on

%Our state is [theta omega]
%A and B come from the state space representation of the system
A = [0,1;0,0]; %i.e. \dot{\theta} = omega
B = [0; -((Area)*(0.02/3)*(-rho/(4*pi*r2^2) + rho/(4*pi*r4^2)))/I];
%this comes from linearizing
        %a simplified
        %equation relating
        %Gamma to alpha


if strcmp(type, 'EventTriggered') == 0
Gain = place(A, B, [-500000 -550001]);
else
Gain = place(A, B, [-500000 -550001]);
end
```

9.3 Event-Trigger Function

```
function [Control] = EventTriggeredControl(u_past, u_desired, Sensitivity)

if abs(u_desired - u_past) > Sensitivity
    Control = 1;
elseif u_desired < 0.01
    Control = 2;
else Control = 0;
end

end
```