The Development of a Rotational Magnetic Actuator to Measure Platelet Biomechanics

Nakul Sridhar

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science in Mechanical Engineering

University of Washington

2018

Committee:
Nathan J. Sniadecki, Chair
Jae-Hyun Chung
Nathan J. White

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

The Development of a Rotational Actuator to Measure Platelet Biomechanics

Nakul Sridhar

Chair of the Supervisory Committee:
Associate Professor Nathan J. Sniadecki
Mechanical Engineering

Platelets play an important role in the hemostasis process, which prevents blood loss in the event of injury. As platelets active and aggregate in response to vascular injury, they generate forces that are critical for forming a stable clot. Currently, there are few technologies that can isolate and quantify the forces that platelets generate. Recently, a microfluidic device has used rigid blocks and flexible posts fabricated with PDMS to induce platelet activation and measure their forces using the post deflections. By embedding the posts with magnetic microparticles, external forces could be applied on the posts using a permanent magnet. Using a slider system, static measurements of post deflections could be taken, during which a magnetic field was acting in an "on" or "off" state. To be able to take dynamic measurements of the microposts under force in real-time, the permanent magnet movement needed to be motorized. To develop such a system, a stepper motor was used to rotate a torque magnet about its axis such that it switched its magnetic dipole. This produced a dynamic magnetic gradient that could be used to apply an oscillatory force on the microposts. The

rotating magnetic actuator was tested on the microfluidic devices in an air medium and showed that it could apply a stable and repeatable magnetic force on the microposts. Moreover, various inputs could easily be programmed to test a wide variety of mechanical properties. In the future, this device can be used to measure the biomechanical properties of platelets as they aggregate in real-time.

Table of Contents

Page

List of Figures

Figure Number Page

List of Tables

Table Number Page

Dedication

To my family, who have always supported me on this journey.

Chapter 1 – Introduction

Platelets are small cells found throughout the blood stream and they play an important role in the hemostasis process, which prevents blood lose during an injury [1]. As platelets activate in response to damage to blood vessels, they generate forces that are critical to properly form aggregates that then proceed to create a stable clot [2].

Currently, there are few techniques that can isolate and quantify the platelet force generation during the hemostasis process [3]. Recently, microfluidic devices have used been used to study platelet biomechanics during the clot formation process [3], [4]. One of the main benefits of these microfluidic techniques are the micro and nano size of the devices, which allows for more precise control to accurately represent the blood environment. In particular, a microfluidic device containing rigid blocks and flexible post structures has been used to study forces generated by platelet aggregations that are activated by inducing a high shear rate [5]. To achieve further control, magnetic particles have been embedded into the block-and-post structures, allowing for external magnetic forces to be applied on to platelet clot formations [6]. However, this device only provided static measurements where the device was magnetized to an "on" or "off" state. Since the activation and aggregation of platelets is a highly dynamic event, there was a need to design a device where real-time control was possible.

In this report, I introduce a motorized actuator that can apply a real-time magnetic force to drive the motion of magnetic microposts to study the dynamic properties of platelet biomechanics over time. I highlight both the mechanical and controls design process that I undertook to build the actuator, as well as the testing and characterization of the micropost movement output in air. I conclude by introducing future research applications that this device can be used for.

Chapter 2 – Background

2.1 – Hemostasis

When an injury occurs, hemostasis is the critical process that causes blood to coagulate, preventing blood loss. When there is a tear in the blood vessels, platelets initially activate and attach to the damaged endothelial cells to form a soft platelet plug in a process known as primary hemostasis. In the next stage, known as secondary hemostasis, fibrinogen is recruited to the wound area and is converted to fibrin that forms a mesh in and around the platelet plug. This stabilizes and solidifies the clot [1].

2.1.1 – Primary Hemostasis

During normal blood circulation, platelets do not come into any significant contact with the intact endothelial cell surface [1], [7].When injury occurs, platelets in the blood are exposed to the subendothelial cell matrix, which contain adhesive proteins such as von Willibrand factor (VWF) and collagen. Platelets can then bind to these adhesive proteins with receptors that are located on the their surface [8]. Multiple receptors take part in the adhesion process. The glycoprotein receptor GPIbα, part of the GPIb-IX-V complex, is the first to arrest the flow of platelets through the blood, as it binds to VFW that is immobilized in the extracellular matrix [1]. This process can occur in regions with high shear rates (>10,000 s^{-1}) caused by the blood flow as it passes by the immobilized VWF [7], [9].

Once the initial platelets bind to the VWF, they begin to activate. This occurs as a feedback process; as the first platelets activate from binding to GPIbα, they release agonists such as ADP and thromboxane that activate nearby platelets. Platelets then bind to other nearby platelets, causing aggregation via a crosslinking of the αIIbβ3 integrin and fibrinogen binding sites that are exposed

during activation [7]. As they activate, platelets also undergo a shape change, as G-actin monomers polymerize into F-actin filaments [10]. Overall, the primary hemostasis process leads to the formation of a soft platelet plug that forms a temporary seal to the break in the blood vessel wall [1], [7].

2.1.2 – Secondary Hemostasis

Secondary hemostasis is the process of introducing fibrin to strengthen the platelet plug. When vascular injury occurs, blood comes into contact with tissue factor (TF) that is found in extravascular tissue. This activates a set of coagulation factors, known as the extrinsic pathway of the coagulation cascade. At the same time, as blood comes into contact with subendothelial collagen, another set of coagulation factors activate in a process known as the intrinsic pathway of the coagulation cascade [11], [12]. One of the key enzymes that is generated during the coagulation cascade is thrombin [1], [11], [12]. Thrombin plays the important role of cleaving fibrinogen to form fibrin, which then forms a crosslinked mesh that binds to the platelet plug. The presence of fibrin strengthens and stabilizes the clot.

2.2 – The Importance of Biomechanics in Hemostasis

Much of the research done on hemostasis has been focused on the biological aspects and processes that occur [3]. However, the entire clot formation progression contains many essential biomechanical processes [1], [3], [7]. Overall, the role of hemostasis is biomechanical in nature, as clots must be mechanically stable enough to prevent blood in a high-pressure region during normal blood flow from moving to a low-pressure outer environment during injury.

One of the most important biomechanical factors to consider is the shear stress on platelets that is caused by friction from flowing blood [3]. Previously, tools have been developed to measure

the effects of shear on platelets, as well as other factors that affect the mechanical integrity of the coagulation process. The next section will provide an overview of some of the devices and methods that are currently in use.

2.3 – Macroscale Techniques to Measure the Biomechanics of Clot Formation

There are many tools currently in use that study the mechanical properties of clot formation. Some of the most popular devices to measure the mechanical behavior of coagulation are thromboelastograghy(TEG) and rotational thromboelastography(ROTEM), which are used both clinically and in research [3], [13]. Another set of techniques, known as rheometry, were originally developed to measure viscous properties in fluids and have now been adapted for studying the elasticity of clots [3], [14] .

2.3.1 – Thromboelastography

TEG and ROTEM are the primary clinical viscoelastic tests for measuring blood coagualation [13]. To obtain measurements using TEG, a 0.36 mL sample of whole blood mixed with clotting agents is placed a cup with a slender pin inserted [Figure 2.1a]. The cup is slowly rotated 4° 45' at a rate of six times per minute to represent the natural flow of blood. As the clot forms, it induces a torque on to the pin, which causes it to oscillate as well. The motion from the pin can then be transduced to trace an oscillation amplitude over time. ROTEM uses a similar technique, but instead of the cup, the pin is rotated and the motion is tracked optically [3], [15].

As the clot starts to form, the oscillation amplitude tends to increase. As the clot begins to break down, the oscillations lessen and begin to converge. Different parameters from the traced curve are then used to paint an overall picture of the health of the clot [13]. These parameters can give information on the initial rate of clot formation, the elasticity of the clot, and the process of clot

lysis [3]. Furthermore, information can be gleaned on the effect of platelet-fibrin interactions on clots, which is especially relevant in trauma situations [16].

However, due to the macroscale nature of TEG and ROTEM measurements, they can be difficult to interpret and pinpoint specific conditions. Also, the mechanical contribution of individual components, specifically platelets, and the effects of high shear rates remain unclear [17]. Therefore, further testing methods are required to better understand those aspects of the coagulation process.

2.3.2 – Rheometry

One technique is to use devices that can provide high shear forces that better simulate the forces that take place in the blood environment. Cone and plate rheometers are tools that can produce a varying velocity profile to apply high shear forces on platelets [14] [Figure 2.1b]. These methods have provided insights into the effects of shear stress on the elastic modulus of clots. Moreover, they have shown that platelet concentration is positively correlated with the elastic modulus [18]. However, rheometry techniques still only provide a macroscale view of clotting biomechanics, are unable to simulate true blood flow dynamics, and require large amounts of blood for effective testing [4].

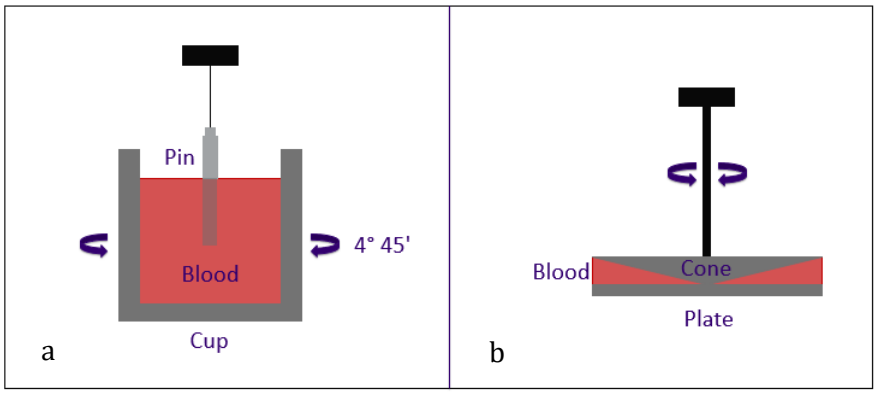

Figure 2.1 – Diagram of common macroscale analysis tools, (a) thromboelastography, (b) cone and plate rheometry

2.4 – Microfluidic Devices

In the last few decades, microfluidic devices have become a popular way to miniaturize flow samples. Experiments can now be designed that match the scale of blood components to better mimic the features that are found in the blood environment [4]. This has the added benefit of requiring much smaller amounts of blood per test. Traditionally, photolithographic techniques used for creating integrated circuits were used to build microfluidic devices. Recently, methods using soft elastomers, such as polydimethylsiloxane (PDMS), have risen in popularity [19]. PDMS can be used to create negative molds of the rigid master molds, which can then be bonded or sealed to other substrates to create flow cells.

Numerous microfluidic devices have been designed to test various aspects of the vascular process and have led to the design of new point-of-care treatment devices. One of the main benefits is that unlike macroscale devices, microfluidic devices can be incorporated with fluorescence microscopy to visually analyze clot formation over time [4]. Moreover, using syringe pumps, flow rates can be easily controlled to create pressure gradients and forces as needed to mimic real blood flow.

2.4.1 – A Microfluidic Device Using Rigid Blocks and Flexible Posts

Previously, a microfluidic device was created to measure the forces generated by platelets after activation as they form a platelet plug [5]. Inside the microchannels, microscale blocks made from PDMS were coated with VWF and collagen. As blood passed over the block, high shear rates were induced, causing the platelets to activate and aggregate onto the block surface as well as a flexible PDMS micropost that was placed behind to the block. As the platelets aggregate, they generate contractile forces that cause the micropost to deflect. The deflection at the tip of the micropost could be visually tracked with a fluorescence microscope. By modeling the micropost as a cantilever beam, the applied force from the platelets could be determined using Equation 2.1,

$$F_{clot} = \frac{E\pi d^4}{8L^3}\delta \qquad (2.1)$$

where F_{clot} is the force generated by the clot, E is the elastic modulus of the flexible post, d is the post diameter, and L is the height of the post. This device could then be used to test forces generated by a clot over time.

2.4.2 – Embedding Magnetic Material into the Block and Post Structures

To run viscoelastic tests on platelet aggregations over time, it was necessary to apply external shear forces on the microposts. A common method of apply external forces to move PDMS structures is to embed them with magnetic material during their fabrication and then actuate them with an external magnet [Figure 2.2]. This method of magnetizing PDMS has been used for many applications, including moving biomimetic cilia [20] and applying forces on cells [21].

Various methods to embed magnetic particles into the block-and-post structures were recently tested [22]. First, nickel nanowires were embedded into the posts, but this produced a low yield where under 40% of posts were upright and responded to an external field. Next, the posts were embedded with iron microparticles that were mixed into the PDMS before curing [6]. This method produced far more upright posts that responded to magnetic fields.

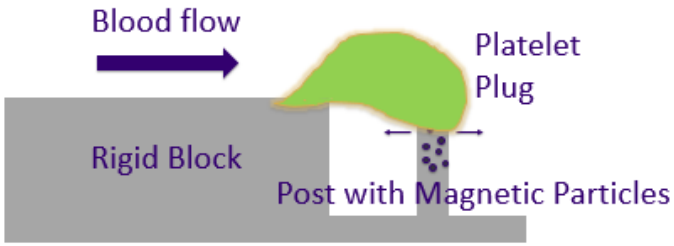

Figure 2.2 – Diagram of platelet plug formation between rigid block and flexible post embedded with magnetic microparticles

To fabricate the magnetic micropost devices, a double replica molding process was used [6]. First, a master mold was fabricated by using deep reactive ion etching to pattern the block and post features on a silicon substrate. Then, PDMS negative molds were fabricated by silanizing the master, pouring uncured 10:1 PDMS mixed with iron microparticles over the mold, and baking at 110 °C for ten minutes. The negative molds were then run through a plasma chamber and silanized. They were then placed on glass slides that held a small drop of PDMS. Inlet and outlet openings were added with a hole puncher. The result was a PDMS device with embedded magnetic particles matching the initial master mold that was ready for experimentation. The resulting device contained posts with a height of 77 microns (μm) and a diameter of 19 μm [Figure 2.3].

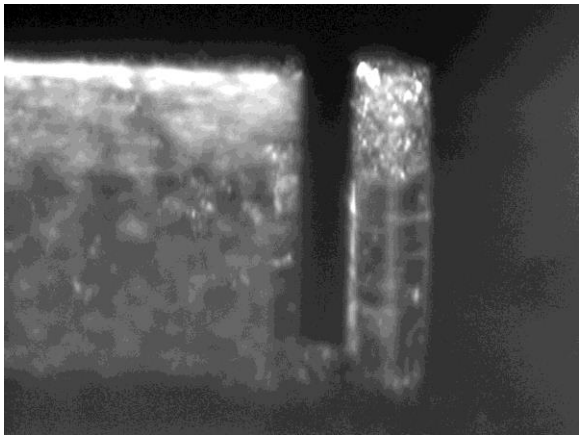

Figure 2.3 – Close up of PDMS block and post with embedded microparticles

2.4.3 – Applying Magnetic Forces on the Microposts

To apply an external magnetic field on the posts, different magnetic techniques were tested [22]. Initially an electromagnetic coil was placed next to the posts, and this could produce fields of up to 25 milliTeslas (mT). To achieve higher fields, a permanent magnet was used. At a close enough distance, fields of over 100 mT were possible. The magnet was placed in a custom-built slide holder

and positioned directly below the microscope. This magnet on a slide system could switch between a static permanent force on the posts and no force.

Using the slider system, the platelet clot elasticity was measured by imaging before and after a magnetic force was applied. Furthermore, the effects of different antagonists on clots could be calculated. However, to measure the dynamic change in platelet elasticity over time as the plug developed, a system with controllable real-time actuation would be required.

Chapter 3 – The Design Process for Building a Real-time Actuation System

3.1 – Project Objective and Overview of the Design Process

To be able to take viscoelastic measurements of platelets under shear in real-time, the external magnetic force needed to be applied dynamically. To achieve this, my aim was to use a stepper motor to drive a permanent magnet such that the magnetic field applied onto the microposts could be controlled by a user input. A major consideration was to design the mechanical system such that it would fit with the microscope stage and the experimental components such that real-time imaging was possible.

There were two major aspects to the design process of the actuation system. One was the mechanical design of the system, where I developed the physical build of the device via prototypes, such that it fit with our constraints and was mechanically stable. The other was the controls design, where I developed the method to control the motor input to the device. Both aspects were developed simultaneously, but here, they are split into two sections to clearly highlight each process.

3.2 – The Mechanical Design Process

During the mechanical design process, the physical build of the actuator was developed via multiple prototypes designed to fit with the space constraints imposed by the microscopes, and to produce a smooth, stable, and repeatable actuation. Each prototype was built with a combination of ordered mechanical parts and custom components designed using SolidWorks and manufactured with an

additive printer (MakerGear M2). In this section, each design iteration will be highlighted to show the evolution of the system that led to the final design.

3.2.1 – Producing a Magnetic Force by Changing the Magnet Distance

Initially, my goal was to change the magnetic force applied onto the microposts by using a motor to vary the distance of a neodymium bar magnet with a north-south dipole [Figure 3.1]. By modeling both the magnet and the microposts as dipoles, the distance between them produces a corresponding magnetic force, given by

$$\vec{F}_{ab} = \vec{\nabla}(\vec{B}_{ab} \cdot \vec{m}_b) \qquad (3.1)$$

where $\vec{B}_{ab}$ is the magnetic field generated by dipole a on dipole b, *and* $\vec{m}_b$ is the magnetic moment of dipole b [23]. As mentioned in Section 2.4.3, a non-motorized magnet slider had been previously designed that could move the bar magnet between an "on" and "off" position and produce a static force on the microposts. To achieve measurements in real-time, I wanted to introduce a motor to drive the slider such that a dynamic force could be produced.

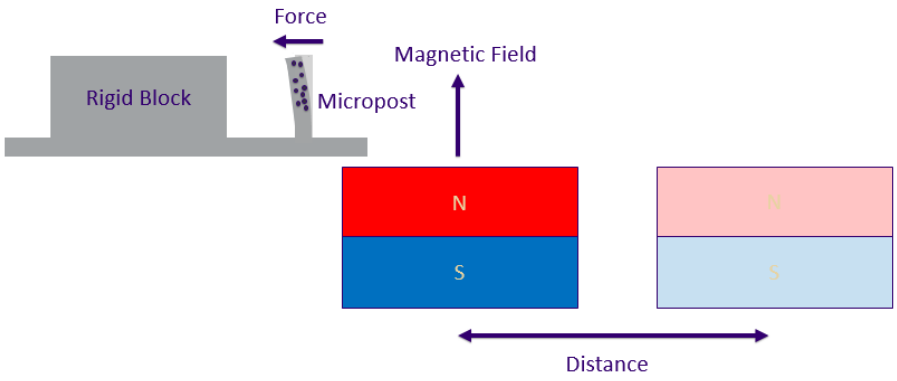

Figure 3.1 – Changing the linear distance between two magnetic dipoles will change the force applied on the microposts

3.2.2 – Design of a Cam-Slider Actuator

To achieve a dynamic linear motion via a slider system, I needed a method to translate the rotary motion produced by the motor into a linear motion. Previous attempts had been made to use a rotating cam attached to the motor shaft to drive the slider [22]. The cam could be designed to geometrically fit a desired profile that corresponded to a specified linear velocity and acceleration. Individual cam and slider parts from the previous designs were still available, so I created a prototype using an old cam and slider, with a newly designed base and supports [Figure 3.2].

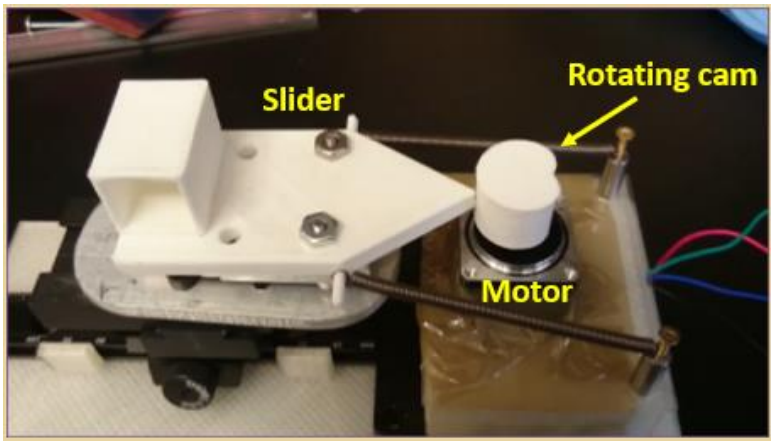

Figure 3.2 – The cam-slider prototype where the rotating cam linearly drives the slider to create an oscillatory movement

Initially, the plan was to use an inverted fluorescent microscope (Nikon Eclipse) for imaging. Due to the presence of an inverted objective, the slider system had to be positioned in such a way that it did not block the objective but still be close enough to apply a force on the microposts. This presented a problem stemming from the lack of space on the microscope stage, necessitating a need for a long extension for the slider to reach the objective. Adding the required extension produced a large torque on the system, making it difficult to keep the slider stable. The presence of a climate-controlled box around the microscope also reduced the available space. The supports needed to hold up the cam-slider were bulky, and the sliding range of motion required additional room.

Furthermore, designing a cam to achieve the desired post movement, in terms of frequency and amplitude, proved challenging. Due to these complications, it was clear that I would need to design a more compact design with less moving parts to be able to fit the constraints and be usable for experimentation.

3.2.3 – Producing a Magnetic Force by Changing the Magnetic Moment Orientation

To eliminate the need for a linear motion to change the magnetic force, the permanent bar magnet was replaced with a torque magnet (Polymagnet 1001122) [Figure 3.3a]. The torque magnet switches its dipole orientation every 90 degrees, which provides a change in the magnetic field gradient as the magnet is rotated about its axis [Figure 3.3b]. By placing the magnet directly below the microposts and rotating about its axis, the applied force could be controlled. Unlike the bar magnet, the torque magnet produced magnetic fields that were far more localized. This was beneficial as it ensured that the magnetic field stayed contained in a much smaller area and did not interfere with the microscope and the electronics.

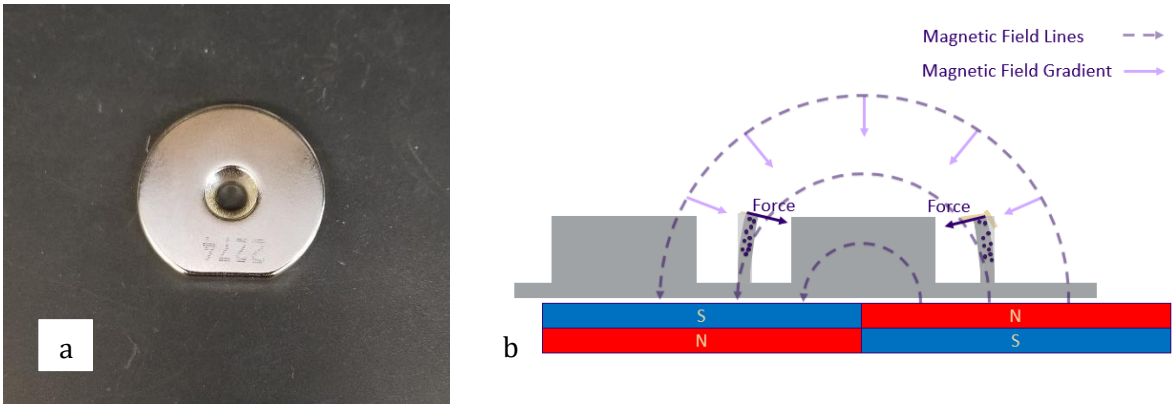

Figure 3.3 – (a) Top view of the torque magnet, (b) Side view showing the magnetic field lines of a torque magnet, which causes a change in the magnetic field gradient that applies a force on the microposts

It was important to quantify how the magnetic field changed as the torque magnet rotated. For this purpose, a gaussmeter (Lakeshore 410) was used to track the field magnitude versus the rotation angle in the r, θ, and z orientations [Figure 3.4]. With the actuator, the magnet would only be moving in the θ direction, so the z orientation coming out of the magnet would be the only field that was not constant. Therefore, the r and θ fields would have a negligible effect on the microposts. Based on the measurements, a region of about 10 degrees on either side of the dipole transition point showed a linear field change for the z orientation as the angle was changed. At larger angles from the transition point, the field gradient leveled off until the next transition point approached. This showed that sweeps below 20 degrees to could be used to apply a stable and linear force on the microposts.

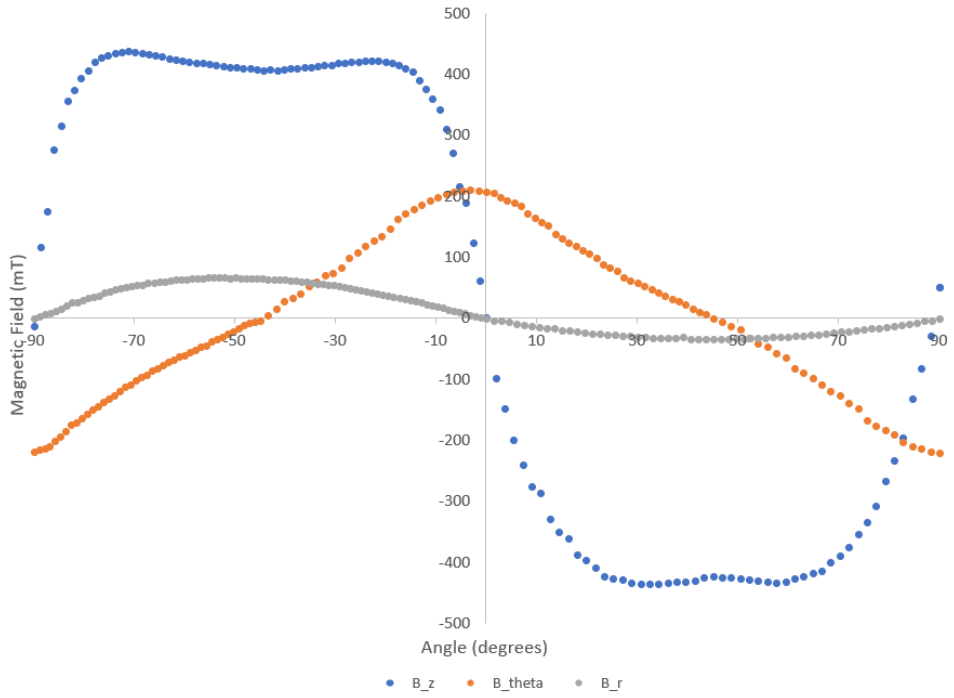

Figure 3.4 – Magnetic field magnitude as a function of angle distance from transition point for z, θ, and r orientation

3.2.4 – Design of a Dual-Axis Rotational Actuator

The first change to simplify the actuator fit issues was to move from the inverted microscope to an upright metallurgical microscope (Nikon LV100ND) [Figure 3.5]. This allowed the torque magnet to be placed directly underneath the glass slide containing the microposts, as the light would reflect back to the objective. To position the microfluidic device, a slide holder was designed and hooked up to a micromanipulator (Eppendorf). The main fit constraint for this microscope was the height limit for the magnet to be positioned under the glass slide such that there was enough working distance to be able to focus on the microposts. A compact design was again vital, as the climate-controlled box for the metallurgical microscope was smaller than the box for the inverted scope.

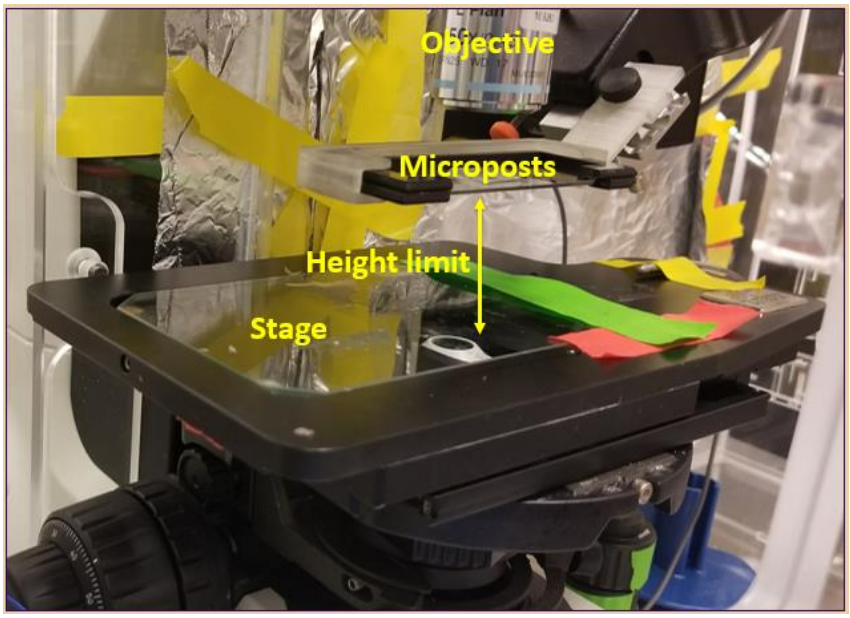

Figure 3.5 – The upright metallurgical microscope stage showing the slide holder for the microchannels and the available space to place the actuator

Simply attaching the torque magnet directly to the motor shaft was not a viable solution as the magnet would sit too high and not fit underneath the microposts. Instead, I designed a dual-axis pulley system, where the motor was inverted and the magnet would sit on a separate pulley. Initially,

the feasibility of such a design was tested using 3-D printed pulleys connected with yarn, where one pulley was driven by the motor [Figure 3.6]. The simple pulley design showed that the motor could drive a pulley on a separate axis without slipping, while significantly lowering the magnet height and freeing up space around the glass slide.

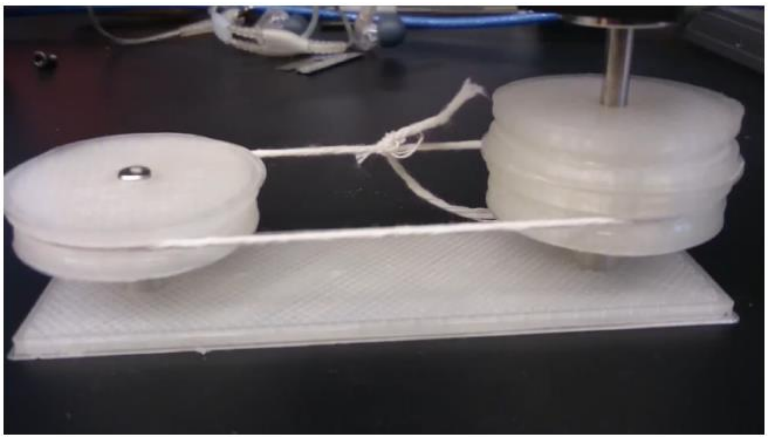

Figure 3.6 – The initial pulley system proof-of-concept with 3D printed pulleys and yarn belt

To better control the pulley movement and further reduce slippage, the 3D printed pulleys were replaced by aluminum T5 timing pulleys attached with a 225 mm urethane timing belt. The pulley axes were fixed on a 3D printed base and the pulleys rotated on a pair of steel shaft collar bearings. Next, a support was added to hold the position of the inverted motor [Figure 3.7]. This prototype was far more compact than the cam-slider, and it was able to produce a sufficient amount of torque to reliably turn the magnet. However, there were some stability issues that needed to be addressed. Due to the inward force generated by the timing belt, the plastic base was warping and not fully sitting on the stage. Also, when changing the rotation direction, there was noticeable slack in the timing belt leading to a delay in the magnet response. Additionally, the vibration from the motor was propagating through the stage into the microscope and could be visibly observed when imaging. Finally, while the height of the magnet was low enough to fit below the glass slide, it was very close to the limit, which left little room for adjustments.

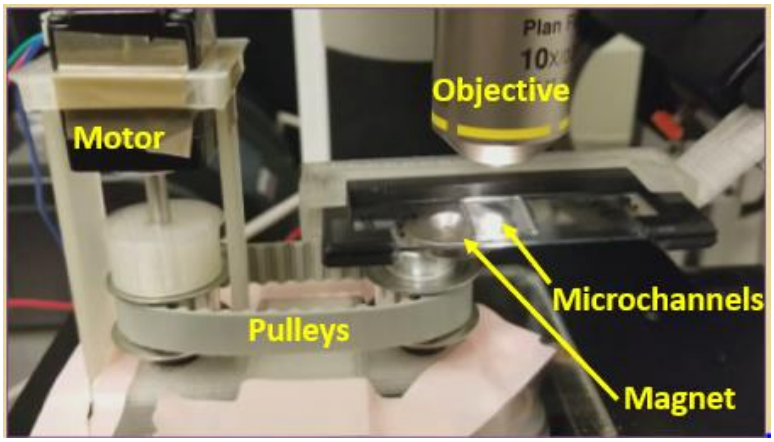

Figure 3.7 – The prototype with T5 pulleys and timing belt with 3D printed supports

To fix these issues, a new design was developed [Figure 3.8]. The major change was moving from the 3D printed plastic to an aluminum base and support, which provided far greater stability for the entire device and removed any warping. To reduce the slack in the belt, a torsion band was placed between the pulleys. Below the base, a thin layer of PDMS was added to dampen any vibrations propagating from the motor. To further lower the magnet height and allow for more leeway to position the glass slide, the shaft collars were replaced with thin plastic washer bearings. This also reduced the friction on the pulleys and produced a smoother rotation. Overall, the aluminum prototype was a compact design that fit the working space available, and it was stable such that the only movement was the motor input and the corresponding magnet rotation. This was the final iteration that was built, and it was then used for characterization and testing.

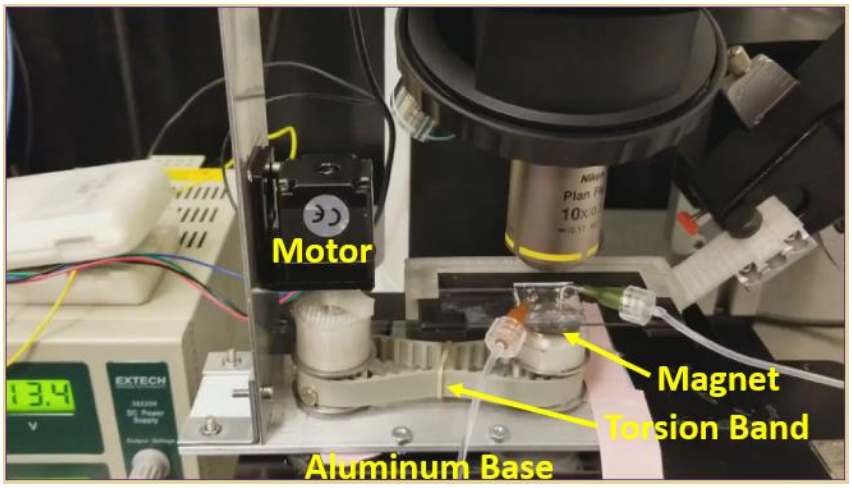

Figure 3.8 – The final actuator design built with aluminum base and support

3.3 – The Controls Design Process

During the controls design process, the electronics required to run and control the motor input were set up. During this process, I developed an understanding of the capabilities of the motor, the motor driver, and the programmable user input, while fine tuning the electronic settings for the experimental requirements. This section will go over the basics of the electronic components, how they fit together, and the control capabilities I had when programming the motor inputs.

3.3.1 – Basics of the Stepper Motor

A bipolar stepper motor (Pololu #1205) was used to drive the actuator. A stepper motor uses sets of electromagnetic coils that energize in phases to rotate the motor in discrete steps [24]. Using a stepper motor comes with the benefit of high accuracy and repeatability, as well as the ability to reliably hold its position [25], [26]. However, unlike a servo motor, stepper motors have no feedback control [27]. The motor chosen for the actuator had a 28x32 mm base, drew 670 mA of current per phase at 3.8 V, while providing a holding torque of 600 g-cm. Each standard full-step was 1.8° (200 steps/revolution).

The Allegro A4988 stepper motor driver was used to control the motor electronics. The driver was connected to a microcontroller (Arduino Uno v3) that was programmed to input step and direction pulses that drove the motor. The driver was microstepping capable, and could operate the motor in full-, half-, quarter-, eighth-, and sixteenth-step modes, cutting down the angle of each step. The driver also included a potentiometer that could control the current flow.

For a minimal wiring connection to run the motor, the following was required. The four motor leads were connected to a stepper motor driver. To power the motor, a power supply provided about 10 Volts and was connected to the driver. A 100 µF capacitor was placed between the voltage in and out pins to protect the driver and motor from shorting. For programming inputs, an Arduino microcontroller was connected to the step and direction driver via *digital in* pins and were able sent information to a computer via a serial USB port [Figure 3.9]. Once a program was started, it would run as an infinite loop until power was removed.

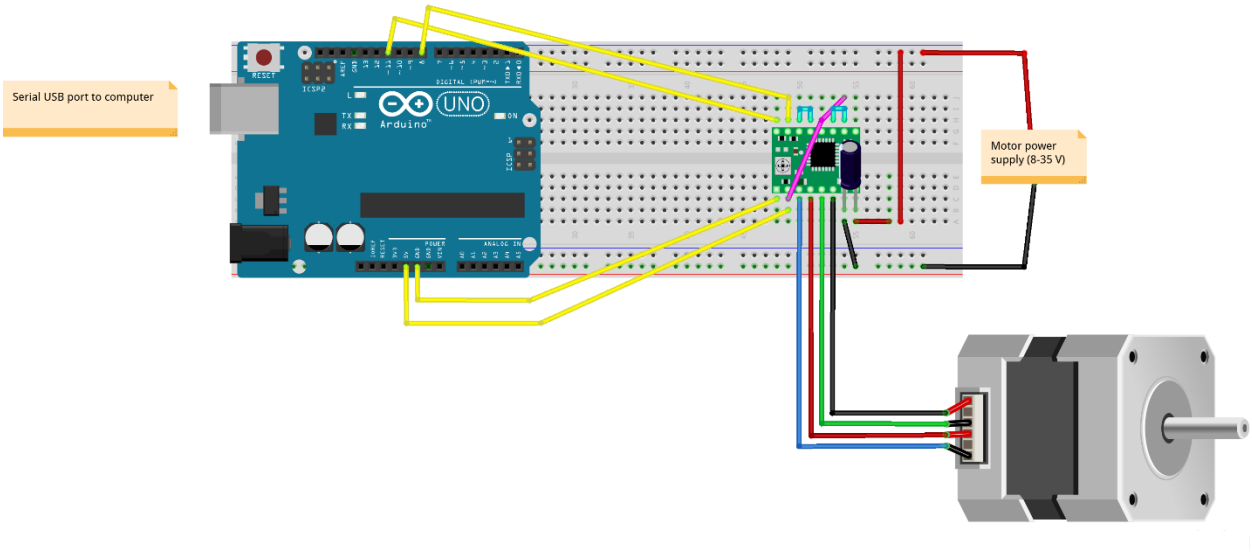

Figure 3.9 – Wiring diagram showing layout of electronics, including Arduino microcontroller and A4988 driver, required to run the stepper motor with eighth-step microstepping

3.3.2 – The Hardwired Variables for the Motor Input

There were four variables that were directly controllable when programming the motor input. Two variables, the step size and the current flow, were controlled through hardwiring of the motor driver. Therefore, these needed to be set before experiments began and were kept unchanged throughout. The step size corresponded to the angle of each discrete step, and it could be changed by activating microstepping combinations of pull-down resistors on the driver [Table 3.1]. Smaller step sizes produced a smoother motion (with more steps per revolution) but were able to provide less torque in the process. To change the current flow, the potentiometer on the driver could be rotated. Ramping up the current flow increased the torque but also increased the vibration propagating through the motor. To choose the most advantageous step size and current flow, each variable was weighted based on being able to provide the cleanest rotation while still having enough torque to avoid slippage.

Table 3.1 – Pull-down resistor combinations on A4988 driver to select microstepping resolution

MS1	**MS2**	**MS3**	**Resolution**
LOW	LOW	LOW	Full Step
HIGH	LOW	LOW	Half Step
LOW	HIGH	LOW	Quarter Step
HIGH	HIGH	LOW	Eighth Step
HIGH	HIGH	HIGH	Sixteenth Step

To find the optimal microstep size, the major characteristics of each step size were charted based on a standard sweep angle of 10 degrees [Table 3.2]. After testing each size with the imaging

setup, the full-step and half-step sizes were immediately disregarded due to the presence of visible vibration interference in the captured video. By lowering the step size to at least quarter-steps, any visible vibration disappeared. The other major consideration for choosing the step size was to maintain enough steps per cycle such that no data was lost via imaging. According to the Nyquist sampling theorem shown in Equation 3.2, the image sampling rate (f_s) needed to be at least twice that of the discrete input frequency (f_{max}) to prevent any data loss [28].

$$f_s > 2f_{max} \tag{3.2}$$

At the largest possible field of view, the camera maintained a maximum frame rate of 45 frames per second. Therefore, each input cycle would require at least 22 steps. Based on Table 3.2, quarter-steps were very close to the sampling limit; so instead, eighth-steps were chosen to provide more data points, while still generating sufficient torque to drive the system.

Table 3.2 – Characteristics of different microstep resolutions based on a 10-degree magnet rotation

Step size	Steps per revolution	Steps per 10 degrees	Theoretical max rotational velocity (rad/s)	Vibration interference
1	200	5	16	Y
1/2	400	11	8	Y
1/4	800	22	4	N
1/8	1600	44	2	N
1/16	3200	89	1	N

Once the step size was chosen, the current flow was set with the built-in potentiometer. The ideal current flow was experimentally found to be approximately 0.3 amps, which produced a voltage

in the power source of about six volts. This again provided a vibration-free input with a sufficient driving torque. Once they were set, the microstepping size and the current flow were not altered for the rest of the experimentation process.

3.3.3 – The Programmable Variables for the Motor Input

The other two controllable variables were number of steps before a direction switch and the delay between each step. These were set by programming the Arduino microcontroller and corresponded to the sweep angle and the rotational velocity. Using Arduino Software, the *DigitalWrite()* command was used to specify either a step or direction switch, while the *Delay()* command held the program for a specified number of milliseconds between each discrete step. By implementing *while* loops using the two commands, a wide variety of motor input programs could be created, including sine curves, step functions, ramps, and chirps. The motor was capable of reading delays as small as to one millisecond, which provided an upper limit for the input rotational velocity.

3.3.4 – Additional Electronic Components

Because of the need to eventually measure the phase lag in viscoelastic fluids, a method of timestamping the inputs and outputs was required. To achieve this, the Arduino was connected to a data acquisition system (DAQ) and programmed to send a digital pulse at every direction switch to a timing program in LabView. This could be matched with a timestamp of every image frame from the camera so that the input magnet orientation and output deflection could be compared during analysis.

One of the disadvantages of using a stepper motor was that it provided no feedback control. If the input program was reset, the motor had no method of tracking its previous “home” location. This could lead to shifts in the magnet orientation every time the input program was reset or changed. To solve this issue, a physical limit switch that acted as a homing device was installed and wired to

the Arduino [Figure 3.10]. At a reset, an Arduino *setup()* loop was implemented such that the pulley would rotate until a 3D printed rod activated the switch. When the switch was detected as on, the motor would switch direction and then step a predefined distance, which ensured that the previously set magnet position was maintained throughout individual tests.

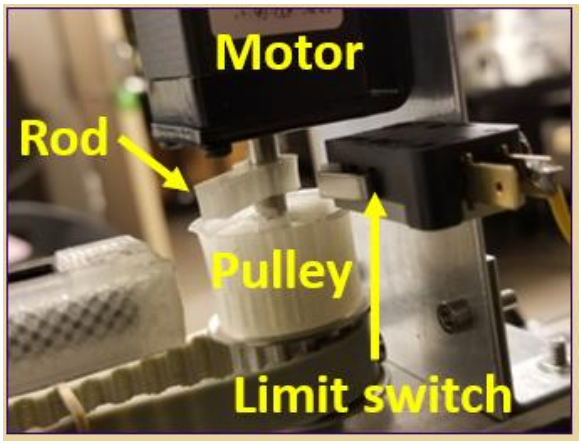

Figure 3.10 – The limit switch set up where a 3D printed rod is attached to the motor shaft and hits the limit switch to activate the homing sequence

3.4 – Discussion

I focused on two major aspects during the design process, the mechanical build and the electronics control of the actuator. For the mechanical side, my main goal was to build the actuator such that it would fit the physical constraints required to fit with the experimental setup with the upright metallurgical microscope. I achieved this using a torque magnet that applied force by simply rotating the magnet about its axis, which drastically cut down on the required movement of the device. For the electronics aspect, I was able to control the motor in a way that provided a precise and stable rotation of the magnet that fit with the constraints imposed by the motor, the driver, and the microscope camera.

Chapter 4 – Characterization of the Actuator

4.1 – Overview of the Characterization Tests

To determine whether visual tracking of the actuated posts produced the expected response to the motor inputs, the post movements were characterized in an air medium. The characterization plan was to focus on two main aspects of the deflections: the amplitude and the frequency. For the input, I programmed the magnet to rotate with a constant delay between steps, which produced a triangle wave with a constant frequency [Figure 4.1]. At high frequencies, this could be modeled with a sine curve.

For the amplitude test, I verified whether the post deflections showed a linear increase corresponding to a linear increase in the magnet sweep angle. For the frequency test, I tested how closely the post deflection frequency matched to the input. I also quantified the presence of background noise and phase lag in the system and demonstrated the ability to program various alternate inputs that would be important for future viscoelastic tests.

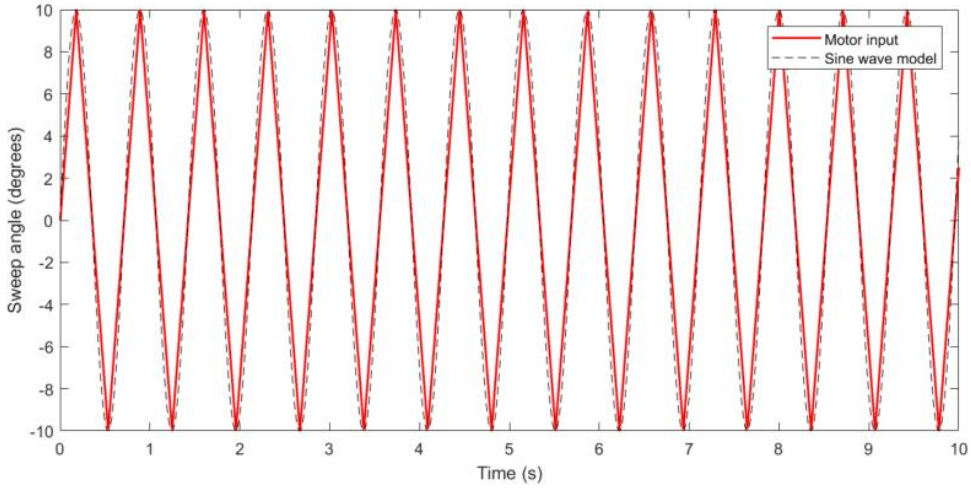

Figure 4.1 – Motor rotation with varying sweep angle at constant frequency compared with sine wave model

The block and post devices with embedded magnetic particles were fabricated using the process described in Section 2.4.3. To improve the embedding process, an extra step was added, in which the devices were placed in a centrifuge to drive the particles into the posts. Image analysis was performed using a combination of manual image processing and MATLAB code. First, the raw images were converted to a binary format. By using both a size and circularity threshold, the micropost outlines could then be tracked, allowing the movement of their centroids to be recorded [Figure 4.2]. The centroid data was sorted for each individual post and compiled, such that each post had a corresponding x and y position, area, and a timestamp for every frame. Usually for each video, a set of 3-5 posts that would respond to movement could be visualized, allowing them to be tracked simultaneously.

For characterizing maximums in the post movement, the magnet was positioned to produce post deflections that were perpendicular to the flow direction of the channel. For testing blood samples, the force would normally be applied parallel to the flow direction, so that the posts pulled on platelet plugs as they formed. However, when testing large deflection amplitudes, the parallel movement produced a deflection that would bring it too close to the neighboring block, sometimes causing them to stick together. By changing the motion to perpendicular to the flow, the posts were free to deflect fully without any objects in the way. To change the deflection direction, the magnet simply needed to be rotated 90 degrees so that the sweep orientation was switched.

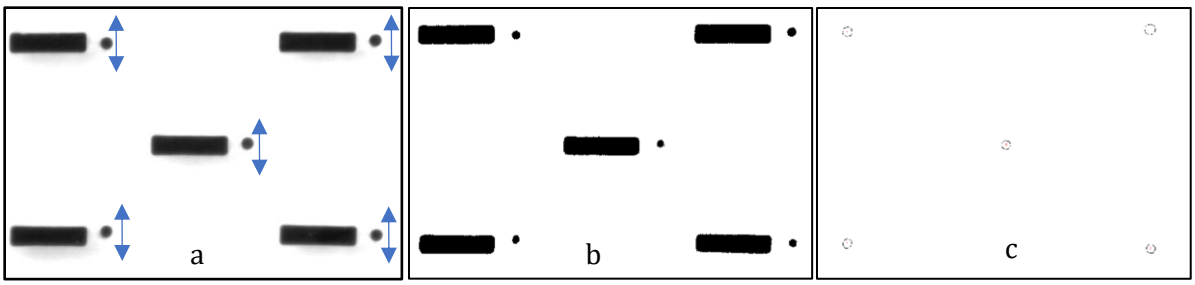

Figure 4.2 – (a) Raw image file of block and post structures; arrows show post movement direction perpendicular to the channel flow direction, (b) conversion to binary format, (c) with size and geometry threshold applied to track post outlines and corresponding centroids

4.1.1 – Converting from Input to Output Variables

As mentioned in Section 3.3.3, the Arduino program followed a motion based on two variables, the step size and delay. However, for the characterization I was calculating theoretical input angular velocity and frequency values in degrees and Hertz. Therefore, Equations 4.1-4.3 were used to convert between the input and output variables.

$$a = \frac{s}{1600}\,(360) \tag{4.1}$$

$$\omega = \frac{2\pi}{1600d} \tag{4.2}$$

$$f = \frac{\omega}{a(\frac{\pi}{180})} \tag{4.3}$$

Here, a is the sweep angle in degrees, s is the number of steps the motor takes before a direction switch, ω is the angular velocity, d is the delay between each step in milliseconds, and f is the frequency in Hertz.

4.2 – Linearity of the System

Based on the magnetic field tracking that was performed in Section 3.2.4, a region where the applied magnetic field gradient increased linearly was determined. This held true for approximately ten degrees on either side of the magnet transition point. According to Equation 3.1, if the permanent magnet was placed directly below a micropost and centered exactly at the transition point, the applied force would only act along one direction and all other components would cancel out. Therefore, the magnetic post deflection amplitude would theoretically follow the linear gradient from the magnet as the sweep angle increased.

The magnet was programmed to input a constant sinusoidal sweep at a set angle and was centered a closely as possible to the transition point. The height of the magnet was raised to be as close as possible to the glass slide containing the microposts without touching it. For each sweep angle, a video was taken at 45 fps for 10 seconds. Various sweep angles were tested between 1-24 degrees, while the frequency was kept at a constant rate of 1.41 Hz for all tests. The process was repeated for four different devices that were all fabricated separately. Each test produced an output of sinusoidal waveform of all the posts in the device [Figure 4.3]. From this, the average deflection amplitude of every post, as well as the average of the all posts in a test, was plotted against the corresponding sweep angle [Figure 4.4]. The average deflections of all the posts from each device were then plotted to compare the variability in different devices [Figure 4.5].

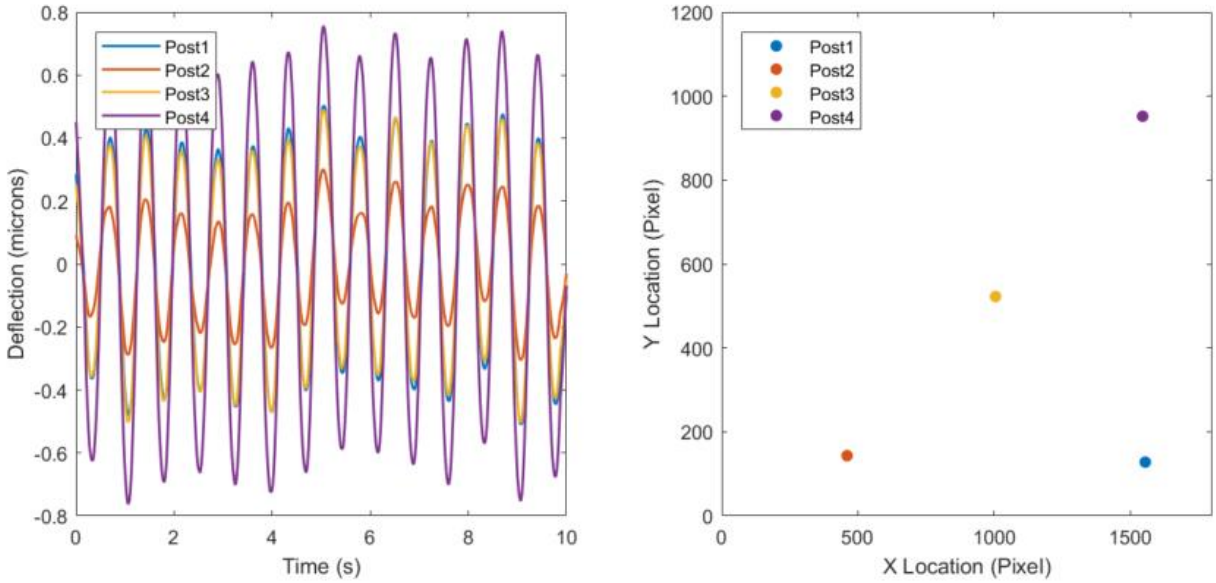

Figure 4.3 – (a)Deflection of all posts in a single device at eight-degree sweep angle, (b) Locations of each post in relation to each other

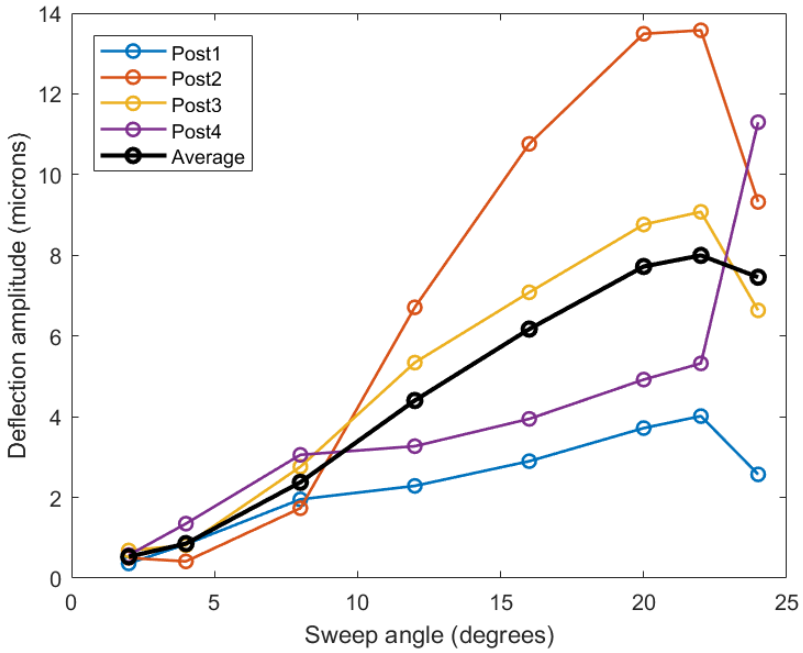

Figure 4.4 – Average deflection amplitude for individual posts in one device as a function of sweep angle, and average deflection of all posts in one device

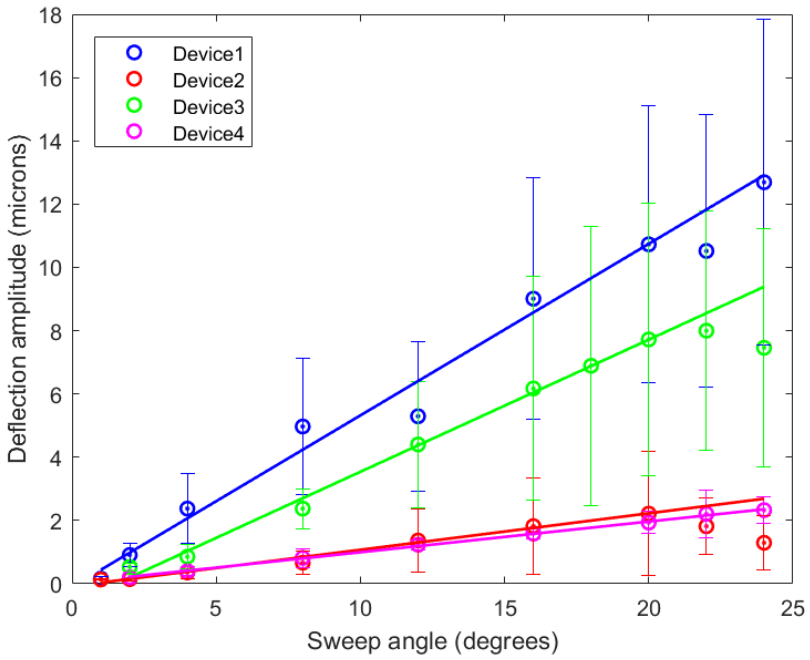

Figure 4.5 – Average post deflection for four different devices vs sweep angle, with lines of best fit shown with standard deviation error bars; variability was greater for devices exhibiting more movement and increased with sweep angle

The average amplitudes for individual posts did not follow a particular curve, but when taking a mean for each device, a linear pattern could be traced. Individual posts in the same device showed a wide range of deflections, with values between 3-15 μm seen at large sweep angles. Devices that exhibited greater deflections also tended to a have a much larger variability in the individual posts. The variability also was seen to increase with the sweep angle. The linear rise was consistently seen for a sweep angle of up to 20 degrees, but deflections flattened or decreased after that point. A flattening was expected knowing the magnet gradient, but the decrease was probably due to a slight error in centering the input sweep at the transition point, which would cause an offset in the applied magnetic field gradient. The variance in the individual posts could be attributed to the location of the post in relation to the magnet transition point or the amount of embedded magnetic material present in the post. These effects could be further quantified in the future.

4.3 – Quantifying the Presence of Baseline Noise

Analysis of the amplitude tests showed that with a one-degree sweep angle, the peak amplitudes could not be clearly deduced from the waveform. At two degrees, the amplitudes could be seen, but there was a large amount of drift present [Figure 4.6]. At larger sweep angles, the drift tended to not be as visible. I hypothesized that this drift was caused by the presence of some amount of baseline noise inherent to the system. To test the amount of noise, videos of the microposts were taken with the actuator turned off and the resulting deflection was plotted [Figure 4.7].

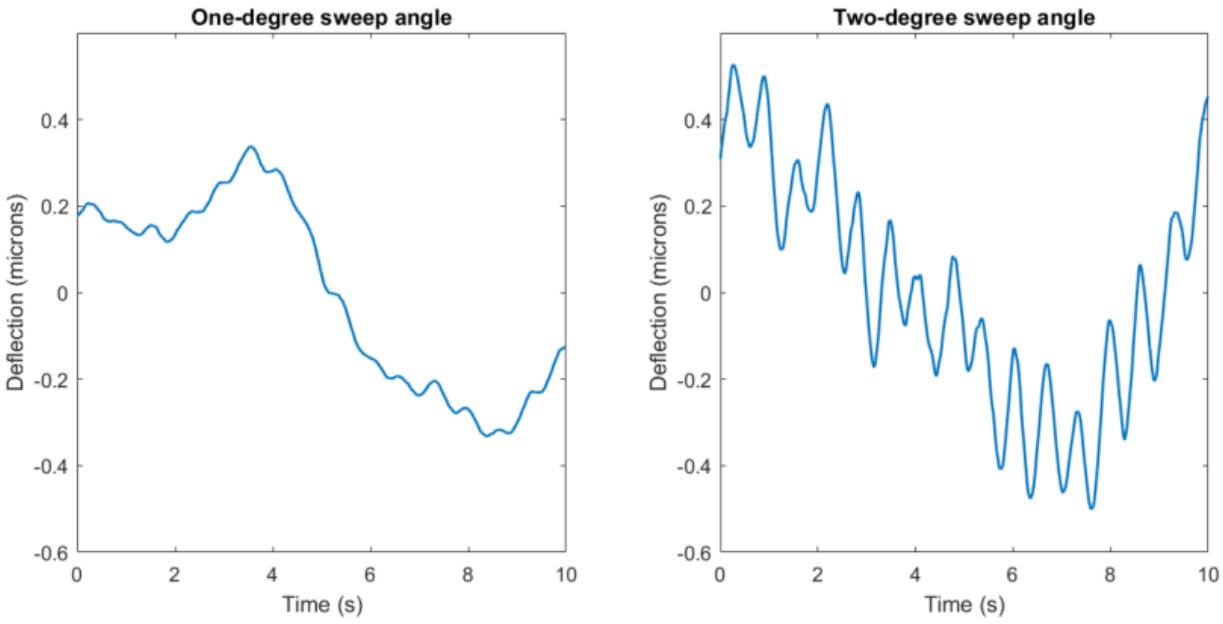

Figure 4.6 – Deflection waveform for one-degree and two-degree sweep angles for the same post; no oscillation could be tracked at one-degree

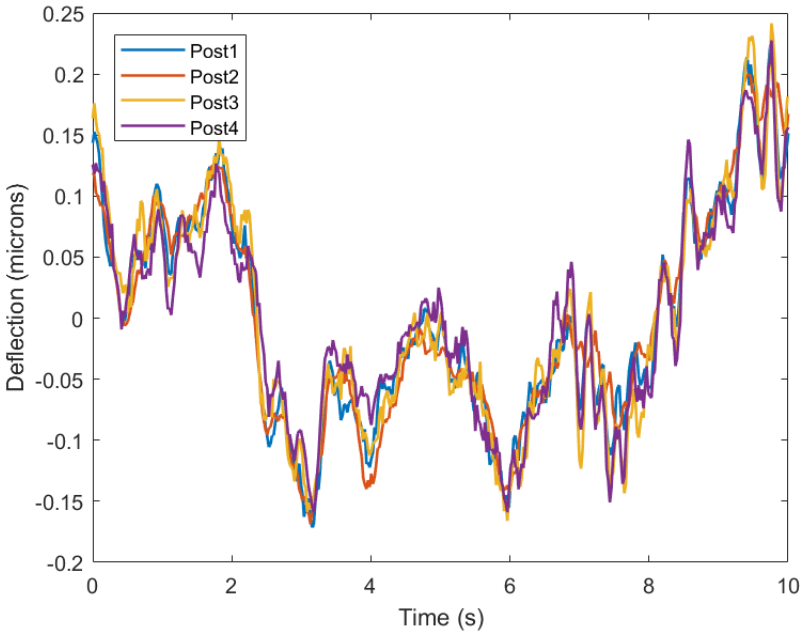

Figure 4.7 – Post deflections in a single device with actuator turned off

Three devices were tested, and the noise was found to be in a range between ±0.25 μm. For each device, the maximum deflections above and below the center point were measured [Table 4.1]. All the posts in each individual device followed the same general noise curve, which suggests that the primary noise source is external movement of the device or experimental setup, rather than from errors in the image analysis.

Table 4.1 – Tracking of maximum deflection with actuator turned off

Device	**Max deflection above center (μm)**	**Max deflection below center (μm)**
Device 1	0.242	0.171
Device 2	0.147	0.199
Device 3	0.186	0.213

4.4 – Matching the Input and Output Frequencies

An important feature when considering the responsiveness of the actuation system was how closely the output and input frequencies matched with each other. The existence of a matching output frequency to that of calculated theoretical frequency values would suggest that the actuator was properly transferring motion to the posts, meaning that there was a minimal amount of mechanical slack or lag that would produce errors. Frequencies were tested to measure to values: how well the magnitudes of the output frequencies matched to the input and how much phase lag was present between the input and output oscillations.

4.4.1 – Measuring the Frequency of the Micropost Deflections

To measure the frequency of the post deflections, a constant frequency oscillation was input and the timestamp of each resulting peak in the sinusoidal waveform of individual posts was extracted. The difference in time for each peak was taken to calculate the periods, and this was converted into frequencies using Equation 4.4,

$$f = 1/T \tag{4.4}$$

where f is the frequency and T is the period of oscillation. The frequencies for the waveform were then averaged to produce a mean frequency for an individual post. The mean frequency for all the posts in a single device were then plotted in comparison to a theoretical input frequency of 1.4063 Hz, which was calculated using Equation 4.3 based on the number of steps and delays that were programmed [Figure 4.8]. The mean frequency and range is shown for three separated devices with for a sweep angle of four degrees [Table 4.2]. Finally, to verify that the output frequencies continued to match for a range of inputs, the devices were tested with five different of input frequencies from 0.14 – 4.7 Hz [Figure 4.9].

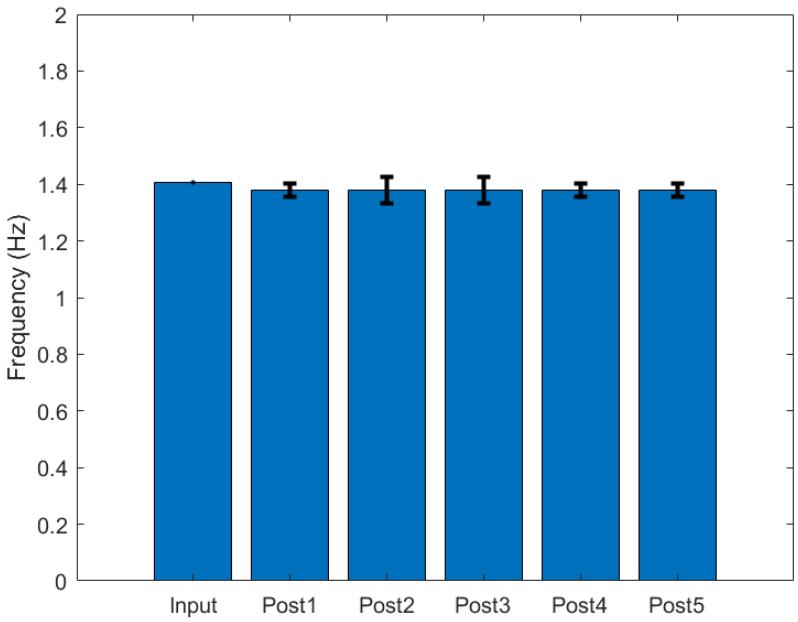

Figure 4.8 – Average frequency of post movements for individual posts in a single device compared to input frequency calculated using Equation 4.3, with error bars showing standard deviation for frequencies of individual peaks

Table 4.2 – Constant frequency characteristics for individual posts in a single device and comparison to other devices for a four-degree sweep angle; calculated input frequency was 1.4063 Hz

Device	**Mean frequency (Hz) for all posts**	**Average Percent Error (%) from theoretical frequency**	**Range of values (Hz) between individual posts**
Device 1	1.380	2.37	0.0036
Device 2	1.389	1.21	0.0041
Device 3	1.372	2.43	0.0003

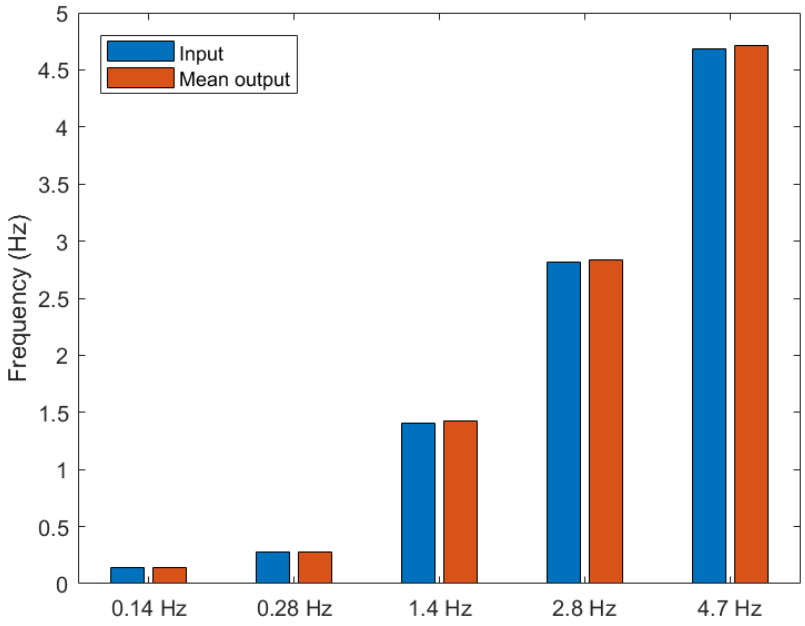

Figure 4.9 – Mean frequencies of post movements across three devices compared to various input frequencies

There was very little variance in average frequencies for individual posts in a single device, which meant that the location of the post in relation to the transition point of the magnet plays no role in the output frequency. This could be deduced from the low difference in the range of frequencies found, with values in the range of thousandths of Hertz, which was essentially negligible. There was more variance found in the individual peaks of each waveform, but taking the mean value produced a value that was very close to the input frequency. For all the tested devices across the range of inputs, mean post frequencies were found to be within 3% of the theoretical input value. Most of this error probably stemmed from rounding errors when converting from motor steps to degrees. Overall, this data showed that for the range of tested input frequencies, the magnitude of the post frequencies matched closely.

4.4.2 – Quantifying the Phase Lag in the System

Due to the use of a pulley system to transfer the motion of the motor shaft, there was a likelihood that some amount of lag existed between the oscillations of the magnet and the microposts. To quantify this, a constant frequency was input, and the resulting timestamps from the waveforms of individual posts were compared with timestamps of a digital signal sent at the direction switch of the motor. These values were acquired using the DAQ system mentioned in section 3.3.4. The time difference between the input direction switch and the peak deflection times for a single post in three different devices were then plotted [Figure 4.10].

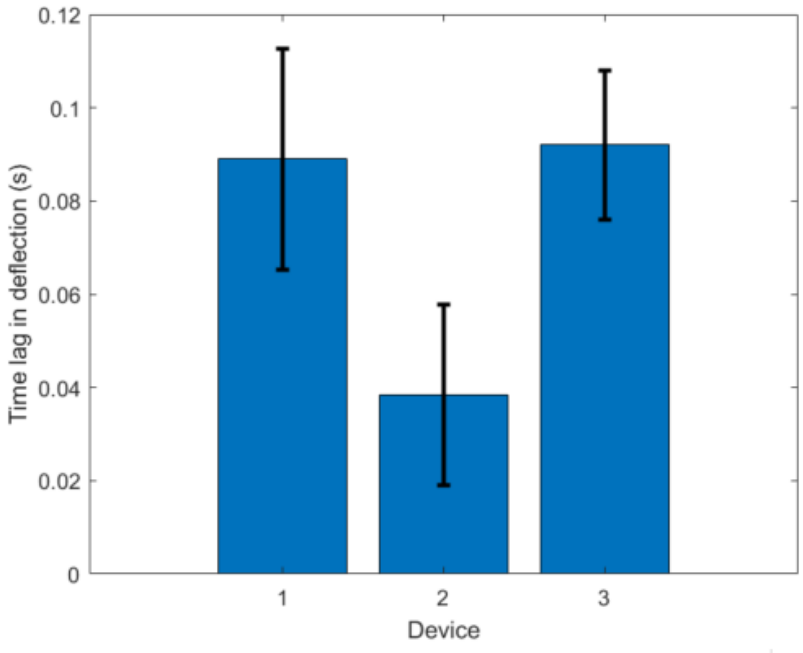

Figure 4.10 – Average phase lag between time of input magnet direction switch and time of corresponding peak deflection amplitude for three different devices, with error bars showing standard deviation of lag for individual peaks

For the three tested device, the results showed an overall average of 0.07 seconds of delay in the post deflections. Based on the input frequency, the input period was calculated to be 0.711

seconds, so the delay corresponded to close to 10% of the period. This data suggests the existence of slack in the pulley as the motor switched direction.

4.5 – Other Input Capabilities

For the characterization of the actuator, a simple sine input was used. However, with small alterations to the input code, a wide variety of inputs could be programmed easily. This allowed the actuator to potentially be used for various rheological tests to dynamically measure viscoelastic properties in a microfluidic setting.

Unlike Newtonian fluids, viscoelastic fluids such as blood have a time dependence when a stress is imposed upon them [29]. One of the common methods of characterizing this dependence is with a creep test, where stresses are applied and relaxed, to measure how a material can recover to its original state. In the case of microfluidic devices, dynamic viscoelastic environments have not been extensively studied [30]. With this actuator, a creep function can be input to test the viscoelastic properties of fluids in the microchannels using the deflections of the microposts. As a proof-of-concept, a preliminary creep input was implemented and the micropost deflection was measured [Figure 4.11].

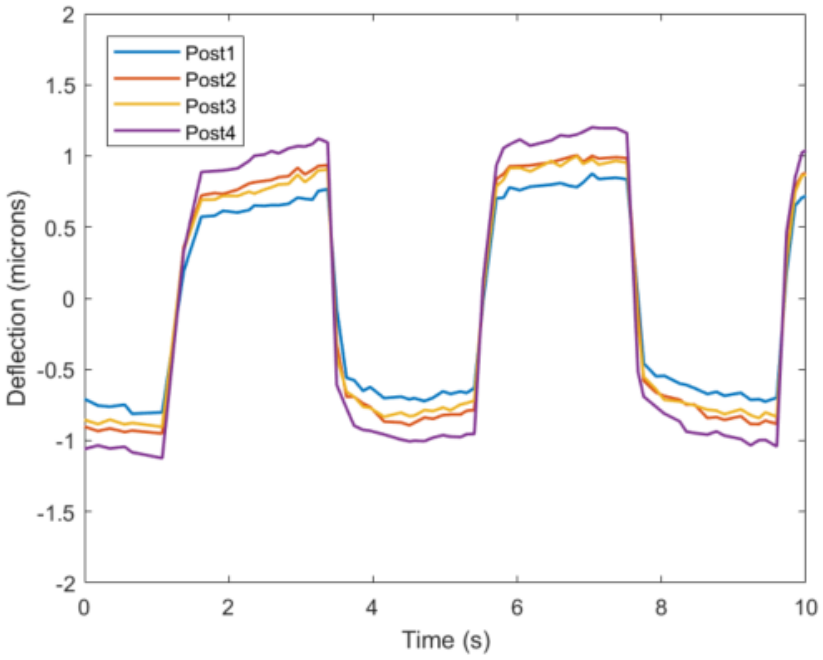

Figure 4.11 – Post response to input creep function showing creep recovery curves of individual posts

Another common viscoelastic test can be created by using a stress ramp function. By applying a steadily increasing force, the yield stress of a material can be determined [31]. A hold step can be added to provide information on material relaxation time. A proof-of-concept for a steadily increasing force was also tested [Figure 4.12].

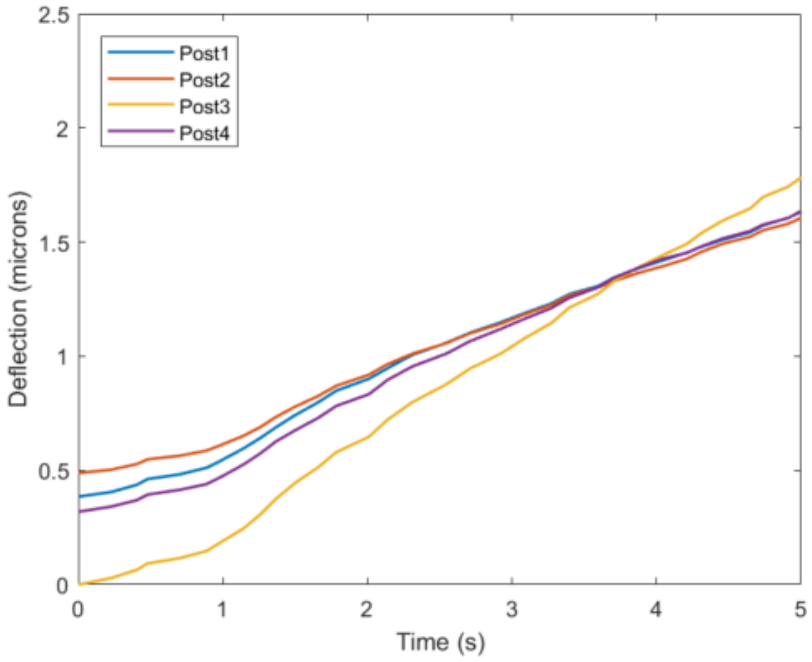

Figure 4.12 – Post response to stress ramp function input showing constant increase in deflection amplitudes over time

4.6 – Discussion

During the characterization process, I used a constant frequency sinusoidal input to quantify the response of the magnetic microposts as input sweep angle and frequency. In terms of the sweep angle, I was able to confirm that the linear magnetic field gradient found on the torque magnet was able to transfer to the posts to provide a linear increase in deflection. For a constant frequency input, I showed that the response matched closely, both in terms of individual post and through different devices. I also established the presence of a phase lag between the input and output oscillations. Finally, I introduced other input capabilities, such as the creep and ramp test. More work can be done to measure the response of a varying frequency input, as well as measuring the upper limits of frequencies that can be reached.

Chapter 5 – Conclusions and Future Directions

In this paper, I have presented a motor-driven actuator designed for the purpose of obtaining real-time measurements of viscoelastic biomaterials in a microfluidic device. In particular, it can be used to measure various biomechanical properties of platelets as they aggregate and start to clot. In this chapter, I will review the project goals, summarize the design process that I undertook, and introduce future research directions and applications for this device.

5.1 – Summary of Work

Previously, it had been shown that a microfluidic device containing PDMS block and post structures could induce platelets to activate and begin clotting to the posts such that their forces could be measured from the post deflections [5]. Further, it was shown that magnetic material could be embedded in the posts such that they would respond to external magnetic fields [6]. Using a slider-system, static measurements could be taken with the magnetic field on or off. The goal of this project was to design a motorized device that could magnetically apply external shear forces on platelets in real-time as they clot.

During the design process, I developed a mechanical actuator where the rotation of a torque magnet was driven by a stepper motor. The major constraint for the physical build was the limited space available to place the actuator such that it would fit with the microscope and experimental setup, particularly where the height of the magnet was low enough to fit below the microposts. To ensure that the actuator build was compact and the magnet was placed at a low height, a dual axis system was developed, where the motor was inverted and drove the magnet rotation via a set of pulleys. For the motor electronics, I set up the hardwired variables with the motor driver such that

the motor produced a smooth rotation with enough torque to produce slippage. This left the input rotation to be controlled by two programmable variables, the step size, and the delay between steps.

Once the actuator was built, the response of the microposts was characterized in terms of amplitude and frequency by using a simple triangle wave input. The results showed that the post deflection amplitudes could be modeled linearly for an input sweep angle of up to 20 degrees. The frequencies of the post motion also matched closely with the input frequency. I was also able to quantify the amount of baseline noise in the system and the amount of phase lag between the post and magnet motion. These tests show that the microposts respond to the applied magnetic field in a reliable and controllable way. More characterization could be done in the future to measure the effects of varying the frequency, as well as quantifying the frequency limits.

5.2 – Future Directions

To gain a better understanding of the differences in the movement of individual posts, more characterization tests could be run. In particular, the impacts the of the relative post locations and the amount of embedded magnetic particles in the microposts could be studied. For the post locations, the force direction and magnitude for each individual post could be mapped to see if the effect of its relative location to the magnet are consistent through different devices. For the embedded particles, the cross sections of posts could be studied to calculate the amount of particles as a percentage of the post height to see their effect on the post movement.

Furthermore, the actuator can be used to perform a variety of real-time biomechanical tests on platelets that improve our understanding of their function during the clotting process. One region of interest is to measure the change in stiffness of a platelet plug as it forms over time. This can be achieved by using the actuator to apply an oscillatory force on the plug and measuring the change in

post deflections over time. Another potential test would be to measure the critical force required to break a clot, which could be done by linearly increasing the applied force over time.

Additionally, this device could be used to quantify biomechanical properties of any viscoelastic biomaterial in a microfluidic scale. I have shown it is possible to input common viscoelastic tests such as a creep and ramp functions, and these could be used to quickly test the response of various materials.

5.3 – Final Conclusions

Magnetic actuation is a promising technique in microfluidics that allows for forces to be applied at a distance, while still maintaining reliability and control. Using this technique, I have designed a motorized actuator and shown that it can control the movement of magnetic microposts in a microfluidic device. In the future, it will hopefully be used to measure biomechanical properties of platelets in real-time and help us gain a better understanding of platelet function.

References

[1] A. J. Gale, "Current Understanding of Hemostasis," *Toxicol. Pathol.*, vol. 39, no. 1, pp. 273–280, 2011.

[2] S. Feghhi and N. J. Sniadecki, "Mechanobiology of Platelets: Techniques to Study the Role of Fluid Flow and Platelet Retraction Forces at the Micro- and Nano-Scale," *Int. J. Mol. Sci.*, vol. 12, no. 12, pp. 9009–9030, Dec. 2011.

[3] R. Tran *et al.*, "Biomechanics of haemostasis and thrombosis in health and disease: from the macro- to molecular scale," *J. Cell. Mol. Med.*, vol. 17, no. 5, pp. 579–596, May 2013.

[4] C. Zhang and S. Neelamegham, "Application of microfluidic devices in studies of thrombosis and hemostasis," *Platelets*, vol. 28, no. 5, pp. 434–440, Jul. 2017.

[5] L. Ting, S. Feghhi, A. Karchin, W. Tooley, and N. J. White, "Clot-On-A-Chip: A Microfluidic Device To Study Platelet Aggregation and Contractility Under Shear," *Blood*, vol. 122, no. 21, pp. 2363–2363, Nov. 2013.

[6] K. S. Bielawski and N. J. Sniadecki, "A Magnetic Post Approach for Measuring the Viscoelasticity of Biomaterials," *J. Microelectromechanical Syst.*, vol. 25, no. 1, pp. 153–159, Feb. 2016.

[7] K. Broos, H. B. Feys, S. F. De Meyer, K. Vanhoorelbeke, and H. Deckmyn, "Platelets at work in primary hemostasis," *Blood Rev.*, vol. 25, no. 4, pp. 155–167, Jul. 2011.

[8] M. Saboor, Q. Ayub, S. Ilyas, and Moinuddin, "Platelet receptors; an instrumental of platelet physiology," *Pak. J. Med. Sci.*, vol. 29, no. 3, pp. 891–896, 2013.

[9] S. Gogia and S. Neelamegham, "Role of fluid shear stress in regulating VWF structure, function and related blood disorders," *Biorheology*, vol. 52, no. 5–6, pp. 319–335.

[10] E. L. Bearer, J. M. Prakash, and Z. Li, "Actin Dynamics in Platelets," *Int. Rev. Cytol.*, vol. 217, pp. 137–182, 2002.

[11] D. Gailani and T. Renné, "Intrinsic Pathway of Coagulation and Arterial Thrombosis," *Arterioscler. Thromb. Vasc. Biol.*, vol. 27, no. 12, pp. 2507–2513, Dec. 2007.

[12] S. Palta, R. Saroa, and A. Palta, “Overview of the coagulation system,” *Indian J. Anaesth.*, vol. 58, no. 5, pp. 515–523, 2014.

[13] S. Collins, “Thromboelastography: Clinical Application, Interpretation, and Transfusion Management,” *AANA J.*, vol. 84, no. 2, pp. 129–134, Apr. 2016.

[14] M. M. Alves, C. Rocha, and M. P. Gonçalves, “Study of the rheological behaviour of human blood using a controlled stress rheometer,” *Clin. Hemorheol. Microcirc.*, vol. 53, no. 4, pp. 369–386, Apr. 2013.

[15] A. Sankarankutty, B. Nascimento, L. Teodoro da Luz, and S. Rizoli, “TEG® and ROTEM® in trauma: similar test but different results?,” *World J. Emerg. Surg. WJES*, vol. 7, no. Suppl 1, p. S3, Aug. 2012.

[16] N. J. White, E. J. Martin, D. F. Brophy, and K. R. Ward, “Examining platelet-fibrin interactions during traumatic shock in a swine model using platelet contractile force and clot elastic modulus,” *Blood Coagul. Fibrinolysis Int. J. Haemost. Thromb.*, vol. 22, no. 5, pp. 379–387, Jul. 2011.

[17] B. R. Branchford, C. J. Ng, K. B. Neeves, and J. Di Paola, “Microfluidic technology as an emerging clinical tool to evaluate thrombosis and hemostasis,” *Thromb. Res.*, vol. 136, no. 1, pp. 13–19, Jul. 2015.

[18] C. J. Glover, L. V. McIntire, L. B. Leverett, J. D. Hellums, C. H. Brown, and E. A. Natelson, “Effect of shear stress on clot structure formation,” *Trans. - Am. Soc. Artif. Intern. Organs*, vol. 20 B, pp. 463–468, 1974.

[19] J. Kuncova-Kallio and P. J. Kallio, “PDMS and its Suitability for Analytical Microfluidic Devices,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 2486–2489.

[20] B. A. Evans, A. R. Shields, R. L. Carroll, S. Washburn, M. R. Falvo, and R. Superfine, “Magnetically Actuated Nanorod Arrays as Biomimetic Cilia,” *Nano Lett.*, vol. 7, no. 5, pp. 1428–1434, May 2007.

[21] N. J. Sniadecki *et al.*, “Magnetic microposts as an approach to apply forces to living cells,” *Proc. Natl. Acad. Sci.*, vol. 104, no. 37, pp. 14553–14558, Sep. 2007.

[22] K. S. Bielawski, “Magnetic Devices and Techniques for the Study of Viscoelasticity of Biomaterials and Myocardial Forces,” Thesis, 2016.

[23] K. W. Yung, P. B. Landecker, and D. D. Villani, “An Analytic Solution for the Force Between Two Magnetic Dipoles,” *Physical Separation in Science and Engineering*, 1998. [Online]. Available: https://www.hindawi.com/journals/psse/1998/079537/abs/. [Accessed: 14-May-2018].

[24] M. Jenkins, “Hybrid stepper motors,” in *IEE Colloquium on Permanent Magnet Machines*, 1988, pp. 7/1-7/2.

[25] C. Muñiz, R. Levi, M. Benkrid, F. B. Rodríguez, and P. Varona, “Real-time control of stepper motors for mechano-sensory stimulation,” *J. Neurosci. Methods*, vol. 172, no. 1, pp. 105–111, Jul. 2008.

[26] M. Bodson, J. N. Chiasson, R. T. Novotnak, and R. B. Rekowski, “High-performance nonlinear feedback control of a permanent magnet stepper motor,” *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 1, pp. 5–14, Mar. 1993.

[27] L. C. Alaoui and A. Lebrun, “Feedback control of a permanent magnet stepper motor,” in *, Proceedings of the IEEE International Symposium on Industrial Electronics, 1997. ISIE ’97*, 1997, pp. 1006–1011 vol.3.

[28] I. Amidror, “Sub-Nyquist artefacts and sampling moiré effects,” *R. Soc. Open Sci.*, vol. 2, no. 3, Mar. 2015.

[29] G. B. Thurston, “Viscoelasticity of Human Blood,” *Biophys. J.*, vol. 12, no. 9, pp. 1205–1217, Sep. 1972.

[30] A. E. Koser, L. Pan, N. C. Keim, and P. E. Arratia, “Measuring material relaxation and creep recovery in a microfluidic device,” *Lab. Chip*, vol. 13, no. 10, p. 1850, 2013.

[31] R. Brummer, *Rheology Essentials of Cosmetic and Food Emulsions*. Berlin Heidelberg: Springer-Verlag, 2006.

Acknowledgements

I would like to thank my advisor Nathan Sniadecki for his guidance throughout this process; Nikita Taparia for working with me on this project, helping me generate design ideas and in running the experiments; Kevin Beussman, Andrea Leonard, Molly Mollica, and Ty Higashi for their advice on device improvements and helping me better understand this topic; and my committee members Nathan White and Jae-Hyun Chung for their support.

Appendix

A. MATLAB Codes for Image Analysis

Calculate average peak-to-peak amplitude for each post and plot vs sweep angle

```
clc, clear all, close all
px2um = 2.9299;
testnumber = 9;     %Number of test for each post
post = [1 2 3 4 5];     %Select which post

for i = 1:testnumber

    %Load file for each sweep angle
    file = uigetfile('*.*');
    load(file);

    %Normalize deflection around zero
    mean1 = mean(p1y(post(1),:))/px2um;
    mean2 = mean(p1y(post(2),:))/px2um;
    mean3 = mean(p1y(post(3),:))/px2um;
    mean4 = mean(p1y(post(4),:))/px2um;
    mean5 = mean(p1y(post(5),:))/px2um;

    movement = (smoothdata((p1y(post(1),:)./px2um)-mean1));
    movement2 = (smoothdata((p1y(post(2),:)./px2um)-mean2));
    movement3 = (smoothdata((p1y(post(3),:)./px2um)-mean3));
    movement4 = (smoothdata((p1y(post(4),:)./px2um)-mean4));
    movement5 = (smoothdata((p1y(post(5),:)./px2um)-mean5));

% Plot deflection waveform for a sweep angle
%   figure
%   plot(tvector,movement,'Linewidth',1.3)
%   hold on
%   plot(tvector,movement2,'Linewidth',1.3)

    % Find peaks and valleys
    pks = findpeaks(movement,'MinPeakProminence',0.02)';
    vlys = findpeaks(movement*-1,'MinPeakProminence',0.02)';
    pks2 = findpeaks(movement2,'MinPeakProminence',0.02)';
    vlys2 = findpeaks(movement2*-1,'MinPeakProminence',0.02)';
    pks3 = findpeaks(movement3,'MinPeakProminence',0.02)';
    vlys3 = findpeaks(movement3*-1,'MinPeakProminence',0.02)';
    pks4 = findpeaks(movement4,'MinPeakProminence',0.02)';
    vlys4 = findpeaks(movement4*-1,'MinPeakProminence',0.02)';
    pks5 = findpeaks(movement5,'MinPeakProminence',0.02)';
    vlys5 = findpeaks(movement5*-1,'MinPeakProminence',0.02)';

    %Remove edge peaks/valleys
```

```
    vlys = vlys*-1;
    pktopk = pks(2:length(pks)-2,:)-vlys(2:length(pks)-2,:);
    vlys2 = vlys2*-1;
    pktopk2 = pks2(2:length(pks2)-2,:)-vlys2(2:length(pks2)-2,:);
    vlys3 = vlys3*-1;
    pktopk3 = pks3(2:length(pks3)-2,:)-vlys3(2:length(pks3)-2,:);
    vlys4 = vlys4*-1;
    pktopk4 = pks4(2:length(pks4)-2,:)-vlys4(2:length(pks4)-2,:);
    vlys5 = vlys5*-1;
    pktopk5 = pks5(2:length(pks5)-2,:)-vlys5(2:length(pks5)-2,:);

    %Average deflection of peak for each sweep angle
    testavg(i,1) = mean(pktopk);
    testavg(i,2) = mean(pktopk2);
    testavg(i,3) = mean(pktopk3);
    testavg(i,4) = mean(pktopk4);
    testavg(i,5) = mean(pktopk5);
end

%Average of all posts
postavg = mean(testavg,2);

% Plot avg amplitude vs sweep angle
sweep = [1 2 4 8 12 16 20 22 24];
plot(sweep,testavg(:,1),'-o','Linewidth',1.3)
hold on
plot(sweep,testavg(:,2),'-o','Linewidth',1.3)
plot(sweep,testavg(:,3),'-o','Linewidth',1.3)
plot(sweep,testavg(:,4),'-o','Linewidth',1.3)
plot(sweep,testavg(:,5),'-o','Linewidth',1.3)
plot(sweep,postavg,'-ok','Linewidth',2)

 xlabel('Sweep angle (degrees)'), ylabel('Tip deflection (microns)')
 legend('Post1', 'Post2','Post3','Post4','Post5','Average')
 legend('location','northwest')
```

Calculate mean frequencies for post deflections and compare to input

```
clc, clear all, close all

post = [1 2 3 4 5];  %Select which post
postnumber = 3;
inputfreq = 1.40625; %Hz - Input frequency from motor
peaknumber = 11; %Number of peaks

%Load file for each sweep angle
file = uigetfile('*.*');
load(file);

for i = 1:postnumber
    movement(i,:) = smoothdata(p1y(post(i),:));
end

% figure
 plot(tvector,movement(1,:))
[pks,locs] = findpeaks(movement(1,:),tvector,'MinPeakProminence',.02);
[pks2,locs2] = findpeaks(movement(2,:),tvector,'MinPeakProminence',.02);
[pks3,locs3] = findpeaks(movement(3,:),tvector,'MinPeakProminence',0.02);
% [pks4,locs4] = findpeaks(movement(4,:),tvector,'MinPeakProminence',0.02);
% [pks5,locs5] = findpeaks(movement(5,:),tvector,'MinPeakProminence',0.02);


% Keep vectors same length
locs1 = locs(1:peaknumber);
locs2 = locs2(1:peaknumber);
locs3 = locs3(1:peaknumber);
% locs4 = locs4(1:peaknumber);
% locs5 = locs5(1:peaknumber);

locs = [locs1 locs2 locs3];

%Calculate periods
for i = 1:postnumber
    periods(i,:) = diff(locs(:,i));
end

%Convert to frequencies
outputfreq = 1./periods;

%Calculate mean freq and stdev
for i = 1:postnumber
    meanfreq(i) = mean(outputfreq(i,:));
    stdev(i) = std(outputfreq(i,:));
end

figure
allfreqs = [inputfreq meanfreq];
stdevs = [NaN stdev];
```

```
bar(allfreqs)
hold on
errorbar(allfreqs, stdevs,'k.','Linewidth',2)

ylim([0 2])
axisnames = {'Input','Post1','Post2','Post3','Post4','Post5'};
set(gca,'xticklabel', axisnames);
ylabel('Frequency (Hz)')
```

B. Arduino Codes for Stepper Motor Input

Sine curve for characterization with limit switch activation – Input sweep angle, delay

```
//simple A4988 connection
//jumper reset and sleep together
//connect  VDD to Arduino 3.3v or 5v
//connect  GND to Arduino GND (GND near VDD)
//connect  1A and 1B to stepper coil 1
//connect 2A and 2B to stepper coil 2
//connect VMOT to power source (9v battery + term)
//connect GRD to power source (9v battery - term)

//Set up integers
int stp = 8;      //connect pin 8 to step (step control)
int dir = 11;     // connect pin 11 to dir (direction control)
int a = 0;        //  step counter
int d = 10;        // delay counter
int val = 3;      // voltage tracking to pin 3 - connected to DAC and LabVIEW
int swtch = 2;   // limit switch (NO) connected to pin 12
int start = 50;  // steps to start position
int steps = 4;    // steps for given angle (10 deg ~ 44 steps)


//Select outputs/inputs
void setup()
{
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(val, OUTPUT);
  pinMode(swtch, INPUT_PULLUP);
  Serial.begin(9600);


//Homing and set motor zero position
  while (digitalRead(swtch))
  {  // Do this until the switch is activated
    digitalWrite(dir, LOW);      // (HIGH = anti-clockwise / LOW = clockwise)
    digitalWrite(stp, HIGH);
    delay(5);                        // Delay to slow down speed of Stepper
    digitalWrite(stp, LOW);
    delay(5);
}


  while (!digitalRead(swtch))
  { // Do this until the switch is not activated
    digitalWrite(dir, HIGH);
    digitalWrite(stp, HIGH);
    delay(10);                       // More delay to slow even more while
moving away from switch
    digitalWrite(stp, LOW);
    delay(10);
  }
```

```
  delay(500);

//Step to start position
digitalWrite(dir, HIGH);
  while (a<start)
  {
     a++;
     digitalWrite(stp, HIGH);
     delay(d);                    //speed control - delay between each
microstep
     digitalWrite(stp, LOW);
     delay(d);
  }
  delay(2000);
  a==0;
}



void loop()

{

  digitalWrite(dir, HIGH);
        if (a < steps )                 //sweep "a" steps in direction 1
        {  Serial.println(a);

          digitalWrite(val, HIGH);      //send high signal to DAC
corresponding to dir 1
            a++;
            digitalWrite(stp, HIGH);
            delay(d);                   //speed control - delay between each
microstep
            digitalWrite(stp, LOW);
            delay(d);                   //keep all four delays the same for
smooth steps
        }
        else if (a >=steps && a < 3*steps)     // sweep "a" steps in
direction 2
        {Serial.println(a);
          digitalWrite(val,LOW);         //send low signal to DAC
corresponding to dir 2
            digitalWrite(dir, HIGH);
            a++;
            digitalWrite(stp, HIGH);
            delay(d);
            digitalWrite(stp, LOW);
            delay(d);
        }
        else if (a>=3*steps && a<4*steps)       //sweep back to zero pt
        {Serial.println(a);
          digitalWrite(val,HIGH);
            digitalWrite(dir, LOW);
            a++;
            digitalWrite(stp, HIGH);
            delay(d);
            digitalWrite(stp, LOW);
```

```
            delay(d);
        }
        else
        {
          digitalWrite(val,LOW);
            a = 0;       //reset counter
        }
}
```

Creep function – Input sweep angle, hold time, delay

```
//Set up
int stp = 8;  //connect pin 8 to step (step control)
int dir = 11;  // connect pin 11 to dir (direction control)
int a = 0;     //  step counter
int d = 1;      // delay counter (ms)
int h = 500;   // hold
int c = 0;      //instance counter


//Select outputs
void setup()
{
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
  Serial.begin(9600);
  delay(15000);
}


void loop()

{
    digitalWrite(dir, LOW);

        if (a < 44 )  //sweep "a" steps in direction 1
        {
          a++;
          digitalWrite(stp, HIGH);
          delay(d);      //speed control - delay between each
microstep
          digitalWrite(stp, LOW);
          delay(d);    //keep all four delays the same for smooth steps
        }
        else if (a==44)
        {
          a++;
          c++;
          delay(h);

        }
        else if (a > 44 && a < 89)        //sweep "a" steps in direction 2
        {
          digitalWrite(dir, HIGH);
          a++;
          digitalWrite(stp, HIGH);
          delay(d);      //speed control - delay between each
microstep
          digitalWrite(stp, LOW);
          delay(d);    //keep all four delays the same for smooth steps
        }
        else
        {
          a = 0; //reset counter
          c++;
```

```
            delay(h);
          }

}
```

Ramp test - Input sweep angle, variable delay (3 selections)

```
//Set up integers
int stp = 8;  //connect pin 8 to step (step control)
int dir = 11;  // connect pin 11 to dir (direction control)
int a = 0;     //  step counter
int d = 200;      // delay counter (ms)
int d2 = 100;
int d3 = 50;
int h = 1;   // drop delay (fast)
int i = 0;  // Loop counter


//Select outputs
void setup()
{
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
}


void loop()

{
    digitalWrite(dir, LOW);
    if (i<1)
      {
        if (a < 66 )  //sweep "a" steps in direction 1
        {
          a++;
          digitalWrite(stp, HIGH);
          delay(d);         //speed    control    -    delay    between    each
microstep
          digitalWrite(stp, LOW);
          delay(d);    //keep all four delays the same for smooth steps
        }
        else if (a==66)
        {
          a++;
          delay(h);
        }
        else if (a > 66 && a < 133)        //sweep "a" steps in direction 2
        {
          digitalWrite(dir, HIGH);
          a++;
          digitalWrite(stp, HIGH);
          delay(h);         //speed    control    -    delay    between    each
microstep
          digitalWrite(stp, LOW);
          delay(h);    //keep all four delays the same for smooth steps
        }
        else
        {
          a = 0; //reset counter
          delay(h);
```

```
          i++;
        }
      }

      if (i<2)
      {
        if (a < 66 )  //sweep "a" steps in direction 1
        {
          a++;
          digitalWrite(stp, HIGH);
          delay(d2);         //speed    control    -    delay    between    each
microstep
          digitalWrite(stp, LOW);
          delay(d2);    //keep all four delays the same for smooth steps
        }
        else if (a==66)
        {
          a++;
          delay(h);
        }
        else if (a > 66 && a < 133)        //sweep "a" steps in direction 2
        {
          digitalWrite(dir, HIGH);
          a++;
          digitalWrite(stp, HIGH);
          delay(h);         //speed    control    -    delay    between    each
microstep
          digitalWrite(stp, LOW);
          delay(h);    //keep all four delays the same for smooth steps
        }
        else
        {
          a = 0; //reset counter
          delay(h);
          i++;
        }
      }

      if (i<3)
      {
        if (a < 66 )  //sweep "a" steps in direction 1
        {
          a++;
          digitalWrite(stp, HIGH);
          delay(d3);         //speed    control    -    delay    between    each
microstep
          digitalWrite(stp, LOW);
          delay(d3);    //keep all four delays the same for smooth steps
        }
        else if (a==66)
        {
          a++;
          delay(h);
        }
        else if (a > 66 && a < 133)        //sweep "a" steps in direction 2
        {
          digitalWrite(dir, HIGH);
```

```
          a++;
          digitalWrite(stp, HIGH);
          delay(h);         //speed    control    -    delay    between    each
microstep
          digitalWrite(stp, LOW);
          delay(h);    //keep all four delays the same for smooth steps
        }
        else
        {
          a = 0; //reset counter
          delay(h);
          i++;
        }
      }

      else
      {
        i=0;
      }

}
```