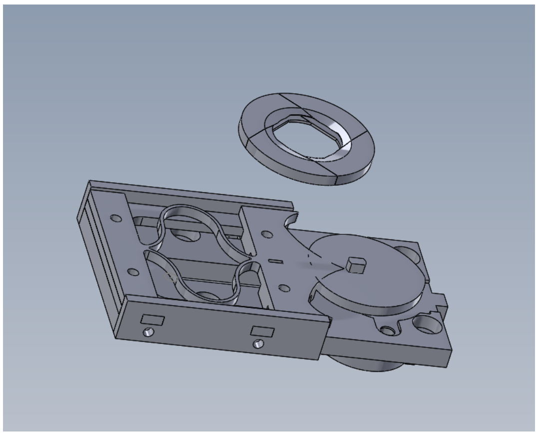
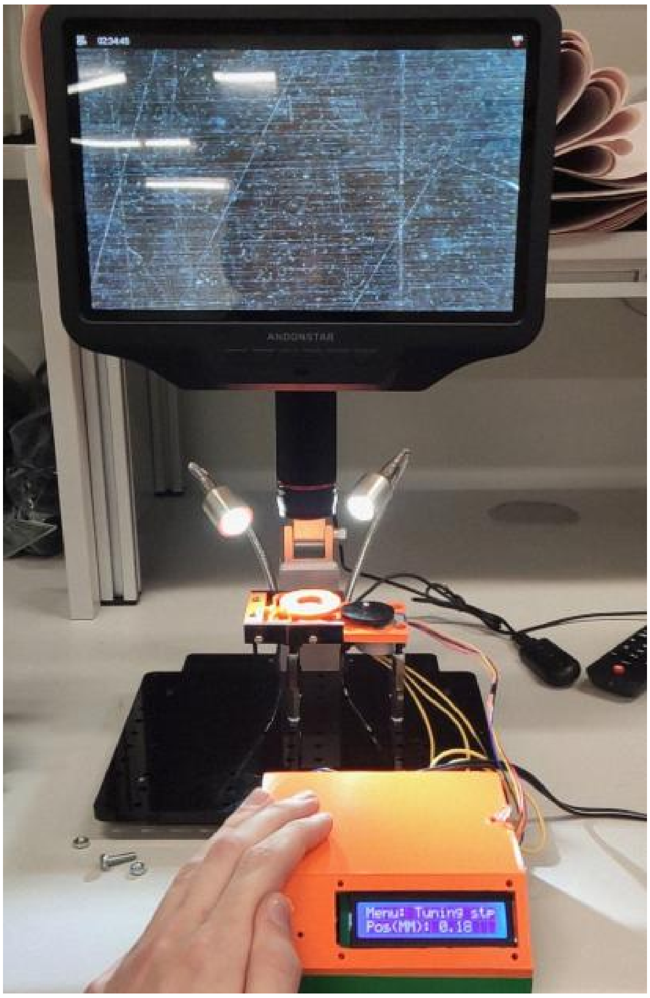
Compliant Microscopy stage - Final Report





I. Introduction

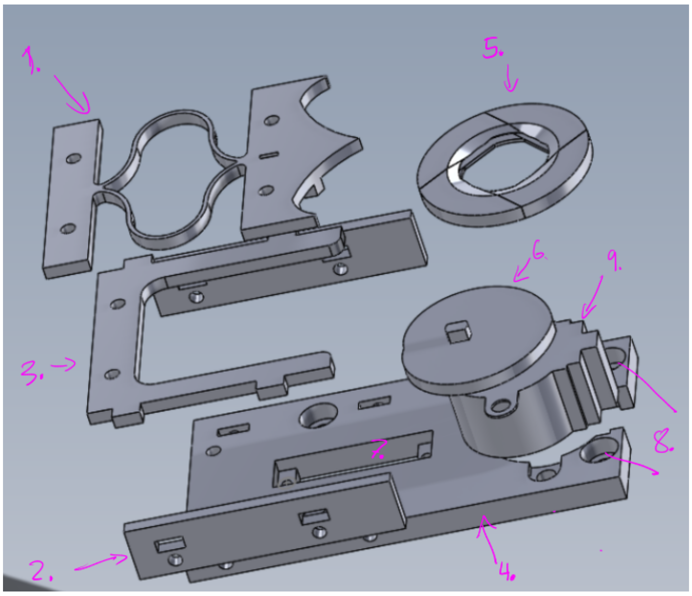
Our project's aim was to make a microscopy stage that is easy to manufacture. This is due to high end microscopy stages being prohibitively expensive and therefore providing an alternative that is possible to manufacture with hobby tools while retaining a certain degree of accuracy.

The microscopy stage was required to be 120x80x70mm and we specified it to 100x75x70mm, It was also required to have a compliant mechanism perform a critical role in the microscopy stage we specified it to be the sample stage. It was required to move the sample 7mm with 0.5mm steps with
The project consisted of:

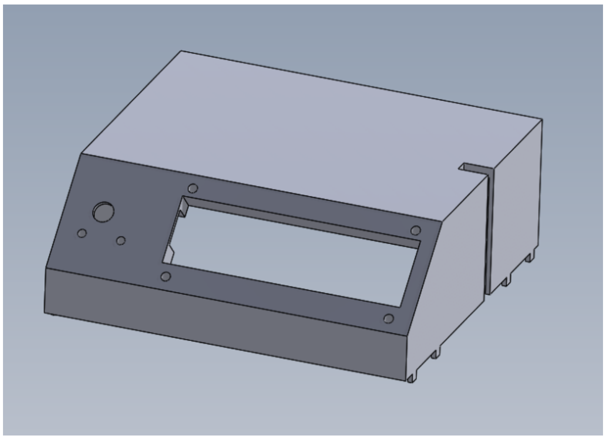
- 1. Designing a compliant mechanism
- 2. Designing a rotation to linear conversion mechanism
- 3. Designing a holder to hold sample onto the compliant mechanism
- 4. Designing an interface between the microscope and mechanism
- 5. Identify and program the needed control electronics
- 6. Design a UI between control software and user.

I have personally designed the UI and the compliant mechanism/rotary to linear converter in order to have a seamless transition into code. So I have done points 5 & 6 with strong influence on 1 & 2. Where I have worked on all the code and had a strong impact on design choices for parts integrating with electronics like the position of the sliding potentiometer and what electronics gets to be used. I was also in charge of manufacturing the design so I could make some adjustments to the design in order to rapidly prototype the initial group ideas but still left the decisions to Max who was in charge of creating the solidworks designs.

Design Overview
Files attached with same names(part name in brackets)



1. Compliant stage
2. guiding walls
3. Spacer/separator
4. Mounting frame
5. Sample holder (magnet holder)
6. Rotary to linear converter RTL (spiral V3)
7. Sliding potentiometer slot (bourns PTA2043-2015CPB103)
8. Mechanical breadboard interface (M4 standoffs and screws)
9. Stepper motor (28BYJ-48)



10. UI housing
11. Bottom panel (UIhousing_2)
12. LCD (2002A I2C display)
13. Rotary encoder

Needed:
2: 14x6x2mm magnets for the underside of the compliant stage
1: 32mm outer 20mm inner diameter and 1.6mm thick magnet
1: 2002A I2C display
1: stepper motor control board (attached SM_driver PCB and schematic)
1: rotary encoder with push button (KY-040 with breakout board for mounting)
At Least 200x200mm of 3mm acrylic
At least 100g of PETG spool

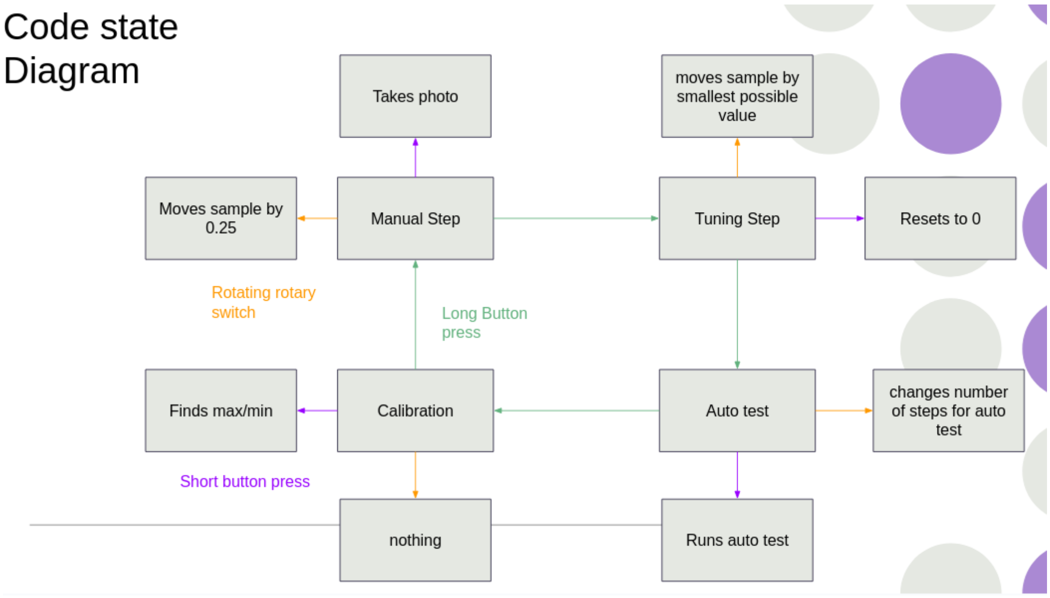
Physical:
Our Microscopy stage works by converting the change in angle to a linear change in radius up to a highest point of 8mm. The change in radius compresses the compliant stage and this in turn moves the sample. The distance the sample travels is the same as the change in radius. The compliant stage also has a cutout for the sliding potentiometer in order for the software to confirm that the sample has moved the distance that was set. The guiding walls are made to be adjustable so that the friction does not stall the stepper motor whilst keeping

the y error low. The spacer is needed in order to decrease the friction the compliant stage has with the mounting frame.

Electronics:
I have chosen a rotary switch and a 2x16 character LCD display for the UI and a sliding potentiometer for the motor feedback. The controller is an Arduino Uno. The code needs to be slightly adjusted to a previous iteration in order to work without the potentiometer but it is not a necessary part for its function. But it is recommended to have the potentiometer due to the much higher increase in accuracy as it acts like a guiding rail and decreases the necessity for the walls further decreasing friction and it also makes the code more adaptable for someones need as the steps can be adjusted to a users need.

UI:
The screen is up to the designer and the front panel should be adjusted for the screen used by the person. The screen, the user input, and their necessary mounting should fit within a 39x117mm footprint for the face of the UI box.

The UI consists of 4 menus:



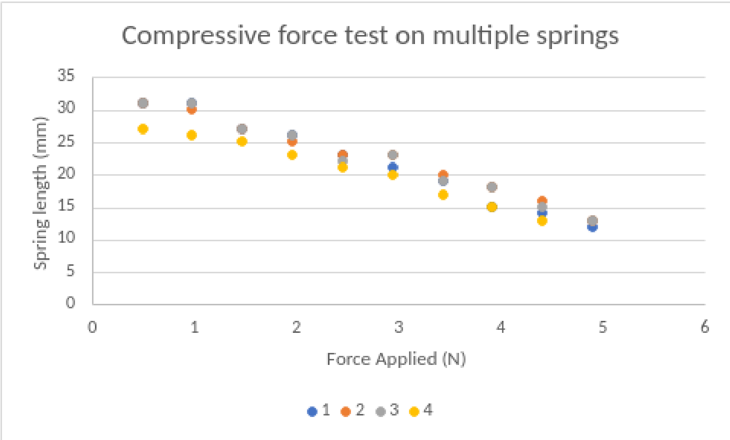
With the 4 inner boxes being the 4 menus.

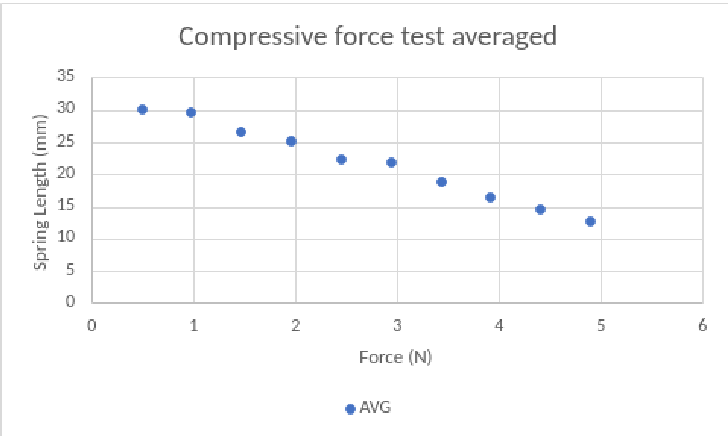
Testing

When testing we went through preliminary tests on seeing which shapes work and then we went onto iterating on a semi-final design into which we added the potentiometer slot and the 2nd part of the sample holding magnets. From the preliminary testing we learned that when the spring is compressed in the X dimension it expands in the Y dimension and if it is limited from doing such we get a non-linear spring constant.

Spring constant test

Compress the spring using 50 gram weights and measure the spring size. Multiplying the weight by 9.8 to get force, F= -kx. The gradient of the graphs below is the spring constant. The average spring constant is 0.276 N/m

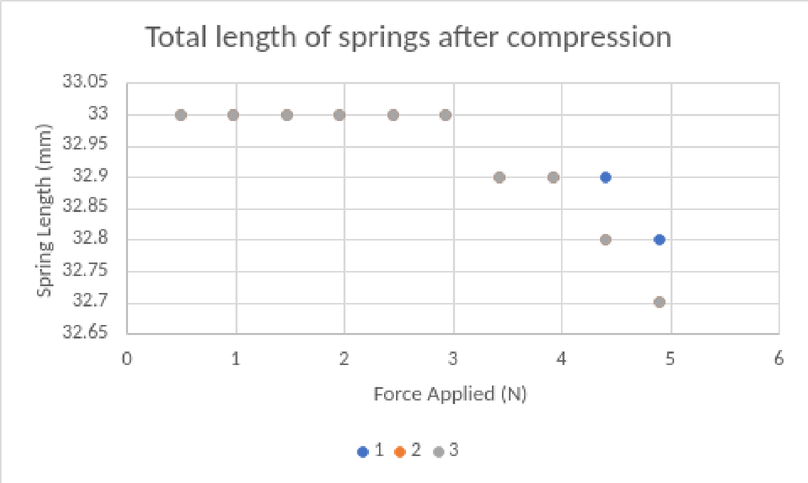




As the testing results show that although each spring might have a different spring constant they all have a linear spring constant.

Repeated compression test.

After performing the spring constant test we compressed the spring with increasing weights. Then we took the weights off and measured its length and recorded it. When combined with the previous test we can see that 8mm is not within the deformation area. But when its assembled the spring is compressed by the inner radius of the RTL. Therefore the values working range does go into the permanent deformation region. Hence the spring is designed to be replaceable.



We ran out of time to perform the accuracy test using the microscope and an image processing software to check for our design going out of focus and checking the error of individual steps. Although we personally have not performed the tests for accuracy of return and movement range, We have the result data from our project being tested by Gideon Gouws and Hamish Colenso.

	Start Position (pxls)	End Position (pxls)	Return to Start Position (pxls)	Forward Displacement (mm)	Error in Return Position (mm)
x	1784	180	1784	**7.712**	**0.000**
y	536	520	536	**-0.077**	**0.000**

Issues with the design and future improvements.

Currently when the spring is returning it performs a jittering movement which doesn't happen when performing a single quick return, but when performing a single return you aren't guaranteed to return to the same place and it can shift the sample due to momentum.

Another issue is the backlash when rotating in the opposite direction this backlash appears from the motor and the potentiometer hence the accuracy is decreased when at the edges of movement and reversing the direction of the motor. This can be helped with by only turning in one direction and returning to 0 after passing the 8mm mark.

Rewriting the code in C will allow for a higher degree of control and the implementation of a more intuitive UI that implements interrupts. As when the device is powered it has to recalibrate during which it ignores user input which is bad UX design.

Developing a sample holder that doesn't require magnets is a possible future consideration as this would majorly cut the price of production. Another cut to the price would be creating a potentiometer that doesn’t need to be bought and made out of readily available materials.