Fall 2021 Reflection

Over the past semester, I have accomplished the following:

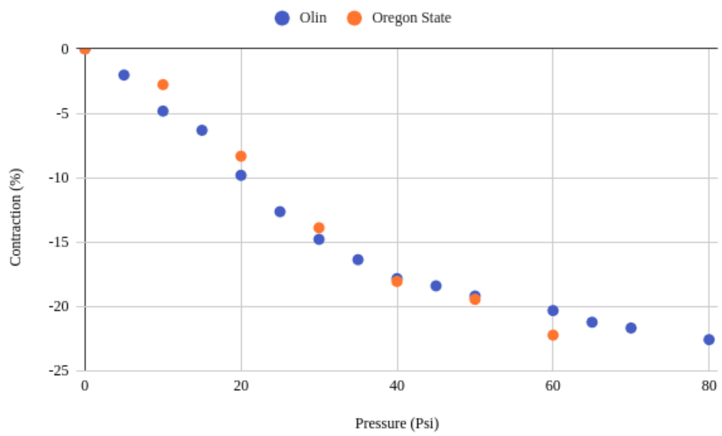
- Constructed and characterized four pneumatic "McKibben" muscles for use in constructing various soft robots.
- Designed and fabricated multiple muscle mounting parts to arrange pneumatics muscles into a wide array of different soft robot designs.
- Set up ROS motion capture packages to measure the shape changes of the soft robots. Experimentally obtained motion capture data using these motion capture packages
- Created scripts to automate the process of data collection and processing
- Implement geometric curve-fitter to extract curvature from sparse motion capture data by utilizing the underlying structure of the soft robots.
- Created and presented a poster providing an overview of the work to an undergraduate research symposium

At the end, there are also some remarks on personal learnings and retrospective.

Table of Contents

Muscles

A singular inflated muscle

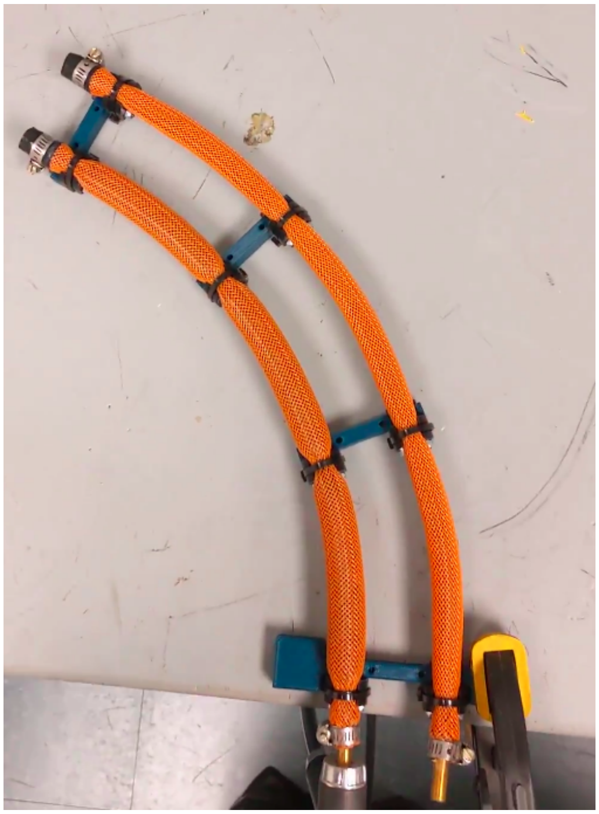


Muscle contraction vs pressure - confirms that the muscles constructed here are equivalent proportionally to the muscles constructed at Oregon State

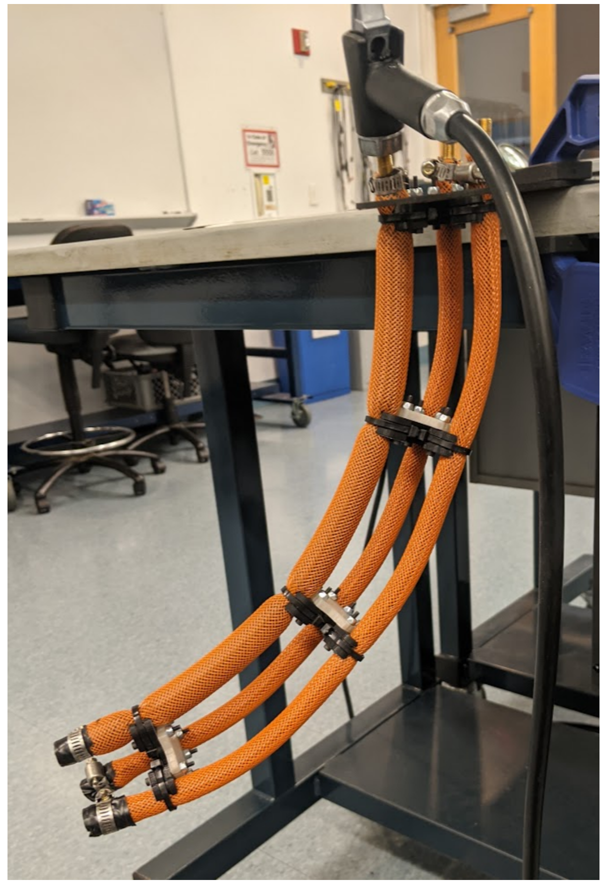
The first thing I did this semester was constructing a four separate McKibben muscle pneumatic actuators. The construction involved a simple rubber tube with a weaved nylon sleeving around it. Because of the weaved pattern in the sleeving, if the ends of the tube are clamped then inflating the tube will cause the muscle to contract, much like a “Chinese Finger Trap”.

After constructing the muscles, I had to characterize them to input into the model. Specifically, since the model takes in individual muscle lengths as its input parameters, I had to characterize how the muscles change length in proportion to its total length, according to input pressure changes. After measuring the muscle’s length at 15 different pressures, I arrived at the curve shown above, which I then applied a polynomial fit to within the matlab curve fitting toolbox (not shown here). From this experimental muscle characterization data, we can see that the the muscles I constructed at Olin, despite being half the length of the ones at Oregon State, contract the same amount proportionally to its total length.

Assembled Soft Robots



“Planar” arm - 2 mucsles, motion restricted to within a plane.



“Spatial” arm: A planar arm with an additional muscle is now able to bend in three dimensions

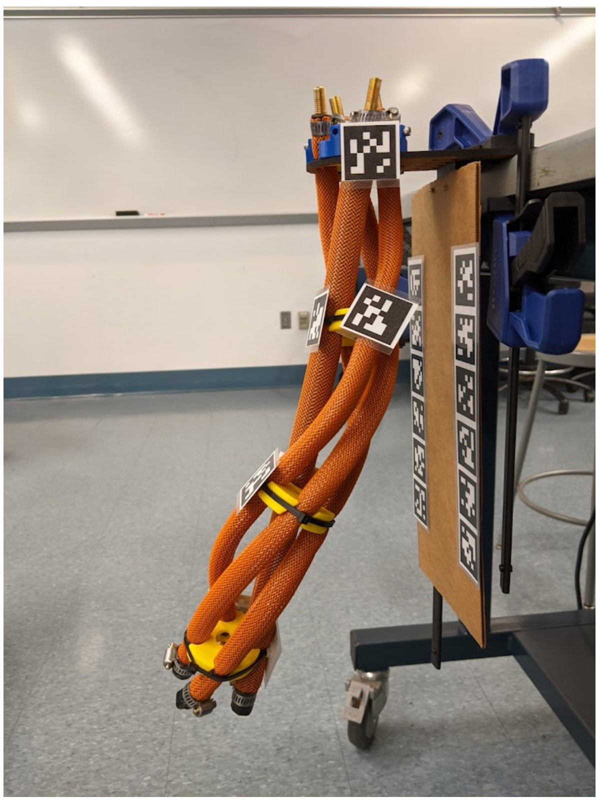
By twisting the mounting positions for the muscles we can create a helical muscle structure

After constructing the muscles, I could then go on to create an array of different soft-robot structures. Shown above are a 2D “planar” arm, a 3D “spatial” arm, and finally a helical arm. Not pictured here were additional wider “planar” arm designs, and ones that incorporated more muscles.

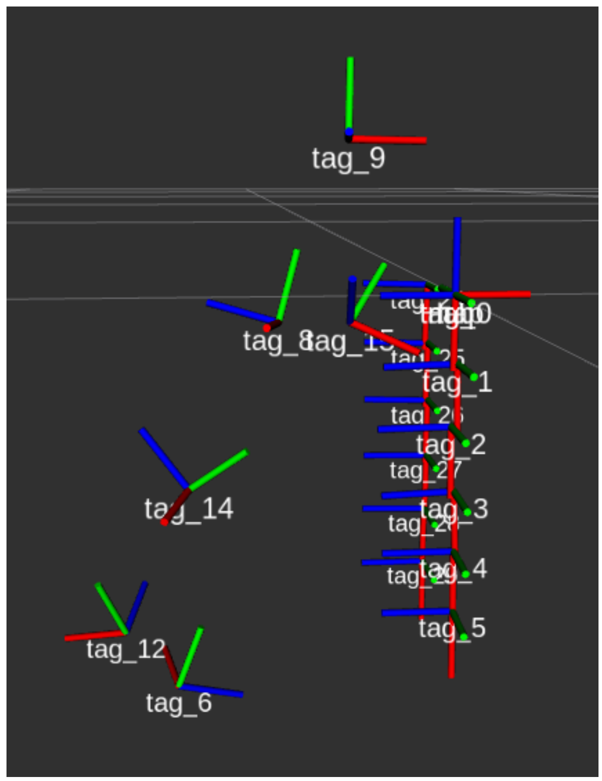
Much like with biological muscles, we can achieve a wide range of different bending motions simply by varying how the individual muscles are constrained to each other. Thus, despite only constructing four total individual muscles, by varying the design of the separator mounts connecting between the muscles we can completely change how the arms actuate.

Now that I had constructed the arms, I can then begin to look into capturing their geometry.

ROS Motion Capture

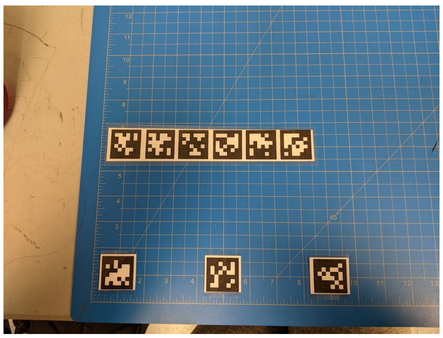


Actuated helical arm attached to experiment jig

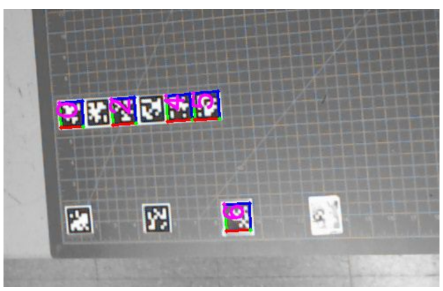


Corresponding captured transformation frames

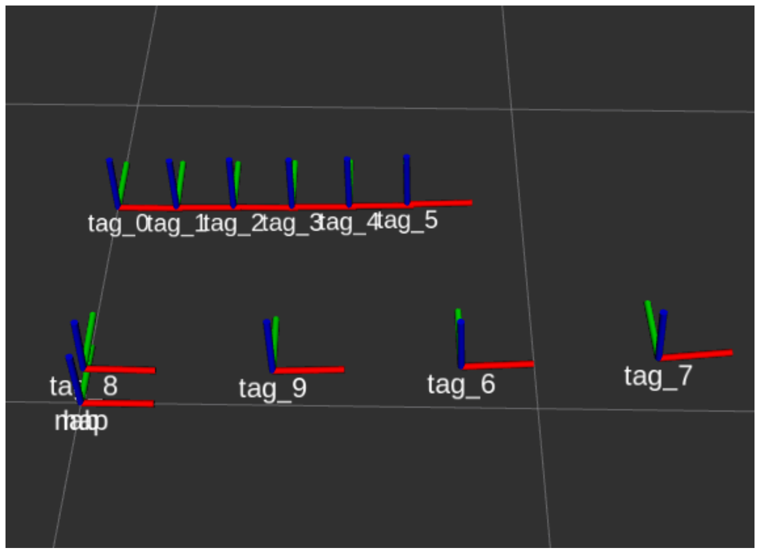
Now that we have constructed the robot arms, we would like to quantitatively capture the shape of the arms so that we can actually compare against our proposed model. Specifically, the model calculates not just positions but entire 6-DoF *poses* (position and orientation) of frames-of-reference travelling along the curve. Thus, both due to our lack of access to a sophisticated off-the-shelf motion capture system, and as a means to capture these poses, we attached Apriltag fiducial markers along the length of the robot arms. We then record footage of a (callibrated) webcam moving around the arm, and process this footage with TagSLAM to arrive at a final mapping of all of the attached fiducials.



Raw image of benchmark test



Example of tag detection. While not all tags are detected here they are all detected at some point over the footage



Tag poses estimated by TagSLAM based on the camera footage.

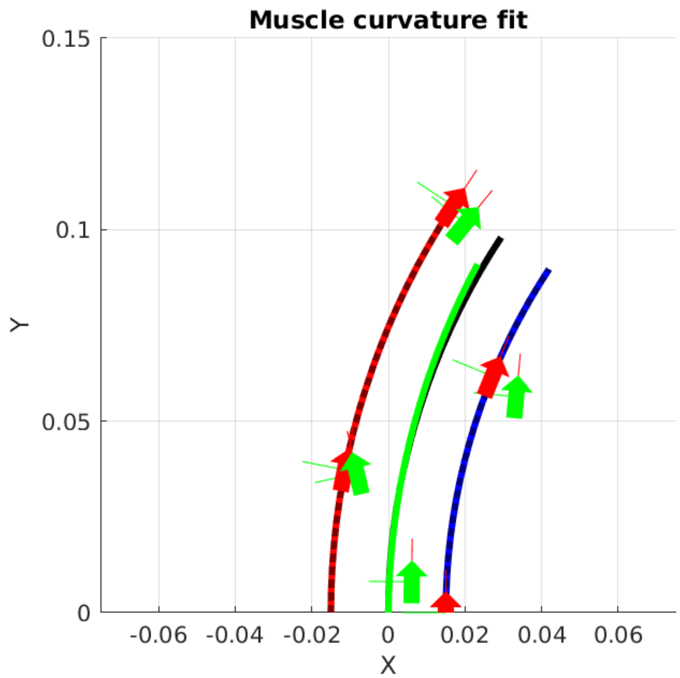
We can see in the above “benchmark” test the workflow of capturing camera footage, performing Apriltag detection, and then finally using TagSLAM to transform these detection into transformation frame poses. We can see that TagSLAM is indeed capable of accurately reconstructing an arrangement of tags.

Scripting

Created a suite of `roslaunch` and bash scripts so that, data capture and labelling each only takes a single command. Details can be found in the Github repository. Speficially, data collection and processing scripts can be found under `/launch` and `/scripts`.

Curve Fitting

An example planar arm with attached Apriltags



Example of simulated curve fitting. Red arrows represent the poses a tag *should* have, and Green arrows represent noisy measurements of those poses. The green curved line represents the fitted curvature, while the black curved line is the actual curvature.

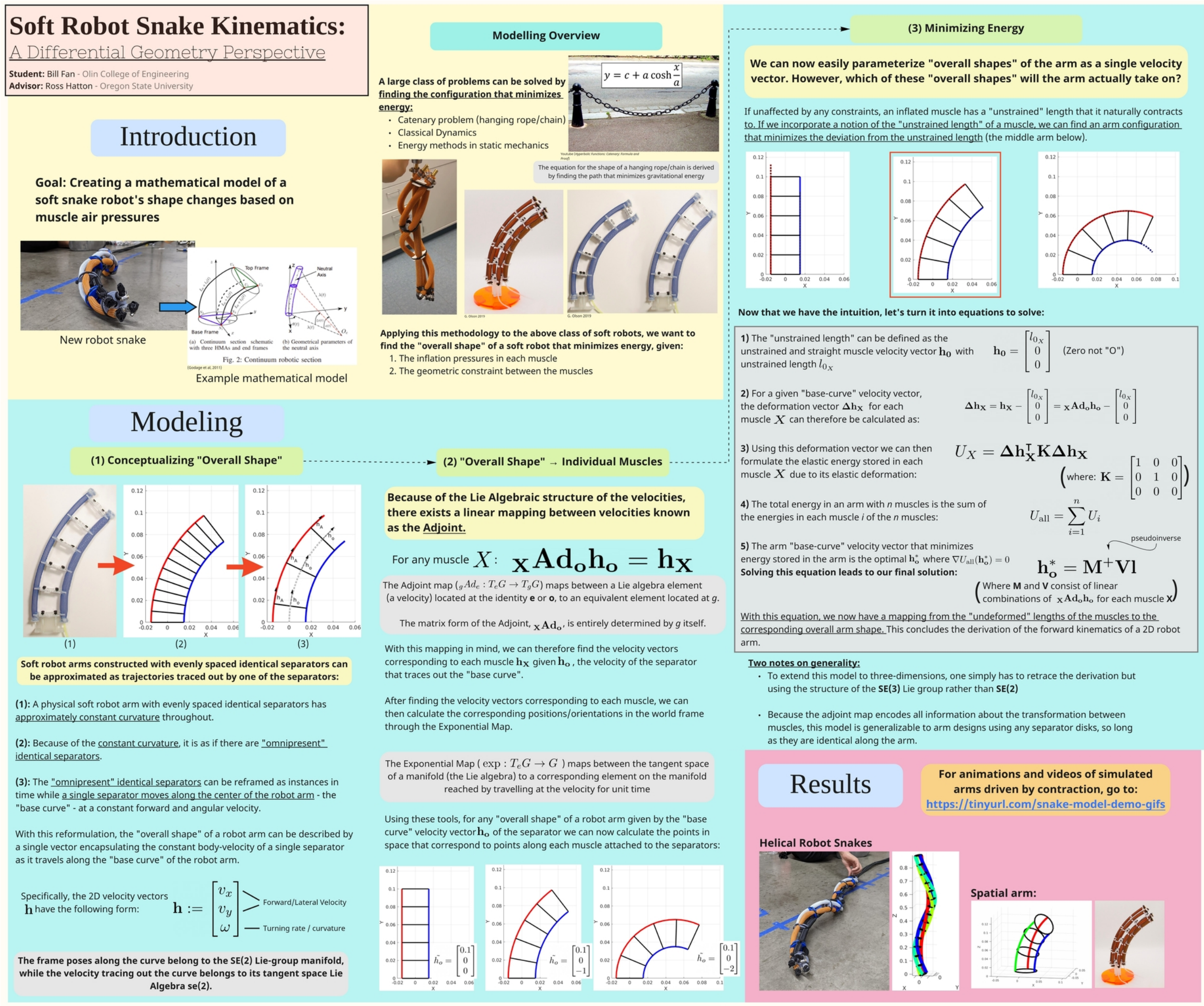
Traditionally with curve-fitting, a large amount of datapoints are needed to create an accurate and robust fit. Here, however, the problem is a bit different: while we only have four measurement points along the length of the robot arm, each of those points contains both a position and orientation. Additionally, if we know *a priori* where they are attached onto the robot arm, for any given arm curvature we will then know where the tags *should* end up. Thus, utilizing the known geometry of the arm, we can then transform the problem into finding the overall arm curvature that minimizes the discrepancy between where the tags *should be*, given this curvature, and where the tags *actually are.* This problem is elegantly expressed in the language of transformation matrices and differential geometry for robotics:

$$\min_{h_o} \sum_{i=0}^{n} \log(\exp(h_o) g_{oi} g_i^{-1})$$

Where g_i denotes the transformation matrix that represents the tag's measured position, while $\exp(h_o)g_{oi}$ is the "supposed" tag position. Multiplying by the inverse of a transformation gives the corresponding transformation between the two - the displacement / discrepancy between the "should be" and actual position of each tag. Finally, the logarithm map (matrix logarithm) reduces the transformation to the Lie Algebra - critical for when extending the framework to three dimensions.

This optimization process was implemented in Matlab, and numerically solved using `fminsearch()`. The result can be seen in the figure to the right. However, unfortunately I did not have enough time to fully implement rosbag file reading to apply this curve fitting to my measured experiment data. This will be happening within the first weeks of winter break.

Research Symposium



As suggested by an Olin alumn, I applied to and was accepted to present at the Undergraduate Math Symposium event hosted by the University of Illinois at Chicago. The event was held virtually as a poster session on Gather Town. For this event, I had to create a poster that showcased my work, including some of the math behind it. The poster was created hastily over the course of three days, so it definitely does not represent my best work, but it is useful as a presentation tool.

Personal Learning

At the start of the semester, I began this research project as a continuation of my work at Oregon State University this past summer. At the time, I set out with the following schedule and list of goals/deliverables:

9/20 - Physical model parts ordered

9/27 - Muscle assembly and characterization

10/06 - Planar arm model assembeled

10/20 - Planar arm dataset collected

11/03 - Spatial arm model assembly & dataset collection

11/10 - Helical arm model assembly & dataset collection

11/24 - Model comparison metrics and results

12/16 - Paper rough draft

Now that we are at the end of the semester, I can see that while I was able to somewhat follow the schedule for the first half of the semester, finishing collection of the raw footage for a Planar Arm dataset by late October. However, there were a couple things that happened afterwards that threw me significantly off schedule:

- It took me a not-insignificant amount of time to setup and use TagSLAM and write automation scripts.
- While I had collected a dataset on-time, the experiment setup and muscle attachement hardware was poorly designed and and not very robust. The robot arm had to be held down by my foot while I inflated and deflated it. Redesigning and fabricating all the parts ended up taking a significant amount of time.
- Finally, to present at the symposium I ended up dedicating an entire week's worth of work time to creating the poster.

Part of the learning experience with this project is the fact that this is really the first time I've had to work on a hardware-focused project in over two years. It took a while to not only get used to having to do a significant amount of CAD and fabrication work, but to also relearn how to get over the activation energy of starting to work on the hardware. Initially, the thought of doing CAD work filled me with dread, which lead to me finishing the software work first, while leaving the bulk of my design and fabrication work for the last three weeks of the semester.

Overall, over this project I've definitely sharpened a lot more of my hardware skills and ability to manage hardware projects, while also keeping my software, robotics math, and ROS skills fresh.

Next Steps

Over break I plan to finish the rosbag to matlab file reading so that I finish creating a quantitative comparison between my model's estimated poses and the measured real one.