SoftNote: Resistive stretch sensing to assist instrument learning for people with visual impairments

Catherine Tianhong Yu: tianhony@andrew.cmu.edu
Jiaqi Liu: jiaqil3@andrew.cmu.edu
Yunqi Willa Yang: yunqiy@andrew.cmu.edu

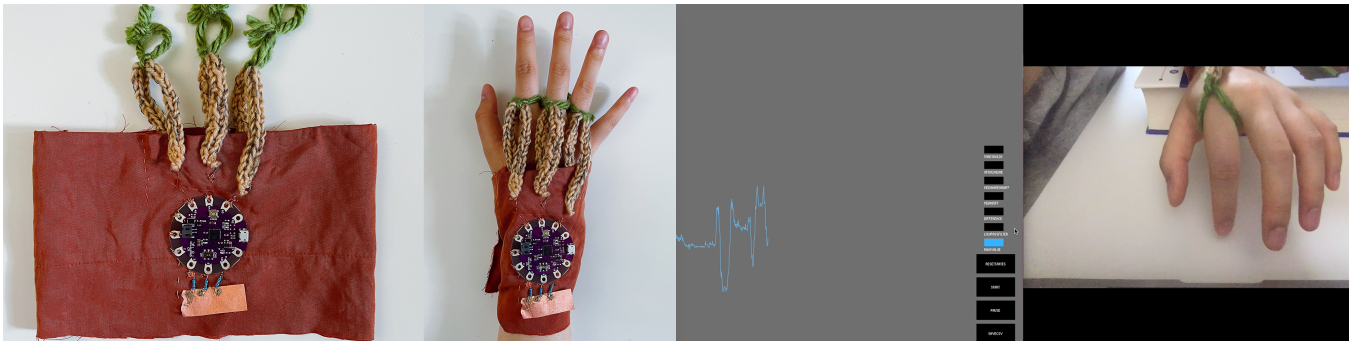

Figure 1: Left: final prototype; Middle: prototype in use; Right: Signal Visualization with one finger down.

1 INTRODUCTION

Advances in e-textiles and wearable technologies have enabled on-skin sensing and computing. In this project we explore gesture detection in connection to resistive stretch sensing. Specifically, our project is motivated by the piano learning experience for people with visual impairments as it is hard for them to keep track of fingerings when practicing. Beyond piano learning, our approach also fits into the larger picture of hand gesture detection as it detects finger tapping and movements. The video demo can be found at https://youtu.be/xu_MilIgwIM.

1.1 Materials

Materials we used include Arduino Lilypad, conductive yarn for sensor fabrication, conductive fabric and threads for knitting and connecting components, resistors, and non-conductive yarn and fabric.

1.2 Hardware system design

Figure 2 below shows our system schematic. To measure the resistance value, we add R1s so that the analog pins can use the voltage divider to calculate sensor values.

Given that stretching conductive yarns will lead to a decrease in their resistance, there needs to be a protective resistor R_{buff} in our schematic in series with the yarn. We also learned that by making the sum of R_{yarn} and R_{buff} closer to R_1, we could make the signal less noisy. Right now the stretch sensor's resistance is about 50ohm when it's not stretched. Therefore we chose 47 ohm for R_{buff} and 100 ohm for R_1 such that 50 + 47 is about 100.

We initially did not add the protective R_{buff} and chose 10 ohm for R_1, which gave us clean signal but unfortunately burned down our Lilypad as the resistance is too small.

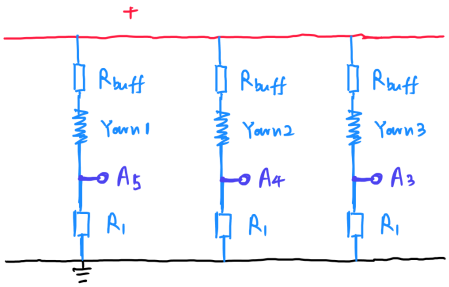

Figure 2: System schematic

1.3 Sensor fabrication

When it comes to the fabrication process for each sensor, we got close to the optimal solution for us during multiple experiments. The key stages for it are choices of yarns, exploration of sensor making approaches and sewing on the limited space. The following presents the road blocks we encountered during the sensor fabrication process, and they shaped our current sensor fabrication solution.

1.3.1 Choices of yarns. The behavior of yarns and our access to them both influence our choices of yarns. First, we select conductive yarns instead of resistive yarns. Among the types of yarn we have access to, the latter one provides a wider range of readings, but they showed irregular changes as it is stretched, i.e. even if the yarn is at rest, the resistance changes during our experiments. On the contrary, the conductive yarn presented more consistent readings and it is sensitive enough for our purpose, even with a small range of change. So we moved forward with conductive yarns, which are shown as #3 and #5 in Figure 3. Furthermore, we chose #5 for our final sensor demo. #3 showed cleaner data than #5 which is

Catherine Tianhong Yu: tianhony@andrew.cmu.edu
Jiaqi Liu: jiaqil3@andrew.cmu.edu
Yunqi Willa Yang: yunqiy@andrew.cmu.edu

Figure 3: Conductive yarns we have access to: 1. Thick blend of polyester (80%) and AISI 316L stainless steel (20%). 2. Thin, 2/50 Nm, 20% Stainless Steel 80% Polyester. 3. Stainless Steel conductive yarn (not sure about the precise compositions). 4. 1000D/2 Short carbon fiber resistive yarn (not sure about the precise compositions). 5. Adafruit Thin Conductive Yarn, 100% 316L stainless steel fiber by spinning and weaving. 6. Adafruit Stainless Thin Conductive Thread, 316L stainless steel.

available to our team member in China, but two of us who are making the sensors only have access to #5.

Besides various conductive yarns, we also experimented with different normal yarns, while considering stretchability and thickness for each of them. The stretchability sets the range of resistance changes, and whether the changes are enough to provide available signals. We should also consider the thickness of yarns, as our toy knitting machine can only knit relatively thin yarn, like the rightmost one of Figure 4. It's also what we selected for our sensor demo.

1.3.2 Exploration of sensor making approaches. To build the sensor with appropriate stretchability for our finger movements and make it relatively reproducible, we tried multiple combinations of conductive yarns and normal yarns. There are two major approaches standing out. The first one is using knots with thick yarn, which is shown as #1 in left picture of Figure 5. The knot shows visible resistance changes with only 60cm conductive yarns per sensor, as it uses two strands per row. The approach saves materials for our limited resources. However, it turns out that the total resistance might be too low, and it have even burnt out our first Lilypad.

Then we tried a new approach using longer conductive yarns, which means we should turn to knitting fabrication method because it can knit longer conductive yarns into our sensors. Using

Figure 4: Normal yarns we have access to: 1. 100% acrylic chunky yarn. 2. 100% acrylic thin yarn.

Figure 5: Left: different fabrication methods (1. knotting; 2. knitting); Right: our toy knitting machine.

Figure 6: Multiple fabric layers for sewing on the limited space.

knitting machine also makes sensors less difficult to duplicate. The right picture of Figure 5 shows the toy knitting machine we experimented with. We finally adopted the second method, and used 200cm conductive yarn per sensor whose resistance value was 50ohm.

1.3.3 Sewing on the limited space. SoftNote is designed to be a lightweight fabric on wrists, where we are actually limited by the space there for sewing yarns in intricate schematics. It took a lot of efforts for us to avoid conductive touching each other. Our solution

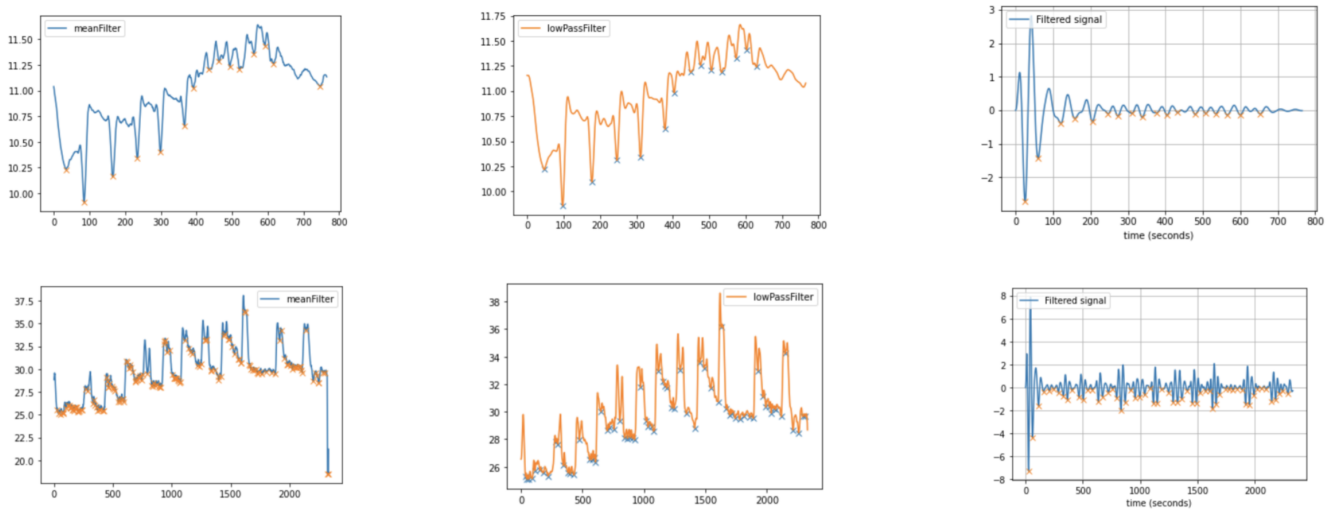

Figure 7: Three different filters applied to two sets of data. Top: data recorded from a knitted sensor using conductive yarn with cleaner data; Bottom: data recorded from a knitted sensor using conductive yarn with noisier data. Left: median filter with the same parameters; Middle: low-pass filter with the same parameters; Right: band-pass filter with the same parameters.

behind it was adding in 3 extra fabrics as layers and sewing branches of our schematic on each of them. Figure 6 zooms in on these fabrics.

1.4 Signal processing

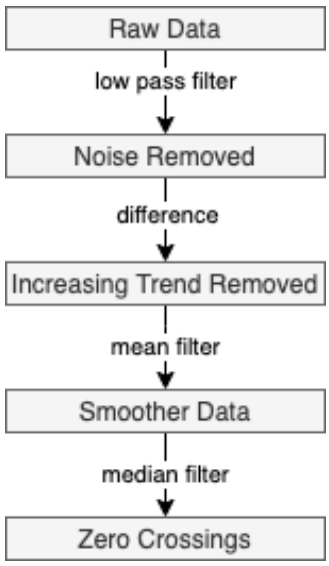

Figure 8: Signal processing pipeline for stretch detection.

1.4.1 Filter choices. Initially, we recorded some data, shown in the top row in Figure 7, using a sensor made using conductive yarn with cleaner data and tuned the filter parameters for that set of data. Then we applied the same filters with tuned parameters to another set of data, shown in the bottom row in Figure 7, using a sensor made using conductive yarn with noisier data, which is the one we are using for our final prototype. At the time, we only had one sensor fabricated and we were hoping to find filters that can easily generalize because our handmade sensors don't have identical behaviors. Mean filter and low pass filter performed significantly better than the band pass filter. We ended up using low pass filter because of the high frequency noises in our data.

1.4.2 Pipeline. In this section, we will discuss the signal processing pipeline(Figure 8).

In figures 9-15, they are screenshots taken from our visualization interface. In each of the screenshot, the gesture is a longer tap followed by three more rapid taps. All of the screenshots are taken one after another. In other words, the first screenshot is the first long tap followed by three rapid taps, the second screenshot is the next first long tap followed by three rapid taps. All of the screenshots are

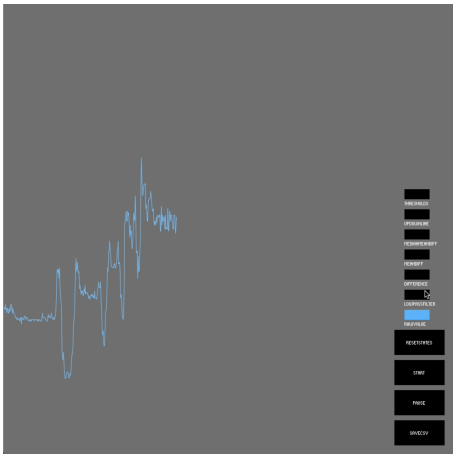

Figure 9: Light blue curve: raw data.

Catherine Tianhong Yu: tianhony@andrew.cmu.edu
Jiaqi Liu: jiaqil3@andrew.cmu.edu
Yunqi Willa Yang: yunqiy@andrew.cmu.edu

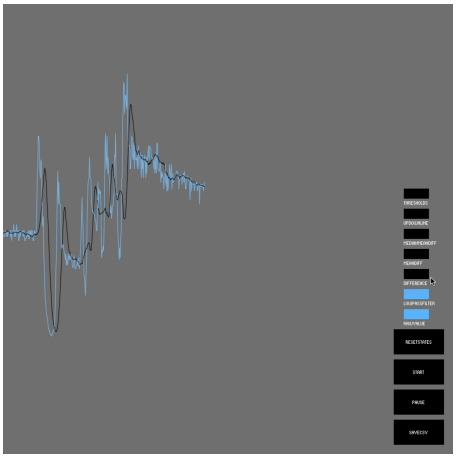

Figure 10: Light blue curve: raw data; black curve: low-pass filter applied.

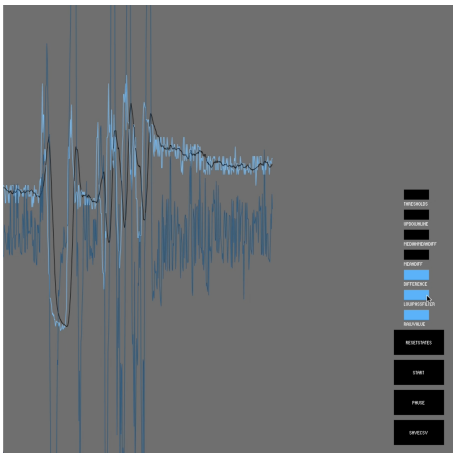

Figure 11: Light blue curve: raw data; black curve: low-pass filter applied; dark blue: differences.

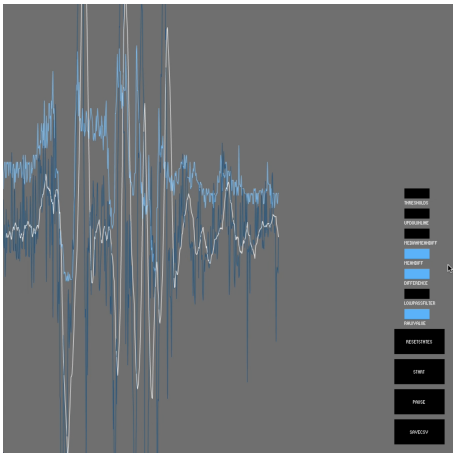

Figure 12: Light blue curve: raw data; dark blue: differences; white curve: mean filter applied to the differences.

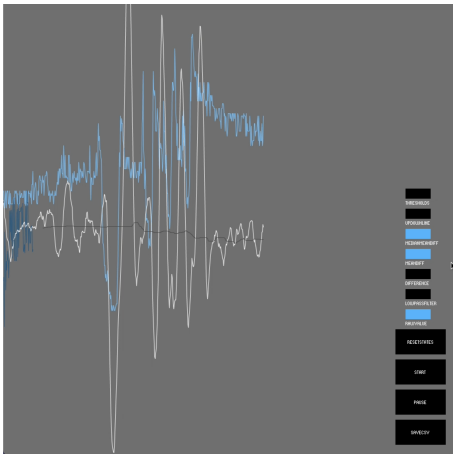

Figure 13: Light blue curve: raw data; white curve: mean filter applied to differences; dark grey curve: median filter applied to the means.

taken in in the same use of a sensor to capture raw value changes over time within a single use.

Firstly in figure 9, there is the raw data, which is the resistance measured using voltage divider. The sensor resistance decreases as it is stretched, and increases as it is relaxed. We observed that the sensor value does not return to its start value after we return to the start position, but the trend of stretched and restored is quite obvious.

In Figure 10, low pass filter is applied to clean up the high frequency noise as explained in the *filter choices* section.

Recommended by the professors, differences(using `numpy.diff()`) between neighboring values are calculated so that the data is not affected by the increasing resistance as the sensor is being used. This step is very important because sensor value does not return to a consistent value and calculating the difference preserves the stretched v.s. restored changes but eliminates the increasing trend.

In Figure 11, although the light blue curve has a increasing trend, the dark blue curve does not.

Then mean filter is applied smooth the calculated differences shown in Figure 12.

Then we applied the median filter for zero crossing in Figure 13.

Using calibrated thresholds visualized as the horizontal lines in Figure 14, which we will discuss further in details in the *calibration* section, stretchedrestored can be detected. Stretched detection is visualized by the black vertical line, and restoration is visualized by the white vertical line. This is the end of our signal processing pipeline.

Finally let's look at the raw data with where stretch and restoration detection are made in Figure x. Also note how much the raw resistance has increased compared to the first screenshot(Figure **??**).

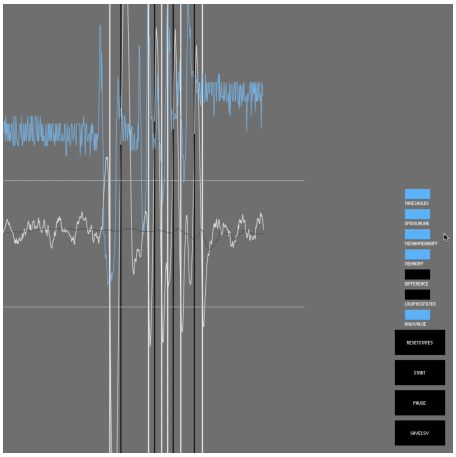

Figure 14: Light blue curve: raw data; white curve: mean filter applied to differences; dark grey curve: median filter applied to the means; white horizontal lines: thresholds; white vertical lines: indicators for stretched; white vertical lines: indicators for restored.

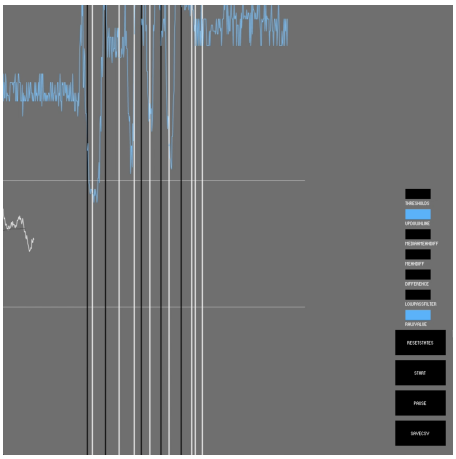

Figure 15: Light blue curve: raw data; white curve: mean filter applied to differences; white horizontal lines: thresholds; white vertical lines: indicators for stretched; white vertical lines: indicators for restored.

1.4.3 Calibration. Before we start detecting with the sensors, we need to first calibrate the sensors(Figure 16). We ask the user to tap 5 times with each finger to find the thresholds using the 5 peaks detected. Calibration is needed before every single use because sensor initial resistance is not predictable. We suspect this behavior could caused by the the following reasons fatigue in the knitted sensor as it becomes looser after it's used.

A interesting observation that we don't have a clear explanation for is that after the sensor is stretched after a use, when we used it later, the initial resistance values are not the same as the earlier resistance values at the end of the last use. Therefore, instead of predicting what the resistance value ranges are for each use, we prefer calibrations before each use.

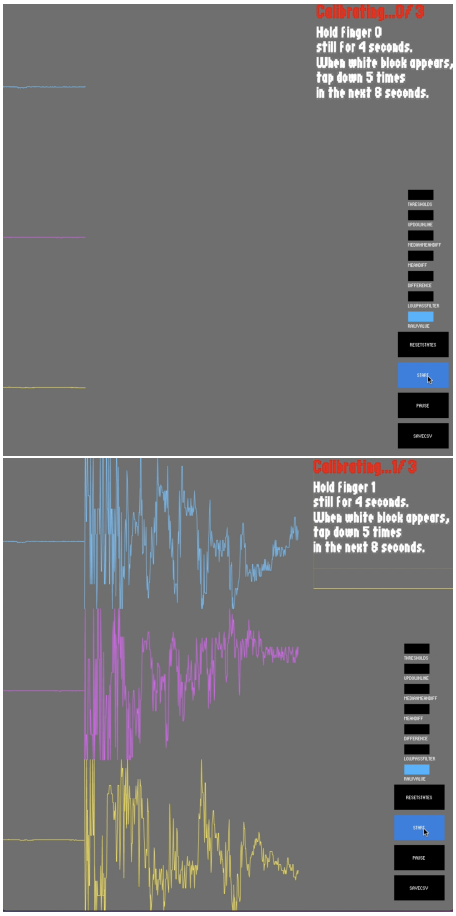

Figure 16: Calibration process for each sensor: rest for 5 seconds(top), tap 5 times within 8 seconds(bottom).

2 PROTOTYPE RESULTS

(1) Hardware: SoftNote fabrics with 3 sensors and one Lilypad, shown left in Figure 1
(2) Functionality: indicating whether a figure moves by collecting data for resistance changes of the attached sensor. For example, when the ring finger taps, the blue light is activated, as is shown in Figure 17.
(3) Code: Github repository https://github.com/willa-yunqiy/05499.

3 LIMITATIONS

The greatest limitation was that data is very noisy because conductive yarn is very noisy. Again we had access to different conductive yarns, the accuracy improves a lot when used with the conductive yarn that is cleaner. We read that using resistive yarn instead of conductive yarn gives a larger range for data change which

Catherine Tianhong Yu: tianhony@andrew.cmu.edu
Jiaqi Liu: jiaqil3@andrew.cmu.edu
Yunqi Willa Yang: yunqiy@andrew.cmu.edu

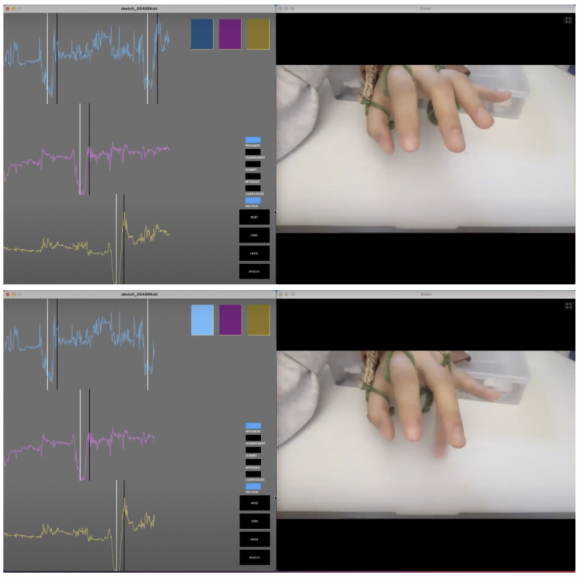

Figure 17: The blue light is activated when the ring finger taps. (Blue light - Forth finger, Red light - index finger, Yellow light - thumb).

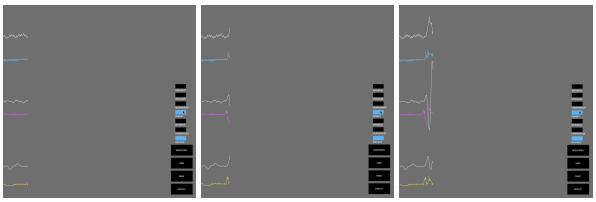

Figure 18: The moving finger is visualized as the pink curve, but all other fingers show significant changes as well even though those fingers are 'not moving'.

could improve accuracy[1], but unfortunately, we were not able to purchase a resistive yarn that was not extremely noisy.

Another limitation is finger movement dependency. In Figure 18, we can see that when only one finger(visualized as the pink curve) moves, data for other fingers also change significantly. This is because natural muscle movements dependency. It's hard to keep the other fingers completely quiet when one finger is moving. Noisy data is another factor here. There might also be other factors contributing to the finger movement dependency like sensor placements.

4 FUTURE WORKS

4.1 DIY Low-Cost Gesture Recognition

While our current demo shows a proof of our ideas and applicable results, we could further refine it for better usability and higher accuracy for gesture recognition.

- Communication: To make the sensor more user friendly and practical in multiple scenarios, a bluetooth module could be added to make it wireless.
- Output: The output signals can also be integrated with audio feedback, which could benefit people with visual impairment, or we could provide API interfaces for gesture triggering.
- Sensor: For the sensor itself, maybe we could experiment with more yarns and find one thinner. Adjusting the knitting mechanism might help with removing the folding in our current sensors.
- Signal Processing: More research could be done for handling finger dependences with software.

4.2 Other Use Cases

We explored our use case of piano playing, the stretch change is actually quite small compared with existing works like elbow bend detection. And the fact that we were able to achieve acceptable accuracy gives us the confidence with cleaner raw data, there are many opportunities. Using a industry knitting machine allows much tenser knit stitches, less space between yarns, and using thinner yarns. Right now each sensor is 3 stitches wide, but it can easily become 10 stitches wide with the same width using industrial knitting machines. Prior works have shown that resistive yarn have a larger range of changes than conductive yarn, so using other yarns should also yield better results.

So far, we have only detected whether the sensor is being stretched. Future works can explore how much the sensor is being stretched. Last but not the least, knitting can easily produce flat 2D sheets and even complex 3D shapes. For our prototype, we are really treating the knitted stretch sensor as a 1D string. Future works could look into stretch sensing from different directions.

5 TEAM MEMBERS CONTRIBUTION

In the early stage, all team members contributed to idea brainstorming, concept development, and project scoping. As Catherine was not in Pittsburgh, she was focusing on all the coding, including signal processing and debugging tools. Jiaqi and Willa were focusing on sensor fabrication. Specifically, Jiaqi explored knitting techniques and fabricated all the resistive sensors and Willa designed the system schematic and sewed components together. We believe that the work was evenly distributed among team members.

REFERENCES

[1] Maureen Selina Laverty. *Knitted Stretch Sensors.* 2018. URL: https://www.maureenselinalaverty.com/knitted-stretch-sensors.