5-2022

Automatic testing of organic strain gauge tactile sensors.

Brian P. Goulet
University of Louisville

AUTOMATIC TESTING OF ORGANIC STRAIN GAUGE TACTILE SENSORS.

By

Brian P. Goulet

B.S. University of Louisville, 2020

A Thesis

Submitted to the Faculty of the

J. B. Speed School of Engineering of the University of Louisville

in Partial Fulfilment of the Requirements

for the Degree of

Master of Science in Electrical Engineering

Department of Electrical & Computer Engineering

University of Louisville

Louisville, Kentucky

May 2022

AUTOMATIC TESTING OF ORGANIC
STRAIN GAUGE TACTILE SENSORS

By

Brian P. Goulet

B.S. University of Louisville, 2020

A Thesis Approved on

April 28th, 2022

by the following Thesis Committee:

____

Dr. Dan O. Popa

____

Dr. John F. Naber

____

Dr. Michael L. McIntyre

____

Dr. Cindy K. Harnett

____

Dr. Ruoshi Zhang

DEDICATION

This thesis is dedicated to my parents (Elizabeth and Gregory), siblings (John and Timothy), and my extended family and friends, both in Louisville and the DMV, who have taught, supported, and influenced me in this work.

ACKNOWLEDGMENTS

I would like to thank Dr. John F. Naber for giving me the initial opportunity to enter research at the University of Louisville. Without the initial offer, I never would have gone on to do research at a higher level and would not have found myself as a member of the NGS Group. I would also like to thank Dr. Dan O. Popa for allowing me to join the NGS group and for his guidance. Without his teachings, this work would never be completed. I would also like to thank Dr. Ruoshi Zhang for his experience in leading the project and allowing me to help. I would also like to thank Dr. Michael L. McIntyre and Dr. Cindy K. Harnett for serving on my committee and teaching me during my time at the University of Louisville.

I would also like to thank all other members working on the NGS KAMPERS project. Dr. Ji-Tzouh Lin, Olalekan O. Olowo, Bryan Harris, Kavish Sudan, Douglas J. Jackson. Their work helped this thesis and allowed me to complete it. I would also like to thank members outside of the KAMPERS project who helped me. Dr. Andriy Sherehiy, Dr. Moath Alqatamin, Dr. Thomas Roussel, Dr. Sumit K. Das, Danming Wei, Jacob Berdichevsky, and Brooke Ritz. All of them have assisted throughout this work and all has been greatly appreciated.

I would also like to thank the National Science Foundation(NSF) for their support through the following project: #1849213 RII Track-1: Kentucky Advanced Manufacturing Partnership for Enhanced Robotics and Structures[54].

Finally, I would like to thank my family and friends, both in Louisville and in the DMV. Without their support, I would not have been able to complete this work and my life would be incredibly different

ABSTRACT

AUTOMATIC TESTING OF ORGANIC STRAIN GAUGE TACTILE SENSORS.

Brian P. Goulet

April 28th, 2022

Human-Robot Interaction is a developing field of science, that is posed to augment everything we do in life. Skin sensors that can detect touch, temperature, distance, and other physical interaction parameters at the human-robot interface are very important to enhancing the collaboration between humans and machines. As such, these sensors must be efficiently tested and characterized to give accurate feedback from the sensor to the robot.

The objective of this work is to create a diversified software testing suite that removes as much human intervention as possible. The tests and methodology discussed here provide multiple realistic scenarios that the sensors undergo during repeated experiments. This capability allows for easy repeatable tests without interference from the test engineer, increasing productivity and efficiency. The foundation of this work has two main pieces: force feedback control to drive the test actuator, and computer vision functionality to guide alignment of the test actuator and sensors arranged in a 2D array. The software running automated tests was also made compatible with the testbench hardware via LabVIEW programs.

The program uses set coordinates to complete a raster scan of the SkinCell that locates individual sensors. Tests are then applied at each sensor using a force controller. The force feedback control system uses a Proportional Integral Derivative (PID) controller that reads in force readings from a load cell to correct itself or follow a desired trajectory. The motion of the force actuator was compared to that of the projected trajectory to test for accuracy and time delay. The proposed motor control allows for dynamic force to stimulate the sensors giving a more realistic test then a stable force.

A top facing camera was introduced to take in the starting position of a SkinCell before testing. Then, computer vision algorithms were proposed to extract the location of the cell and individual sensors before generating a coordinate plane. This allows for the engineer to skip over manual alignment of the sensors, saving more time and providing more accurate destinations.

Finally, the testbench was applied to numerous sensors developed by the research team at the Louisville Automation and Robotics Research Institute (LARRI) for testing and data analysis. Force loads are applied to the individual sensors while recording response. Afterwards, postprocessing of the data was conducted to compare responses within the SkinCell as well as to other sensors manufactured using different methods.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Over the past 70 years, the field of robotics has continued to develop and integrate itself into our daily human lives and begin to automate tasks that were originally only performed by humans. Robotic labor has many benefits over human labor by having much greater precision and speed, not tiring, etc.. While amazing, robotics still cannot replace everything we do. This is where human-robot interaction technology can augment a robot's capabilities and allow it to complete many tasks. The end goal of this are humans and robotics working alongside each other and interacting like how people to people would. To reach that goal, robots will need to gather and process data in the same ways that humans do to put us all on an even footing. In the recent years, incorporating a similar sense of touch into robotics has been researched by a good number of groups around the world.

The sense of touch helps humans identify what and when they are interacting with another object using skin. Robotic skin is seen to incorporate the sense of touch, but studies have proven that a simple tactile solution does not do enough[2]. Organic electronics have been seen as a possible solution due to their flexibility and piezoresistive characteristics of materials[3]. Someya and Sekitani propose a printed large area pressure sensor that has the flexibility to wrap around a prosthetic hand as seen in Figure 1.

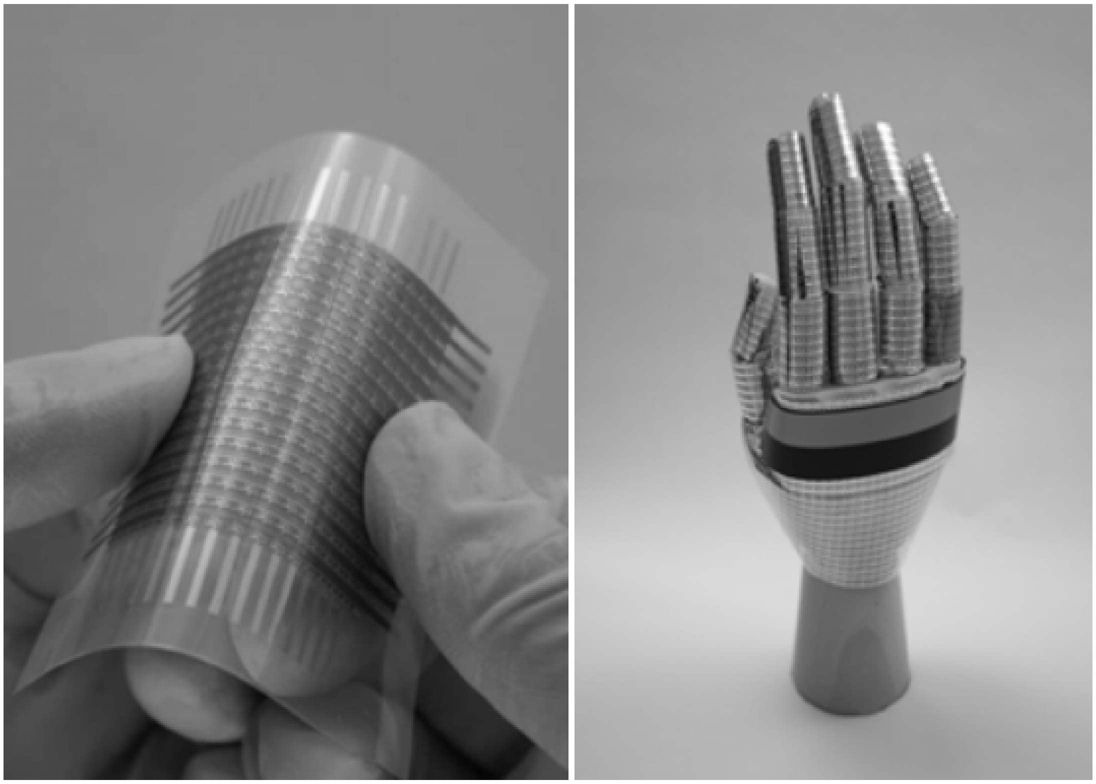


Figure 1: Electronic artificial skin made from organic transistors that can stretch around a hand[1]

The Next Gen Systems(NGS) group at the University of Louisville is developing organic strain gauge sensors based on Poly (3,4-ethylene dioxythiophene)-poly(styrene sulfonate), or PEDOT:PSS used in the Kentucky Advanced Manufacturing Partnership for Enhanced Robotics and Structures(KAMPERS) project[54]. The sensors are fabricated with different methods in house that each provide different levels of sensitivity and durability. An example of a 4x4 laminated tactile sensor called "SkinCell" can be seen in Figure 2. When manufacturing sensors, it is important to collect data and characterize the different types in a way that leads to finding the best way to produce them. This is not possible without accurate and reliable equipment to collect test data.

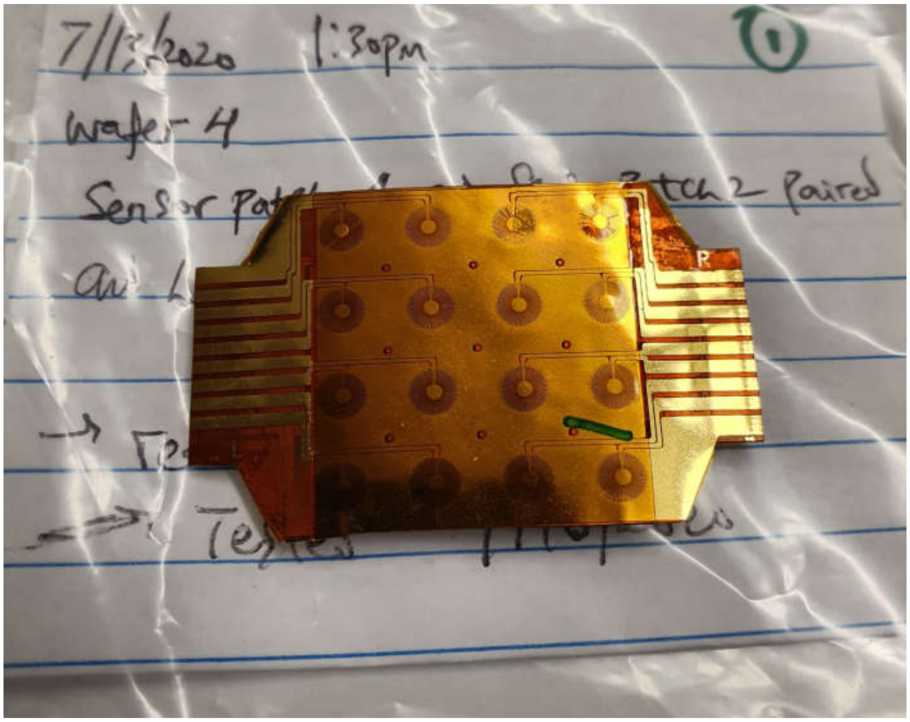


Figure 2: NGS-produced SkinCell, 4 by 4 skin sensors with 3.5mm vertical and horizontal spacing between each sensor

Previous testbenches have been designed by the NGS group for the purpose of force loading. These focused on testing single pressure sensors with force applied via an indenter[55, 56, 57] However, none of the previous testbenches added the XY raster scan to quickly move between each sensor to proceed to the next test. Also, sensors tested in previous works were not designed to have multiple on a single SkinCell. Therefore, there was a need for an updated testing bench to efficiently travel between points and apply tests to allow for quick and thorough data collection on an entire SkinCell.

The result of this work is a new automated testbench designed to allow easy, repeatable testing styles over any type of strain-based sensor. In the testing environment, the user can choose a premade test and have it run without having the need to align a sensor to any manual coordinates. These tests apply a desired strain on the sensors by moving a force actuator until it reads a force that is desired by the user. These can be static applications of forces as well as dynamic waveforms based off real time mathematical equations. By creating this, we can easily see sensor response and characterize trends and the quality of the manufacturing process. This allows us to wean out bad manufacturing processes and learn the response of them at a much quicker rate.

1.2 Thesis Contributions

1.2.1 Testbench Design and Development

The contribution of this work was to create a fully autonomous testing bench for any strain-based sensor without substantial manual coordinate input from the test engineer. The testbench was designed to primarily test the Skin Sensors made by NGS but has the flexibility in its options to test sensors outside this scope as well. LabVIEW was used to create the entire software suite needed for the bench. LabVIEW allows easy access to all the required hardware needed to run the automated test setup. Using this testbench, we can easily test and characterize the sensors used in the skin sensor project. My contributions to the project compared to previous works from the NGS group were as follows

1) Tuning of the testbench to run tests automatically across a 4x4 SkinCell as opposed to singular sensors[54,55,56]

2) Increase in the force control loop cycle by an order of magnitude in comparison with previous benches.
3) The experimental evaluation of the testbench reliability by collecting data from over a dozen NGS-designed sensors during the project

The testbench includes built in force control, that has the system dynamically adjust to whatever type of force the test engineer wants. In addition, the force is able to follow mathematical equations, such as sine and triangle waves, to allow dynamic force loading to be applied to the sensor cells. The testbench also includes automated movement between the individual sensors, ensuring that every sensor is tested in a single test.

To measure data from the sensors, an outside python Linux script was developed by the NGS group that runs separately from the real time target. To account for this, datalogging with accurate time stamps was developed to run in parallel with the testing setup. This allows for easy alignment of the two data files afterwards.

In addition, a fully integrated computer vision script was developed to allows for the system to automatically the individual coordinates of the sensors from the entire cell. Once extracted, a full coordinate system is generated where every sensor is located. The script automatically considers if the sensor is laid in at an angle when generating its coordinate plane. The code for the computer vision portion is all run-on LabVIEW PC, then the data is transferred over to the real time target.

Finally, the testbench was applied to sensors in the KAMPERS project and results were used in the following publications

1. O. O. Olowo, R. Zhang, Z. Yang, B. Goulet, and D. O. Popa, “Organic piezoresistive robotic skin sensor fabrication, integration and characterization,” in *Volume 2: Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability*, 2021.
2. R. Zhang, J.T. Lin, O. O. Olowo, B. Goulet, B. Harris, and D. O. Popa, “SkinCell: A Modular Tactile Sensor Patch for Physical Human-Robot Interaction” *IEEE Sensors Journal,* Under Review, 2022.
3. O. O. Olowo, R. Zhang, A. Sherehiy, B. Goulet, A. Curry, D. Wei, Z. Yang, M. Alqatamin, and D. O. Popa. “Inkjet Printing of PEDOT: PSS Inks for Robotic Skin Sensors” *International Manufacturing Science and Engineering Conference*. American Society of Mechanical Engineers, Accepted, 2022.
4. O. O. Olowo, D. Wei, D. Ratnayake, B. Goulet, A. Curry, A. Sherehiy, R. Zhang, and D. O. Popa. “PEDOT: PSS polymer Aerosol Jet-printing for Robotic Skin Sensors.” *IEEE International Conference on Flexible, Printable Sensors and Systems,* Accepted, 2022.

1.2.2 Testing and Characterizing Skin Sensors

The sensors designed by the NGS group needed to be tested to see the worthwhileness of printing types and manufacturing styles. To identify how these sensors responded, extensive testing was performed on the different sensor types. Each of the 16 sensors on a cell were tested in two ways, both static and dynamic. The static testing involved two separate tests with different forces, the first with 1 newton and the second with 2 newtons. The dynamic test involved a two separate 4-stage ladder tests. The first test would focus on lower ranges of force, at 0.5, 1, 1.5, and 2 newtons. The second ladder test starts at the lower range, but gradually greatens to run at higher forces. Those force values are 0.5, 2, 4, and 6 newtons. Once the tests were completed, post processing was done on the gathered data to see the resulting performance of the individual sensors and to look for any trends.

1.3 Challenges with the project

Developing the testing bench brough in a slew of challenges for myself. Learning and getting used to LabVIEW, along with incorporating the high level of code development needed for this took a few weeks. The hardware, while capable, did begin showing its age through a slew of breakdowns and crashes that occurred and deleted several weeks' worth of development. The load cell, which is used to measure the current force response, was prone to high levels of noise and needed to be replaced and designed around.

The motor amplifiers brought another cause for concern. The wiring setup for the amp that was initially present was prone to breaking down, and the number of wires made debugging prolonged. The age of the wiring system led to the connector to one of the motor amps to oxidize, breaking the amp in the process. To fix this, a new PCB and chassis setup were designed to remove the wiring and to put more safety measures in place for the amplifier.

Development on the FPGA for the CompactRIO presented its own challenges. FPGA development with LabVIEW is substantially different then its RTL counterparts and was needed to be learned over time. Additionally, any development of the FPGA would cause the RT bench to go unusable, at first this was thought because it corrupted the original software. However, after about a month of debugging it was discovered that activating the FPGA would cause the Softmotion engine to break down, and the whole system needed to be wiped in order to stop the FPGA from running.

1.4 Thesis Organization

The Thesis is organized in the following way:

Chapter 2 is background research of the following topics: Control Loop Systems, Human Robot Interaction, Electronic skin, and, finally, Computer object recognition. Chapter 3 describes the automation profile designed for the testbench, as well as the force controller and motor controller designed for more accurate testing. Chapter 4 introduces the work in computer object recognition for extracting automation coordinates. Finally, chapter 5 concludes by summarizing the work of this thesis and discuss the future work that can be done.

CHAPTER 2

LITERATURE SURVEY

2.1 Force Control

Control systems are a series of devices or equations that manages, directs, or regulates the behavior of another device. Control systems commonly help a system go to a desired trajectory of use while minimizing the amount of percent error from that trajectory. Control systems have existed for at least 2000 years, with some of the earliest being implemented in water clocks in 270 B.C.[20]. It would not be until the end of World War 2 for classical control techniques to cement themselves, and the integration into electronic systems would come next[20]. Control systems would continue developing and become one of the driving factors of automation of labor in tasks[27]. The systems make it so robots can perform the same tasks as humans with a higher degree of accuracy.

One of the most common control system types is the closed loop control system. A closed loop control system compares the current output parameter of a system to one that is desired by the user. The difference is calculated, and the controller updates the value the system should be at to mitigate the difference, before repeating this loop. Medical devices are a common area for closed loop control systems. Anesthesia levels in a patient under surgery must be constantly monitored and adjusted to keep the patient safe. Schwilden et al. propose the use of a control feedback system loop to automatically update levels of anesthesia in the body[21]. The resulting experiment showed promise but made note that it must be designed with an emergency fail safe in mind due to the dangers it can pose if unmonitored.

Control systems are often used in testing environments where a robot applies a specific action to another system. This action is primarily done using either position control, based on the current position of the robot, or force control, based on the current force applied by the robot. Force control provides a stable groundwork for controls that must manipulate something that interacts with the environment around it[22]. The general application of a force control in a robotic arm is to read in the current force being applied, then have the controller compensate to try and match the desired force[23].

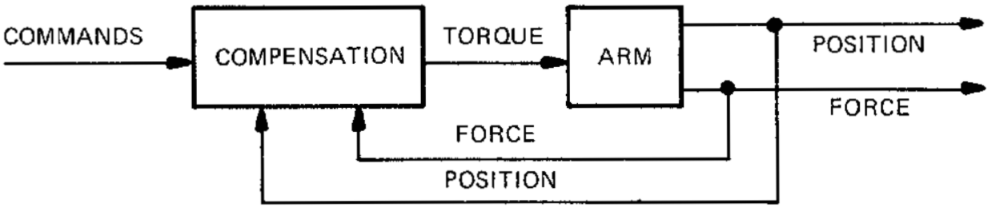


Figure 3: General diagram of a servo force control application[23]

Both force control and position control are the primary methods for producing compliant motion. Compliant motion is a motion that is constrained by a task. Force control is considered an active compliance that is implemented via a software control loop[24]. The programmability of a force control system allows it to be used for specific application use cases that best fit it, while developing secondary programs for separate applications.

Implementation of force control can be done in two primary ways. The first is a hardware approach using a force sensor such as a load cell to read the current value of the force applied. The second is using force estimation models to follow trajectories considering by the current force and position[25]. This force and position control can be referred to as impedance control. For a comparison between explicit force control and impedance control, see the work in Komati et. al[58]. Impedance control is seen as the relationship of the applied force and the error of the position or the velocity[26]. Jung et al. propose the use of a force control structure with a robust position control algorithm that provides a simple solution to minimizing force error on a Puma 560 robotic arm. To do this, the system takes a summation of the position, force, and velocity error transitioned to diagonal NxN matrixes and summed with the current reference end point acceleration and the adaptive law proposed to compensate for the remaining error. The resulting sum is compared to the joint angular velocity of the arm before being multiplied through by the estimate of the next definite inertia matrix. Afterwards, a final summation of the previous value and a delay for stability system is transferred to the robot. The full schematic can be seen in Figure 4.

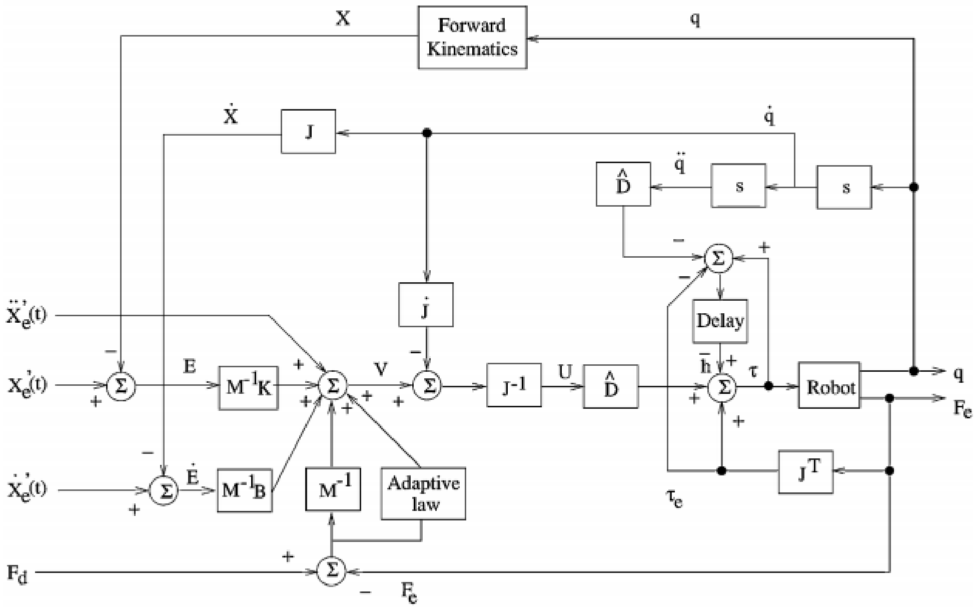


Figure 4: Robust control law schematic proposed by Jung et al.[26]

The proposed controller can adapt to its environment, even if the full dynamics of the robot are unknown. When changes in the stiffness of the environment occurred, the robot arm managed to adapt itself and change its force output to achieve stability[26]. Adaptive impedance controllers that can learn in an unknown environment have been proposed by the NGS group, see Alqaudi et. al and Singh for examples in this work[59, 60].

2.2 Human Robot Interaction

The use of electronic skin is primarily to allow more complex interactions with robotics. This field of study is commonly called human-robot interaction (HRI). HRI is a field of study that researches and develops the way humans cooperatively interact with robotic systems. HRI can be separated into 4 areas of application [12].

- Human supervisory: Where a human watches over a robot and updates its work as needed.
- Remote Control: Where a human directly controls a robot in areas that may be hazardous or impossible to get to for the human.
- Automated Control: Where the robot is in direct control and provides a service to the human without manual input.
- Social Interaction: Where the robot aids those who normally cannot interact human to human and allows the human to develop something in response

Physical human robot-interaction is seen in areas where humans work side by side with robots to accomplish tasks. These primarily human supervisory and automated control. In industry, humans work alongside machinery every day in manufacturing lines. These robots can impose safety critical situations to operators and those who depend on them. As such, cooperative robotics must be designed with safety critical features in mind, and why many industries have not moved all features over to HRI[17]. To avoid injury, safety systems are developed that can help avoid collisions, limit the impact force to an acceptable level if a collision does occur, and sensors designed to read and respond to human navigation[16].

To accommodate for collision-based injuries, robotic skin has been proposed as a solution. Duchaine, Vincent, et al. propose the creation of a flexible robotic skin for use in safety configurations with robotics. The robot skin proposed is capable of sensing multiple contact points through that can pinpoint the spatial location of collisions[18]. The skin is fabricated from polyimide films with electrically conducted ink that is laid on top of a pressure rubber sheet.

To test the skin, a collision experiment was made that has the robotic arm collide with a force cell at a specific velocity. The velocities were chosen between 100mm/s up to 1m/s in stages of 100mm. The activated skin was compared to both the deactivated skin and the robotic arm with no skin mounted. The maximum measured force was read from the load cell for each collision and compared, and a unified pain threshold was set to 50N. When looking at the results, comparisons were only able to make between the velocities of 100, 200, and 300 mm/s, afterwards the values become high enough to where it does not matter. The deactivated and without skin models never managed to go under the 50N pain threshold, even at 100mm/s velocity. With the robotic skin, every value stayed well below the pain threshold, with 1m/s velocity only achieving roughly 30N of max force[18].

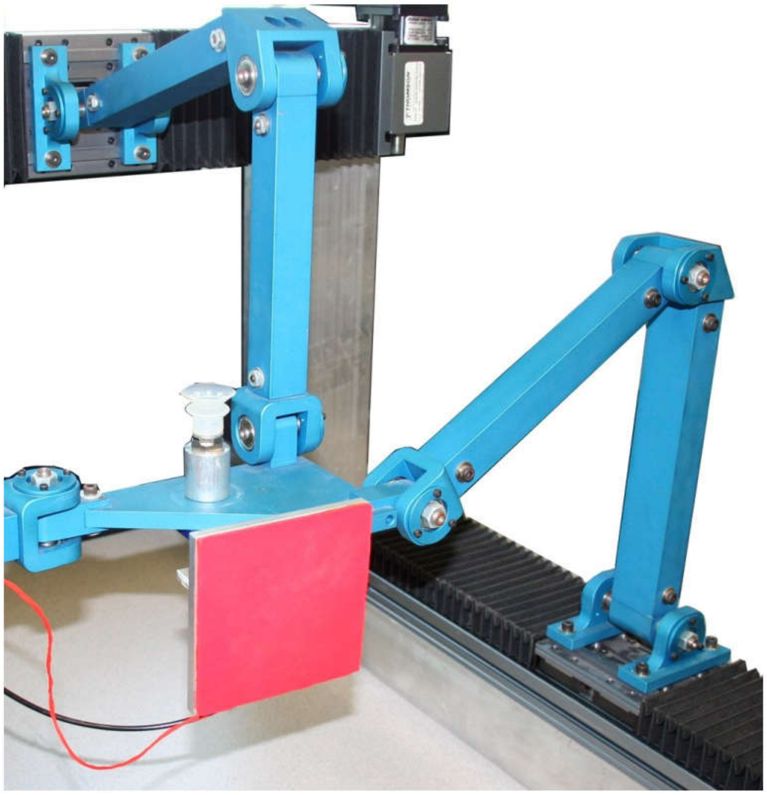


Figure 5: Robotic Arm with skin mounted to its end[18]

The development of safety measures is important so that we may ingrain more HRI into our daily lives. Real time software is developed to coincide with the hardware to lessen collisions. This software produces the fast response that allows safe interactions to occur and to minimize the danger to all parties[19]. Past work of the NGS group has explored physical human-robot interaction using classical and/or neuroadaptive robot controllers and robotic skins. These algorithms autocalibrate tactile information from SkinCells, and transform this information into guiding motions for real time feedback[13, 14, 15].

2.3 Electronic Skin

Skin acts as the primary way the body communicates with the outside world. Recent technological advancements have started developing electronic skin for robotics by simulating nerve receptors vis sensors[6]. Biological skin in animals uses a multitude of nerve receptors underneath the skin to translate necessary information to the brain. Skin receptors each translate different types of data such as temperature, pain, and mechanical stimuli that are then translated to the brain directly [5]. Electronic skin is developed to try and mimic these receptors in the form of differing sensors; however, these signals cannot be directly transferred to the brain. Data communication protocols, signal encoding and processing, and signal transmission all must be developed to convey this information[4].

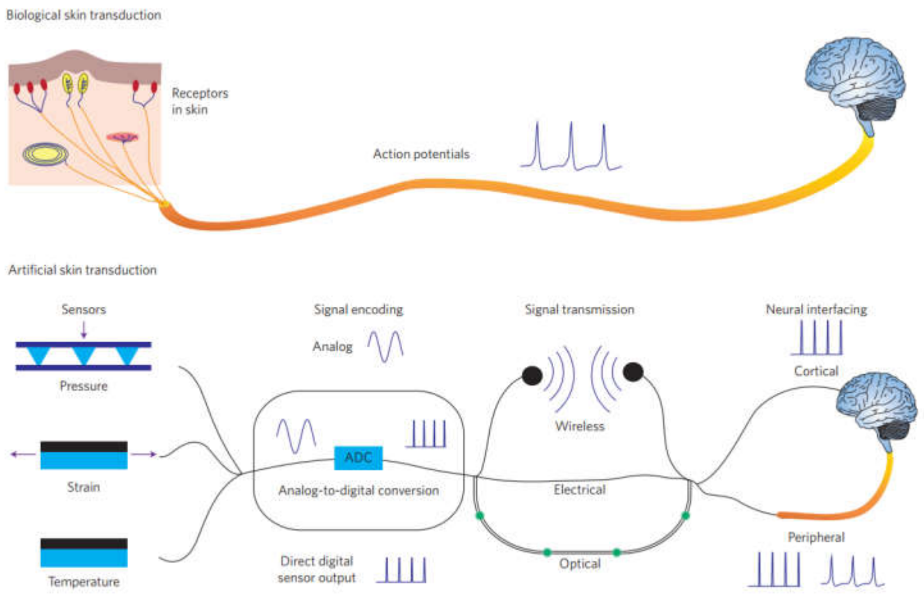


Figure 6: Comparison of steps for sensory stimulus, both natural and artificial[4]

Development of the first electronic skins began in the mid-1970s, culminating with General Electric creating an infrared sensor based robotic skin in 1985[7]. The infrared skin was able to be aware of its adjacent surroundings which allowed it to move around and avoid objects[7]. It would take until the 1990s for researchers to begin experimenting with flexible materials. In 1997, a sensor array was proposed by Jiang et al. that developed a MEMS sensor array on individual silicon islands[8]. This was one of the first proposed electronic skin sheets and allowed for shear stress sensors to be developed on the islands. Flexible electronics have been used as a development basis since.

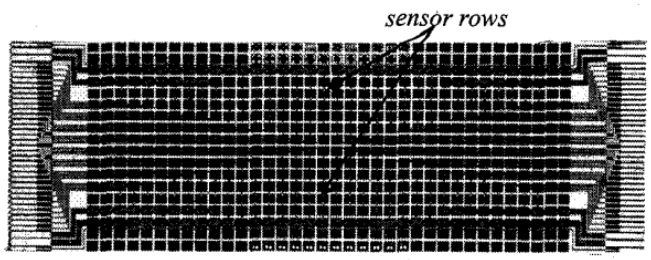


Figure 7: Finished flexible shear sensor array proposed by Jiang et al.[8]

The goal of the electronic skin should be to replicate the sense of touch in humans as much as possible. Human nerve receptors are incredibly sensitive and can fell force differences as little as. To replicate this, sensors must be designed that can notice incredibly small changes in applied force. Besides that, human nerves stretch over the entire body and have millions of receptors that each produce their own response. To replicate this, the sensors must be small, allowing hundreds or thousands to be eventually placed on the material. With hundreds of sensors comes the need to process vast quantities of data from individual points. This means the "brain" of the operation will need to be able to differentiate the types of signals from each other, and at a quick enough rate as to not be overwhelmed[9].

Li et al. propose an electronic skin constructed from an all-fiber structure[10]. The fiber structure allows for an elastic and breathable skin while being able to incorporate pressure sensing, energy harvesting, and motion positioning. The skin is constructed using three layers of nanofibers, made of polyvinylidene fluoride, carbon, and polyurethane. These layers act as the sensing, electrode, and substrate layers respectively. To test the sensitivity of the skin, external pressure was applied while the voltage response was recorded down. The skin was determined to have a 0.18V/kPa response over a single electronic skin pixel, each with an area 1cm^2. The results of the sensitivity do not compare relatively well to another nanostructure proposed by Chen et al., which has a voltage response of 3V/kPa and is designed specifically for harvesting energy[11]. However, the other structure cannot produce the same elasticity or breathability [10].

The use of tactile sensors as a way to do pressure sensing for electronic skin has risen in popularity over the recent years. Two types of sensors have become used widespread at measuring tactile force: force-sensing resistors(FSRs) and force sensing capacitor[48, 51]. Both of these types of sensors can be easily made into singular tactile, and also into groups to form the sensor arrays. In addition, new types of tactile are receiving more attention. A tactile based on vision that uses an optical rubber that uses illumination as its sensing parameter[48 52]. The force resolution is determined by image brightness, which includes a resolution increase due to the high pixel density in these sensors. Besides a visual sensor, Y. Tenzer, L. P. Jentoft and R. D. Howe propose using multiple MEMS barometer sensors. The sensors are then encapsulated in a rubber mold, which helps act as the pressure conversion between the sensor at the force[48 53]. The small footprint of the sensor allows multiple sensors to be placed in the same piece of rubber, on the same PCB.

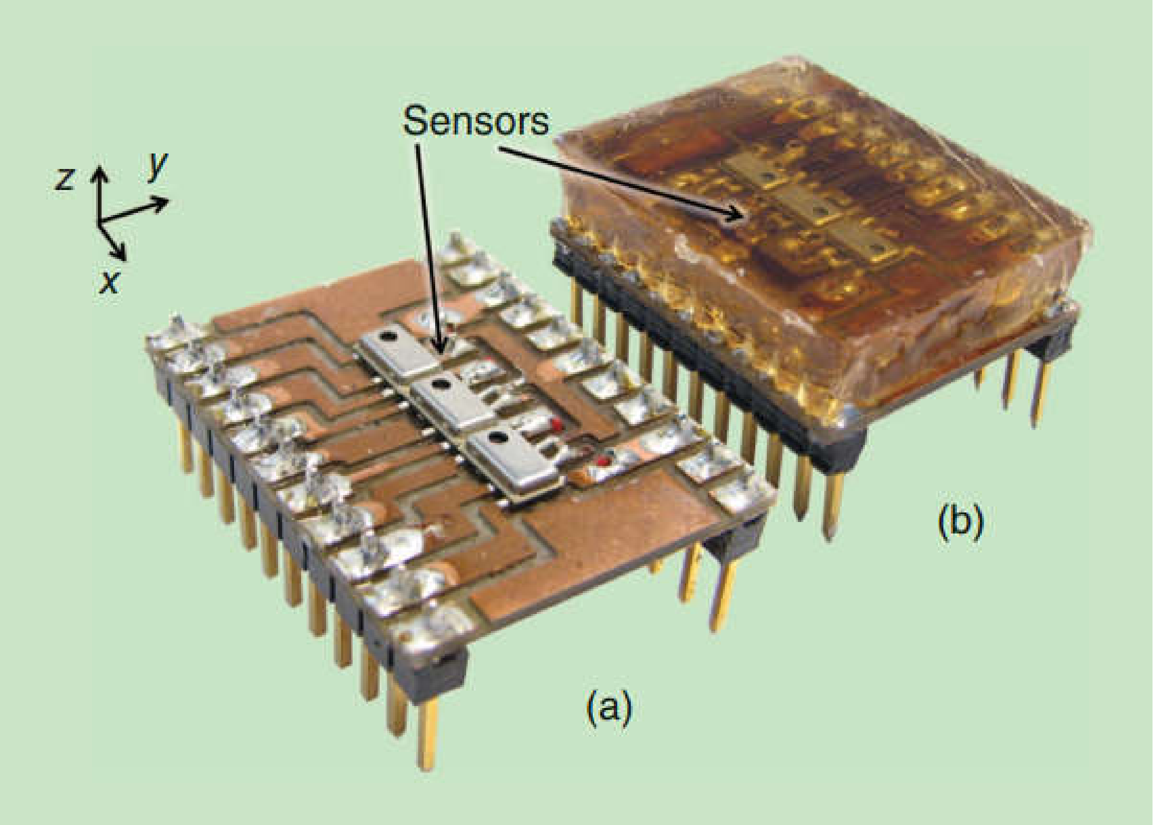


Figure 8: a) Sensor layout of the barometer mems setup b) setup encapsulated in rubber mold[53]

Sensors at the NGS lab at the University of Louisville have been developed over the past decade. They consist of electrode arrays that are fabricated by cleanroom or 3D printing techniques using the organic piezoelectric material PEDOT:PSS deposited in a thin film above each electrode[61, 62]. Sensors are arranged in 4x4 array to form a SkinCell. They also include a molded elastomer top and bottom, along with electronic circuitry for local signal processing, and interfacing of the tactile information to a robot[63]. Finally, dynamic models of these sensors were extracted and included in simulation models on robot surfaces[64].

2.4 Digital Image Processing

The ability to recognize what and where an object is allows a laborer to find the information needed to perform a task without wasting excess time and energy. To give this ability to computers and robotics was a task eagerly worked on[28]. Computer vision is a field of science that specializes in computers being able to gain information from images and videos. Object detection acts as a subset of computer vision that deals with trying to find an object and where it is located. This is done by developing models and scripts that can extract this information from a picture while referencing its relative location.

To accurately find an object, the computer should first be able to recognize general shapes. Shape detection techniques Human vision at the lowest level primarily utilizes edge and regional information to determine a perceptual unit[29]. This argument proposes that humans do not need the full depth of object reconstruction to recognize what an object is. Line recognition is a common technique used to find the edges of a shape[30]. To do this, a histogram is implemented focusing on a relevant parameter, such as brightness or color. Similar data points are tracked across a region of interest to locate a line. Thenmozhi et al. propose the use of shape detection to identify insects on field crops[31]. A script is designed to extract the outlining shape of an insect from a picture of crops growing by outlining the outer edge of the insect. To test this, multiple images with an insect on a crop were ran through the script. The script was able to identify the general outline of the insect and generate an estimated area of the insect along with its location on the image.

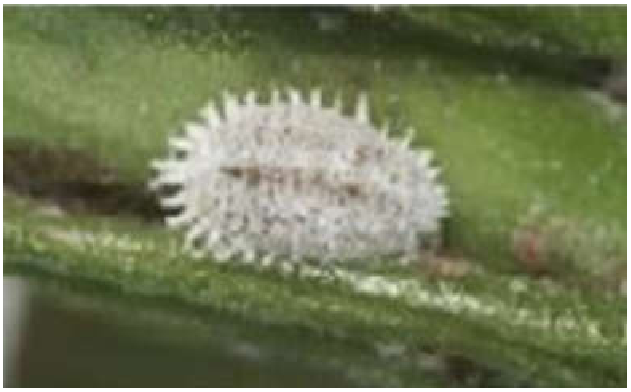


Figure 9: Image of insect on plant before identification[31]

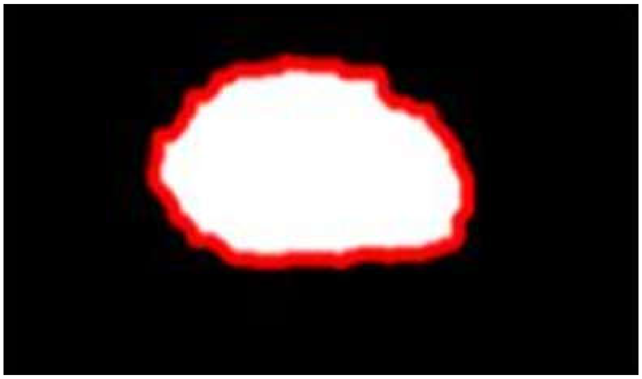


Figure 10: Shape identification of insect on plant[31]

Line detection of images does have drawbacks. If the camera does not produce a clear enough image, it can be susceptible to noise making the algorithm inaccurate[32]. More complicated edge detection techniques that circumvent this issue are also prone to being very time consuming and producing false positives while being incredibly more complex in design. Tone mapping algorithms, that map a set of colors across the image and build lines from the map, can be used as an effective way to find finer lines[33]. However, these techniques can suffer from high color contrast and requires an image to be processed in an RGB color span for the best effect, which can be very computationally heavy. Fitting preconstructed models to photometry is a way to get around this[34]. Rigid model fitting can be accomplished when the shape of a target object is known in advance and the system will be working with identifying that underlying target. Pattern recognition of rigid models allows a computer to search for a predetermined model and match it to multiple locations, identifying the information needed[35].

CHAPTER 3

TESTBENCH, AUTOMATION PROFILE, AND FORCE CONTROL.

When developing any type of sensor, it is important to test and characterize the responses to help refine the manufacturing process. However, a test done through human labor is hard to replicate. Human labor adds uncertainty that is out of a user's control and must be removed. To do this, a testbench is created that allows a test to be done by machines instead. Machine testing is much more controlled and repeatable, giving a better view of how a sensor performs. However, this is only if the testbench is properly implemented.

Due to a computer failure, the NGS team was unable to use any of the previously designed testing software made by previous students, and instead created a stopgap solution for force applications. Testing was done by having the user move the motor positions based on fixed increments: 0.2, 2, 10, and 20mm. The user would select which axis they wished to move and then move one of the motors in either a positive or negative direction. Force was applied by shifting the Z axis down a fixed distance and seeing how much force was applied before readjustments were done. The tests could see functionality of the sensors but could not accurately compare responses between each other. The restrictions of the bench were seen and a new one would need to be developed.

This work implemented a complete software redesign of the skin sensor testbench to create a repeatable, easy to use, and autonomous testing suite. The current hardware setup offered everything needed to create this setup, so changes were not made. The testbench is able to run a predetermined choice of test from the user on all 16 sensor cells without any manual labor required. The Force indention is controlled by a force control algorithm to allow the engineer to test at specific values. Software development was changed from LabVIEW 2011 to LabVIEW 2014 to remain compatible with the testbench and the development PC.

3.1 Testbench hardware

As stated previously, the testbench configuration runs mostly the same hardware as the previous testing setups. When optimized, the system supplies the full range of motions needed to run every test. For future reference, the hardware is listed as such.

- Newport UTM150CC Motorized Linear Travel Stages(2)
- Newport UTM100CC Motorized Linear Travel Stage
- National Instruments CompactRIO(cRIO) model 9074
- National Instruments 9516 Motor Drive modules(3)
- National Instruments 9205 Analog to Digital Converter Module
- Transducer Techniques MLP-10 Load Cell
- Transducer Techniques TMO-1-24 Load Cell Signal Conditioner Board

The NI cRIO is a real time embedded controller that can run LabVIEW code needed to operate the testing suite. The first 150CC and only 100CC linear stages are used to control the X and Y positioning of the testbed respectively. Since the Y linear stage offers 100mm of travel range compared to the 150mm on the other two, it is important that modifications to testbench position are made with it in mind. The X linear stage is laid on top of the Y linear stage and then the SkinCell to be tested is put on top of that. The X and Y stages are used to control the alignment of the SkinCell to the force indenter that is attached to the final linear stage. The final 150CC linear stage mounts the indention tool that is used to apply force to the individual sensors. The indention is connected to the MLP-10 load cell. This is a new piece of hardware put in place to replace the previous MLP-25 load cell. The MLP-10 offers a smaller range of capability compared to the previous load cell, with a max range of 44.8 newtons instead of 111.2 newtons. However, putting more than 20 newtons of force on a sensor can cause irreversible damage to it, so the 111-newton range is completely unnecessary. Additionally, the MLP-10 offers a reduction in noise over the signal read in, which will be explained when discussing the signal conditioner board.

Connected to the load cell is the Load Cell Signal Conditioner Board. The board takes in a signal from the load cell and converts it to an analog voltage, with a max output voltage of 8V representing the max force value. When the MLP-25 was connected, it created a volt to newton ratio of 0.0719. The MLP-10 load cell has a ratio value of 0.179 volts to newton in comparison. This is important when considering general signal noise added to the force value. The TMO-1 adds a max 5mV of noise to the output voltage. On the MLP-25, this equates to .07 newtons of noise to the signal. The MLP-10 only produces a max .028 Newtons of noise to the signal. When dealing with something as small as newtons, it is important to minimize as much outside interference as possible so that estimates can be accurate.

The final pieces of hardware are three I/O modules that connect to the cRIO. The 9516 Motor drives are used to power and control the three linear stages. Lastly, the 9205 ADC connector takes in the voltage from the load cell board and allows it to be read into the LabVIEW programming environment. Figure 11 shows the full testbench setup with labels on the corresponding parts.

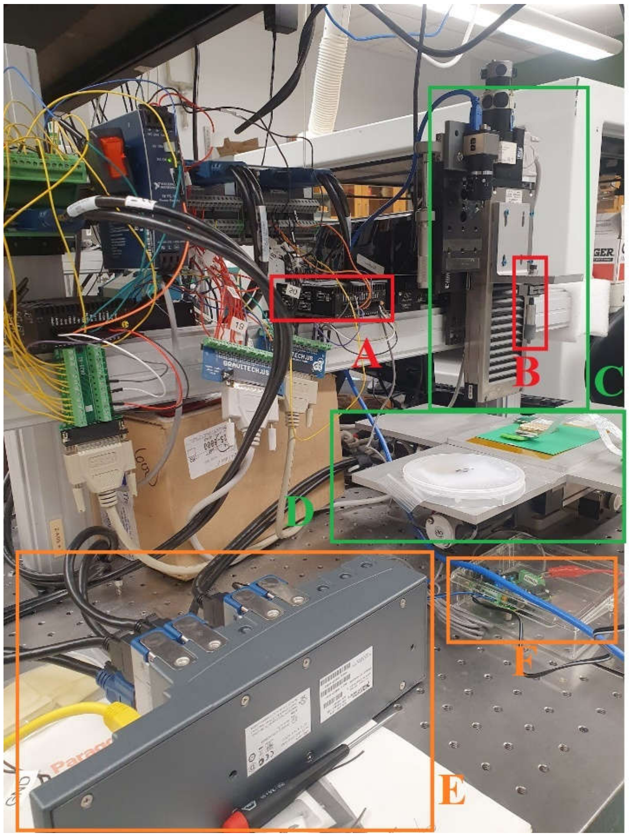


Figure 11:Testbench Hardware: A) Maxon Motor Amplifier. B) Transducer Techniques Load Cell C) Z Linear Motor Stage. D) X and Y Linear Motor Stages E) NI CompactRIO with 3xNI 9516 modules and 1 NI 9205 Module F) Transducer Techniques Signal Conditioner Board

3.2 Testbench Connection PCB Design

While the hardware of the previous setups remains, it cannot be said the same for the connection solution. The current connection solution revolves around a 37-pin terminal then having individual go to their respective ports. This solution can cause the entire system to go down if even one wire is misplaced, and the number of wires can make debugging the solution and incredibly long process. The previous solution also ended up being a reason as to why the Maxon Motor Amplifier broke down by having an internal fuse pop. A new solution is designed in order to ensure safety of the motor amp, and to apply the most up to date setup[36].

A PCB was designed to replace the wiring setup used previously in the testbench. The PCB carry's the 37 and 25 pin D-sub connectors, and a 16-pin connector to connect to the Maxon Motor amplifier. The 24V power supplies receives separate screw terminals instead of hijacking a previous position. Wire connections are replaced with PCB traces to create a more stable connection between the three parts. Added safety features are included to make sure that the Maxon Motor Amplifier no longer is in danger. A 5x20mm fuse was added to the previous power line, as well as a power LED to ensure that that The schematic, PCB layout, and 3D model of the PCB can be seen in Figure 12, Figure 13, and Figure 14.

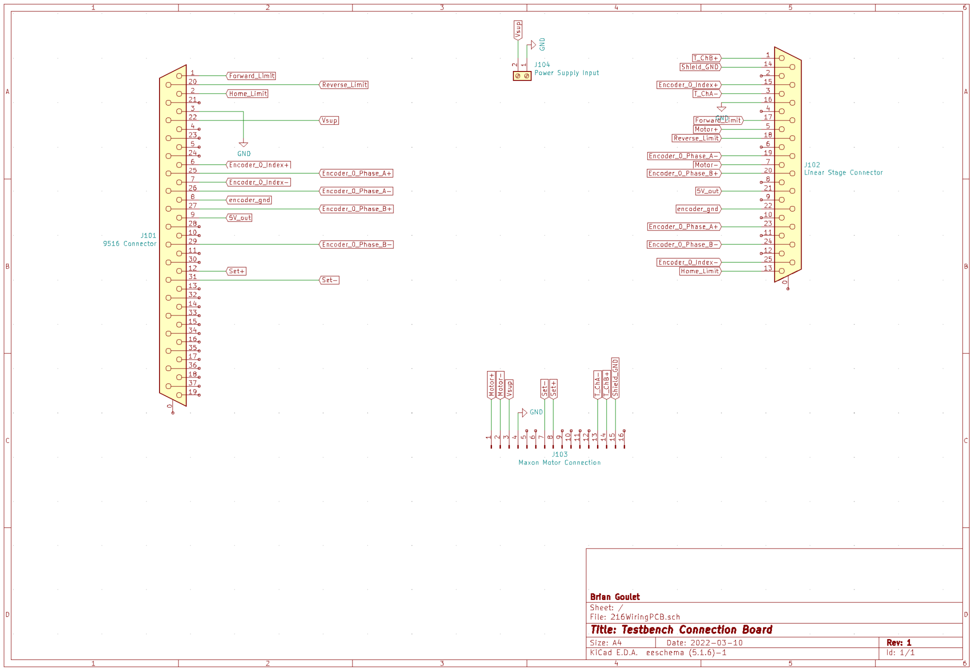


Figure 12: Electronic Schematic for connection board

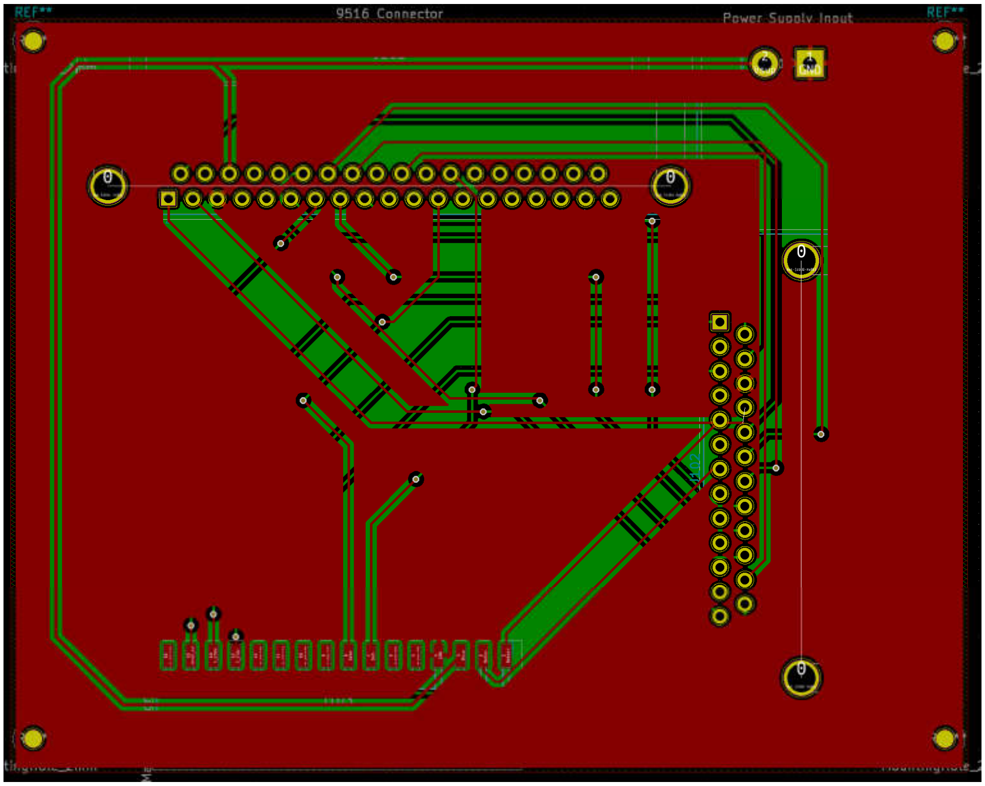


Figure 13: PCB trace design of the connector board

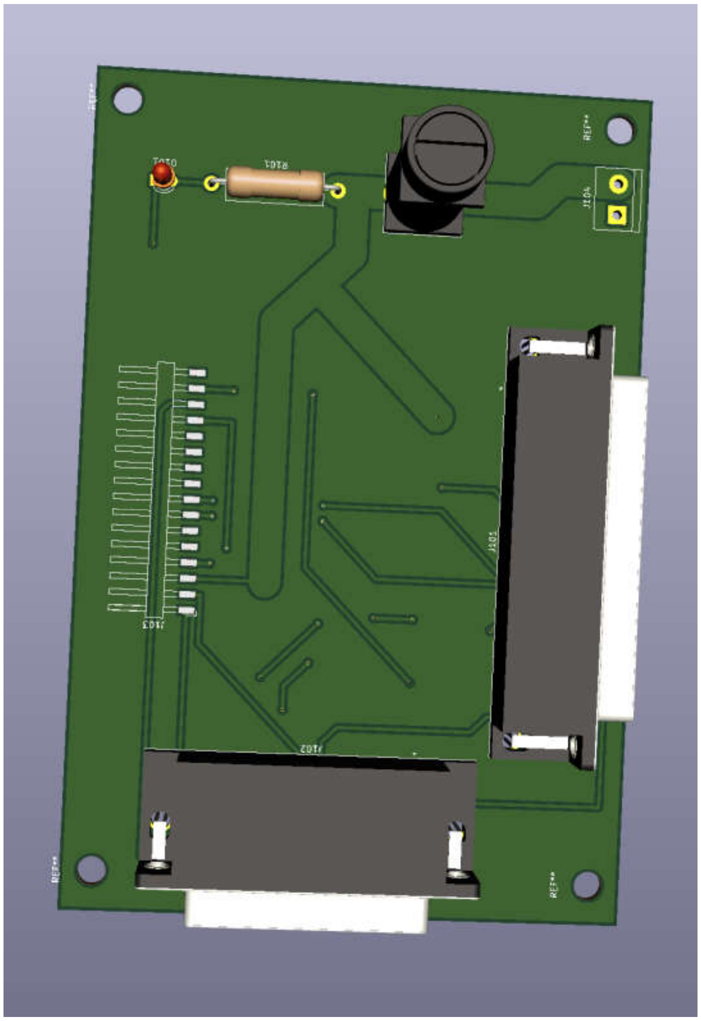


Figure 14: 3D Model of Completed Board

3.3 Testbench framework

The design of the testbench framework is finalized before code development can begin. The testbench features that the framework should be designed around are listed as the following

- Force control testing that allows the test engineer to choose from a variety of testing types

- Automated movement between each sensor on a singular cell after completing a sensor test
- Timestamp generation to match force loader data to that of the sensor response
- Internal error recognition that allows for the system to safely stop a test and return to a reset without damaging the sensor
- Manual takeover of the automated test if the engineer wants to work manually from that point on

The goal of the software is to allow the engineer to preselect an existing test then have that test to completion without necessary interference. If the engineer wants to take control of the system, they have the option to while the test is running. This allows the engineer to rerun tests on a specific SkinCell for clarification purposes. The testbench also needs to recognize when internal faults occur and specifically fix itself in a way so that no damage is done to the SkinCells or the testing hardware. Lastly, the data response file is generated through a separate programming environment outside of the Real Time controller. Accurate timestamps must be generated by the real time system so that the data can be aligned with that of the data response file, while also not impacting the performance of the system. Figure 15 shows a block diagram of the proposed testing setup.

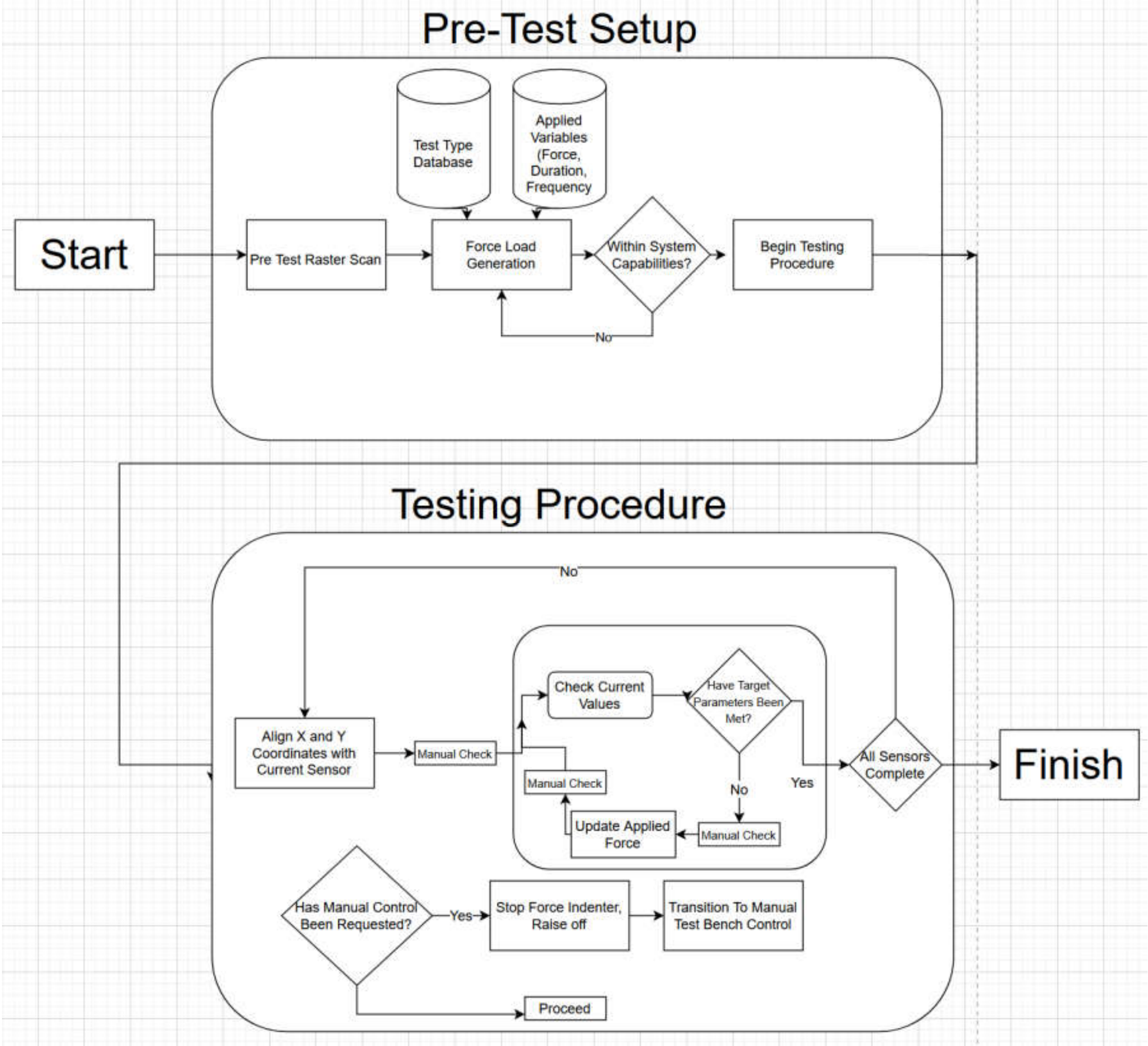

Pre-Test Setup
Start
Pre Test Raster Scan
Test Type Database
Applied Variables (Force, Duration, Frequency
Force Load Generation
Within System Capabilities?
No
Begin Testing Procedure
Testing Procedure
No
Align X and Y Coordinates with Current Sensor
Manual Check
Check Current Values
Have Target Parameters Been Met?
Manual Check
No
Yes
Update Applied Force
Manual Check
All Sensors Complete
Finish
Has Manual Control Been Requested?
Yes
Stop Force Indenter, Raise off
Transition To Manual Test Bench Control
Proceed

Figure 15: Block Diagram of the Developed Testing Setup

3.4 Reading in force from ADC

The ADC signal read in by the 9205 module comes out as a voltage that needs to be converted to the force read by the load cell. Voltage can be successfully transformed into the force of the system if a voltage to force ratio is known. To find this, a scale was placed underneath the force actuator and a set distance was put in place. The mass placed upon the scale was used in newtons law to solve for the force applied before dividing by the current voltage amount. Solving for k gives a volt to newton ratio of -40. Equation (1

) gives the equation to convert voltage to newtons, where K is the ratio, F is the force exerted on the scale, and V is the voltage value of the ADC.

$$K = \frac{F}{v}$$

(1)

3.5 Sensor Cell Profiling

The testbench is designed around solely testing the SkinCells. Each cell contains 16 sensor cells in a 4x4 grid setup with 7mm spacing both horizontally and vertically between sensors. Figure 16 shows the dimensions of a generic SkinCell.

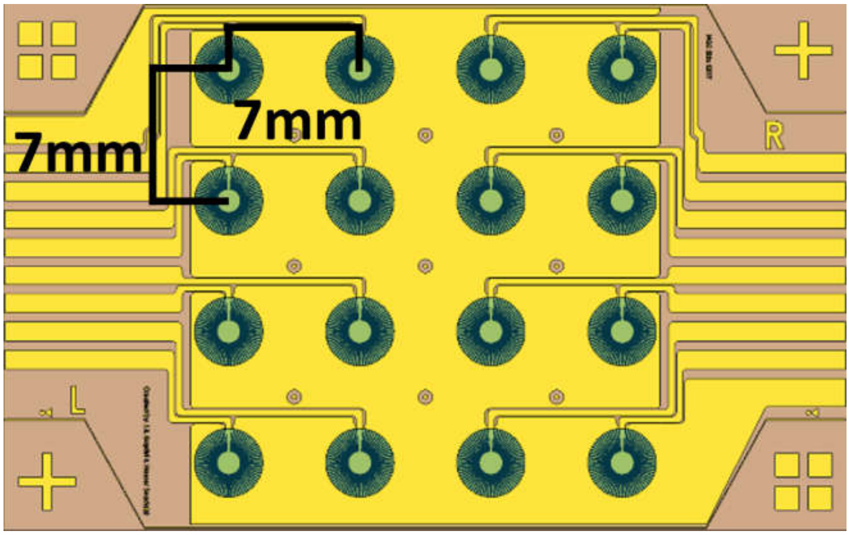


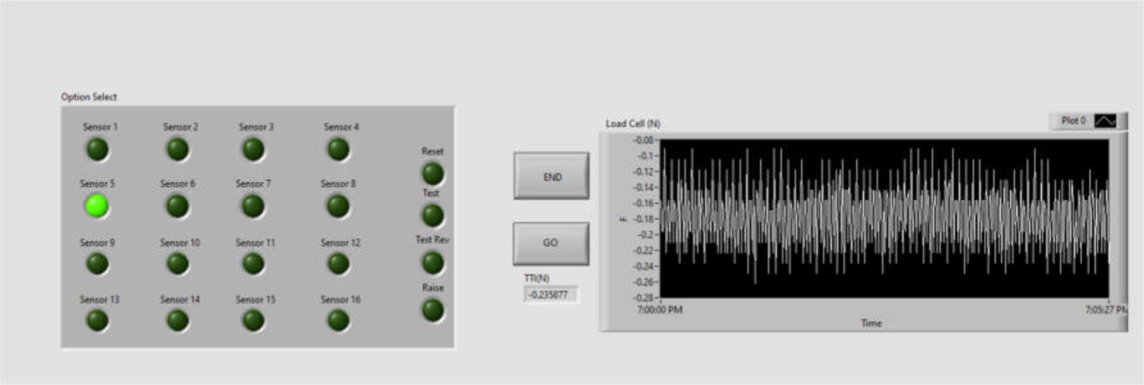
Figure 16: 4x4 SkinCell Array

LabVIEW does motor control through its Softmotion package. Softmotion is a simplified motor control scheme that allows an engineer to do motor control without having to go through standard Pulse Width Modulation(PWM) signal generation.

Position control of a motor is done by inputting the number of rotations applied to the DC motor that then moves the positioning on the linear stages. A ratio between mm and motor rotations is needed to move the 7mm positioning between the sensors. The max number of rotations calculated to move the UTM150CC 150mm is 75 rotations. Knowing that, the calculation comes out to 2mm/rotation which gives the distance between each sensor 3.5 motor rotations.

3.6 Prototype Manual Testing Suite

A manual test bench was initially created as a base that could be moved to full automation. It is easier to automate an already created task then it is to create it from the ground up. The manual script gives the test engineer a choice to move to 16 locations based on a sensor that is parallel to the testbench. After aligning a sensor, the user can freely move between each sensor and run a basic single point force test. The front panel of the Vi gives the user feedback on the current force value and which step they are currently selected on. Figure 17 and Figure 18 show the front panel and code respectively.



TTI(N) -0.235877

Figure 17: Front Panel of the manual Testing bench running

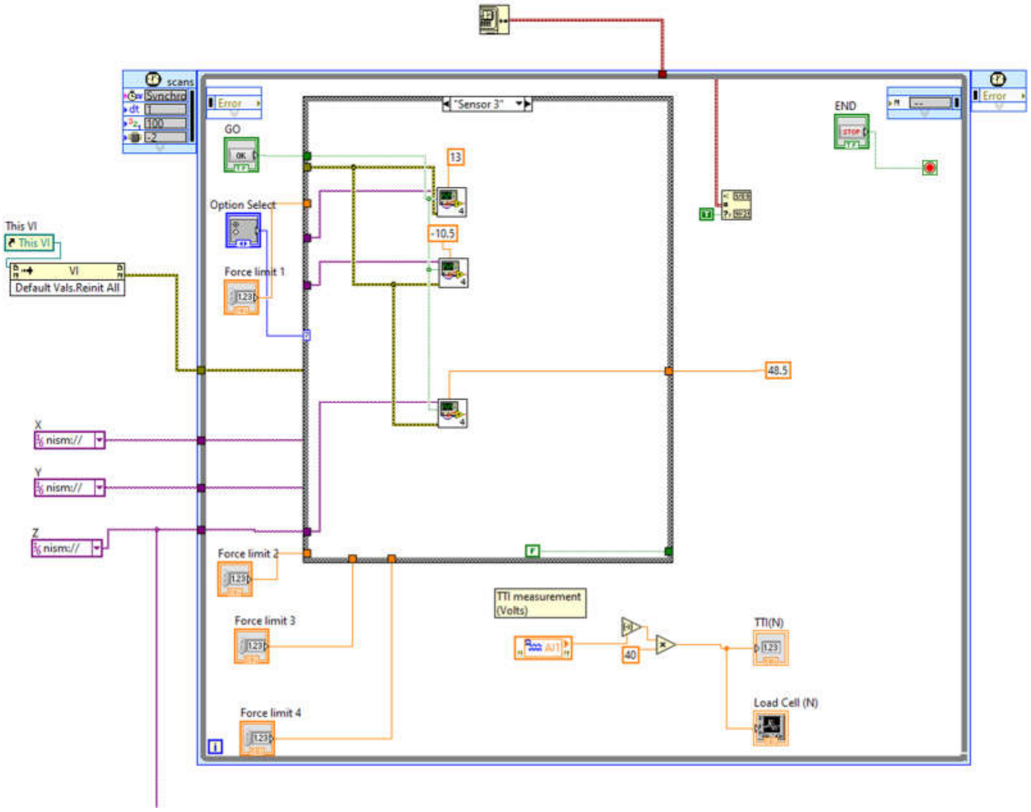


Figure 18: LabVIEW code design for Manual Test Bench

3.7 Automating the test bench

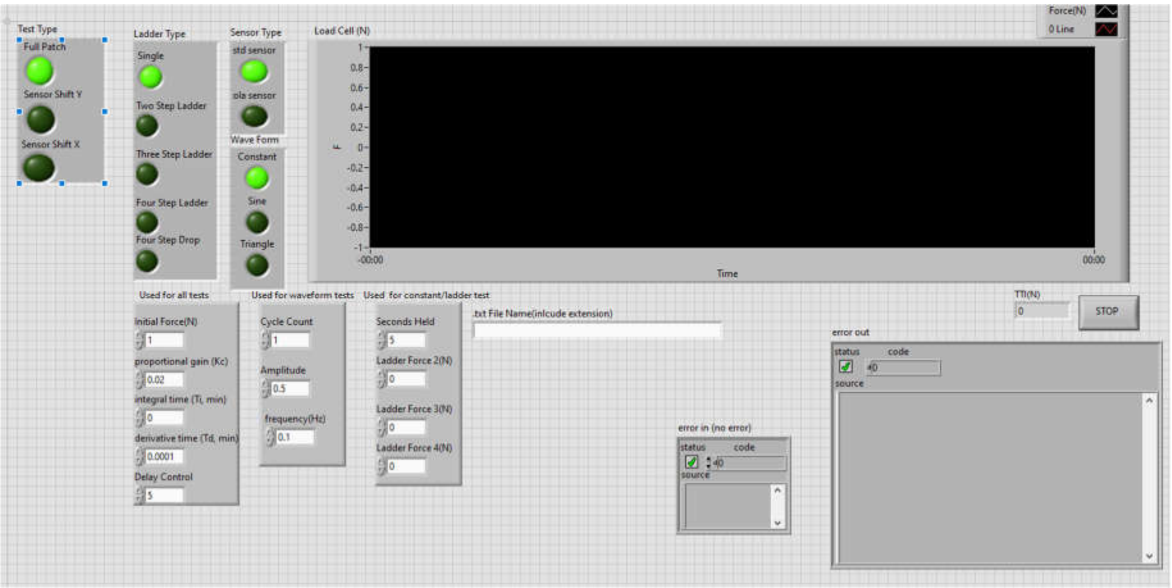
A single test can be broken down into 16 subtests. Each subtest contains the systems needed to test a singular sensor on the 16-sensor cell. Breaking down the steps needed to fill a subtest can be seen as such

1. Align X and Y linear stages to have sensor underneath force indenter
2. Run Force control algorithm over sensor to desired amount
3. Safely raise the force indenter to not damage the cell

Once the 16th subtest is finished, the motor positions reset back to their initialization points while the load cell collects 10 more seconds of data to renormalize it.

This leads a 16-sensor cell to have 49 steps to run a full testing suite. However, both the force load algorithm and the raising motion should remain consistent across each subtest. This leaves only 17 unique steps for moving to the 16 sensor locations and resetting the motor locations at the end. If the sensor is assumed to be in or close to parallel with the top portion of the testbench, the steps can be simplified since the distance between is known. The X linear stage can move 3.5 rotations over for each sensor before it gets to the 4th in the row of sensors. Then, the X stage can shift back 10.5 rotations to the first column while the Y linear stage shifts 3.5 rotations. If the sensor positions are not known, or are at an angle compared to the parallel, then 17 unique steps are generated for the sensor locations.

To automate these steps, a case structure is created in LabVIEW that is controlled by a for loop. Case structures are control subsets that execute when an outside condition is met. The count of the for-loop iteration applies itself to the case structure, which executes the code inside the subset before the for loop iterates again. Stage 0 initializes the motors and gives the load cell a controlled time to normalize data, before moving to the first sensor location. Applying the force algorithm to a sensor is done on 16 stages. To see if the structure is ready to execute a force algorithm, the current loop iteration is read in before being divided by 3, keeping the answer in integer and remainder form. If the remainder is equal to 2, then a force algorithm stage has been reached and the indenter is loaded onto the sensor. The raise command is checked similarly to the force command, except the program checks if there is no remainder before applying a safe raise. The remaining steps are needed to run individually from each other. Since the location of the coordinates of sensors can very, no remainder test is taken and instead the for-loop iteration is compared directly to a specific case structure. To simplify the amount of code being executed in the case structure, sub-vis is created to control the individual motions of each axis. Figure 19, Figure 20, and Figure 21 show the Vis used to create the automated profile.



Initial Force(N) 1
proportional gain (Kc) 0.02
integral time (Ti, min) 0
derivative time (Td, min) 0.0001
Delay Control 5
Cycle Count 1
Amplitude 0.5
frequency(Hz) 0.1
Seconds Held 5
Ladder Force 2(N) 0
Ladder Force 3(N) 0
Ladder Force 4(N) 0
.txt File Name(inlcude extension)
TTI(N) 0
error out
status ☑ code 0
source
error in (no error)
status ☑ code 0
source

Figure 19:Front Panel of the Automated Testing Software

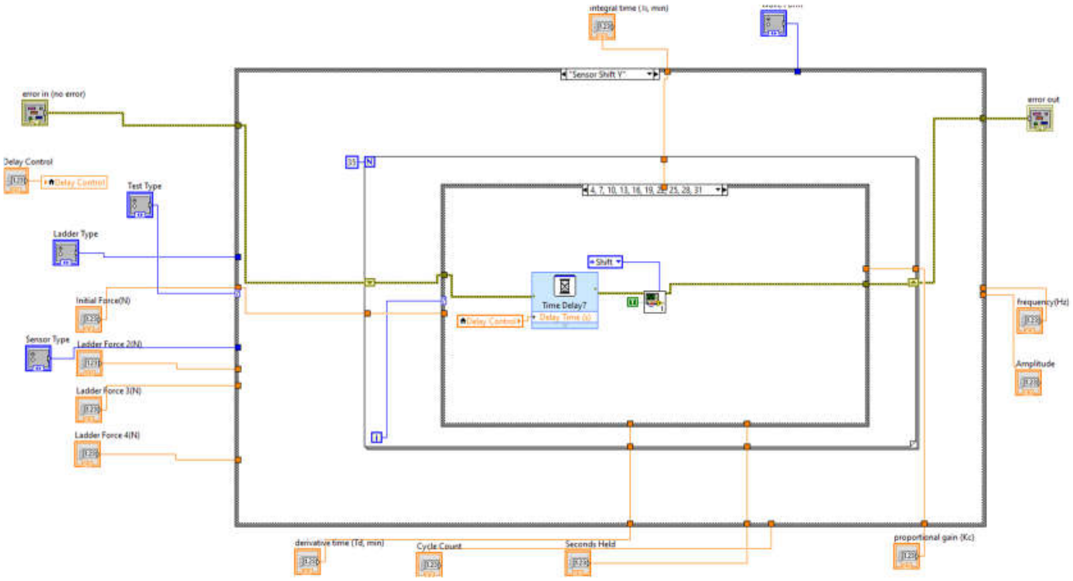


Figure 20: LabVIEW Vi of Automated Loop for Testing

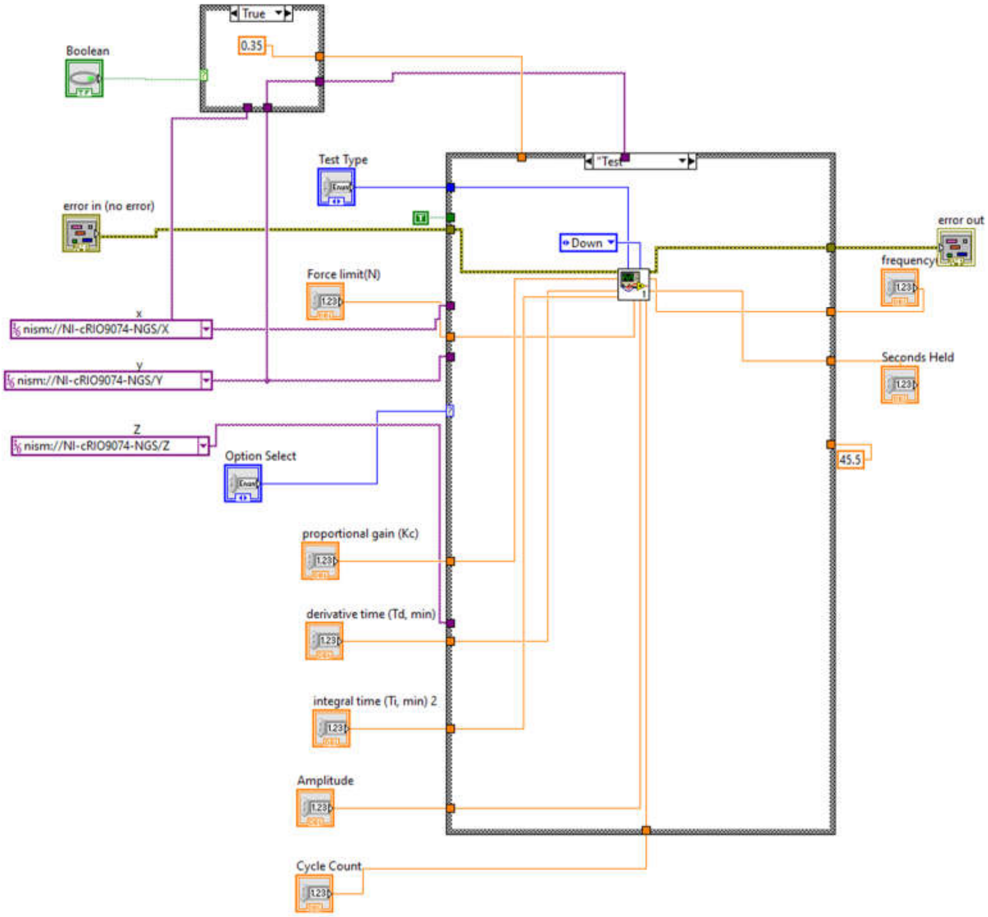


Figure 21: LabVIEW SubVI created to manage individual Automation Steps

3.8 Time Stamp Matching

A separate script developed in python is used to collect the response data of the SkinCell. Accurate timestamping becomes necessary so that the data collected from both programming environments can be aligned. The cRIO runs has its own internal clock that can produce timestamps for every force read iteration. During testing however, timestamps began to lag tremendously behind what the actual time was listed as. Going into the settings of the cRIO, it was discovered that the internal clock was completely lagging from when it started.

To update the time on the cRIO, a PC Vi is created to open a direct com between the PC clock and cRIO clock. Since the cRIO clock begins lagging as tests start running, this Vi must be always run so that timestamps generated can be accurate and data between both platforms can be aligned. Figure 22 shows the PC Vi created to keep the time of the cRIO updated.

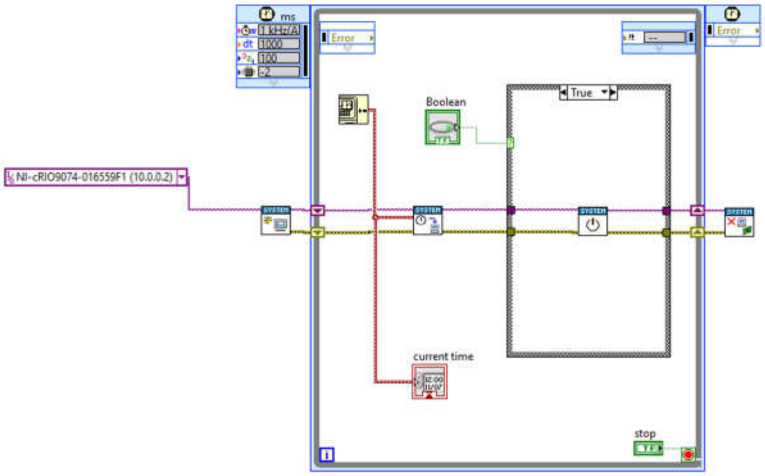


Figure 22: Timestamp Vi created to always run in the background of the RT target

3.9 Incremental Force Controller

Force control is the act of controlling an object based on the amount of force said object is applying. In the case of the testbench, the motor position or velocity of the Z stage should be controlled based on the value the load cell is generating. LabVIEW Softmotion allows motion to be done for a specific distance or at a specific velocity until a stop command is generated. The initial force controller was designed to have the Z axis linear stage run at a speed of 12.56 radians per second, which translates to 4mm/s of linear travel. Once the desired force was reached, the system would automatically generate a stop command and keep the axis in place. However, the cRIO had trouble recognizing when a stop signal occurred, causing overshoots of the desired force. The velocity controller was scrapped in favor of an incremental position-based controller.

The incremental controller was designed to read in the current load cell value then compare it to what the desired value was currently at. If the value was less then the desired value, then the motor was once again stimulated to move a specific distance. Once the force target was equal to or greater than the desired value, the motor loop stopped. Figure 23 shows a block diagram of the code while figure y shows the code represented in LabVIEW.

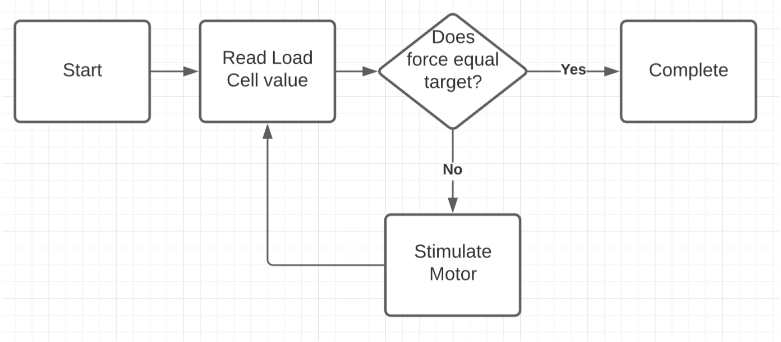


Figure 23: Block Diagram of initial motor

This method of force control is called incremental force control. Incremental force control works by moving a force actuator over small increments until it reaches a static point. The linear actuator moving just a single mm can cause the force applied to shoot up massively. To compensate for this, the motor must move at incredibly small steps so that it does not overshoot its target force. Incremental force controllers can make it to a desired force value, but take a lot of time doing so.

Additionally, the motor controller stops once it reaches the desired destination, but due to external forces the location may be wrong. External system noise or an error in the system can cause the value to be about 10% off, with no additional way to correct itself since the motor stops. Finally, this method of force control only supports singular static point testing. The only test that can be run is one that moves to its desired position then stops and waits for it to release itself. Static force control does not accurately represent what a real-world environment is applied to the sensors, which gives data that does not represent them correctly. This controller was designed to be a stop gap for publications to be done with simple force control and completely scrapped when developing future control systems.

3.10 Force Control: PID Controller

When designing the next force control algorithm, it was decided that a similar base would remain the same between load cell, algorithm, and motor control. However, the control set would offer a much more diverse set of features and move the position of the Z axis faster. A Proportional Integral Derivative(PID) controller was initially chosen as the replacement for the algorithm. A PID controller is a feedback control loop system that's primary goal is to shrink the difference between a desired output and a process variable. The loop runs continuously even when a desired setpoint has been reached so that error correction can occur. Each of the three terms represents a different type of error correction. Proportional tries to minimize the value of the current error. The integral term uses an integral from a certain time back to correct for past error. This term does not integrate from the beginning of loop time since excess values can lead to wind up, which delays the calculation. The derivative term takes the derivative of the current error function and uses it to try and predict future error via slope trajectory. The equation used to run the PID Controller can be seen below in equations 2 and 3.

$$u(t) = K_p(e(t) + \frac{1}{T_i}\int_{t_i}^{t} e(\tau)\, d\tau + T_d \frac{de(t)}{dt}) \tag{2}$$

$$e(t) = r(t) - y(t) \tag{3}$$

Where K_p is the proportional gain, e is the error signal of the difference between the desired point(r(t)) and feedback value(y(t)). T_i and T_d are the integral time and the derivative time, which are the hypothetical time it would take for the integral and derivative terms to catch up to a sudden change in the proportional term. To fully integrate this model into the control loop, the desired value is what the engineer wants the force to be, while the process variable is the current force value from the load cell.

3.11 PID Implementation and Functionality

Creating a PID controller in LabVIEW is simplistic. LabVIEW includes a built in PID control scheme that derives the output[37]. The machine does not come with built in tuning so that is still done manually. The PID function is placed in a loop structure so that it runs constantly. Attached to the process variable section is the current force value read in. The desired value is loaded with the force the engineer wants to test at and is compared. The function block outputs a motor rotation amount in absolute value. Absolute value refers to moving based on the exact motor rotation it has changed in respect to 0 instead of a value relative to its starting point. This value is fed into a

Softmotion straight line move command. Figure 24 shows the structure of this in LabVIEW

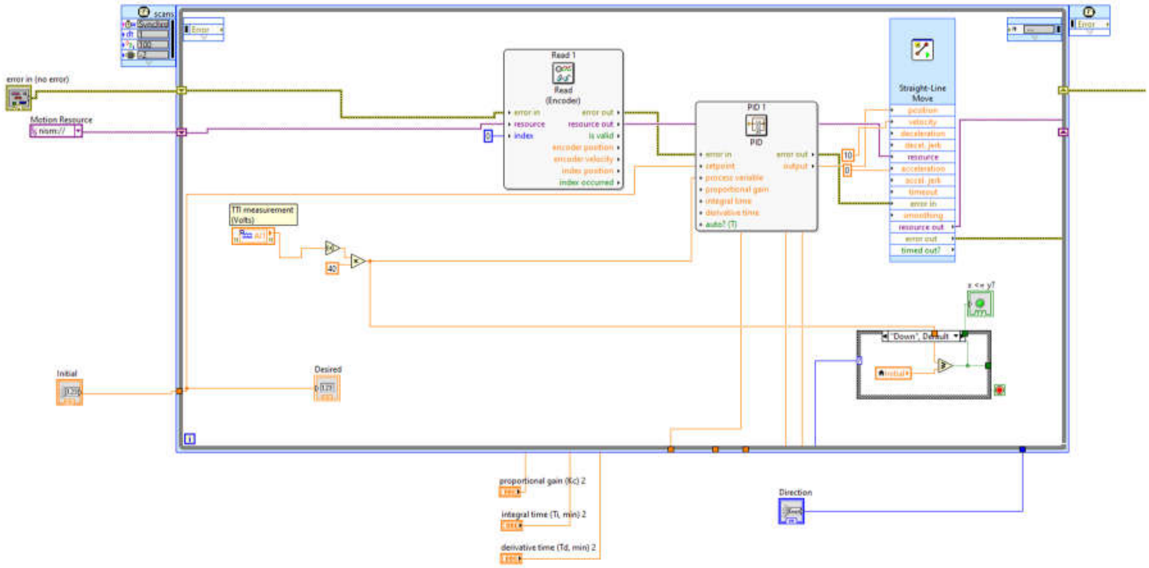


Figure 24: PID Motor Movement Loop in LabVIEW RT

To see the performance difference, the primitive controller and the PID controller were tested to apply the same force, from the same starting positions on top of a soft bedding. The force response over time would be measured to compare the time needed for both to hit 1N. Results can be seen in Figure 25.

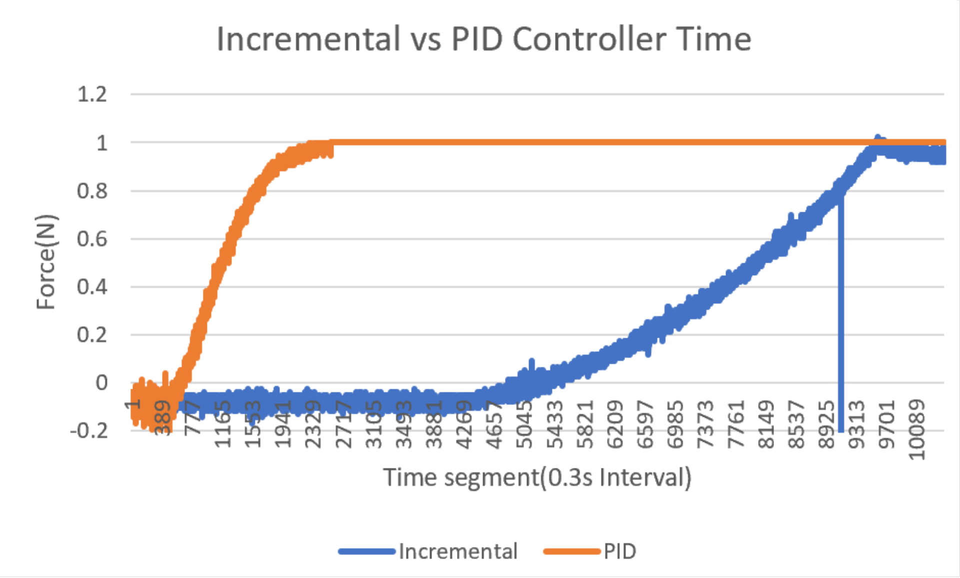
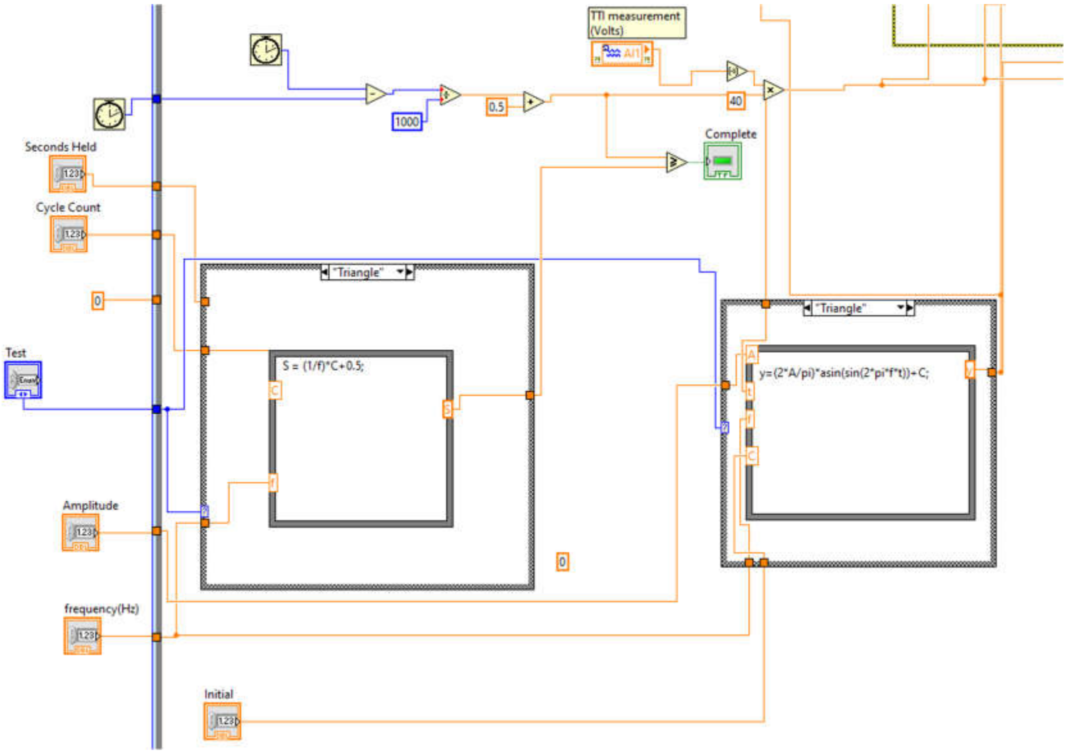


Figure 25: Plotted chart of the Incremental vs PID controller

The PID Controller is able to reach the 1N threshold in 20 seconds. Comparatively the primitive force controller took 104 seconds to reach the 1N point and stopped without any error correction. This results in 5.2 times gain in time alone, resulting in a much faster testing process as well as the other benefits.

3.12 Dynamic Load Testing

Since the PID controller is able to track a desired force, non-static forces can also be applied. In human-to-human interaction, applied force varies as perfect control over force, and the area it is applied to changes. To apply dynamic force, motor positioning should be moved in a way so that a controlled changing waveform is created. To do this in LabVIEW, the tracking point is changed to model a real time equation. LabVIEW function blocks allows the user to write out a mathematical formula without the need to build it using their blocks. The result allows for generation of these waves. Two function blocks are used. The first is to create the period that the waveform is to run for, allowing each one to be stopped after a set time. The second is for the waveform. Two mathematical equations are input into the design: A sine wave and a triangle wave. Each wave allows the user to choose the operating frequency as long as it is within the limits of the system. Figure 26 shows the LabVIEW formula blocks, while Figure 27 shows the results of a sine wave being applied.



```
S = (1/f)*C+0.5;
```

```
y=(2*A/pi)*asin(sin(2*pi*f*t))+C;
```

Figure 26: LabVIEW formula blocks used to create a period and wave

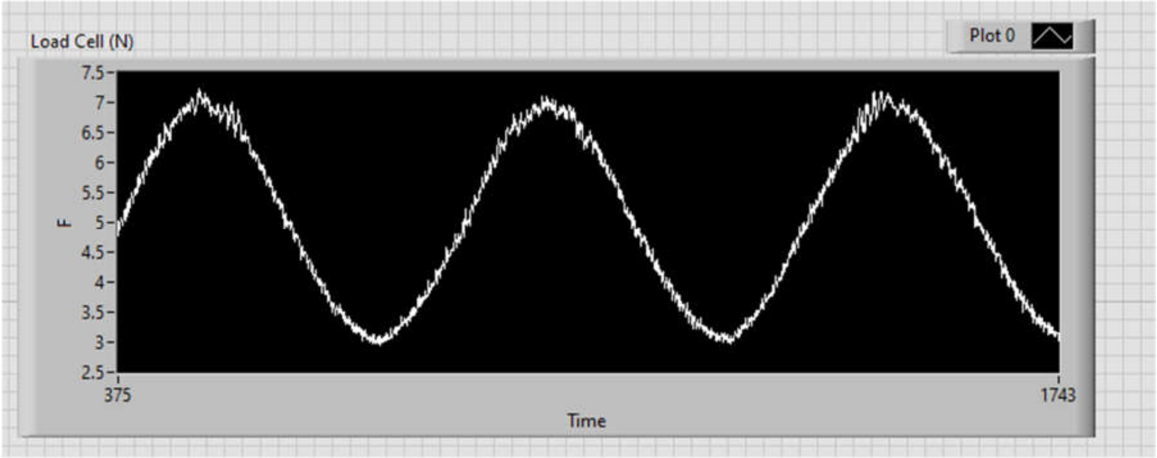


Figure 27: Applied sine wave using PID tracking

3.13 PID Force Controller evaluation and Sensor Displacement

To test the accuracy of the force controller, tests were run to determine how well the force controller tracks a desired dynamic waveform and see the difference between the exact desired setpoint and the current value of the force load cell. The first test set the desired load to an 0.1 Hz sine wave that starts with an offset of 2N and has an amplitude of 1N. 3 cycles were run to see the difference in tracking between the waveforms. The second test involved adding a triangular wave and a sine wave together to make a more stuttering motion. The triangular wave had the same specs as the previous tests sine wave(0.1 hz, 1N amplitude, 2N offset) but the additional sine wave used an 0.5N amplitude along with an initial 0.1hz frequency before adding 0.001hz for every loop cycle of the controller. Additionally, the displacement from a 0 point was also measured slightly above the SkinCell to identify the exact amount of position changed when applying these forces.

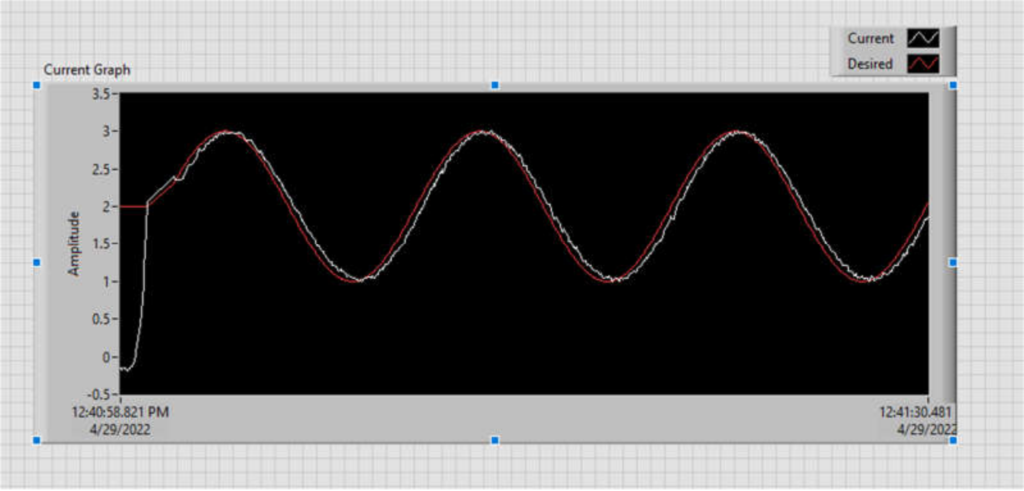


Figure 28: Sine Wave Tracking. red wave is desired while white is actual

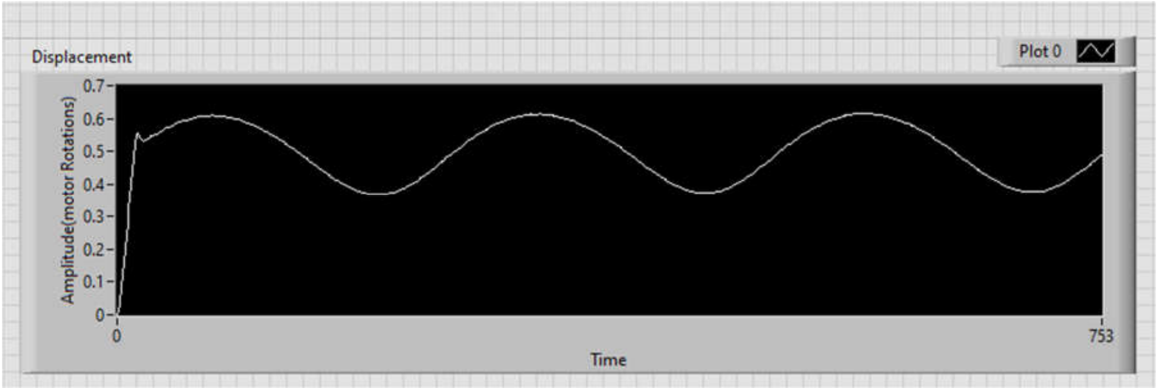


Figure 29: Displacement tracking of an applied sine waveform

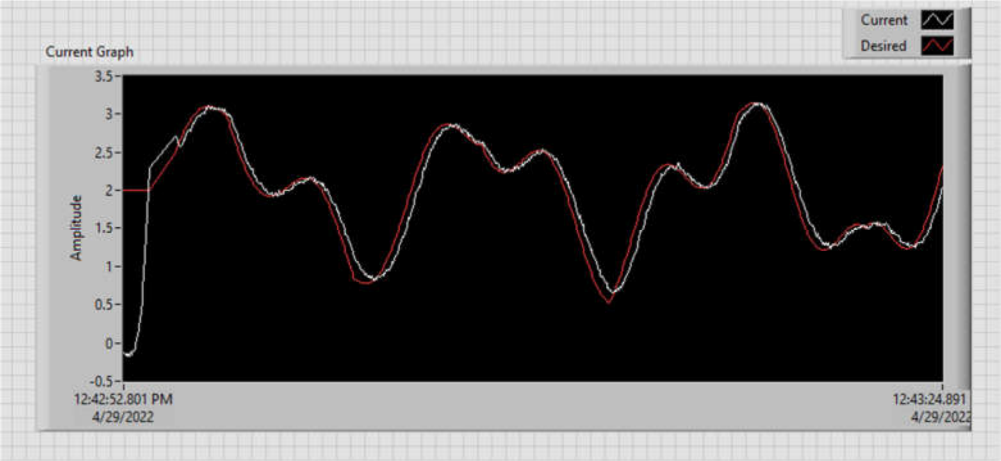


Figure 30: Triangular + Sine wave tracking. red is desired and white is actual

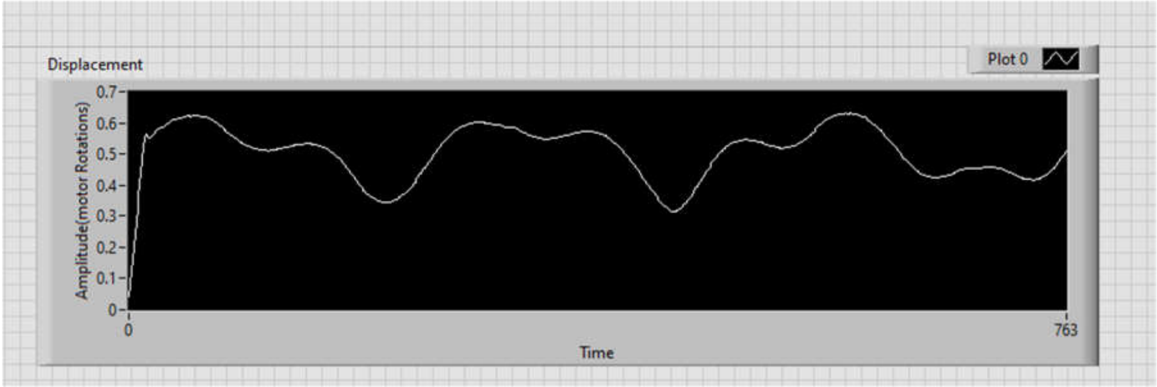


Figure 31: Displacement tracking of the Triangular + Sine wave

The resulting tracking results of the desired vs actual value shows that the Z linear stage is able to accurately track the desired waveforms wanted, albeit with a little delay in the wave to compensate for the difference between the time the calculations are done and when the movement command is issued. The measured error was average out to be 0.021 during the sine wave and 0.039 on the triangular wave. This could be attributed to the noise value of the load cell, as while small can still affect the PID controllers' calculations. Tracking the displacement allows us to see how much distance the linear Z actuator needs to move in order to apply Newtons. Going by both waveforms after the initial offset is applied, it is observed that the Newton to motor rotation ratio is 0.1 motor rotations per Newton. Converting this value to Micrometers gives us a ratio of 50 Micrometers of change required to apply a Newton.

3.14 PID Real Time Issues

The Real-Time based PID controller allows the test bench to run tests more accurately or with dynamic loads. The controller however is still without its faults. A controller needs to update with new values at a fast rate so that it can quickly recalculate its next position. The update rate for the PID controller loop is runs around 11Hz when running the entire system in motion. That update rate causes the actual value of the force sensor to lag the desired value by a significant amount. The lag can lead to a test not following its desired trajectory to completion or an overcorrection that breaks off from the control loop. This creates unrepeatable testing scenarios. A minimum update frequency of 60Hz was targeted so that the control loop can respond accordingly to changes in the force actuator.

To minimize the lag the of the control loop, a timing experiment was done by running the three individual components of the control loop singularly, then finding the difference between loop time. The results can be seen in Table 1 below

Loop Element	Period(ms)	Limiting freq (Hz)
Load Cell Read	1	1000
PID Calculations	2	500
Softmotion Command	40	25

Table 1: Timing Analysis Results of individual loop Components

Both the read in of the force actuator value from the load cell and the PID calculations cause minimal delay. The Softmotion command for a straight line-move however limits the loop frequency to a maximum of 25 Hz. With the Sub-Vi system in place, this limits the speed down more to the 11Hz seen when running the automated test setup. Increasing the speed of the refresh rate requires an optimization or workaround of the Softmotion engine used to move the linear stages.

3.15 Softmotion engine developments

Several ideas were proposed to remedy the speed of the Softmotion engine.

1. Remove the Softmotion engine entirely and work at the hardware level of the 9516 modules.
2. Optimize the loop to run calculations asynchronously from the Softmotion engine and only update when it can accept a new value
3. Remove unnecessary bloat from the Softmotion movement modules so that only the necessary code pieces are used

.

All three of these methods offer a possible solution, but not all of them were in reach. The CompactRIO also offers a Fully Programmable Gate Array(FPGA) to be used instead of the Real Time Processer. The FPGA can be programmed on a hardware level to create direct pipelines between executed code and the 9516 modules used to control the linear stages. This mitigates the final two issues but still leaves the new motor controller. FPGA code requires more complicated code then its Real Time counterpart, but if done correctly can easily fix the solution. However, National Instruments specifically hides the FPGA motion control from the user, and instead requires them to use a write-scan-read method for applying data to the 9516 modules[38]. To properly generate a motor command, a spline model of a desired trajectory is needed to guide the motor. After discussing this with someone who has done previous FPGA work with the 9516, they recommended that this method not be pursued, as NI does not entirely support this method and instead has specific modules for FPGA motor control to use instead.

Running two separate loops allows the PID controller to run at a separate rate then the calculations can work. In theory the PID controller should be outputting the optimal data needed to move onto the next location while the straight line move command is still occurring. There are some issues with this approach, however. For starters, the updated PID values could cause an overshoot and not be able to course correct due to the calculations for it occurring while the straight line move command is still moving. These overshoots and overcorrections make it incredibly difficult to follow a dynamic waveform precisely, causing unrepeatability in the system. Another issue stems performance on the cRIO. Running two separate time structures previously has caused issues with both logging data with inaccurate time stamps and updating at a fast rate. Adding a third loop would complicate the system more and in practice has been seen to fall behind in performance. Due to these issues, this method was also not pursued.

The final method involves altering the movement command used for Softmotion. LabVIEW does offer alternative ways to move a motor without the use of straight-line move. The method Write.vi is a Softmotion module that writes data to a motor axis and allows positional override to occur. This allows our data to overwrite the second the PID is done calculating, resulting in up to 300Hz.

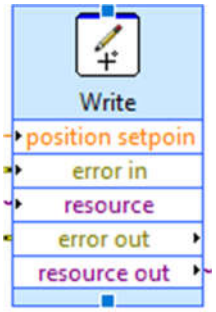


Figure 32: LabVIEW Softmotion Write ExpressVI[39]

Softmotion Write also comes with its own setbacks. Position overwrite allows the motor to constantly move to its desired location. But to do that, the overwrite command pushes the motor try and get to that position as fast as possible. Acceleration curves are not used and instead the motor moves as quick as it can to correct its new position. When trying to have the motor settle in on a point, the sudden changes cause a ringing effect to occur when trying to stabilize. This ringing can cause a shift in the force applied upwards of 0.2N in both the positive and negative direction of the desired target, and the PID is not able to compensate for this instantaneous change. Figure 33 demonstrates the ringing effect that occurs from this.

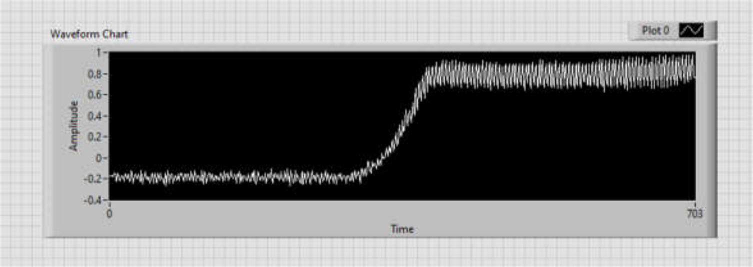


Figure 33: Ringing effect from the Softmotion Write command

The write command did not fully solve the problem, but it did offer insight into another solution. Creating and override program for the straight line move command would allow us to keep using the Vi and possibly move it to a faster speed[40]. To do this, the straight line move command was opened up and broken down into individual steps.

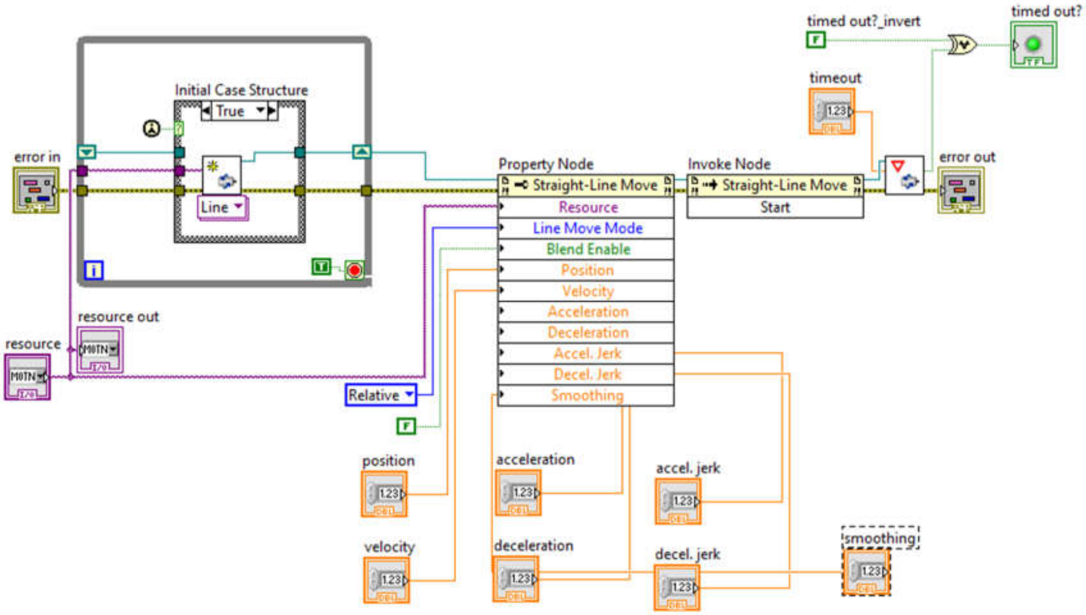


Figure 34: Straight Line Move Vi individual parts

The Vi starts with an initialization phase before loading the data in and sending it over to the 9516 modules. Afterwards, a wait until command is done, which does not allow any other Vi components to run until the entire process is completed. The wait until command is what is causing the time loss, as it has the motor control run through it's entire deacceleration state and check. This method is locked behind NI copyright, and requires employee access in order to edit the internals. However, it is possible to completely get rid of it instead. By deleting this method, the wait process no longer occurs and instead a stop command can be placed. This removes the de-acceleration period, but that it not required for the system. The ringing in the previous write subVi was caused by not a lack of de-acceleration, but a lack of acceleration of the motor. The instantaneous movement made it so overcompensating was required when going to the next spot. By keeping the straight line moves acceleration curve, then manually setting the clock refresh rate so that it runs at a given rate slower than the max, a controlled ramp is created which gives a faster loop rate time without the ringing.

3.16 Velocity Control

Besides position control, velocity control came up as an option to use. The same Softmotion write Vi has the ability to overwrite velocity as well. In theory, the velocity fed in from the PID controller should be able to follow along an applied waveform by changing the velocity. In practice, the linear stages responded to the velocity control to an extent. However, the motor amplifier setup currently is done with position control in mind. The wiring will have to be changed to have the motor amplifier produce the correct signal. Additional testing is done to verify this method.

CHAPTER 4

SENSOR COORDINATE PROFILING USING COMPUTER VISION

Controlled force algorithms and automation do not matter however if the sensor locations are unknown on the cells. The system must be able to learn where the location of every individual sensor is on a cell to test thoroughly. In addition, the system must use this information to generate a motion profile in an efficient manor to not halt work done by the engineer.

This work proposes the use of a fixed-point camera that can acquire images of the testing area with a SkinCell present. Then, using LabVIEW image processing techniques, extract sensor pixel locations. Once pixel locations are taken, the coordinates are converted to distance between the force actuator and the sensors. Finally, the distances are converted from pixels to motor rotations before being sent to the Real-Time Controller. Figure 35 shows the data flow model of the entire system including a Camera connected to LabVIEW 2014 via micro-USB.

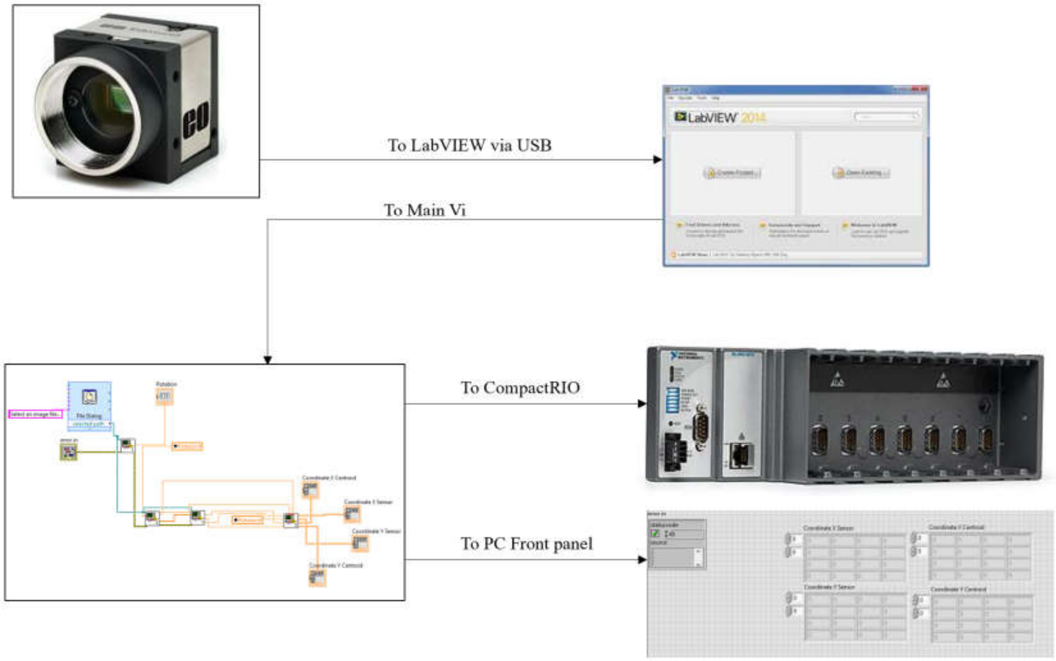


Figure 35: Data flow between Camera and Real-Time controller including LabVIEW

4.1 Camera and mounting equipment selection and setup

In this setup, the camera remains stationary at a fixed point. The testbench moves to acclimate to the sensors predetermined Field of View(FOV). Keeping the camera stationary over the rest of the bench benefits the autonomy of the system.

For this project, the Edmund Optics EO-1312C 1/1.8" CMOS Color USB Camera is selected. The camera is connected over a USB type 3 connection and gives images at a resolution of 1280x1024 pixels, exceeding the necessary requirement for this project. The camera also supports the RGB32 format which lets the script be developed with base color in mind. Furthermore, the camera comes in a small volumetric package of 34 x 32 x

41.3 mm. This allows us to easily mount the camera without worrying about weight and size. The camera can be seen in Figure 36[41].

Figure 36: Edmund Optics EO-1312C 1/1.8" CMOS Camera

The position of the mount is important to the project. The camera must not interfere with any of the linear stages, while still being in a position to get the information it needs. The X and Y linear stages take up most of the room on the testbench table, ruling out the possibility of building a mount on the table. The camera is mounted to the same metal bar that the linear Z stage is mounted on. Mounting on the metal bar fastens the camera system in place. The camera captures a view straight down over the initial position of the X linear stage. The Y linear stage then shifts back 100mm to move the sensor starting point underneath the cameras FOV.

A mount was created to attach the camera to the metal bar. A Thor Labs XT66P2 was used to attach the mount to the metal bar. On top of that is a Newport 423 Series manual linear travel stage. The manual travel stage shifts the mount horizontally along the metal bar. This allows FOV calibration to be done if needed. Attached to that linear stage is a Newport 433 vertical manual linear stage. The linear stage sets the height of the camera which is used to change the depth of the FOV. Finally, a screw mount for the camera is 3D printed to attach the camera to the vertical linear stage. Figure 37 shows the mount over the setup.

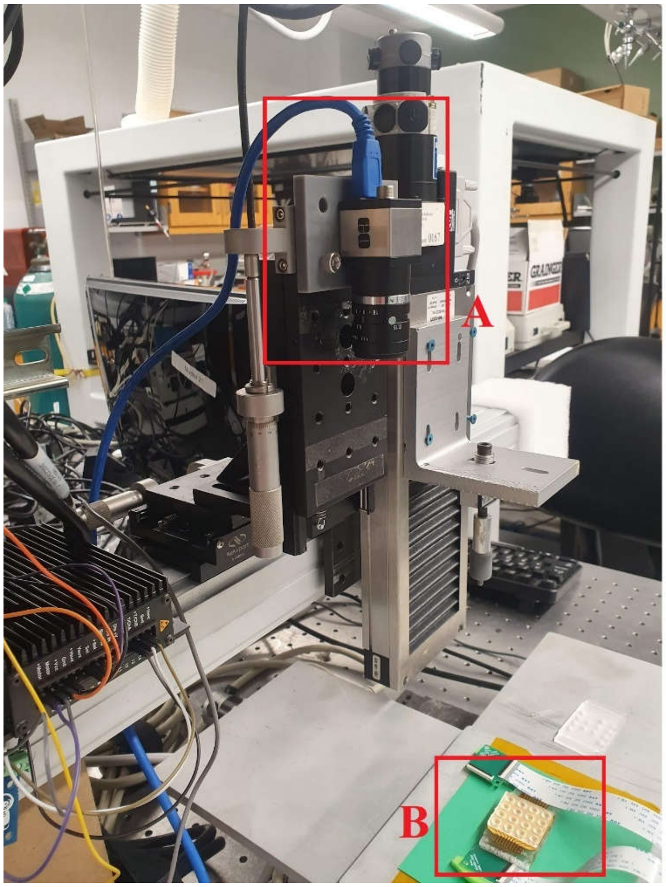


Figure 37: A) Camera connected to mount B)Sensor underneath camera

4.2 Software Design and Coordinate Extraction

4.2.1 Creating a standard image framework

When working with image processing for data extraction, it is important to have a similar image type for the script to work with. Having a constant image framework allows the script to be TaylorMade for the task.

The following prerequisites are needed for an image to be used as a candidate for data extraction.

1. The image includes the SkinCell after it has been setup for testing.
2. The image includes a silicon cover overlaying the SkinCell. The silicon cover must include an indention shape for the script to recognize, and to provide better application of force in tests
3. The background color of the image must contrast with the cell, cover, and data clip attachments.

The image seen in Figure 38 manages to meet all the prerequisites of the image framework stated above. Meeting every prerequisite allows the script to work most accurately and create the best sensor coordinate array. Development of the script was done in NI Vision Assistant, a software specifically designed to incorporate CV into LabVIEW[42].

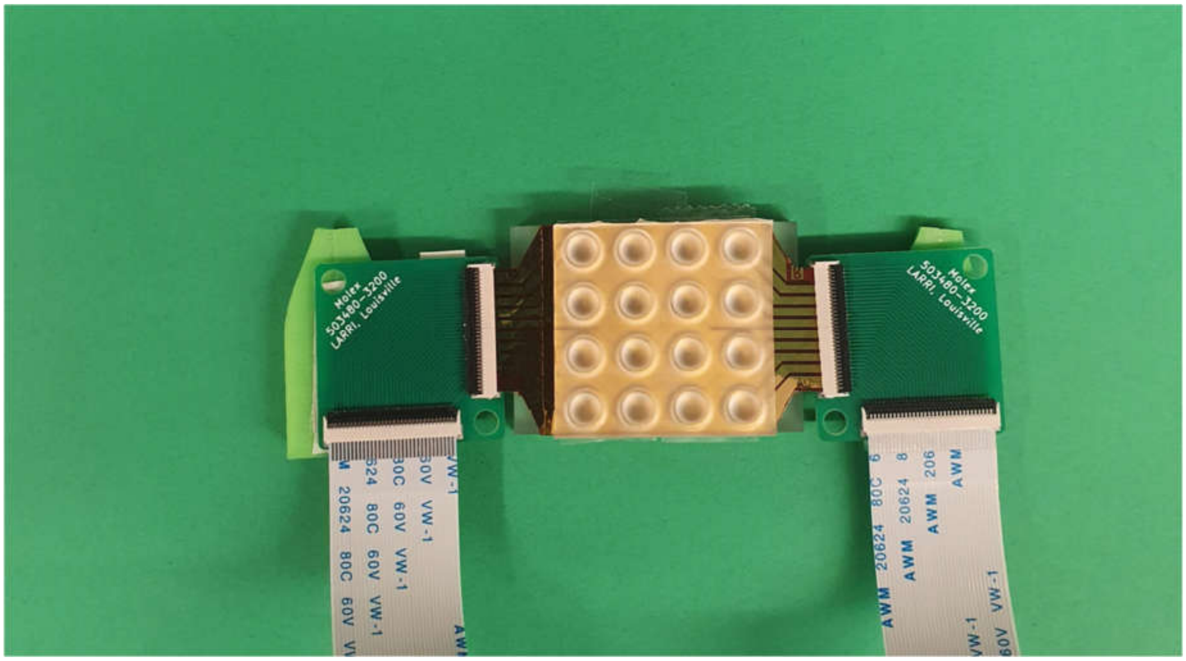


Figure 38:Image that meets all the prerequisites of the image framework

4.2.2 Acquiring the image in LabVIEW

The images are taken using the included Edmund optics software, uEye cockpit. The software automatically optimizes the camera to work at its best settings and gives noticeably sharper images then the windows camera app. Once a photo is taken, it is saved to the LabVIEW project directory over the same location. The current test photo is only important to the current run, so once the test is done it can be rewritten over as to not take up excess storage for pictures. The LabVIEW file then scans the image in and proceeds with it.

4.2.3 Image processing techniques

After taking the image, it is not possible for the computer to immediately tell where all the relevant parts of the image are located. The base image represents an incredibly complex array of pixels that cannot be easily worked with. To get an image in a state that we can extract the information needed, it must be altered in a way so that the computer can isolate the parts of the image that are needed.

The following techniques are used throughout the computer vision script to help modify the image to the point where necessary data can be found with certainty.

1. Gray scaling
2. Binary filtering
3. Object Morphology
4. Image Masking
5. Shape Pattern recognition

4.2.3.1 Gray scaling and the RGB spectrum

Digital images are made up of x by y pixel matrixes. Every pixel is a assigned a color value, and the individual pixels combine to create a digital image. Colored images such as the one in figure 2 have their individual color value created using the RGB spectrum.

The RGB color spectrum is a color model that uses the primary colors of light, Red, Green and Blue, to create different colors. The different values of color intensity are added together and can combine to produce a spectrum of visible colors. In digital images, the RGB324 standard is used for individual pixel coloring. The RGB24 model allocates 8 bits to each color to represent intensity of light giving 256 intensity values per color, making over 16 million color possibilities. All these possibilities of color require large amounts of computational power to classify objects in images. Instead, images are converted over to a grayscale format.

A grayscale is an image that does not use a color spectrum. Instead, the value of a pixel is represented by a single value that corresponds to the intensity of light at its location. Grayscale images have less data per pixel, which can simplify computer vision algorithms. The image in Figure 5 shows the prior image in Figure 39 after converting over to grayscale by removing a color plane.

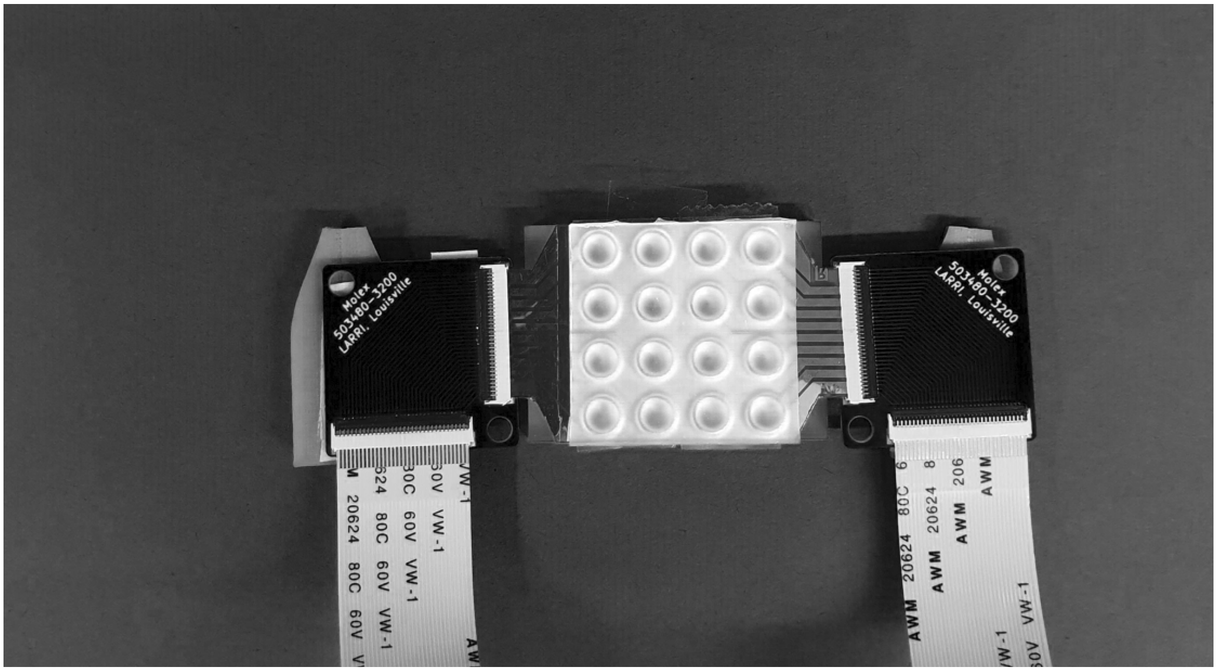


Figure 39: Prior image converted over to a grayscale equivalent

4.2.3.2 Binary Filtering

Once an image is converted over to a grayscale equivalent, the pixel values can be represented in a singular byte. To focus on parts of an image with a certain value of light, a threshold can be applied to the intensity values. A binary threshold filters out particles that do not meet within the desired threshold, removing what is deemed unnecessary. Figure 40 shows the prior image after going through a binary threshold that removes all intensity values that are lower than 98 on the intensity spectrum.

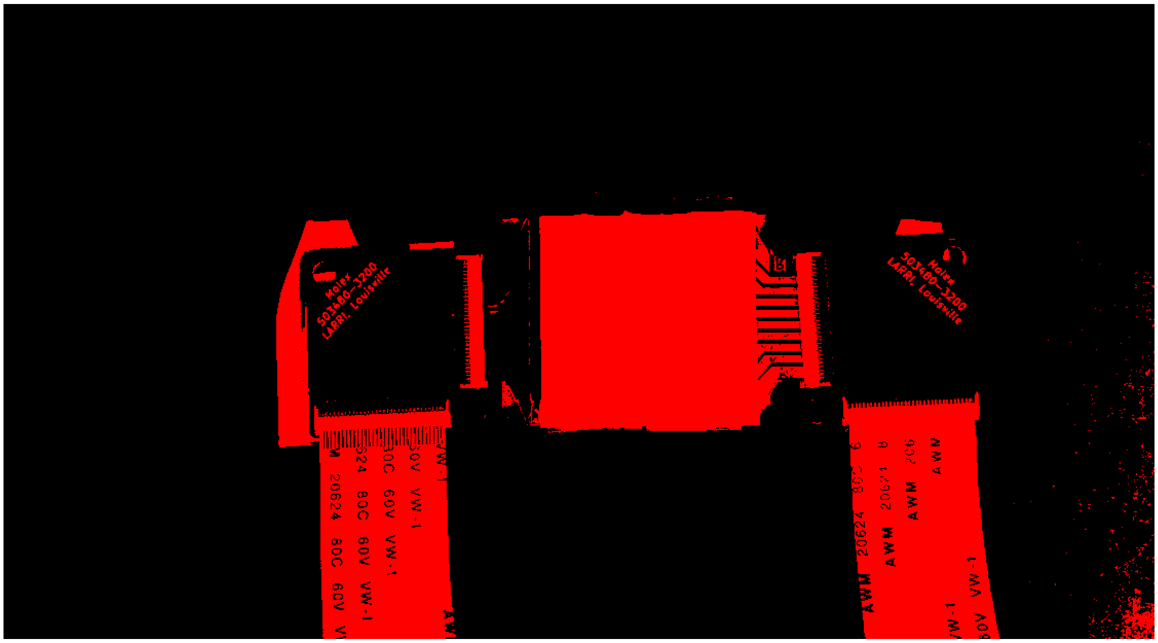


Figure 40: Image after passing through a binary threshold filter

4.2.3.3 Object Morphology

After determining the correct binary filter that includes everything needed, the image can still possess excess objects outside of what we are looking for. Morphology techniques are used to help get rid of remaining pieces and get the final object needed. Returning to figure 6, object morphology can be performed to isolate the SkinCell object so that we can analyze it.

First, objects that are connected to the border are removed. Since our point of focus is on the middle of the image, we can get rid of everything attached to the edges. Then, small objects are removed from the picture to get rid of the remaining isolated light dots. After, the image is slightly eroded. Image erosion takes objects from the outside and slowly eats away. This is useful as it can get rid of the connector that is partially attached to the sensor. Afterwards, we can remove slightly larger objects so that all that remains is our SkinCell. Next, holes in the remaining objects are filled in. This makes it so that a singular object is all that remains in the image. Finally, since all we have is the location of our SkinCell, we can perform an analysis on it, and extract the needed data to generate a coordinate plane. Figure 41 shows the image that as in figure after it has gone through the steps of object morphology.

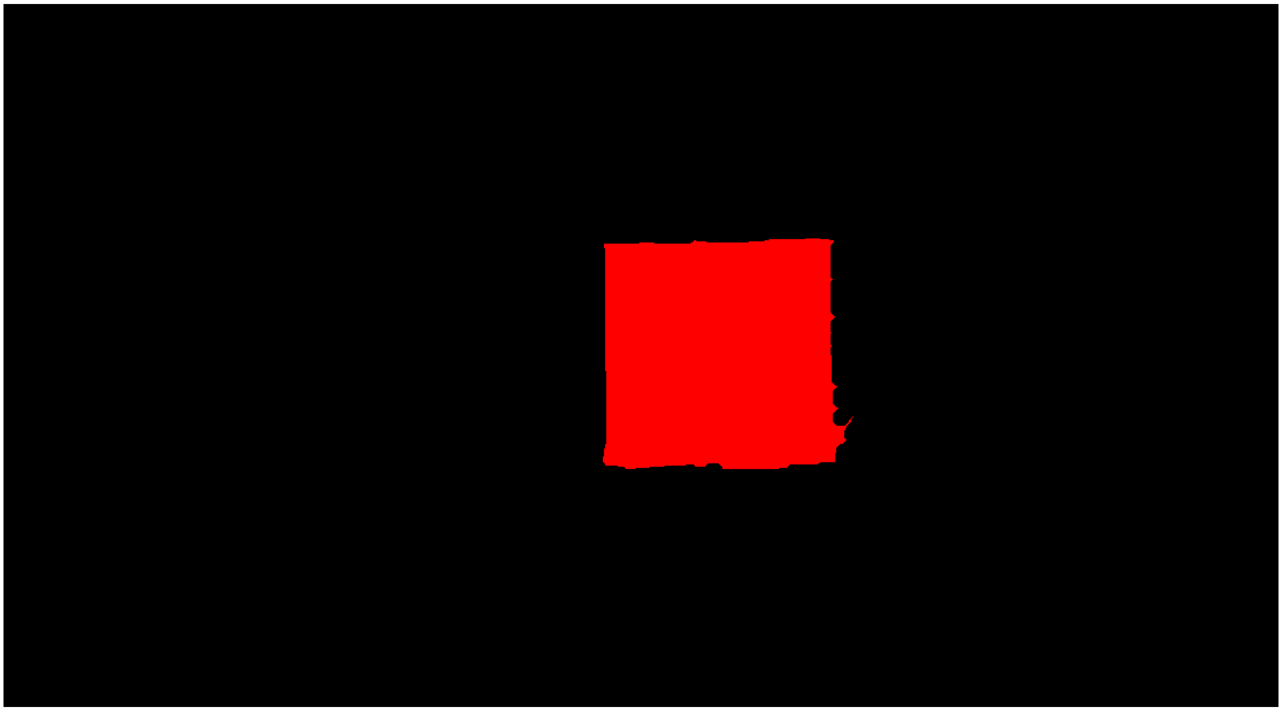


Figure 41: Image after object morphology, isolating the SkinCell from the rest of the image

4.2.4 Information needed to extract from image

After acquiring the image in LabVIEW, it is taken through a slew of post processing steps to classify key objects that are relevant to generating a sensor coordinate grid. To generate that grid, the following information is needed from the image.

1. Angle of deprivation

2. Sensor cell center location
3. At least one sensor location

The angle of deprivation is the angle difference between the left data clip attachment and the right one. The angle is used in

4.2.4.1 Main Vi.

The main computer vision script Vi consists of the LabVIEW camera interface to acquire images, and four sub-Vis in a serial connection. The first sub-Vi is used to find the location of the SkinCell. Both the centroid of the SkinCell and a bounding rectangle of the cell are found and labeled. Both points are used in the next sub-Vis

The second sub-Vi is used to extract the angle of deprivation. The bounding rectangle of the previous Vi is used to create an image mask of the SkinCell. Then, a straight-line detection profile is run across the top of the SkinCell. Once generated, the starting and ending points are used to calculate the angle of deprivation using equation (4).

$$\emptyset = \tan^{-1}\frac{dY}{dX} \tag{4}$$

Where $\emptyset$ is the angle of deprivation, and dY and dX are the vertical and horizontal changes of both center of masses. The third sub-Vi is used to locate the best-looking circular edge on the cell and the location it is at. The final sub-Vi takes the information it has learned and generates a 16-sensor plane with the locations of each sensor. The main Vi can be seen in Figure 42.

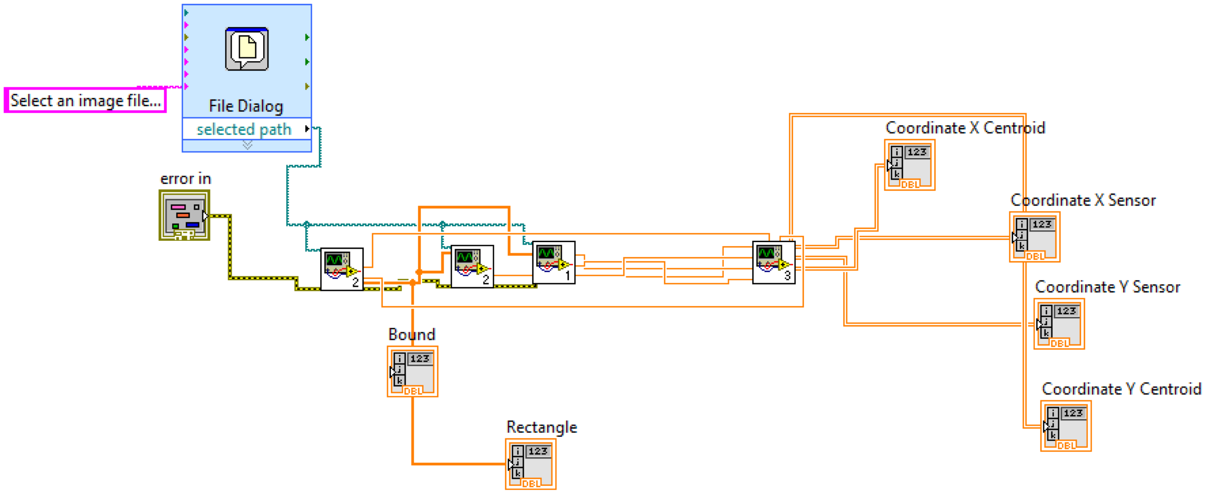


Figure 42: Main sub-vi for computer vision script

4.2.4.2 Sensor Cell Location

The angle detection sub-Vi begins by creating its own reference of the acquired image from the camera, then begins running through post processing techniques. The image is changed from a full color RGB to grayscale by removing the Hue color plane. Once it has been removed, a binary threshold is set to filter out unwanted light. Afterwards, particle morphology is performed on the new image to fully isolate the pixel area of the SkinCell from the rest of the image. This can be seen in the above figure 7. Particle analysis is performed on the object and two points of data are extracted, the center of mass coordinate and a bounding rectangle of the cell. Both are used to make separate coordinate systems. The SkinCell location Vi can be seen in Figure 43.

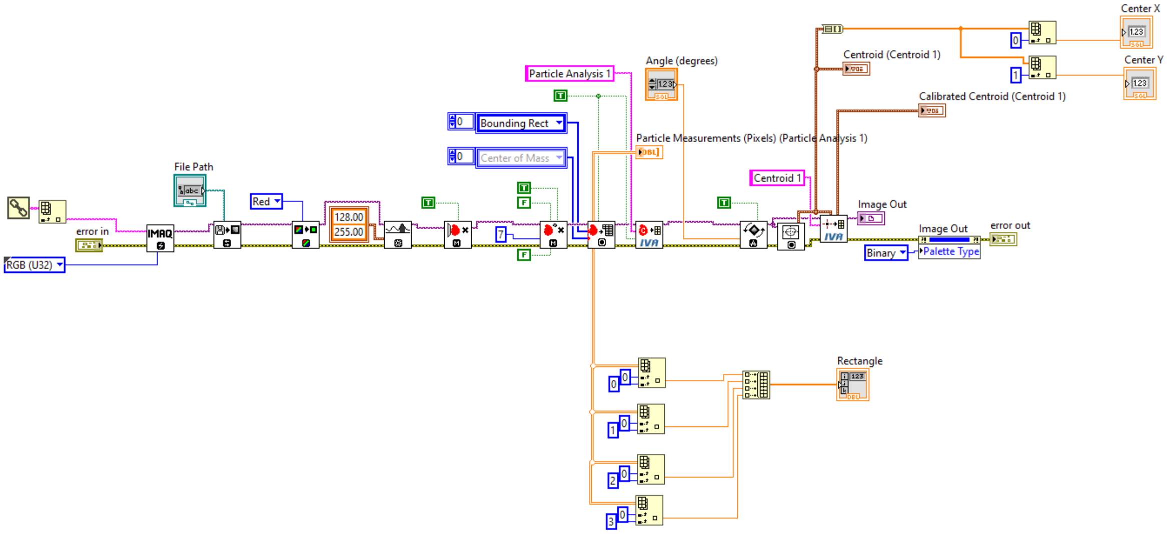


Figure 43:SkinCell isolation Vi

4.2.4.3 Angle Detection Sub-Vi

The angle detection sub Vi begins by creating its own reference of the base image. Since LabVIEW does permanent change to an image, references need to be created to alter an image differently. The bounding rectangle from the previous Sub Vi is transferred over to find the location of the SkinCell. A straight-line detection algorithm is run with line going from bottom to top over the top half of the bounding rectangle. Figure 44 shows the algorithm finding the top edge of a tilted SkinCell.

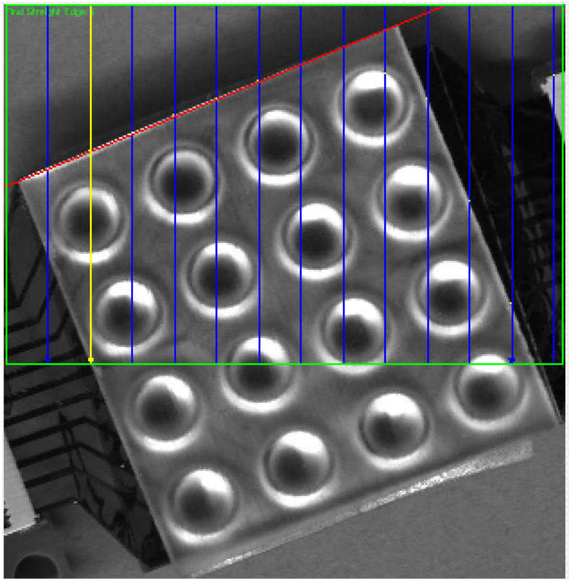


Figure 44: Edge Algorithm used to find the top edge of the SkinCell

Object analysis is then performed on the line generated across the top of the SkinCell. The silicon cover over the top is designed as a square, so the angle across the top is consistent with the rest of the cell. The beginning and ending pixel points are taken of the line, and the angle is found using the previously stated equation 1. Figure 45 shows the Angle detection sub-Vi.

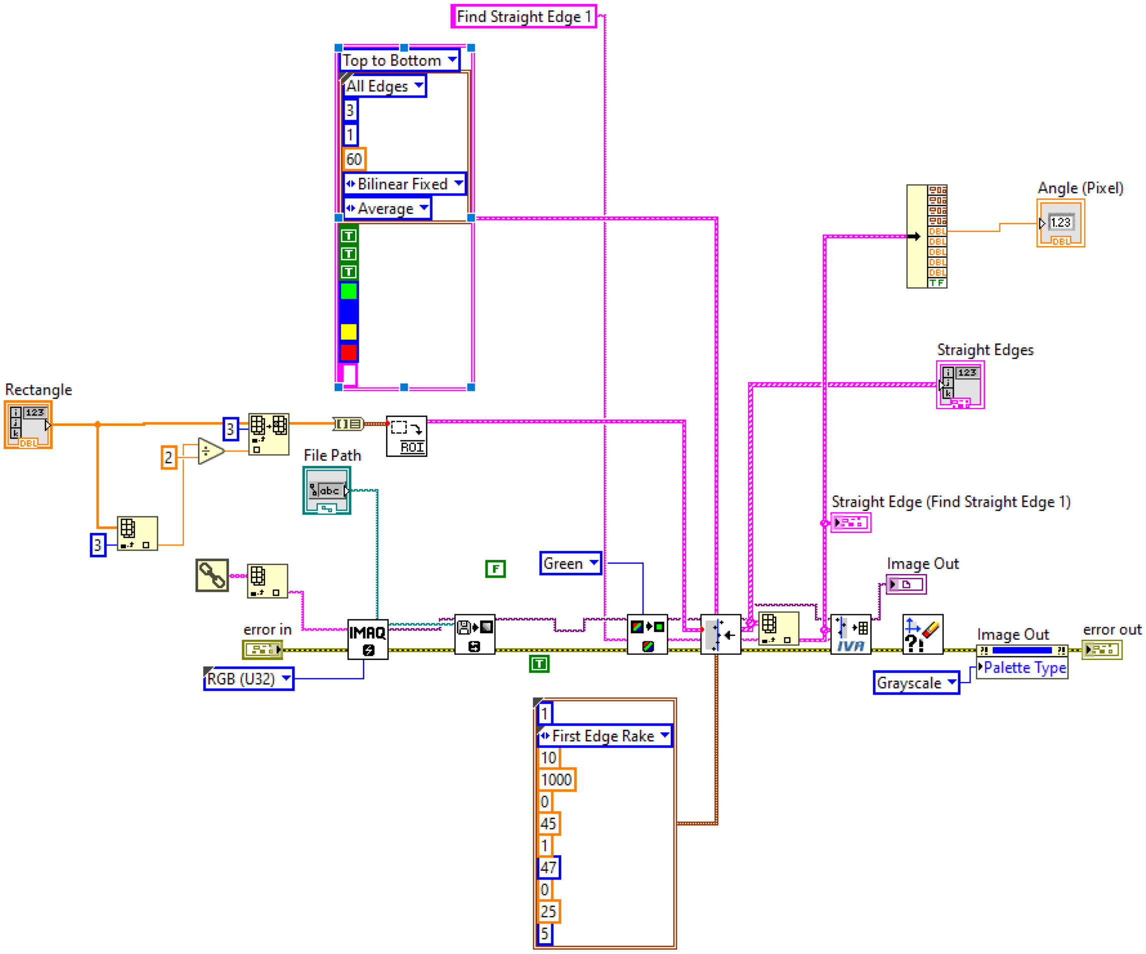


Figure 45: Angle detection Sub-Vi

4.2.4.4 Circular sensor detection sub-Vi

The previous bounding rectangle generated from the sensor SkinCell is transferred over to the current sub Vi so that an image mask can be generated of the bounding rectangle. The image mask is generated over the original image creating a focus of only the SkinCell. A circular edge test is run around the image afterwards. The test searches for what it can find as the best circular edge in the picture before having it ready for analysis[43]. A post circular edge test can be seen in Figure 46.

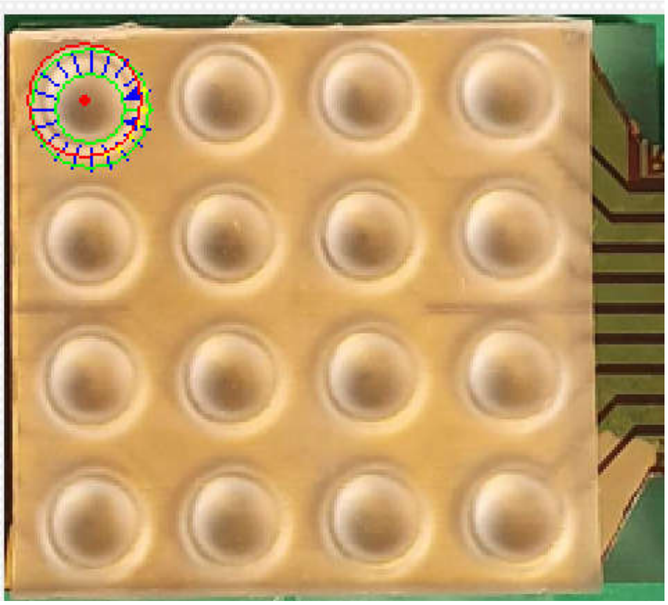
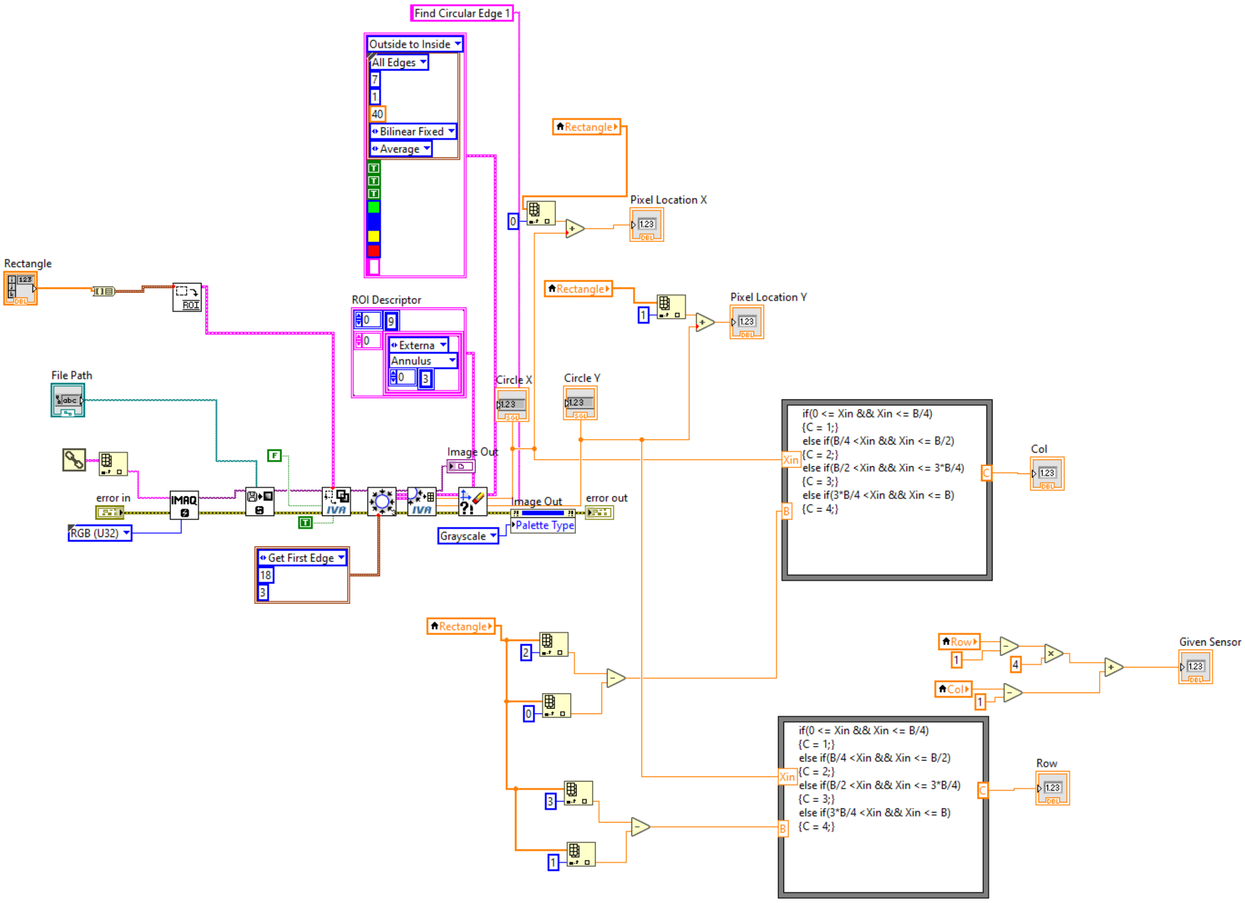


Figure 46: Image mask post circular edge test

Since this type of circular edge test focuses on finding the best-looking circle throughout, it only gives us at most one sensor and not which number sensor it is. To find out which sensor has been chosen, a 4 by 4 grid is generated based on the length and width of the image mask. The sensor is then found by comparing where the x and y coordinates land in the pixel grid before being transferred to the next part of the program. Figure 47 shows the full sub-Vi.



```
if(0 <= Xin && Xin <= B/4)
{C = 1;}
else if(B/4 <Xin && Xin <= B/2)
{C = 2;}
else if(B/2 <Xin && Xin <= 3*B/4)
{C = 3;}
else if(3*B/4 <Xin && Xin <= B)
{C = 4;}
```

```
if(0 <= Xin && Xin <= B/4)
{C = 1;}
else if(B/4 <Xin && Xin <= B/2)
{C = 2;}
else if(B/2 <Xin && Xin <= 3*B/4)
{C = 3;}
else if(3*B/4 <Xin && Xin <= B)
{C = 4;}
```

Figure 47: Circular edge detection Sub Vi

4.2.4.5 Pixel to Motor rotation conversion.

The pixel locations do not mean anything if they cannot be converted over to a usable data format for the motor. LabVIEW motion is designed with inputting motor rotations as an absolute distance. The linear stages used in this test bench travel 2mm per rotation. A pixel per millimeter value can be used to convert from pixels to mm to motor rotations. To calculate pixel per millimeter, a piece of clay was indented at two separate locations. Both indentations shared the same x axis values but were 10mm apart vertically. An image of the clay indention can be seen in Figure 48.

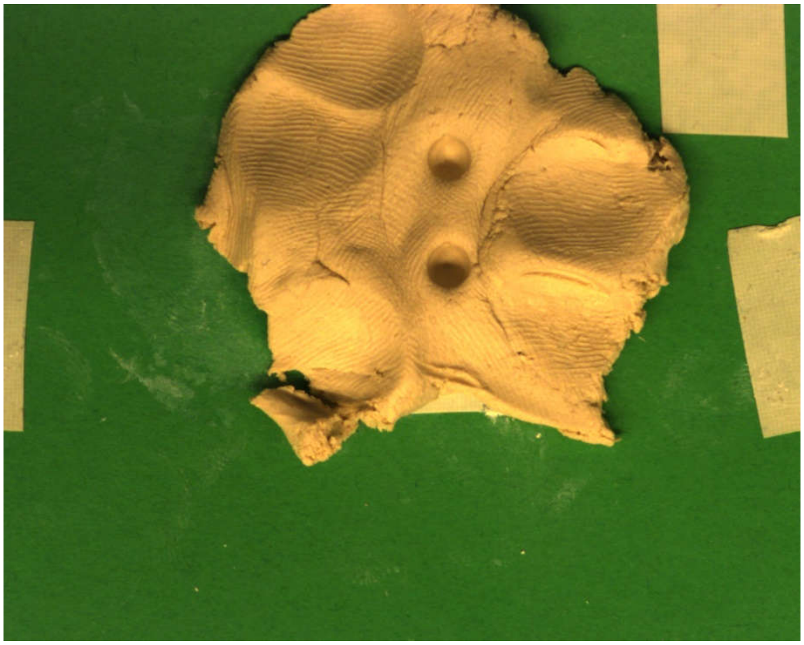


Figure 48: Clay indentions to extract pixel per millimeter ratio

Measuring the distance between the center points of the indentions gives a length of 175 pixels and gives 17.5 pixels per millimeter and 35 pixels per motor rotations. Using this value, the Starting X and Y motor rotations can also be calculated. The leftmost point of any picture taken is 20.1428571428 rotations while the highest most Y point is -24.9714285714 rotations

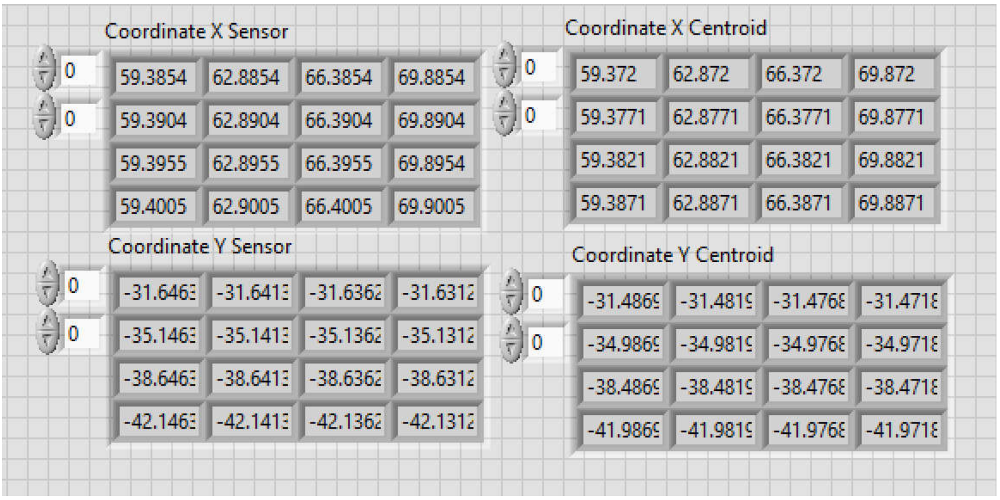
4.2.4.6 Coordinate generation sub Vi

The final step of is the coordinate generation based off the previous data. Two sets of coordinates are generated. The first set of coordinates is based of the center of mass location of the SkinCell found in the SkinCell location Vi. The second set of coordinates is based off the circular edge selection. The circular edge test coordinates result in more accurate sensor locations when working properly. However, in case of it not functioning properly, the second grid of coordinates are created from the center of mass as a redundancy check. Equations (5) and (6) are used to calculate the X and Y coordinate positions.

$$X = X_i + 3.5(C - C_i)\sin(\emptyset) + 3.5(R - R_i)\cos(\emptyset) + a \tag{5}$$

$$Y = Y_i - 3.5(C - C_i)\cos(-\emptyset) - 3.5(R - R_i)\sin(-\emptyset) + b \tag{6}$$

Where X$_i$ and Y$_i$ are initial positions, C and R are the row and column of the sensor coordinates, C$_i$ and R$_i$ are the row and column of the initial position, a and b are the minimum motor rotations captured by the image(a is 20.14 and b is -24.97), and Ø is the angle calculated from the initial step. Figure 49 and Figure 50 show the resulting coordinates generated as well as them mapped to a sensor.



Coordinate X Sensor

59.3854	62.8854	66.3854	69.8854
59.3904	62.8904	66.3904	69.8904
59.3955	62.8955	66.3955	69.8954
59.4005	62.9005	66.4005	69.9005

Coordinate X Centroid

59.372	62.872	66.372	69.872
59.3771	62.8771	66.3771	69.8771
59.3821	62.8821	66.3821	69.8821
59.3871	62.8871	66.3871	69.8871

Coordinate Y Sensor

-31.6463	-31.6413	-31.6362	-31.6312
-35.1463	-35.1413	-35.1362	-35.1312
-38.6463	-38.6413	-38.6362	-38.6312
-42.1463	-42.1413	-42.1362	-42.1312

Coordinate Y Centroid

-31.4869	-31.4819	-31.4768	-31.4718
-34.9869	-34.9819	-34.9768	-34.9718
-38.4869	-38.4819	-38.4768	-38.4718
-41.9869	-41.9819	-41.9768	-41.9718

Figure 49: Coordinates generated

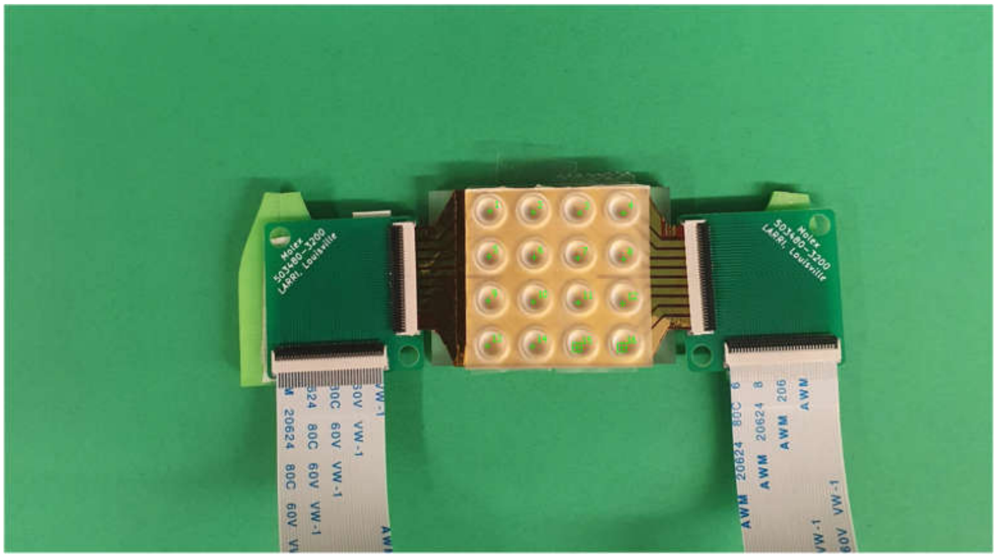


Figure 50: Coordinates generated mapped to the individual sensors

Figure 51 and Figure 52 show the first and second half of the coordinate generation script.

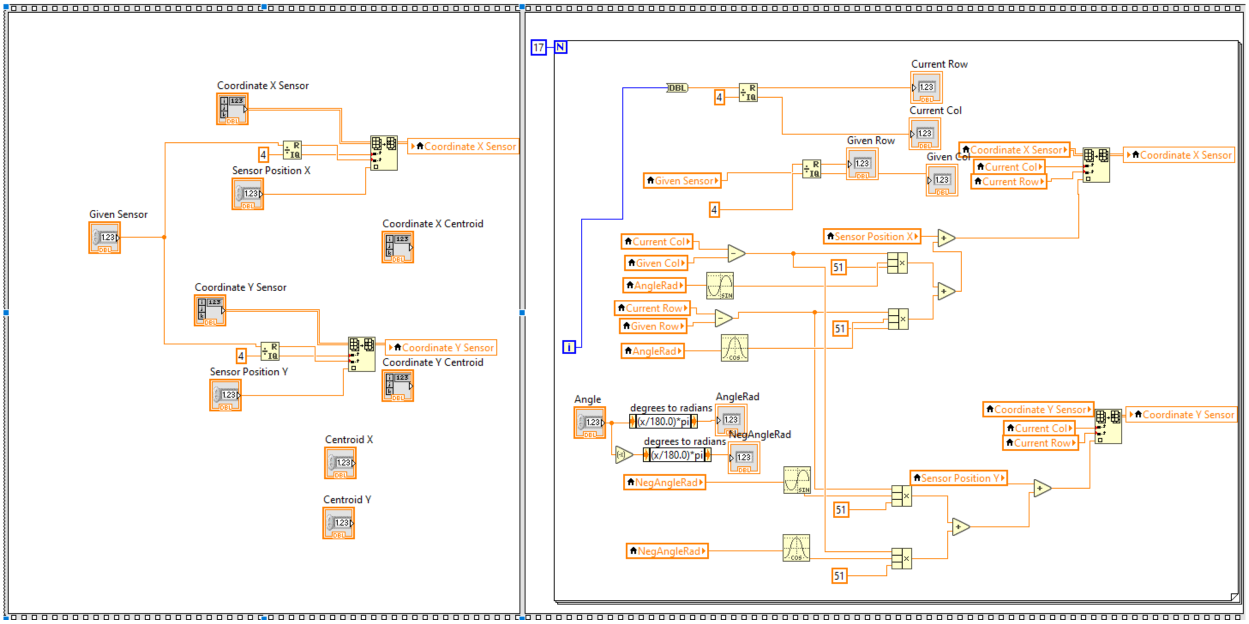


Figure 51: First half of coordinate generation script, for circular edge method

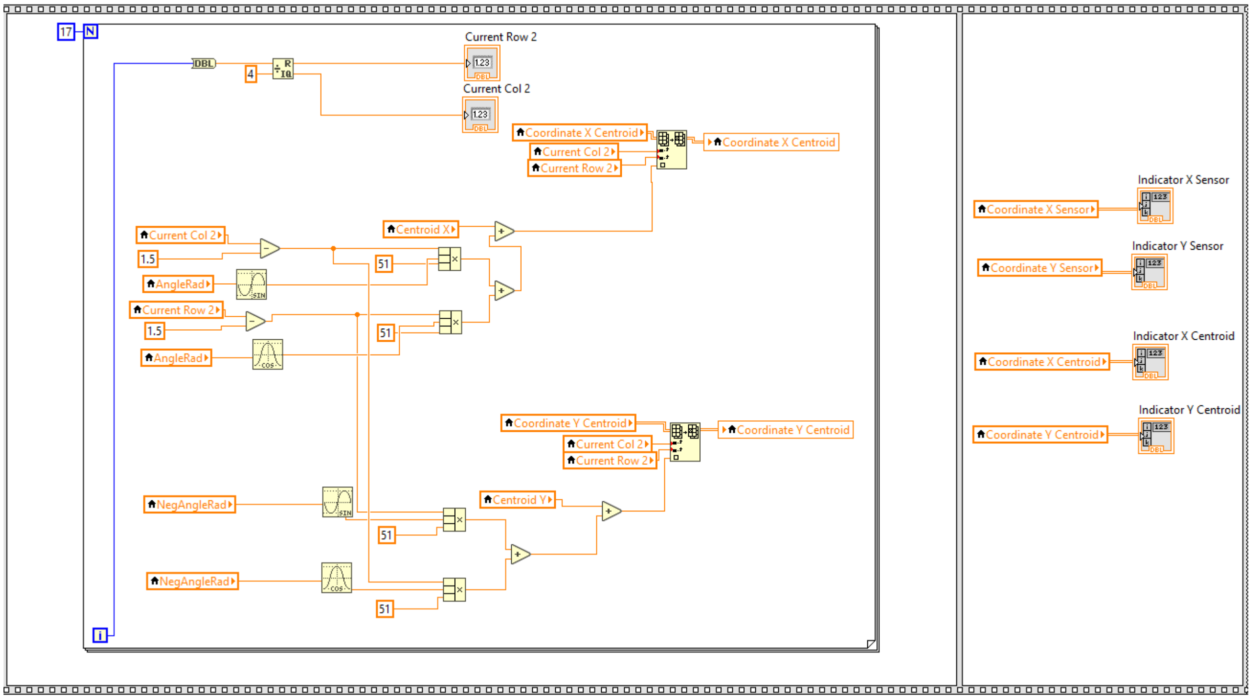


Figure 52: Second half of coordinate generation script, for centroid method

After generating the coordinates, they must be transferred over to the cRIO so that the motor can use them. A global network variable is created on the cRIO that copied over the coordinate locations. Once copied, a Boolean flag is turned on telling the rest of the program that the new coordinates have been found and the test can begin. When a test is completed, this flag can be optionally turned back off. This is so that multiple tests can be run without having to find the same coordinates.

4.3 Accuracy Experimentation

To test the accuracy of the coordinate generation, an experiment was setup. The experiment looks over three distinct images. The first image in Figure 53 is the SkinCell set parallel to the camera with little angle. The second image in Figure 54 has the SkinCell at an exaggerated negative angle to the camera. The third image in Figure 55 has the SkinCell at a exaggerated positive angle to the camera. Manual coordinates were measured by human hand and compared to the computer vision generated coordinates.

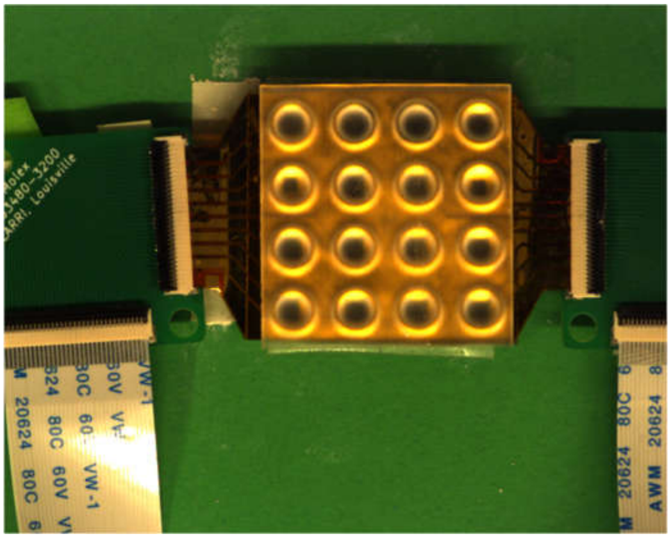


Figure 53: Parallel test image

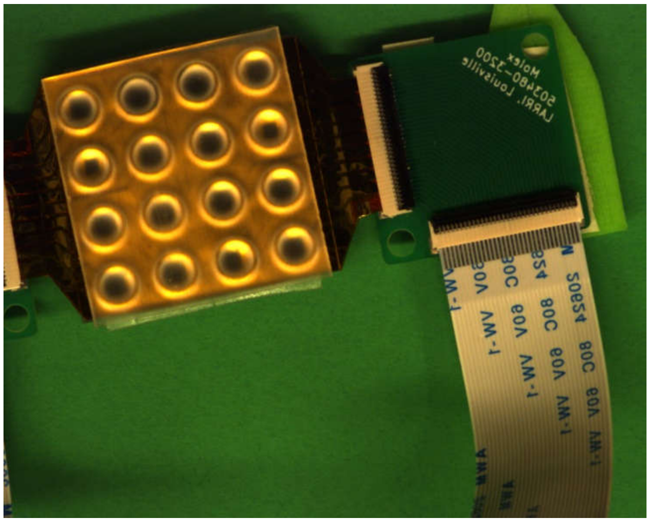


Figure 54: Exaggerated Negative angle test image

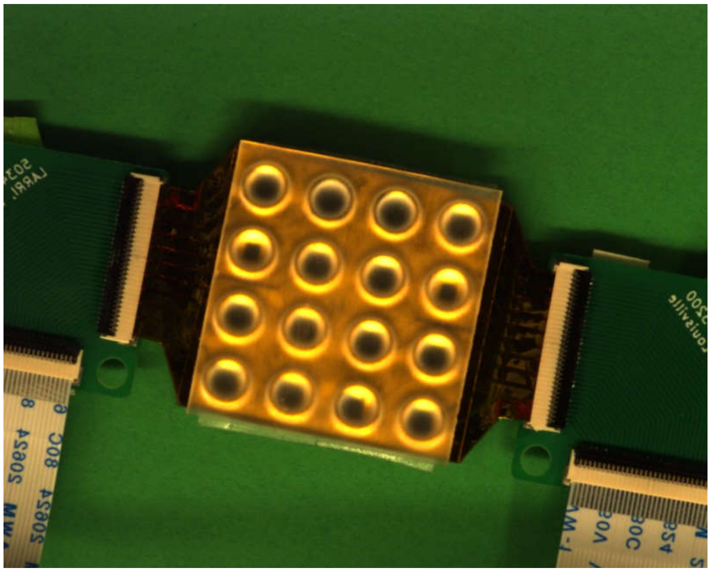


Figure 55: Exaggerated positive angle test image

The full results can be seen in the tables down in Appendix A. The mostly parallel sensor provided the best overall results. The average X coordinate error between the two results is 0.166% with a standard deviation of 0.0138. The Y coordinate error averaged at 0.382% and a standard deviation of 0.0454. The script performed worse when mapping the coordinates of the cells at an angle. In the exaggerated negative angle, the X coordinate has an average error of 1.61% and a standard deviation of 0.3. The Y coordinate average error is at 2.16% and a standard deviation of 0.51, the highest of any test. Finally, the X coordinates for the positive angle had an average percent error of 1.77% with a standard deviation of 0.27. The average Y error was 0.497% with a standard deviation of 0.29. While the angle results performed worse overall, the values of the coordinates still aligned well enough to match with every sensor on the cell to the force indenter. Further adjustments to the script in the future can bring down that percent error even more.

CHAPTER 5

SENSOR TESTING AND ANALYSIS

Electronic SkinCells developed by the NGS group were tested and characterized on the automated testbench for a multitude of publications. All SkinCells were developed with PEDOT:PSS, as a strain gauge sensing element on linear interdigitated electrode(IDE) structures[45]. A micro patterned gold structure within the PEDOT:PSS is used as the strain element. As strain is applied to the gold pattern, resistance in the conductive structure changes[45]. Where the SkinCells differ from one another is how they were fabricated, and the patterned structure in which the PEDOT:PSS is deposited on[45].

The force controller was loaded in to perform a ladder test on each sensor on a cell. Due to different testing philosophies between sensors, the original publication ladder forces do not match the later testing forces. However, a ratio of the output to force is developed across all sensors to see the sensitivity, which allows for fair comparison to occur. The sensor outputs are recorded in data point fed into an ADC. Once post processing on the data is completed, the voltage response of each sensor is aligned to the force applied, and the sensitivity ratio is generated.

SkinCells were placed on top of a bedding made from Polydimethylsiloxane (PDMS), more commonly called silicone[48]. A silicone encapsulation cover is places on top of the of the skin sensor. This cover serves to provide a complete conformance of the applied force for strain transfer[48]. A simulation of the application of the bedding and cover can be seen in Figure 56.

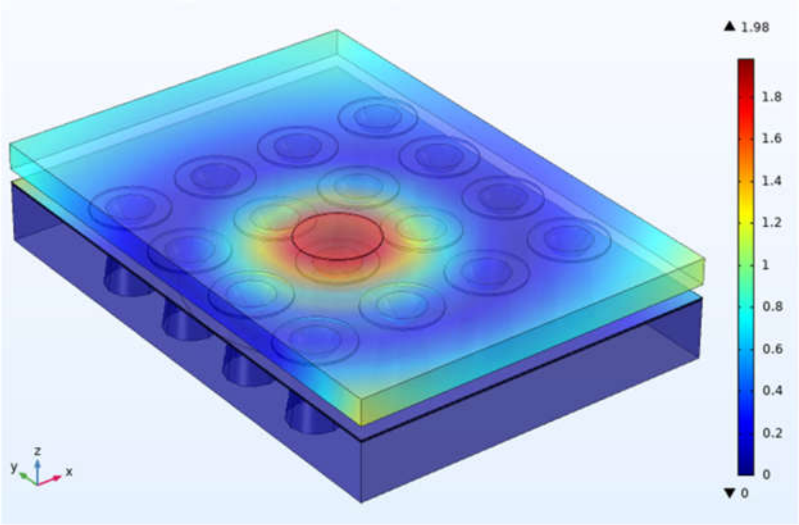


Figure 56: Simulation of skin sensor bedding and cover showing force application[48]

5.1 2021 Circular Tree Geometry

5.1.1 Sensor Description

The skin sensors developed in this batch were used to investigate different geometries[47]. The fabrication was done in a cleanroom environment that uses both gold etchant and wet etching to produce more consistent responses across the skin sensor cells. Sensors in the previous fabrication were designed with a 3.65mm star shape sensing structure[45]. The star shape structure allows for a symmetric omni axial response across the entire sensor[46].

The star shaped structure of the previous sensors was replaced by a circular tree structure. The circular tree structure aims to improve on the star structure of the previous design by improving the spatial resolution of the sensing element. The circular tree offers a denser sensing element then the previous star shaped design, utilizing the space to a more efficient degree. Each circular tree element has a diameter of 2mm, but with a much more compact design to match or surpass the previous star design in sensitivity. Figure 57 and Figure 58 show a comparison of the star shape and circular tree shape elements.

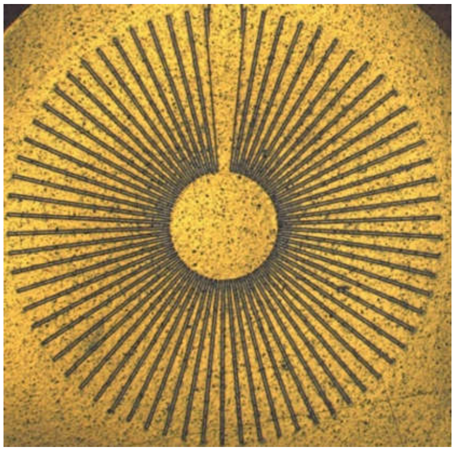


Figure 57: Star Shaped Spoke Structure[46]

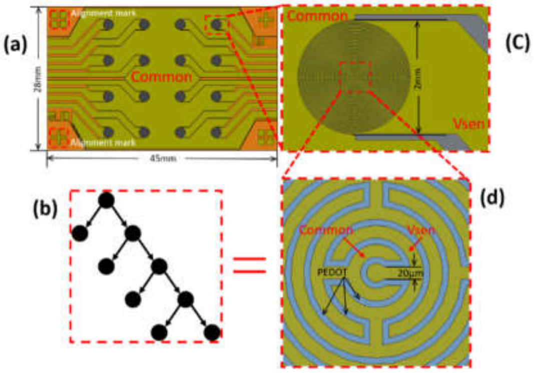


Figure 58: Circular Tree Structure Sensors[47]

5.1.2 Experiment Methodology and Results

A force ladder was applied to each sensor over the center of the circular tree cell. A 3.9 mm diameter indenter was used to apply force. The indenter itself is bigger than the circular tree structure, allowing force to be applied over the sensing region and partially outside the structure as well. The force ladder applied began with 2N, then 5N, then 8N, and finally 10N before stepping back down across the same forces[47]. Figure 59 shows both the force load applied as well as the response from four of the best performing sensors.

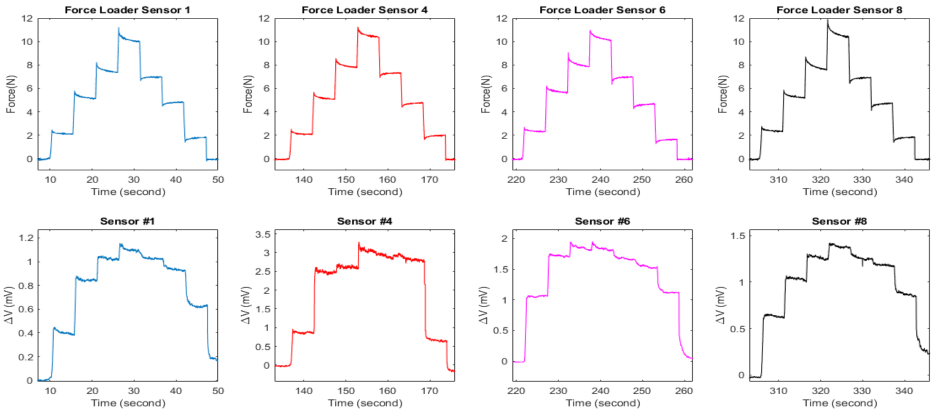


Figure 59: Performance of Skin Sensors 1,4,6, and 8 with variable force[47]

The sensitivity seen in these sensors gives an average response of 0.2mV/N when the applied force ranges between 0-2.7N. Afterwards, a response of 0.07mV/N is seen going from 2.7-10N. When a sensor begins to respond to force, the initial applied force generates the highest increase in voltage across all these sensors, as well as the more average performing ones. Previously, the star shape sensor showed a response of 0.0051mV/N[47], while further testing stated it to be 1125nV/N when applying 0-0.5N, and 412nV/N when applying 0.5-2.3N. The sensitivity readings indicate a gain of 1000 times more when applying force at low values[47]. Overall, this indicates that a switch to the Circular Tree design should be done for future Skin Sensors. Full results for this sensor can be seen in Appendix B1.

5.2 2022 IEEE Sensors Paper

5.2.1 Sensor Description

Skin sensors developed for this trial were developed for the OctoCan project. The OctoCan is a structural electronic device that contains a human robot interaction-based controller that receives human pressure data from mounted SkinCells. The data received from the sensors is used to control a robotic arm positioning. Figure 60 shows a model of the Octocan with the robotic arm.

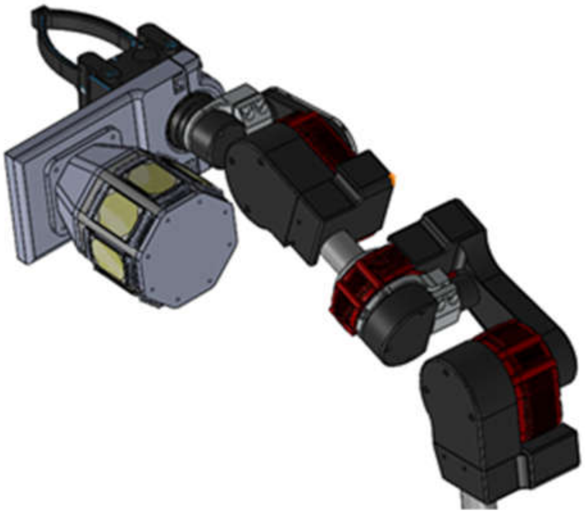


Figure 60: Octocan attached to the robotic arm[48]

The sensors developed use the star shape structure used in previous batches, however the fabrication process is slightly different. The previous batch used a wet lift off etching photolithography process to create the sensors. The new sensors used a gold etchant solution to improve the yields. Gold etchant is able to dissolve tiny particles that can cause a short circuit of the skin sensors after liftoff[48]. This should produce more working sensors on a cell then previously.

5.2.2 Experiment Methodology and Results

The SkinCell is placed between the bedding and cover on top of the X and Y linear stages for a raster scan. A For these sensors, two separate types of tests were run. The first was a series of four static force application tests that applied a force of 0.5N, 1N, 1.5N, and 2N. Static force tests were chosen for this application instead of ladder or dynamic tests to test the sensors after a time so that the strain gauges could reform back to their initial position. The individual trials were layered over top of each other to see performance at each layer. The star shaped sensors produced voltages in the couple hundred microvolt range. The sensitivity was worse than the circular tree structure, which makes sense given that the circular tree was designed to replace the star. The sensors did track the force indenter, raising and lowering voltage change with respect to force. Figure 61 shows the force indenter applied to sensor 4 and the results it generated. Figure 62 shows the calibration results across all sensors, with extra analysis for sensors 4, 6, 9 and 13.

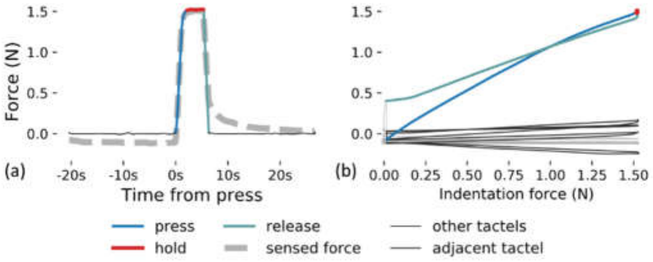


Figure 61: a) Force Indenter measurements from 1.5N test for sensor 4. Dashed line is used for sensor response, solid line is for force feedback. b) Hysteresis plot for sensor 4, grey represents all other sensors on cell[48]

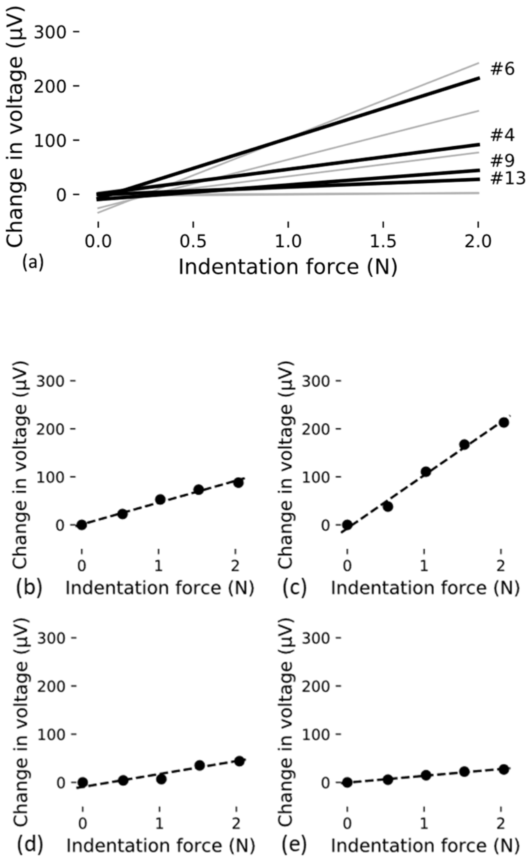
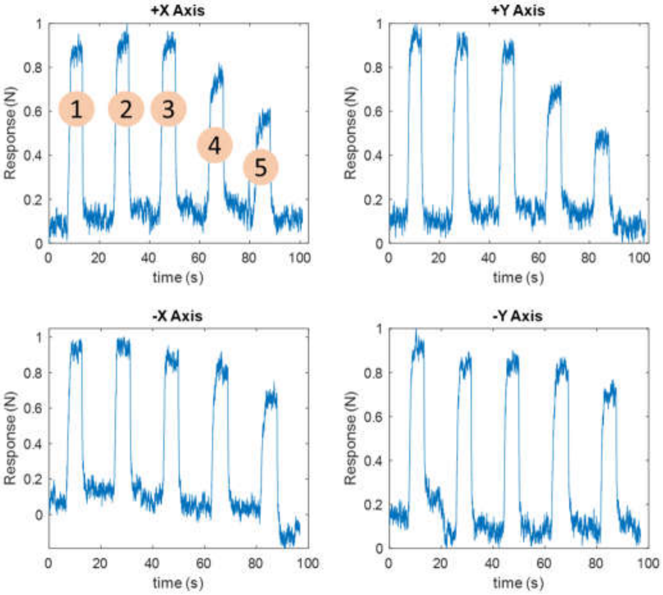


Figure 62: a) SkinCell sensitivity calibration profile for all sensors. b) c) d) and e) represent graphs for individual sensors 4, 6, 9, and 13 respectively[48]

Secondary tests were run on individual sensors to test the response as the point of indention was shifted. The indenter would apply force at the center, before moving 0.35mm away from the sensor. This motion was done 5 times to where the indenter would end up being completely off the sensor. This test was run in each of the four cardinal directions of a sensor to see the degradation of response. Figure 63 shows the experiment results being applied to sensor ?. The first three responses show similar responses in each of the cardinal directions; however, a tapering is seen in the +X and +Y direction responses respectively. At 2.1mm away from the sensor, a drop of roughly 30% is seen on both the X and Y change, and a drop of 50% is seen on peak 5 at 2.8mm away from the center[48]. The negative direction changes saw a drop of 30% maximum. Full results for this experiment set can be seen in Appendix B2.



S

Figure 63: Single SkinCell indention offset experimentation. First peak aligns with the center of sensor before moving off[48]

5.3 Inkjet printing sensors

5.3.1 Sensor Description

The skin sensors developed in this batch were meant to try and improve upon the circular tree structures from the previous year. The sensors were printed using the Nordson Pico Pulse inkjet system, which is a part of the NeXuS system being developed by LARRI. The inkjet system allows for the PEDOT:PSS deposition process to be done outside of the cleanroom. This reduces the costs of manufacturing by removing cleanroom usage and reduces time from a 5-hour deposition down to 5 minutes[49].

A second SkinCell was fabricated using the Pico Pulse but differed by having different geometry across the individual sensors. Seven different geometries were sketched into the cell, each based on either the circular tree shape or the star shape design. Sensor types A and B are based on the star structure, with both having a trace width of 10um and a gap of 20um. The two sensors differ in the full diameter of the sensor, with type A at 3.65mm and type B at 3.04mm. Sensor type A represents the original star shape structure designed and tested previously. Sensor types C, D, and E represent circular tree sensors with 10um traces and a 20um gap, with different diameters. Type C has a 0.46mm diameter and represents the smallest sensor on the cell. Type D is measured at 1.0mm while type E is measured at 1.48mm. Sensor type F has the same topology as sensor types A and B but differs with a much larger spoke length of 0.44mm and a diameter of 2.16mm. Finally, sensor type G is the control group, using the previous circular tree shape at 2.0mm. Figure 64 shows the SkinCell, with part a having the original circular tree, b showing the different designs on the sensor, and c showing a close up of the different geometries[49].

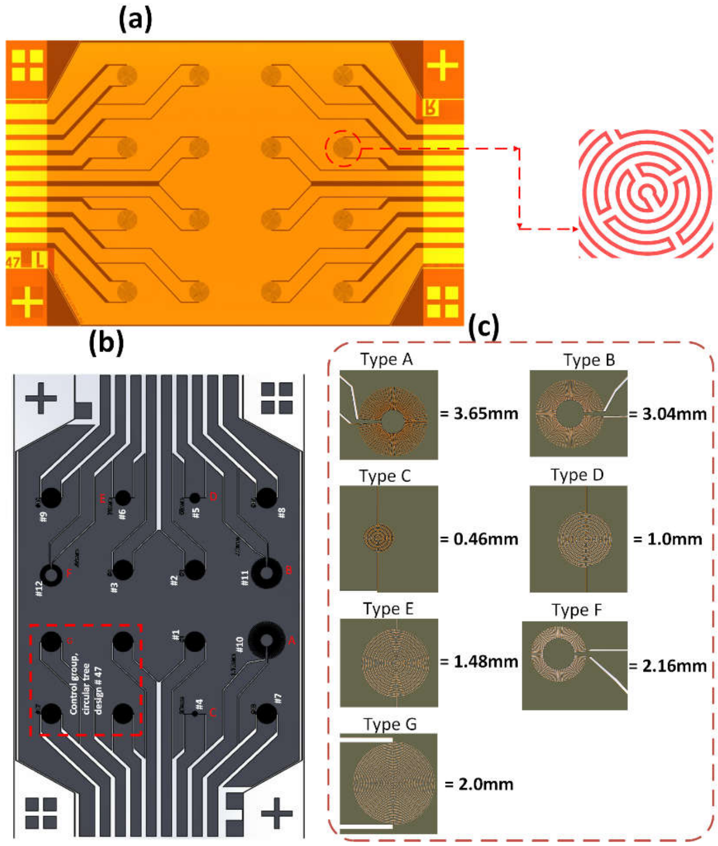


Figure 64: Pico Pulse Sensor structure a)Circular Tree Structure Design B) Cell with different Geometries C) Closeup of different Geometries[49]

5.3.2 Experiment Methodology and Results

A force ladder was applied across each sensor, with force values of 0.5, 2, 4, and 6 newtons. Figure 65 Shows the applied force ladder used for testing the sensors. Afterwards, the sensitivity of each sensor as well as sensor types was recorded. Sensitivity charts were developed comparing the different sensor geometries, and a main sensitivity chart was developed to compare the current sensor to those made in 2020.

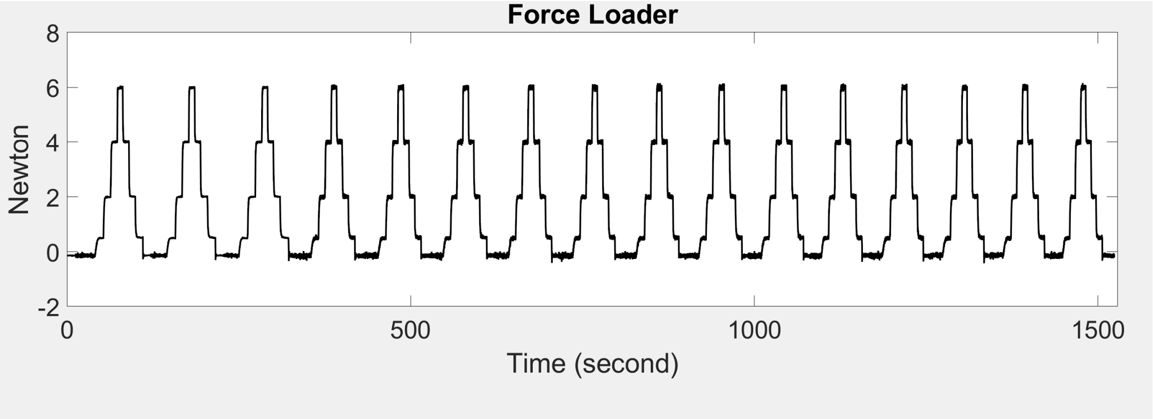


Figure 65: Force Ladder Data Applied across all 16 sensors

On working sensors, an average sensitivity of 11.42mV/N is seen with a 22.84mV response at 2N. This indicates a significant improvement over the previous sensors which showed a sensitivity of 0.2mV/N. Sensitivity increases linearly across the test before beginning to drop off past 6N of force[49]. Figure 66 a) shows a comparison between the previous year's sensor sensitivity and the new sensors. b) shows an applied force ladder to a well performing sensor and the response given. c) shows a comparison of the differing geometries, and all appear to be sensitive based on the use of the inkjet techniques. Sensor type G, the 2mm circular tree structure, does show a far better ability to track the indented force, with more defined steps between each layer[49].

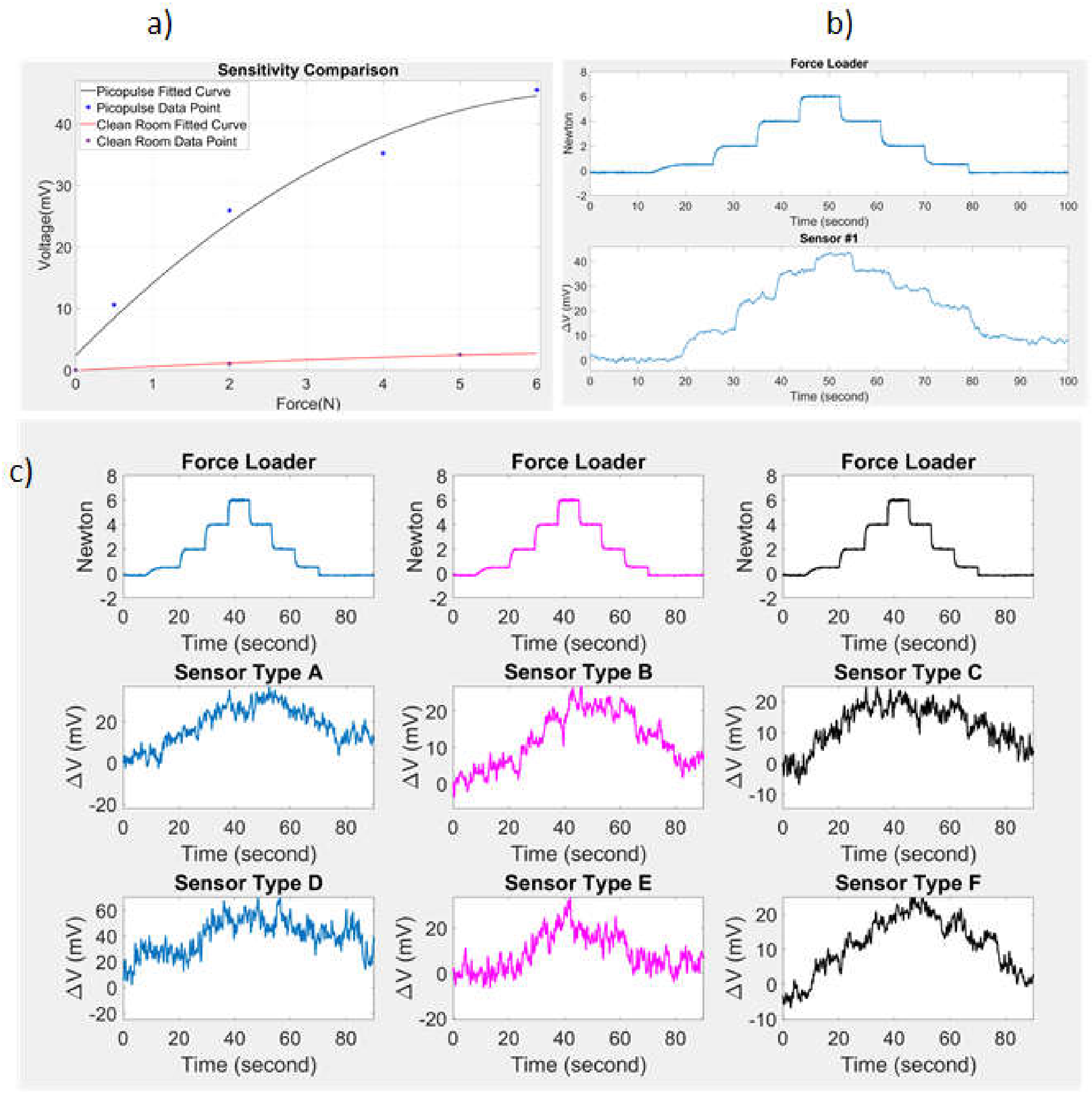


Figure 66: a) Sensitivity Readings between the previous circular tree sensors and the new ones. b) Response graph of the circular tree structure from the similar geometry cell. c) Response graphs of skin sensors with different geometry[49].

The increased sensitivity does show another problem with the inkjet printed SkinCells. The sensors almost all show an incredible amount of noise in their response signals. These disruptions alter a signal tremendously and when the force indenter is stable can see a change in voltage up to multiple millivolts in both the positive and negative directions. Some of the manufactured sensors also produce a peaking effect. This occurs when a force is applied, and the sensor shoots up to an incredibly high value and not returning down to its initial position. This could be seen as a possible error in the data logging software, but after examination and updates, the peaking effect still existed. Figure 67 shows the peaking effect occurring in sensor 16, happening after the 4N is applied. Future manufacturing of this style of sensor will need to adjust the noise and the peaking effect to be used in the future. Full results can be seen in Appendix B3 and B4.

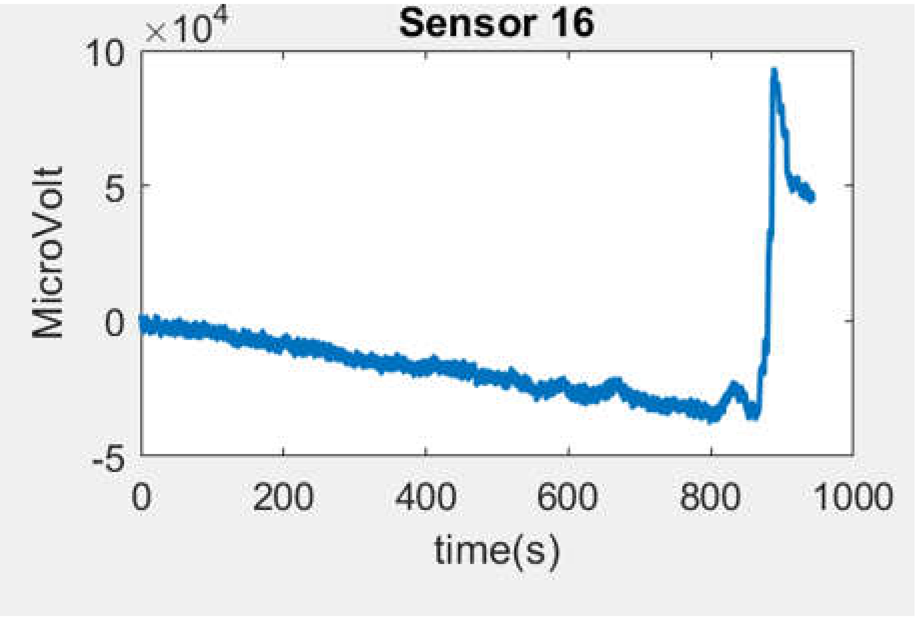


Figure 67: Peaking effect seen in sensor 16.

5.4 Optomec Aerosol Printing Sensors

5.4.1 Sensor Description

Skin sensors manufactured in the experiment were once again done outside the clean room on the NeXuS. Instead of using Pico Pulse for the deposition, the Optomec

Decathlon Aerosol inkjet was used. Like the Pico Pulse, the aerosol printing system is used to shave off manufacturing time of cleanroom deposition and to remove the human error associated[50]. The aerosol printed sensors come with a PEDOT:PSS thickness of 100nm, which is smaller than previously deposited inks. The sensor was created using the same different geometries as the last one, so that the different printing devices could be compared to one another.

5.4.2 Experiment Methodology and Results

The previous force ladder test done for the PicoPulse sensors is once again done here, as it is the standardized test run on sensors from this point forward. Sensitivity levels of were considerably lower than both the PicoPulse sensor and also the previous cleanroom manufactured sensors. However, the sensors showed a far greater ability to track force with much less noise. Working sensors saw an almost 1 to 1 track between the force and the response generated, even when further testing was done with dynamic waveforms. Full results can be seen in Appendix B5 and B6.

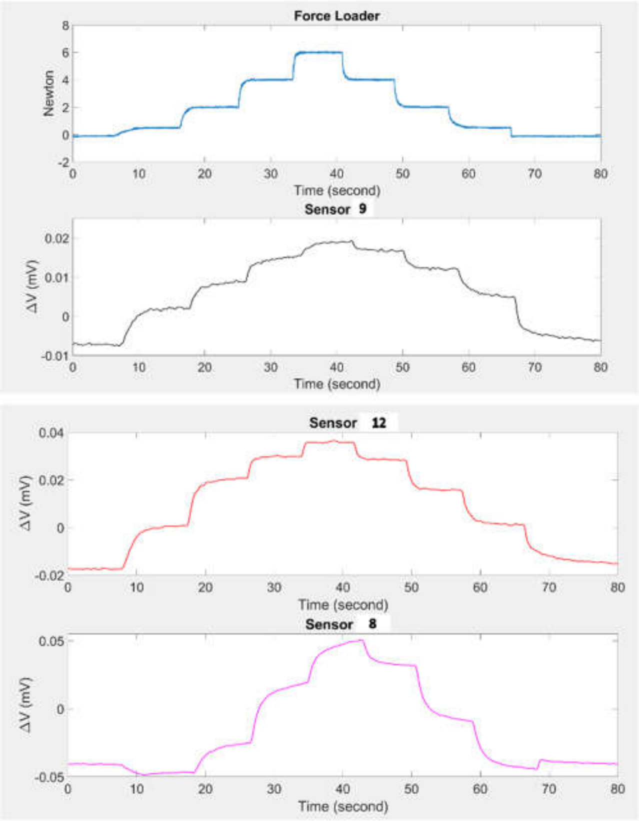


Figure 68: Force Response of Aerosol sensors[50]

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we focused on the development of the automated test bench for use of testing sensors made by the NGS group, and the results of applying the testing bench to different skin sensors. We explored the use of an automation profile to complete a raster scan of a skin sensor in real time, and the use of a camera and computer vision scripts in order to automate the sensor alignment process. Also, the development of force control was done to allow repeatable testing and the application of dynamic waveforms. Finally, the bench was applied to a variety of skin sensors manufactured by the team, and data analysis was done to determine the effectiveness of the sensors.

6.1 Conclusion

Additional software suites were designed in order to accurately test and characterize the data response of sensors. Using LabVIEW as a development software, we were able to control an indenter based on the amount of force it was applying. Using this, we could guide it along a predetermined path to apply desired forces at a quick rate. Additionally, the tracking was proven to be well enough to follow a time varying waveform and apply force mimicking that waveform to a substrate.

Using the force controller, we are able to create repeatable force tests that allows us to fairly examine the sensors in a cell. The knowledge provided by this allows us to test and characterize our sensors to one another to see compare the performance between the individual sensors, as well as sensors made using different manufacturing techniques or batches. This information allows us to determine which deposition techniques and geometry design to pursue for future testing and development.

The automation profile implemented allows for tests to be done with minimal human labor necessary. By having the system move between each sensor on a cell and applying roughly the exact same force to each location allows the engineer to focus on the data response of the of the SkinCell. The implementation of the camera and computer vision scripts allows the testbench to automatically generate its own raster scan coordinates. This skips over the manual alignment step previously required and takes out the need for a parallel angle to be used so that each sensor is hit. This removes the arduous process entirely and allows quicker access to the testing setup, which allows the testing engineer to run more tests on the same and different sensors in the same amount of time.

Finally, applying the automated test bench to sensors developed by the NGS team allows us to accurately compare both the sensors and the cells to one another. Testing sensors with the circular tree structure showed an improvement over the initial star structure of roughly 1000 times greater in terms of sensitivity. The updated printing processes, using both the Opotomec printing system and the PicoPulse printing system, show benefits over the previous sensors in different aspects. The PicoPulse manufactured sensors gave off a much higher sensitivity then the clean room sensors, but the high sensitivity causes incredible variance in readings, and trouble with tracking. The sensors generated using the Optomec printing system offered slightly less sensitivity then those in a full clean room process. However, the sensors offered the ability to create responses mirroring the waveform applied. This increase in tracking, and faster time over their clean room counterparts, allows the sensors to be easily made and used in applications that require measuring distinct changes in sensitivity.

6.2 Future Work

In the future, the automated test bench can be used to quickly test and characterize new skin sensors, as well as any type of strain-based sensor developed by the NGS teams. Developing extensions to the computer vision scripts will allow the system to auto align itself to other sensors made as well. Additionally, a manual selection mode can be added, so that the user can select where they would like the motor to go incase of a fault in the script.

The motor control solution currently can apply time varying waveforms and steady state correction based on the current force. In the future, this can be updated further to track custom time varying waves that the user wants. Doing this allows us to follow specific waveforms that might better simulate a response given under a certain application. In addition, the motor can be updated to run oscillating waveforms at a faster rate then it is currently possible. Impedance control methods can also be developed, so that both the force and control can be more fine tuned in the real time environment.

REFERENCES

[1] T. Someya and T. Sekitani, “Printed skin-like large-area flexible sensors and actuators,” *Procedia Chem.*, vol. 1, no. 1, pp. 9–12, 2009.

[2] C. Bartolozzi, L. Natale, F. Nori, and G. Metta, “Robots with a sense of touch,” *Nat. Mater.*, vol. 15, no. 9, pp. 921–925, 2016.

[3] S. Stassi, V. Cauda, G. Canavese, and C. F. Pirri, “Flexible tactile sensing based on piezoresistive composites: a review,” *Sensors (Basel)*, vol. 14, no. 3, pp. 5296–5332, 2014.

[4] A. Chortos, J. Liu, and Z. Bao, “Pursuing prosthetic electronic skin,” *Nat. Mater.*, vol. 15, no. 9, pp. 937–950, 2016.

[5] V. E. Abraira and D. D. Ginty, “The sensory neurons of touch,” *Neuron*, vol. 79, no. 4, pp. 618–639, 2013.

[6] X. Wang, L. Dong, H. Zhang, R. Yu, C. Pan, and Z. L. Wang, “Recent progress in electronic skin,” *Adv. Sci. (Weinh.)*, vol. 2, no. 10, p. 1500169, 2015.

[7] M. L. Hammock, A. Chortos, B. C.-K. Tee, J. B.-H. Tok, and Z. Bao, “25th anniversary article: The evolution of electronic skin (e-skin): a brief history, design considerations, and recent progress,” *Adv. Mater.*, vol. 25, no. 42, pp. 5997–6038, 2013.

[8] F. Jiang, Y.-C. Tai, K. Walsh, T. Tsao, G.-B. Lee, and C.-M. Ho, “A flexible MEMS technology and its first application to shear stress sensor skin,” in *Proceedings IEEE The Tenth Annual International Workshop on Micro Electro Mechanical Systems. An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots*, 2002.

[9] J. C. Yang, J. Mun, S. Y. Kwon, S. Park, Z. Bao, and S. Park, "Electronic skin: Recent progress and future prospects for skin-attachable devices for health monitoring, robotics, and prosthetics," *Adv. Mater.*, vol. 31, no. 48, p. e1904765, 2019.

[10] Z. Li, M. Zhu, J. Shen, Q. Qiu, J. Yu, and B. Ding, "All-fiber structured electronic skin with high elasticity and breathability," *Adv. Funct. Mater.*, vol. 30, no. 6, p. 1908411, 2020.

[11] X. Chen *et al.*, "Flexible fiber-based hybrid nanogenerator for biomechanical energy harvesting and physiological monitoring," *Nano Energy*, vol. 38, pp. 43–50, 2017.

[12] T. B. Sheridan, "Human-robot interaction: Status and challenges: Status and challenges," *Hum. Factors*, vol. 58, no. 4, pp. 525–532, 2016.

[13] J. Rajruangrabin and D. O. Popa, "Enhancement of manipulator interactivity through compliant skin and extended Kalman filtering," in *2007 IEEE International Conference on Automation Science and Engineering*, 2007

[14] S. Cremer, S. K. Das, I. B. Wijayasinghe, D. O. Popa, and F. L. Lewis, "Model-free online neuroadaptive controller with intent estimation for physical human–robot interaction," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 240–253, 2020.

[15] S. Cremer, I. Ranatunga, S. K. Das, I. B. Wijayasinghe, and D. O. Popa, "Neuroadaptive calibration of tactile sensors for robot skin," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016.

[16] A. Zacharaki, I. Kostavelis, A. Gasteratos, and I. Dokas, "Safety bounds in human robot interaction: A survey," *Saf. Sci.*, vol. 127, no. 104667, p. 104667, 2020.

[17] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Found. Trends® Hum.–Comput. Interact.*, vol. 1, no. 3, pp. 203–275, 2007.

[18] V. Duchaine, N. Lauzier, M. Baril, M.-A. Lacasse, and C. Gosselin, "A flexible robot skin for safe physical human robot interaction," in *2009 IEEE International Conference on Robotics and Automation*, 2009.

[19] D. Kulić and E. A. Croft, "Real-time safety for human–robot interaction," *Rob. Auton. Syst.*, vol. 54, no. 1, pp. 1–12, 2006.

[20] S. Bennett, "A brief history of automatic control," *IEEE control syst.*, vol. 16, no. 3, pp. 17–25, 1996.

[21] H. Schwilden, H. Stoeckel, and J. Schüttler, "Closed-loop feedback control of propofol anaesthesia by quantitative EEG analysis in humans," *Br. J.*

Anaesth., vol. 62, no. 3, pp. 290–296, 1989.

[22] S. Chiaverini and L. Sciavicco, “The parallel approach to force/position control of robotic manipulators,” *IEEE Trans. Rob. Autom.*, vol. 9, no. 4, pp. 361–373, 1993.

[23] J. Maples and J. Becker, “Experiments in force control of robotic manipulators,” in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, 2005.

[24] M. T. Mason, “Compliance and force control for computer controlled manipulators,” *IEEE Trans. Syst. Man Cybern.*, vol. 11, no. 6, pp. 418–432, 1981.

[25] A. Alcocer, A. Robertsson, A. Valera, and R. Johansson, “Force estimation and control in robot manipulators,” *IFAC proc. vol.*, vol. 36, no. 17, pp. 55–60, 2003.

[26] S. Jung, T. C. Hsia, and R. G. Bonitz, “Force tracking impedance control of robot manipulators under unknown environment,” *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 3, pp. 474–483, 2004.

[27] D. C. Slaughter, D. K. Giles, and D. Downey, “Autonomous robotic weed control systems: A review,” *Comput. Electron. Agric.*, vol. 61, no. 1, pp. 63–78, 2008.

[28] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *arXiv [cs.CV]*, 2019.

[29] H. Moon, R. Chellappa, and A. Rosenfeld, “Optimal edge-based shape detection,” *IEEE Trans. Image Process.*, vol. 11, no. 11, pp. 1209–1226, 2002.

[30] T. Mainzer, "Genetic algorithm for shape detection." *Technical Report no. DCSE/TR-2002–06* (2002).

[31] K. Thenmozhi and U. S. Reddy, “Image processing techniques for insect shape detection in field crops,” in *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 2017.

[32] I. Kumar, J. Rawat, and H. S. Bhadauria, "A conventional study of edge detection technique in digital image processing." *International Journal of Computer Science and Mobile Computing* 3.4 (2014): 328-334.

[33] G. Mane, K. Panchal, and S. Sable. " Analysis Of Various Techniques Used For Implementation Of Video Surveillance System." *International Journal of Engineering Research and Applications* 3.3 (2013): 292-297.

[34] P. Suetens, P. Fua, and A. J. Hanson. "Computational strategies for object recognition." *ACM Computing Surveys (CSUR)* 24.1 (1992): 5-62.

[35] A. Rosenfeld, “Image pattern recognition,” *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 69, no. 5, pp. 596–605, 1981.

[36] “LSC 30/2, 4-Q-DC servoamplifier in module housing,” *Online shop for high precise drive systems by maxon | maxon group*. [Online]. Available: https://www.maxongroup.com/maxon/view/product/250521. [Accessed: 24-Mar-2022].

[37] “PID function block,” *PID Function Block - LabVIEW 2018 Real-Time Module Help - National Instruments*. [Online]. Available: https://zone.ni.com/reference/en-XX/help/370715P-01/lvfbhelp/pid_fb/. [Accessed: 11-Feb-2022].

[38] “NI 9516 (FPGA interface),” *NI 9516 (FPGA Interface) - CompactRIO Reference and Procedures (FPGA Interface) - National Instruments*. [Online]. Available: https://zone.ni.com/reference/en-XX/help/370984T-01/target4devicehelp/9516_io_reference/. [Accessed: 02-Mar-2022].

[39] “Write,” *Write - LabVIEW 2018 SoftMotion Module - National Instruments*. [Online]. Available: https://zone.ni.com/reference/en-XX/help/371093W-01/nimclvfb/dt/eng/generated/write_00b10083/. [Accessed: 15-Mar-2022].

[40] “Straight-line move mode,” *Straight-Line Move Mode - LabVIEW 2018 SoftMotion Module - National Instruments*. [Online]. Available: https://zone.ni.com/reference/en-XX/help/371093W-01/nimclvfb/dt/eng/generated/straightlinemove.mode_00a10014/. [Accessed: 18-Mar-2022].

[41] “IDS UI-1240SE-C-HQ EO-1312C camera: Edmund Optics,” *Edmund Optics Worldwide*. [Online]. Available: https://www.edmundoptics.com/p/eo-1312c-118-cmos-color-usb-camera/26850/. [Accessed: 26-Jan-2022].

[42] NI. (n.d.). *Ni vision assistant Tutorial - National Instruments*. National Instruments. Retrieved November 10, 2021, from https://www.ni.com/pdf/manuals/372228m.pdf

[43] “IMAQ find Circular Edge 2 vi,” *IMAQ Find Circular Edge 2 VI - NI Vision 2019 for LabVIEW Help - National Instruments*. [Online]. Available: https://zone.ni.com/reference/en-XX/help/370281AG-01/imaqvision/imaq_find_circular_edge_2/. [Accessed: 08-Feb-2022].

[44] "Using the labview shared variable," *NI*. [Online]. Available: https://www.ni.com/en-us/support/documentation/supplemental/06/using-the-labview-shared-variable.html. [Accessed: 10-Feb-2022].

[45] D. Wei, R. Zhang, M. N. Saadatzi, O. O. Olowo, and D. O. Popa, "Organic piezoresistive pressure sensitive robotic skin for physical human-robot interaction," in *Volume 1: 14th International Conference on Micro- and Nanosystems (MNS)*, 2020.

[46] J. R. Baptist, R. Zhang, D. Wei, M. N. Saadatzi, and D. O. Popa, "Fabrication of strain gauge based sensors for tactile skins," in *Smart Biomedical and Physiological Sensor Technology XIV*, 2017.

[47] O. O. Olowo, R. Zhang, Z. Yang, B. Goulet, and D. O. Popa, "Organic piezoresistive robotic skin sensor fabrication, integration and characterization," in *Volume 2: Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability*, 2021.

[48] R. Zhang, J.T. Lin, O. O. Olowo, B. Goulet, B. Harris, and D. O. Popa, "SkinCell: A Modular Tactile Sensor Patch for Physical Human-Robot Interaction" *IEEE Sensors Journal,* Under Review, 2022.

[49] O. O. Olowo, R. Zhang, A. Sherehiy, B. Goulet, A. Curry, D. Wei, Z. Yang, M. Alqatamin, and D. O. Popa. "Inkjet Printing of PEDOT: PSS Inks for Robotic Skin Sensors" *International Manufacturing Science and Engineering Conference*. American Society of Mechanical Engineers, Accepted, 2022.

[50] O. O. Olowo, D. Wei, D. Ratnayake, B. Goulet, A. Curry, A. Sherehiy, R. Zhang, and D. O. Popa. "PEDOT: PSS polymer Aerosol Jet-printing for Robotic Skin Sensors." *IEEE International Conference on Flexible, Printable Sensors and Systems,* Accepted, 2022.

[51] X. Sun *et al.*, "Recent applications of different microstructure designs in high performance tactile sensors: A review," *IEEE Sens. J.*, vol. 21, no. 9, pp. 10291–10303, 2021.

[52] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "GelSlim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," *arXiv [cs.RO]*, 2018.

[53] Y. Tenzer, L. P. Jentoft, and R. D. Howe, "The feel of MEMS barometers: Inexpensive and easily customized tactile array sensors," *IEEE Robot. Autom. Mag.*, vol. 21, no. 3, pp. 89–95, 2014.

[54] NSF Award Search: Award #1849213 - RII Track-1: Kentucky Advanced Manufacturing Partnership for Enhanced Robotics and Structures. [Online]. Available: https://www.nsf.gov/awardsearch/showAward?AWD_ID=1849213

[55] K. R. Shook, A. Habib, W. H. Lee, and D. O. Popa. "Experimental testbed for robot skin characterization and interaction control." *Next-Generation Robots and Systems*. Vol. 9116. SPIE, 2014.

[56] Shook, R. Kyle. "Experimental testbed for robot skin characterization and interaction control." M.S. Thesis, Department of ME, University of Texas at Arlington, 2014.

[57] Sahasrabuddhe, Ritvij. "Development, Testing and Characterization of Electronic Skins for Robots.", M.S. Thesis, Department of EE, University of Texas at Arlington, 2015.

[58] B. Komati, M. R. Pac, I. Ranatunga, C. Clévy, D. O. Popa, and P. Lutz, "Explicit force control vs impedance control for micromanipulation," in *Volume 1: 15th International Conference on Advanced Vehicle Technologies; 10th International Conference on Design Education; 7th International Conference on Micro- and Nanosystems*, 2013.

[59] B. Alqaudi, H. Modares, I. Ranatunga, S. M. Tousif, F. L. Lewis, and D. O. Popa, "Model reference adaptive impedance control for physical human-robot interaction," *Contr. Theory Technol.*, vol. 14, no. 1, pp. 68–82, 2016.

[60] S. K. Singh and D. O. Popa, "An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments," *IEEE Trans. Rob. Autom.*, vol. 11, no. 6, pp. 912–921, 1995.

[61] F. Mirza, R. R. Sahasrabuddhe, J. R. Baptist, M. B. J. Wijesundara, W. H. Lee, and D. O. Popa, "Piezoresistive pressure sensor array for robotic skin," in *Sensors for Next-Generation Robotics III*, 2016.

[62] M. N. Saadatzi, J. R. Baptist, Z. Yang, and D. O. Popa, "Modeling and fabrication of scalable tactile sensor arrays for flexible robot skins," *IEEE Sens. J.*, vol. 19, no. 17, pp. 7632–7643, 2019.

[63] M. N. Saadatzi, J. R. Baptist, I. B. Wijayasinghe, and D. O. Popa, "Characterization of large-area pressure sensitive robot skin," in *Smart Biomedical and Physiological Sensor Technology XIV*, 2017.

[64] S. Cremer, M. N. Saadatzi, I. B. Wijayasinghe, S. K. Das, M. H. Saadatzi, and D. O. Popa, "SkinSim: A design and simulation tool for robot skin with closed-loop pHRI controllers," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1302–1314, 2

APPENDIX A: ACCURACY EXPERIMENTATION RESULTS

Sensor Number	X Coordinate(Hand Measured)	X Coordinate(Computer Vision)	Percent error
1	59.5	59.3854	0.192605042
2	63	62.8854	0.181904762
3	66.5	66.3854	0.172330827
4	70	69.8854	0.163714286
5	59.5	59.3904	0.184201681
6	63	62.8904	0.173968254
7	66.5	66.3904	0.16481203
8	70	69.8904	0.156571429
9	59.5	59.3955	0.175630252
10	63	62.8955	0.165873016
11	66.5	66.3955	0.157142857
12	70	69.8954	0.149428571
13	59.5	59.4005	0.167226891
14	63	62.9005	0.157936508
15	66.5	66.4005	0.14962406
16	70	69.9005	0.142142857

Table 2: X coordinate accuracy testing results for parallel sensor

Sensor Number	Y Coordinate(Hand Measured)	Y Coordinate(Computer Vision)	Percent error
1	-31.5	-31.6463	0.464444444
2	-31.5	-31.6413	0.448571429
3	-31.5	-31.6362	0.432380952
4	-31.5	-31.6312	0.416507937
5	-35	-35.1463	0.418
6	-35	-35.1413	0.403714286
7	-35	-35.1362	0.389142857
8	-35	-35.1312	0.374857143
9	-38.5	-38.6463	0.38
10	-38.5	-38.6413	0.367012987
11	-38.5	-38.6362	0.353766234
12	-38.5	-38.6312	0.340779221
13	-42	-42.1463	0.348333333
14	-42	-42.1413	0.336428571
15	-42	-42.1362	0.324285714
16	-42	-42.1312	0.312380952

Table 3: Y coordinate accuracy testing results for parallel sensor

Sensor Number	X Coordinate(Hand Measured)	X Coordinate(Computer Vision)	Percent error
1	57.5	56.2942	2.09704348
2	60.5	59.7305	1.27190083
3	64	63.1669	1.30171875
4	68	66.6033	2.05397059
5	58	56.9585	1.79568966
6	61.5	60.3949	1.79691057
7	65	63.8312	1.79815385
8	68.5	67.2676	1.79912409
9	58.5	57.6229	1.49931624
10	62	61.0592	1.51741935
11	65.5	64.4956	1.53343511
12	69	67.932	1.54782609
13	59	58.2872	1.20813559
14	63	61.7236	2.02603175
15	66	65.16	1.27272727
16	69.5	68.5963	1.30028777

Table 4: X coordinate accuracy testing results for negative angle sensor

Sensor Number	Y Coordinate(Hand Measured)	Y Coordinate(Computer Vision)	Percent error
1	-35.5	-36.375	2.46478873
2	-35	-35.7106	2.03028571
3	-34	-35.0462	3.07705882
4	-33.5	-34.3819	2.63253731
5	-39	-39.8113	2.08025641
6	-38	-39.147	3.01842105
7	-37.5	-38.4826	2.62026667
8	-37	-37.8182	2.21135135
9	-42.5	-43.2477	1.75929412
10	-42	-42.5833	1.38880952
11	-41	-41.919	2.24146341
12	-40.5	-41.2546	1.86320988
13	-46	-46.6841	1.48717391
14	-45	-46.0197	2.266
15	-44.5	-45.3553	1.92202247
16	-44	-44.691	1.57045455

Table 5: Y coordinate accuracy testing results for negative angle sensor

Sensor Number	X Coordinate(Hand Measured)	X Coordinate(Computer Vision)	Percent error
1	64.5	65.7231	1.89627907
2	68	69.136	1.670588235
3	71	72.549	2.181690141
4	75	75.9619	1.282533333
5	64	64.9473	1.48015625
6	67	68.3602	2.030149254
7	70.5	71.7732	1.805957447
8	74	75.1861	1.602837838
9	63	64.1715	1.85952381
10	66.5	67.5844	1.630676692
11	70	70.9974	1.424857143
12	73	74.4103	1.931917808
13	62	63.3957	2.251129032
14	65.5	66.8086	1.997862595
15	69	70.2215	1.770289855
16	72.5	73.6345	1.564827586

Table 6: X coordinate accuracy testing results for positive angle sensor

Sensor Number	Y Coordinate(Hand Measured)	Y Coordinate(Computer Vision)	Percent error
1	-28.7	-28.5305	0.590592334
2	-29.5	-29.3063	0.656610169
3	-30	-30.0821	0.273666667
4	-31	-30.8579	0.458387097
5	-32	-31.9434	0.176875
6	-33	-32.7192	0.850909091
7	-33.5	-33.495	0.014925373
8	-34.5	-34.2709	0.664057971
9	-35.5	-35.3564	0.404507042
10	-36	-36.1322	0.367222222
11	-37	-36.908	0.248648649
12	-38	-37.6838	0.832105263
13	-39	-38.7693	0.591538462
14	-40	-39.5451	1.13725
15	-40.5	-40.3209	0.442222222
16	-41	-41.0967	0.235853659

Table 7: Y coordinate accuracy testing results for positive angle sensor

APPENDIX B: COMPLETE TESTING RESULTS FOR SKIN SENSOR

B1: 2020 Cleanroom Circular Tree Geometry Design

Table 8: Average Response Value of sensors: 0 indicates nonfunctioning sensors

Sensor#	2N response(mV)	6N response(mV)	8N response(mV)	10N response(mV)	Sensitivity average(mv/N)
1	0.375	0.835	1.021	1.141	0.1141
2	0	0	0	0	0
3	0	0	0	0	0
4	0.647	1.21	1.435	1.81	0.181
5	0	0	0	0	0
6	1.052	1.71	1.81	1.93	0.193
7	0	0	0	0	0
8	0.619	1.03	1.25	1.43	0.143
9	0	0.52	0.81	0.93	0.093
10	0	0.558	0.861	1	0.1
11	0.629	0.953	1	1.07	0.107
12	0	0.67	0.985	1.09	0.109
13	0	0.438	0.589	0.746	0.0746
14	0	0.727	0.975	1.15	0.115
15	0	0	0	0	0
16	0.7477	0.947	1.03	1.136	0.1136

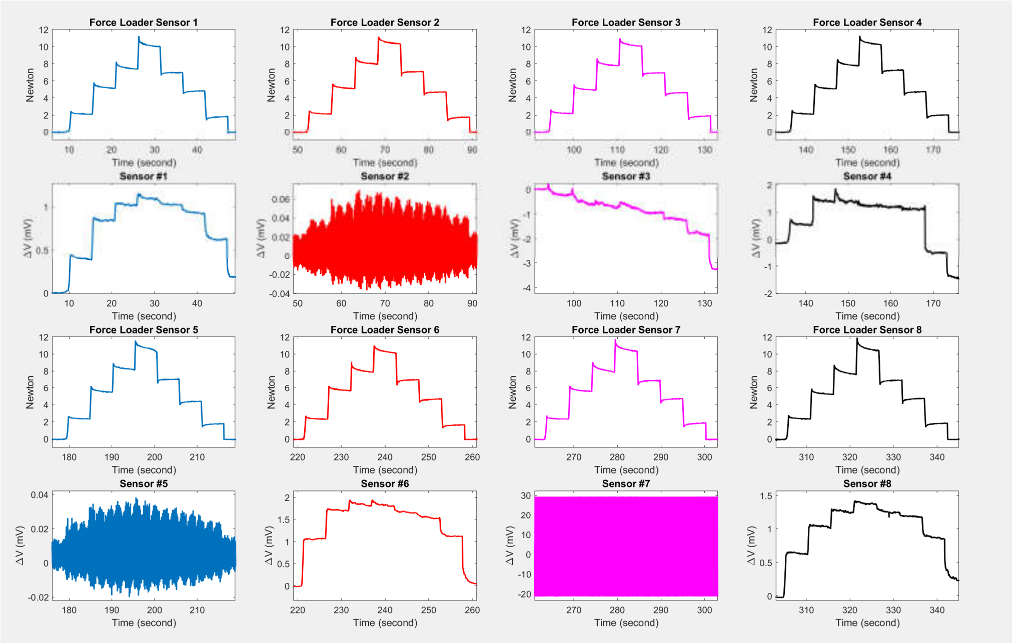


Figure 69: Visualization of Force load and sensor response for sensors 1-8

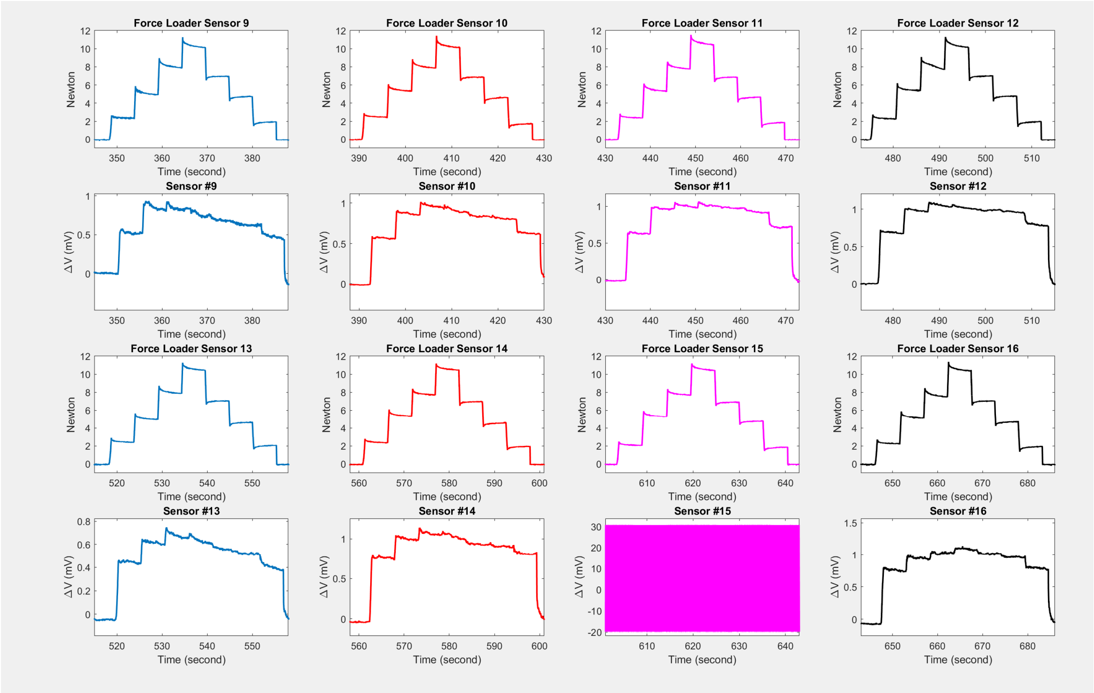


Figure 70: Visualization of Force load and sensor response for sensors 9-16

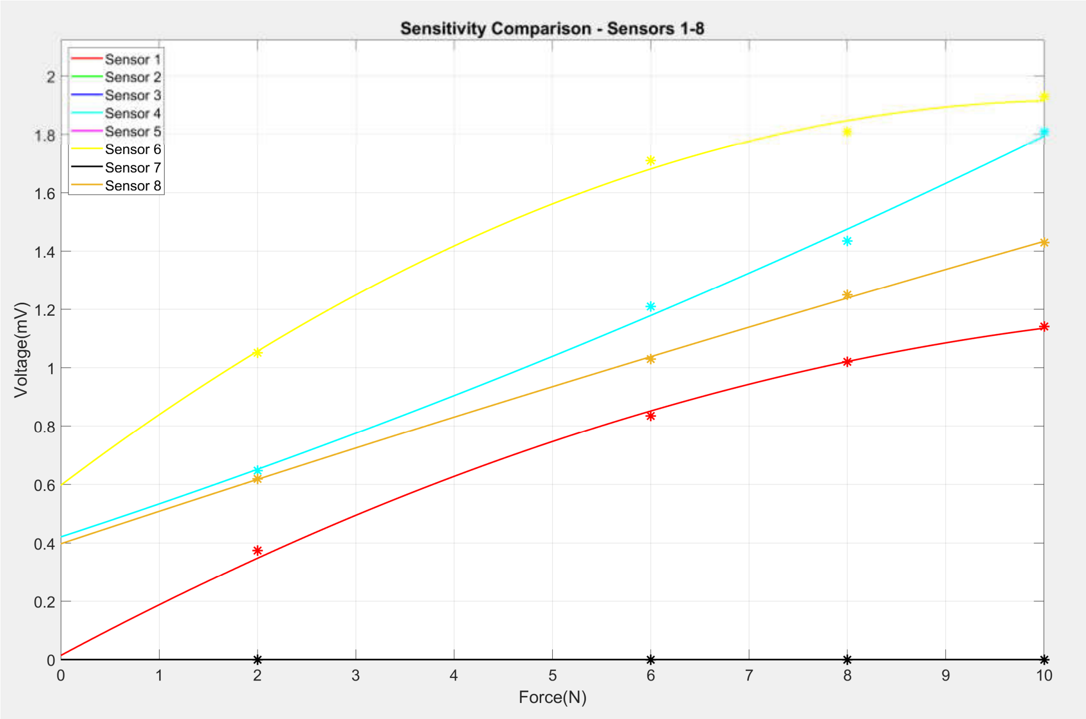


Figure 71: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit.

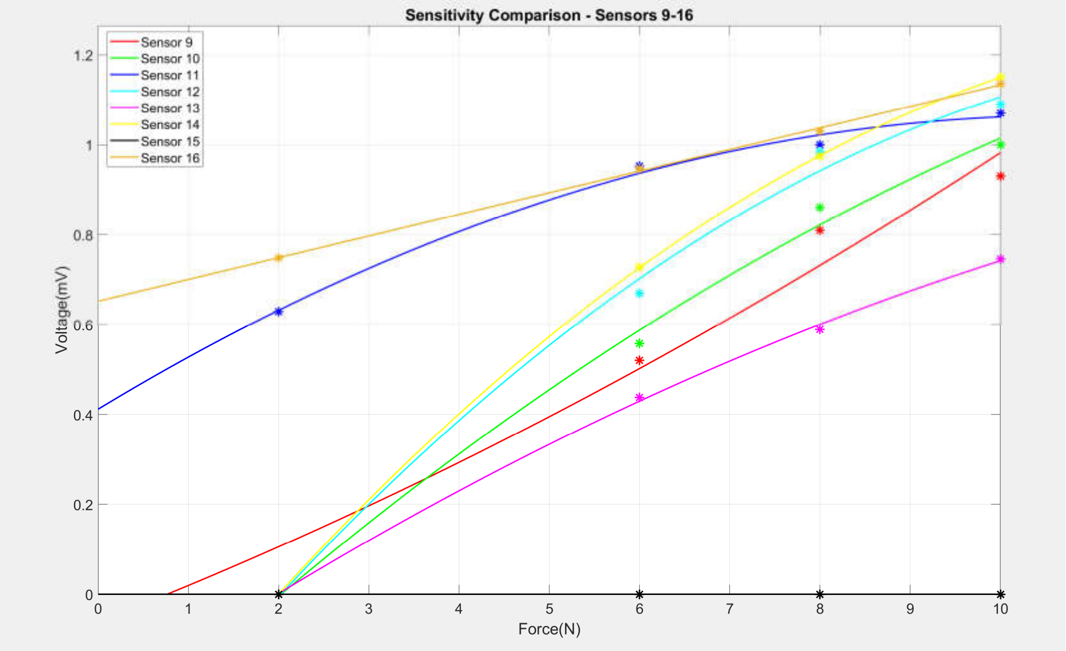


Figure 72:Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit.

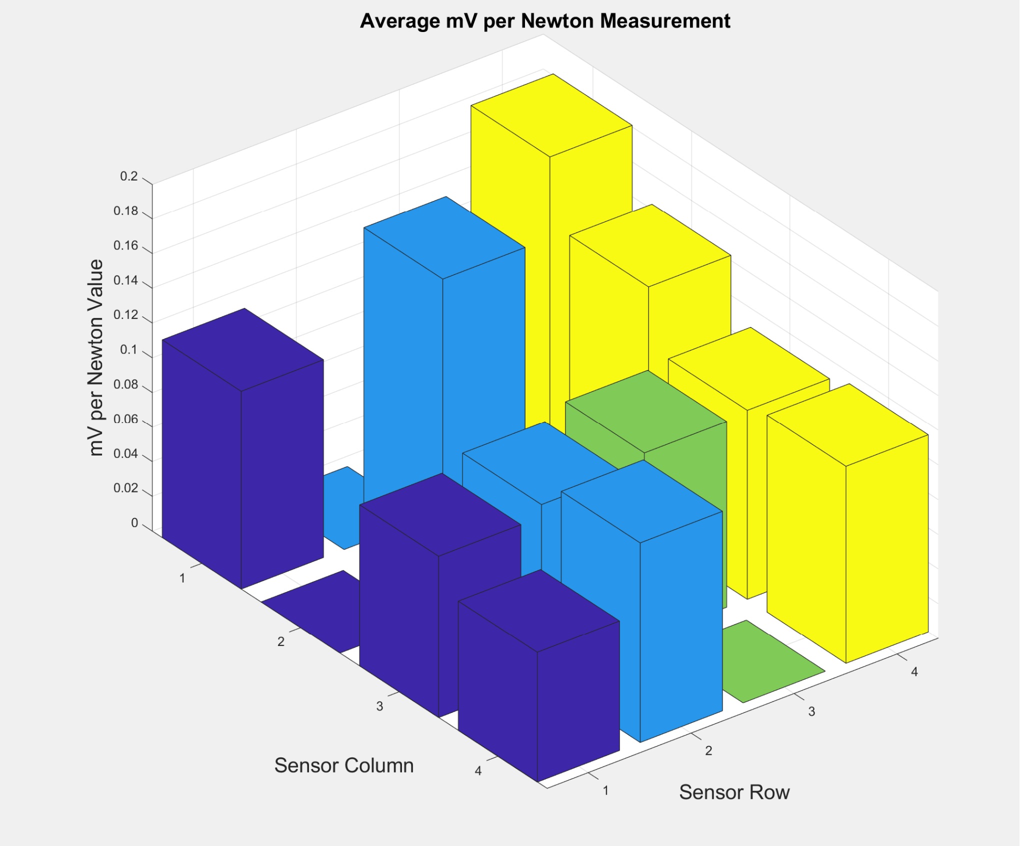


Figure 73: Average mV per Newton Measurement visualization

B2: 2021 IEEE Skin Sensor Publication Measurements

Sensor #	0.5N response(mV)	1N response(mV)	1.5N response(mV)	2N response(mV)	Sensitivity average(mv/N)
1	0.06565	0.059	0.0716	0.0868	0.0434
2	0	0	0	0	0
3	0	0.00132	0.03589	0.042	0.021
4	0.0251	0.0642	0.0813	0.0924	0.0462
5	0.1752	0.9517	2.035	2.151	1.0755
6	0.0475	0.104	0.186	0.212	0.106
7	0	0	0	0	0
8	0	0	0	0	0
9	0.00435	0.0312	0.0626	0.0806	0.0403
10	0.0241	0.0528	0.071	0.0846	0.0423
11	0	0.0112	0.0163	0.031	0.0155
12	0.00684	0.01673	0.0235	0.0278	0.0139
13	0.0122	0.124	0.124	0.124	0.062
14	0.0355	0.1513	0.185	0.2022	0.1011
15	0.0164	0	0.691	0.631	0.3155
16	0	0	0	0	0

Table 9: Average Response Value of sensors: 0 indicates nonfunctioning sensors

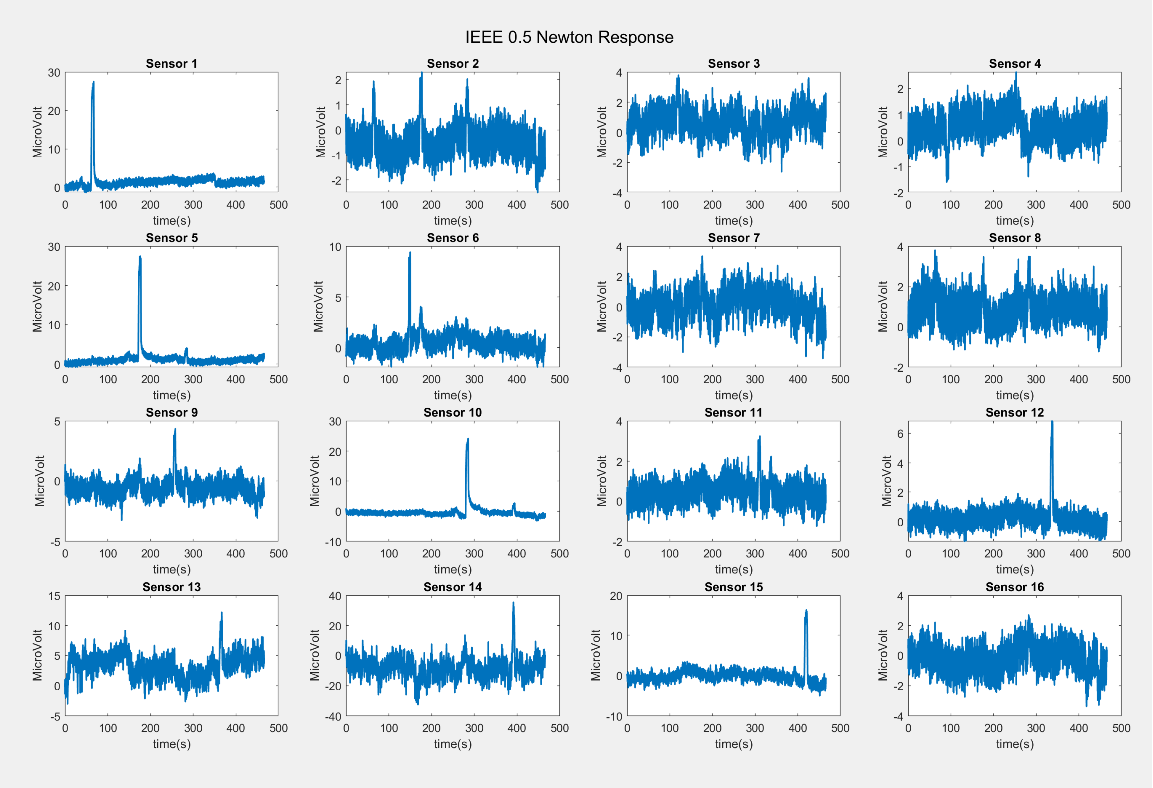


Figure 74: Visualization of 0.5 Newton Test Raw Response Data

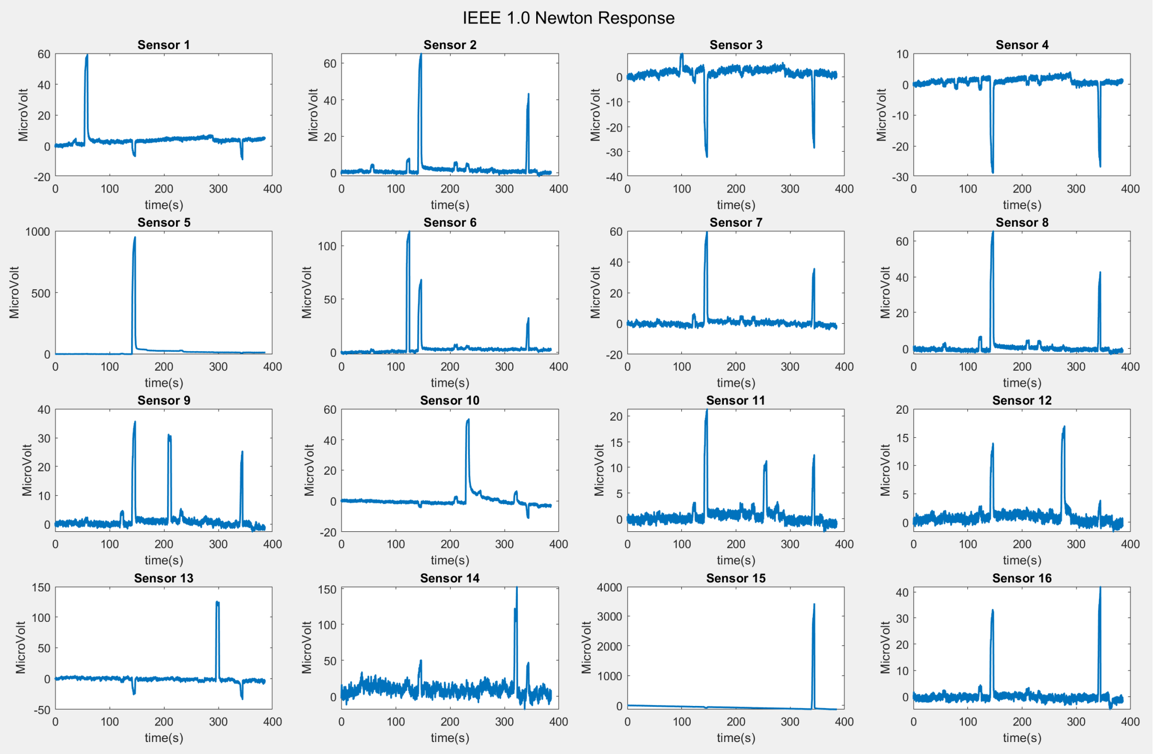


Figure 75: Visualization of 1.0 Newton Test Raw Response Data

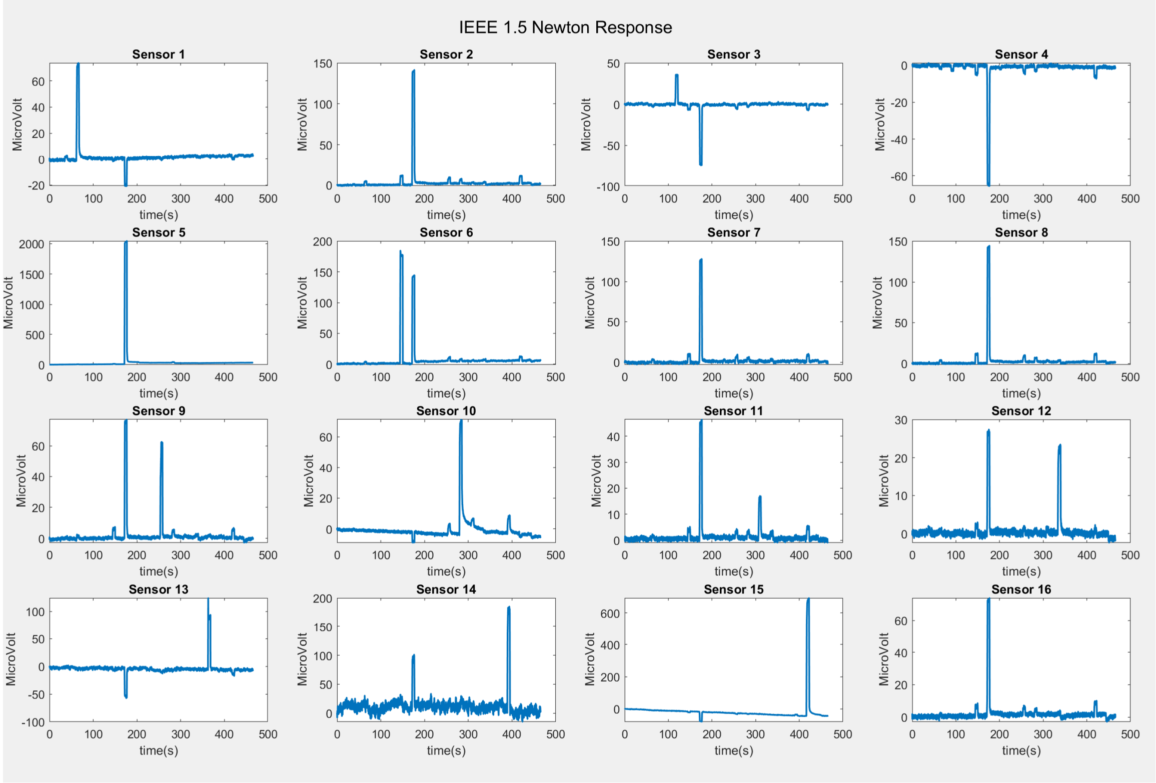


Figure 76: Visualization of 1.5 Newton Test Raw Response Data

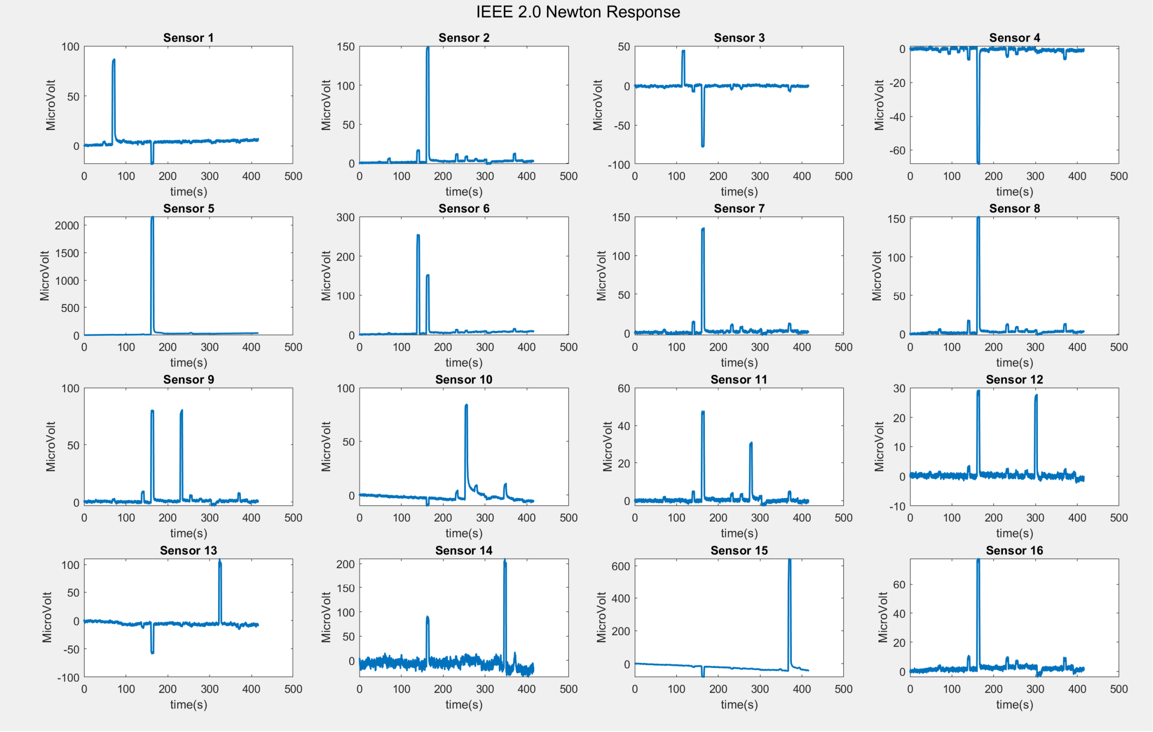


Figure 77: Visualization of 2.0 Newton Test Raw Response Data

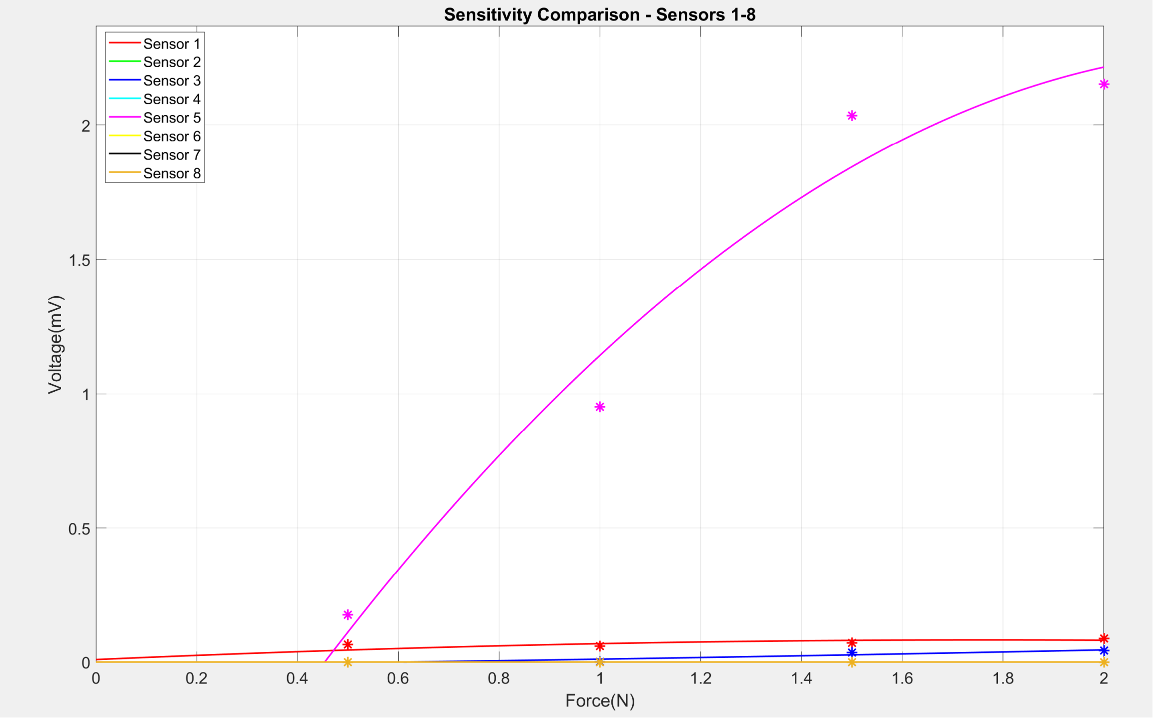


Figure 78: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit.

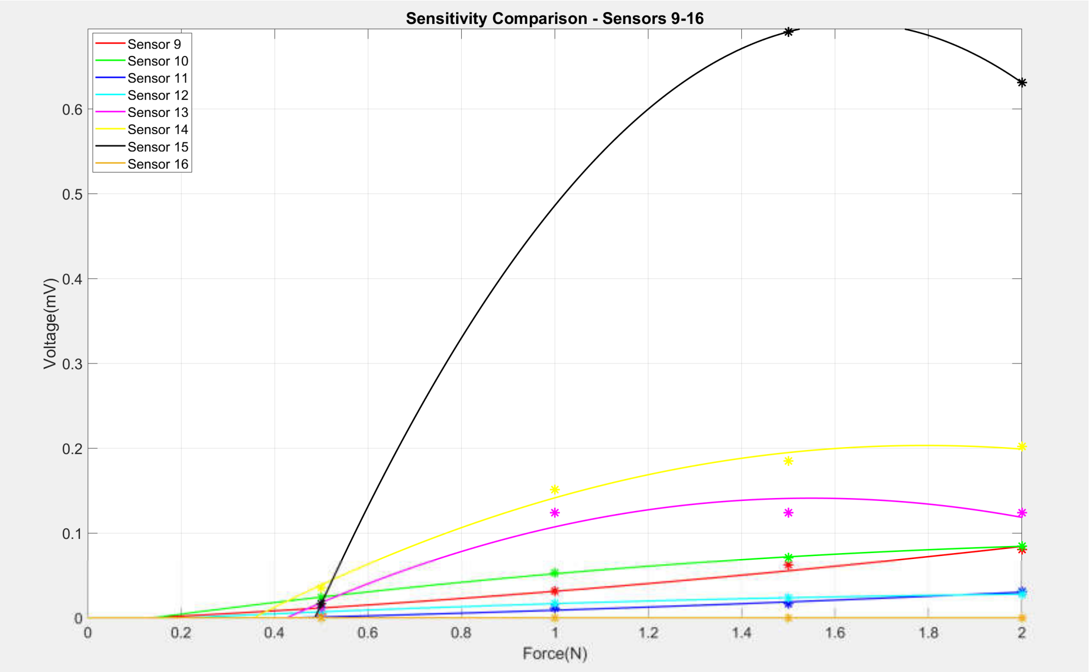


Figure 79: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit

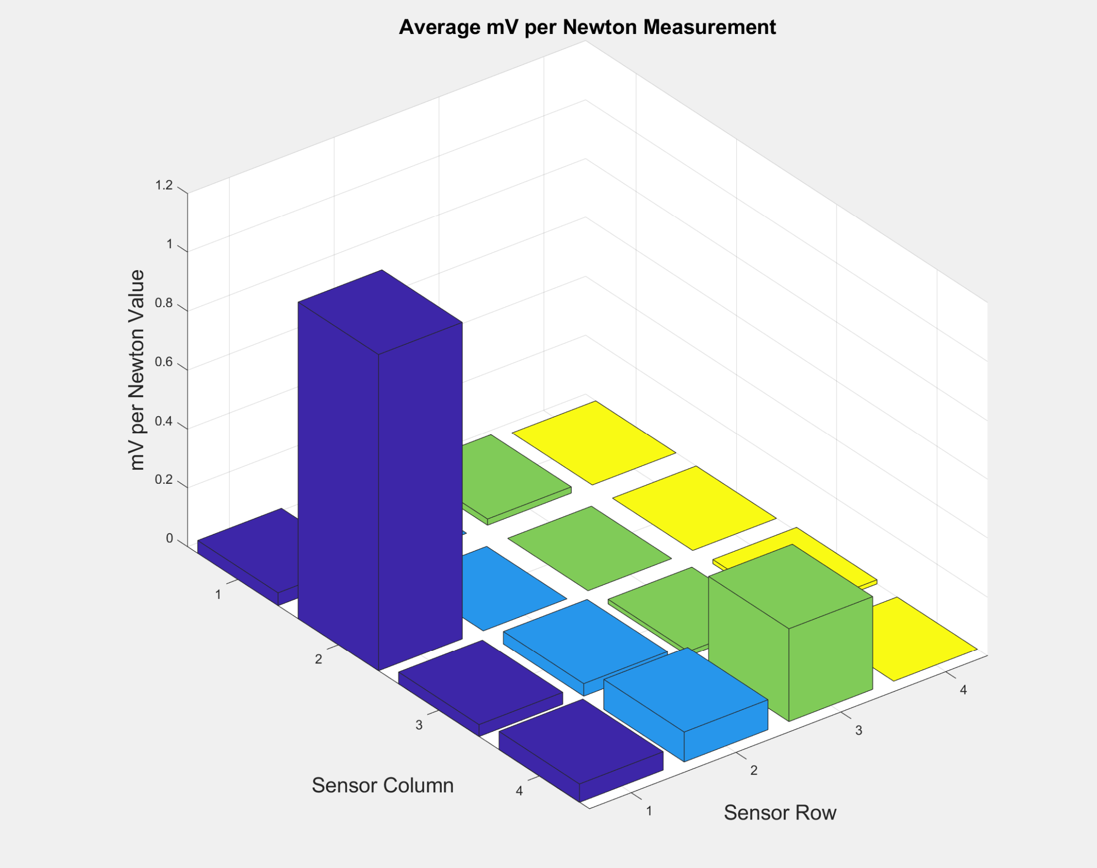


Figure 80: Average mV per Newton Measurement visualization

B3: PicoPulse Deposited Circular Tree with Differing Structures

Sensor #	0.5N response(mV)	2N response(mV)	4N response(mV)	6N response(mV)	Sensitivity average(mv/N)
1	8.875	18.57	34.461	45.55	7.591666667
2	0	0	0	0	0
3	6.39	11.64	17.71	24.2424	4.0404
4	8.59	18.899	31.15	37.838	6.306333333
5	0	0	0	0	0
6	0	0	0	0	0
7	4.724	11.012	15.686	24.85	4.141666667
8	8.5	15.96	21.14	30.84	5.14
9	7.08	11.14	15.06	23.2	3.866666667
10	0	0	0	0	0
11	5.976	10.67	20.491	31.32	5.22
12	11.91	25.11	50.122	67.47	11.245
13	0	0	0	0	0
14	10.61	19.66	23.56	34.01	5.668333333
15	5.5	11.72	18.3	25.71	4.285
16	6.55	12.07	17.5	24.19	4.031666667

Table 10: Average Response Value of sensors: 0 indicates nonfunctioning sensors

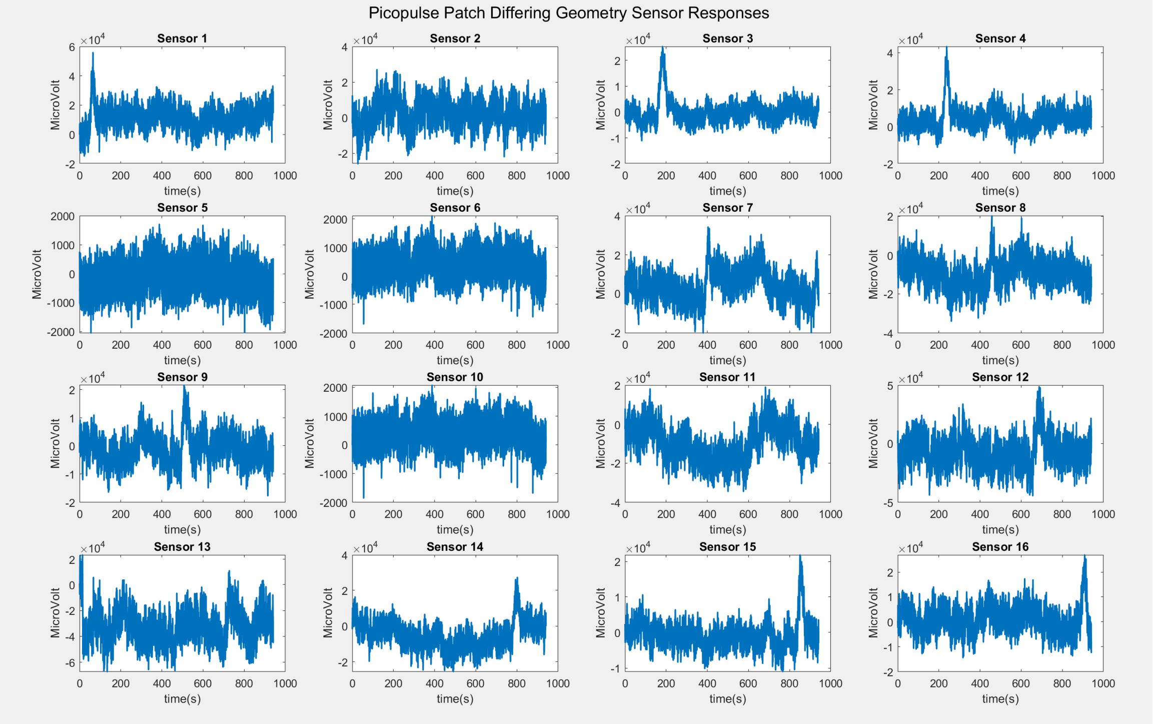


Figure 81: Visualization of Test Raw Response Data Across the Entire Test

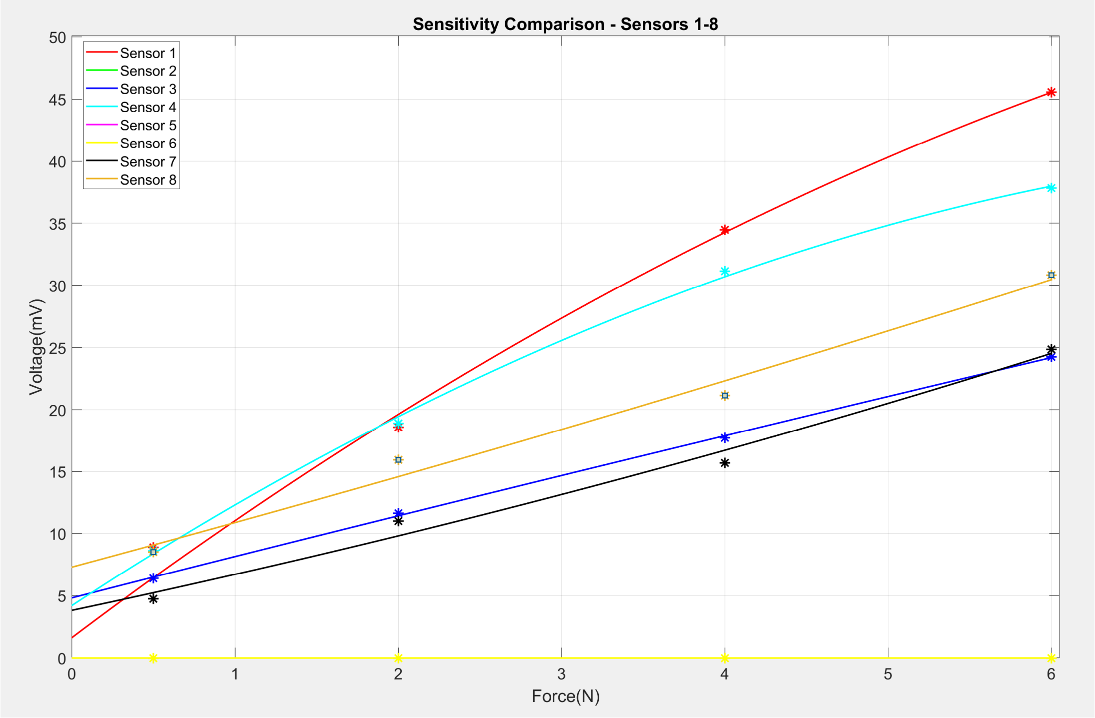


Figure 82: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit.

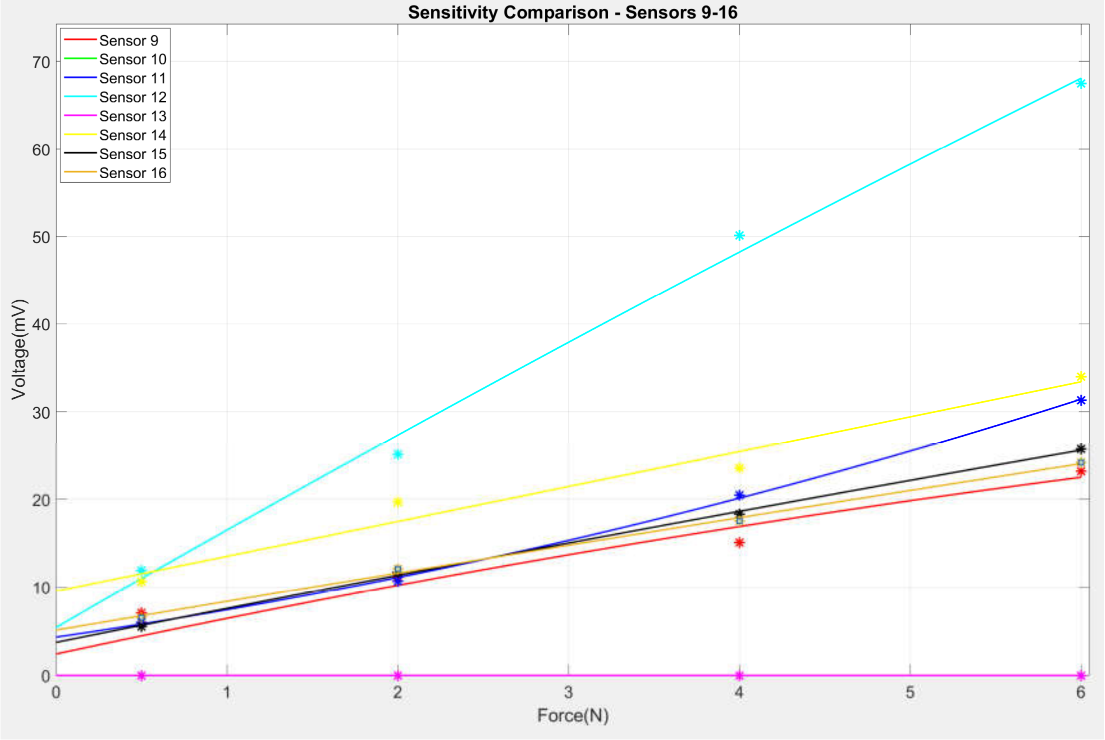


Figure 83: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 9-16 including a poly fitted line of best fit.

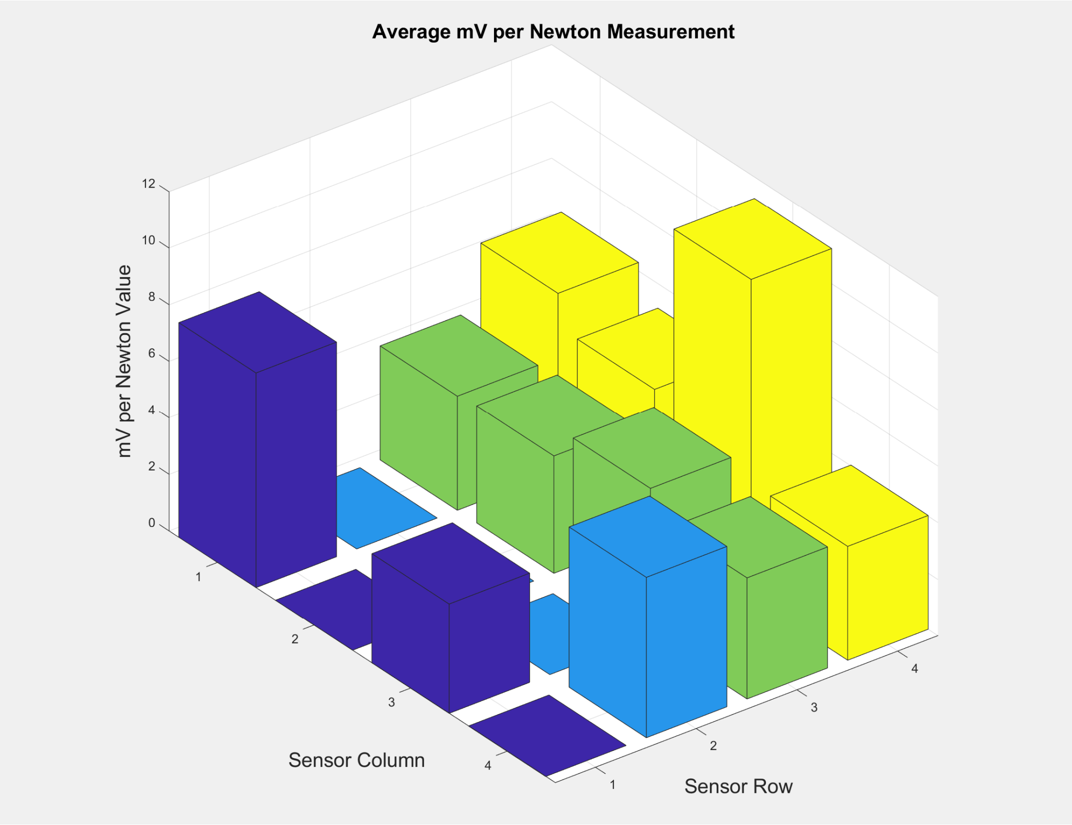


Figure 84: Average mV per Newton Measurement visualization

B4: PicoPulse Deposited Circular Tree with Similar Structures

Sensor #	0.5N response(mV)	2N response(mV)	4N response(mV)	6N response(mV)	Sensitivity average(mv/N)
1	12.11	24.51	35.32	42.68	7.113333333
2	0	0	0	0	0
3	9.37	17.35	48.77	67.21	11.20166667
4	0	0	0	0	0
5	8.08	13.6	17.23	33.01	5.501666667
6	6.1	12.47	21	38.21	6.368333333
7	6.91	13.71	17	21.15	3.525
8	0	0	0	0	0
9	9.5	15.2	19.87	25.75	4.291666667
10	6.48	14.65	19.94	41.7	6.95
11	0	0	0	0	0
12	3.54	8.66	15.12	28.74	4.79
13	0	0	0	0	0
14	10.61	14.5	21.73	65.5	10.91666667
15	8.57	16.82	51.68	60.38	10.06333333
16	0	0	0	0	0

Table 11: Average Response Value of sensors: 0 indicates nonfunctioning sensors

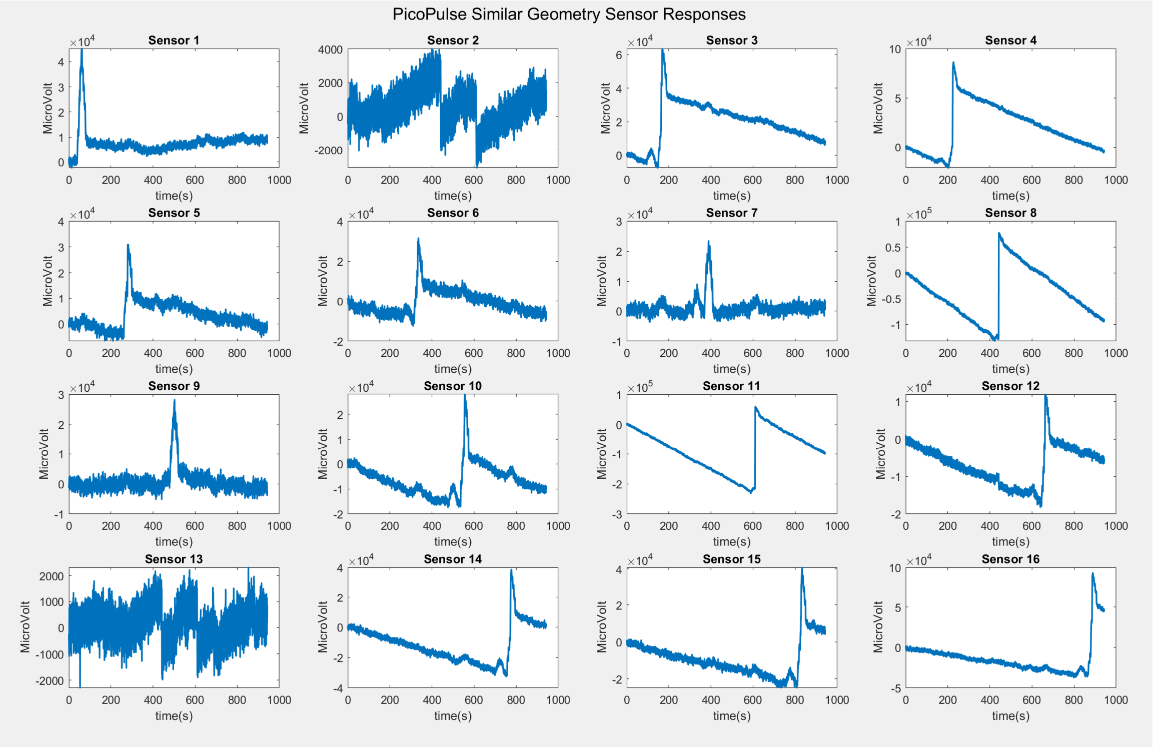


Figure 85: Visualization of Test Raw Response Data Across the Entire Test

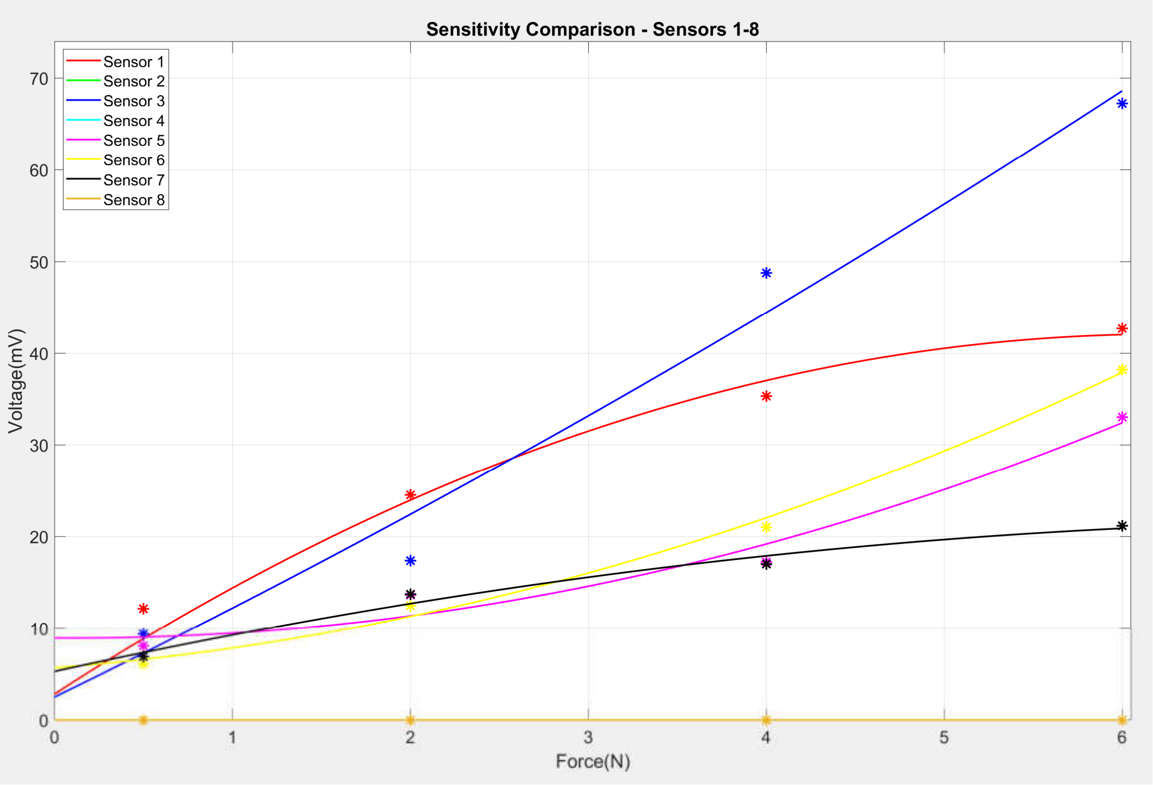


Figure 86: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit.

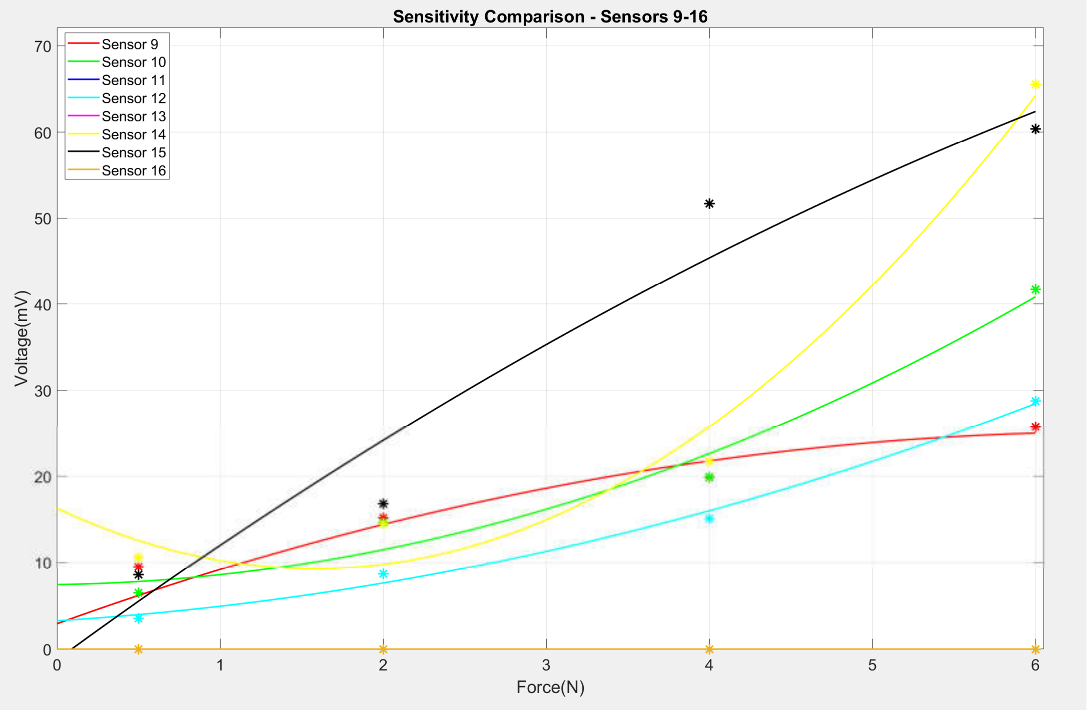


Figure 87: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 9-16 including a poly fitted line of best fit

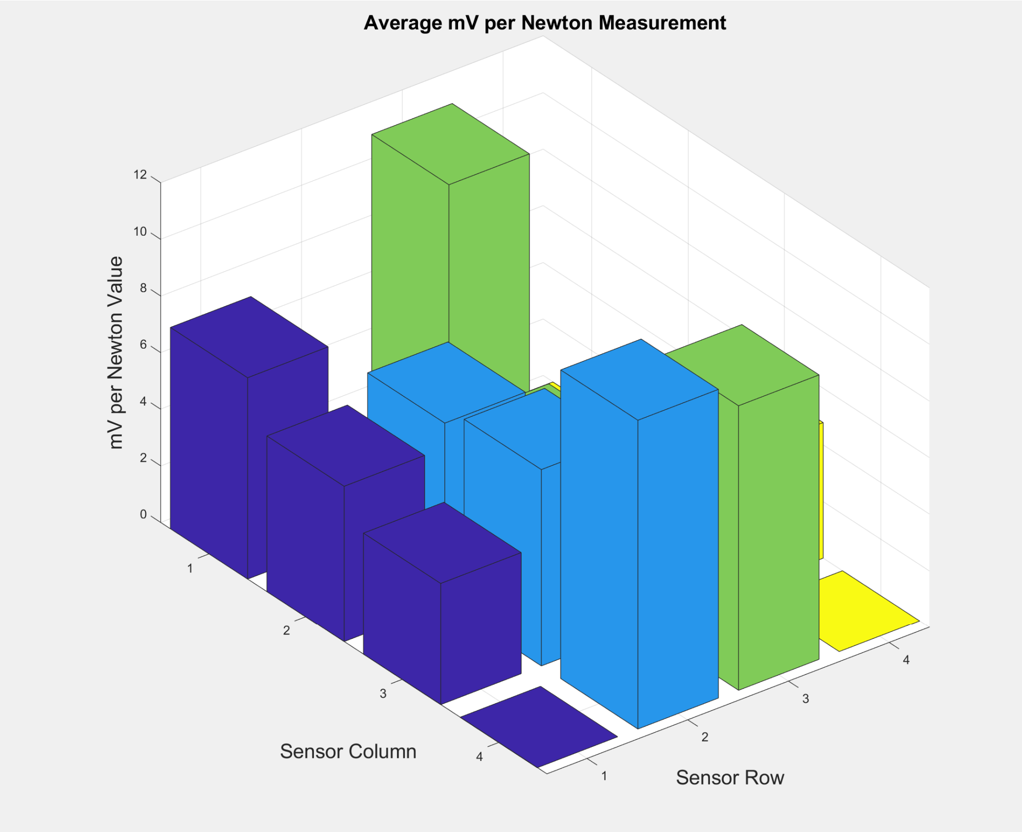


Figure 88: Average mV per Newton Measurement visualization

B5: Optomec Fabricated Sensor –SkinCell 1

Sensor #	0.5N response(mV)	2N response(mV)	4N response(mV)	6N response(mV)	Sensitivity average(mv/N)
1	0.01144	0.02454	0.03694	0.05164	0.008606667
2	0.0366	0.0446	0.05505	0.06085	0.010141667
3	0.0041	0.0142	0.0229	0.0291	0.00485
4	0.010805	0.01774	0.02347	0.02727	0.004545
5	0.009098	0.01333	0.017715	0.021995	0.003665833
6	0.00855	0.01435	0.01805	0.02115	0.003525
7	0.018	0.036	0.047	0.0537	0.00895
8	0.012	0.0198	0.025	0.0278	0.004633333
9	0.01891	0.02815	0.0337	0.036	0.006
10	0.009235	0.01558	0.0193	0.0233	0.003883333
11	-0.009	-0.00445	0.0026	0.0112	0.001866667
12	0.03243	0.04495	0.0531	0.0659	0.010983333
13	-0.004655	0.0237	0.06625	0.09735	0.016225
14	0.033	0.046	0.054	0.062	0.010333333
15	0.026	0.0331	0.0396	0.0471	0.00785
16	0.02	0.0355	0.0465	0.0565	0.009416667

Table 12: Average Response Value of sensors: 0 indicates nonfunctioning sensors

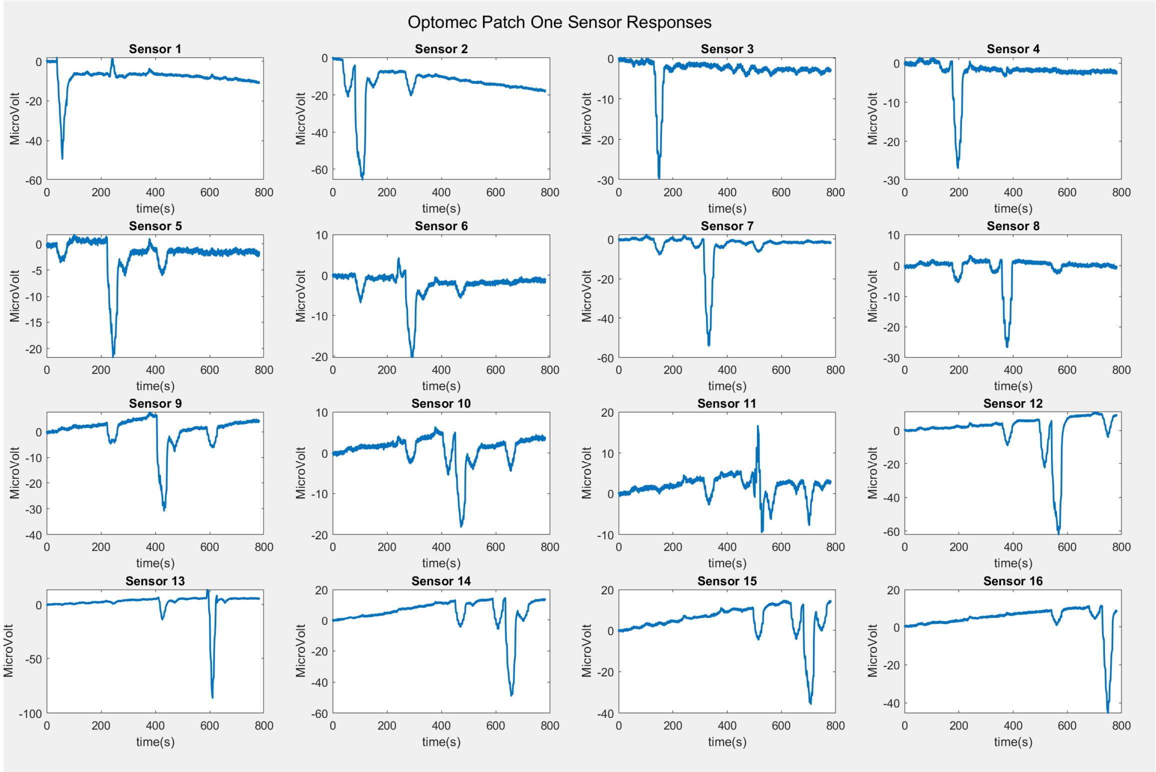


Figure 89: Visualization of Test Raw Response Data Across the Entire Test

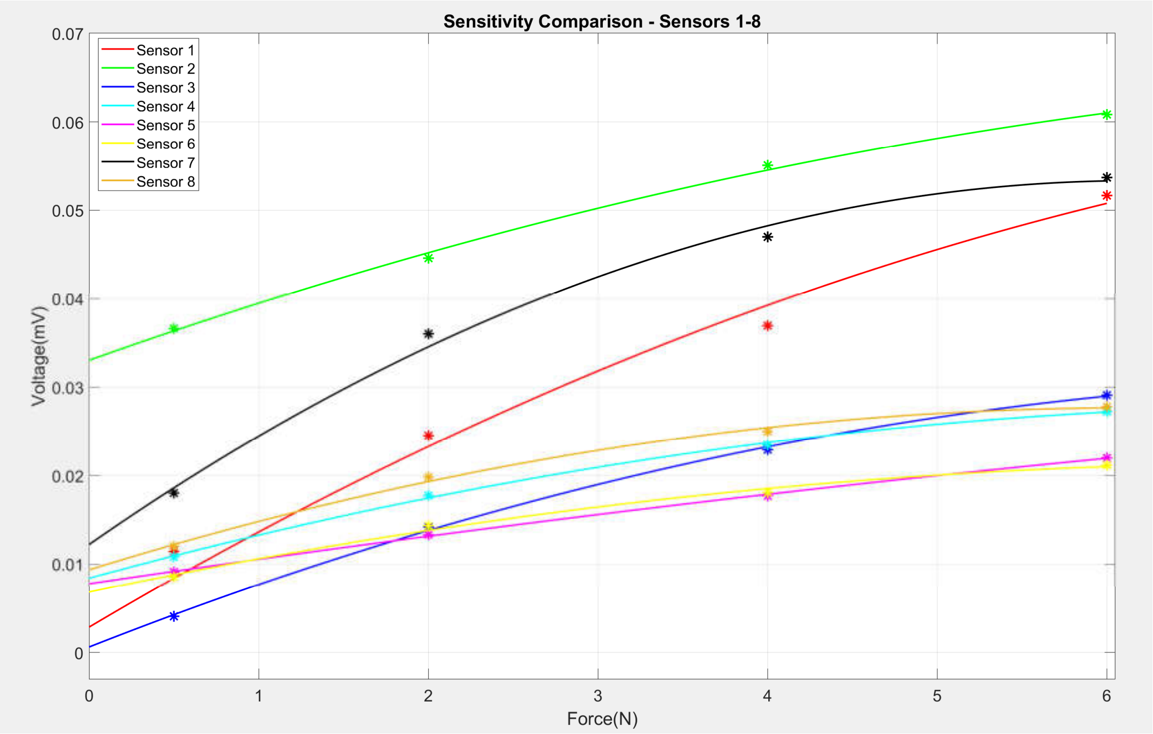


Figure 90: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit

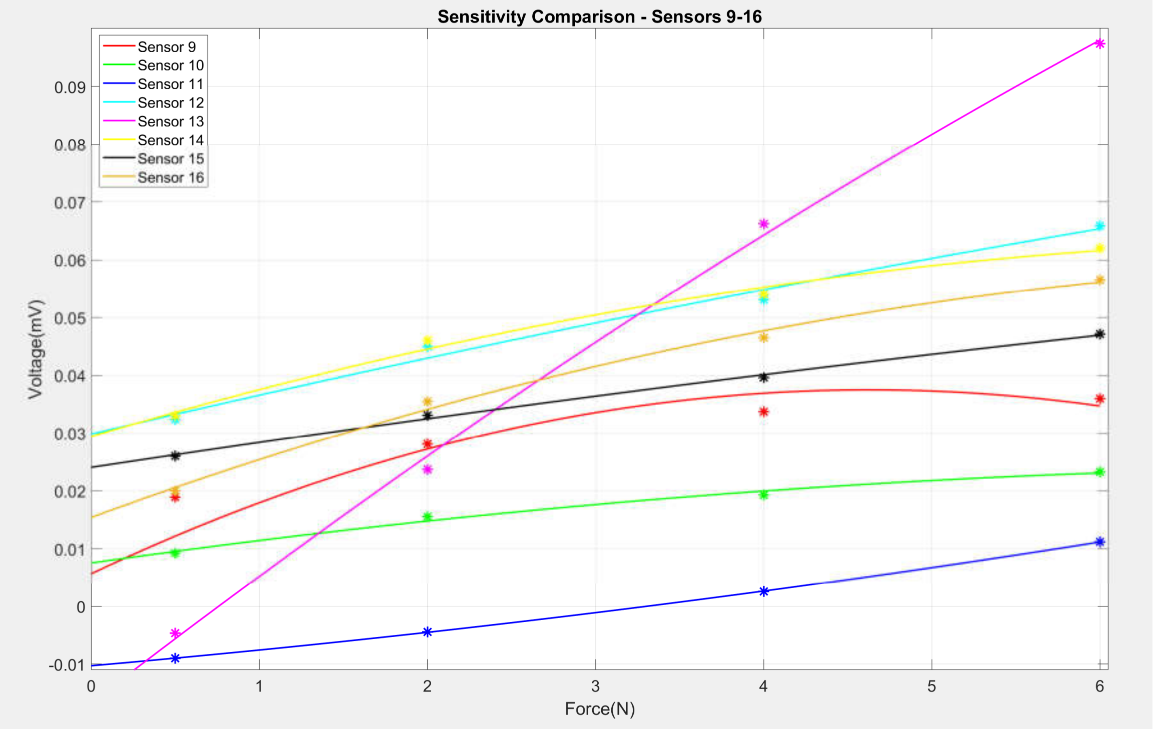


Figure 91: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 9-16 including a poly fitted line of best fit

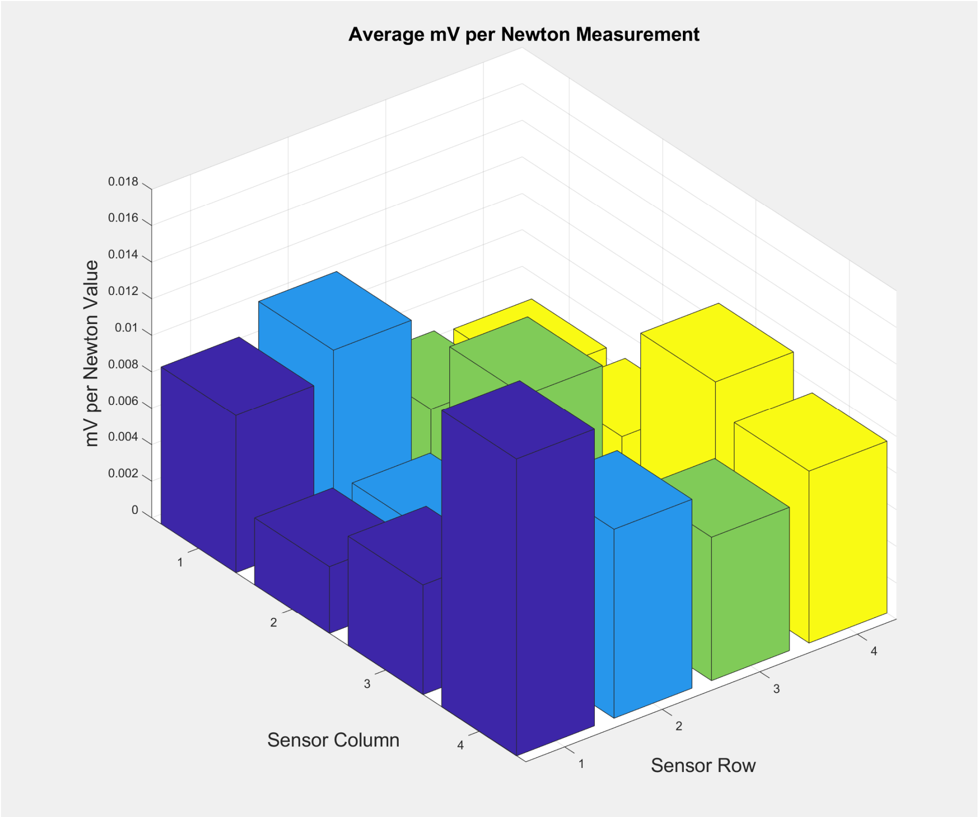


Figure 92: Average mV per Newton Measurement visualization

B6: Optomec Fabricated Sensor –SkinCell 2

Sensor #	0.5N response(mV)	2N response(mV)	4N response(mV)	6N response(mV)	Sensitivity average(mv/N)
1	0.00995	0.0221	0.0267	0.0315	0.00525
2	0.00682	0.01401	0.017625	0.021835	0.003639167
3	0.02778	0.03748	0.04628	0.05108	0.008513333
4	0.01272	0.02805	0.0418	0.0572	0.009533333
5	0.005	0.012625	0.01919	0.02289	0.003815
6	0.007635	0.01558	0.019465	0.022535	0.003755833
7	0.0086	0.01892	0.02198	0.02562	0.00427
8	0.01152	0.0159185	0.019965	0.023235	0.0038725
9	0.01995	0.03056	0.03935	0.0454	0.007566667
10	0.0046	0.01305	0.017605	0.020165	0.003360833
11	0.007364	0.01066	0.01371	0.01615	0.002691667
12	0.018158	0.02322	0.02715	0.03055	0.005091667
13	0.0058	0.0096	0.01215	0.01815	0.003025
14	0.0154	0.02455	0.0334	0.0381	0.00635
15	0.0229	0.02707	0.0324405	0.0386705	0.006445083
16	0.0175	0.03798	0.04319	0.05204	0.008673333

Table 13: Average Response Value of sensors: 0 indicates nonfunctioning sensors

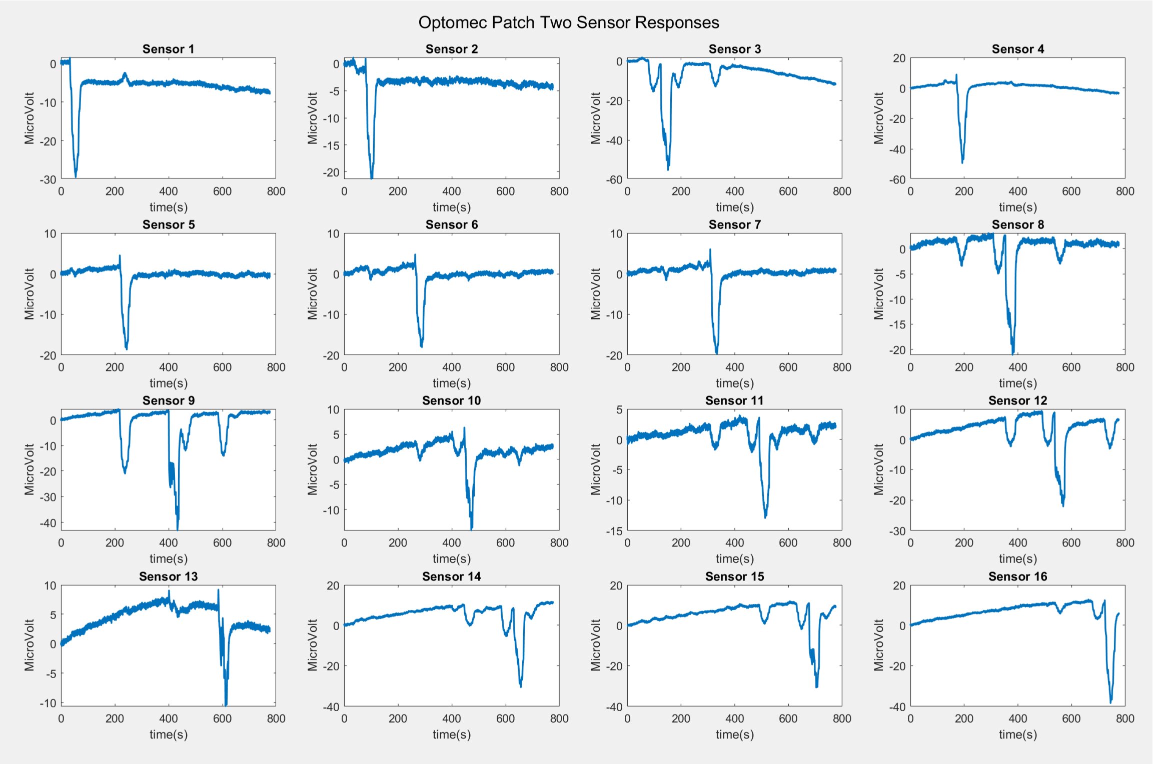


Figure 93: Visualization of Test Raw Response Data Across the Entire Test

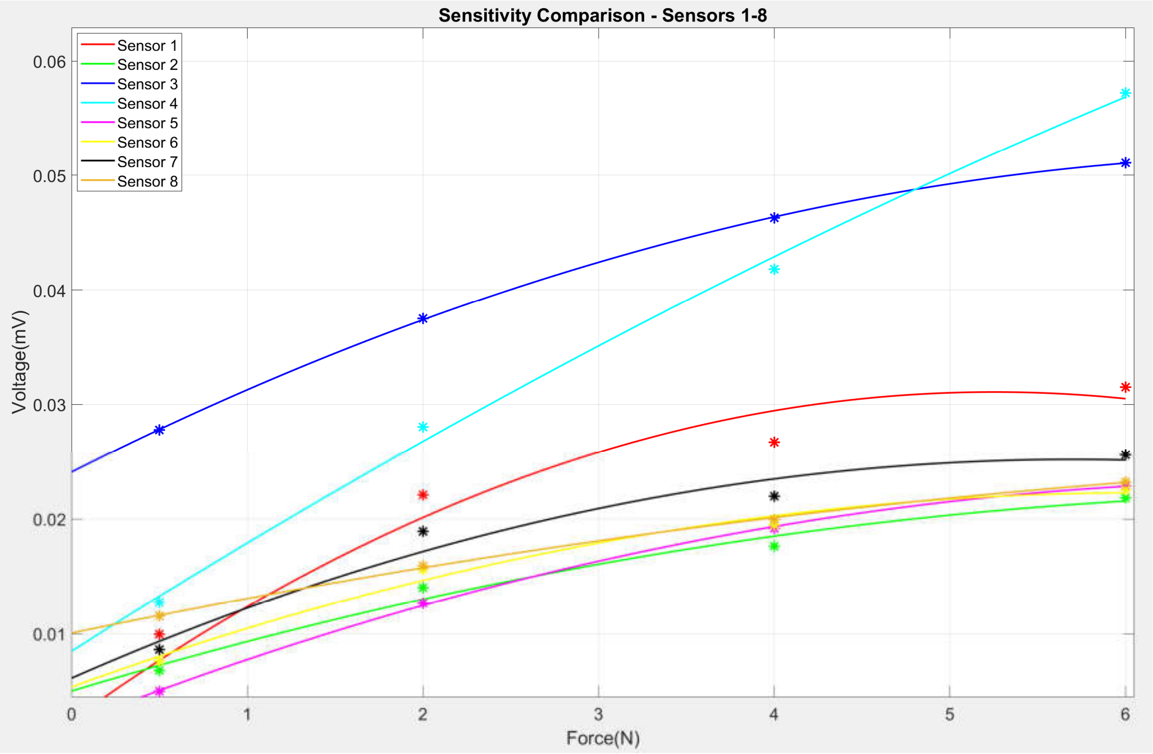


Figure 94: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 1-8 including a poly fitted line of best fit

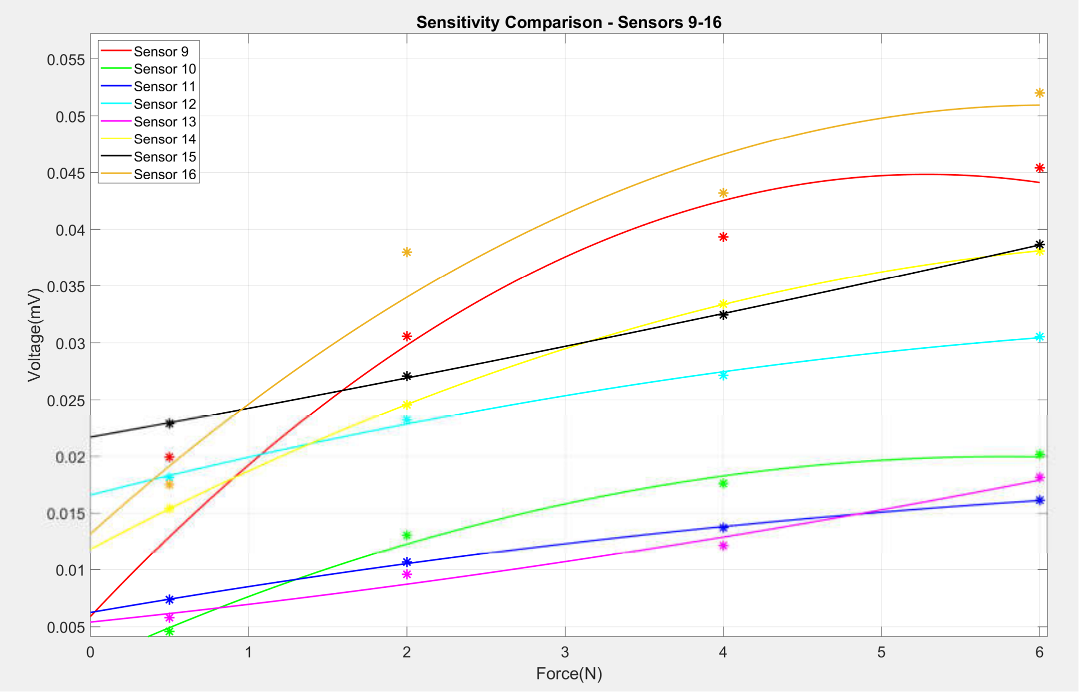


Figure 95: Comparison of working Sensitivity, Voltage with Respect to Force, Across Sensors 9-16 including a poly fitted line of best fit

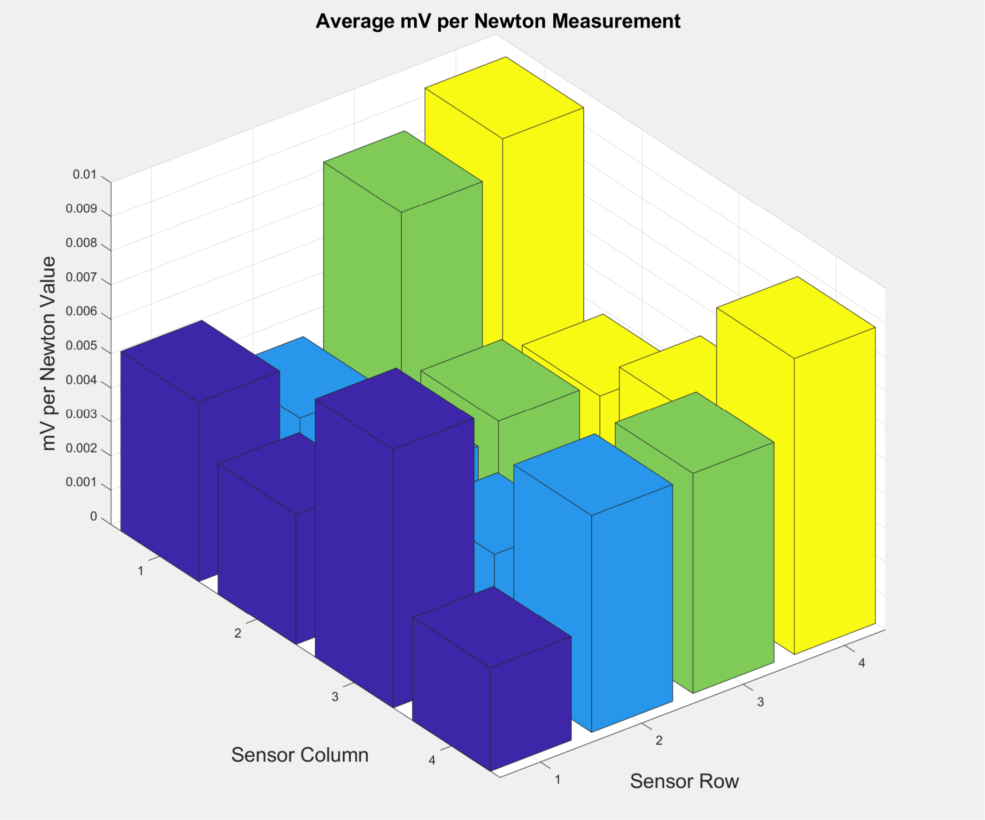


Figure 96: Average mV per Newton Measurement visualization

CURRICULUM VITA

NAME: Brian P. Goulet

DOB: Columbia, MD – Oct 14, 1997

EDUCATION: B.S., Electrical Engineering
University of Louisville
2016-2020