HIGH SCHOOL EDITION
Journal of Student Research

Soft Robotics in Body Assistance: Intelligent Rehabilitation Device with Soft Continuum Actuation

Ria Jain[1], Kunal Kapur[2], Jiaqi Wang[3], Yin yu[4], Diarmid Flatley[4], and Lina Lim[4]

[1]Dublin High School, Dublin, CA, USA
[2]Evergreen Valley High School, San Jose, CA, USA
[3]Woodbridge High School, Irvine, CA, USA
[4]University of California, Santa Barbara, CA, USA

ABSTRACT

In an effort to improve rehabilitation devices, the applications of soft robotics technologies to prosthetics and physical therapy were explored, particularly due to the benefits of the inherent properties of soft materials. A conceptual design for a soft robotics device prototype is proposed to assist with physical therapy for wrist tendonitis and arthritis, carpal tunnel syndrome, fractures and sprains, and compromised motor skills due to chronic stroke. The device assists in four motions that are commonly performed in wrist therapy: flexion, extension, and rotation at the proximal radioulnar joint (supination and pronation) using soft pneumatic actuators to guide movements. The distinct directions were achieved by varying the lateral and radial strain limiting layers. The device uses embodied intelligence to make the device dynamically adaptable in real time, allowing for a customizable recovery process. A detailed model of the device was developed, and the viability of the design was assessed using a suite of state-of-the-art simulation tools and limited hardware prototyping. Simulations were performed through integration of Rhinoceros 3D, Grasshopper 3D, Firefly, an Arduino microcontroller, biosensors, Python scripting, and visual parametric programming. Pressure and materials were simulated and tested in Simulia Abaqus and Autodesk Fusion 360. Several parametric variations were tried using simulations and the predictions revealed that rubber silicone at a pressure of 10 kiloPascals is the optimal choice.

I. Introduction

In recent decades, robots have become integrated into many aspects of daily life across a wide range of fields and industries. Alongside, soft robots have rapidly emerged and are receiving more widespread attention due to their increased capabilities, resilience, and deformability, presenting plethora of new applications. Rigid robots are often limited in their capabilities due to their inherently rigid structure, meaning they can only be controlled through a series of rigid links. Furthermore, conventional robots are more susceptible to breakage and lack the ability to adapt to their environment dynamically. Although rigid robots can be accurate, precise, and efficient, an increased demand in Human-Computer Interaction (HCI)—the communication and interplay between technologies and humans—has spiked both new interest and technological breakthroughs in soft robotics because they are considerably more adjustable than their rigid counterparts.

Due to the flexible qualities of their materials, soft robots are being increasingly explored in wearable devices in prosthetics and physical therapy. An example of a rigid rehabilitation device is SaeboFlex, which uses multiple splints to move the fingers. Although functional, the device is quite bulky and uncomfortable, and cannot—by the nature of the material—be positioned on the human body well. Therefore, users of the device have expressed difficulty and inconvenience with the fitting process [12]. By definition, soft robots are built of flexible materials (e.g. rubber, silicone, elastomers, nylon 6, etc.), which contrast the rigid build of traditional solid metals used in conventional robots. Because its materials are flexible, a soft robot is much better suited to be a wearable device on the body, giving a more effective physical therapy experience. In devices that are attempting to improve the motor skills of a body part, the optimal performance is achieved when the device matches the shapes and contours of the body closely because that is when the device will have the most control, ensuring that its movement is transferred to the limb effectively. Because rigid robots are simply a series of joints and links, bodily geometries are virtually impossible to achieve. Additionally, soft robots can use a variety of actuation techniques, including pneumatic inflation and deflation, pressure regulation of the actuator, response to stimuli using a material's inherent properties through environmental and material exploitation, and many others. The most important feature of soft robotics actuators, in this application, is continuous actuation. In rigid robots, the actuation is discrete, which means there is a set number of degrees of freedom (DOF) that are controlled. However, soft robotic actuators can be intelligently controlled using embodied intelligence, soft sensors, and intricate approximation algorithms based on neural networks and reservoir computing, which improve the quality of rehabilitation by creating an increased capacity for imitating human movement. Finally, traditional robots are often heavy and bulky and, therefore, lack portability and mobility, meaning they can mostly only be used in specially designated facilities, inconveniencing the user financially, timewise, and energy-wise. Conversely, soft robots can be designed as portable and lightweight wearable devices, which is both convenient and effective for its purpose.

In this work, a new prototype was developed to assist with physical therapy for wrist tendonitis and arthritis, carpal tunnel syndrome, fractures and sprains, and compromised motor skills due to chronic stroke. A literature survey of similar devices is presented in the next section. The techniques employed in the design of the novel device proposed in this work is discussed next, followed by details on the modeling of the actuation process. Finally, simulations and analyses of results are presented, and key conclusions are drawn at the end.

II. A Review of Current Soft-Robotics Devices

The current state-of-the-art in the applications of soft robotics to body assistive devices, particularly prosthetics and physical therapy, is presented, which helped the current research in identifying potential areas for improvement.

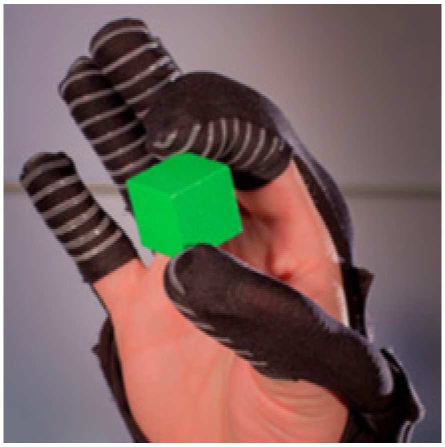


FIG. 1. Soft Robotic Glove for Task Specific Training. A gripping device by the Harvard University Biodesign Lab to assist with hand motor control using pneumatic actuation. Credit: The Wyss Institute.

Harvard University Biodesign Lab has designed a biomimetic grasping glove for patients who have lost hand motor abilities (see Figure 1). In order to control hand movements, the researchers used soft actuators made of elastomer, a material that is valuable for its simultaneous resilience and deformability. The research in this lab motivated the examination of constraining actuators, similar to the techniques used in this glove. These materials are biologically inspired, which was part of the research in biomimicry conducted in this lab. The actuators were controlled pneumatically through manipulating the internal air pressure via a separate manual control box. This device motivated the consideration of biomimetics for the research on the methods and materials for our device. The success of this device prompted the present research to further explore intuitive control and the use of embodied intelligence based on intelligent prediction techniques for control, rather than using a manual control box.

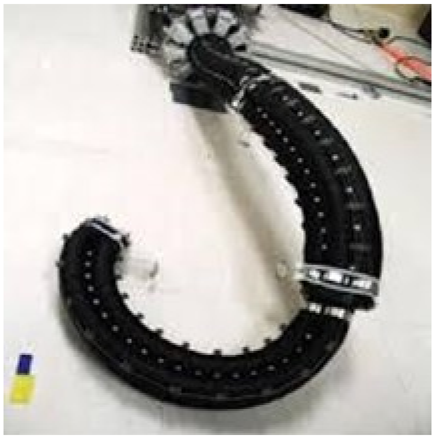


FIG. 2. X-Limb, a 3D printed soft robotic prosthetic hand. X-Limb is one of the first 3D-printed soft robotic hands, used to understand the compromise between practicality and performance in such devices Credit: Mohammadi et al.

Because soft robotics is still an emerging and rapidly evolving field, there is a general lack of balance between practicality and performance in devices. A new device, X-Limb (see Figure 2), explored the trade-off between design feasibility and effectiveness so that the device is (1) comfortably wearable by the user, (2) practically manufacturable in large scales, (3) easily customizable to a liberal range of grasping movements, and (4) reasonable in size and weight. This case study gave the present research the consideration of exploiting the inherent properties of a material as an alternative to brute-force control methods and potentially reduce the amount of needed control.

FIG. 3. OCTARM VI Soft robotic actuator controller. The OCTARM, a biomimetic soft continuum actuator controller. Credit: I. Walker, Clemson University

Another example of the biomimetic controllers is the OCTARM controller series. In general, continuum actuators are more difficult to control. However, new breakthroughs in computer science and artificial intelligence provide new methods for the intelligent control of such systems. OCTARM VI (see Figure 3), is a dynamic manipulator for soft continuum robots, taking biomimetic inspiration from an octopus tentacle. It was constructed using McKibben pneumatic actuators which are high-strain extensor air muscles commonly used in prosthetic and physical therapy devices. OCTARM VI has data acquisition boards for input (actuator length). The output is the voltage level of the pressure regulators. This device motivated the consideration of reservoir computing in nonlinear systems in the proposed device as a means of accomplishing intuitive control in lieu of manual control.

There has also been some research in using soft robots for wrist rehabilitation. One device has explored using pneumatic actuators to imitate flexion, extension, radial and ulnar deviation, and circular movements of the wrist [21]. The rubber actuators artificially imitated muscle movement and prompted the present research to consider the materials of the device and explore the costs and benefits of each to determine which are best suited for wrist rehabilitation on the basis of portability, convenience, functionality, and nature.

In the following section, the novel the device proposed in the current work is described along with the techniques employed.

III. Device Techniques

A. Device Overview

The proposed device prototype specifically focuses on the rehabilitation of the wrist. Such a device can be useful in a number of applications: chronic stroke victims, who often have weakened dexterity and motor skills in the hands and wrist; carpal tunnel syndrome improvement physical therapy exercises, both preceding and succeeding corrective surgery; wrist tendonitis and arthritis, which is quite common; and wrist sprains and fractures. In all of these instances, it is valuable to gradually complete physical therapy exercises to recover strength. In response to this, our device implements four main motions: flexion, extension, and rotation at the proximal radioulnar joint (supination and pronation) through controlled actuator pressurization and embodied intelligence.

B. The Actuation Process

The main factors considered for the actuation process were: comfort, mobility, practicality, and effectiveness. Based on the research that was carried out, three different methods for actuation were down-selected: (1) response to stimuli based on the inherent properties of a material, (2) manipulating pressure through on-device regulators, and (3) motors. The first method was not very effective for our purposes, simply because a proper stimulus or a proper material to react to the stimulus could not be identified. The third method was determined to be bulky and rigid. Ultimately, pneumatics (method 2) was chosen since it provided the user with the most comfort, had the most potential for mobility, and complied with our actuator design the best [11].

As seen in Figure 4, the device has a secure brace that wraps around the wrist and contains a total of four actuators on the wrist. Included on the device is the microcontroller, a flex sensor to measure the angle of the wrist (including direction), a haptic sensor for ease of control, pneumatic tubes for inflation, and other components.

For a device to be successful, the following qualities need to be assessed: (1) actuation techniques and properties, (2) actuation performance, and (3) practicality of such a device. In this paper, these areas are assessed through design exploration using computer-aided design (CAD) models and simulations of the actuation process using a number of commercial simulation software as described in later sections.

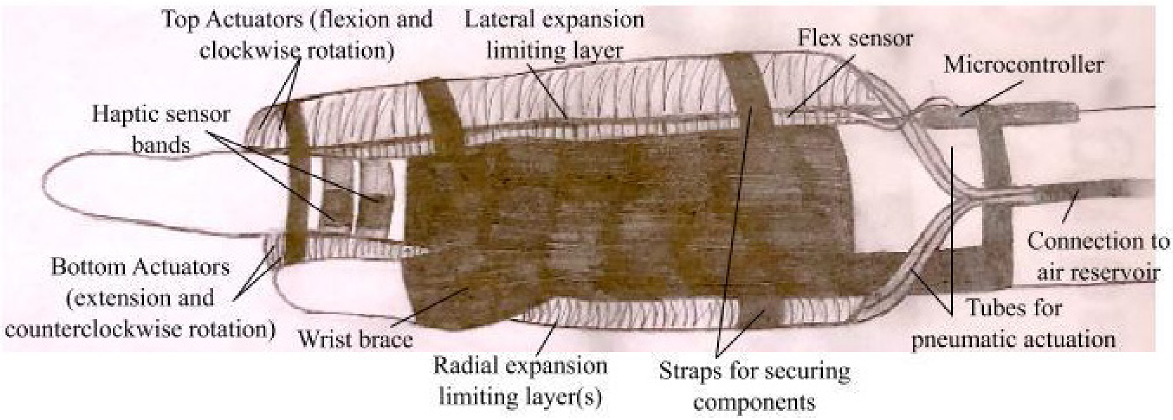

Top Actuators (flexion and clockwise rotation)
Lateral expansion limiting layer
Flex sensor
Microcontroller
Haptic sensor bands
Connection to air reservoir
Bottom Actuators (extension and counterclockwise rotation)
Wrist brace
Tubes for pneumatic actuation
Radial expansion limiting layer(s)
Straps for securing components

FIG. 4. The initial sketch of our prototype developed prior to modeling and simulating.

C. Actuator Motions and Control

In order to achieve the four motions of the device described above, the actuator movement in the direction of interest was constrained in two ways. The first was the control of lateral expansion (see Figure 5). When the actuator is inflated, it is constrained such that it would not expand in length. This was achieved using a limiting layer of a material with a higher Young's modulus, a measure of how easily an object can stretch or deform. The second was the control of radial expansion (see Figure 6). This requires two similar radial constraints, one in the clockwise direction and the other in the counterclockwise direction to allow for both supination and pronation. This method was adapted from Harvard University Biodesign Lab's design, but expanded the capabilities of the actuators beyond bending. For each desired wrist motion, a specific combination of constraints was used (see Figures 7-10). When the actuator is placed on a certain part of the hand, it will cause the wrist to imitate that same motion.

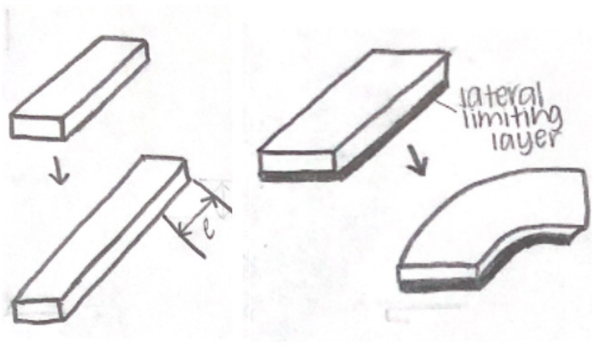


FIG. 5. When no lateral limiting layer is present, the actuator expands laterally. When the layer is added, the actuator bends instead, as desired, because it is no longer constrained to expand in length.

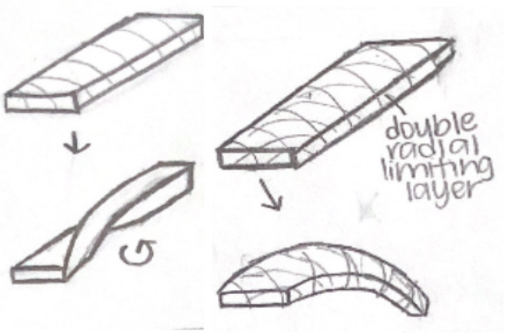


FIG. 6. When the radial limiting layer is added in one direction the actuator will twist when inflated. When it is constrained in both directions, the actuator bends.

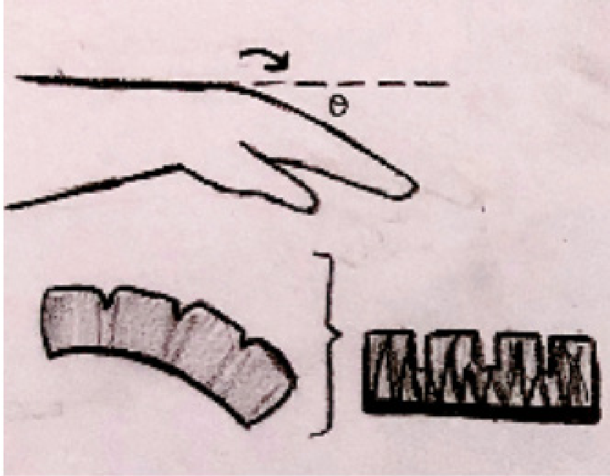


FIG. 7. Flexion. In order to get the wrist to flex, it is important to add a strain limiting layer to constrain lateral expansion and radial limits in both directions, so that the actuator bends.

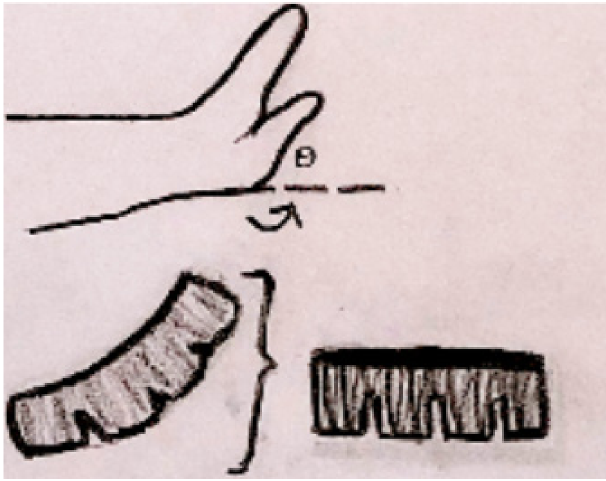


FIG. 8. Extension. Similar to Figure 7, three constraining layers are used. Essentially, it is the same actuator, but by placing it on the bottom of the hand, the opposite motion is achieved.

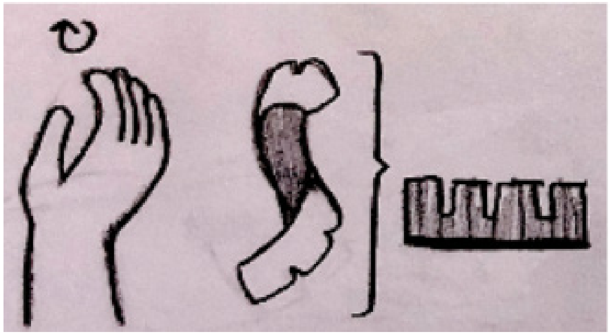


FIG. 9. Supination and pronation (right hand). In addition to bending, the actuator must now twist, which means one less radial constraint is added (see Figure 8).

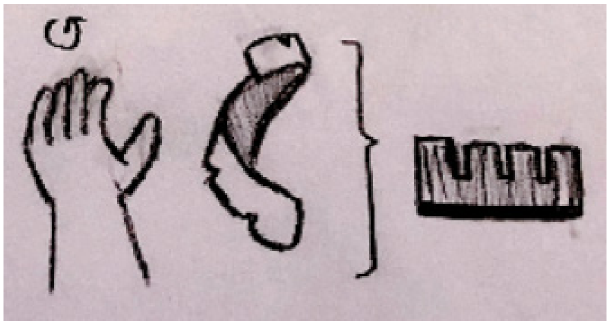


FIG. 10. Supination and pronation (left land). Similar to Figure 9, we use the same layers. By placing the actuator below, the wrist rotates the wrist the opposite way.

The proposed actuators would need a feedback control system to manage, process, and regulate the actions of a device. The feedback controllers take an input, typically through sensors, and regulate and manipulate the output of the system through the use of a microcontroller. This allows the device to constantly adapt to the environment dynamically. Control systems can generally be represented through block diagrams as shown in Figure 11. There are two main control systems that are typically used: setpoint and tracking. Setpoint controllers use feedback to maintain equilibrium, which is at a given setpoint, hence the name. Tracking controllers use manipulator algorithms to define the model of an articulated robot to analyze and perform dynamic and kinematic functions to track the trajectories of the robot. For the proposed device a tracking controller would be more suitable as the actuation process is continuous. Setpoint control would require taking inputs that could be directly compared to a threshold value. Because the robot is soft, it is difficult to directly measure the shape of the actuator. Instead, a tracking controller would be able to function with many unknown parameters of the actuator systems, since all the DOF cannot be controlled in soft actuators. This device would likely need to be modeled as a time-variant; nonlinear system as discussed in the next section.

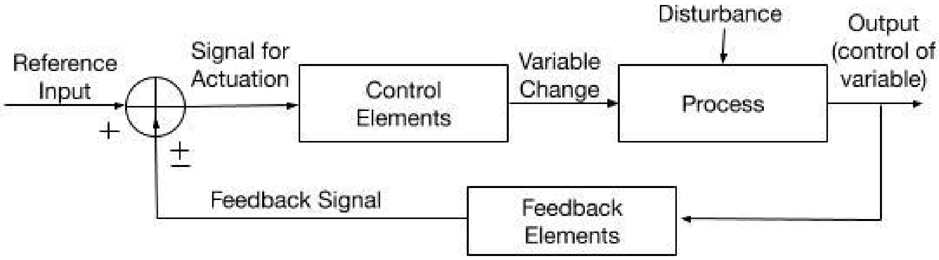


FIG. 11. A generic control system block diagram. Block diagram for the feedback control system for the proposed device prototype.

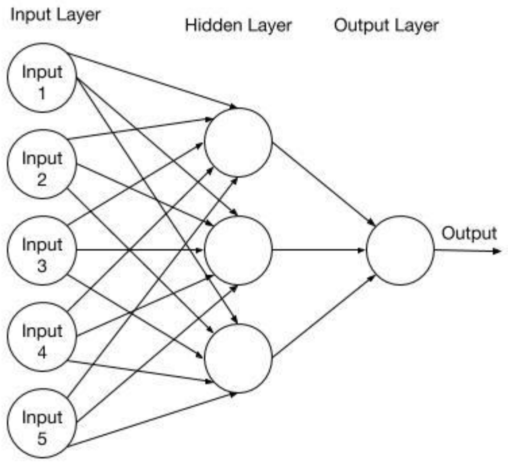
D. Embodied Intelligence

Due to the inherent properties of soft robots, including their lack of rigidity and complex trajectory, soft sensors are often used in control systems. Not every parameter of a system can be directly measured for control, either because the parameter is simply unmeasurable or there is no appropriate apparatus to do so. Soft sensors are used to collect other parameters using other flexible sensors on the device to collect data which are then used to make a prediction of the true parameter. In the proposed device, neural networks could be utilized to count the completed repetitions to track the user's progress and accordingly determine the desired internal air pressure and threshold angle, which would be the target outputs for the control system.

Despite the fact that it is not necessary to regulate every parameter of a soft system, the proposed device requires a certain degree of sensing to control the behavior of the actuators. Because these actuators are soft and flexible, incorporating tactile sensors to measure changes to the system based on external manipulation – in this case pressure – is achieved through the use of stretchable electrical wiring and dynamic transducers that are designed for such devices. This allows pressure data to be directly collected from the system, which can be utilized in artificial neural networks and reservoir computing to control exercise difficulty. To incorporate computation, the device exploits the inherent rich dynamics in the material itself, where the soft body is used as a reservoir, complimented by the nonlinearity of the body. Therefore, the collected pressure data can be used in the input layer of the artificial neural network, eventually used to predict the state of other parameters in real time.

Neural networks work by imitating the intricate connections of neurons in the brain for pattern recognition, in the form of providing outputs. Fundamentally, they function through an organization of layers that are used to process information from an input layer to produce an output neuron (see Figure 12). The input layer has one data point (neuron) for each piece of input, which is then passed through a series of hidden layers, which are responsible for processing the inputs, and transfers information to the output layer. Within the hidden layers, there are a few main things that define the processing (see Figure 13). First, each neuron has a weight, which quantifies the impact of the numerical value on the output. This can be both positive and negative. Conceptually, the weight can be seen as a quantification of how strong the neurons are connected. Second, bias, which is an offset in the network, acts as a constant that is added to the weighted total to modify the output. Then, the weighted sum is inputted into an activation function. One common example is the sigmoid function, because it exists only between 0 and 1, so it can be useful to predict probability. The output is given in the output layer, which has one neuron per output of interest.

The proposed device aims to adjust the rehabilitation process—that is, the range of motion and the amount of provided support—based on how much the user has advanced in their recovery process. The aforementioned sensor techniques will be employed as a means of measuring user performance and device trajectory through tactile pressure sensors. Once the input layer is used to predict an output of the pressure layer, using a middle reservoir to process inputs and predict different outputs in real time, the pressure is varied in the actuators and controlled through constant regulation and prediction of the pressure. The pressure levels are regulated using tracking control, which is discussed in a later section. The pressure would be regulated by controlling solenoid valves that regulate air flow in and out of the actuator, instructed by a microcontroller incorporated in the device. Additionally, by using a fluidic flex sensor to measure the number of times that a user has been able to cross a threshold value, the device is able to quantify how much the user's wrist can bend. As the user becomes comfortable with achieving flexion, extension, and rotation to a certain angle, the support from the actuators (i.e., pressure) would decrease to gradually train the wrist to independently achieve these four motions. To adjust the pressure of these actuators, the microcontroller varies the amount of voltage provided to the pressure regulators on the actuators, allowing for the variation of both the actuator angle and rigidity, allowing the degree to which the wrist is being bent and supported to vary throughout the actuation process. The means of controlling actuator pressure based on predictions from data on completed repetitions and angle magnitude. The actuation methodology proposed for the device is described in the next section.



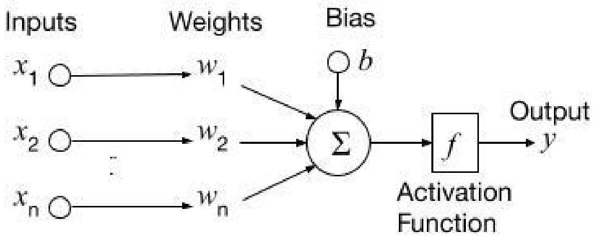


FIG. 12. Neural Network Map. A map of a neural network used to predict the amount of voltage sent to pressure regulators to control the device without manual control.

FIG. 13. Neural network layers map. Outline of the hidden layers that will be in the reservoir of the network. Allows the prediction algorithms to be functional.

IV. Device Modeling

A model of the proposed device was developed based on its features and capabilities. The model allowed for four unique features: (1) allowing the device to move in the four aforementioned directions, achieved by the use of four distinct actuators; (2) intuitive control of the device by using the flex sensor for flexion and extension to sense the position of the hand, and two haptic sensors pressed by the thumb for rotation; (3) control over the difficulty of the exercise in the form of levels, done by regulating the air pressure: the higher the air pressure in the actuator, the more assistance is given to the user's wrist, so by using reservoir computing and prediction algorithms, intelligent adaptation to the user's needs is made possible; and (4) returning the wrist to a normal position after each completed repetition. By reading the flex sensor, the actuators would automatically deflate. This threshold angle value can also be manipulated for sub levels. The device was made of the materials and components listed in Table 1.

TABLE 1. Device components and materials. The components and materials used in the device. Electronic components, jumper wires, and air tubes are omitted.

Component	Material or Product	Quantity
Silicone Elastomer (actuators)	ELASTOSIL® M 4601 A/B	160g
Microcontroller	Arduino Mega 2560 Rev 3	1
Toggle Switches	Nilight 90012E Heavy Duty Rocker Toggle	4
Potentiometers	Rotary Potentiometer - 10k Ohm, Linear	4
Air Pump	Gikfun Mini Air Diaphragm Pump EK1856	1
Valves	SMC VQ110U-5M valve	4
Pressure Sensors	Autex Pressure Sensor 100 Psi	4
Flex Sensor	3" Bidirectional Flexible Bend Sensor	1
Haptics Sensor	GHH Digital Touch Sensor	2
24V DC input jack	Chanzon DC Power Connector	1
24V to 12V converter	Cllena DC/DC Voltage Converter Regulator 24V to 12V	1
24V to 5V converter	DKPLNT 50W DC-DC 12V 24V to 5V 10A Converter	1
10 K Ohm resistors	EDGELEC 10K ohm Resistor	4
Velcro (to hold actuators)	Strenco 2 Inch Adhesive Black Hook and Loop Tape - 5 Yards	1
Wrist Brace	Comfy Brace Night Wrist Sleep Support Brace	1

A. Control System Modeling

The control system for this device is represented in Figure 9, shown below. The system begins with the flex sensor reading, which will vary as the wrist moves and changes its bending angle. This reading is processed by the microcontroller, an Arduino Mega, which processes the input and accordingly controls the solenoid valves and the air pumps. For example, when the wrist bending angle increases, the solenoid valves would open and compressed air would be pumped through the tubes and valves into the corresponding actuator. As a result, the actuator would inflate. When the wrist bending angle is outside a certain threshold, the controller would instruct the air pump to suck the air out from the actuator for deflation. When the valves are closed, pressure in the actuators remains constant. The control occurs in a feedback loop, meaning the actuators are constantly controlled based on the flex sensor reading.

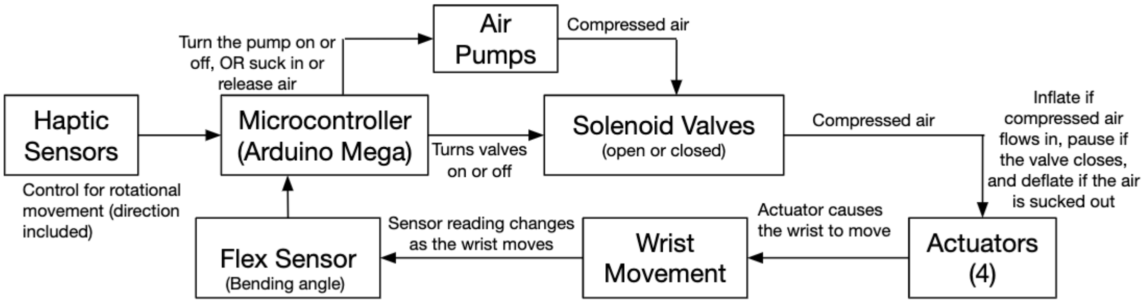

Air Pumps
Compressed air
Turn the pump on or off, OR suck in or release air
Haptic Sensors
Microcontroller (Arduino Mega)
Turns valves on or off
Solenoid Valves
(open or closed)
Compressed air
Inflate if compressed air flows in, pause if the valve closes, and deflate if the air is sucked out
Control for rotational movement (direction included)
Flex Sensor
(Bending angle)
Sensor reading changes as the wrist moves
Wrist Movement
Actuator causes the wrist to move
Actuators (4)

FIG. 14. Device control system block diagram. The control system block diagram, specific to our prototype.

The control system is developed upon an open-source control board developed by Soft Robotics Toolkit [3]. A visual circuit diagram of the configuration is shown in Figure 15 below. Most of the electrical components from Table 1 are represented, as well as the wiring. The connections to the Arduino Mega, a microcontroller developed by Adafruit Industries, are also shown. The solenoid valves are controlled by the Arduino's signals to the Field Effect Transistor (FET) switches, which function by controlling voltage to vary output current. The potentiometers are used to manually control the actuators if needed. The pressure sensors are used to monitor the actuators. Since four actuators are used, there are four sets of valves, valve switches, and FET switches. The 5V power controller is used to power all the other components except the air pump, which uses 12V power input. The sensors are powered by the micro-controller, which uses its native power input.

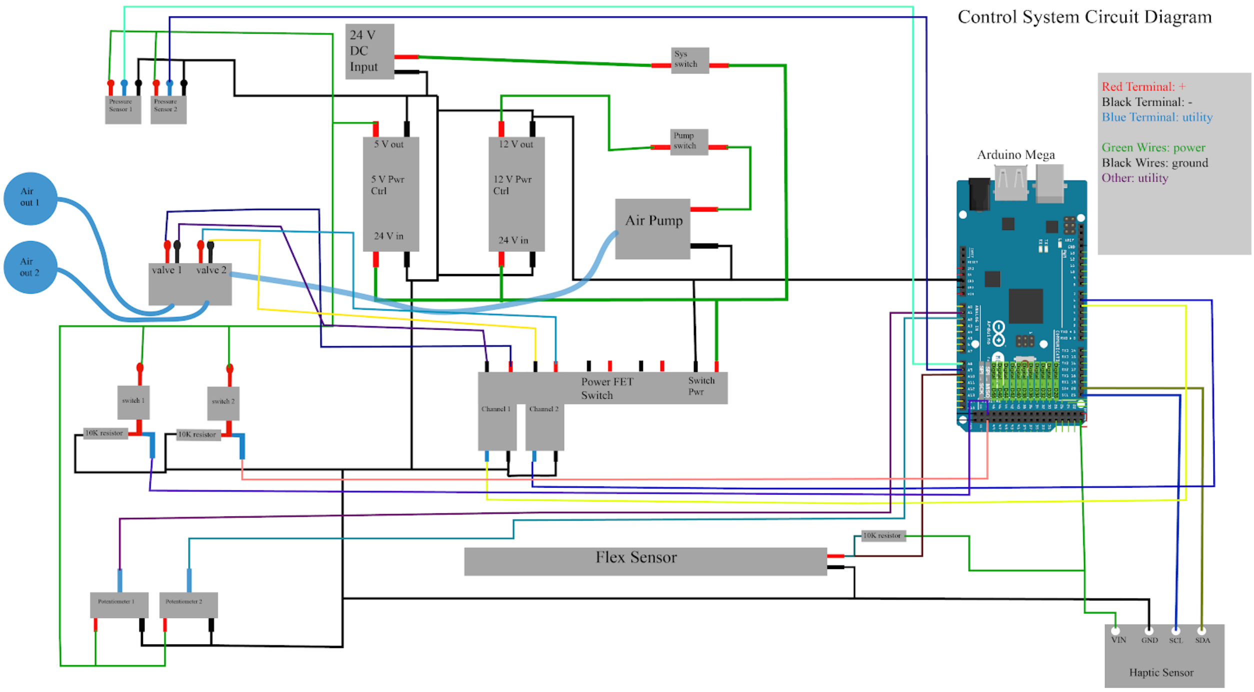


FIG. 15. Visual wiring diagram. The electrical wiring configuration of the control system. Two additional air outlets, valves, switches and potentiometers are omitted to ensure readability.

B. Actuator Design and Modeling

The actuators used in this device are a modified version of the Pneumatic Network Actuator [1] originally developed by Soft Robotics Toolkit [2] and pneu-nets actuator [7]. The actuator dimensions (see Figure 16) are modified to better suit the device. The actuators have air cavities that inflate (see Figure 17). The simulations performed using the model developed and the results are described in the next section.

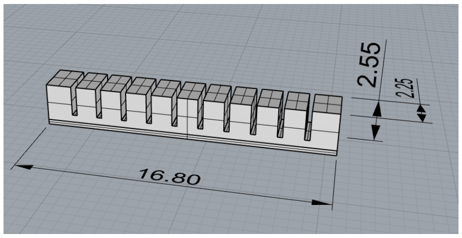


FIG. 16. Actuator CAD model top view. Actuator top and front views with extensible dimensions.

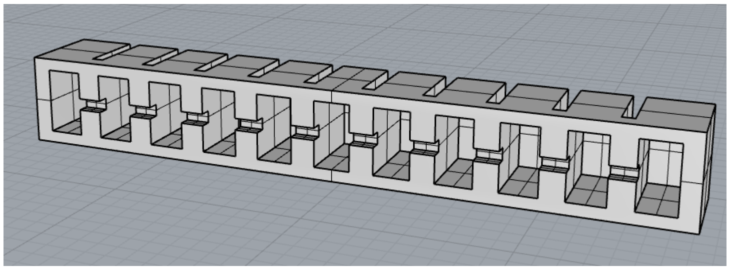


FIG. 17. Actuator CAD model bottom view. The bottom view of the actuator, showing the parts and the inner cavities.

V. Simulations, Results, and Analyses

A. Actuator Inflation Simulation in Simulia Abaqus

When considering the development of the control system, it is important to determine the amount of pressure needed to inflate the actuator. The inflation process was simulated using the Simulia Abaqus software by Dassault Systèmes. The material of the actuator simulated was silicone rubber Elastosil M4601. A thin, 0.10 millimeter thick layer of paper fabric served as the lateral expansion limiting layer. The pressure of the actuators was varied to obtain the bending angle (see Table 2 for results). The time of inflation was set to two seconds. The desired amount of pressure needed per actuator was determined. Because the human wrist only needs to bend to a maximum angle of ninety degrees [4], 10 kPa is a sufficient maximum pressure for an individual actuator (as 10 kPa gives an angle just slightly above ninety degrees). The inflated actuator is shown in Figures 18-21.

TABLE 2. Simulia Abaqus actuator simulation results. Simulation results, showing the relationship between actuator pressure and approximate bending angle.

Pressure (kPa)	Actuator Bending
10 kPa	95.5°
25 kPa	156.6°
40 kPa	216.0°
55 kPa	251.8°

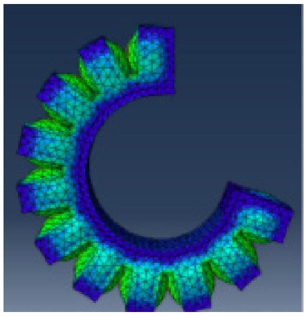


FIG. 18. Actuator at a pressure of 10 kPa. Bending angle was well above 90 degrees, at around 251.8 degrees.

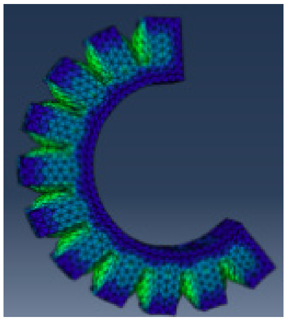


FIG. 19. Actuator at a pressure of 25 kPa. Bending angle was well above 90 degrees, at around 216.0 degrees.

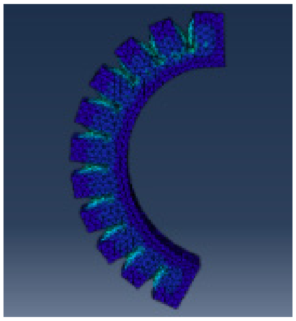


FIG. 20. Actuator at a pressure of 40 kPa. Bending angle was above 90 degrees, at around 156.6 degrees.

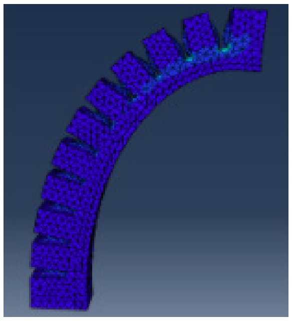


FIG. 21. Actuator at at a pressure of 55 kPa. Bending angle was slightly above 90 degrees, at around 95.5 degrees.

B. Actuator Control Simulation with Arduino in Rhinoceros 3D

To model the movement of the actuators, controlled by biosensory input from the user, a model was created using computer-aided design (CAD) modeling in Rhinoceros 3D (Rhino), a three-dimensional CAD modeling software. First, the complete model was constructed, including the same components from the initial sketch (see Figure 13). The individual models of the four actuators in their proper orientations (i.e., facing upward or downward based on whether they are to be placed on top of or underneath the wrist) were included. The lateral and radial limiting constraints can be seen in the model's dark pink layers (see Figure 22). The red geometries show an outline of the bending motion that occurs with each actuator (see Figures 23-24). To simulate the bending, Grasshopper, an integrated programming environment within Rhino, was used. A parametric arc was created to bend the geometries along, initially working with a slider to manipulate the trajectory of the arc on the XZ plane. The script used for this purpose is presented in Figure 25.

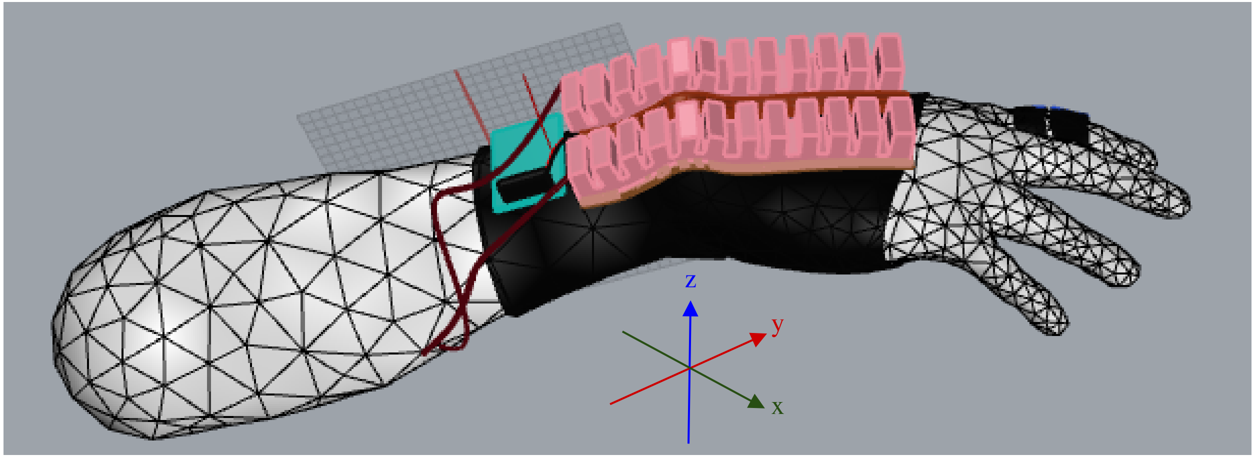


FIG. 22. Full device CAD model from Rhinoceros 3D. The model includes all the components from the initial sketch, including the two top actuators.

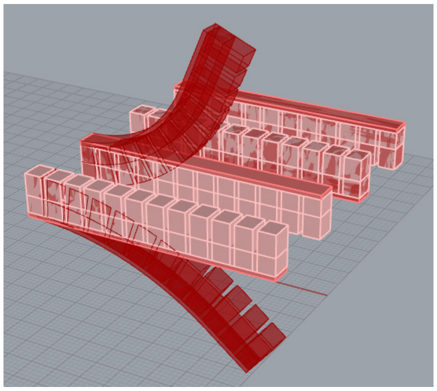


FIG. 23. Flexion and extension. Depicts the movement of the actuators when the tilt sensors detect angle chance.

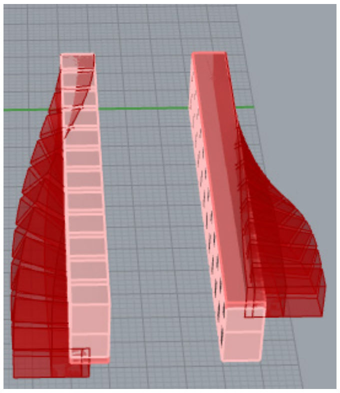


FIG. 24. Rotation for both hands. Depicts the movement when the haptic sensors are touched.

Although the sliders were useful to initially model the actuator movement, the physical device used two biosensors as a catalyst for movement. Therefore, an Arduino Uno board and two sensors were used to control the functionality of the device. There were few modifications made to the design due to the limitations of the availability of our resources. The sensors were modified to replace the flex sensor and two haptic sensors with two tilt sensors with one haptic sensor. The simulation activity was constrained by the number of digital pins available on the Arduino Uno. In the future, this constraint can be lifted by using an Arduino Mega board which has an expanded set of pins.

The electrical configuration (see Figure 26) begins with sensors (see Figures 27) which are connected to the Arduino using a solderless breadboard and jumper wires. This allows current to flow from the board's power and ground pins to the power rails on the breadboard. Each sensor pin is connected to digital input pins on the breadboard. Two tilt sensors were necessary for the two directions of vertical movement, flexion and extension. To combat the lack of another touch sensor, we utilized a toggle to switch between the two rotation directions.

There are a number of software that were used to achieve this simulation. The first was the Arduino Integrated Development Environment (IDE), in which code was written to identify a number of parameters. This included the baud rate, or a quantification of the system's data transmission speed, which was 115200 bits per second in our prototype. In addition, input and output pins were defined in the code. Finally, instructions regarding the processing of the sensor inputs and how to modify or present the output were programmed, using techniques like loops, conditionals, and arrays. This program, called a sketch, was uploaded through a COM (communication) port in the serial communication interface, connected via a USB cable. Sensor readings were eventually utilized as a basis for controlling the geometry constructed in Rhino via Grasshopper. The sensor reading from the Serial Monitor in the Arduino IDE was transferred using an extension software called Firefly. The Firefly Firmata was interfaced with the Arduino sketch (see Figure 28) for carrying out the simulations with ease as the model parameters were defined in the graphical interface, like the baud rate, COM port, and port availability (see Figure 30). To ensure that the reading was constantly updating in the interface, the timer function was added in the Grasshopper environment. Finally, once this reading successfully appeared in Grasshopper, Python scripting (see Figure 29) was used to interpret this result and ultimately control the slider. This was done by defining the output variable, initializing it to the high point of the domain of the arc, and incrementally changing the value of the x coordinate when the sensor reading was true, or “1.” The output of the Python scripting component was fed into the inputs for the x and z values of the arc along which the 3D geometry bends. The results are shown in Figures 31-34).

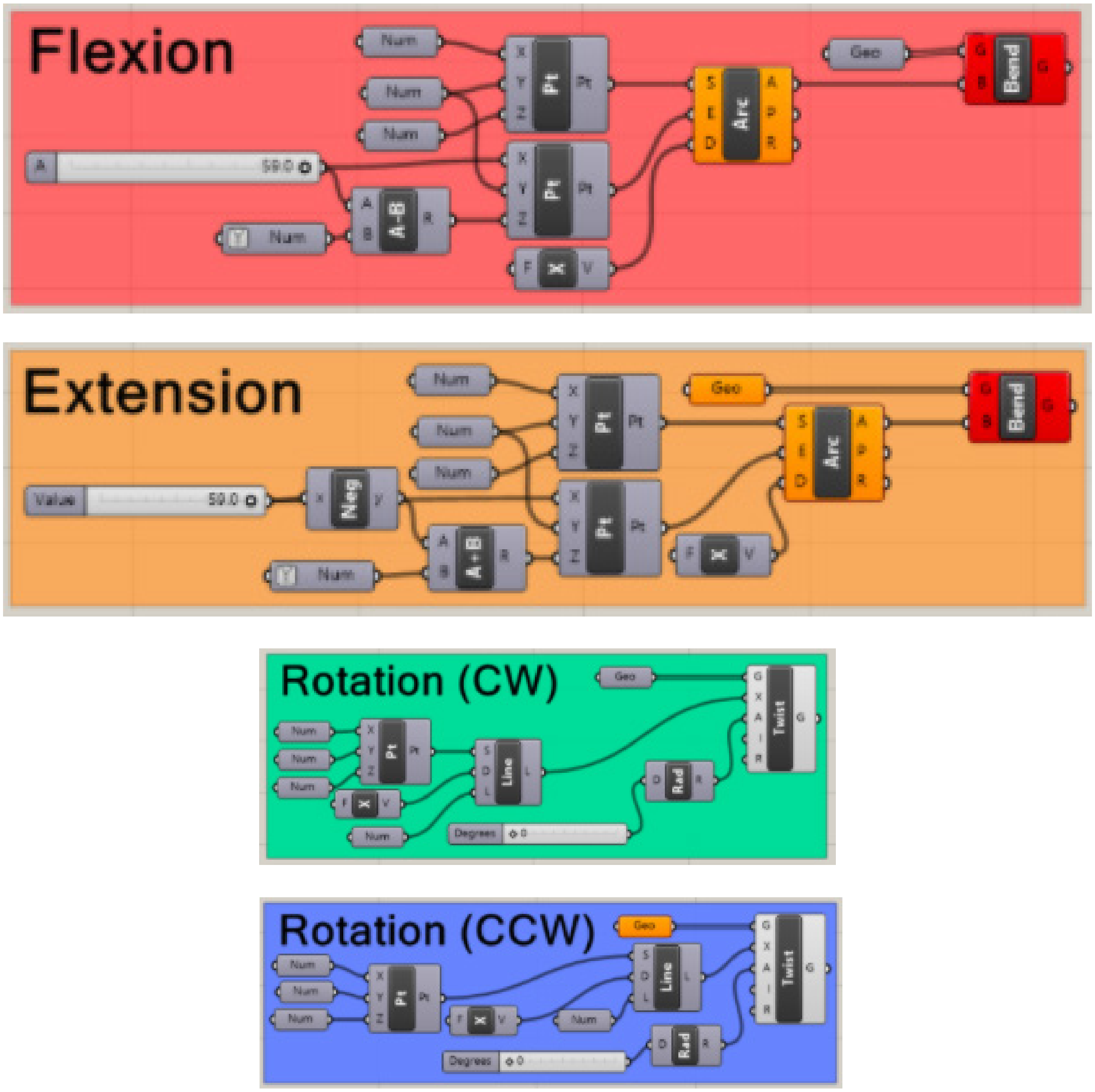


FIG. 25. Grasshopper scripts.

Initial Grasshopper script to control the geometries created in Rhino using sliders, which were later replaced by controlling the actuators using data, shown in the yellow panels (see Figure 20) from the digital sensors, and Python scripting components (see Figure 21).

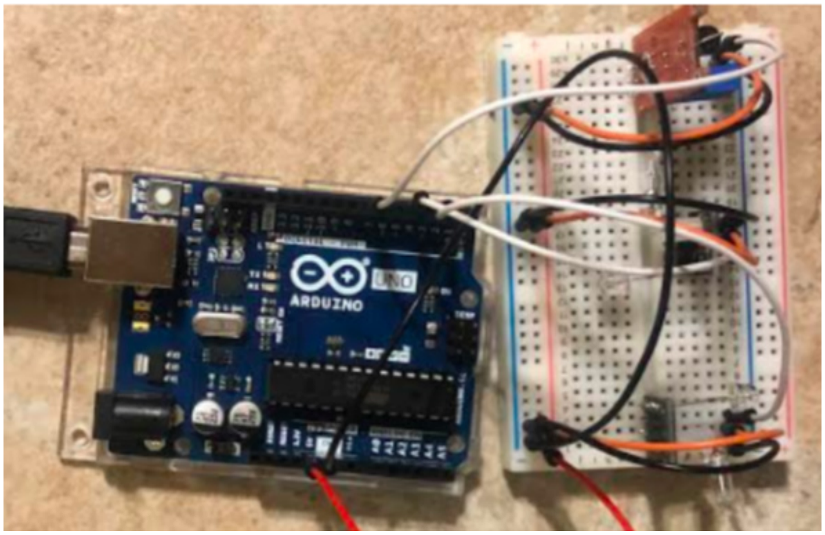


FIG. 26. Electrical configuration. Breadboard configuration of the Arduino board and sensors.

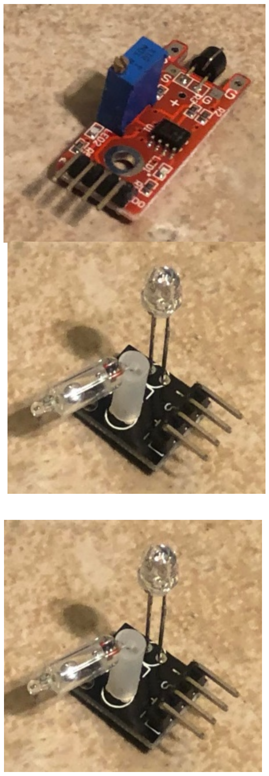
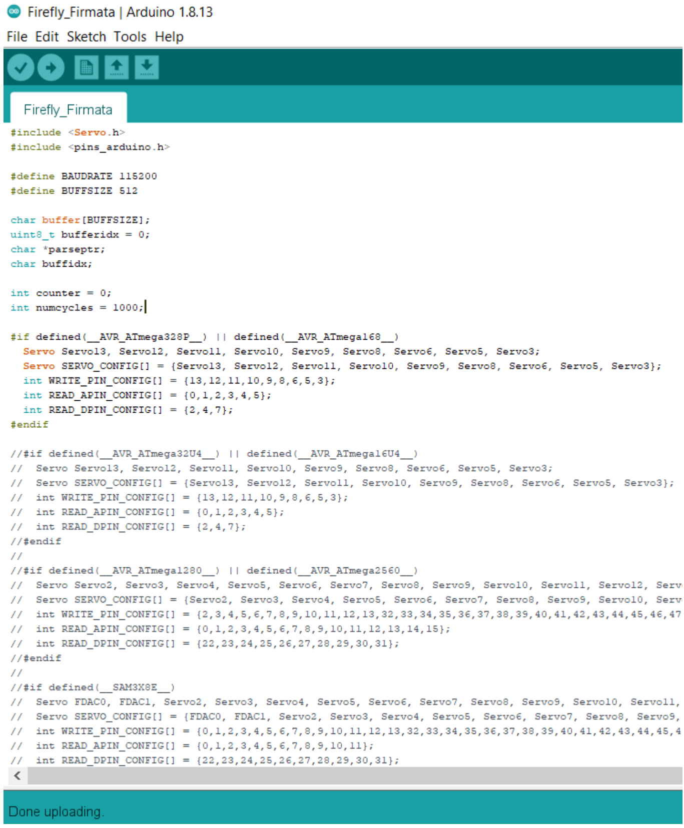
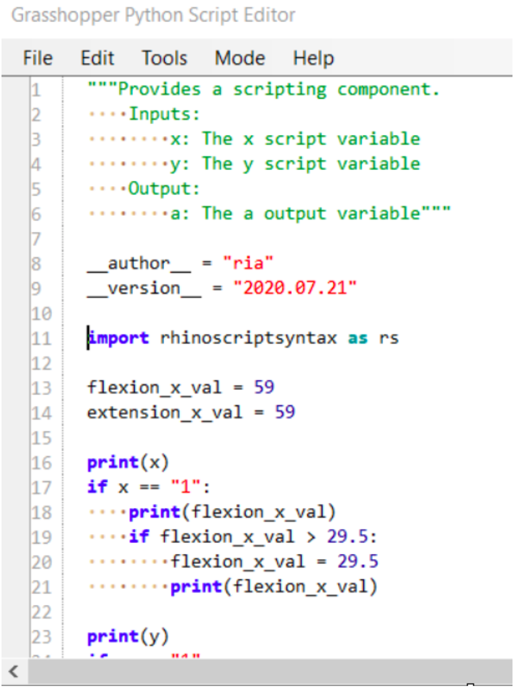


FIG. 27. Tilt and touch sensors used for input.



```
#include <Servo.h>
#include <pins_arduino.h>

#define BAUDRATE 115200
#define BUFFSIZE 512

char buffer[BUFFSIZE];
uint8_t bufferidx = 0;
char *parseptr;
char buffidx;

int counter = 0;
int numcycles = 1000;

#if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__)
  Servo Servo13, Servo12, Servo11, Servo10, Servo9, Servo8, Servo6, Servo5, Servo3;
  Servo SERVO_CONFIG[] = {Servo13, Servo12, Servo11, Servo10, Servo9, Servo8, Servo6, Servo5, Servo3};
  int WRITE_PIN_CONFIG[] = {13,12,11,10,9,8,6,5,3};
  int READ_APIN_CONFIG[] = {0,1,2,3,4,5};
  int READ_DPIN_CONFIG[] = {2,4,7};
#endif

//#if defined(__AVR_ATmega32U4__) || defined(__AVR_ATmega16U4__)
//  Servo Servo13, Servo12, Servo11, Servo10, Servo9, Servo8, Servo6, Servo5, Servo3;
//  Servo SERVO_CONFIG[] = {Servo13, Servo12, Servo11, Servo10, Servo9, Servo8, Servo6, Servo5, Servo3};
//  int WRITE_PIN_CONFIG[] = {13,12,11,10,9,8,6,5,3};
//  int READ_APIN_CONFIG[] = {0,1,2,3,4,5};
//  int READ_DPIN_CONFIG[] = {2,4,7};
//#endif
//
//#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
//  Servo Servo2, Servo3, Servo4, Servo5, Servo6, Servo7, Servo8, Servo9, Servo10, Servo11, Servo12, Serv
//  Servo SERVO_CONFIG[] = {Servo2, Servo3, Servo4, Servo5, Servo6, Servo7, Servo8, Servo9, Servo10, Serv
//  int WRITE_PIN_CONFIG[] = {2,3,4,5,6,7,8,9,10,11,12,13,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47
//  int READ_APIN_CONFIG[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
//  int READ_DPIN_CONFIG[] = {22,23,24,25,26,27,28,29,30,31};
//#endif
//
//#if defined(__SAM3X8E__)
//  Servo FDAC0, FDAC1, Servo2, Servo3, Servo4, Servo5, Servo6, Servo7, Servo8, Servo9, Servo10, Servo11,
//  Servo SERVO_CONFIG[] = {FDAC0, FDAC1, Servo2, Servo3, Servo4, Servo5, Servo6, Servo7, Servo8, Servo9,
//  int WRITE_PIN_CONFIG[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,32,33,34,35,36,37,38,39,40,41,42,43,44,45,4
//  int READ_APIN_CONFIG[] = {0,1,2,3,4,5,6,7,8,9,10,11};
//  int READ_DPIN_CONFIG[] = {22,23,24,25,26,27,28,29,30,31};
```

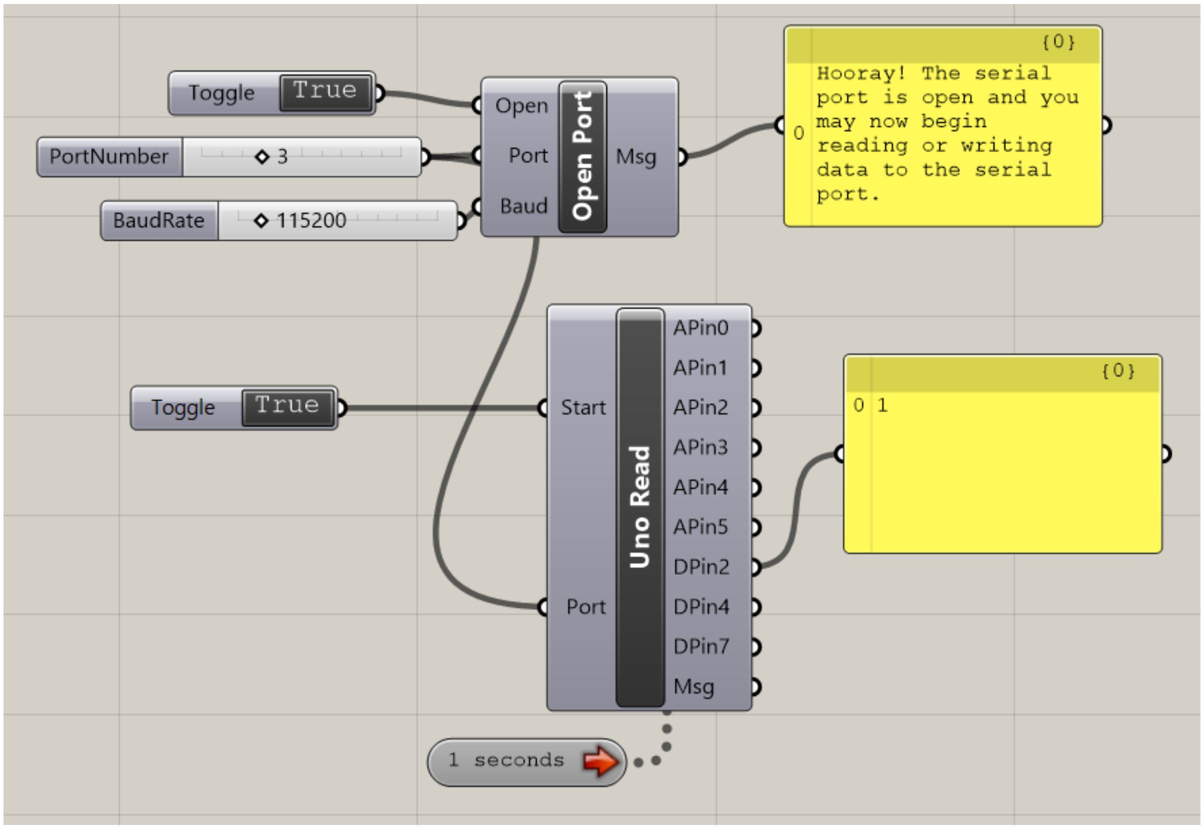
Done uploading.



```
"""Provides a scripting component.
    Inputs:
        x: The x script variable
        y: The y script variable
    Output:
        a: The a output variable"""

__author__ = "ria"
__version__ = "2020.07.21"

import rhinoscriptsyntax as rs

flexion_x_val = 59
extension_x_val = 59

print(x)
if x == "1":
    print(flexion_x_val)
    if flexion_x_val > 29.5:
        flexion_x_val = 29.5
        print(flexion_x_val)

print(y)
```

FIG. 28. Sketch. The Arduino sketch in the IDE.

FIG. 29. Python script. Written in Grasshopper.



```
Hooray! The serial
port is open and you
may now begin
reading or writing
data to the serial
port.
```

FIG. 30. Grasshopper script. Configuration of parametric scripting components to integrate Arduino.

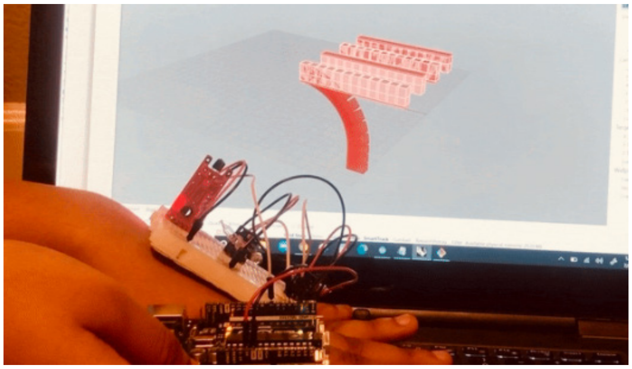


FIG. 31. Flexion. Because the sensor is tilted down, the flexion actuator is pressurized and bends down in Rhino.

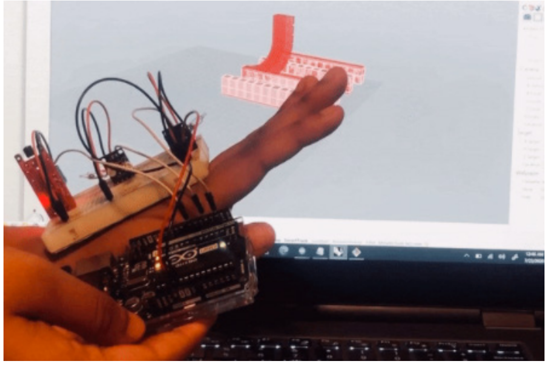


FIG. 32. Extension. Because the sensor is tilted up, the extension actuator is pressurized in Rhino.

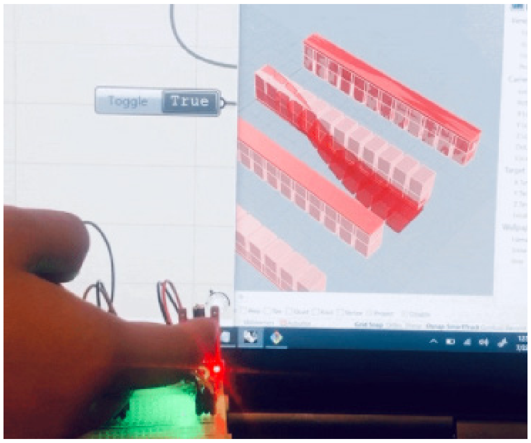


FIG. 33. Supination and pronation (R). Since the toggle is set to true for CW, and the sensor is being touched, the actuator twists for the right hand in Rhino.

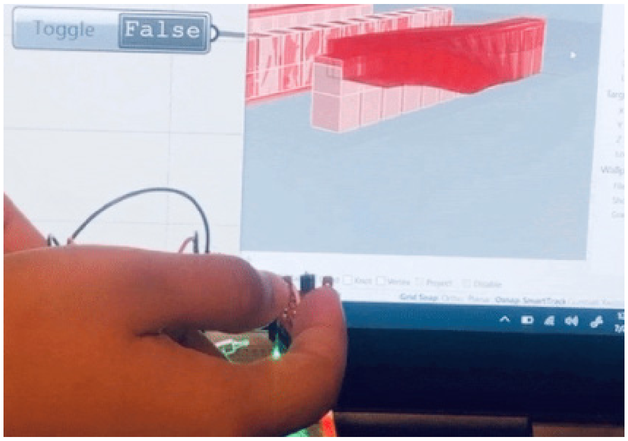


FIG. 34. Supination and pronation (L). Since the toggle is set to false for CCW, and the sensor is touched, the actuator twists for the left land in Rhino.

C. Materials Simulation in Autodesk Fusion 360

To simulate the effectiveness and suitability of various materials, tests were performed in Autodesk Fusion 360. A CAD model was developed of a single inflatable cavity of the actuator and then tested three materials of that cavity using a static force. These materials were nylon (polycaprolactam) 6, rubber silicone, and low density polyethylene (LDP). Overall, these materials were lightweight and had low Young's moduli (see Table 3). The Young's modulus is a ratio of stress to strain (in the elastic zone) , measuring a material's resistance to deformation when a force is placed on it. A material of a low Young's modulus will deform easily, and vice versa [6]. We considered the safety factor and displacement of each material to determine the optimal material for our device.

TABLE 3. The materials used and their respective weights needed for the device and their Young's moduli.

Material	Estimated Total Weight of All Material (g)	Young's Modulus (GPa)
Nylon 6	126.8	2.758
Rubber silicone	141.2	0.003
Low Density Polyethylene	101.2	0.124

The graph shown in Figure 36 shows the resulting safety factor of each material as the load increased on all sides. The safety factor is a ratio of strength to expected strain. The target safety factor ranges based on the uncertainty and risk of the design [13]. Any safety factor above four is generally considered conservative. Fusion 360 offers a maximum safety value reading of 15. So, from the simulation results, it was found that Nylon 6 could withstand up to 2 PSI while keeping a safety factor equal to or greater than 15. LDP and rubber silicone were relatively close in safety factor as the PSI increased and both gradually decreased in safety factor after 0.75 PSI. Through all materials exhibited high safety factors, Nylon 6 was significantly higher than Rubber Silicone and LDP. The graph in Figure 37 shows the materials' displacement increased as the load increased. All the materials showed roughly linear changes in maximum displacement as the PSI increased. Rubber silicone had the steepest slope (and therefore the largest displacement), followed by LDP, followed by Nylon 6. Rubber silicone was the ideal material due to its high deformation even at low pressures and high Young's modulus. Though rubber silicone had the lowest safety factor, it was still well above the threshold value of four. Despite the higher weight of, the difference in weights was negligible,

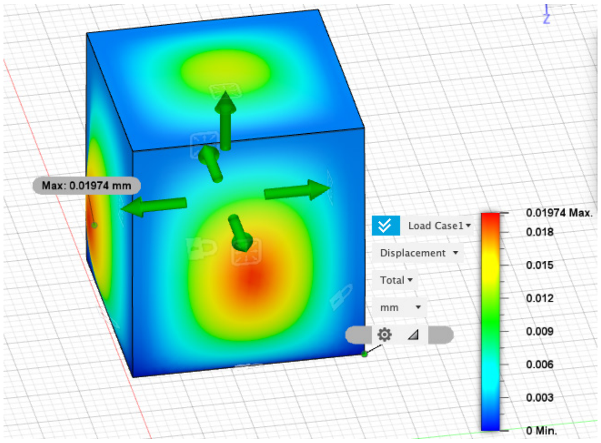


FIG. 35. Single Actuator Cavity. Model of a single cavity from Autodesk Fusion 360.

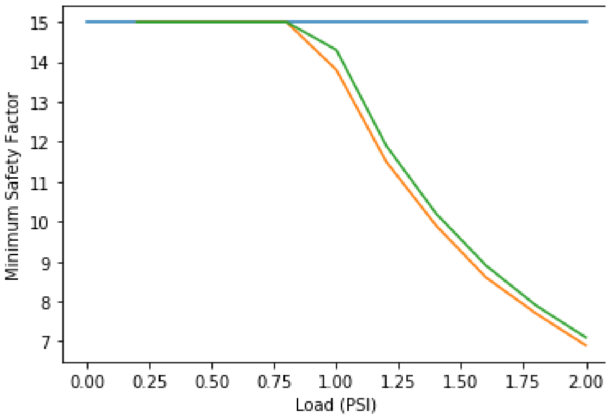


FIG. 36. Graph of safety factor and load.

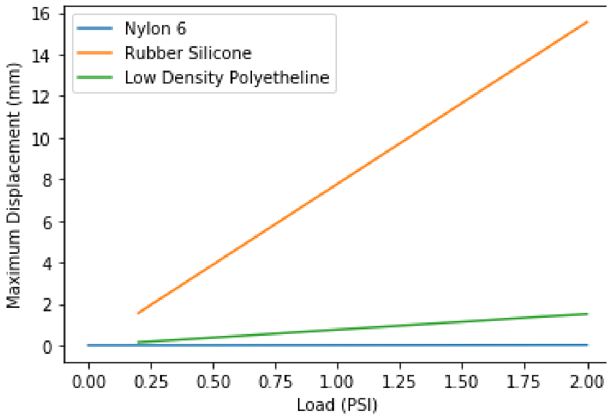


FIG. 37. Graph of displacement and load.

VI. Conclusions

The prototype offered four ranges of motion, intuitive control, and a dynamically adaptable physical therapy process. Through many simulations in Rhinoceros 3D, Grasshopper, Kangaroo, Firefly, Simulia Abaqus, and Fusion 360, the optimal material and pressure necessary in our device were tested. The optimal material was silicone rubber, Elastosil M4601 A/B. In addition, movements in all four directions of wrist motion: extension, flexion, rotation (supination and pronation) were successfully simulated. It was found that only needed 1-2 PSI of pressure was needed to fully extend the wrist to 95.5 degrees, which is sufficient. The maximum capabilities of the device is 251.8 degrees, at around 8 PSI. Because these conclusions are purely theoretical, a possible future work would be to construct a physical prototype and obtain experimental data. Additionally, a mobile application that can collect data from the physical therapy process could be developed. This would involve collecting information from flex and haptic sensors to gauge the user’s performance and develop a custom exercise plan. Finally, the device could be further developed for other limbs using a similar concept.

Acknowledgments

This research was conducted during the Summer Research Academies program at the University of California, Santa Barbara. We would like to thank the following people for their assistance in our research: Ms. Yin Yu (instructor), Mr. Diarmid Flatley (teaching assistant), Dr. Lina Kim (director), and the UCSB Summer Research Academies Staff.

References

1. Mosadegh, Bobak, Panagiotis Polygerinos, Christoph Keplinger, Sophia Wennstedt, Robert F. Shepherd, Unmukt Gupta, Jongmin Shim, Katia Bertoldi, Conor Walsh, and George W.. "Pneumatic Networks for Soft Robotics That Actuate Rapidly." Advanced Functional Materials 24.15 (2014): 2163-170. https://doi.org/10.1002/adfm.201303288

2. "PneuNets Bending Actuators." Soft Robotics Toolkit, https://softroboticstoolkit.com/book/pneunets-bending-actuator. https://softroboticstoolkit.com/book/pneunets-modeling.

3. "Fluidic Control Board." Soft Robotics Toolkit, softroboticstoolkit.com/book/control-board. https://softroboticstoolkit.com/book/control-board.

4. Hersh, Erica. "Wrist Flexion: What's Normal, What's Not and Exercises to Improve It." Healthline, Healthline Media, 24 June 2019, www.healthline.com/health/wrist-flexion.

5. Ogura,, Keiko & Wakimoto, Shuichi & Suzumori, Koichi & Nishioka,, Yasutaka. (2009). Micro Pneumatic Curling Actuator - Nematode Actuator. 462 - 467. 10.1109/ROBIO.2009.4913047. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4913047&isnumber=4912969

6. Didier, P, et al. "Mechanical Stability of Custom-Made Implants: Numerical Study of Anatomical Device and Low Elastic Young's Modulus Alloy." Materials Science & Engineering C, vol. 74, 2017, pp. 399–409. 10.1016/j.msec.2016.12.031

7. Whitesides, George M. "Soft Robotics." Angewandte Chemie International Edition, vol. 57, no. 16, 2018. https://doi.org/10.1002/anie.201800907

8. Cooper, Cynthia. Fundamentals of Hand Therapy : Clinical Reasoning and Treatment Guidelines for Common Diagnoses of the Upper Extremity. Second ed., Mosby, 2013.

9. Thuruthel, Thomas George, et al. "Soft Robot Perception Using Embedded Soft Sensors and Recurrent Neural Networks." Science Robotics, vol. 4, no. 26, 2019, p. eaav1488. 10.1126/scirobotics.aav1488

10. Yazhou Wang, et al. "New Structure of Pneumatic Networks Actuators for Soft Robotics." Journal of Engineering (Stevenage, England), vol. 2019, no. 13, 2018, pp. 273–277. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8707349&isnumber=8707161

11. Andriske, Larissa, et al. "Patient and Therapist Experiences of the SaeboFlex: A Pilot Study." Occupational Therapy International, vol. 2017, no. 1, 2017, p. 6. 10.1155/2017/5462078

12. Musto, Joseph C. “The Safety Factor: Case Studies in Engineering Judgment.” International Journal of Mechanical Engineering Education, vol. 38, no. 4, 2010, pp. 286–296. https://doi.org/10.7227/IJMEE.38.4.2

13. Polygerinos, Panagiotis, et al. “Soft Robotic Glove for Hand Rehabilitation and Task Specific Training.” Harvard School of Engineering and Applied Sciences, N.D. https://biodesign.seas.harvard.edu/files/biodesignlab/files/2015_-_polygerinos_-_icra_-_soft_robotic_glove_for_hand_rehabilitation_and_task_specific_training.pdf
signlab/files/2015_-_soft_robotic_glove_for_hand_rehabilitation_and_task_specific_training.pdf

14. Charfeddine, M. & Jouili, Khalil & Braiek, Naceur. (2015). Approximate input-output feedback linearization of non-minimum phase system using vanishing perturbation theory. doi: 10.4018/978-1-4666-7248-2.ch006.

15. Mohammadi A, Lavranos J, Zhou H, Mutlu R, Alici G, Tan Y, et al. (2020) A practical 3D-printed soft robotic prosthetic hand with multi-articulating capabilities. PLoS ONE 15(5): e0232766. https://doi.org/10.1371/journal.pone.0232766Polygerinos, Panagiotis, et al. “Soft Robotic Glove for Hand

16. Kaneko, Hiromasa, et al. “11th International Symposium on Process Systems Engineering.” *ScienceDirect Computer Aided Chemical Engineering*, 2012, https://www.sciencedirect.com/topics/computer-science/soft-sensor

17. Charfeddine, M. & Jouili, Khalil & Braiek, Naceur. (2015). Approximate input-output feedback linearization of non-minimum phase system using vanishing perturbation theory. 10.4018/978-1-4666-7248-2.ch006.

18. Dubey, K.A., et al. “High Performance Polymer Nanocomposites for Structural Applications.” *ScienceDirect Materials Under Extreme Conditions*, 2017, https://www.sciencedirect.com/topics/engineering/elastomeric-matrix

19. B. Xian, D. M. Dawson, M. S. de Queiroz and J. Chen, "A continuous asymptotic tracking control strategy for uncertain nonlinear systems," in IEEE Transactions on Automatic Control, vol. 49, no. 7, pp. 1206-1211, July 2004, doi: 10.1109/TAC.2004.831148.

20. D. Braganza, D. M. Dawson, I. D. Walker and N. Nath, "A Neural Network Controller for Continuum Robots," in IEEE Transactions on Robotics, vol. 23, no. 6, pp. 1270-1277, Dec. 2007, doi: 10.1109/TRO.2007.906248.

21. H. Al-Fahaam, S. Davis and S. Nefti-Meziani, "Wrist rehabilitation exoskeleton robot based on pneumatic soft actuators," 2016 International Conference for Students on Applied Engineering (ICSAE), Newcastle Upon Tyne, UK, 2016, pp. 491-496, doi: 10.1109/ICSAE.2016.7810241.