ROB-GY 6413

Robots for Disability

Sensorized Hand Rehabilitation Therapy Device

Project Report submitted to Professor Kapila in partial fulfillment of the requirements of the course

Arunagiri Adhithian R - ar7404

Erin Butler - ecb9931

Nithya Muralidaran - nm3671

December 2022

NYU | TANDON SCHOOL OF ENGINEERING

TABLE OF CONTENTS

Abstract
After a stroke, patients are often prescribed a version of hand rehabilitation in which a ball is squeezed in order to build strength of the weakened hand. While some progress may be made in randomly squeezing the ball, it is hard to measure progress over the course of rehabilitation when using devices without feedback. Additionally, when patients are not able to gauge how much they should be squeezing, they often do not progress as quickly as expected through the rehabilitation process, leading to abandonment of the exercises. This project develops a prototype of a new hand rehabilitation therapy device equipped with pressure sensors and LEDs to provide feedback throughout the course of rehabilitation exercises.

Objective
The objective for this project is to develop a first prototype of a sensorized rehabilitation therapy device able to provide patients with visual feedback during hand rehabilitation exercises in order to improve strength quicker and decrease rate of abandonment of therapy devices.

Background
A stroke occurs when blood flow to a certain part of the brain is restricted. Depending on the location and intensity of the stroke, its effects will vary. Some common side effects post-stroke include loss of motor function or paralysis of one side of the body, speech and language problems, changes in behavior, and memory loss [1]. The project intends to address loss of motor ability and strength in the arm and hand. It has been shown that intense motor therapy post-stroke can improve motor ability in the long run and is most effective when this therapy occurs as close to the time of stroke as possible [2].

There are a variety of motor rehabilitation approaches for stroke patients including one-on-one time with a physical or occupational therapist and supervised robotic assistive rehabilitation [3]. These forms of rehabilitation are time consuming for the therapist. Patients are often prescribed resistance based therapy exercises for between sessions and provided with a resistance ball [4]. Based on the patient's ability, the level of resistance in the ball can be changed over time to encourage further development. Resistance balls and bands increase access to therapy when one-on-one time with a doctor is limited for any reason [5]. These resistive devices are used by many elderly patients, not just those who have suffered from stroke.

Depending on the level of injury and ability of the patient, a variety of different rehabilitation exercises can be prescribed [6]. This is intended to work different muscle groups in the hand and arm and is an attempt to provide task specific rehabilitation.

Figure 1: Different hand therapy exercises.

As with many forms of assistive devices, rehabilitation devices and exercises are often met with abandonment [7] . In the case of resistive therapy balls and resistive bands, this abandonment can be caused by a lack of interest in a repetitive task as well as not seeing improvement. There is no feedback for the patient with these devices telling the patient they have reached the desired compression. This project intends to address this issue.

Prototype Hardware

A goal of this project is to keep the device as low profile as possible so it will remain a mobile device similar to a resistance ball. Additionally, the device is intended to be ambidextrous and can be used by numerous patients, regardless of strength or size of hand. The device created has a 3D printed cylindrical core fitted with 5 finger-tip sized pressure sensors mounted between layers of foam. The foam provides a soft, malleable surface that deforms as the patient applies pressure when completing their exercises.

Figure 2: Cylindrical core of sensorized therapy device.

Of the pressure sensors tested, the Walfront Thin Film Pressure Sensor had the best results. When placed on a hard surface, the sensor has a 2kg capacity when pressed directly. The sensors are connected to the analog pins of an Arduino Uno, which provides a voltage value corresponding to the level of compression. In the proposed design, rather than placing the sensors directly on the 3D printed core, they are placed between layers of foam. This increases the amount of pressure needed to max out the sensor, providing an increased range of measurable levels of applied pressure during rehabilitation. The device is equipped with 5 of these sensors: one intended for the thumb or palm depending on the exercise and one for each of the remaining fingertips.

Figure 3: Walfront Thin Film Pressure Sensor

The device is also fitted with 5 LEDs embedded in the foam covering the device. Each LED is associated with one of the pressure sensors and is illuminated when the threshold for that pressure sensor is reached. The LEDs are also used to help the patients walk through the calibration phase on startup and each time a new hand is using the device for rehabilitation. Each LED has a color coded marker painted on the foam indicating where the patient should place their finger in order to get accurate pressure readings. However, since the device is calibrated for each hand and exercise, as long as there is minimal movement of finger placement during the exercises, the device will work as intended.

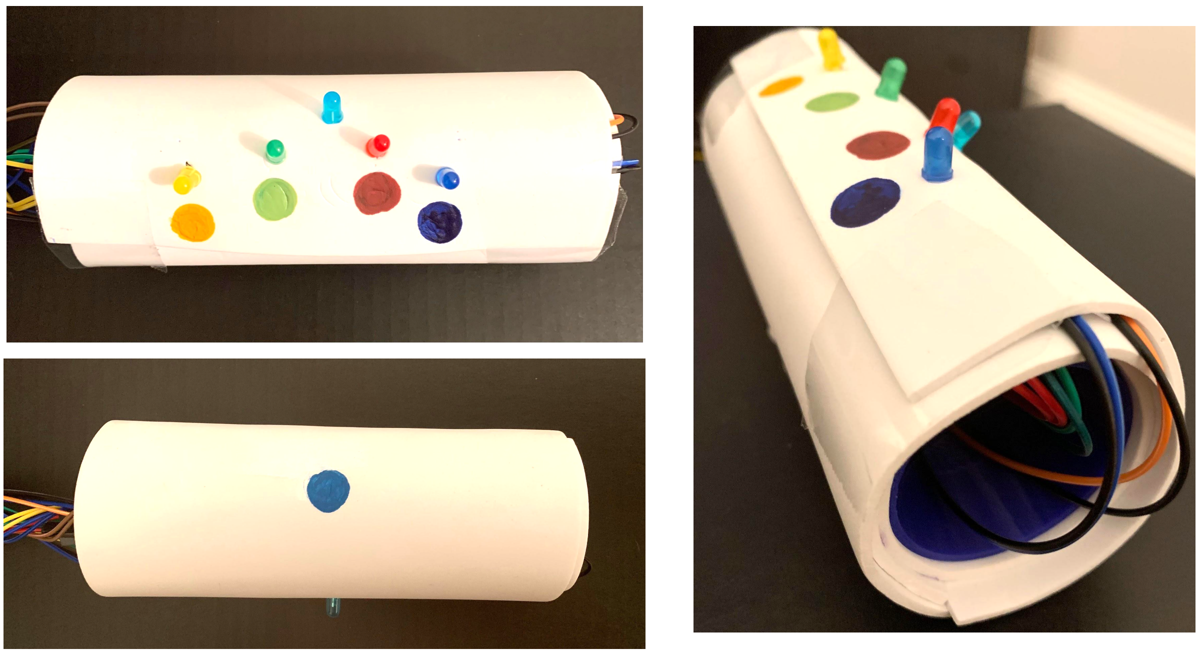

Figure 4: Completed device prototype: Finger tip location, thumb/palm position, side view of 3D printed cylinder

Code Flow

On startup or when the device is reset between change of hands or exercises, the device enters calibration mode. The LEDs will flash to indicate the start and end of calibration. The calibration function determines the minimum and maximum pressure the patient applies. While the green LED is lit, the patient is directed to hold the device loosely for five seconds with their fingers placed over the markers. The minimum pressure value detected for each sensor is stored. Then, the red LED is lit while the patient is directed to hold the device tightly for five seconds. In a similar manner, the maximum value detected for each sensor is stored.

When the device is given to the patient, a goal percentage of maximum pressure is assigned based on patient ability and at the discretion of the therapist. It is common practice for patients to be instructed to apply a certain percentage of maximum pressure for each repetition during their exercises. With this in mind, a threshold value is found for each sensor as the goal percentage of each maximum value recorded. The LEDs will flash again and the device will reach rehabilitation mode.

The device runs in an infinite loop during rehabilitation mode. The pressure value of each sensor is read and compared to its corresponding threshold value as the patient is instructed to repetitively squeeze the device. As the patient squeezes the device, if the pressure value of any sensor rises above the threshold, the corresponding LED will be illuminated. As the patient releases pressure, the LEDs will turn off as the pressure recorded falls below the threshold value.

Having each pressure sensor calibrated individually to find a threshold value is important for longevity of the device. Wear and tear on the device is expected with increased use which will lead to reduced sensitivity of the pressure sensors. Calibrating with each new use will negate the effects of this wear and tear.

Circuit Diagram

The following circuit diagram shows how the hardware of the device was connected to the Arduino Uno. A 9V battery is used to power the device. The pressure sensors are connected to power through the 3V power source on the board. Five analog pins of the Arduino are used to measure the pressure of each sensor. The LEDs are powered through the Arduino's digital pins. A common ground from the Arduino connects the sensors and LEDs.

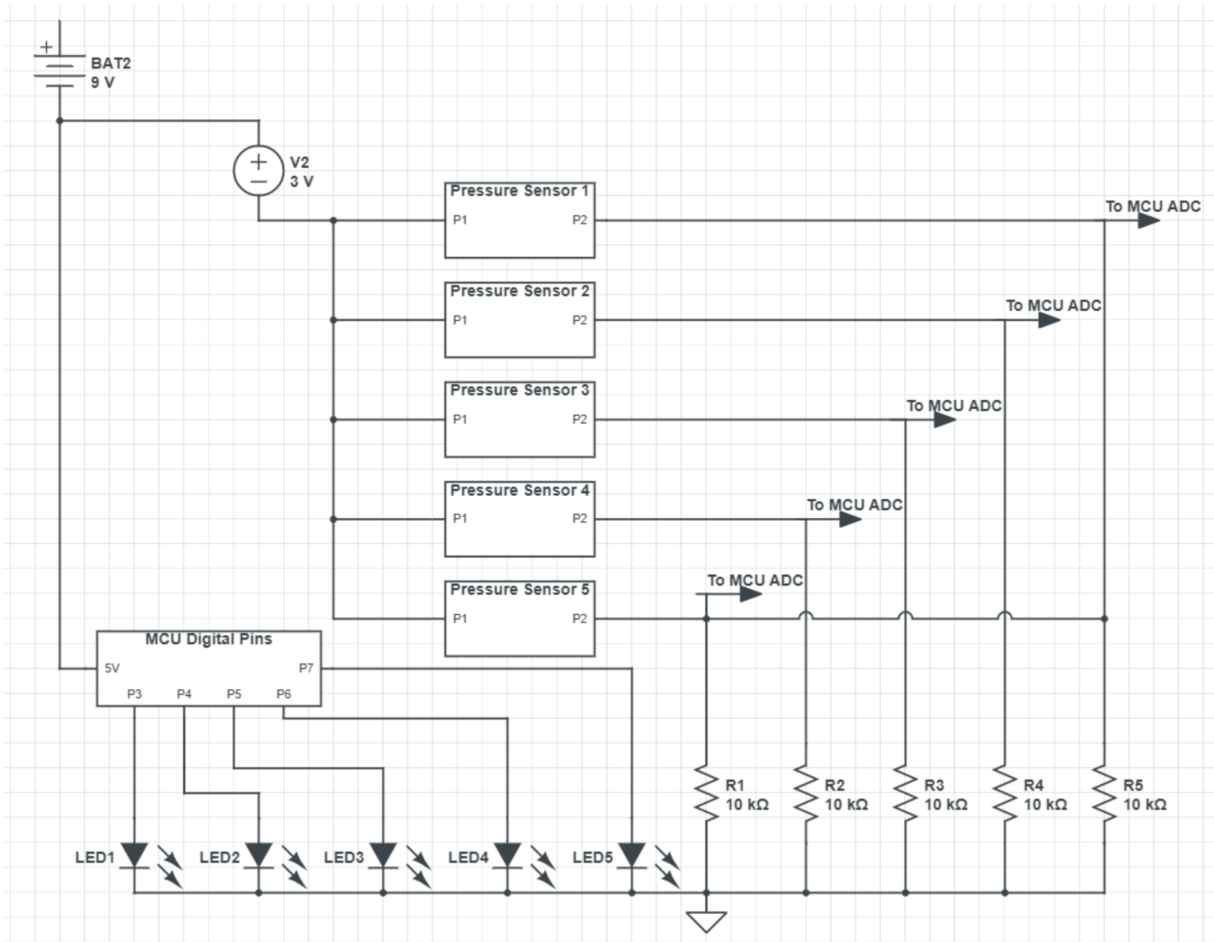

Figure 5: Circuit diagram used to connect sensors and LEDs to an Arduino Uno.

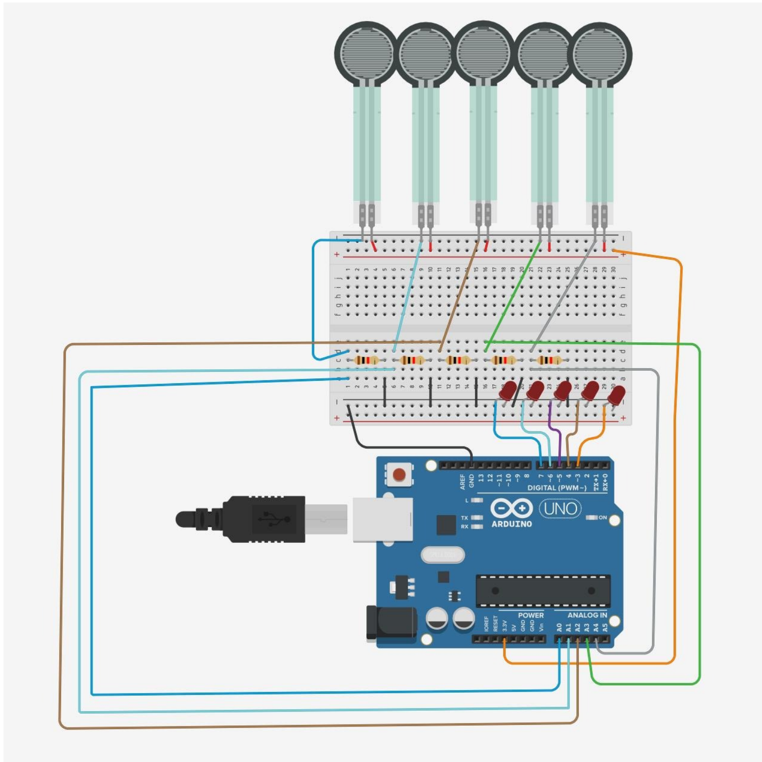

Figure 6: Arduino Uno connections

Results

During testing with able bodied users, a goal percentage of 80 percent was found to provide a good threshold. A significant amount of pressure was required to light each of the LEDs in a variety of hand holds. First, a traditional “egg grip” (see Figure 1) was tested with four fingertips and the palm applying pressure to the sensors. Then, a “grab” (see Figure 1) was tested. For stroke patients, especially those in early stages of rehabilitation, these are the two most common grips used in rehabilitation exercises. It is assumed that a goal percentage of 50 to 80 percent would also be a good fit for stroke patients as their maximum values would be significantly less than that of an able bodied person.

Otherwise, as seen in the accompanying video, the device works as expected. As the user increases and decreases pressure on all markers, the corresponding LEDs light up. As pressure is released, the LEDs turn off. The video also shows each of the sensors being pressed individually to showcase the LEDs lighting respectively.

Impact

The inspiration for this device was to address a need in the motor rehabilitation of stroke patients. By providing visual feedback of reaching a threshold, the hope is patients will be more inclined to continue using the device for their exercises. The device can also be provided to others who would benefit from similar rehabilitation. For example, athletes who have suffered a wrist or hand injury, people who have undergone any form of surgery on the arm, or children with delayed muscle development could benefit from using this device. The advantages of the device being calibrated for each individual are evident when considering these various use cases.

While developing the device, a few areas of improvement were identified, however due to various constraints could not be included in the first prototype. The first improvement would be to reduce the overall footprint of the device and eliminate the wires connecting the handheld part of the device to the Arduino Uno. Using a smaller microcontroller, either of the shelf or custom, and incorporating a custom printed circuit board would be beneficial. The goal for achieving this improvement would be to house the microcontroller and any other hardware inside the handheld part of the device. This would also serve to improve the safety of the device and make it less prone to error by means of wires becoming detached.

Multiple software improvements could also be made. Including a way to track progress between different sessions, such as saving information about maximum pressure applied to each sensor could be beneficial to the therapist in determining how the patient is progressing through therapy. Once the patient has become familiar with the device, different modes of rehabilitation could also be developed. A mode where the goal is to hold each sensor lit for five seconds would help in improving stamina. Using the LEDs to indicate which sensor should be pressed without changing the hold on the device would help the patient improve fine motor control of individual fingers. Games to see how many times a specific goal can be achieved in a certain amount of time could also be developed.

The various improvements are aimed at improving interaction with the device and preventing patients from abandoning their rehabilitation. Providing stimulating activities may also prove beneficial to the cognitive ability of some patients as well as making the device useful and more accessible to a wider range of people. By producing a more engaging device, increased rehabilitation progress in the months immediately after stroke or other injury will lead to improved motor function in the long run.

Code

The following code was written in the Arduino IDE and uploaded to the Arduino Uno connected to the rehabilitation device.

```
/*
 * Erin Butler, Nithya Muralidaran, Arunagiri Ravichandran
 * Robots for Disability - ROBGY 6413
 * Final Prototype Project
 * Sensorized Therapy "Ball" for Improved Rehabilitation Feedback
 * Professor Kapilla
```

```
 * Fall 2022
 */

// Initialize variable for how many pressure sensor pins there are.
#define count 5

// Initialize array of sensor pins.
byte Pins[count] = {A0, A1, A2, A3, A4};
// Initialize array of LEDs
int LEDs[count] = {3, 4, 5, 6, 7};

//Initialize calibration values.
int minimum[count];
int maximum[count];
float goal = .8; // Goal pressure percentage as a decimal.
int threshold[count];

// Initialize an array for storing the voltages read from the sensor pins.
int voltages[count];

// Initialize functions.
void calibrateMode();

void setup() {
  Serial.begin(9600); // For debug

  // Set the analog pins to input. Set the LED pins to output.
  for (int i = 0; i < count; i++){
    pinMode(Pins[i], INPUT);
    pinMode(LEDs[i], OUTPUT);
  }
}

void loop() {
  /*
   * Run calibration to get minimum and maximum values. Calibration
   * needs to happen every time the device is turned on, especially
   * when device is switched between hands. Calibration will also
   * determine threshold values for each sensor.
   */
```

```
    Serial.println("Running calibration."); // For debug.
    calibrateMode();
    delay(2000);
    Serial.println("MIN   MAX   THRESHOLD"); // For debug.
    Serial.print(minimum[0]); // For debug.
    Serial.print("\t"); // For debug.
    Serial.print(maximum[0]); // For debug.
    Serial.print("\t"); // For debug.
    Serial.println(threshold[0]); // For debug.

    /*
     *  The following while loop is run infinitely until therapy is
     *  finished, the device is switched to a new hand, or a new
     *  exercise is to be completed.
     */
    while(1){
      Serial.println("Begin therapy."); // For debug.
      while(1){
        // Read the voltage value of each of the pressure sensors.
        for (int i = 0; i < count; i++) {
          voltages[i] = analogRead(Pins[i]);
        }
        Serial.print(voltages[0]); // For debug.
        Serial.print("\t"); // For debug.
        Serial.println(threshold[0]); // For debug.

        // Turn on LED if corresponding sensor has reached its threshold.
        for (int i = 0; i < count; i++){
          if (voltages[i] > threshold[i]) {
            digitalWrite(LEDs[i], HIGH);
          }
          // Turn LED off if measured value falls below the threshold.
          else {
            digitalWrite(LEDs[i], LOW);
          }
        }
        delay(80);
      }
    }
  }
```

```
void calibrateMode(){
  // Reset the minimum and maximum array values.
  for (int i = 0; i < count; i++){
    minimum[i] = 1000;
    maximum[i] = 0;
  }

  // Flash all LEDs 5 times to indicate entering calibration mode.
  for (int i = 0; i < 5; i++){
    for (int j = 0; j < count; j++){
      digitalWrite(LEDs[j], HIGH);
    }
    delay(300);
    for (int j = 0; j < count; j++){
      digitalWrite(LEDs[j], LOW);
    }
    delay(300);
  }

  // Instruct user to hold the device loosely.
  Serial.println("Hold the device loosely.");
  // Turn the green LED on to indicate to the user to hold loosely.
  digitalWrite(LEDs[2], HIGH);
  delay(1000);
  // For each pressure sensor, find the minimum value detected.
  int t = millis();
  while((millis() - t) < 5000){
    for (int i = 0; i < count; i++) {
      voltages[i] = analogRead(Pins[i]);
      if (voltages[i] < minimum[i]){
        minimum[i] = voltages[i];
      }
    }
  }
  digitalWrite(LEDs[2], LOW);

  // Instruct user to hold the device tightly.
  Serial.println("Hold the device tightly.");
  // Turn the red LED on to indicate to the user to hold tightly.
```

```
    digitalWrite(LEDs[1], HIGH);
    delay(1000);
    // For each pressure sensor, find the maximum value detected.
    t = millis();
    while((millis() - t) < 5000){
      for (int i = 0; i < count; i++) {
        voltages[i] = analogRead(Pins[i]);
        if (voltages[i] > maximum[i]){
          maximum[i] = voltages[i];
        }
      }
    }
    digitalWrite(LEDs[1], LOW);

    /*
     *  Based on the goal percentage set above, determine threshold values
     *  for each sensor.
     */
    float temp[count];
    for (int i = 0; i < count; i++){
      temp[i] = (float (maximum[i] - minimum[i])) * goal;
      threshold[i] = minimum[i] + int (temp[i]);
    }

    // Flash all LEDs 5 times to indicate exiting calibration mode.
    for (int i = 0; i < 5; i++){
      for (int j = 0; j < count; j++){
        digitalWrite(LEDs[j], HIGH);
      }
      delay(300);
      for (int j = 0; j < count; j++){
        digitalWrite(LEDs[j], LOW);
      }
      delay(300);
    }
  }
```

References

[1] “Effects of Stroke.” *Www.stroke.org*, https://www.stroke.org/en/about-stroke/effects-of-stroke.

[2] Author links open overlay panelJudith DSchaechterPersonEnvelope, et al. “Motor Rehabilitation and Brain Plasticity after Hemiparetic Stroke.” *Progress in Neurobiology*, Pergamon, 2 June 2004, https://www.sciencedirect.com/science/article/pii/S0301008204000565?via%3Dihub.

[3] Iosa, M., et al. “Seven Capital Devices for the Future of Stroke Rehabilitation.” *Stroke Research and Treatment*, Hindawi, 13 Dec. 2012, https://www.hindawi.com/journals/srt/2012/187965/.

[4] *Resistance Training in Stroke Rehabilitation ... - Sage Journals*. https://journals.sagepub.com/doi/abs/10.1177/0269215520932964?journalCode=crea.

[5] Kwak CJ, Kim YL, Lee SM. Effects of elastic-band resistance exercise on balance, mobility and gait function, flexibility and fall efficacy in elderly people. J Phys Ther Sci. 2016 Nov;28(11):3189-3196. doi: 10.1589/jpts.28.3189. Epub 2016 Nov 29. PMID: 27942147; PMCID: PMC5140827.

[6] *Hand Exercises for Stroke Patients of All Ability Levels - Flint Rehab*. https://www.flintrehab.com/hand-exercises-for-stroke-patients/.

[7] Sugawara AT, Ramos VD, Alfieri FM, Battistella LR. Abandonment of assistive products: assessing abandonment levels and factors that impact on it. Disabil Rehabil Assist Technol. 2018 Oct;13(7):716-723. doi: 10.1080/17483107.2018.1425748. Epub 2018 Jan 15. PMID: 29334475.