Design and Validation of the Automated Vestibular Evoked Tapper of Ocular Reflex (AVETOR)

By

SHREYANK B DESAI

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Biomedical Engineering

Written under the direction of

William Craelius

And approved by

____

____

____

New Brunswick, New Jersey

May 2018

ABSTRACT OF THE THESIS

Design and Validation of the Automated Vestibular Evoked Tapper of Ocular Reflex (AVETOR)

by SHREYANK DESAI

Thesis Director:

William Craelius

This thesis describes the design, fabrication, and testing of the AVETOR prototype, a device for evaluating vestibular function in relationship to balance. Current diagnostic methods of vestibular function tend to be uncomfortable, lack specificity in diagnosing the problem, and yield inconsistent results. AVETOR takes advantage of the fact that ocular muscles respond to vestibular activity, and hence can be a measure of the same. The AVETOR prototype consists of an automated, mechanical striker to stimulate the vestibular system, sensors to detect muscle activity, and a processor to record and save the test results. The device uses an Arduino Pro Mini/Dynabrush board capable of controlling the striker and processing the signals from the sensors. Protocols for testing were programmed via the Arduino IDE. The AVETOR prototype was tested on three subjects with and without the application of a 2 Hz tapping stimulus at the Fz location of the forehead over a period of 8 seconds, as well as with and without purposeful eye movement. The results indicate that AVETOR meets the required specifications relating to striker control (frequency, duration, and threshold conditions) and data capture (sensor recording and file management). FMG was used as a novel modality to detect VOR responses. The FMG response of the inferior oblique muscles detected a consistent response during purposeful eye movement, but little to no response during the tap stimulation. However, as this is an untried method additional trials are needed. Further improvement on the data acquisition methodology, use of sensors, and studies on the striker's ability to induce VEMPs are recommended, but out of the scope of this thesis.

DEDICATION:

To my sister, for always believing in me and pushing me to be better.

ACKNOWLEDGEMENTS:

I would like to offer my special thanks to the those who have assisted me on this project. Thank you to Dr. William Craelius for guiding me and providing advice to help me accomplish my goals. Thank you to Dr. Ashley Wackym, without whom this project could not have taken place. Thank you to my committee members, Dr. Mark Pierce and Dr. Gary Drzewiecki, for their invaluable input. As well as to the volunteers who assisted me in testing my device, they are greatly appreciated. And lastly, to the Senior Design team (Tyler Agnew, Mahruza Chaudhary, David Ivashenko, Gautham Pavuluri, and Apoorva Sinha), who made working in the lab an enjoyable experience.

TABLE OF CONTENTS:

INTRODUCTION:

The Clinical Problem

Superior Canal Dehiscence, Meniere's disease, and Benign Paroxysmal Positional Vertigo are only three of the many vestibular-related balance disorders that plague 35% of Americans over the age of 40 each year (Agrawal, Ward, & Minor, 2013). Their main symptoms include dizziness, vertigo, and imbalance, but can also lead to vision, hearing, and cognitive difficulties. Diseases of the vestibular systems can affect the ability of an individual to complete normal, everyday tasks leading to a decline in the quality of life (Watson et al., 2018; Zalewski, 2015). Understanding how the vestibular organ responds to stimuli will better help us diagnose any abnormalities or diseases of the vestibular organs (Fetter, 2007; Zalewski, 2015).

Proper treatment requires accurate diagnoses, however, current diagnostic methods of vestibular function tend to be uncomfortable for patients, as well as stimulate multiple components of the vestibular system at once causing a lack of specificity in determining the element causing the problem. Results of these tests also have high variability between each patient making it difficult to discern a normal response from one that is abnormal (Noohi et al., 2017). A relatively new technique exploits vestibular evoked myogenic potentials or VEMP, which has shown promise as a diagnostic tool for vestibular disorders from studies in small animals. External stimulation of the vestibular system elicits VEMPs in the muscles surrounding the neck (cVEMP) and eyes (oVEMP) that are specific to saccular and utricular function, respectively. Studies in eliciting VEMPs in guinea pigs and cats have shown muscles responses similar to those found in humans (Curthoys et al., 2012). These muscle responses can be analyzed to determine any abnormalities specific to the otoliths and their corresponding nerves. This thesis describes the design, fabrication, and testing of the AVETOR (Automated Vestibular Evoked Tapper of Ocular Reflex) prototype, a device for testing vestibular responses. The end goal is to develop a device that is portable, cost effective, and comfortable for the patient.

Human Balance Mechanism

The three components that influence balance via sensory input are vision, proprioception, and the vestibular organs. Firstly, the eyes provide visual cues about our surroundings. Secondly, proprioceptors located in the muscles, joints, and skin that are sensitive to pressure and stretch changes, tell the body how it is oriented relative to itself. Lastly, the vestibular system gathers information about the body’s motion, equilibrium, and spatial orientation letting the brain know how the body itself is moving. These sensory inputs work together in reflex arcs to help maintain balance, explained further in this section.

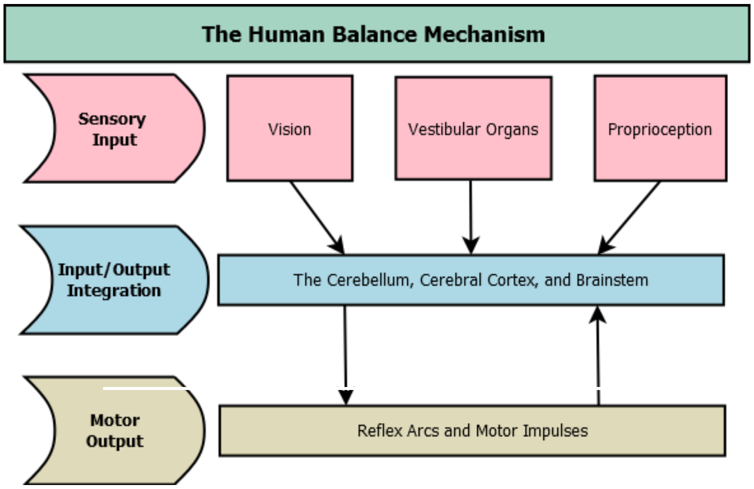

The Human Balance Mechanism
Sensory Input
Vision
Vestibular Organs
Proprioception
Input/Output Integration
The Cerebellum, Cerebral Cortex, and Brainstem
Motor Output
Reflex Arcs and Motor Impulses

Figure 1 - Balance in the human body consists of sensory input, I/O integration, and motor outputs

A vestibular organ is located in each inner ear. Each vestibular organ consists of the otoliths (the utricles and saccules) and three semicircular canals that are oriented at 90 degrees to each other. The semicircular canals detect rotational movement while the otoliths detect gravitational movement. The utricles and saccules detect movement in the horizontal and vertical planes, respectively. The receptive organs, the macula in the otoliths and the cristae in the semicircular canals, have long cilium, called kinocilium, that detect the movement of endolymph fluid as the head and body move. The kinocilium is pre-polarized and thus is providing a constant tonic signal. This means that movement in either direction of the kinocilium causes a change in signal. The two vestibular systems work together but receive opposite signals as they are mirror images of each other. That means if one receptor has an increase in signal in the left vestibule, the corresponding receptor in the right inner ear would receive a decrease in signal (Gray; Khan & Chang, 2013; Swenson, 2006).

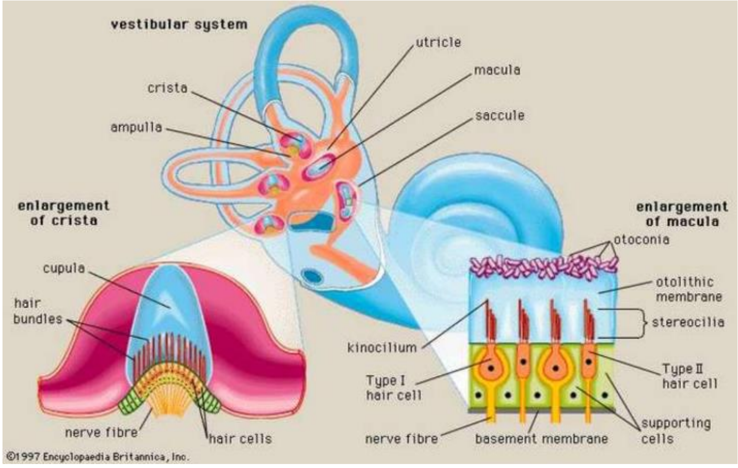


Figure 2 - Vestibular organ, cristae, and macula. ("Vestibular System: Anatomy," 1997)

The vestibular system consists of three main reflex arcs related to balance: the vestibular-ocular reflex (VOR), the vestibular-spinal reflex (VSR), and the vestibular-colic reflex (VRC). The most understood reflex is the VOR, it is used to control eye movements and is important in stabilizing images while they move; it is the reflex that allows the eyes to focus on a stationary object while the head is turning. The activity of this reflex and the corresponding muscle responses is what is measured with the AVETOR device. The VSR and VCR are less understood, but what is known is that the VSR regulates posture and gait and the VCR regulates head stability (Craig, 2016; Gray; Khan & Chang, 2013; Swenson, 2006). Pathologies within the vestibular organs and their reflexes can lead to deficiencies in these reflex arcs and impair an individual's balance. For example, during a vertigo attack in a patient with Meniere's disease, the VOR responses have been shown to be highly erratic due to fluctuations in vestibular function (Yacovino, Hain, & Musazzi, 2017). Thus, monitoring and testing vestibular functionality and reflexes is key in diagnosing any abnormalities in the vestibular organs.

Testing Vestibular Function

Current methods of testing vestibular function include but are not limited to the rotational/tilt chair testing, head thrust/impulse test, caloric vestibular stimulation (CVS), and galvanic vestibular stimulation (GVS) (Amin, 2016). CVS, one of the most common tests, involves irrigation of water, either colder or warmer, than body temperature into the ear canal. This hyperpolarizes the kinocilium and causes a caloric nystagmus of both eyes – the eyes move to one side and then dart back to the center as if the head was turning (Gray). The temperature of the water determines whether there is a contralateral or ipsilateral response. This method takes 15 minutes to wear off and can be uncomfortable for patients. It also only stimulates the horizontal semicircular canals and only allows testing of one ear at a time. CVS also only has a sensitivity of 37.7% and a specificity of 51.2% when diagnosing Meniere's disease, the VOR related illness mentioned earlier (Egami et al., 2013), low sensitivity and specificity can lead to inconsistent diagnoses. The horizontal head thrust/impulse test involves forceful movement of the head and is used to determine VOR dysfunction. This test has a sensitivity and specificity of 71% and 82%, respectively (Farrell & Rine, 2014). However, this exam can be unsafe for the patient and is vulnerable to bias during diagnosis (Hain, 2018). GVS, on the other hand uses an electric current to stimulate the vestibular nerve, which is again dangerous, but also stimulates both the semicircular canals and the otoliths (Noohi et al., 2017). None of these methods test only the function of the otoliths - for that we need to a test to measure VEMPs.

VEMP or vestibular evoked myogenic potentials is a modality developed by Colebatch and Halmagyi in 1994. (Colebatch, 2012; Hecker et al., 2014; Wackym et al., 2012). The test involves stimulating the otoliths with an external stimulus. The muscles responses of the patient are then measured using electrodes. The differences in the peaks and troughs of the EMG can be analyzed to determine abnormalities in the vestibular organs (Craig, 2016; Hain, 2018; Hecker et al., 2014). There are two different VEMPs the cervical, or cVEMP, and the ocular, or oVEMP, named after the location from which they measure the muscle response. The cVEMP is measured from the sternocleidomastoid muscle and tests the functionality of the saccular receptors and the inferior vestibular nerve. The oVEMP is measured from the inferior oblique muscle and tests the functionality of the utricular receptors and the superior vestibular nerve (Craig, 2016; Wackym et al., 2012). VEMP thus allows specificity in testing the otolith organs that the other tests do not provide.

As mentioned, we want to design a device that initiates and records VEMPs for later use in diagnostics. There are two main stimulating techniques, air conducting sound (ACS), that uses high decibel tones to send vibrations to the otoliths, and bone conducted vibrations (BCV) which uses head taps via a reflex hammer to send vibrations through the skull (Noohi et al., 2017; Wackym et al., 2012). The AVETOR uses a BCV as opposed to an ACS because of several advantages: (1) BCV, unlike ACS, can be used on young children, geriatrics, and patients with conductive hearing loss (Hecker et al., 2014; Wackym et al., 2012), leaving very little restrictions on who the test can be administered. (2) BCV can also produce a response with a smaller stimulus. A 70-dB stimulus at 500 Hz can generate a response in the otolith neurons, while ACS needs 120-130 dB stimulus to generate a response at the same frequency (Curthoys et al., 2012), see figure 3 from Curthoy et al where they demonstrate this phenomenon. (3) A drawback of ACS is that it can stimulate other parts of the vestibular system besides the otoliths (Hain, 2018), which we do not want, as we primarily want to study the otoliths. (4) BCV, on the other hand, can activate the otoliths in both inner ears with one stimulus (Hecker et al., 2014), which allows for simultaneous testing of both vestibular organs. (5) Lastly, the BCV, is more comfortable for patients than ACS (Noohi et al., 2017). Overall, BCV is a better method of stimulating the otoliths than ACS.

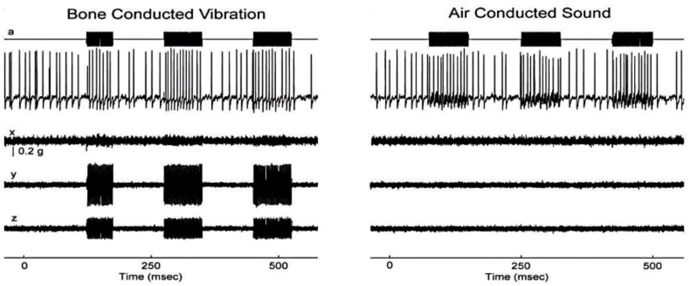


Figure 3 - BCV vs ACS stimulation at 500Hz in the same otolith neuron, (Curthoys, Vulovic, & Manzari, 2012). Top trace is stimulus, second is response of neuron, last 3 are triaxial accelerations of the mandible

To record the VEMPs responses themselves, the most common method is using electrodes to measure the electromyograms (EMGs) of the motor neurons in the sternocleidomastoid and inferior oblique muscles. Instead, we take a novel approach by using force myography (FMG) as our muscle response detector. Force myography is a technique developed at Rutgers, in which FMG sensors measure the variation in surface pressure attributed to change in shape of the muscle as it contracts and relaxes. FMG signals have been shown to be more consistent and produce biphasic waveforms that last longer than EMG signals when measuring muscle activity (Yungher, Wininger, Barr, Craelius, & Threlkeld, 2011). We believe that this method would be beneficial over the EMG method because the amplitude of the VEMPs detected are very small and susceptible to interference from noise and excess skin (Hecker et al., 2014). The FMG, on the other hand, would give steadier, more reliable signals and can be calibrated by providing an initial pressure.

The AVETOR prototype consists of a striker to administer head taps as the BCV stimulus, force sensors to record the induced muscle responses, and an Arduino/Dynabrush board used to control the peripheries and save the acquired data. The following table indicates the required specifications of the device. These specifications are made to mimic a portable BCV striker developed by Wackym et al (Wackym et al., 2012) as it is a similar product. The device is required to fire at a frequency of 2 Hz for a duration of 8 seconds. AVETOR is only allowed to fire when the pressure applied by the striking tip is above an adequate force threshold but still comfortable for the patient. Lastly, the device is required to record and save the incoming sensor signals to a file(s) on the SD card. In the following sections, we will delve into more detail on device construction and function and well as describe how closely AVETOR met the specifications.

AVETOR Device Specifications

Specification #	**Specification Description**	**Unit of Measure**	**Marginal Value**	**Ideal Value**
1	Striking frequency	Hertz	1 – 10 Hz	2 Hz
2	Striking duration	Seconds	Variable	8 seconds
3	Threshold Requirement: Striker only fires when proper placement and pressure are met	Scaled units in Arduino	> 800 units (HIGH signal)	1023
4	File Management: Save data from 2 sensors in new file after each run	# of files	1- 2 files	1 file
5	Continuous signal recording during strike administration	Seconds	Variable (dependent on striking duration)	8 seconds

Figure 4 – AVETOR Device Specifications

METHODS:

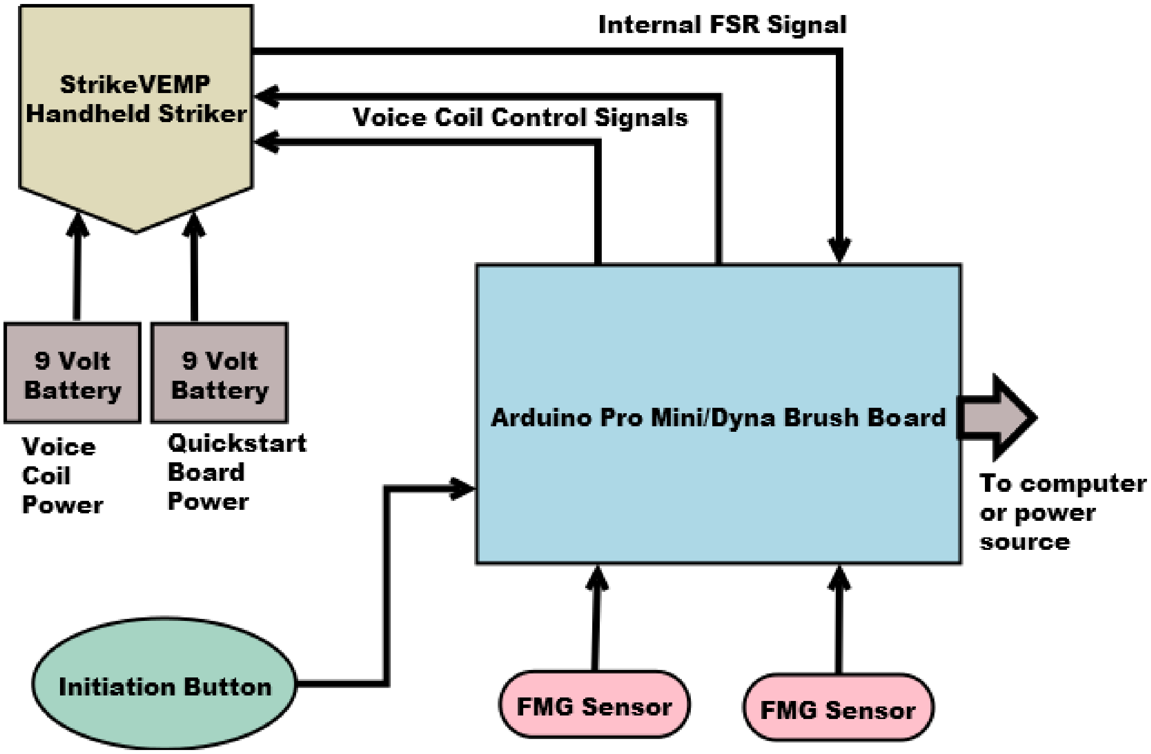


Figure 5 - AVETOR Prototype schematic

Device Components

The AVETOR prototype consists of a programmable controller, a StrikeVEMP striker, 2 FMG sensors, an initiation button, and power sources, see figure above. The controller is the brain of the device that sends and receives data from the peripheries, it also houses the data acquisition and processing mechanism. The striker takes the place of the reflex hammer in producing BCV to stimulate the vestibular organs. The 2 FMG sensors detect the induced VEMPs and send the signals back to the controller. The initiation button is used to begin the striker and data acquisition process. In this section, we will go into further details of each component.

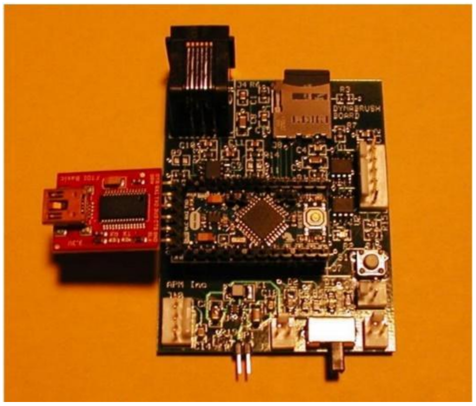


Figure 6 - Arduino Pro Mini/Dynabrush board and the attached FTDI breakout board

The AVETOR controller consists of an Adafruit Arduino Pro Mini connected to a Dynabrush board designed in the RuRehabLab and produced by Applied Processors and Measurement, Inc. The Arduino board contains the ATMega328 processor (3.3V DC, 8MHz) that can be programmed with the open source Arduino IDE (version 1.8.5). The Arduino is connected to a computer for development via Sparkfun's FTDI BASIC breakout board. As mentioned, the Arduino Pro Mini is mounted on the Dynabrush board. The Dynabrush board contains signal processing circuitry, a removable SD card, connections and switches for an external power source, Auxiliary I/O ports, 2 LEDs, and a 6-pin registered jack (RJ45). In application mode, the breakout board is disconnected and both boards are powered by 2 AA batteries connected to the Dynabrush board. The RJ45 is used to

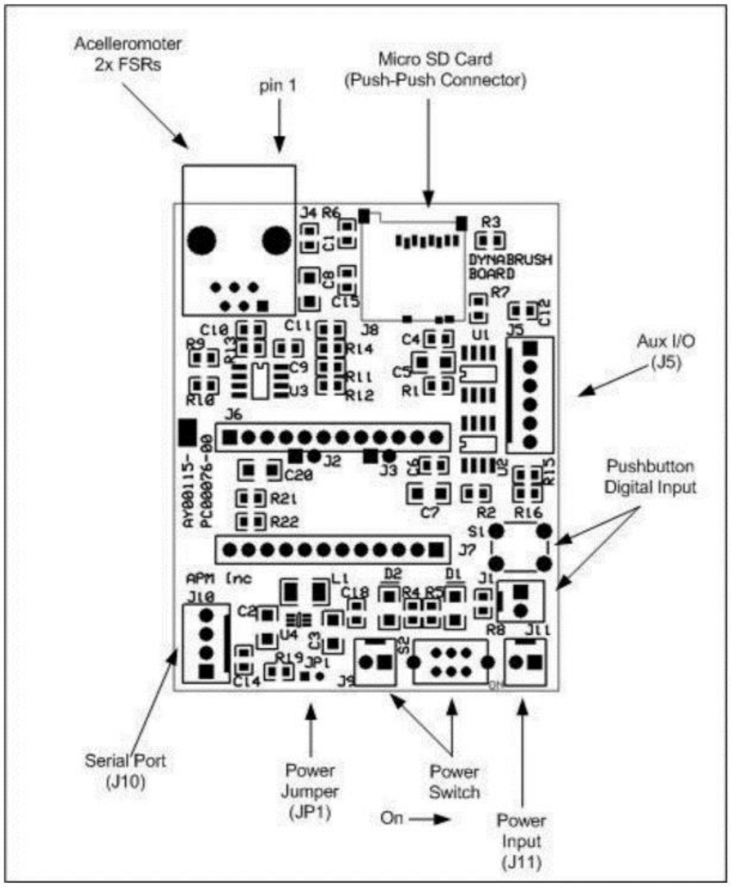


Figure 7 - Components of the Dynabrush board

connect the FMG sensors to the board while the Auxiliary I/O ports are used to control the striker. The Dynabrush board also contains ports for attachment to an accelerometer (RJ45) which are not used in this application but can be used to monitor head position/movement.

StrikeVEMP Striker

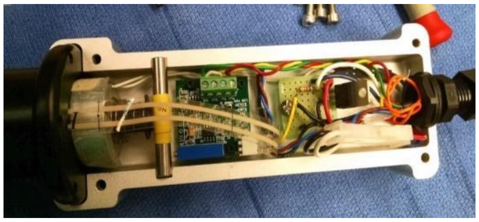


Figure 8 -Internal of the StrikeVEMP striker, including the metal pin, spring, FlexiForce board, and switching circuit (from left to right)

The StrikeVEMP striker is an ergonomically designed automated reflex hammer developed for the StrikeVEMP system by Dr. Ashley Wackym. It comprises of a voice coil motor, the FlexiForce QuickStart board, an internal FSR sensor and a switching circuit. The striker requires two separate power sources, a 9-volt battery for the voice coil motor and another 9-volt battery for the FlexiForce board. A metal pin accompanied by a spring inside the voice coil provides the striking force, similar to that of a reflex hammer. As current is applied to the voice coil, a magnetic field is formed that propels the metal pin forward and compresses the surrounding spring. When the current is removed, the magnetic field disperses. This allows the compressed spring to relax and push the metal pin back. As the metal pin moves back and forth, it contacts the striking tip that provides an even, flat surface for force administration on to a patient. The striking tip is attached in to the striker housing allowing it to move to deliver the necessary force. Successive application and removal of current produces a repeated striking mechanism that is needed to stimulate the vestibular system. The charging and discharging of the voice coil is controlled by the switching circuit that receives input in the form of a square wave from the controller. The internal FSR sensor, or force sensitive resistor (FlexiForce A201

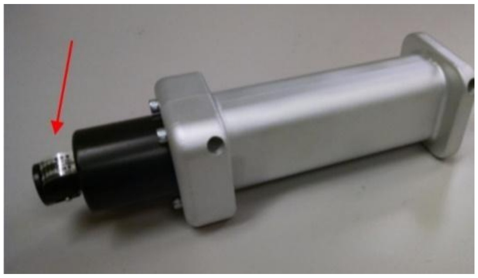


Figure 9 - StrikeVEMP striker, red arrow points to internal FSR sensor at the striker tip

sensor), provides a placement check for the striker. It is located in the striking tip and sends its signal to the FlexiForce board, which continues the signal to the controller. The force detected at the FSR must be above a specific threshold before allowing the striker to fire. This ensures that the striker is placed on the patient with enough pressure and contact to properly administer the striking force. Once the signal from this wire is above the threshold, the striker is primed to fire. The entire striking mechanism, FlexiForce board, internal FSR sensor, and switching circuit are located within an aluminum enclosure. A cable leaves from the opposite end of the striker, containing the wires necessary to control and power it. The cable contains 7 wires, 2 to power the voice coil and another 2 to power the FlexiForce Board, while the last 3 carry signals. Of the signal carrying wires, one carries the signal from the internal FSR that is used to check for proper placement of the striker. The last two wires connect to the switching circuit in the striker and receive the square wave that causes the striker to fire.

FMG Sensors

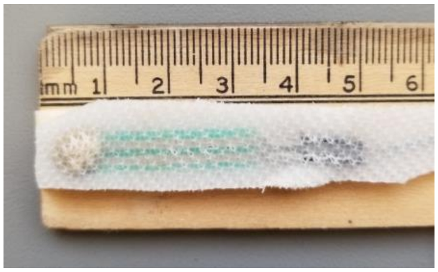


Figure 10 - FMG sensor embedded in silicon, note markings on ruler are in centimeters

Data acquisition is conducted by two FMG sensors that connect to the Dynabrush board via the RJ45 port. These sensors are flexible FSRs embedded in silicon. The silicon makes the sensors more comfortable for the subject and provides easier application of pressure. Each FSR is applied as part of a voltage divider circuit with a fixed resistor (valued at 10K Ohms) located on the Dynabrush board. The FSR works by changing its resistance, as pressure is applied. When no pressure is applied the FSR acts as an infinite resistor or open circuit. As more pressure is applied the resistance in the resistor gradually decreases. This affects the voltage loaded across the FSR and a fixed resistor in the voltage divider circuit. The FMG signal is the voltage loaded over the fixed resistor. The more pressure applied, the less resistance in the FSR and a greater voltage (and current) loaded over the fixed resistor, On the other hand, lower applied pressure leads to greater resistance in the FSR, and thus lowers the voltage load at the fixed resistor. The change in voltage load across the fixed resistor is sent to the controller and is recorded as the FMG signal. However, before the signal reaches the Arduino, it is first processed in the Dynabrush board. The signal passes through a noninverting amplifier to increase its signal strength and then a low pass filter, which removes any high frequency electrical noise. Once the data acquisition is complete, it is saved to the SD card mounted on the Dynabrush board.

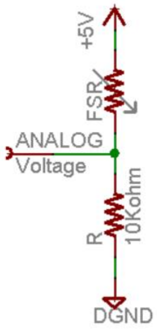


Figure 11 - Voltage divider circuit with a FSR and a fixed resistor. In AVETOR the voltage input is 3.3V. The FMG signal is the analog voltage on the fixed resistor.

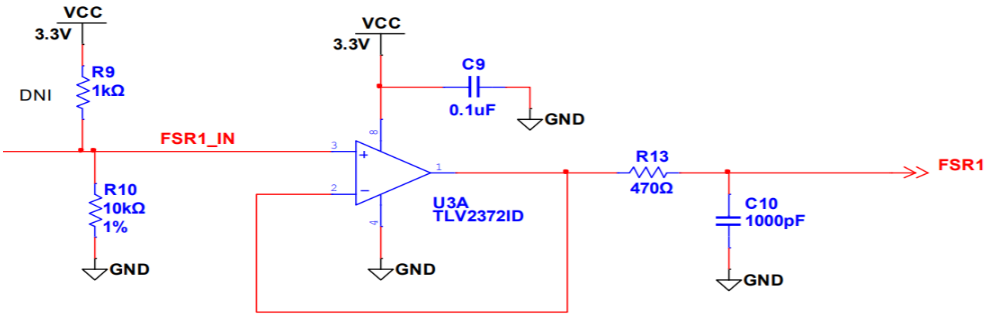
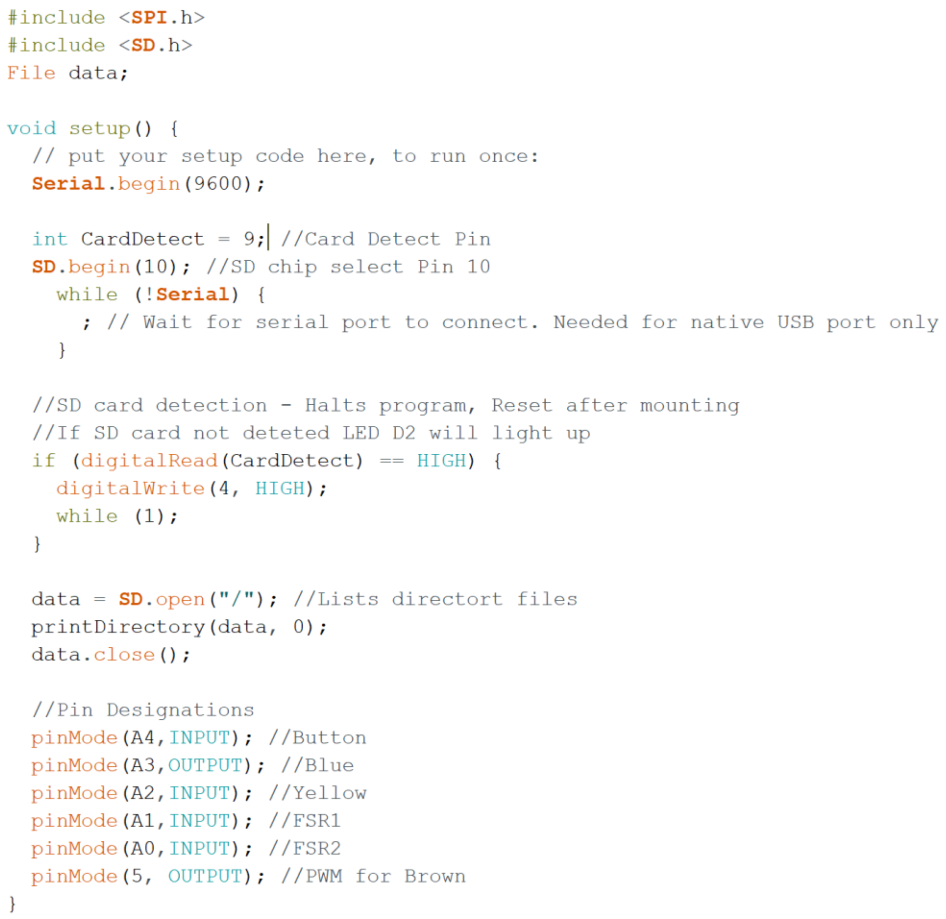


Figure 12 - Signal processing circuitry from Dynabrush board: amplifier and low pass filter

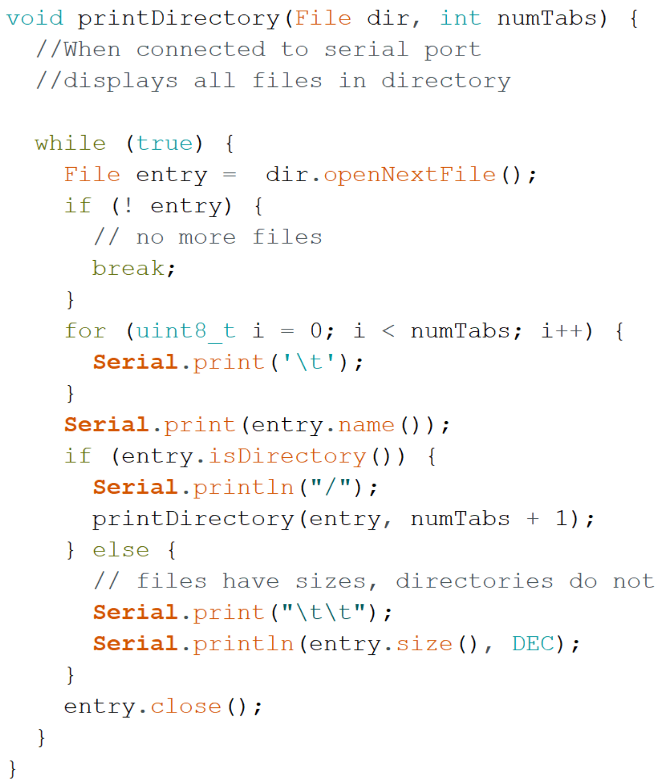
Controller Program

The program running on the Arduino Pro Mini consists of 5 functions: setup, loop, NewFile, StrikeRecord, and printDirectory. In the Arduino IDE, the setup function is always called once when the controller is first powered or is reset. This function designates the I/O pins, opens the serial port, and connects to the SD card. It provides defensive programming by stalling the program if a SD card is not mounted; this feature also lights up the D2 LED as a warning. Mounting the SD card and then resetting the Arduino restarts the program. Within the setup function, the printDirectory function is called. This Arduino provided function, prints a list of the files on the SD to the serial port. See figures below for the code in the setup and printDirectory functions.



```
#include <SPI.h>
#include <SD.h>
File data;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  int CardDetect = 9; //Card Detect Pin
  SD.begin(10); //SD chip select Pin 10
    while (!Serial) {
      ; // Wait for serial port to connect. Needed for native USB port only
    }

  //SD card detection - Halts program, Reset after mounting
  //If SD card not deteted LED D2 will light up
  if (digitalRead(CardDetect) == HIGH) {
    digitalWrite(4, HIGH);
    while (1);
  }

  data = SD.open("/"); //Lists directort files
  printDirectory(data, 0);
  data.close();

  //Pin Designations
  pinMode(A4,INPUT); //Button
  pinMode(A3,OUTPUT); //Blue
  pinMode(A2,INPUT); //Yellow
  pinMode(A1,INPUT); //FSR1
  pinMode(A0,INPUT); //FSR2
  pinMode(5, OUTPUT); //PWM for Brown
}
```

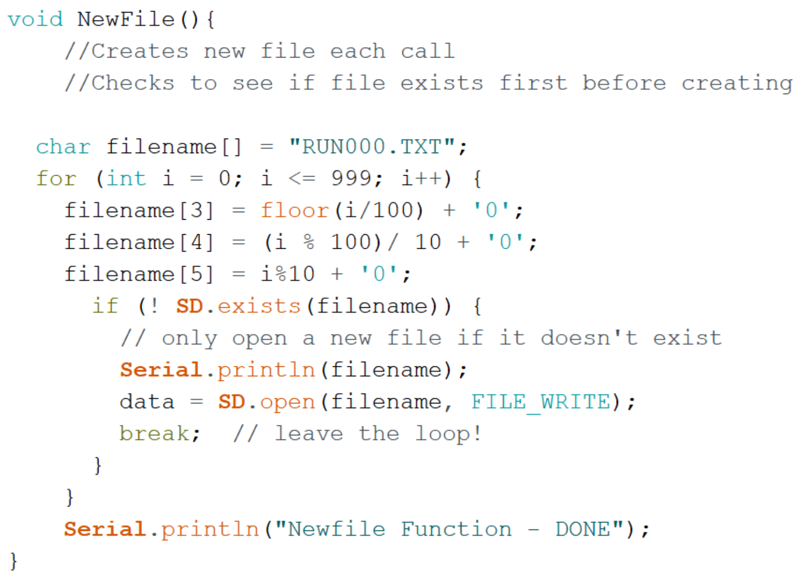
Figure 13 - The setup function code



```
void printDirectory(File dir, int numTabs) {
  //When connected to serial port
  //displays all files in directory

  while (true) {
    File entry =  dir.openNextFile();
    if (! entry) {
      // no more files
      break;
    }
    for (uint8_t i = 0; i < numTabs; i++) {
      Serial.print('\t');
    }
    Serial.print(entry.name());
    if (entry.isDirectory()) {
      Serial.println("/");
      printDirectory(entry, numTabs + 1);
    } else {
      // files have sizes, directories do not
      Serial.print("\t\t");
      Serial.println(entry.size(), DEC);
    }
    entry.close();
  }
}
```

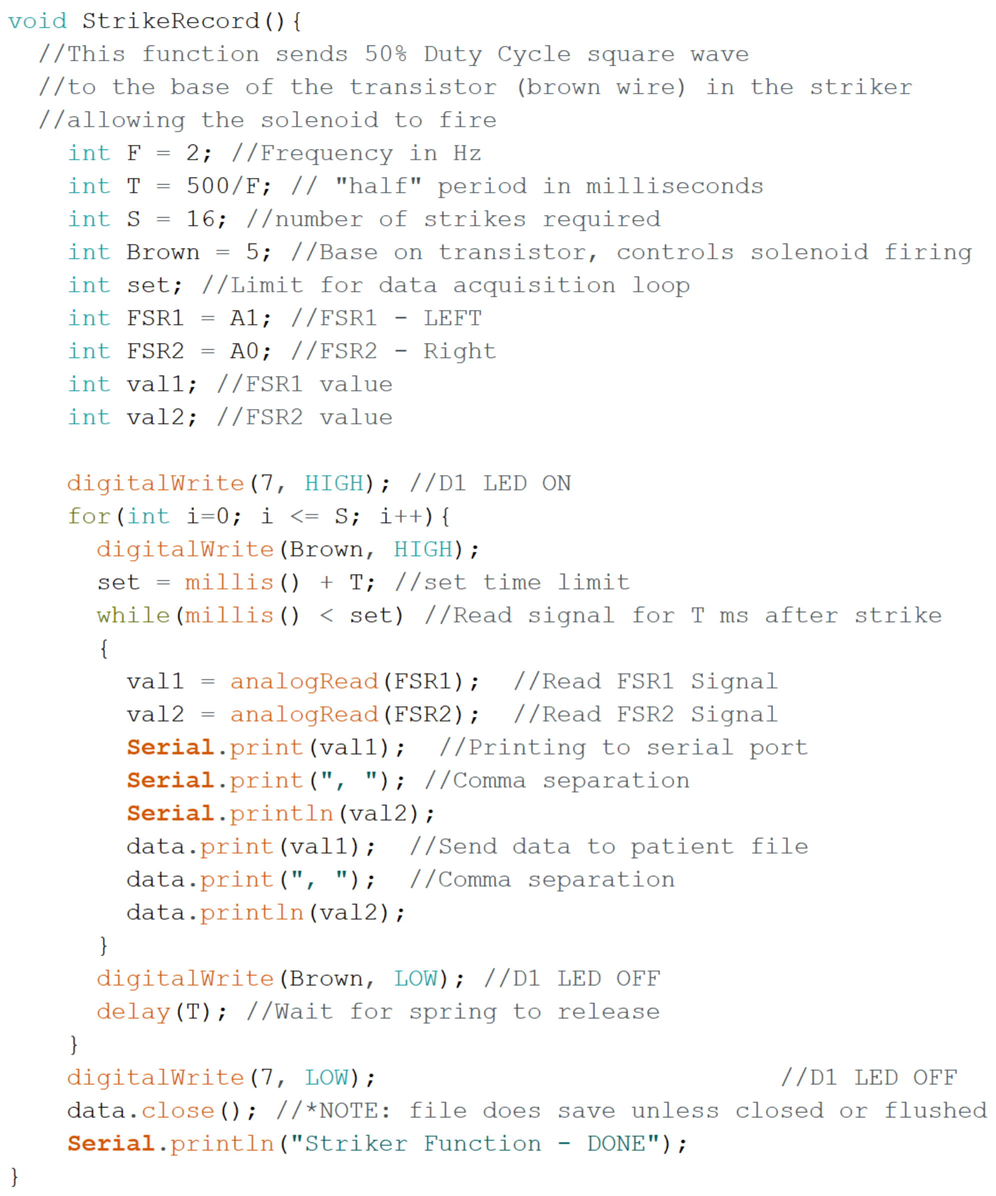
Figure 14 - The printDirectory function code

The NewFile and StrikeRecord functions are the most important in the controller. NewFile creates a new file to log the data for each test (Guest, 2012). It cycles through the SD card comparing file names until it finds one that does not currently exist. It then creates and opens the file to be ready to receive data. The file is named as “RUNXXX.TXT”, where the three X’s represent the file or run number. Each test result is saved in its own text file. The StrikeRecord function causes the striker to file, as well as records the FMG signals coming into the sensors. The striker is fired by sending a sequence of square pulses to it, with a 50% duty cycle. The frequency of which can be specified in the code, as well as the number of strikes. Between each strike, the FMG signal is recorded for half the period of the square wave and saved to the currently opened file. The FMG signal is also printed to the serial port to view in real time. The function then closes the file saving the data. During the duration of this function, the D1 LED is on to show that data acquisition is occurring. Both NewFile and StrikeRecord functions print a message to the serial port to signify their completion.



```
void NewFile(){
    //Creates new file each call
    //Checks to see if file exists first before creating

  char filename[] = "RUN000.TXT";
  for (int i = 0; i <= 999; i++) {
    filename[3] = floor(i/100) + '0';
    filename[4] = (i % 100)/ 10 + '0';
    filename[5] = i%10 + '0';
      if (! SD.exists(filename)) {
        // only open a new file if it doesn't exist
        Serial.println(filename);
        data = SD.open(filename, FILE_WRITE);
        break;  // leave the loop!
      }
    }
    Serial.println("Newfile Function - DONE");
}
```

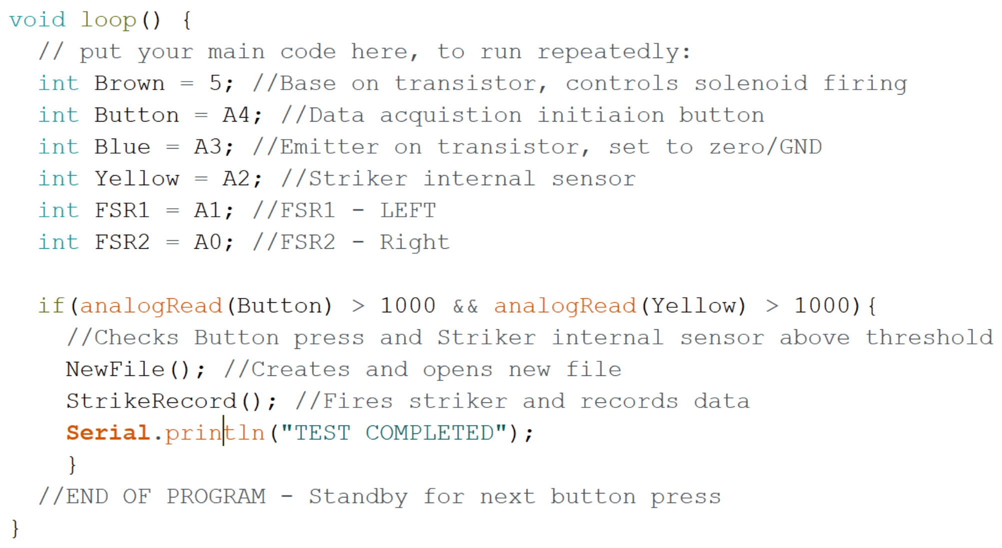
Figure 15 - The NewFile function code



```
void StrikeRecord(){
  //This function sends 50% Duty Cycle square wave
  //to the base of the transistor (brown wire) in the striker
  //allowing the solenoid to fire
    int F = 2; //Frequency in Hz
    int T = 500/F; // "half" period in milliseconds
    int S = 16; //number of strikes required
    int Brown = 5; //Base on transistor, controls solenoid firing
    int set; //Limit for data acquisition loop
    int FSR1 = A1; //FSR1 - LEFT
    int FSR2 = A0; //FSR2 - Right
    int val1; //FSR1 value
    int val2; //FSR2 value

    digitalWrite(7, HIGH); //D1 LED ON
    for(int i=0; i <= S; i++){
      digitalWrite(Brown, HIGH);
      set = millis() + T; //set time limit
      while(millis() < set) //Read signal for T ms after strike
      {
        val1 = analogRead(FSR1);  //Read FSR1 Signal
        val2 = analogRead(FSR2);  //Read FSR2 Signal
        Serial.print(val1);  //Printing to serial port
        Serial.print(", "); //Comma separation
        Serial.println(val2);
        data.print(val1);  //Send data to patient file
        data.print(", ");  //Comma separation
        data.println(val2);
      }
      digitalWrite(Brown, LOW); //D1 LED OFF
      delay(T); //Wait for spring to release
    }
    digitalWrite(7, LOW);                           //D1 LED OFF
    data.close(); //*NOTE: file does save unless closed or flushed
    Serial.println("Striker Function - DONE");
}
```

Figure 16 - The StrikeRecord function code

Loop is the core function running in the background that is responsible for calling the other two functions. Its main purpose is to analyze the signals coming from the internal FSR in the striker and the initiation button. If the signal from the internal FSR is above the threshold, while the initiation button is pressed, the loop function calls NewFile and StrikeRecord. It also prints a message to the serial port to signify its and the test's completion and then waits for the next button press and threshold requirement. Until both conditions are met, the controller is in standby and waits for the requirements.



```
void loop() {
  // put your main code here, to run repeatedly:
  int Brown = 5; //Base on transistor, controls solenoid firing
  int Button = A4; //Data acquistion initiaion button
  int Blue = A3; //Emitter on transistor, set to zero/GND
  int Yellow = A2; //Striker internal sensor
  int FSR1 = A1; //FSR1 - LEFT
  int FSR2 = A0; //FSR2 - Right

  if(analogRead(Button) > 1000 && analogRead(Yellow) > 1000){
    //Checks Button press and Striker internal sensor above threshold
    NewFile(); //Creates and opens new file
    StrikeRecord(); //Fires striker and records data
    Serial.println("TEST COMPLETED");
    }
  //END OF PROGRAM - Standby for next button press
}
```

Figure 17 - The loop function code

Device Operation

To operate the AVETOR prototype, first, plug the microUSB portion of the FTDI cable into the FTDI basic board that is attached to the Arudino/Dynabrush board. Second, connect the other end of the FTDI cable into a USB port on a computer or laptop that is running the Arduino IDE software, this will power up the controller. Open the Serial Monitor under the Tools tab of the IDE. At this point the program will run the SD check, to see if it is mounted. If it is, the printDirectory function will run and display the current files on the memory card. If not, the program will halt, and the D2 LED will turn on, indicating the missing SD card. Insert the SD card and press the reset button located on the

Arduino Pro Mini. Next connect the two 9-volt batteries to power the voice coil and FlexiForce QuickStart board within the striker.

The next step is to attach the FMG sensor to the desired area on the patient. To measure cVEMPs attach the FMG sensors along the upper one third of the sternocleidomastoid muscle. To measure oVEMPs attach the FMG sensors right below the eye, aligned with the pupils. Have the patient don a cuff or goggles to apply a base pressure to the sensors. Firmly grasp the striker and place at the Fz location of the patient's forehead (center of the hairline). Gently apply the pressure from the tip until the internal FSR sensor is above the threshold and press the initiation button. The NewFile function will open a new file and indicate is conclusion. Then the StrikeRecord function will run. It will fire the striker while the Serial Monitor displays the values at the FMG sensors; the D1 LED will also turn on. Once the data acquisition is complete, the D1 LED will turn off and the StrikeRecord function will indicate is conclusion.

At this point the Arduino can be reset to allow printDirectory to show the addition of the new file to the list of files on the SD card, or the test can be administered again by having the internal FSR reach threshold and pressing the initiation button. The data from the next test will be stored in a new file. To power off the device, remove the 9-volt batteries from their sockets and unplug the FTDI cable from the PC.

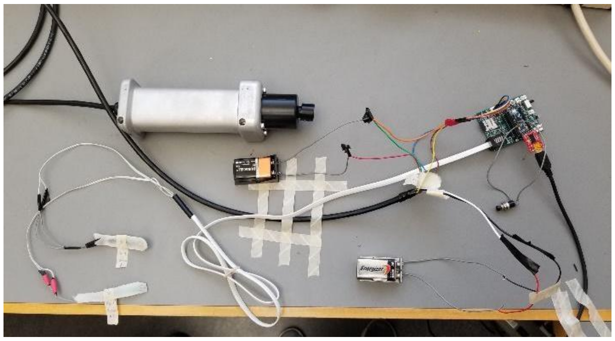


Figure 18 - The AVETOR device, includes: StrikeVEMP striker, FMG sensors, power sources, and Arduino/Dynabrush board

Figure 19 - Goggles worn by subjects during testing, applies base pressure to the FMG sensors

Testing Muscle Activity

My goal in the thesis was to a design a device for evoking and recording the VOR response. I did not test using EMG responses, but rather used FMG sensors. These sensors were never used for the VOR response, and therefore may constitute a new sensor modality. Thus, to confirm or deny this phenomenon, two main goals must be achieved. First, do the FMG sensors detect ocular muscle activity? Second, can the FMG sensors detect the VOR response? Determining the accuracy of the response is out of the scope of the thesis, as it requires a trained medical professional. It is important to note that the testing protocol is used to determine the functionality of the AVETOR device and its ability to meet its specifications, listed in the Introduction section. Since, this is also a novel approach to measuring the VOR response, further testing to determine the correlation of variations to FMGs must be completed but is out of the scope of this thesis.

The testing procedure included measuring the ocular muscle responses in 3 volunteers. To measure the inferior oblique muscle activity, the FMG sensors were placed right below each eye, aligned with the pupils. The FSR1 sensor was associated with the patient’s left eye and the FSR2 sensor with the patient’s right. Goggles were worn above the sensors to provide a base pressure to the sensors. Subjects were asked to keep their eyes open and look up during each test to expose more of the inferior oblique muscle to the sensors. The muscle activity was measured multiple times. Either with no eye movement to determine a baseline, or with the eyes moving to determine the effect of inferior oblique contraction on the sensors, or with the striker to check for vestibular stimulation and VEMP production. The striker was placed at the Fz location, the center of the forehead at the edge of the hairline. It was then fired 16 times at a frequency of 2 Hz (total striking time - 8 seconds). The following section contains the results of the testing to determine the feasibility of the FMG sensors to detect ocular muscle activity by using the AVETOR prototype. Improvements to the functionality of the device were conducted during testing to achieve the desired specifications.

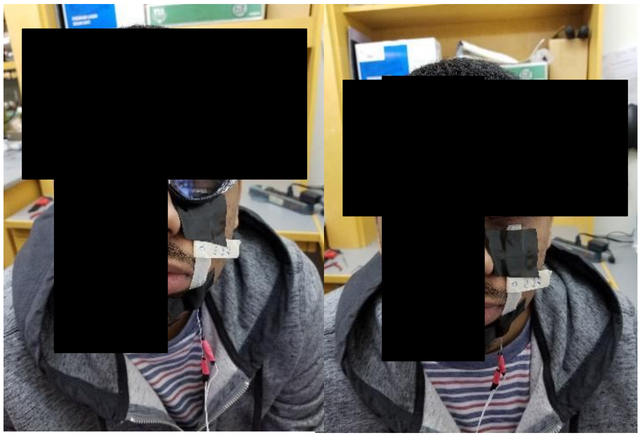
Consent

The subjects read and understood the testing process and signed the informed consent form approved by the Rutgers IRB (Protocol #20170001973).

RESULTS:

The results below contain the FMG signals over time collected by the AVETOR prototype and then plotted with Microsoft Excel. The x – axis contains subsequent data points relating to time. However, these are not a continuous recording of the FMG signal due to the nature of the data acquisition. The y – axis is the FMG signal in scaled millivolts. When receiving voltage, the Arduino scales the signal between 0 units and 1023 units, with 0 being GND or 0V and 1023 being 3.3V (the max voltage powering the Arduino) loaded over the fixed resistor. It is important to note that all FMG signals received were below 200 units and often, below 100 units. This means that the voltage loaded across fixed resistor was very small. Thus, the FSR had a great resistance and did not receive a large pressure.

Subject 1



[redacted] [redacted]

Figure 20 - Subject 1 with taped sensors and goggles

For the first test subject, only one eye was tested. The sensor was placed below the subjects left eye and taped to the skin on the cheek for the duration of the tests. In previous trial runs, it was determined that taping the entirety of the sensor to the cheek, as opposed to just right below the eye, was more comfortable for the subject, it also allowed the sensor to be flush with the subject’s cheek.

The graph below contains the results of the first test. The subject was asked not to move his eyes to prevent muscle contraction. As you can see, there is change in the signal from 100 - 400 samples. I believe this is due to the patient moving during the test therefore changing the pressure on the sensor. Before and after this movement, the pressure measured at the sensor was fairly constant.

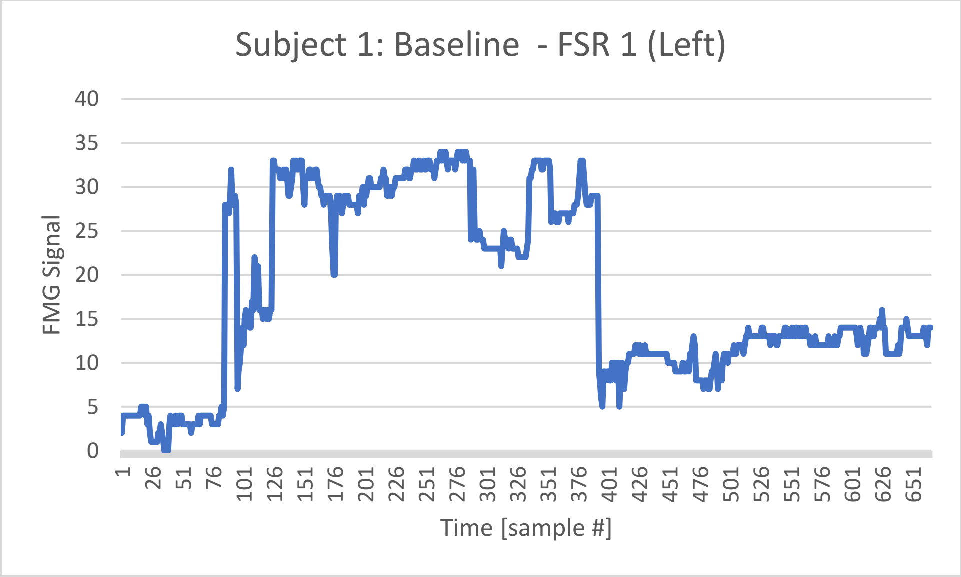


Figure 21 - Subject 1, baseline test, no movement of eyes

The next two graphs contain data from the muscle contraction on the eyes. In this part of the testing, the subject was asked to move his eyes vertically (U/D) for one test and horizontally (L/R) for the next. This type of movement produces a response from the contraction and relaxation of the inferior oblique and possibly the inferior rectus muscles around the eyes. There is rapid variation in the FMG signal corresponding to the eye movement. The U/D motion created more varied and larger peaks compared to the L/R motion. This could be due to the sensor being in better contact with the muscle, therefore receiving a larger pressure change.

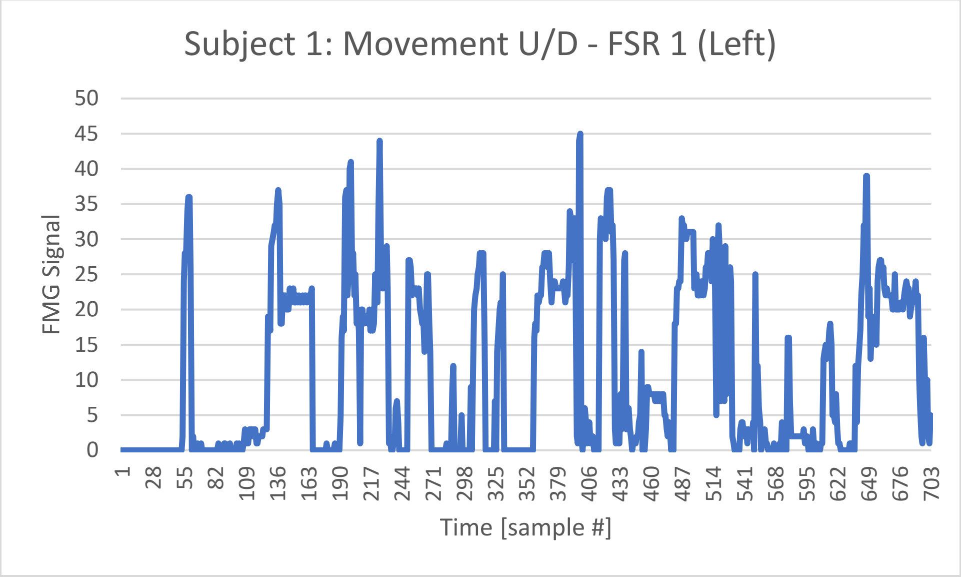


Figure 22 - Subject 1, purposeful eye movement, vertical

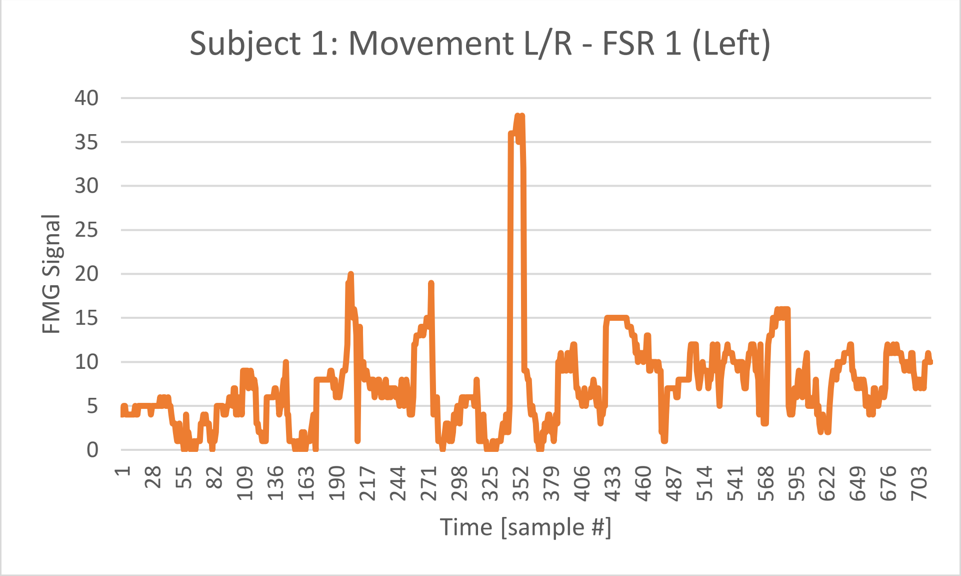


Figure 23 -Subject 1, purposeful eye movement, horizontally

The last graph for this subject contains data collected as the subject received strikes at the Fz location of the forehead. As this is his left eye, it corresponds to VEMPs produced by the right vestibular organs, as ocular VEMPs are contralateral responses. There is not as much variation in the oVEMP data compared to either of the purposeful eye movement test results, but more compared to that of the baseline data set. The peak between 300 –

400 samples, is consistent with pressure pulses common with muscle contractions. However, because of the data acquisition method that was applied during subject 1's testing it is difficult to be sure as to what this peak actually relates to. It is possible that this is the ocular muscle's response, but it is equally likely to be from the patient moving during the exam.

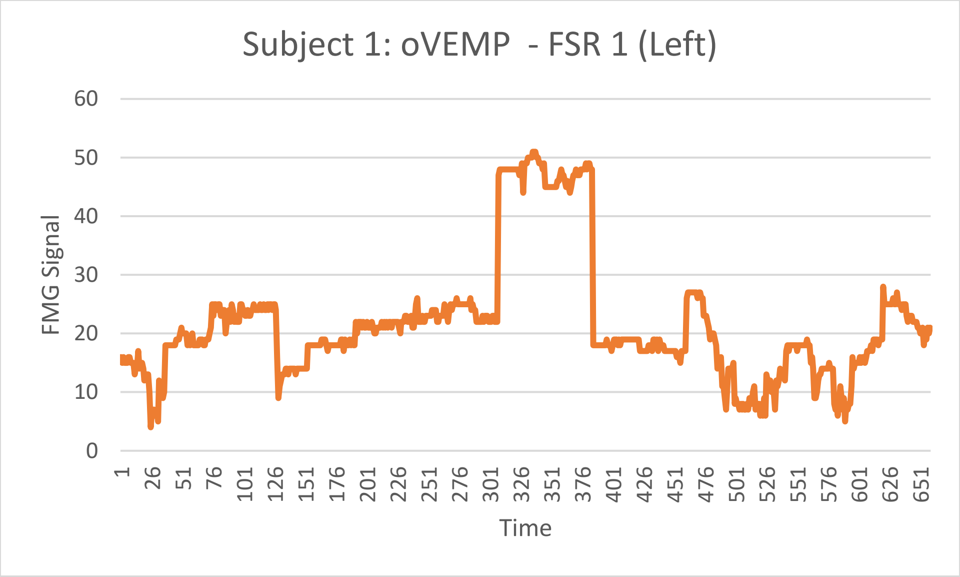


Figure 24 - Subject 1, strike administration, there is variation in this FMG response most likely due to blinking

Subject 2

The next set of charts are from a second subject. For this subject muscle responses from both eyes were taken during testing. The sensors were taped to the subject's face and covered with goggles to apply a base pressure. Similar tests to the first subject were

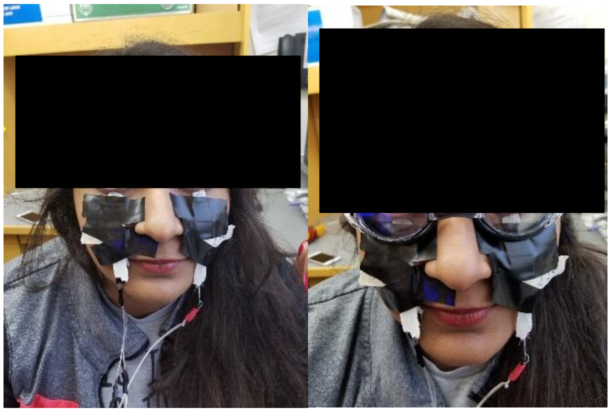

[redacted]
[redacted]

Figure 25 - Subject 2 with taped sensors and goggles.

administered, however additional tests with improvements in the code were also given. These changes will be explained later in this and the discussion sections.

In this first graph, the subject was asked to look forward and not blink or look around. This was to measure the baseline pressure compared at both FMG sensors. As you can see, both sensors have different base values, FSR1 is approximately at 20 units while FSR2 is approximately 55 units. This difference in values is could be due to differences in the silicon coating of the sensors or to a lack of calibration in the FMG sensors. However, both signals do remain constant throughout the test, with any variation due to noise.

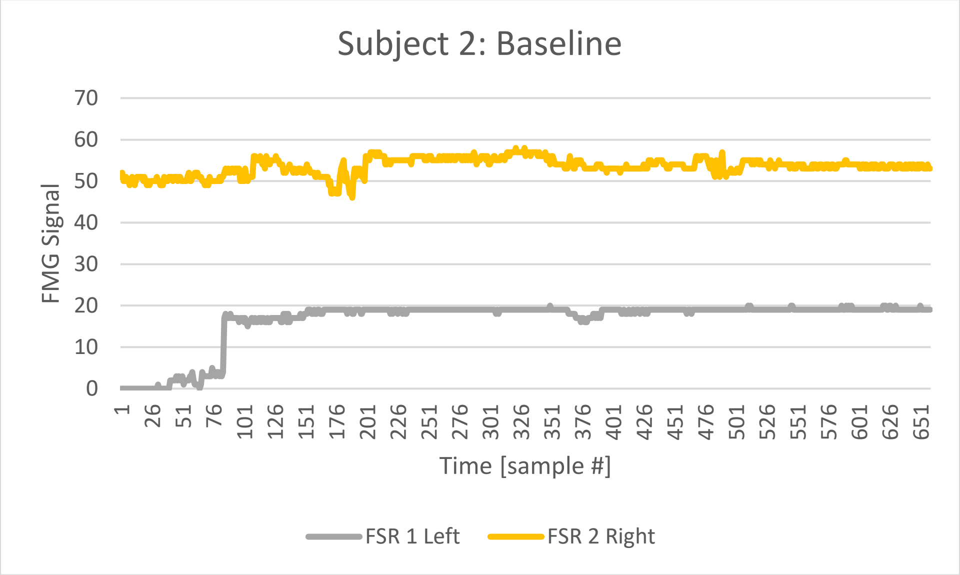


Figure 26 - Subject 2, baselines test, no eye movement

The next graph is from the subject moving her eyes vertically for duration of the test. Here the signals are much more varied compared to the baseline. They are not consistent, like the response received from subject 1. However, there is an overlap of similar responses between both eyes from ~140 – 300 data points.

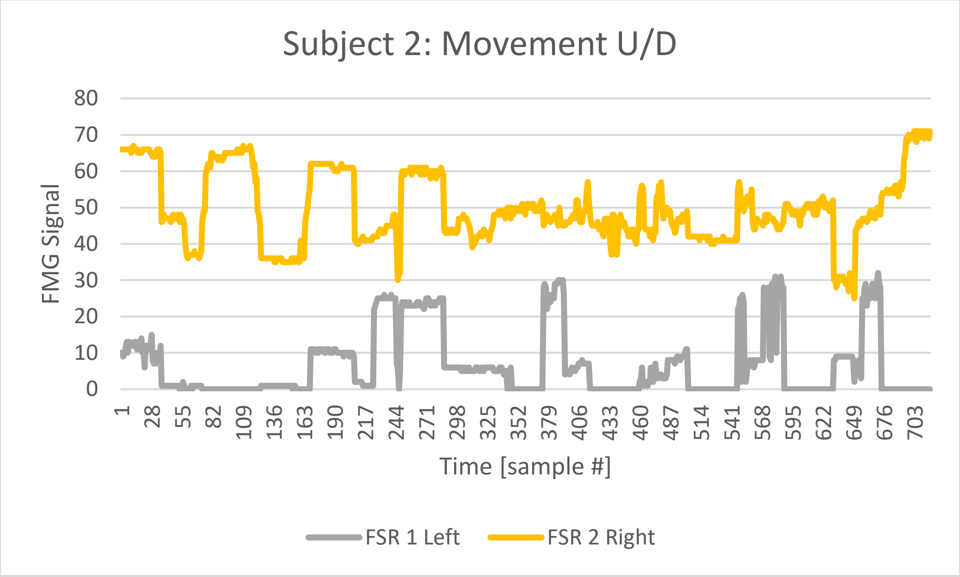


Figure 27 - Subject 2, purposeful eye movement, vertical

The graph below contains data acquired during strike administration. Like subject 1, there is more variation compared to the baseline signal, but not as much compared to the purposeful eye movement signal.

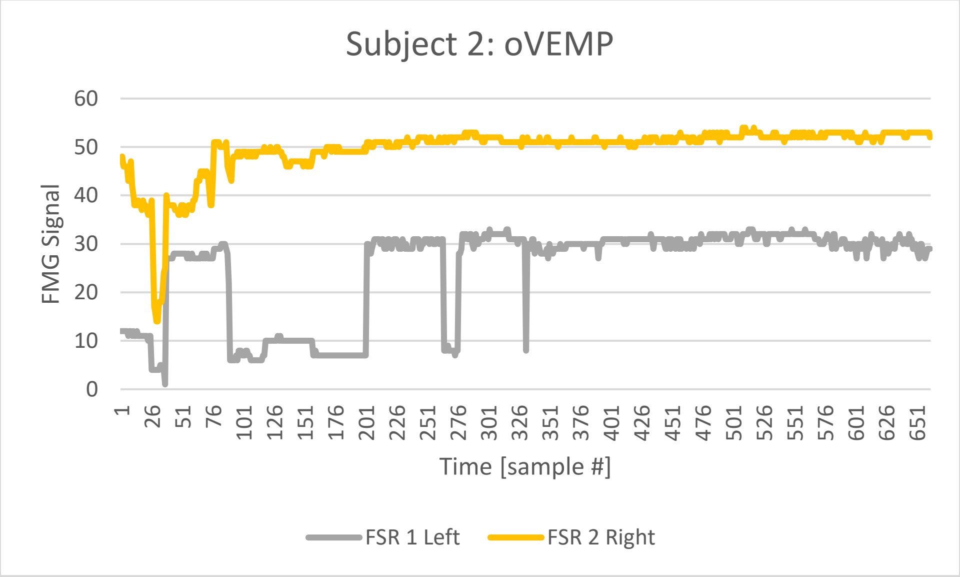


Figure 28 - Subject 2, strike administration

Due to the order in which the striker fires and data acquisition takes place in the code, it came to my attention that the signal recorded after each strike is linked directly to the signal from the previous strike. Since data collection only occurs during the high portion of the square wave sent to the striker it is not a continuous reading of the FMG signal. This makes the FMG signal appear to be one entire signal, even though it's pieces of signals strung together. For the next 2 tests, I incorporated a gap into the data acquisition code that separates the signals acquired after each strike. Now the data from each strike could be analyzed separately. The first two of the following four graphs are from purposeful eye movement tests while the last two are from the strike administration tests. The first two graphs have variation in the FMG signal as the sensors are detecting the movement of the eye, whereas the last two graphs, show little to no change in signal, except at the first two strikes. The subject stated that she blinked during the beginning of that test, attributing to this change. The only major difference in these results compared to other results for subject 2, is that the FMG signal from FSR2 jumped from 55 units, its baseline, to 155 units for both the eye movement and striker tests then and continued to climb for the striker test. This leads me to believe that the change is not due to the VOR response, but a change in the base pressure applied to FSR2 as this occurred for both tests.

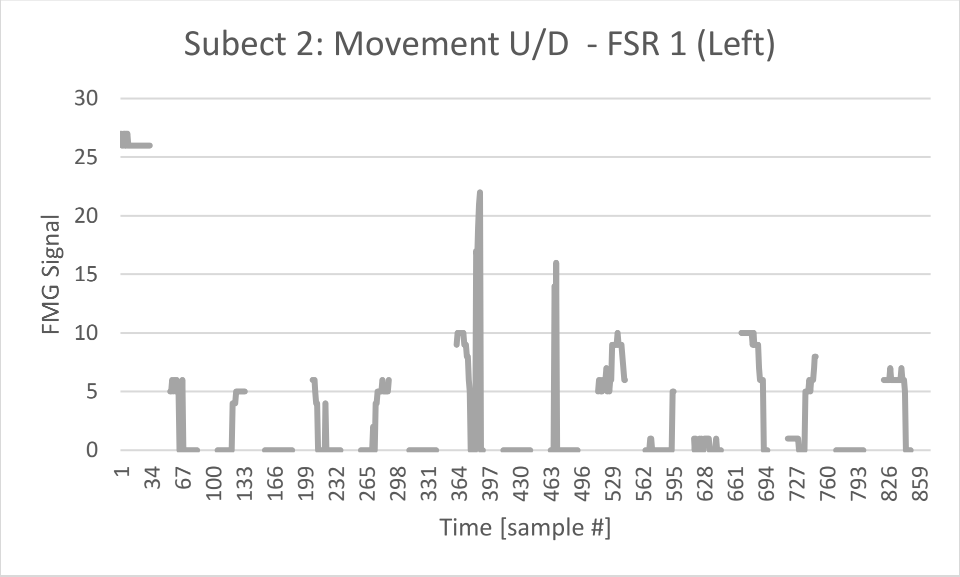


Figure 29 - Subject 2, purposeful eye movement, vertical, from FSR1. Using gap in code.

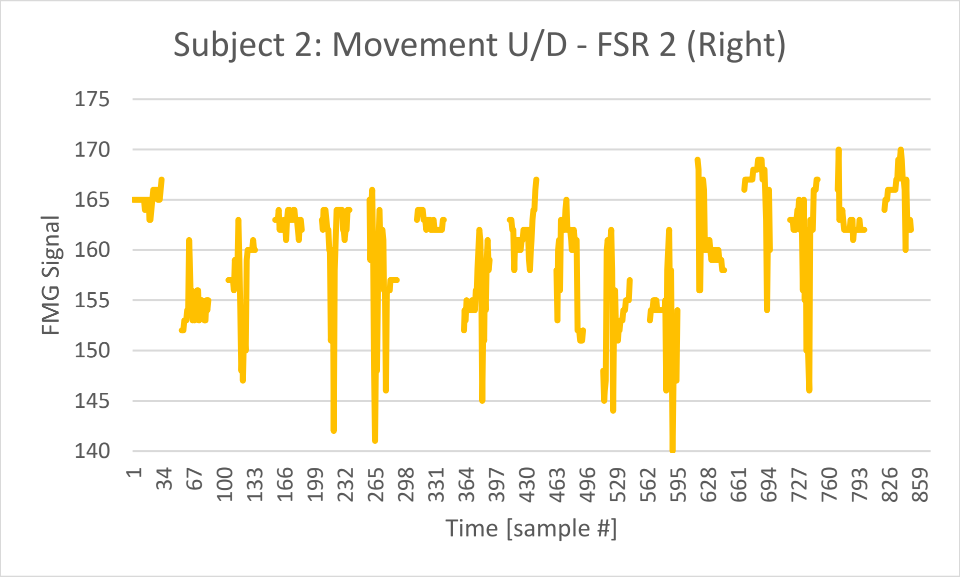


Figure 30 - Subject 2, purposeful eye movement, vertical, from FSR2. Using gap in code.

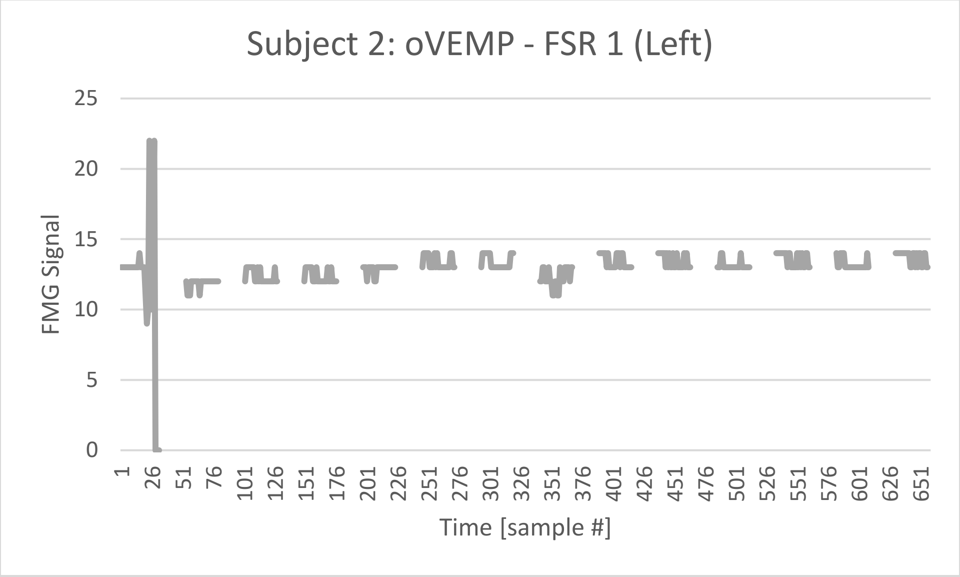


Figure 31 - Subject 2, strike administration test, from FS1. Subject blinked at beginning of test.

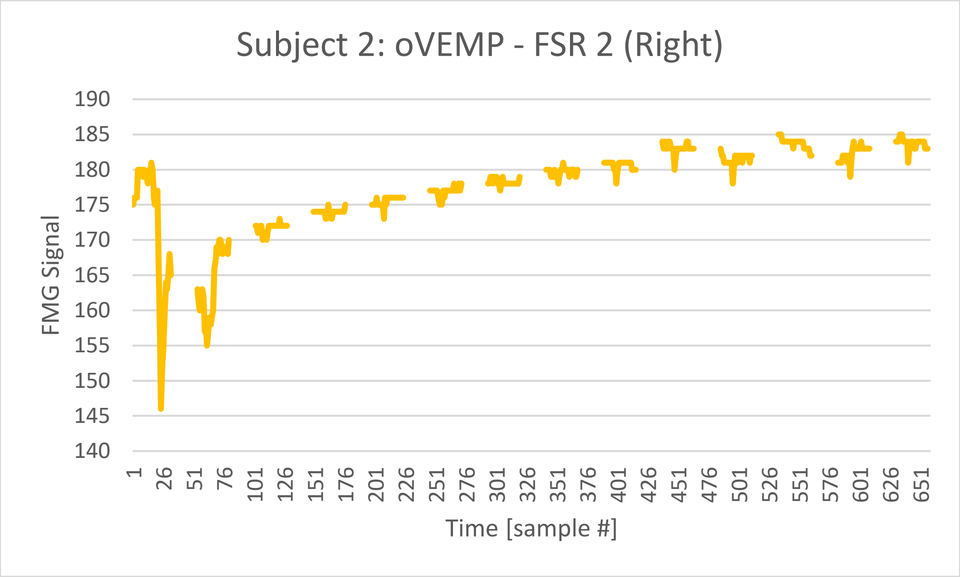


Figure 32 - Subject 2, strike administration test, from FSR2. Subject blinked at beginning of test.

An additional test was taken on subject 2. In this test, I asked the subject to move her eyes vertically for the duration of the test. This test used an altered code that only acquires data for a continuous 10 seconds and has no striker functionality. In the results below, we can identify a consistent pressure change due to the movement of the eyes, indicating that the FMG sensors can indeed detect the muscle response.

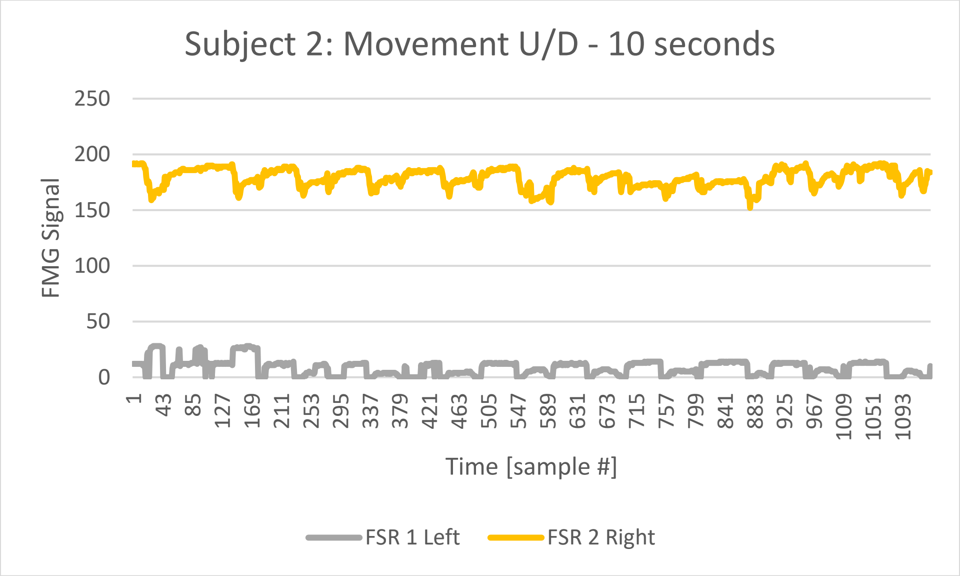


Figure 33 - Subject 2, purposeful eye movement, vertical, for continuous 10 seconds

Subject 3

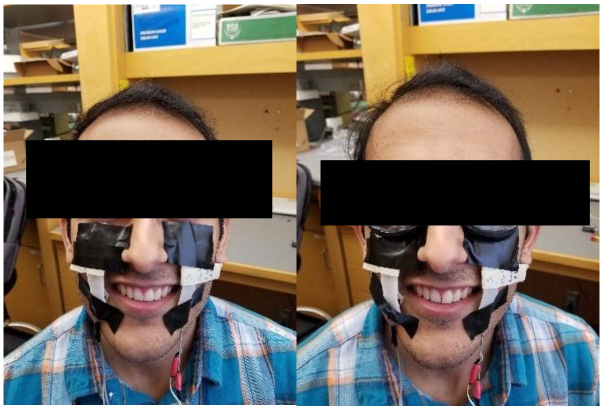

[redacted]
[redacted]

Figure 34 - Subject 3 with taped sensors and goggles

The data in this section are from the third subject. He underwent the same testing procedure as the first two test subjects. However, for this subject the code was further improved prior to testing. For these tests, the StrikeRecord function was altered to acquire data during both the high and low portions of the square wave sent to the striker, providing a longer signal. I deemed this must be necessary to recording the muscle responses after analyzing subject 2's continuous eye movement and comparing it to the more erratic results of the previous exams. The gap introduced in the code for subject 2's tests was also included in the following tests. These improvements are further explained in the discussion section. The following graph is the baseline FMG signal from subject 3. It was requested that the subject did not to blink or move his eyes for the duration of the test. Analyzing the results thus far, it appears that FSR1 is more prone to random fluctuations, compared to FSR2, this is especially noticeable in the baseline tests, as seen below.

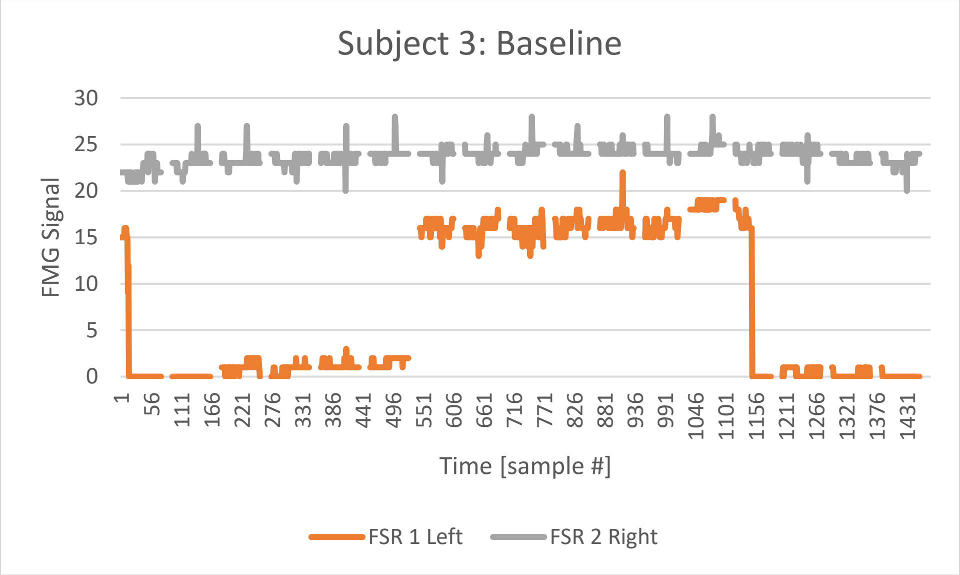


Figure 35 - Subject 3, baseline test, no eye movement. Using new data acquisition code.

The following two graphs are of the subject moving his eyes vertically. Like subjects 1 and 2, there is great variation in these results, compared to those of the baseline. Comparing the ranges of FSR1 and FSR2, we can see that the signal from FSR2 moves both above and below its baseline. While FSR1 stays mostly above, if we consider its baseline to be at 0. This is important because FMG signals are biphasic, thus the sensors need to have a proper base pressure applied to them to show a decrease in pressure if the muscle retracts from the sensor during contraction/relaxation. This is achieved in the FSR2 sensor but not the FSR1 sensor.

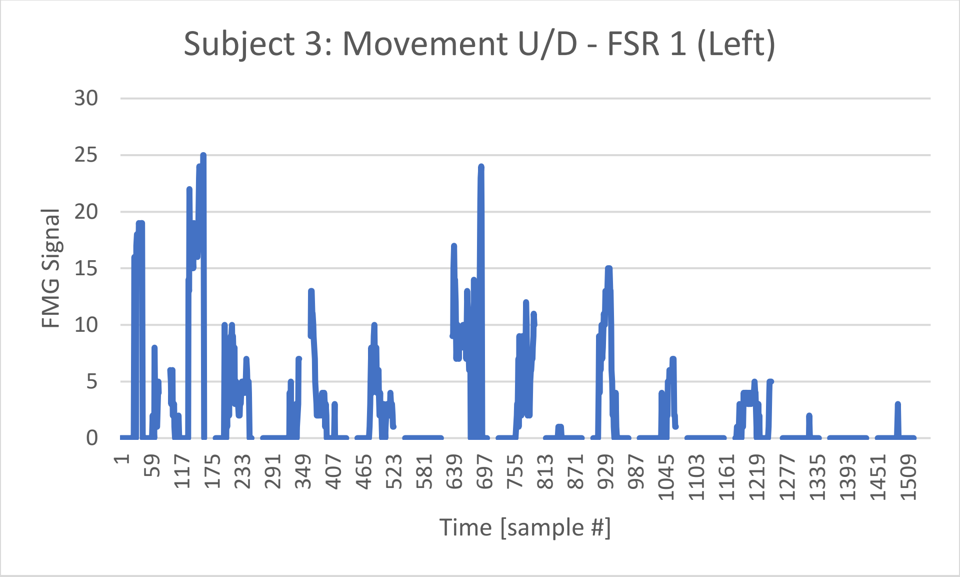


Figure 36 - Subject 3, purposeful eye movement, vertical, from FSR 1. Using new data acquisition code.

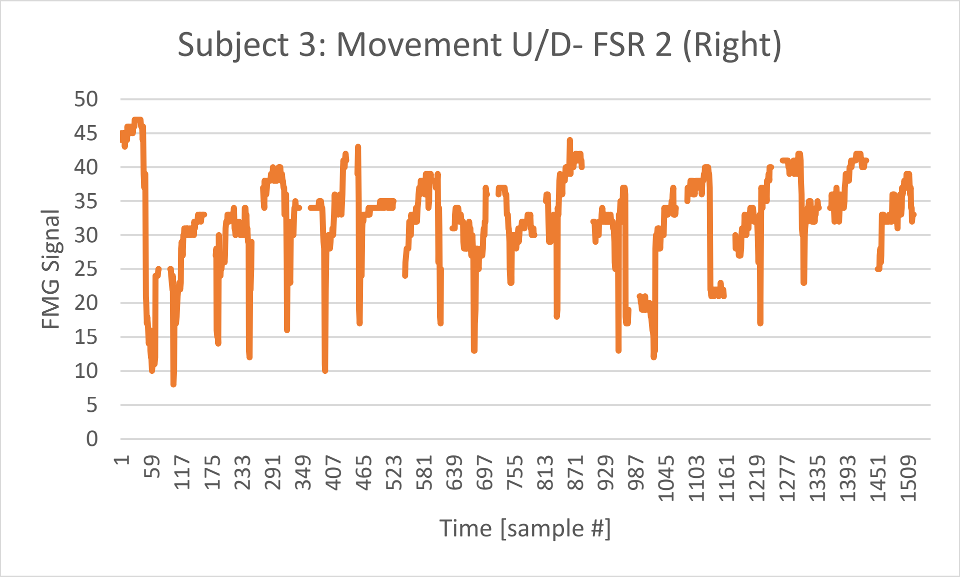


Figure 37 - Subject 3, purposeful eye movement, vertical, from FSR 2. Using new data acquisition code.

An additional eye movement test was taken from this patient, and the results are shown below. For this test, the subject was asked to move his eyes in coordination with the strikes. This allows for better data capture of the eye movement as the recording takes place post strike firing. This showed clear results of the eye movement on the FMG sensors, as opposed to the irregular variations seen in the previous graphs. This result is similar to that of subject 2, when her eye movement was recorded continuously for 10 seconds.

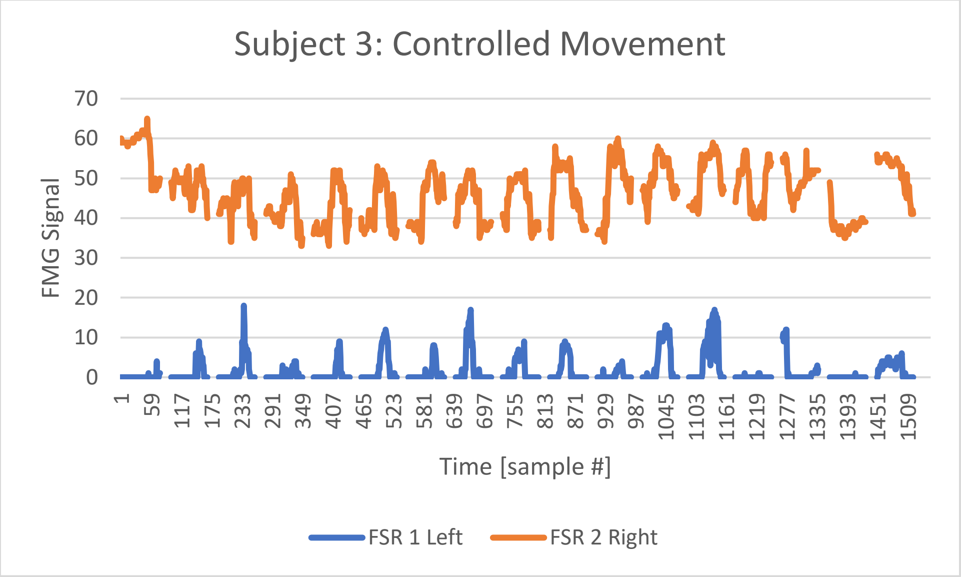


Figure 38 - Subject 3, purposeful eye movement, vertical at point when striker fires. Using new data acquisition code.

This last graph contains the results of strike administration to the subject. As with the previous subjects, there is little variation in the data. This makes it clear that the sensors are detecting very little ocular muscle activity during strike administration. This implies that either the FMG sensors cannot detect the VOR response or that the induction of oVEMPs did not occur.

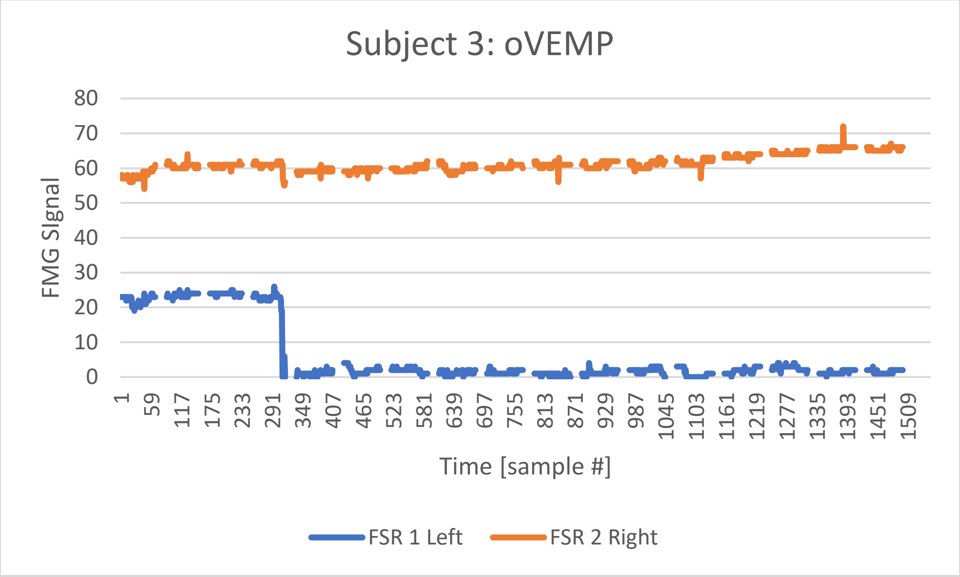


Figure 39 - Subject 3, strike administration test. Using new data acquisition code.

Striker Test

A test to measure the striker force was also completed. In this test, one of the FMG sensors (FSR2) was attached to the tip of the striker. The striker was then placed against a wall with a weight placed behind it to prevent it from moving. For this test, the threshold requirement from the internal FSR was removed so that no external pressure needed to be applied to the FMG sensor or the internal FSR sensor prior to firing. The striker was then fired and the FMG signal was recorded. The results are below.

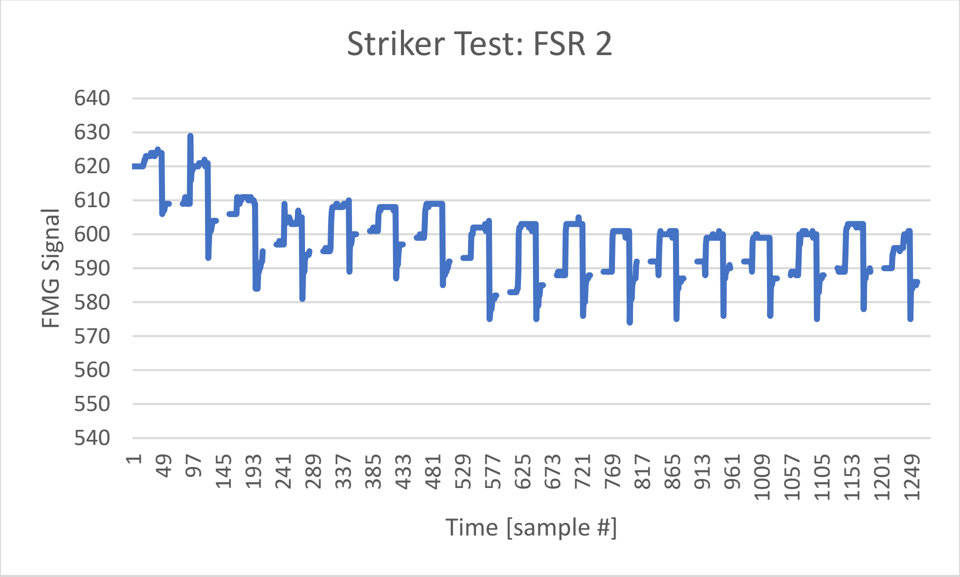


Figure 40 - Striker force test results

From the results, it appears that the force applied by the striker was consistent. The small decline in the signal is attributed to the striker moving away from the wall, as the weight holding it in place was not heavy enough to prevent it from moving completely. The value at the sensor is dependent on the pressure applied to the sensor before the test; however, the change in the force is consistent and represents the force applied by the striker.

FSR and Striker Testing

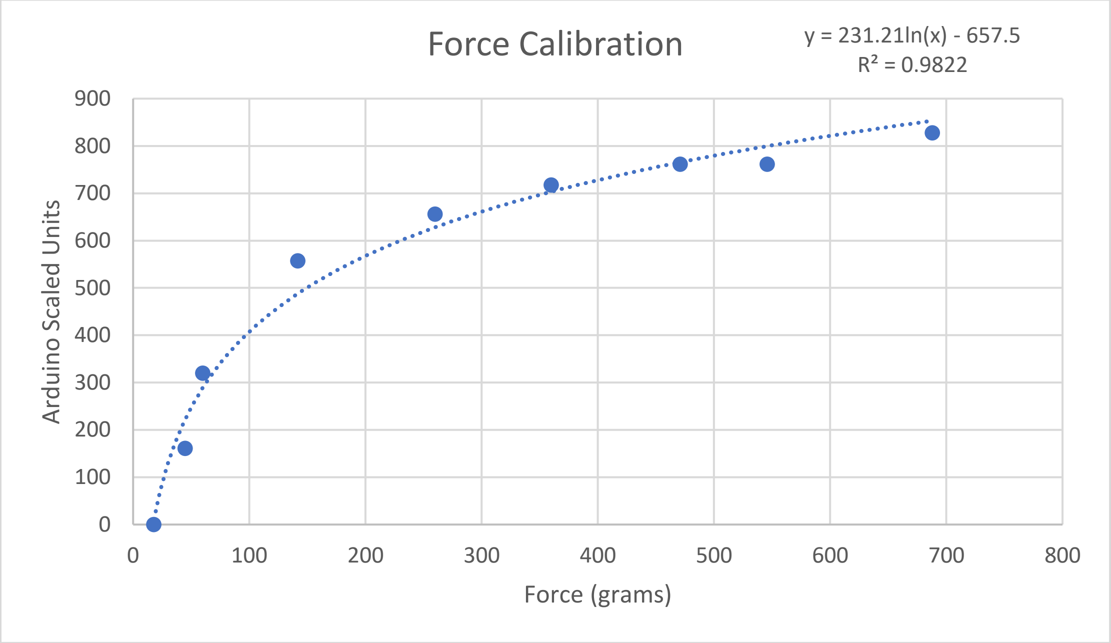


Figure 41 - FSR sensor reading calibrated against forced applied and measured by the EXTECH Instruments FG-5000 Force Gauge. Note the nonlinearity of the FSR signal. The y -axis is in Arduino scaled units representing voltage where 1023 units = 3.3V and 0 = 0V. The x-axis is in grams and represents force.

The graph above shows data from the calibration of a force sensitive resistor. This allows us to determine the force in grams corresponding to the value received at the Arduino in scaled units (0 – 1023). Force data was collected using the EXTECH Instruments FG-5000 Force Gauge. The force gauge was used to apply pressure to the FSR and the value of the FSR was recorded for 10 seconds to the Arduino. The recorded signal was then averaged and graphed along with the corresponding force value in grams. Varying amounts of pressure were applied to the sensor to create the calibration curve above. A logarithmic trendline was generated and its equation is located at the top right corner of the graph, where x is force and y is Arduino scaled units. It is clear from the values displayed and the trend line that the force measured by the sensors is not linear in nature, as was expected due to previous testing of FSR sensors (Shain, 2009). This means the force applied does not directly correlate with the measurement at the sensors. The values recorded must be modified prior to analyzing the data to achieve a better understand of how much force is generated from the contraction and relaxation of the ocular muscles. It is important to take note that at low amounts of force, the value detected by the sensor is still at 0 units. This means that if the force applied is not above a minimum threshold, that the sensor will not detect a response, even if one occurs.

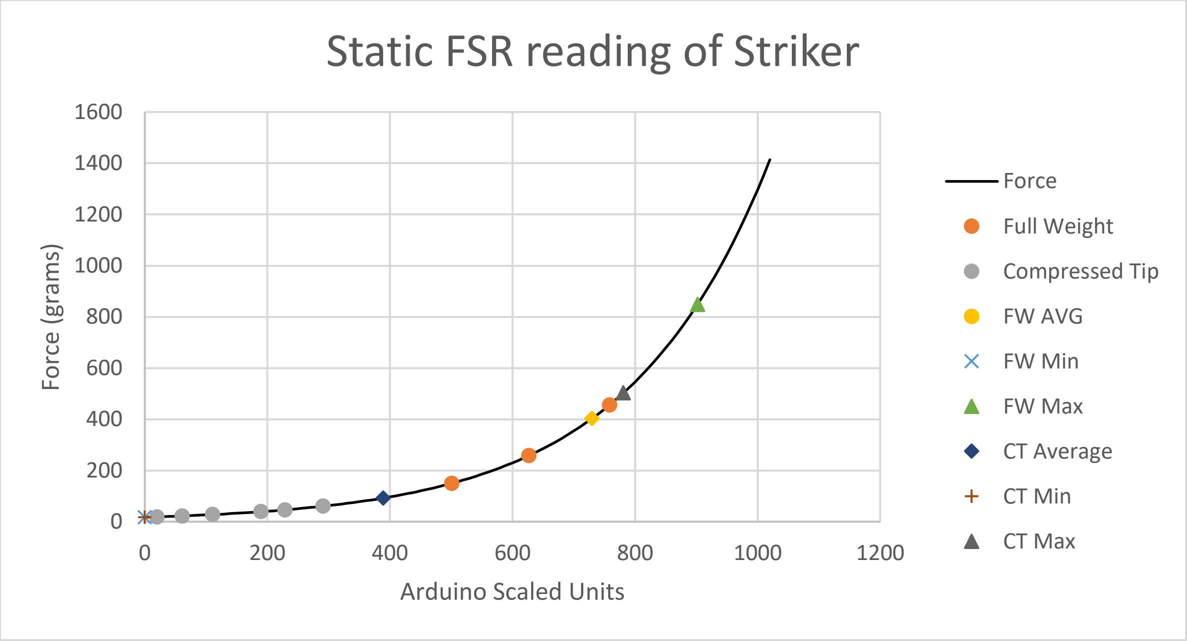


Figure 42 – Static FSR readings with varying striker tip compression. The striker was positioned multiple times in the vice grip until the tip was fully compressed, the FSR signals were recorded and averaged, the corresponding forces were calculated. Strikes were administered at this position and the average, max, and min force were calculated. The same tests were given when the full weight of the striker was applied to the FSR. Data was collect from the FSR sensor in Arduino scaled units (1023 = 3.3V, 0 = 0V) and converted to force using the equation from the Force Calibration curve.

Static FSR data was collected to measure the amount of Force applied by the striker. For this test the FSR was taped flat to the table and the striker was positioned vertically above while being held in a vice grip. The striker was lowered until its striking tip was fully compressed but not allowing its entire weight to be applied to the FSR sensor. The signal at the FSR was collected for a period of 10 seconds and averaged. The striker was repositioned multiple times to measure the varying amounts of pressure needed to compress the tip. The grey markers on the figure above are the averaged values recorded from the FSR to compress the tip, these range from 18 to 60 grams of force. Strikes were administered from this compressed tip position and the signal was converted into force. The average force detected was 92 grams, with a min of 17 grams and max of 502 grams during strike administration. Static measurements while the full weight of the striker was placed on the FSR were recorded. The average values of the FSR reading during repeated positioning were 149 grams, 258 grams, and 456 grams. Strike administration was taken during this position as well with average force of 729 grams, a min of 17 grams, and a max of 456 grams. These force measurements were generated from the equation of the force calibration curve equation. It is important to remember as stated above that a minimum threshold of force needs to be applied to the sensor to receive a reading. The sensor reading a value of 0 units corresponds to 17 grams due to the equation generated in the force calibration chart and thus is not an accurate measurement of the force detected.

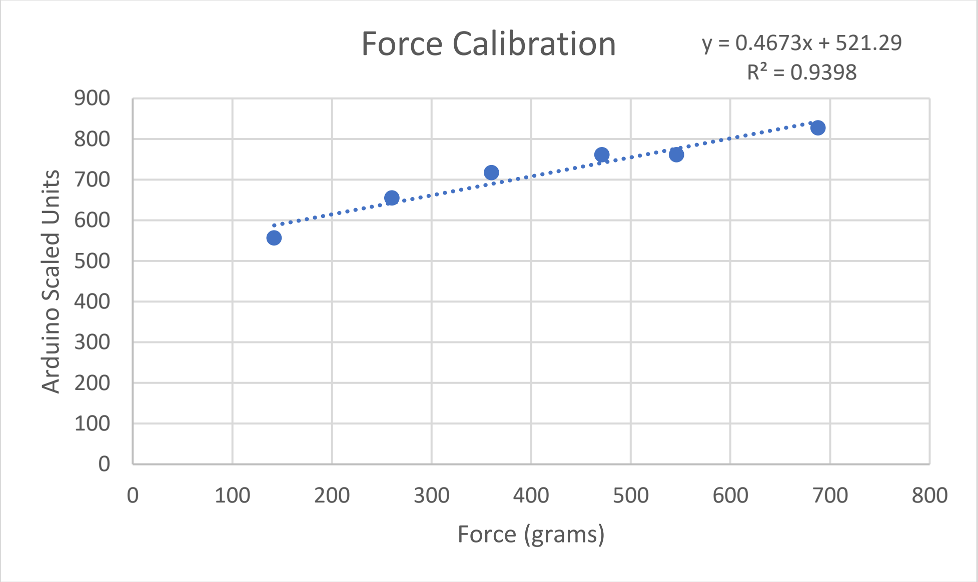


Figure 43 - Force calibration of force measurements over 100 grams and its linear trendline. Force data was collected from the EXTECH Instruments FG-5000 force gauge and correlated with an average FSR signal that collected over 10 seconds. The x – axis represents force in grams. The y – axis represents voltage in Arduino scaled units, where 1023 = 3.3V and 0 = 0V.

Signals from the FSR tend to become more linear as greater force is applied. The above graph takes only the data points over 100 grams of force and generates a linear regression, the equation to which is in the top right corner of the graph. This trend can be used to correlate higher values of FSR readings to force in grams as lower values of force give a more nonlinear relation between signals recorded and actual force applied.

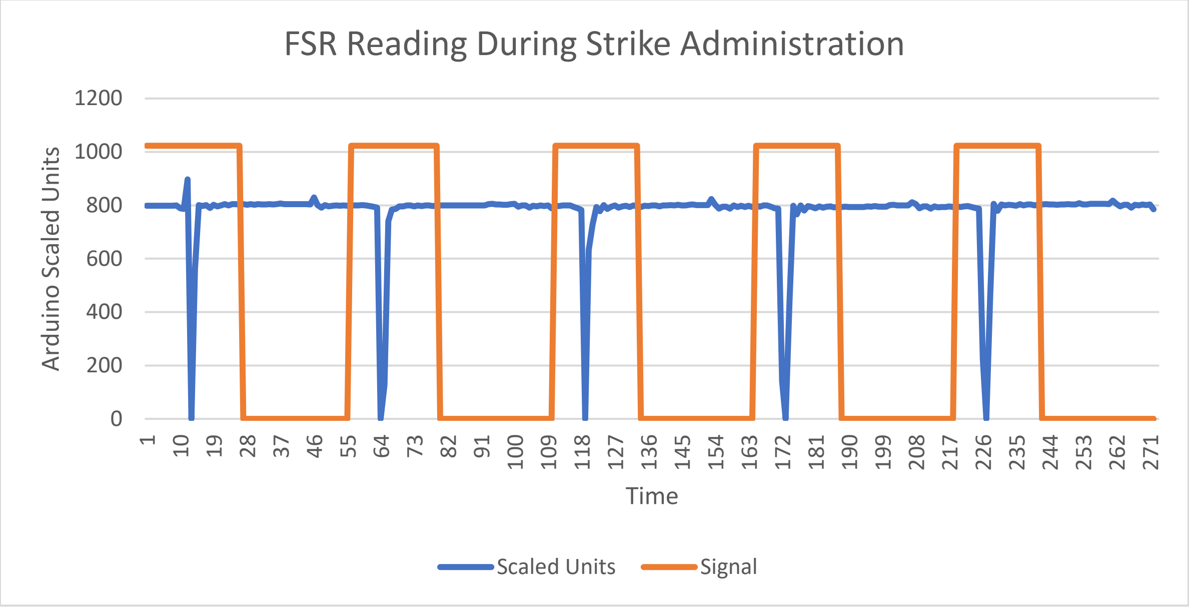


Figure 44 - FSR signal during strike administration over a period of 2.5 seconds at a frequency of 2 Hz for 5 strikes and the striking signal. Data was collected at 108 samples per second. The x- axis is in .01 seconds. The y -axis in Arduino scaled units with 1023 units representing 3.3V and 0 representing 0V.

The strike delivered from the striker is an impulse similar to how a reflex hammer delivers force. The striker was again placed in the vice grips and allowed to apply its full weight to the FSR sensor. The striker was fired for 2.5 seconds at a frequency of 2 Hz for 5 strikes. Data was collected at a calculated sampling frequency of 108 samples per second. The image above demonstrates the force collected in Arduino Scaled Units from the striker. The impulse generated from the striker is clear, however, the force generated is not. This is because the FSR receives too great a force than it can handle and becomes saturated. This makes it difficult to determine the actual force applied by the striker because no matter how much force is applied the signal at the FSR will no longer change. This is a limitation due to the size of the FSR and further testing to determine the actual amount of force applied must be determined.

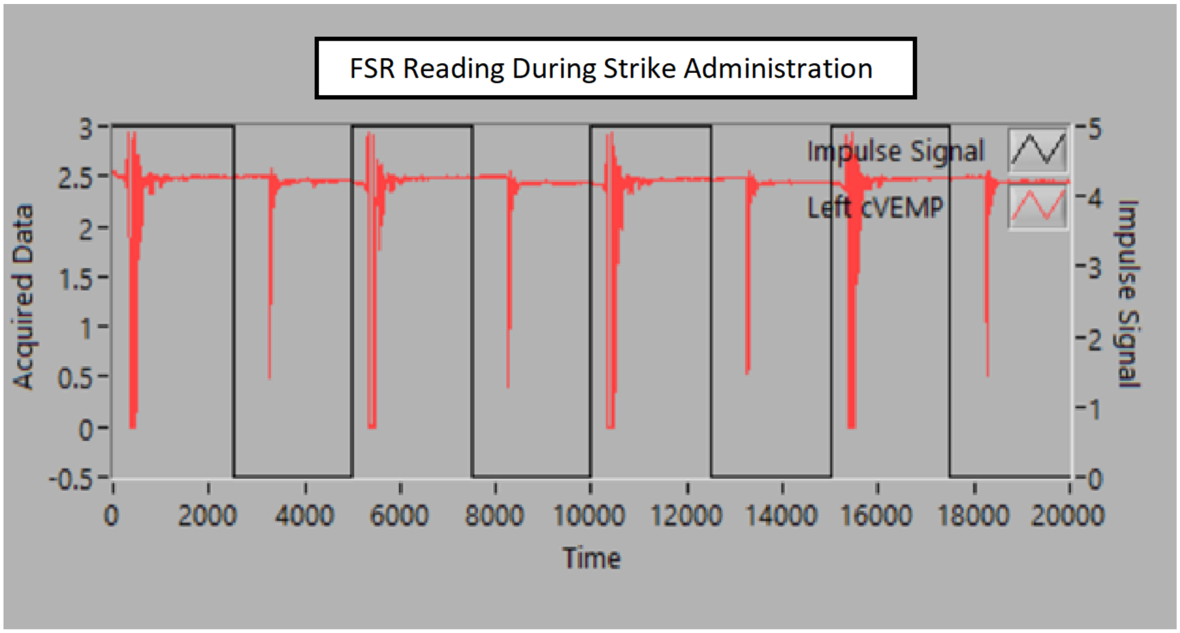


Figure 45 – FSR reading during striking administration collected from the StrikeVEMP LabView program. Striker was fired at a frequency of 2 Hz for 2 seconds for a total of 4 strikes. A total of 10000 data points was collected over 2 seconds. The FSR signal is the voltage loaded over a fixed 10k Ohm resistor in a voltage divider circuit. The y – axis is voltage recorded from the FMG sensor while the x-axis is time in 0.1 millisecond intervals

The same test was repeated using the StrikeVEMP LabView program to allow for a higher resolution reading of the FSR values during strike administration, see image above. The striker was fired at a frequency of 2 Hz for 2 seconds for a total of 4 strikes. As opposed to the Arduino Pro Mini, the LabView can collect data at a rate of 10000 samples per second. This increased resolution allows us to see the slight variation and spring like nature of signal that the FSR is detecting during strike administration. The first peak occurs within approximately 500 milliseconds after the high signal is sent to the striker. This means the firing of the strike has a latency, due to the generation of the magnetic field needed to propel the metal pin in the striker. A secondary peak, with a latency of approximately 750 milliseconds after the low signal is sent is due to the spring releasing and retracting the metal pin in the striker. However, the same saturation effect of the FSR is seen and thus the force cannot be properly measured, as it is passed the ability of the FSR to measure.

DISCUSSION:

<table>
<tr><th colspan="7">AVETOR Device Specifications and Accomplishments</th></tr>
<tr><th>Spec. #</th><th>Specification Description</th><th>Unit of Measure</th><th>Marginal Value</th><th>Ideal Value</th><th>Achieved ?</th><th>Achieved Value</th></tr>
<tr><td><b>1</b></td><td>Striking frequency</td><td>Hertz</td><td>1 – 10 Hz</td><td>2 Hz</td><td>Yes</td><td>2 Hz</td></tr>
<tr><td><b>2</b></td><td>Striking duration</td><td>Seconds</td><td>Variable</td><td>8 seconds</td><td>Yes</td><td>8 seconds</td></tr>
<tr><td><b>3</b></td><td>Threshold Condition: Striker only fires when proper placement and pressure are met</td><td>Scaled units in Arduino</td><td>> 800 units</td><td>1023 (HIGH signal)</td><td>Yes</td><td>>1000</td></tr>
<tr><td><b>4</b></td><td>File Management: Save data from 2 sensors in new file after each run</td><td># of files</td><td>1- 2 files</td><td>1 file</td><td>Yes</td><td>1 file</td></tr>
<tr><td><b>5</b></td><td>Continuous signal recording during strike administration</td><td>Seconds</td><td>Variable (dependent on striking duration)</td><td>8 seconds</td><td>No</td><td>Split Recording (negligible data loss)</td></tr>
</table>

Figure 46 – AVETOR Device Specifications and Accomplishments

The goal of this thesis was to develop a prototype that meets the required specifications as listed in the table above. Of the 5 listed specifications, 4 were achieved completely, with the last one only partially achieved. The AVETOR device can fire the StrikeVEMP striker at the required 2 Hz frequency for a duration of 8 seconds, these parameters can also be further modified in the code for any desired frequency and number of strikes. Both the threshold requirement and the initiation button signal are needed to allow the striker to fire. This allows for a consistent and repeatable testing procedure. The AVETOR only fires if it is properly placed at the Fz location of the forehead and enough pressure is applied allowing the internal FSR sensor to reach the designated threshold value. As the reading from the internal FSR is received in scaled Arduino units and not yet calibrated to a force, the marginal value was listed as greater than 800 units and ideally 1023 units (a completely HIGH signal). The goal was to provide enough force at the Fz location and remain clearly above random noise, while still being comfortable for the patient. During testing, it was determined that a threshold of greater than 1000, can accomplish these goals. The AVETOR prototype is also capable of recording data from both FMG sensors and saving the results to one file by separating the values by a comma. The data is saved on the SD card mounted and named as RUNXXX.txt, where the X's represent the file number. AVETOR uses the novel approach of FMG to detect muscles responses, as opposed to the more common method of EMG. From the results, it appears that the ocular muscle responses can be registered by the FMG sensors. This is evident in every purposeful eye movement test, but especially in subject 2's continuous eye movement test and subject 3's controlled eye movement tests. These results both show the repeated vertical movement of the eyes. On the other hand, the VOR responses taken during strike administration do not seem to register above baseline noise. This could be due to a lack of calibration in the FMG sensors or because EMG is required to measure these responses. However, this is not a requirement of the thesis and the sensors were used only to test the functionality of the device. Further testing of both modalities needs to be conducted. Lastly, due to the serial processing nature of Arduino Pro Mini, recording of a continuous signal was not achievable. This drawback led to many improvements in the development of this prototype, which are described in the section below.

Controller Improvements

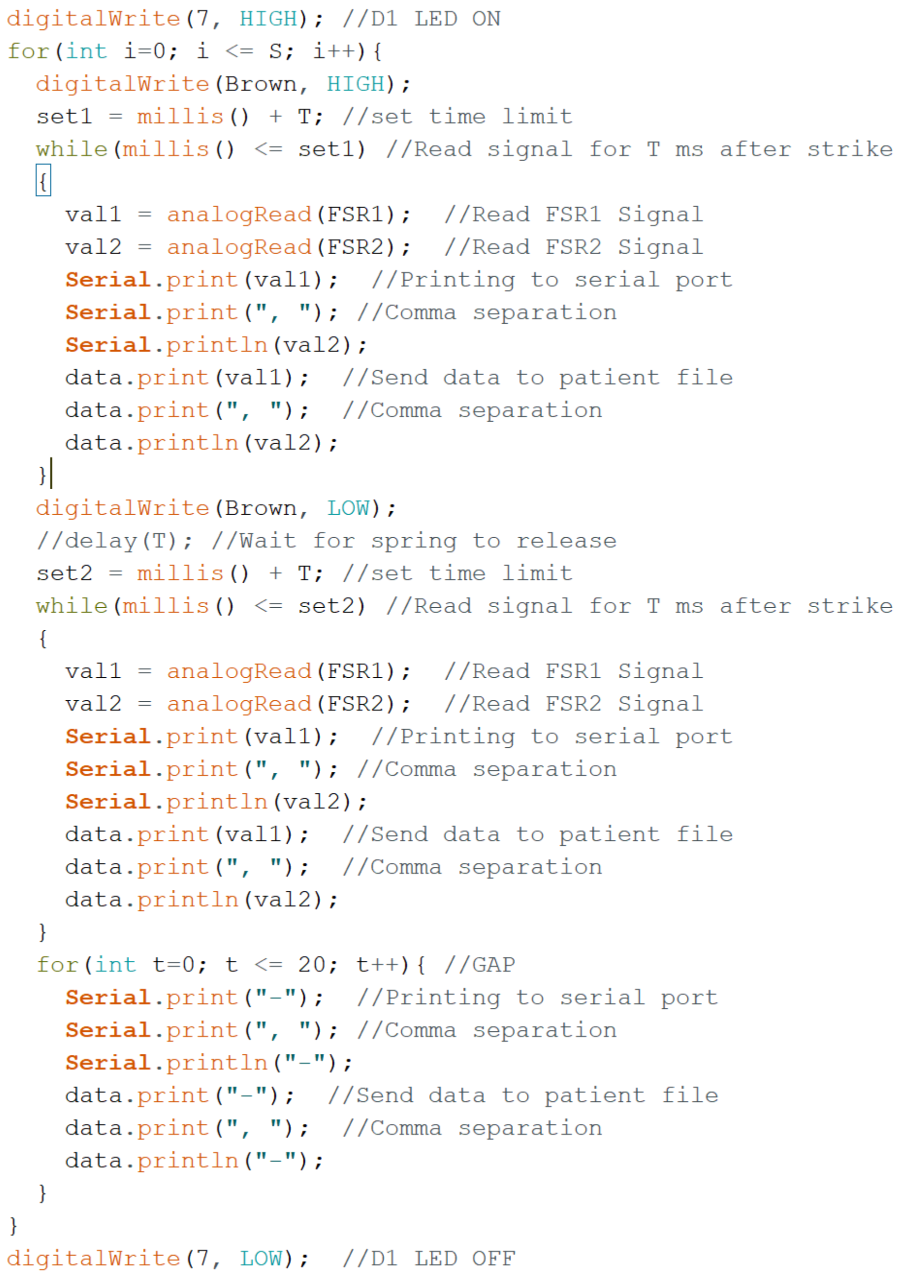
An innate condition of the Arduino Pro Mini is that it has only one processor (ATmega328), thus, it can only execute one line of code at a time and does not allow for two functions to run simultaneously (Halle & Bens, 2008). This method of serial processing works well when sending just a square wave to the striker to fire it or just acquiring an uninterrupted signal of data. However, data acquisition needs to occur immediately after striking, since the VOR response occurs within 100-150 milliseconds, we run into a problem (Craig, 2016; Curthoys et al., 2012). During the development and testing of the device, improvements were made to the accommodate this condition.

The StrikeRecord function was originally 2 separate functions, Striker and Record. The Striker function sent high/low signals to the striker with a delay for half the period between each change in signal. The delay function halts the Arduino processor, preventing it from executing its next command until the specified time has passed. The Record function recorded signals continuously in a while loop for a designated period using the millis function. The millis functions returns the number of millisecond that has passed since the Arduino has been powered on or restarted, it represents an internal clock/counter for the Arduino. Due to Arduino only processing functions serially, and not in parallel, a new method of firing the striker and recording data needed to be developed.

To accomplish this in the code, the square wave was split into two parts. In the first half a high signal was sent to the striker firing it. For the remainder of the high signal, data was collected from the FMG sensors using a loop with a limit set to half the period plus the current time. Data acquisition was then stopped, and a low signal was sent to the striker for the second half of the square wave period using the delay function. This low signal is required to allow the magnetic field to subside and allow the spring to retract the metal pin in the striker. Removing this delay would prevent the striker from firing properly. This was the code used to test subject 1.

Prior to testing subject 2, the code was modified to add a gap in the data acquisition between each strike. This prevents the data from stringing together when it was collected in pieces and only during the high period of the square wave. This allows us to analyze the response from each strike separately.

The code was further modified to remove all delays and acquire data during the low portion of the square wave as well, see figure below. As seen in subject 3's results, this greatly improved data acquisition for the moving eye tests. This firing and data collection method works better than the previous two, however, there is a tendency for the controller to get stuck in the data collection loop. This requires the Arduino to be reset and the test to be administered again. It is possible that it is due to a problem with the breaking condition of the data collection loops. As mentioned, the breaking condition is to check for when current time passes a set limit. This limit is set as the time right before entering the data collection loop plus half the period of the striking frequency. Further testing of the code and the breaking condition must be completed to avoid this problem in the next version of the prototype.



```
digitalWrite(7, HIGH); //D1 LED ON
for(int i=0; i <= S; i++){
  digitalWrite(Brown, HIGH);
  set1 = millis() + T; //set time limit
  while(millis() <= set1) //Read signal for T ms after strike
  {
    val1 = analogRead(FSR1);  //Read FSR1 Signal
    val2 = analogRead(FSR2);  //Read FSR2 Signal
    Serial.print(val1);  //Printing to serial port
    Serial.print(", "); //Comma separation
    Serial.println(val2);
    data.print(val1);  //Send data to patient file
    data.print(", ");  //Comma separation
    data.println(val2);
  }
  digitalWrite(Brown, LOW);
  //delay(T); //Wait for spring to release
  set2 = millis() + T; //set time limit
  while(millis() <= set2) //Read signal for T ms after strike
  {
    val1 = analogRead(FSR1);  //Read FSR1 Signal
    val2 = analogRead(FSR2);  //Read FSR2 Signal
    Serial.print(val1);  //Printing to serial port
    Serial.print(", "); //Comma separation
    Serial.println(val2);
    data.print(val1);  //Send data to patient file
    data.print(", ");  //Comma separation
    data.println(val2);
  }
  for(int t=0; t <= 20; t++){ //GAP
    Serial.print("-");  //Printing to serial port
    Serial.print(", "); //Comma separation
    Serial.println("-");
    data.print("-");  //Send data to patient file
    data.print(", ");  //Comma separation
    data.println("-");
  }
}
digitalWrite(7, LOW);  //D1 LED OFF
```

Figure 42 - Modified portion of the StrikeRecord function, note lack of delay and use of millis in data acquisition and the gap code

Due the split in data collection, a loss of data can still occur between the two data recording loops in the code, although it may be negligible. This method may also disrupt the timing of the strikes and acquisition because the code itself takes time to run, with each line of code taking a different length of time. This would prevent the square wave from having a 50% duty cycle.

A possible solution to this problem would be to use a modality that can run functions simultaneously, but still maintain the portability and compactness of the AVETOR prototype. My suggestion is a cellular phone or tablet. They would both have the processing power capable of handling multiple functions at once. The ports on the phone or tablet could be used to connect to the FMG sensors and receive data, which can also send a signal to fire the striker. This would also eliminate the need for a computer to modify the code on the controller or view the saved files post data acquisition. A user interface on a phone or tablet could allow for easier modification to the parameters of the striker or FMG sensors, as well as allow for real time data analysis. As of now, the saved files need to be transferred to a computer and require secondary software to recreate the signal in graphical form. The circuitry used for signal processing from the Dynabrush board could be integrated into the FMG sensor connector that attaches to the phone or tablet. This method would streamline the process of data acquisition and analysis.

Force Calibration and Striker Characterization

Multiple tests were completed to calibrate the force applied to an FSR sensor to the value received to the Arduino. It was determined that the relationship between force and FSR value were not linear, but instead logarithmic. This is important in understanding how much force is being detected by the sensors as well as how much force the striker is applying. Testing of the striker force application determined that the striker administers force in the form of impulses similar to a reflex hammer. There are in fact two peaks that are measured by the sensor during both the low and high portion of the square wave signal. This was determined during analysis of the data from the StrikeVEMP LabView program as it had higher resolution compared to the Arduino because of its faster sampling rate. This increased resolution allows us to see the slight variation and spring like nature of signal that the FSR is detecting during strike administration. The first peak occurs within approximately 500 milliseconds after the high signal is sent to the striker. This means the firing of the strike has a latency, due to the generation of the magnetic field needed to propel the metal pin in the striker. A secondary peak, with a latency of approximately 750 milliseconds after the low signal is sent is due to the spring releasing and retracting the metal pin in the striker However, the same saturation effect of the FSR is seen and thus the peak force cannot be properly measured, as it is passed the ability of the FSR to measure. This is due to the striker applying more force than the FSR sensor can detect. This means that even if greater force is applied, the signal from the sensor will not change, so determining the peak force by using the FSR sensor is not possible. However, from previous studies, the force applied by the striker is at least 237.3 +/- 3.3 N with a linear acceleration of 1.8 g (Wackym et al., 2012), thus we can assume the force applied to the FSR sensor is at least at this value. Our results showed that the risk of head injury was lower for linear head acceleration than for angular head acceleration, and it was lower for frontal impact than for lateral impact. Many head injury detection system begin measurement at 10 g while our device only delivers 1.8 g (Wackym et al., 2012).

(Wackym et al., 2012).

Signal Detection

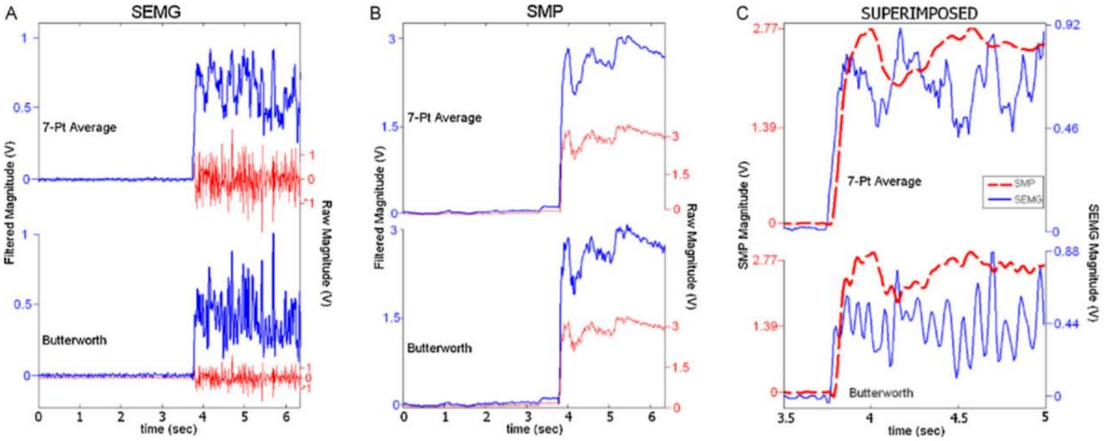


Figure 47 - Comparison of EMG and FMG responses in quadriceps muscle during isometric contraction, (Yungher, Wininger, Barr, Craelius, & Threlkeld, 2011).

In the AVETOR prototype the detection of ocular muscle activity occurs via the novel approach of force myography, rather than the more common electromyography. It is important to understand how FMG functions and the correlation it has to EMG. In EMG, the signal is the electrical activity that the motor neurons send to the muscles causing them to contract. While in FMG, the signal is the surface muscle pressure that is produced from the change in shape of the muscle as it contracts and relaxes. Both signals indicate a change in muscle activity. The image above, from Yungher et al, shows surface EMG and FMG (surface muscle pressure - SMP) signals from the quadriceps muscles during an isometric contraction (Yungher, Wininger, Barr, Craelius, & Threlkeld, 2011). It is clear that the EMG signals are more erratic, and variable compared to FMG signals (Part A versus Part B). However, the image also shows that the FMG signals are consistent with detecting muscle activity like EMG signals (Part C). Since oVEMP responses have small amplitudes, are susceptible to electrical noise and interference from variations in skin thickness (Hecker et al., 2014), I believed that FMG would be a better modality to record these muscle responses as these qualities do not affect their signals. It is important to note that the quadriceps are large muscles and pressure response is much greater than that of the ocular muscles.

Analyzing the data from the results we can see that FMG sensors are able to detect pressure changes due to eye movement. It is clear comparing the baseline no eye movement tests to those of the vertical or horizontal movement tests, that pressure changes at the sensors were detected. The graph of subject 2's vertical eye movement for the continuous 10 seconds showed a clear and consistent change in pressure related to the eye movement. The graph of subject 3's controlled movement had similar results, showing repeatability. We can assume that this pressure change is due to the change in shape of the ocular muscles during eye movement. The change in pressure in all three subjects' tests for eye movement was approximately 25 in the scaled Arduino units. If each scaled unit is 3.23 mV (3.3V/1023 units), that converts the pressure change to 83 mV at the FSR. This relates to a very small applied pressure, which is expected due to the small size of the ocular muscles.

Analyzing the muscle responses from the strike administration trials, we see results more closely resembling the baseline no movement FMGs. This means that very little to no pressure changes were detect during these tests. This could be due to a lack of induction of VEMPs, insensitivity of the sensors to these VEMPs, or both. The image below is a depiction of the cVEMP (left) and oVEMP (right) EMG responses taken from Chiarovano et al (Chiarovano, Darlington, Vidal, Lamas, & de Waele, 2014). These EMG signals are much smaller and at a lower frequency compared to the EMG signals of the quadriceps contraction above. During vibration induced eye movements, like the VOR response, the eyes moves less than 0.5 degrees (Curthoys et al., 2012). This small movement may not produce enough of a muscle contraction to provide a change in pressure at the FMG sensors. While in the purposeful movement trials the subjects moved their eyes as far as possible in either direction. It is possible that the FMG sensors cannot detect the minute ocular muscle responses in the oVEMPs, and that EMG is required. As this was a novel approach to detecting VOR responses, it is not a requirement of the device, but rather a check for feasibility.

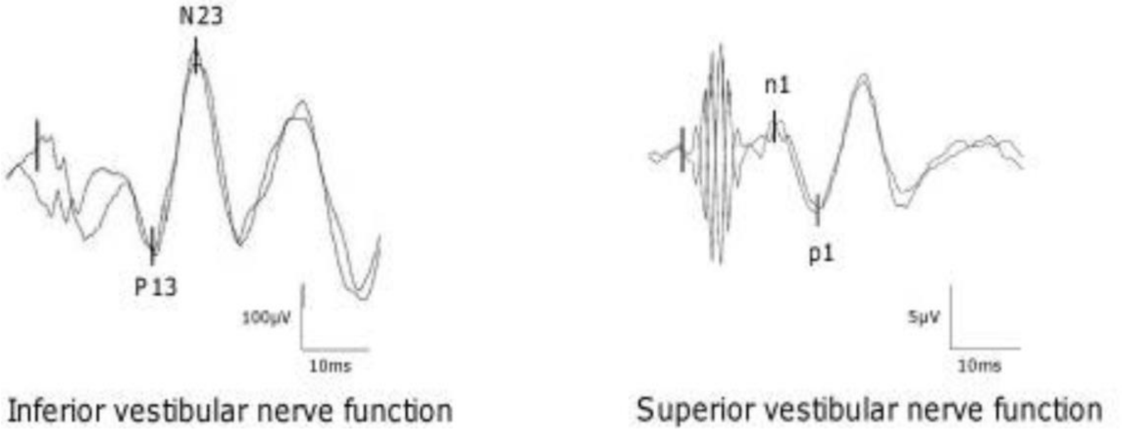


Figure 48 - Normal oVEMP responses of a healthy individual, (Chiarovano, Darlington, Vidal, Lamas, & de Waele, 2014)

However, if the FMG modality will continue to be used for VOR response measurement, the a few concerns must be addressed. Firstly, the lack of calibration makes the analysis of the signal quite difficult. Any changes in the base pressure can greatly change the results, as was seen in subject 1 and subject 3 in the no movement tests. This lack of calibration can even be seen between the two sensors on an individual patient, as with subject 2, where one sensor was reading a base of 50 units and the other a base of 20 units. Calibration of the sensors must be done prior to further testing. This requires a baseline and maximum pressure to be applied to the sensors and then related to the output received at the Arduino. Also, gradual addition of pressure at the FSR and recording the coinciding value can help create a calibration curve (Wininger, Kim, & Craelius, 2008).

This would help understand the amount of pressure the ocular muscles are applying to the sensor. Another concern is the lack of enough base pressure, as seen with FSR1, this prevents suitable collection of data as a decrease in pressure would not register. Appropriate placement of the sensor and goggles that apply pressure at the correct location would solve this problem.

CONCLUSION:

The purpose of this thesis was to describe the design, fabrication, and testing of the AVETOR prototype. This device consists of the StrikeVEMP striker to stimulate the vestibular system, 2 FMG sensors to measure the muscle activity, and a controller to record and save the responses via Arduino/Dynabrush board. The goals of this thesis were to develop a prototype capable of firing the StrikeVEMP striker at a 2 Hz frequency for 8 seconds, have the device record a continuous FMG signal, and the save the data onto a mounted SD card.

The results determined that the device could indeed fire the StrikeVEMP striker at the required specifications and record the FMG signal to a single file onto the SD card, creating a new file for each run of the test. The FMG sensors were able to detect a change in pressure due to purposeful ocular movement of approximately 83 mV at the FSR or 25 scaled units in Arduino. However, the FMG sensors were not able to detect any significant VOR response in pressure during strike administration. As this is a novel approach additional testing of this modality needs to be completed. Due to the serial processing nature of the Arduino Pro Mini, the AVETOR device is not able to record a continuous signal, but instead split signal with negligible data loss. Further research must be conducted to improve upon the shortcoming of this device. Through this research and possible improvements lined out in this thesis, the goal of developing a portable and cost effective diagnostic tool to elicit and measure VOR responses may be achievable.

REFERENCES:

Agrawal, Y., Ward, B. K., & Minor, L. B. (2013). Vestibular dysfunction: prevalence, impact and need for targeted treatment. *J Vestib Res, 23*(3), 113-117. doi:10.3233/VES-130498

Amin, M. S. (2016, Feb 10, 2016). Vesitbuloocular Reflex Testing.

Colebatch, J. G. (2012). Mapping the vestibular evoked myogenic potential (VEMP). *J Vestib Res, 22*(1), 27-32. doi:10.3233/VES-2011-0438

Chiarovano, E., Darlington, C., Vidal, P. P., Lamas, G., & de Waele, C. (2014). The role of cervical and ocular vestibular evoked myogenic potentials in the assessment of patients with vestibular schwannomas. *PLoS One, 9*(8), e105026. doi:10.1371/journal.pone.0105026

Craig, J. (2016). Vesitbular Evoked Myogenic Potentials (VEMP): How Do I Get Started. *Balance/Dizziness/Vestibualar Issues.* Retrieved from https://www.audiologyonline.com/articles/vestibular-evoked-myogenic-potentials-vemp-16713

Curthoys, I. S., Vulovic, V., & Manzari, L. (2012). Ocular vestibular-evoked myogenic potential (oVEMP) to test utricular function: neural and oculomotor evidence. *Acta Otorhinolaryngol Ital, 32*(1), 41-45.

Fetter, M. (2007). Vestibulo-ocular reflex. *Dev Ophthalmol, 40*, 35-51. doi:10.1159/000100348

Gray, L. Vestibular System: Structure and FunctionNeuroscience Online: UTHealth: McGovern Medical School.

Guest. (2012, May 26, 2012). Topic: Open a new file on an SD card everytime Arduino reboot. *Programming Questions.* Retrieved from http://forum.arduino.cc/index.php?topic=107324.0

Egami, N., Ushio, M., Yamasoba, T., Yamaguchi, T., Murofushi, T., & Iwasaki, S. (2013). The diagnostic value of vestibular evoked myogenic potentials in patients with Meniere's disease. *J Vestib Res, 23*(4-5), 249-257. doi:10.3233/VES-130484

Farrell, L., & Rine, R. M. (2014). Differences in Symptoms among Adults with Canal versus Otolith Vestibular Dysfunction: A Preliminary Report. *ISRN Rehabilitation, 2014*, 10. doi:10.1155/2014/629049

Hain, T. C. (2018). Ocular Vesitbular Evoked Myogenic Potential (oVEMP) Testing. Retrieved from https://www.dizziness-and-balance.com/testing/VEMP/ovemp.html

Halle, & Bens. (2008, 2008). Topic: Can't figure out how to run two functions at once Retrieved from http://forum.arduino.cc/index.php?topic=41079.0

Hecker, D. J., Lohscheller, J., Schorn, B., Koch, K. P., Schick, B., & Dlugaiczyk, J. (2014). Electromotive Triggering and Single Sweep Analysis of Vestibular Evoked Myogenic Potentials (VEMPs). *IEEE Trans Neural Syst Rehabil Eng, 22*(1), 158-167. doi:10.1109/TNSRE.2013.2252627

Khan, S., & Chang, R. (2013). Anatomy of the vestibular system: a review. *NeuroRehabilitation, 32*(3), 437-443. doi:10.3233/NRE-130866

Noohi, F., Kinnaird, C., DeDios, Y., Kofman, I. S., Wood, S., Bloomberg, J., . . . Seidler, R. (2017). Functional Brain Activation in Response to a Clinical Vestibular Test Correlates with Balance. *Front Syst Neurosci, 11*, 11. doi:10.3389/fnsys.2017.00011

Shain, A. H. (2009). *CHARACTERIZATION OF THE FLEXOR DIGITORUM SUPERFICIALIS AS A PREDICTOR OF GRASPING STRENGTH.* (Master of Science), Rutgers University.

Swenson, R. (2006). Vestibular SystemReview of Clinical and Functional Neuroscience: Darthmouth Medical School.

Vestibular System: Anatomy. (1997). Encyclopedia Britannica Inc.

Wackym, P. A. (2015). StrikeVEMP: Automated Portable Bone-Conducted Force Stimuli for Recording Acceleration-Produced coVEMPs Portland,OR: National Institute of Health.

Wackym, P. A., Ratigan, J. A., Birck, J. D., Johnson, S. H., Doornink, J., Bottlang, M., . . . Black, F. O. (2012). Rapid cVEMP and oVEMP responses elicited by a novel head striker and recording device. *Otol Neurotol, 33*(8), 1392-1400. doi:10.1097/MAO.0b013e318268d234

Watson, M. A., Black, O., & Crowson, M. (2018). About Vestibular Disorders. *The Human Balance System.* Retrieved from http://vestibular.org/understanding-vestibular-disorder

Wininger, M., Kim, N. H., & Craelius, W. (2008). Pressure signature of forearm as predictor of grip force. *J Rehabil Res Dev, 45*(6), 883-892.

Yacovino, D. A., Hain, T. C., & Musazzi, M. (2017). Fluctuating Vestibulo-Ocular Reflex in Meniere's Disease. *Otol Neurotol, 38*(2), 244-247. doi:10.1097/MAO.0000000000001298

Yungher, D. A., Wininger, M. T., Barr, J. B., Craelius, W., & Threlkeld, A. J. (2011). Surface muscle pressure as a measure of active and passive behavior of muscles during gait. *Med Eng Phys, 33*(4), 464-471. doi:10.1016/j.medengphy.2010.11.012

Zalewski, C. K. (2015). Aging of the Human Vestibular System. *Semin Hear, 36*(3), 175-196. doi:10.1055/s-0035-1555120