POSTURE GUIDANCE CHAIR

By

Alex Shen

Emre Ulusoy

Pablo Corral Vila

Final Report for ECE 445, Senior Design, Spring 2018

TA: Kexin Hui

02 May 2018

Project No. 58

Abstract

The posture guidance chair consists of a system of sensors mounted on a chair with haptic feedback. The system collects data using eight force sensitive resistors and a distance sensor, and the haptic feedback is provided by four vibration motors. The sensor data is gathered by a microcontroller and sent to the classification software on a nearby computer through Bluetooth. The classification software uses binary and multi-class Support Vector Classification to determine if the use is exhibiting poor posture and how to correct it with 98% and 95% accuracy, respectively. The user can interact with the system in real-time through a Graphical User Interface (GUI) to customize the classifier decisions and view pressure readings from the sensors. The system can be calibrated accurately to users weighing up to approximately 200 pounds. The chair system can be powered with four AA batteries for an average of 20 hours of continuous use.

Contents

1. Introduction

1.1 Background

According to the American Chiropractic Association, back pain is the leading cause of disability worldwide, where it is estimated that 31 million Americans experience back pain [1]. As a potential cause of back pain, poor body positioning for prolonged periods of time adds stress to the muscles and bones [2]. It's possible to adopt bad posture subconsciously which can be effectively countered by maintaining awareness and repeatedly countering the bad posture [4]. As technology advances and more jobs require sitting for long periods of time, it's becoming more important to develop safe methods that maintain the health and well-being of users [3]. Thus, a product that can aid in the correction of posture and isn't physically intrusive and disruptive is a desirable solution.

When it comes to deciding whether a posture is good or bad, the distinction can be quite difficult to make. Figure 1 shows the differences between four common postures. In the context of this project, we consider flat, long lordosis, and short lordosis positions to be generally acceptable with preference for short lordosis as mentioned in [7].

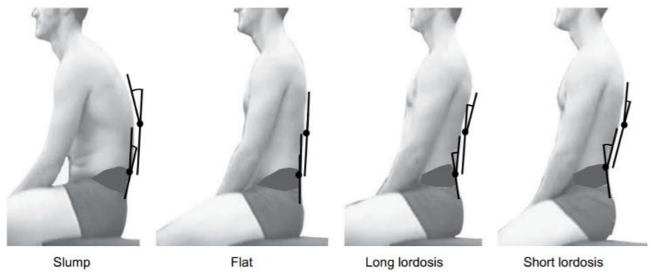

Figure 1 Four common sitting postures. Adapted from [7].

Along with the variations in back angles, an illustration of some acceptable sitting angles is shown in figure 2. While the back position may be the same, the pressure distribution can be drastically different.

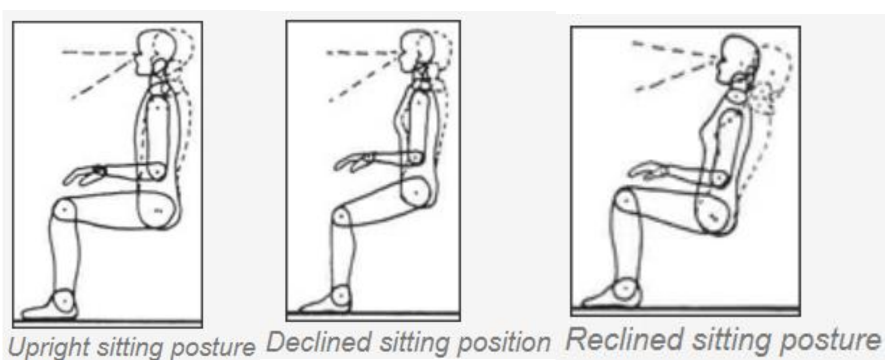

Figure 2 Three good working positions according to OSHA. Adapted from [6].

1.2 Functionality

We created a system of sensors embedded on a chair that can detect poor posture and notify the user of the presence of poor body positioning. Eight pressure sensors on the seat and the back of the chair recognize the weight distribution of the user, and a distance sensor captures the presence of the user and helps detect the mid-upper spinal curvature. The sensor data is collected by a microcontroller and sent to a computer through Bluetooth and processed. This usage of wireless transmission enables mobility of the user in the chair. The processing software uses the sensor data to classify the user's posture according to Support Vector Classification models configured with 98% accuracy for good/bad classification and 95% accuracy for spatial motor activation classification. These models have been trained on various forms of the postures in figures 1 and 2 with success. The interface of the system includes a visualization of sensor data as well as haptic feedback as directed by classification result. The pressure visualization allows the user to gain insight on their posture, and the haptic feedback, from vibration motors, guides the user away from poor posture. This system addresses the problem of prolonged poor sitting posture since it notifies and educates the user about the risk of the user's positions. An important feature of the system is that there is a calibration mechanism that can generalize the results of the system to users of various weights and preferences. To emphasize the non-intrusive and non-disruptive characteristics, the system allows mobility of the chair of up to 30 feet from the computer and can an average of 20 hours with 4 AA batteries.

1.3 Subsystem Overview

The chair system will need six modules: power supply, sensors, control system, data stream, user interface, and software. Figure 3 shows the block diagram of how the six modules interact with each other. The power supply block provides power to all hardware components using 6V of battery voltage and steps it down into the 3.3V and 5V needed for other components. The sensors block is responsible for providing the readings of the user's posture. The controller system brings the sensor data together and uses the data stream block to communicate each sample to the computer software wirelessly, and it controls the activation of vibration motors in the user interface. From there, the software processes the data and displays the real-time data in the user interface block GUI. Vibration activation data is then sent from the software to the data stream for the control system to process.

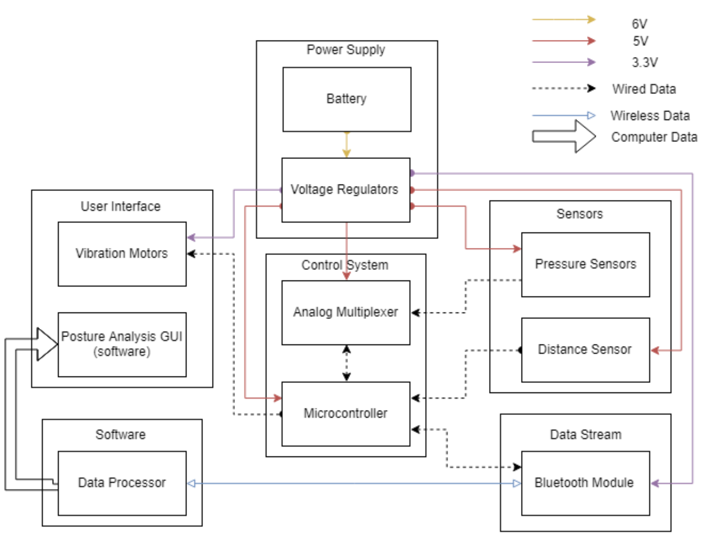

Figure 3 Block Diagram of the system. The modules are Power Supply, Sensors, Data Stream, Control System, User Interface and Software.

2 Design

Figure 4 shows the rough physical layout of the system. The pressure sensors on the seat attempt to record pressure changes in the areas near the knee and under the thigh, ischial tuberosities, lumbar, and shoulders. These sensor placements were found to be the most important and distinguishable areas for detecting different postures [8]. The distance sensor is mounted on the back, where it will sense the presence of a user and help in detecting the distance from the user's spine. The distance detected from the user's spine will help determine if the user is slouching along with the changes in weight distribution. A single distance sensor is used because the user is expected to sit directly in front of the sensor and more sensors will interfere with each other. The rest of the components are mounted behind the chair to avoid physical damage.

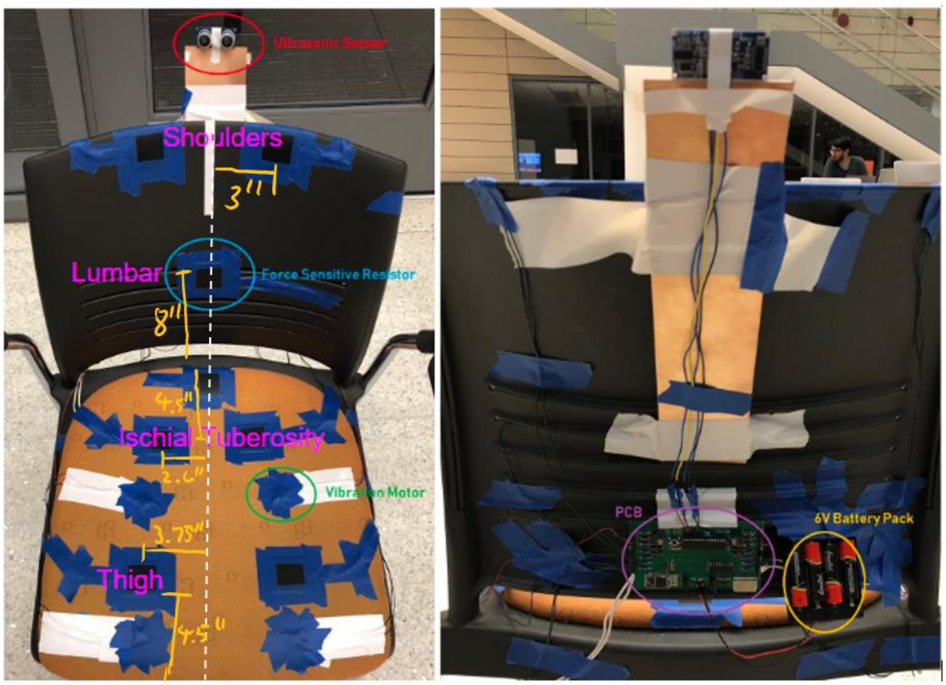

Figure 4 Layout of the hardware components on the chair. The front is on the left and the back is on the right image Sensor placement measurements are based on the findings from [8].

2.1 Power Supply

2.1.1 Battery

2.1.1.1 Design Procedure

The power for the hardware components comes from 4 AA batteries in series, which yields 6 V. The off-the-shelf batteries are sufficient for the use of the project because the average current draw was measured at ~125 mA. This results in an average total power consumption of ~750 mW. The required voltages are 5 V for the pressure sensors, microcontroller, distance sensor, and analog multiplexer and 3.3 V for the vibration motors and Bluetooth module. These values are quite small, so there's no need to design a specialized battery since the voltage can be regulated to the appropriate levels for the components. The capacity for current draw is approximately 2500 mAh for around 100 mA current [17]. Using equation (1), the amount of time the batteries can supply ~125 mA of current is a little under 20 hours. Therefore, the batteries are sufficient for powering the entire system.

$$\text{time} = \frac{\text{capacity}}{\text{discharge}} \tag{1}$$

A design alternative was to use power from a wall outlet. However, the use of a wall outlet would require a power cable attaching the chair to an outlet. This would limit the mobility of the chair and reduce its ability to be moved around a room.

There are no design details for this component because we did not design it.

2.1.2 Voltage Regulators

2.1.2.1 Design Procedure

The voltage regulators have the duty of maintaining the desired voltages from the battery to the 3.3 V and 5 V values. Since there are two distinct voltages, we use two voltage regulators for each voltage level. Although there are many off-the-shelf options that can be used for this component, we decided to make our own basic linear voltage regulators. A naïve implementation the drop the 6 V input from the batteries would be to use a voltage divider circuit. However, this would waste power by dissipating it through a series resistor. A basic linear voltage regulator is implemented because it is significantly easier to analyze and tune compared to more advanced configurations. This decision to implement our own voltage regulators was made to add complexity to the project and make use of the immediately available parts we have access to.

2.1.2.2 Design Details

Each voltage regulator consists of a BJT NPN transistor, a Zener diode, and a resistor. They take advantage of the Zener diode property of approximately constant voltage when it is in breakdown mode.

To calculate the components, we used the fact that the NPN transistor MPSU06 has an emitter-base voltage of 0.6 V [19]. Knowing that and the desired output voltage, we can easily calculate the Zener diode voltage by adding the emitter-base voltage to the voltage output (5.6 V for the 5 V regulator and 3.9 V for the 3.3 V regulator). To calculate the resistor value, we first calculated the voltage across it, which is the amount of voltage to dissipate until it reaches the Zener diode breakdown voltage. Then, using equation (2), the resistance can be computed as a function of expected load current.

$$R_v = \frac{V_{Rmin}}{I_{Dmin} + \frac{I_{Lmax}}{h_{FE}+1}} \tag{2}$$

I_{Dmin} is the minimum current in the Zener diode, which was taken from their datasheets: 20 mA for both 1N5228 and 1N5232 [18]. I_{Lmax} is the maximum load current, which was taken from making calculations of the elements that are going to be supplied by each linear regulator. We concluded that the 5 V regulator, which powers the sensors, the microcontroller, and the analog multiplexer, has a load current of ~120 mA, and the 3.3 V regulator, which powers the Bluetooth module and vibration motors, has a maximum current of ~128 mA. The parameter h_{fe}, which relates the currents through base and collector, is 60 for the MPSU06 transistor [19]. With the defined parameters above, R_v was determined to be 18 Ω for the 5 V regulator and 100 Ω for the 3.3V regulator.

Using the above parts and values, the voltage regulator circuits are constructed according to figure 5.

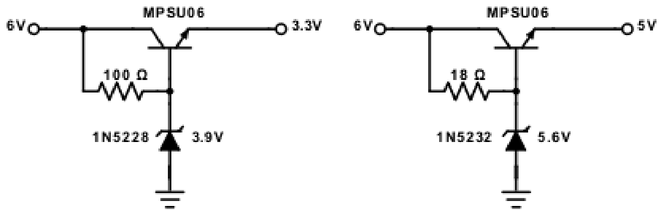

Figure 5 Voltage regulator circuits. The 3.3 V regulator is on the left and the 5 V regulator is on the right.

Since the voltage needs to be reduced from the 6V batteries, this results in some power dissipated in the voltage regulators. This current is roughly 20 mA using 6 V each. Therefore, 240 mW of power will be used up to regulate the voltages of 5 V and 3.3 V from the 6 V source. While it's possible to reduce the power consumption by using the true minimum current of 1 mA through the zener diodes, the best practice is to use the test current as outlined in the datasheet [18].

We did some simulations for voltage regulators using Simscape (a Matlab extension). We entered the next parameters: a Zener resistance of 0.1 Ω for the Zener diodes and input voltages of 5.5-6V, and we modeled the load as resistances. Knowing the output voltage and curr ent, we used Ohm's law to get the resistance values (80 Ω for the 3.3 V regulator and 60 Ω for the 5 V regulator).

The results were satisfactory: a voltage of ~3.2 V on the 3.3 V regulator, and a voltage of ~4.8 V on 5V one. These values are within the tolerance margin outlined in the requirements and verification table. The results are shown for the 5.5V and 6V inputs for the 5V regulator and similar results on the 3.3V regulator in figures 6 and 7.

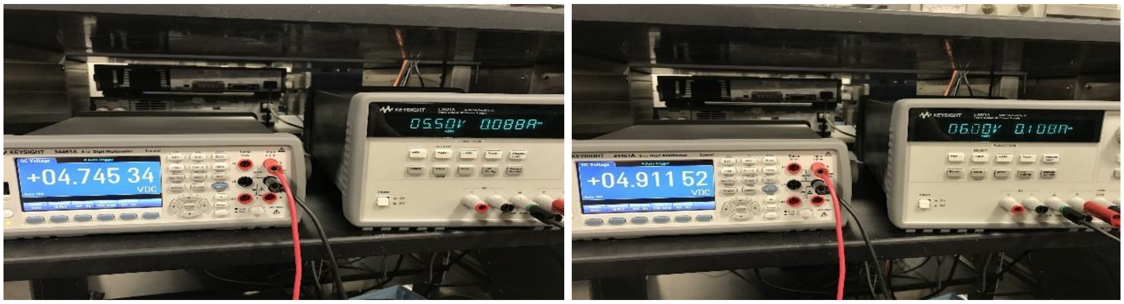

Figure 6 Simulation results of the 5 V voltage regulator.

Figure 7 Waveform of the 3.3 V voltage regulator. The supplied voltage is constant at around 3.2 V for an input of 6 V.

2.2 Sensors

2.2.1 Pressure Sensors

2.2.1.1 Design Procedure

To measure the pressure distribution of a user's weight around the chair, force sensitive resistors (FSRs) are used. 8 FSRs are sufficient to identify the posture of the user based on the placements of figure 4. Load cells are an alternative to FSRs, and they can provide more accurate and sensitive readings. However, load cells are more difficult to work with since many other components are needed to amplify and detect the signal changes. There are also issues with placing them physically on the chair since the center tab needs elevation to allow displacement and the metal bracket will not be comfortable to sit on. It should also be noted that we are not concerned with the actual weight being detected as we are about the distribution an imbalance of force.

2.2.1.2 Design Details

As seen in figure 8, we will take advantage of the fact that the force sensitive resistor varies in resistance. A constant 5 V voltage input with a constant 10 kΩ resistor in series in a resistor divider circuit will allow changes in vout to be directly attributed to the change in resistance of the FSR.

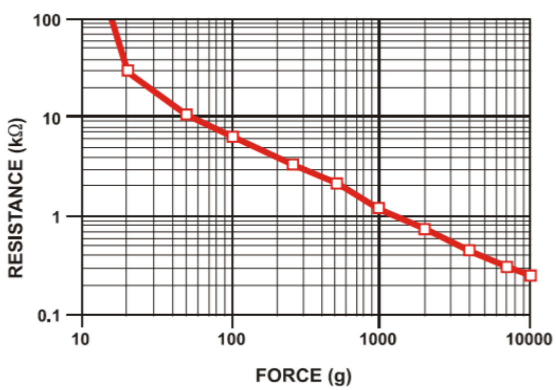

Figure 8 Resistance vs. Force for a Force Sensitive Resistor. Adapted from [11].

Using the decreasing log-log relationship of resistance to force, a voltage divider circuit can be used to get an approximately increasing linear relationship between voltage and force [11]. Figure 9 shows the circuit for each force sensitive resistor. The 1 kΩ series resistor of the voltage divider circuit was selected by testing various resistors in table 1 and selecting the value that provided the best range to support a wide range of the typical weight of a sitting user.

Series Resistor (kΩ)	Typical reading of a 160 lb user in a neutral position (range: 0 – 1023)
10	980
4.7	870
3.3	800
2.2	720
1	640

Table 1 10-bit analog-to-digital converted readings of an FSR with a varied series resistor.

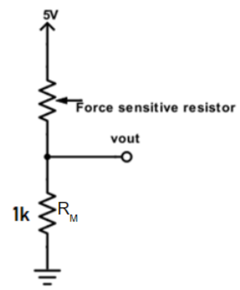

Figure 9 Voltage divider circuit.

Equation 3 shows the relationship between the output voltage being read and the FSR resistance that varies with force.

$$Vout = \frac{R_m Vin}{(R_M + R_{FSR})} \tag{3}$$

The linear readings of voltage provide a powerful realization that will be essential to the calibration topic: the readings can be shifted with to mimic different weights.

2.2.2 Distance Sensor

2.2.2.1 Design Procedure

The distance between the back of the user and the back of the chair is measured by the HC-SR04 ultrasonic sensor. It helps detect the slump posture and is also used to detect the presence of a user since there's no reason to process the sensor data if the system is not being used. Alternatives to the ultrasonic sensor include options such as LIDAR and infrared sensors. However, LIDAR is much too expensive for this project for such a small feature. Infrared sensors are also not viable since most reliable sensors require a minimum of 10 cm, which is difficult to account for on the chair

2.2.2.2 Design Details

The ultrasonic sensor is placed at the top back of the chair to detect the upper back. As shown in figure 1, the angle of the back can be useful to determine when a user is slumping over. The ultrasonic sensor has a range of distances between 2 cm and 4 m. Since the chair is approximately 50 cm from the front to back, we don't expect valid values past around 50 cm.

The ultrasonic sensor operates by sending an ultrasonic wave and detecting how long it takes for the wave to be reflected back. To convert the reading to a meaningful value such as distance. Since the waves travel at the speed of sound and they make a round trip, the distance can be measured using equation (4) where 340 m/s is the speed of sound and the distance is divided in two to get the one-way distance.

$$\text{distance} = 340 * \frac{\text{roundtriptime}}{2} \tag{4}$$

Unfortunately, there is a major flaw with the ultrasonic sensor: it is sensitive to angles. Observe the diagram in figure 10. The ultrasonic sensor is not guaranteed to receive the reflected wave if an angle is greater than ~15°. This is problematic since a slouching user can result in very high readings or very small readings if the reflection comes back at a later reading. Therefore, the sensor cannot be relied on completely. This issue is mitigated by using the ultrasonic sensor reading as a backup and only using it to determine the presence of a user when the values are valid.

Figure 10 Ultrasonic measuring at an angle. The wave is reflected away from the sensor, which results in inconsistent/garbage readings.

2.3 Control System

2.3.1 Microcontroller

2.3.1.1 Design Procedure

The ATmega328p is used as the microcontroller. This microcontroller was selected over alternatives due to the popularity, resulting in abundant documentation, as well as familiarity and ease of programmability since it is used in an Arduino Uno.

Other features such as 6 Pulse Width Modulation (PWM) pins, 10-bit analog-to-digital converter (ADC), and programmability using ArduinoISP with Arduino code [13].

2.3.1.2 Design Details

Figure 11 shows how the microcontroller was implemented. It has an 16 MHz external clock which is more reliable than the 8 MHz internal clock [13]. There are other features such as a button to reset the program and capacitor to smooth the power spikes.

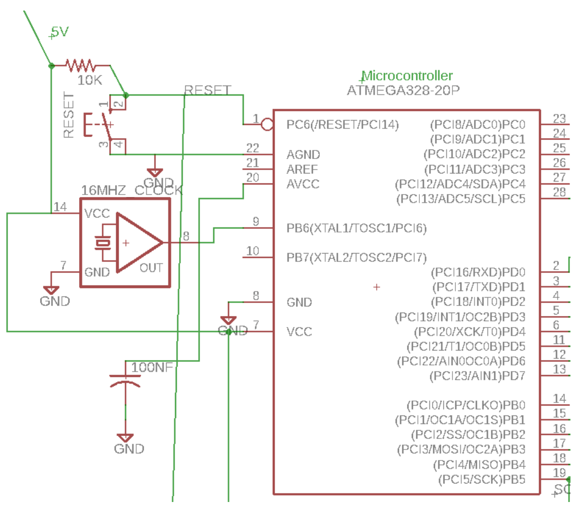

Figure 11 Microcontroller schematic.

When receiving data from the computer, the microcontroller must decode it to determine what to do with the vibration motors. Since there are only 4 vibration motors, a single byte can be used to encode which motors to activate. The method of conversion is to convert to one-hot encode the 4 most significant bits of each byte to the vibration motors. This makes it so that each bit of a received bye has the information indicated in figure 12. The reason a single byte is used is because it is much more efficient than sending multiple bytes while providing enough information for the motors. The intensity in the bottom 4 bits maps onto the range 0 to 255 in 255/16 increments to enable a PWM output to control the motor intensity. These numbers were chosen since the ATmega328p uses the values 0 – 255 for analog output and 16 intensity settings is more than enough to be distinguished by users.

Front left motor	Front right motor	Back left motor	Back right motor	Intensity scaling (4 bits)

Figure 12 Computer to microcontroller byte encoding for the activation of vibration motors. Each motor bit is active high.

The microcontroller mainly behaves as the central link between components and an interface between hardware and software. It polls all the pressure sensors from the analog multiplexer using digital output, takes ultrasonic sensor readings, sends and receives data through Bluetooth, and controls the intensity of the vibration motors. This polling occurs every 100 ms since 10 samples per second is sufficient for the measurement of a user's posture. Having more readings would be redundant since the user is not expected to be making very quick movements while sitting. Since the microcontroller has a single core and no multithreading capabilities, these tasks must be quick and/or non-blocking to avoid slowing down the entire system. Figure 13 shows what the control flow of the microcontroller will look like. It will continuously poll data from the sensors and the Bluetooth module to decide what to do.

The ADC in the microcontroller uses 10 bits to convert a voltage of between 0 and 5 V into a value between 0 and 1023. This is highly granular with for the FSR readings [13].

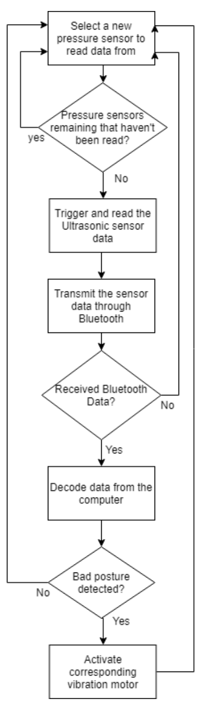

Figure 13 Microcontroller program flowchart.

2.3.2 Analog Multiplexer

2.3.2.1 Design Procedure

The CD4052 8-to-1 analog multiplexer is used for the project. Since there are 8 FSRs to take analog readings from and only 6 analog input pins on the ATmega328p microcontroller, a multiplexer must be used to select them one at a time.

2.3.2.2 Design Details

The analog multiplexer is selected by three digital pins of the ATmega328p.

2.4 Data Stream

2.4 Bluetooth

2.4.1.1 Design Procedure

We will be using a Bluetooth connection to transmit data between the chair and the computer. This part was selected because it is very easy to use and can be found for a reasonable price. The Bluetooth stack we chose is an embedded system approach, because we are implementing a Bluetooth peripheral device. The Bluetooth module we have picked out is the HC-05. It is a simple TX/RX pipeline, and is powered by 3.3 V [15].

The range is up to approximately 30 feet. The HC-05 also has a 2.4 GHz frequency and 9600 bits per second default transmission rate configurable up to 1,382,400 baud rate [15]. The module is expected to have a 100% accuracy unless the connection fails.

2.4.1.1 Design Details

The transmission of data is sent as comma-separated value (CSV) since the Bluetooth data is received together as a row. CSVs are also easy to parse in software.

To compute the transmission rate needed, we use the 10 samples per second that the microcontroller sends. Since each a pressure sensor reading can take values up to 1023, this is broken down into 4 bytes. For a with a distance reading of around 3 bytes. With 8 commas separating 4*8 pressure sensor bytes and 3 bytes of distance with 1 newline byte, the maximum data sent per row is 44 bytes. Therefore, up to 44 bytes*10 samples/second = 3,520 bits per second. Therefore, the default transmission rate of 9,600 bits per second is enough for our purposes.

2.5 User Interface

2.5.1 Vibration Motors

2.5.1.1 Design Procedure

We will use 4 C1034B018F vibration motors mounted with two on the left side of the seat and two on the right side. This allows us to move the user left, right, front, or back. Vibration motors work nicely since there are not as disruptive as using aural methods and can be subtle yet effective in notifying the user. This was inspired by cell phone vibration. These are also small enough to not be uncomfortable when sitting on them.

2.5.1.1 Design Details

These vibration motors operate on 2.7~3.3V with a starting voltage of 2.3V, which scales linearly with intensity [16]. They are toggled individually by the microcontroller, 4 of the PWM pins. The motors have a maximum steady-state current draw of ~80 mA, which is about double the ~40 mA maximum current output of a digital pin of the ATmega328p [13], [16]. So a transistor is needed to supply the proper amount of current without damaging the microcontroller.

When a motor is at steady-state, the change in current is small or 0 so the voltage is essentially 0. When the power is switched off, the change in current becomes negative. This will result in a negative voltage which means the inductor is now supplying current proportional to the change in current from the source. This can result in catastrophic voltage levels that may damage components in different parts of the circuit. To remedy this issue, we place the vibration motors in a circuit shown in figure 14 below. The parallel reverse-biased (also known as a “flyback”) diode is used to absorb all of the back emf from the motor when the motor is turning off. The NPN transistor allows the microcontroller to supply a low current digital output that will connect the motor to ground and turn it on.

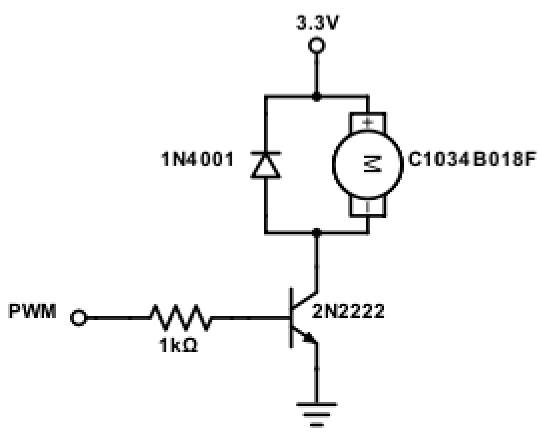

Figure 14 Vibration motor circuit. Adapted from [24].

2.5.2 Graphical User Interface (GUI)

2.5.2.1 Design Procedure

The user interface is how the system communicates with the user. The GUI on the computer screen that provides information like the pressure heatmap and posture style. The implementation of the GUI is through TKinter in Python. This provides a local GUI that can interact with the other software components. We had initially planned on delivering the service through a website, but that got in the way of a real-time feedback implementation. Using python already extended the duration of the computation cycles, and adding database operations to the system would slow the system down too much. A potential solution would be to fetch data using a more thread-efficient language such as C++ since it doesn’t have to share a thread with the processor-intensive GUI.

2.5.2.1 Design Details

The GUI was modeled to provide information on the areas being sensed. As seen in figure 15, the areas of interest such as the shoulder, the lumbar, the ischial tuberosity, and thighs are displayed for the user to analyze. This was selected based on the findings of [8]. The learning rate for calibration can be inputted as well. This allows the user to customize what the system considers as “good” posture

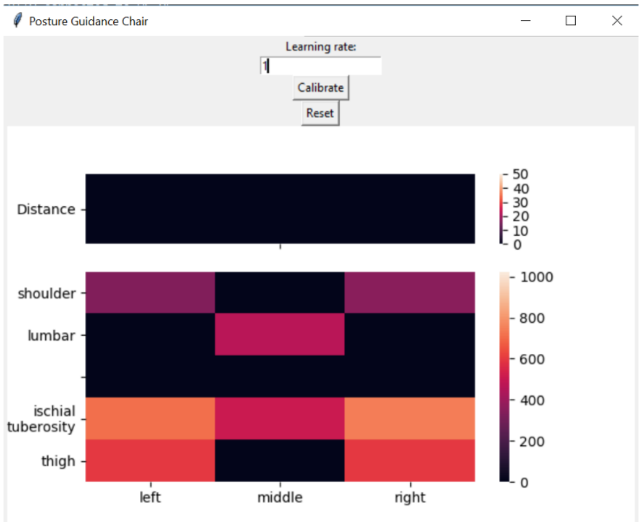

Figure 15 GUI layout

2.6 Software

2.6.1 Data Processor

2.6.1.1 Design Procedure

The software was built using Python 3.6 for ease of prototyping and powerful machine learning capabilities with scikit learn. Our main posture classification algorithm is an implementation of the Support Vector Machine (SVM). We chose to use an SVM because it is reliable with small training sets and works nicely for binary classification (and regression) [20]. Therefore, we have used each sensor's readings as a feature to estimate the decision boundaries that separate the good postures from the bad. Since we have decided to use supervised learning, we had to be careful not to fall victim to overfitting the classifier while we attempted to provide data for various cases and postures that could occur.

2.6.1.2 Design Details

Our final design had a training set made up of 134 bad and 51 good posture features gathered from people with varying weights. To understand what the feature space looks like, we performed a principal component analysis (PCA) on the training data. We obtained the following results:

- Principal Component 1
    - Explains 35.6% of the variance
    - Most relevant features: ischial tuberosities, lumbar
- Principal Component 2
    - Explains 21.1% of the variance
    - Most relevant Features: shoulders
- Principal Component 3
    - Explains 18.9% of the variance
    - Most relevant Features: thighs

In figure 16, we see that the datapoints are nicely separable, so SVM works nicely.

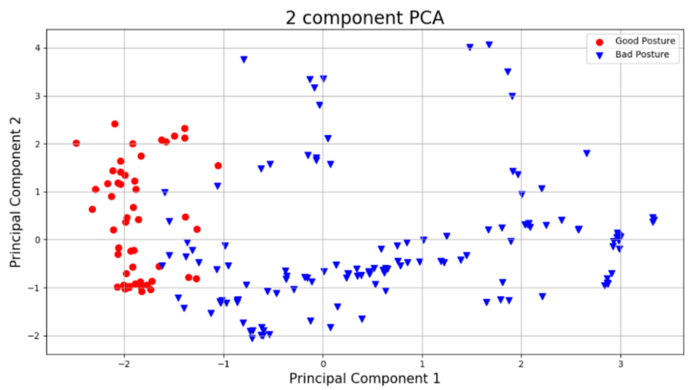

Figure 16 PCA plot of feature space

To obtain the best hyperparameters, we used randomized cross-validated grid search with 5 folds. The same training set was used in the principal component analysis, but obviously a separate data set was used as the test set to determine the accuracy of the classifiers. The binary posture classifier had an accuracy of 98% (C = 0.174, gamma = 0.0281) and the multi-class haptics classifier scored a 95% (C = 356, gamma = 0.0281). The control flow in figure 17 shows how these classifiers are used.

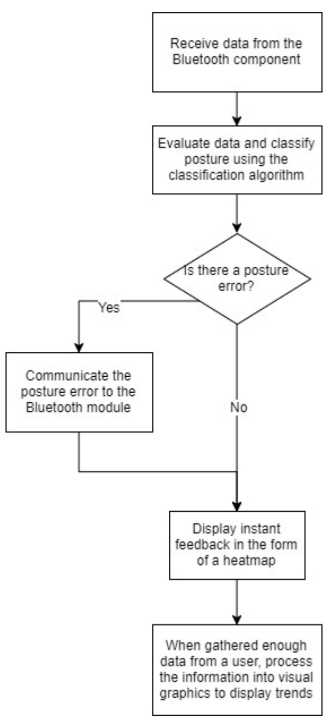

Figure 17 Data processor flow chart

While SVMs were ideal for our objective, we did face some challenges concerning the complexity of the algorithms. Since SVM is a quadratic programming problem, the implementation in Scikit Learn takes between $O(n_{features} \times n^2_{samples})$ and $O(n_{features} \times n^3_{samples})$ time [22]. Since each sensor was used as a feature, $n_{features}$ is 9 and $n_{samples}$ is 185. Any more data in the training set proved to have cycles that were too slow, and the space requirements scaled at a similar rate. Due to the scale invariance of Support Vector Machine algorithms, it is highly recommended to scale/standardize the data with mean 0 and variance 1 [22]. In particular with our dataset, we shifted the pressure sensor readings to a baseline average of all the current pressure sensors.

After we complete the binary classification to determine if a posture is good or not, we need to figure out which motor to send the vibration signal to. Given an exhaustive training set covering most cases on each sensor, we can identify which vibration motor to activate by applying the data to a multi-class SVM, which follows the same design principles as the binary classifier.

To generalize the data to multiple users of different weights, we make use of the fact that the FSR voltage reading is roughly linear with force. Therefore, users of different weight can have their weights shifted to a certain baseline of good posture determined from the training set. This makes. Equation 5 shows how the calibration works when the user specifies alpha. It takes a portion of the input shifted by the previous calibration value w, then creates a new value to be used on future inputs. This allows it not only able to detect user of different weights, but also consider different postures as good. For example, an amputee may have a significantly different weight reading with good posture, so the readings can be shifted to account for the calibrated position. It is expected that users calibrate using what is considered good posture.

$$w' = w + \alpha(baseline - input - w) \tag{5}$$

The limitation of accurate calibration is around a user of 200 pounds since that is when the raw sensor readings become very high and potentially unreliable.

3. Costs

3.1 Parts

Description	Manufacturer	Part #	Quantity	Unit Cost ($)	Total Cost ($)
4-pack AA Batteries	Energizer	39800011329	1	3.59	3.59
Battery Holder, 4AA	Keystone Electronics	2478CN	1	2.80	2.80
Force Sensitive Resistor	Sparkfun Electronics	SEN-09376	8	9.95	79.60
Ultrasonic Sensor HC-SR04	Sparkfun Electronics	SEN-13959	1	3.95	3.95
ATmega328p	Atmel	DEV-10524	1	9.49	9.49
8-input Analog Multiplexer	ON Semiconductor	74HC4051	1	4.00	4.00
Bluetooth Module	Electronica 60 Norte	HC-05	1	6.60	6.60

Vibration Motors	Jinlong Machinery & Electronics	C1034B018F	4	2.88	11.52
				Total	121.55

Table 2: Part cost

3.2 Labor

$$Labor\ cost\ =\ 3 \cdot \frac{\$40}{hour} \cdot \frac{10\ hours}{week} \cdot 16\ weeks \cdot 2.5 = \$48,000 \qquad (6)$$

5. Conclusion

5.1 Accomplishments

Our senior design project, the orthopedic chair, was built using many of the ECE branches we got to be familiar with over the past few years, and ended up working as initially envisioned. Our group mates used their expertise in various fields such as power, control systems and software to successfully build the many components that makes up our project. Our confidence in our project comes not only from the fact that many of our instructors and peers have tested our chair for themselves, but also from the fact that we believe that we have documented the design and the implementation process of the project very succinctly.

5.2 Uncertainties

We do have some regrets about our methods. Using C++ would reduce our cycle latency by a significant amount. We would have also liked to have a broader training sample, but for the scale of the project and our time budget, we did about as well as could have been.

5.3 Ethical considerations

There is a certain risk in this project, which must be handled: the risk of having an electrical system at a place in which a human being is going to sit.

The requirement to handle this problem is using enough isolation measures, such as using insulating materials or putting critical components at places which are not too close to the user. There could be around 100 mA of current flowing around the circuit, so protective measures must be taken to avoid potential injury as a result of contact. Also, there are many components spaced out around the chair. Naturally, there will also be a lot of wires and cables to handle properly so that the user doesn't get hurt by accidentally tampering with them. The primary solution for this is to make the design as seamless as possible and keep things organized.

However, it is impossible to move away from the user all the components. For example, the sensors need to be near the user, and in fact, the pressure sensors will be in contact with him or her. These pressure sensors are the most critical part of the system, so an insulating material will be put between the sensors and the user. This material must be strong enough to electrically isolate the sensors and the user, but at the same time it must be soft, so when the user sits on the chair the pressure sensors will be pressed, despite the fact that there is an insulating material between them.

This aligns with the IEEE code of ethics. To be more specific, it follows point 1 of IEEE code of ethics: to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment [5]. Our project may be useful to improve the people's health, which is obviously an ethical action.

As indicated by IEEE Code of Ethics #3, we must be honest about our claims about what our data represents [5]. This system is intended to improve the posture of the user, but we cannot claim to diagnose or treat potentially serious health issues related to posture. In fact, we'll need to inform the public using disclaimers to see more personalized experts such as their physician or more proven techniques.

5.4 Future work

Future work may involve cleaning up the system to be production-ready. This may involve using custom cushions to hide the sensors, enhancing the software by moving processing to a web server, and using optimal code such as C++.

References

[1] American Chiropractic Association, “Back Pain Facts and Statistics,” *American Chiropractic Association*. [Online]. Available: https://www.acatoday.org/Patients/Health-Wellness-Information/Back-Pain-Facts-and-Statistics. [Accessed: May 2, 2018].

[2] American Chiropractic Association, “Tips to Maintain Good Posture,” American Chiropractic Association. [Online]. Available: https://acatoday.org/content/posture-power-how-to-correct-your-body-alignment. [Accessed: May 2, 2018].

[3] D. Brown, “Experts say posture matters: The good ... and the bad,” GoUpstate, 10-May-2010. [Online]. Available: http://www.goupstate.com/news/20100511/experts-say-posture-matters-the-good--and-the-bad. [Accessed: 08-Feb-2018].

[4] J. Flynn, “Is it possible to cure a bad posture?,” The Independent, 26-Oct-2015. [Online]. Available: http://www.independent.co.uk/life-style/health-and-families/features/is-it-possible-to-cure-a-bad-posture-a6709781.html. [Accessed: 08-Feb-2018].

[5] Ieee.org, "IEEE IEEE Code of Ethics", 2018. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08-Feb-2018].

[6] “UNITED STATES DEPARTMENT OF LABOR,” Occupational Safety and Health Administration. [Online]. Available: https://www.osha.gov/SLTC/etools/computerworkstations/positions.html. [Accessed: 20-Feb-2018].

[7] A. Claus, J. Hides, G. Moseley, and P. Hodges, “Is ‘ideal’ sitting posture real?: Measurement of spinal curves in four sitting postures,” Manual Therapy, vol. 14, no. 4, pp. 404-408, June 2008. [Online]. Available: www.sciencedirect.com. [Accessed Feb. 20, 2018].

[8] Y. Zheng and J. Morrell, “A vibrotactile feedback approach to posture guidance” in Haptics Symposium, 2010 IEEE, 25-26 March 2010 [Online]. Available: IEEE Xplore, www.ieee.org. [Accessed 20-Feb-2018].

[9] L. Martins, R. Lucena, J. Belo, M. Santos, C. Quaresma, A. Jesus, and P. Vieira, “Intelligent Chair Sensor Classification of Sitting Posture,” Communications in Computer and Information Science, 2013. [Online]. Available: www.researchgate.com. [20-Feb-2018].

[10] J. J. Swearingen, C. D. Wheelwright, J. D. Garner, “An Analysis of Sitting Areas and Pressures of Man,” Civil and Medical Research Institute, Oklahoma City, Oklahoma, 1962. [Online]. Available: www.faa.org. [Accessed 22-Feb-2018].

[11] Interlink Electronics, Inc., “Force Sensing Resistor Integration Guide and Evaluation Parts Catalog.” [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf. [Accessed 22-Feb-2018].

[12] ElecFreaks, “Ultrasonic Ranging Module HC-SR04.” [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf. [Accessed 22-Feb-2018].

[13] Atmel, “8-bit AVR Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash,” 2009. [Online]. Available: https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf. [Accessed 22-Feb-2018].

[14] NXP Semiconductors, “74HC4051; 74HCT4051 8-channel analog multiplexer/demultiplexer,” 2016. [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/5/5/3/74HC_HCT4051.pdf. [Accessed 22-Feb-2018].

[15] "HC Serial Bluetooth Products User Instructional Manual", Cdn.makezine.com, 2018. [Online]. Available: https://cdn.makezine.com/uploads/2014/03/hc_hc-05-user-instructions-bluetooth.pdf. [Accessed: 22-Feb-2018].

[16] Jinglong Machinery & Electronics, Inc., “Coin type vibration motor,” 2007. [Online] Available:http://www.vibration-motor.com/products/download/C1034B018F.pdf. [Accessed: 05-Mar-2018].

[17] Energizer Brands, LLC, “Energizer E91 Product Datasheet.” [Online]. Available: http://data.energizer.com/pdfs/e91.pdf. [Accessed: 22- Feb- 2018].

[18] Vishay Semiconductors, “1N5221 - 1N526 Small Signal Zener Diodes,” 2008. [Online]. Available: http://www.vishay.com/docs/85588/1n5221.pdf. [Accessed: 22- Feb-2018].

[19] Motorola, “Bipolar Power Transistor Selector Guide & Cross Reference.” [Online]. Available: http://www.datasheets360.com/pdf/7456842635131963505. [Accessed: 22- Feb- 2018].

[20] S. Lazebnik, “Support Vector Machines,” 2017. [Online]. Available: http://slazebni.cs.illinois.edu/fall17/lec17_svm.pdf. [Accessed: 22- Feb- 2018].

[21] S. Lazebnik, “Neural Networks,” 2017. [Online]. Available: http://slazebni.cs.illinois.edu/fall17/lec18_neural_nets.pdf. [Accessed: 22- Feb- 2018].

[22] “1.4. Support Vector Machines,” 1.4. Support Vector Machines - scikit-learn 0.19.1 documentation. [Online]. Available: http://scikit-learn.org/stable/modules/svm.html. [Accessed: 26-Feb-2018].

[23] “Nearest centroid classifier,” Wikipedia, 17-Feb-2018. [Online]. Available: https://en.wikipedia.org/wiki/Nearest_centroid_classifier. [Accessed: 26-Feb-2018].

[24] How to Build a Vibration Motor Circuit. [Online]. Available: http://learningaboutelectronics.com/Articles/Vibration-motor-circuit.php. [Accessed: 05-Mar-2018].

Appendix A Requirement and Verification Table

Component/Module	Requirement	Verification	Status
Voltage Regulators	The 3.3V voltage regulator must consistently output voltages between 3.0V - 3.6V given an input of 5.5-6V.	1. Provide a 5.5-6V input signal into the 3.3V regulator. 2. Check the output voltage to make sure it's within 3.0-3.6V.	Y
	The 5V voltage regulator must consistently output voltages between 4.5V - 5.5V given an input of 5.5-6V.	1. Provide a 5.5-6V input signal into the 5V regulator. 2. Check the output voltage to make sure it's within 5.5-6V.	Y
Pressure Sensors	A neutral balanced sitting position with two sensors on both sides of the body should output voltages within 0.1V of each other.	1. Place two sensors directly under the subject. 2. Use a leveler to ensure the user is balanced. 3. Measure the voltages to make sure the difference is within 0.1V (~20 difference in readings).	Y
	A position with an upright or slight lordosis should register readings of over 150 higher than a slouching position. The readings under the thigh and ischial tuberosity should be distributed accordingly (ischial tuberosity should increase as the user applies more pressure to the back).	1. Start with a slouching position such as the slump posture in figure 1 of the design document. 2. Record the readings for the lumbar, ischial tuberosity, and under the thigh. 3. Recline to a flat, long lordosis, or short lordosis position and take the same measurements. 4. The lumbar should register over 150 points higher and the ischial tuberosity should increase while the thigh decreases linearly.	Y
Distance Sensor	Distance readings	1. The user should sit on	

	where the user is leaning as far back as possible in the chair should hold valid values of at least 2 cm.	the chair with the distance sensor mounted on the back. 2. Lean back by applying as much pressure as is comfortable on the back. 3. Record the readings of the sensor and confirm the readings are valid	Y
Microcontroller	The microcontroller should be able to decode received bytes that have the form [3:0] one-hot encoded motor ID [7:4] Motor intensity mapping onto the voltage range of 2.3V – 3.3V The intensity should be used to activate the indicated motors.	1. Create a program that sends bytes to all combinations of motor IDs from 0x0 to 0xF 2. Each ID combination should be sent with varying intensities. 3. Verify that the motors are identified in the lower 4 bits and the intensity/voltage changes as the higher 4 bits change.	Y
	Readings from each sensor should be successfully polled by the microcontroller and sent as a CSV row ending with a newline.	1. Have the microcontroller collect sensor data and send them to the computer over Bluetooth. 2. Confirm the CSV format ending in a newline.	Y
Analog Multiplexer	Power dissipation per input must be less than 100 mW.	1. Apply the pressure sensor inputs into the mux. 2. Measure the voltage and resistance to calculate the power as V^2/R.	Y
Bluetooth Module	Between 10 and 25 readings of all sensors should be sent in roughly 1 second.	1. From the computer, have a thread sleep for 1 second. 2. When the thread wakes up, verify that the number of readings in the queue is between 10 and 25.	Y
Vibration Motors	Must be noticeable when sitting on the	1. Provide a ~3.3V power input to the vibration	

	chair when operating on 3.0V+/-0.3V.	motor circuit. 2. A user sitting on the motor should be able to identify which motor is active.	Y
	The current draw of each motor must not exceed 120 mA when applying a ramped PWM signal.	1. Supply power to the vibration motors starting from ~2.3V to ~3.3V via PWM. 2. As the intensity/voltage increases, check the current draw and confirm that a single motor does not draw more than 120 mA.	Y
	Must be spaced at least 1.5-2 inches away from the nearest pressure sensor to avoid interference.	1. Obtain a ruler to measure the distance from each vibration motor to the nearest FSR. 2. Verify that the distances from the motors to the nearest pressure sensor is greater than 1.5-2 inches.	Y
Posture Analysis GUI (Software)	Should be able to receive user-inputted calibration information and send it to the posture processing program.	1. The user should be able to enter a "learning rate" number between 0 and 1.0. 2. Check that the posture processing program receives the learning rate.	Y
	Must update the visualizations and posture indications from the posture processor in real-time (around less than 2 seconds of latency).	1. Send posture data to the GUI from the posture processing program. 2. The GUI should update in less than 2 seconds after receiving the new posture data and decisions.	Y
Software	The supervised learning model	1. A poorly trained model cannot accurately	

	should be trained with at least 100 samples of bad posture and at least 50 samples of good posture.	classify data that are slightly unusual, so provide various, distinct poses. 2. Record the posture data with the pressure sensors and distance sensor. 3. Check that the positions were correctly classified as good/bad.	Y
	The classification accuracy should be at least 85%.	1. Create a test set of posture data that is separate from the training set. 2. Score the binary posture classifier and verify that at least 85% of the data is correctly classified.	Y
	Calibration should be able to learn various positions indicated by the user (possible to learn "bad" posture as "good").	1. The user should lean forward until the classifier consistently labels the position as bad. 2. Input a learning rate above 0.5. 3. Check that the new bad posture is now classified as "good."	Y
	The vibration algorithm should identify the correct motors to activate with 70% accuracy.	1. Apply various biased positions in different directions (left, right, front, back) in a test set. 2. Observe which motors are activated and score the accuracy.	Y