Posture Sensing Smart Chair

Design Document

ECE 445 Spring 2020

TA: Jonathan Hoff

Geonil Kim, Steven Zhou, Brian Hill

1 Introduction

1.1 Problem and Solution Overview:

Back problems due to slouching have become a huge cause for concern as people spend more and more time sitting in cubicles, hunched over a computer for hours upon hours every single day. In fact, according to a University of Washington study, forty-five percent of Americans between the ages of 35 and 55 suffer acute back pain each year [1]. Moreover, other studies carried out by the Social Security Administration identified back pain as the top cause of disability under the age of 45 in the United States [1]. Not only is poor posture unsightly, but it also introduces stress on the neck and spine, causing further muscular tension as the body attempts to compensate for the lack of support.

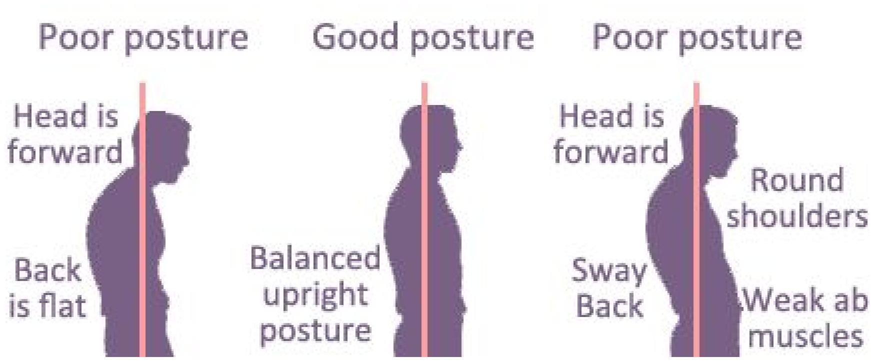


Figure 1: Good and Poor Posture

While maintaining proper posture is something that needs to be taken seriously in perpetuity, we have decided to combat this problem by focusing on time spent sitting in front of a computer, like in a traditional office environment, for example. After all, many adults spend a significant portion of their lives in this position! In response to this, we have set out to create a computer vision system that uses video input of the user's sitting position to determine whether or not they are sitting with proper posture. We will calculate the relative positions of the head, neck, shoulders, upper/lower back, and legs to determine if the user's alignment is safe and healthy. Should the user revert back to a poor sitting position, they will be notified via a combination of flashing lights and vibrations originating from within their chair.

1.2 Background:

From our own experiences with posture-related back problems, we know the hardest part about changing our habits is accountability. While strengthening postural muscles is important in correcting kyphotic posture, it is useless if you are unable to *remember* to put those muscles to use. By using our product, we hope to make people more aware of the way they are sitting and allow them to monitor their progress as they try to break the habit.

Because poor posture is such a widespread problem there currently exist many solutions on the market. A large portion of these solutions fall under the umbrella of electronic wearables [3], like the Upright GO, shown in figure 2, which attaches to the user's back between the shoulder blades. While we did initially consider the idea of designing our own wearable, we ultimately decided against it because we suspect that many people would find them uncomfortable. Other solutions involve harnesses that physically pull the shoulder blades back, shown in figure 3. The problem with these lies in the fact that they don't support the development of stronger postural muscles and therefore won't actually fix the problem. The last major class of solutions to poor posture in the workplace are ergonomic workstations and include things like standing desks and using an inflatable ball instead of a chair. While these solutions have the right idea in mind, the average person does not want to stand or roll around on a ball for eight hours a day; it's exhausting and unsustainable for many! We believe our proposed solution would have many advantages over the existing products on the market.

Figure 2: Upright Go

Figure 3: Harness

Originally, we wanted to approach this chair with a robust method of using various sensors to read a user's sitting posture, but after more research we figured this could be improved for more accuracy and reliability using computer software. Our first proposed idea had two separate subsystems to gather user information -- a back subsystem and a seating subsystem. Although this method may work, it would be more efficient and reliable to reduce the two measuring systems into one. By having just the software subsystem, we do not have to worry about faulty/inaccurate measurements from the sensors nor do we need the subsystems to be dependent on each other. Our computer vision subsystem will be a module that is placed a few feet away from the side of the chair. We believe this is by nature the best way to measure since the camera would not be able to see all of the user's segments from other placements (front and back). In addition, computer vision allows a better visual display of a user's sitting posture. Instead of gathering raw numbers to show the user, there are now live images that detect the different segments and check its alignment. Finally, our new proposed system has the advantage over the previous proposal in that the chair will no longer be anchored by a cord as all communication to the chair will be handled using bluetooth and all power going to the chair will come from a Li-ion battery.

Using computer vision to estimate a human's pose is definitely a well-researched topic with plenty of open source software and data sets that we will be able to utilize in our design. Some of the more well known data sets that are available to the public are the COCO 2018 Keypoint Detection, the MPII Human Pose Dataset, and VGG Pose Dataset[10]. Furthermore, there are even several open source models that are pre trained using these datasets which would eliminate any need for us to tackle this very complex machine learning problem ourselves[10]. The image below shows the results of one model that is trained using different datasets.

COCO KeyPoints MPII KeyPoints

Figure 4: same model trained with different data sets

Easy implementation of models such as the one depicted above are possible using OpenCV's plethora of useful functions. Originally developed by Intel, OpenCV is an open source computer vision and machine learning software library containing interfaces for several languages -- including python, C++, Java, and MATLAB -- as well as support for Windows, Linux, Android and Mac OS users alike[11]. OpenCV can provide the means to utilize these models to effectively classify the user's posture in real-time. By finding pixel locations of several key points on the user we will be able to calculate different angles formed by these points and use these angles to classify posture. This method of calculating angles would provide a strong advantage over our previous method of using pressure and range sensors because it allows us to better *quantify* their posture and give feedback on what the user should specifically improve on. For example, consider a situation in which the user's shoulders begin to round and their posture falls out of the range of what is determined to be acceptable. While both solutions would be able to inform the user of their poor posture, the computer vision solution would ideally be able to specifically notify the user that they need to realign their shoulders.

1.3 Visual Aid

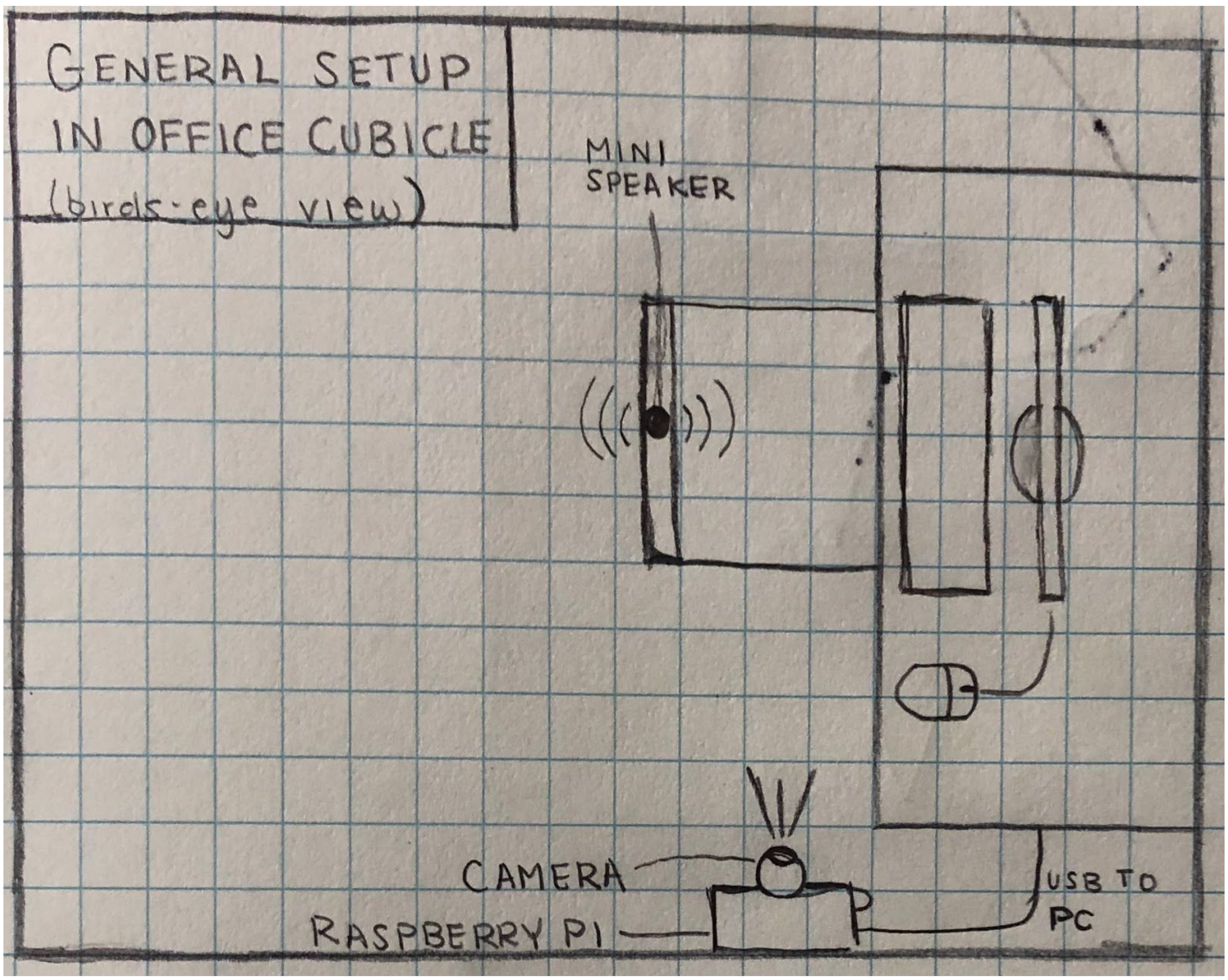


Figure 5: Visual example of how the system might be set up in an office environment

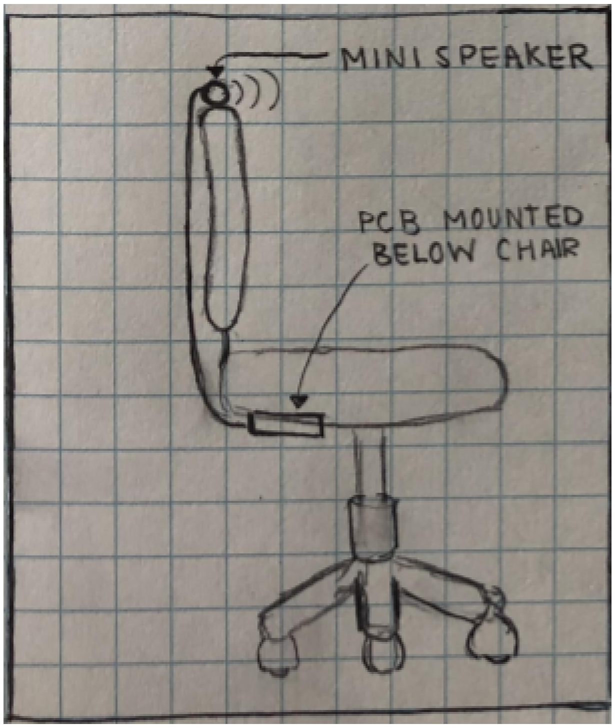


Figure 6: Placement on chair

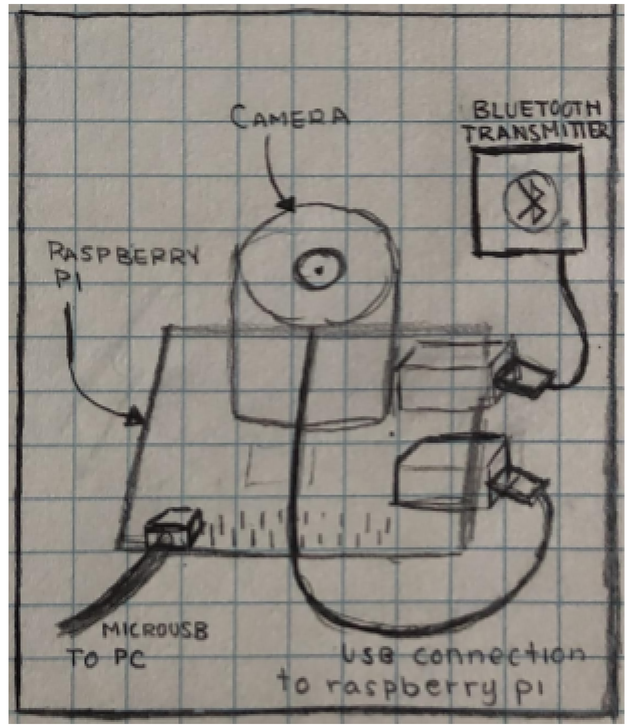


Figure 7: Computer Vision subsystem

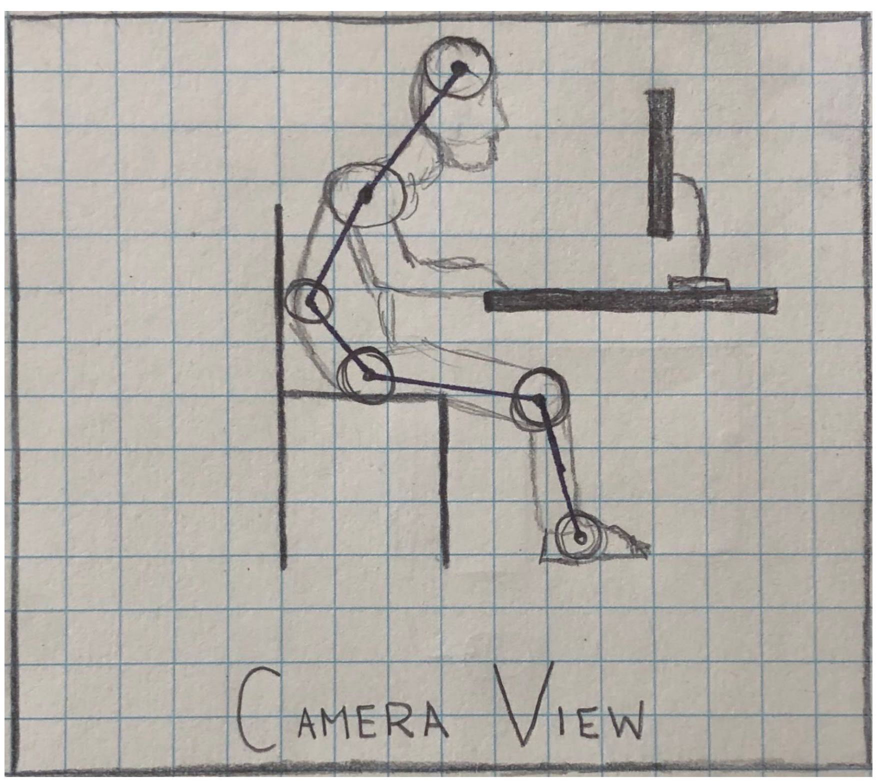


Figure 8: Camera View

1.4 High-level requirements list

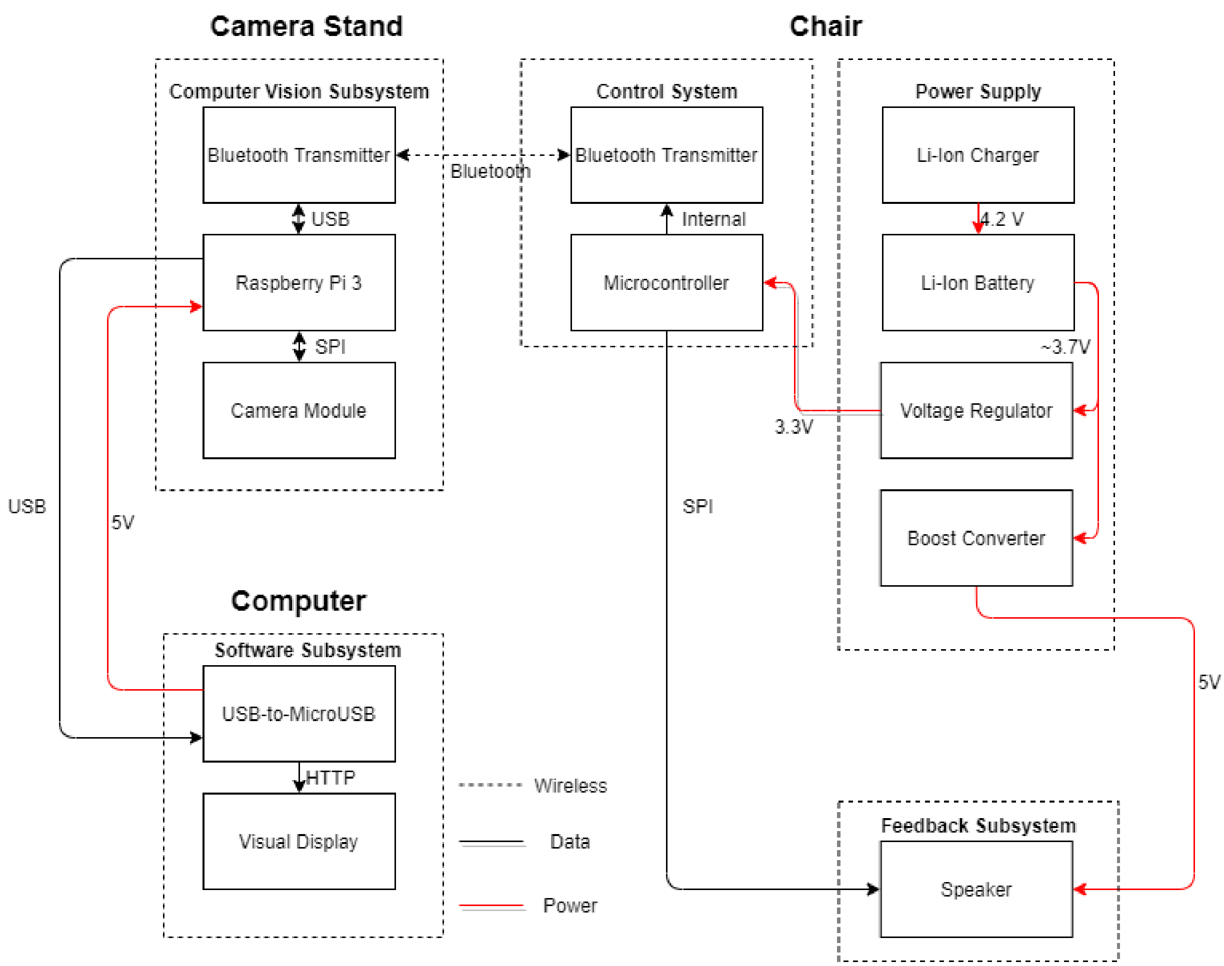
1. Using computer vision software, the system must be able to accurately classify the users posture as either good or poor with at least 70% accuracy.

2. The computer must be able to collectively analyze the data provided from the computer vision subsystem and send feedback back to the chair to notify the user when their posture is poor by sounding a small speaker.

3. The data analyzed must be shown visually through a graphical display -- available via web application -- showing the user's posture over time.

2 Design

2.1 Block Diagram



2.3 Power Supply

2.3.1 Li-Ion Charger

The Li-Ion charger is a voltage limiting device that will supply power to the battery once it is depleted. It will charge the Li-Ion cell batteries through a charging IC with power coming from a microUSB. It shouldn't take longer than 7 hours to fully charge the batteries.

Requirement	**Verification**
Must charge the battery to between 3.7V $\pm$ 5% when supplied with a 5V source from the microUSB	1. Discharge the battery to 3V 2. Power the battery through the microUSB 3. After charging is finished, ensure the battery's voltage is between 3.7V a $\pm$ 5% using the oscilloscope

2.3.2 Li-Ion Battery

Our product will use a 3.7V Li-Ion rechargeable battery. The Li-Ion battery will be responsible for providing the power needed for all subsystems located on the chair. It will provide power for our audio feedback subsystem, the microcontroller, and bluetooth transceiver. The camera module will not receive power from the battery but instead from the PC through the microUSB.

We will be using a 6600mAh Li-Ion battery for this design.

Requirement	**Verification**
Must provide voltage between 3.7V $\pm$ 5%	1. Connect a simple load to the battery to drive 1A 2. Probe using an oscilloscope and ensure the value does not leave the range of 3.7 V $\pm$ 5%
Must power the device for at least 7 hours	1. Leave the chair running for 7 hours 2. Connect an LED from the battery to ground and see if it lights up.

2.3.3 Voltage Regulator

The voltage regulator will be used to supply the battery power to the microcontroller. This is included as a protection measure to prevent damages to the microcontroller. Shunt Capacitors added before and after to filter out any potential high frequency content.

Requirement	**Verification**
Must regulate the voltage to 3.3 V ± 5% for the microcontroller	1. Probe using Oscilloscope 2. Observe and ensure that the value does not leave the range of 3.3 V ±5%

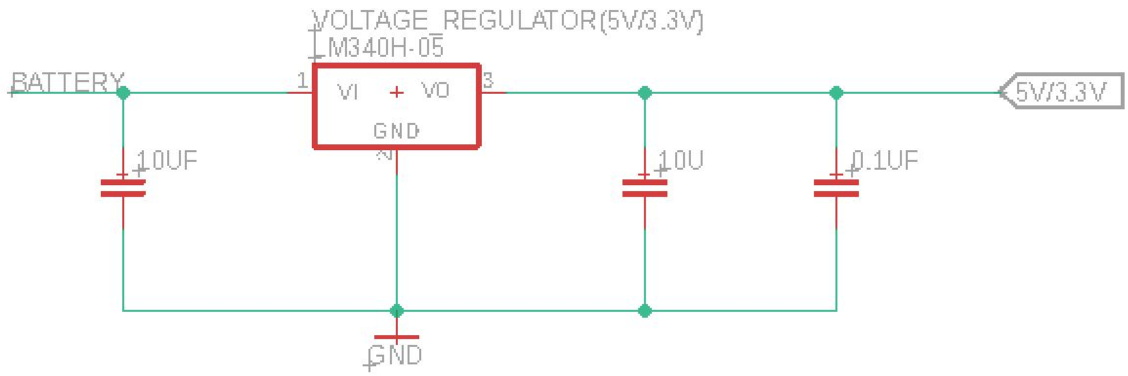


Figure 9: Circuit schematic for Voltage regulator

2.3.4 Boost Converter

The boost converter will be used to supply the battery power to the speaker feedback system. This block is included as a DC/DC converter because the voltage requirements for the feedback system are greater than what is provided by the Li-ion battery.

Requirement	**Verification**
Must supply the voltage of 5 V ± 5% with the input voltage between 3 V and 5 V	1. Input DC voltage between 3V and 5V 2. Probe using Oscilloscope 3. Observe and ensure that the output does

	not leave the range of 5 V $\pm 5\%$

2.4 Control System

2.4.1 Microcontroller

We will be using a high-performance, low-power ATmega328P 8-bit microcontroller for this design. We chose this microcontroller for its affordability and abundance of documentation. We are also most familiar with this microcontroller as it was used in Arduino Uno. Atmega328P has a full suite of programs and system developing tools, such as C compilers, in-circuit emulators, and evaluation kits, which could be used in testing feedback subsystem [6]. The microcontroller will receive the analyzed data from the Raspberry Pi and manage the feedback subsystem.

The main purpose of ATmega328P would be to receive the data from Raspberry Pi and manage the audio feedback subsystem. Once the Raspberry Pi analyzes the user's posture with the computer vision algorithm and concludes that the posture is poor, it will trigger the speaker to play a sound and notify the user. The microcontroller interfaces with an arduino bluetooth transceiver module through which it receives the information from the raspberry pi. Upon receiving a signal declaring the user's posture is poor, the microcontroller will set its I/O pin to high voltage which will trigger a beeping sound on the speaker.

Requirement	**Verification**
Must respond to the bluetooth signal by setting its I/O to high voltage, thus triggering the speaker to sound without noticeable lag (at most 250 ms)	1. Hard code the signal for poor posture to be 'true' 2. See if the speaker sounds within the allowed time span

2.4.2 Bluetooth Transceiver

The bluetooth transceiver is used as a communication pipeline between the chair and the computer vision subsystem. Since the camera and microcontroller is separate from the chair we want to avoid as much wiring as possible to prevent accidental damages as well as increase user mobility. The Bluetooth transceiver will receive the analyzed data from the Raspberry Pi and relay the information to the microcontroller in the chair.

Requirement	**Verification**
Bluetooth transmitter is compatible with	1. Setup test scripts for the raspberry pi to

Raspberry Pi	react towards signals received from transmission. 2. Analyze behavior and code from the raspberry pi after sending signals.
Must not take longer than 300ms to receive data from the Raspberry Pi	1. Record time of transmission after transmission trials.

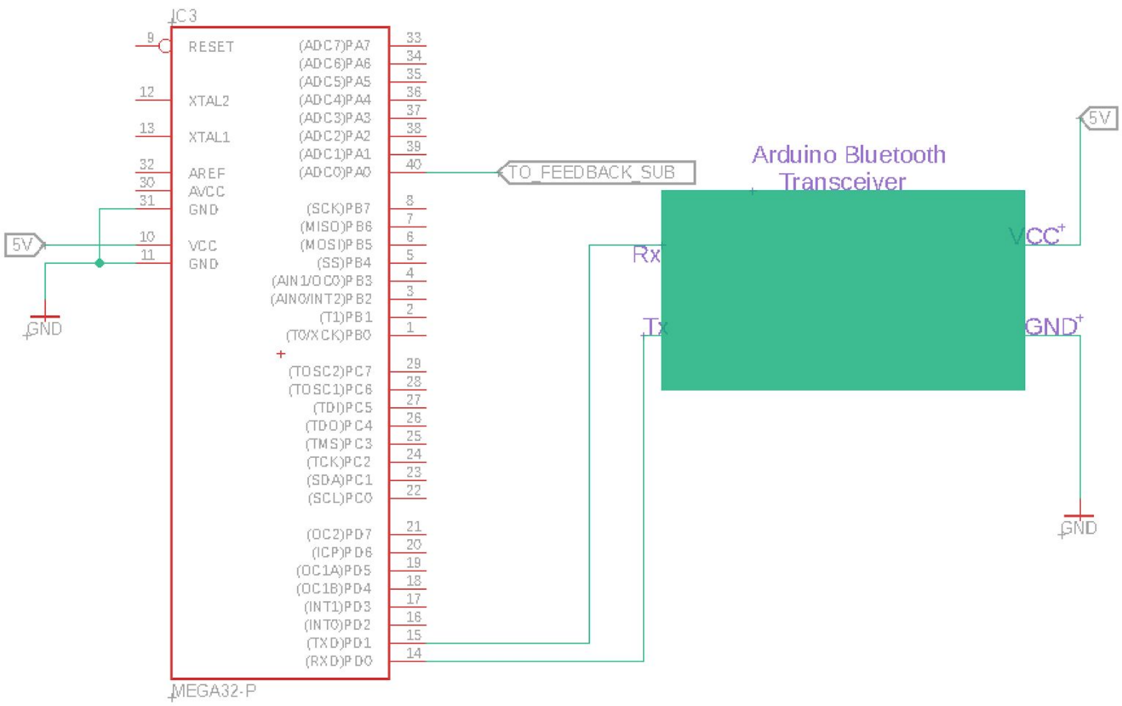


Figure 10: Control system Schematic

2.5 Computer Vision Subsystem

2.5.1 Raspberry Pi 3

The RaspberryPi will be used to process the data received from the camera module and run a realtime multi-person 2D pose estimation algorithm to detect and provide feedback of a user's current sitting posture [9]. To implement this algorithm, we will utilize OpenCV's Human Pose Estimation to track all the segments that are essential in determining the posture (segments are listed in the high level requirements). We will implement additional logic in the algorithm to determine whether the existing posture read from the Raspberry Pi is within the threshold of acceptable posture. Once the microcontroller is done processing the posture, it will send information to our software subsystem for data analysis and relay signals to the feedback subsystem to tell the user of his/her current posture status.

OpenCV's Human Pose Estimation takes the input image and predicts the possible locations of each keypoint in the image with a confidence score [11]. Then, the algorithm has stages that take the image data and the confidence map of the previous stage to use the context of the nearby area from the previous stage. The prediction of keypoints improves after each step, and therefore the confidence map is more accurate after passing through all 4 stages. Finally, after all the key points have been found, the algorithm connects the indices and draws the skeleton.

Requirement	**Verification**
Must be able to accurately detect segments of user posture and move accurately in accordance to the user in < 500 ms.	1. Ensure that the data is being loaded 2. Measure the runtime of the algorithm
Must be able to determine if a posture is good or poor with at least 90% accuracy	1. Sit on the chair and attempt many different postures, including good and bad 2. Count the number of good postures and the attempted postures 3. Ensure the percentage of good posture is over 90%
Must be able to efficiently collect and encode data at every 1 minute interval then send to the computer for data analysis. Upon reading the analyzed data, the module must be able to send signals to notify the user.	1. Sit on the chair and attempt a good posture 2. Once you have a good posture, attempt a bad posture 3. Code in microcontroller program so that it captures the time when the posture goes bad. 4. Attempt in real-time with own stopwatch to estimate that the latency is less than 700ms and compare with computer-captured value.

2.5.2 Camera Module

We will attach the camera module to our microcontroller to record and compare the user's current sitting posture in respect to the chair. This will be placed on the side to see all essential segments of a user. We will be using a 5-megapixel camera as this camera's high quality will allow our computer vision algorithm to accurately analyze the user's posture.

Requirement	**Verification**

Must detect the user's position from head to toe with at least 2 inch of buffer space on all sides.	1. Position camera on the same level as the arm rest and move a few feet back until the camera captures everything. 2. Take snapshots and assure buffers on all sides.
Must be on and responsive whenever the whole system is active.	1. Check power indicator when utilizing chair.

2.6 Software Subsystem

2.6.1 USB-to-MicroUSB

USB-to-MicroUSB cable will be used to power the Raspberry Pi with the computer as well as send posture data to the user's PC.

Requirement	**Verification**
Must be able to access posture data from the Raspberry Pi	1. Graph data in real time on computer and see if graph matches (ie. when the user slouches, it should show up on the graph)
Must provide 5 V ± 5% for the Raspberry Pi	1. Check to see if status LEDs light up on Raspberry Pi

2.6.1 Visual Display

Our software subsystem is mainly responsible for displaying data to the user in a way that describes their seating posture over time. We will be using Ruby on Rails as the framework to build our web application. This project heavily depends on the data measurements from the sensors, thus we'd figured this development tool would be ideal because of its simplicity and efficiency to abstract these measurements. Using the collection of data, we will implement features on the web application such as frequency of maintaining good posture throughout the day and identifying weak points from the sitting posture.

Requirement	**Verification**
Data can be displayed on the web server within 1s of the data being collected.	1. Data will be gathered from the pressure sensors and ultrasonic sensors. 1. Check the data is in the web server in real-time
Data must be collected and stored in timed intervals to show gradual progression of the user seating posture.	1. Have the user sit casually on the chair for an extended period of time (minimum 4 hours). 2. Using a timer to let the user know the system is taking a snapshot of the current posture, read the data and see if there are any outliers. 3. Observe the web application to see if there are snapshots being updated/added on schedule every timed interval.

2.7 Feedback Subsystem

2.7.1 Speaker

This subsystem will be used to alert the user in real time when their posture has been classified as poor. The volume will be minimal but also noticeable so that it will not annoy the user.

Requirement	**Verification**
The volume should be minimal (< 45 dB) but noticeable while sitting down on the chair.	1. Attach the speaker on the chair. 2. Sit on the chair and determine if the sound is noticeable. 3. Use a decibel meter to observe how loud vibrations are.

The subsystem should correctly respond to the microcontroller's signal 100% of the time (beep when turned on and make no noise otherwise)	1. Run a series of test trials by manually setting the Microcontroller I/O pin to toggle between high and low 2. Check the audio feedback. 3. Run 100 trials and ensure that the speaker behaves as expected

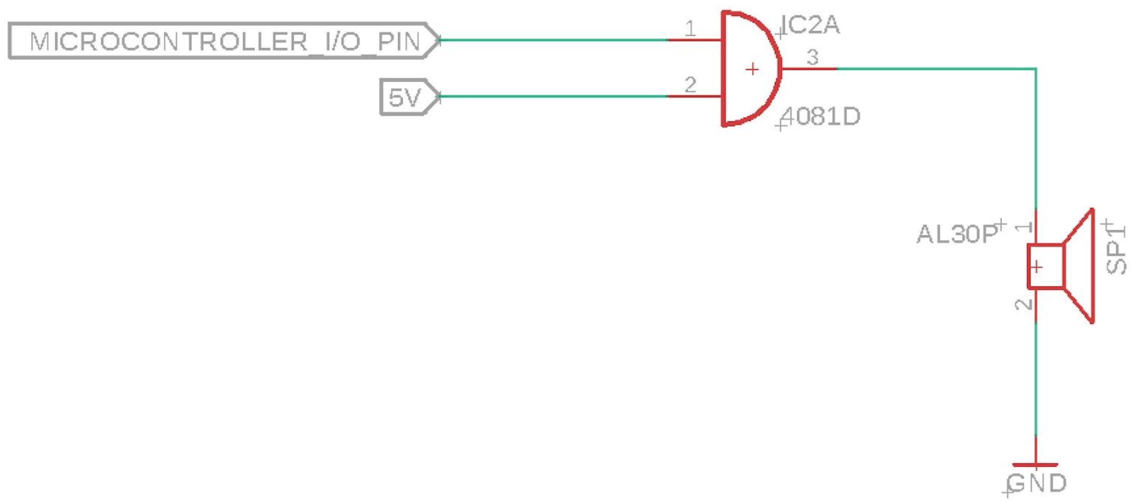
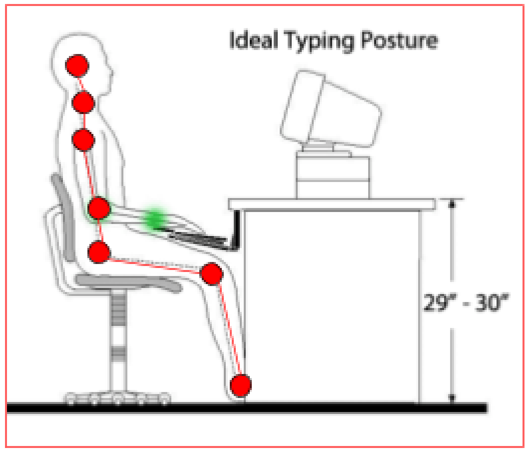


Figure 11: Schematic for feedback subsystem

2.8 Tolerance Analysis

To determine how applicable our design is to the real world, the most vital tolerance that we have to uphold is the analysis of essential joint segments by the computer vision algorithm. There are two major points that are considered from using computer vision to prevent any possible user injuries. Before proceeding to read the status of a person's sitting posture, the computer vision subsystem must be able to detect the following major points from a side view:

1. Head
2. Neck
3. Shoulders
4. Lower back
5. Buttocks
6. Knees
7. Feet



[Essential segments for analysis are marked in red circles]

Since there is no ideal model to pinpoint all joints due to the nature of different body physiques, we utilize the OpenCV Deep Learning Based Human Pose Estimation's suggested keypoint detection datasets[10]. Specifically the COCO (Common Objects in Context) dataset that utilizes over 200,000 images and 250,000 person instances can produce 18 points of the human segments with high accuracy. Since we do not have the equipment necessary to examine the tolerance analysis with our own sample data, we evaluate the OpenPose library derived from our algorithm to determine the accuracy of the datasets.

COCO uses the Object Keypoint Similarity (OKS) metric that measures how close the predicted keypoint is with the ground truth. In the context of our project, how far is the segment read from the computer vision subsystem to the actual segment.

COCO uses mean Precision and Recall as primary metrics: Precision measures how accurate predictions are correct, and recall measures how good all positives are located. The mathematical definition is as defined:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

TP = True positive

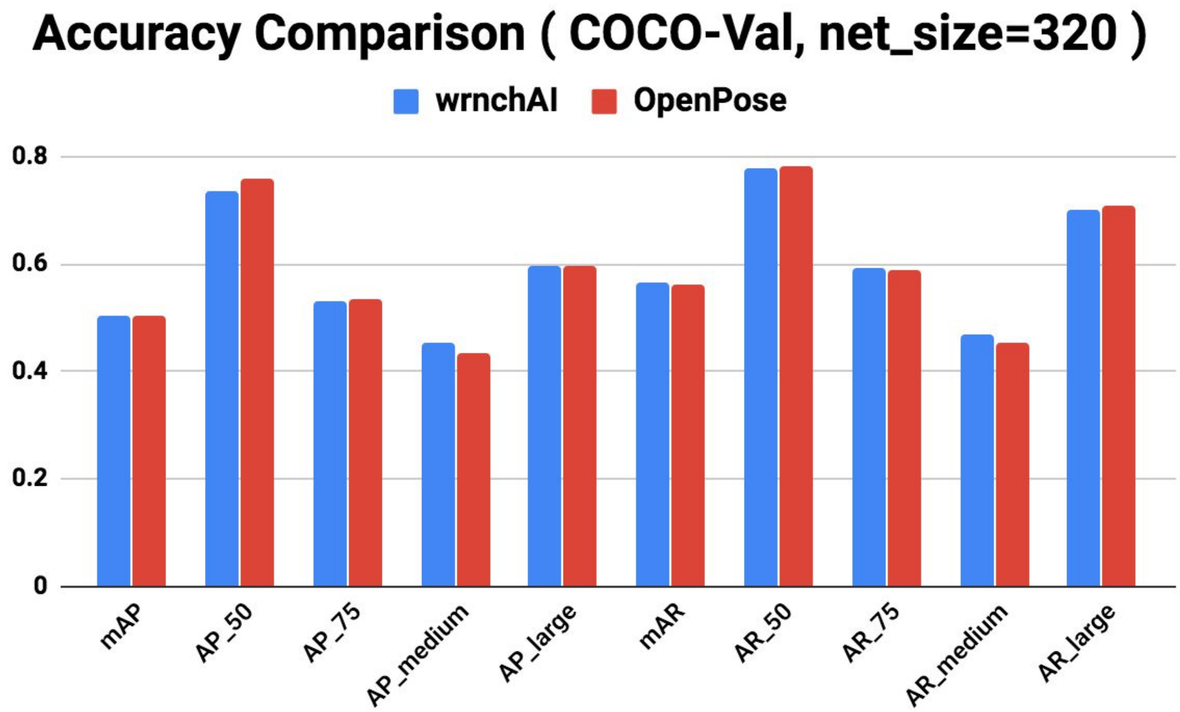
TN = True negative

FP = False positive

FN = False negative

The Precision and Recall metrics are measured at 50% and 75% (AP_50, AP_75, AR_50, AR_75) and Precision and Recall for medium(AP_medium, AR_medium) and large objects (AP_large, AR_large).

Conducting the experiment to seek validation on the accuracy and threshold of the data, the results are as follows:



*We disregard the wrnchAI comparison as the resource for computer vision human posture estimation utilizes the OpenPose library.

From the results of the COCO experiment, we see a 3%-5% tolerance in what we consider an accurate joint detection.

Upon determining good posture from the detected joints, we will run a trial of calibration tests to determine a standardized model of good posture per user. The user is expected to sit in a comfortable yet uphold a good posture for proper readings. Once the calibration is finished, the data will be recorded and averaged out to form a model to compare with future posture from regular use. Angles will be calculated from one segment to another. Due to natural movements on the chair, we will allow a 5-10% threshold in degrees between each segment to be considered as proper sitting posture for the users.

3 Solution Comparison

Difference Overview	Analysis
The most important difference between our two designs is the types of	While changing our design to use a camera for sensing has its trade-offs, we believe that it will ultimately allow for more accurate classification as well as more specificity in the feedback to the user.

sensors used to determine the user’s posture. While our original design utilized both Force Sensitive Resistors and Ultrasonic Range Sensors, our improved design uses a camera and computer vision to gather data about the users posture.	Under our previous approach, posture classification was totally based on initial calibration data from the user. The user would initially perform the calibration by sitting with good posture while the system took a series of measurements. From there, our previous algorithm would classify the users posture as bad whenever the data strayed too far from the initial measurements. While this approach would likely be fairly effective in deciphering between good and bad posture, it wouldn’t be able to tell the user what they need to correct. The computer vision approach, on the other hand, measures relative angles between key points on the user's body and therefore would be better at localizing the problem. If the user’s shoulders rolled forward, for example, the original approach would likely be able to catch this as the range sensors would show higher values that they initially did during calibration but it would not be able to distinguish which part of the body is causing this. The computer vision approach on the other hand would be able to recognize that the angles corresponding to the head and shoulders do not align with their proper values and keep track of this in the web application. With this extra information, the user would better be able to track their weak areas so they know what they should be focusing on throughout the day. Another benefit of using a camera instead of sensors on the chairs is increased comfort. With sensors removed from the back and bottom of the chair, the user will no longer have to feel these indents all day long. Furthermore, since placement of sensors no longer plays a role, our design could reasonably be used on any chair. This is a much more cost efficient as the chair itself was one of the biggest costs of our previous design. One potential drawback of using computer vision is that the design's processing power requirements have greatly increased as video processing is known to be a computationally heavy action. This is not a huge concern for us because this processing is done on the raspberry pi which is powered by USB, so there are no worries about battery drain. In addition, from the average user’s perspective, it would likely not make much of a difference.
Unlike our previous design for the chair which drew power from a standard 120V outlet, our new design uses a Li-ion battery instead.	We decided to use a Li-ion battery instead of using an outlet because using an outlet requires cords that could prevent the chair from moving freely in an open space. This makes our design far more attractive from a user’s perspective, allowing them to roll freely in their workspace.
Our feedback system in our updated design now includes a speaker that makes a small	We feel the speaker has benefits over the LEDs because the LEDs would likely be difficult to see because of their position on the chair. Unless they are bright enough to be noticeable in the user’s peripherals,

beeping noise when the user's posture is poor as opposed to LEDs and vibration motors present in the previous design.	they likely wouldn't ever be seen throughout the day. In addition, using a speaker is an improvement on the vibration motors because it would likely be more noticeable that small vibrations in the chair. Being more noticable, the user is more likely to fix their posture to make the noise stop. While some might see this as bothersome, we think it is important to prioritize results and effectiveness.
Finally the last major difference is the way in which information flows from the chair to the other subsystems. In the previous design, all processing was done on the PCB mounted to the chair and then sent to the user's PC via USB. Our new design now does all the processing on the raspberry pi which is not mounted to the chair.	Similarly to using a Li-ion battery, this eliminates the need for another cord going from the chair to the computer. It does however create a need for wireless transmission of data to the chair through bluetooth for the feedback subsystem.

4 Cost & Schedule

4.1 Cost Analysis

Tentative Supply Cost Chart

Parts	Quantity	Price
Chair	1	\$62.99
Linear Voltage Regulator - 5V L7805CV	1	\$0.50
Linear Voltage Regulator - 3.3V L7833CV	1	\$0.50
Mini Speaker COM-11089	1	\$1.95

ideapro USB Bluetooth Adapter, 4.0 Dongle Micro Bluetooth Transmitter	1	$9.90
LED lights	12	$5.16
HC-06 Wireless bluetooth Transceiver	1	$2.99
ATMega328	1	$1.86
USB to TTL Adapter - FT232RL	1	$12.76
USB Camera	1	$38.99
Total Cost:		**$137.60**

Goals	**Estimated hours**
Installing necessary components to chair + considering design placements	10
Designing, building, and testing PCB	20
Web application + data analysis features	40
Forming successful data stream architecture	15

The workload would be split between us three as even as possible so that we try to avoid some members working harder than others. Therefore, the estimated salary for the labor between three people is roughly the equal. Since the average salary of a ECE graduate is $80,000/yr which is roughly $40 an hour, that is what we are planning to project until completion of the product.

4.2 Schedule

Week	**Steven**	**Geonil**	**Brian**
1	Research algorithms and resources needed for computer vision subsystem	Confirm and order necessary parts	Research connectivity between subsystems (Data transfer, voltage, etc)
2	Implementing/testing code	Develop web framework + parsing data from computer vision	Design and ordering PCB components
3	Optimizing code + testing Raspberry pi	Data visualization on webserver + connectivity between raspberry pi and feedback	Integrate components to chair and verifying compatibility
4	Group meeting: expressing concerns, assistance, briefings	Group meeting: expressing concerns, assistance, briefings	Group meeting: expressing concerns, assistance, briefings
5	Testing chair and function capability	Testing chair and function capability	Testing chair and function capability
6	Algorithm optimization	Web optimization	Debugging
7	Algorithm optimization & Debugging	Web optimization	Debugging
8	Debugging & Testing	Debugging & Testing	Debugging & Testing

5 Ethics and Safety

One of the biggest issues we need to be careful with is the IEEE Code of Ethics #9 which states to avoid injuring others by false or malicious action. Our design uses data from the Raspberry Pi and computer vision to analyze one's posture. Therefore, we need to accurately make measurements and analyze these values in order to prevent providing misinformation to the user and possibly aggravating back or neck injuries. In order to prevent these injuries, we plan to perform some tests by sitting on the chair with good and bad postures and checking if our Raspberry Pi provides the correct data.

Another important risk we need to address is the risk of electrical circuits and components that are placed on a chair where someone will sit on daily. Because we are designing this project with a camera module that is placed away from the chair instead of sensors that are directly attached, we are able to make this chair a little more safe and comfortable for a user who will be sitting on for multiple hours. However, we still need to be careful with PCBs and other small electrical components that will be placed on the chair. By ensuring that nobody will be endangered by sitting on this electrical chair with enough insulation and protection, we believe that our Posture Sensing Smart Chair is in compliance with the IEEE Code of Ethics #1 [10].

We believe that our design follows the ACM Code of Ethics 1.1 in that it contributes to society and human well-being [8]. Like it states, our design supports improving people's poor posture which is an important problem in society today. As the world is becoming more technology-based, people often tend to spend a lot of time sitting on their chair and work in their offices for several hours. Our design contributes to the society by providing feedback to their poor postures and lowering the risk of back or neck injuries.

We also intend to follow the ACM Code of Ethics 1.6: respect privacy. We will not disclose any information or data gathered through this design or share them with any other parties. The data gathered through the chairs will only be for personal use and knowledge.

References

[1] Wang, C. (2020). *Good Posture and its Wealth of Benefits to the Workplace*. [online] Pdfs.semanticscholar.org. Available at: https://pdfs.semanticscholar.org/7755/d5c48864b44937a639fc3c72f4dd3d4d63df.pdf [Accessed 13 Feb. 2020].

[2] Acatoday.org. (2020). *Posture*. [online] Available at: https://acatoday.org/content/posture-power-how-to-correct-your-body-alignment [Accessed 13 Feb. 2020].

[3] Simpson, L., Maharaj, M.M. & Mobbs, R.J. The role of wearables in spinal posture analysis: a systematic review. *BMC Musculoskelet Disord* 20, 55 (2019). https://doi.org/10.1186/s12891-019-2430-6

[4] wikiHow. (2020). *How to Sit*. [online] Available at: https://www.wikihow.com/Sit [Accessed 28 Feb. 2020].

[5] Web.eece.maine.edu. (2020). *Product User's Manual – HCSR04 Ultrasonic Sensor*. [online] Available at: http://web.eece.maine.edu/~zhu/book/lab/HC-SR04%20User%20Manual.pdf [Accessed 28 Feb. 2020].

[6] Ww1.microchip.com. (2020). *8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*. [online] Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf [Accessed 28 Feb. 2020].

[7] "IEEE Code of Ethics," *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 28-Feb-2020].

[8] "The Code affirms an obligation of computing professionals to use their skills for the benefit of society.," *Code of Ethics*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 28-Feb-2020].

[9] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," Cornell University, November 2016. [Online]. Available: https://arxiv.org/abs/1611.08050. [Accessed Apr. 7, 2020].

[10] V. Gupta, "Deep Learning based Human Pose Estimation using OpenCV," *Learn OpenCV*, 29-May-2018. [Online]. Available: https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/. [Accessed: 18-Apr-2020].

[11] "About OpenCV," *OpenCV*. [Online]. Available: https://opencv.org/about/. [Accessed: 18-Apr-2020].