# Small Wearable Analyzer of Gait (SWAG)

Jack M. Shaffery, EE, Alex H.R. Fanolis, EE, Eric A. Grimaldi, EE, and Scott T. Clark, CSE

*Abstract*—**The goal of SWAG (small wearable analyzer of gait) is to provide users of our wearable orthotic with analytical feedback of their specific gait parameters. Current gait analysis is not only expensive, but it can only be performed in a lab setting by a professional. SWAG is an affordable alternative, available to anyone who could benefit from gait analysis. Gait abnormalities affect people of all shapes and sizes, and even healthy individuals can benefit from gait analysis. SWAG will consist of force sensors attached to a shoe insole, coupled with an ankle housed module, which transmits data to a mobile Android device. The use of SWAG will help to identify gait parameters, identify gait abnormalities, and make suggestions to make improvements to a user's gait. It can calculate step times, swing times, cadence, center of gravity, pronation type, and more. SWAG will have applications in not only the medical field, but also as a retail sales device; steering runners towards the right types of shoes for their unique gait.**

## I. Introduction

Gait analysis is the study of the body's motions while walking or running, and specifically targets the movements and forces of the foot and ankle. Currently gait analysis techniques are expensive, usually costing close to $300 for a simple analysis [13]. Current techniques are also very limited; they can only be conducted in a calibrated lab setting by a trained professional. They usually require a series of images, or videos to be taken and analyzed. This process can take anywhere from 45 minutes, to 3 hours [13].

Unfortunately, these limitations are further exasperated by the relatively high prevalence of gait abnormalities in the population. While studies have exclusively been done on the elderly, and those with disabilities, it is accepted that people of all ages, and ranging degrees of health suffer from gait abnormalities. In a sample of noninstitutionalized adults aged 65 or older, 35% were found to suffer from gait abnormalities. [1] Studies of children and adolescents with attention deficit/hyperactivity disorders have found gait abnormalities in 50% of the subjects. [2] Even marathon runners use specific types of shoes to correct or improve over-pronation and supination. Gait analysis has also been used in orthopedic rehabilitation, and can decrease recovery time by allowing professionals to make suggestions to improve mechanics.

Because of the limitations of current gait analysis techniques, people have recently started turning to pseudo gait analysis for results. Many shoe stores offer a "Gait analysis" when you are purchasing running shoes. What this boils down to is an associate watching, or recording a customer walk, and visually characterizing gait to suggest a proper shoe to purchase [13]. These associates don't necessarily have proper training; this can lead to misdiagnosis or a walking gait type.

Making gait analysis more readily available can help to prevent misdiagnosis of conditions, prevent future injuries, improve rehabilitation time in certain orthopedic conditions, and even lead to early intervention in many common disorders. SWAG is an affordable alternative to current gait analysis techniques. SWAG consists of an insole, and anklet, and Android application to analyze gait parameters, and make a user aware of any abnormalities. The insole contains four piezo-resistive pressure sensors that collect a user's weight data at contact points on the foot. The anklet housing, contains a microcontroller capable of compiling the data and sending it to the Android application. The Android application houses the data processing and algorithms, which takes the raw data, and converts it to a representation of gait parameters. The Android application will give a visual representation of a user's step data, and can also display a heat-map representing the user's arch type. The parameters calculated by the application, will be compared to healthy averages, and show their deviation from these averages. Abnormal parameters, will be marked, and brought to the attention of the user.

## II. Design

### A. Overview

SWAG is composed of multiple subsystems that work in cooperation to complete our design as shown in Figure 1. Force distribution data is collected by piezo-resistive pressure sensors that will be placed in the insole of a shoe. These sensors are lightweight and low profile enough to the point where it would not affect a normal walking motion. The calculated voltage data will be sent from the sensors to a microcontroller, which is located in an on ankle housing. Also found in the ankle device is the power supply to provide power for the sensors, as well as the Bluetooth transmitter [10]. The power supply meets the requirements of portability, power distribution, and battery life. It will communicate with a mobile phone in order to transmit the collected sensor data. The use of Bluetooth ensures a sufficient data rate. At the mobile phone is where the raw data is processed by a series of algorithms to compute necessary gait parameters. A functioning conversion of the raw data to a numerical representation of each parameter is an essential part of the working system. The Android app's graphical user interface will then present visual results of the calculated parameters. As a whole, the system predicts arch type and is able to predict, diagnose, rehab, or correct disorders for gait analysis in an affordable and efficient way.

SWAG System Specifications

| Specification | Value |
|---|---|
| Battery Life | 6.34 hrs (one charge) |
| External Circuit | <83 mW |
| Arduino | 232.5 mW (idle) |
| Max Power | 315.5 mW |

Table 1. System Specifications

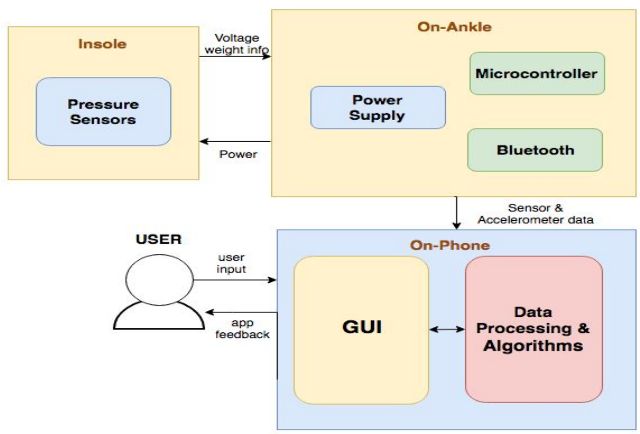


Fig. 1. Block Diagram.

*B. Pressure Sensors*

The user's force distribution across their foot will be measured using a set of pressure sensors in an insole that is to be worn within their shoe. The sensors that were selected were the Tekscan Flexiforce A201 [3], due to their 1mm thickness, power consumption below 2.5 mA per sensor, and accurate performance up to 100 lbs of applied pressure. As part of this block, a small op-amp powered drive circuit had to be constructed for each sensor so that they would output a linearly increasing voltage with increasing force, which depends on a designer chosen fixed resistance value within the drive circuit. The plots of output voltage vs. force is shown in Figure 3, from the Tekscan A201 data sheet, with the plots for various different fixed resistance values shown.

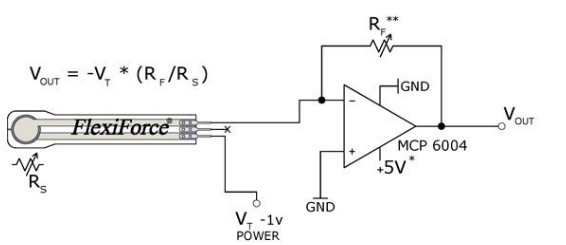


Fig 2. Tekscan recommended drive circuitry

The current design for this block implements four of the sensors at a cost of $65 per set of four, and the drive circuit, shown below in Figure 2, is comprised of parts available in the SDP lab, with the exception of the LTC 660 CMOS Voltage Converter, which is required for the drive circuitry and costs $.33 per single order. This block is preliminarily assembled already, although there still remains the issue of implementing the LTC 660 and reducing noise in the output voltage, so that the reading taken by the microcontroller is more accurate. Implementing the LTC 660 will allow the drive circuit to be taken off of a prototyping board, which may be partially responsible for the noise issues.

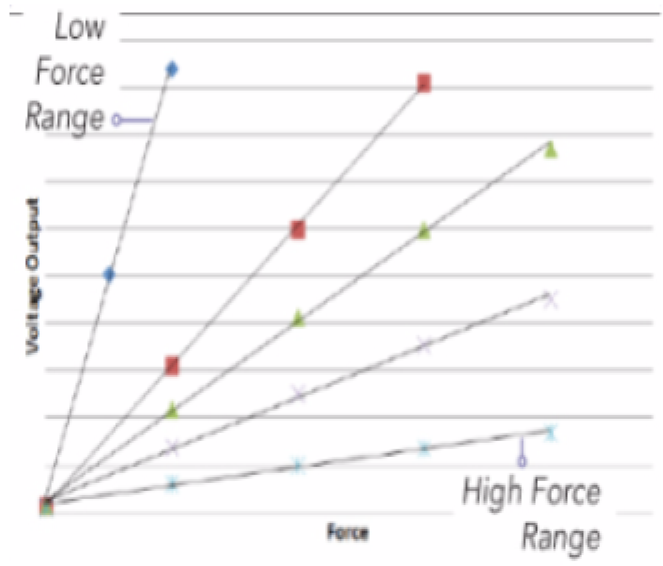


Fig. 3. Voltage output of the drive circuit as a function of the applied force. Note that the various plots are all depending on the chosen value of the fixed resistance value from the negative op amp terminal to the output, and can be used to select the sensitivity of the sensors.

As part of the design of this block, a calibration had to be done to learn the force-voltage relationship for the specific design setup. For this, various weights of known sizes were placed on the sensors and voltage readings were taken. These readings were then plotted to get a quantifiable value for the linear relationship between force and voltage, which could be programmed into the Arduino to be sent as date to the mobile app.

*C. Microcontroller*

The purpose of the microcontroller is to facilitate communication between the pressure sensors worn by the user in the insole and the Android application. The microcontroller reads the analog information produced by the sensors and converts it to digital information usable by the Android application. Additionally, the microcontroller then sends this information via Bluetooth connection to the Android application. Information is gathered and sent by the microcontroller upon request by the Android app.

For this project, the Arduino Uno Revision 3 [9] microcontroller was chosen. This model of microcontroller has an on board analog to digital converter. With this feature built in in the form of 6 analog input pins, it is extremely straightforward to read analog signals produced by the sensors and their driving hardware. Further, the Arduino has dedicated voltage output pins and ground pins which simplify the process of interfacing the sensor drivers and the battery which directly connects to the Arduino. While the Arduino Uno does not have built in BlueTooth capability, the JY-MCU Arduino Bluetooth Wireless Serial Port Module [10] adds simple and utilitarian BlueTooth capability to the Arduino's serial port connection. The fact that the Arduino uses its serial port to communicate with this module allows for straightforward implementation of BlueTooth communication. Both the on board ADC and the JY-MCU Bluetooth module utilize the standard functions in the Arduino's programming library. This allowed for quick integration of the sensors and BlueTooth into the microcontroller block. Additionally, this feature allowed for quick and flexible prototyping and adjusting of the microcontroller block to meet the needs of the project.

Largely, this block of the project falls within the scope of an Embedded Control class Eric had taken at his previous institution. An understanding of basic control theory and previous experience with the Arduino programming environment are the main tools used in the design of the microcontroller stage.

Initial prototypes of the microcontroller block were built using a Raspberry Pi 3. This option seemed attractive due to the on board BlueTooth hardware. However, in practice the Raspberry Pi was not a great fit for the project as it lacked native support for reading analog signals [11] (and thereby requires that we build additional hardware which could result in increased power draw and possible reading error), and its method of interfacing with BlueTooth required much more cumbersome software development than the Arduino.

### *D. Power Supply*

There is a sole power supply implemented in the design, for the microcontroller, which powers the sensors, the processing done on the microcontroller, and the Bluetooth communication with the user's phone. Research indicated that for many projects that use a microcontroller, especially ones meant as a prototype, a simple power bank, usually used to charge cell phones, is used.

The phone power bank was chosen for this design because they cost uniformly less than $20, have a low profile for the ankle housing, and have built in circuitry to regulate the maximum voltage and current supplied, which for the microcontroller chosen cannot exceed 5V at 2.5A, the specified maximum for almost all mobile power banks. This eased the design process considerable, as a power supply circuit did not need to be devised, saving time and space on the ankle housing.

For this design, a 2 Ah power bank will be used. Measurements were taken to determine that the maximum current draw from all four sensors, even at heavy force application, does not exceed 10mA. Assuming less than 10mA draw from both the 3.3V and the 5V pin on the Arduino Uno provides a maximum power consumption of 10mA * (5V + 3.3V) = 83mW. In addition, the idle power consumed by the microcontroller, 232 mW, was taken into consideration to determine a maximum power draw of 83 mW + 215 mW = 315 mW. With a 2 Ah, or 7,200 J battery, this means a battery life of, at worst, 7,200 J/315 mW = 6.3 hours.

### *E. Algorithms*

Housed in the Android applications, our data processing takes the raw sensor data collected by the microcontroller, and converts it into useful gait parameters. Examples of such parameters are: step times, swing times, cadence, center of gravity, pronation type, and arch type. While these are not all the parameters that are calculated, they are some of the more important ones. An important aspect of the data processing is filtering of non-step data. When analysis begins, a user will be instructed to hit a "start analysis" button on the Android app, and then begin walking at a self-selected pace. After taking a set amount of steps, they click "end analysis." At the beginning and end of this data, there will be some sampled data that is not representative of a step, so by searching the data for the first instance of the sensor situated at the heel switching from unloaded to loaded, we know that is where steps begin. Intuitively, you can find the last instance of the toe sensor switching from loaded to unloaded, and that is representative end of the last step. Searching between these two instances, we can identify all the heel strikes and toe offs, and consider the timing of these in relation to each other. We can use these to calculate step times, swing times, and cadence.

One of the more advanced parameters to study is the center of gravity (CoG). We will set the 4 force sensors at set contact points on the insole. We can overlay an x-y plane on the insole, and find the sensors position of this plane. Using these locations, stored in the algorithms, we can calculate the CoG using the following equations:

$$\text{CoG}=(x,y)$$

$$x=\sum_{i=1}^{4} F_i x_i + x_i$$

$$y=\sum_{i=1}^{4} F_i y_i + y_i$$

Fi in these equations is a representation of the force applied to a specific sensor, and xi, and yi, are the x and y coordinates of these sensors respectively. We can use this to visually represent a user's CoG. Based off this CoG data, we can also determine a user's pronation style. The results of these algorithms is the compiled, and sent to the GUI, for visual representation.

### F. Graphical User Interface

The Android application provides an easy to use, clean, and visually appealing interface for the user to interact with. Some of the main goals are to graphically represent data, show progression over time, and provide feedback. The app includes views for arch type, force distribution over a single capture period, and graphs of individual parameters. It is equipped with a menu screen to provide navigational functions.

The arch type will be will be represented by a color-coded heat map. This is generated by taking the values at each sensor, and then using a pre-determined algorithm to approximate the points in between. The force distribution over the time of a few steps will be shown by an animation. The animation will have a color for each sensor, similar to the heat map, at the given time in the walking cycle. Graphs will be used to show the users' results of each parameter compared with the average.

Another notable feature is that it will have a different view for a healthcare professional, as the patient will most likely not have much knowledge about the data being given to them. The patient can send their results to his or her doctor for review. The expert can then provide feedback to the patient based on what he or she sees. The app will also show the user the status of the connection to the Bluetooth device.

Courses we took have provided a solid understanding of the software development process, including techniques and common practices. They also taught how to write clear, maintainable, high-quality code. Mobile application development, however, was not touched upon. Therefore, extended research and prior experience, such as internships, will be essential. Online tutorials and code samples were good sources for learning.

Java programming knowledge, as well as mobile application building skills will be required. The app will be built in Java with the use of Android Studio [12]. Android Studio offers a Java IDE with all the tools necessary for development. The app can be tested through the use of Android Studio's built-in debugger and Android emulator.

## III. Project Management

Shown below in Table 2 is the list of MDR deliverables followed by whether each one was completed by the MDR presentation. As can be seen, all of the promised MDR deliverables were completed on time. The largest remaining design challenges lie in both establishing Bluetooth communications between the microcontroller and the user's phone and establishing a useful graphical user interface that implements the algorithms to produce the desired user experience.

The group members each chose a block that aligned with their area of expertise. Scott Clark, the team's CSE major took the task of designing the graphical user interface. Alex Fanolis, who has experience working with gait analysis in commercial applications, took on the task of researching and implementing algorithms to analyze gait. Eric Grimaldi has worked with Arduinos and printed circuit boards in internships in the past, and as such is working on the majority of the ankle housed block. Jack Shaffery has had prior experience working with analog sensors and measurement tools, as well as soldering and other skills required for designing and creating the required drive circuitry for the sensor and insole design.

Communication was vital in the progress of the project, with most communication occurring via text the mobile app WhatsApp. Here the group members were able to respond to one and others ideas and questions in real time, and they did so frequently to help keep progress on track. Resources and mutual projects like the MDR presentation slides were consolidated in a folder on Google Drive where all members could access and work on different materials together.

The team is functioning well up to time of writing, and they continue regular meetings with their advisor, UMass ECE professor, Aura Ganz. She has been indispensable in the success of the project up to now, providing guidance and suggestions to further progress as the team worked.

| MDR Deliverable | Completed? |
|---|---|
| Identified and Purchased Hardware | YES |
| Establish communication between sensors and microcontroller | YES |
| Identify specific power source | YES |
| Research algorithms for specific gait parameters | YES |
| Rough draft of graphical user interface | YES |

Table 2. Promised MDR deliverables, marked YES or NO to indicate whether they have each been completed or not completed

## IV. Conclusion

Currently, prototypes of each block are assembled. Basic functionality of the pressure sensors has been achieved on a prototype circuit board. Communication between the microcontroller and the sensors has been reliably established, but communication via Bluetooth has only been implemented with a basic placeholder program and not with the Android final application. The final power supply has been chosen. General algorithms for the majority of measured parameters have been written. A test version of the Android application has been developed, though it currently uses mock data and does not communicate with the microcontroller.

Plans for the future include finalizing the circuitry in support of the sensors in the form of a custom printed circuit board, fully establishing the BlueTooth communication between the microcontroller and the Android app, tuning the algorithms and integrating them into the Android application, creating a final anklet housing and insole for the device, and completing overall end to end integration of the final system.

The majority of the work left to be done is in the vein of finalization and integration. In other words, the majority of the difficulties remaining lie in seamlessly the separate subsystems together into one smoothly functioning whole. Bugs are bound to arise when all the parts begin to depend on each other. These will need to be ironed out by rigorously testing the system and iterating on the design until it runs without hiccup.

## Acknowledgment

We would like to thank Professor Hollot for his feedback and guidance throughout the project, especially in regards to setting appropriate goals. We would also like to greatly thank Professor Ganz who took the time to meet with us each week and kept us organized and on the path to success.

## References

[1] https://www.ncbi.nlm.nih.gov/pubmed/16460376
[2] https://www.ncbi.nlm.nih.gov/pubmed/18963991
[3] Tekscan, 'Flexiforce A201 Sensor'. *Tekscan*. N.p., 2014. Web. 15 Dec. 2016
[4] B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.
[5] E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.
[6] J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted for publication.
[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style),” *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].
[9] https://www.arduino.cc/en/Main/ArduinoBoardUno
[10] https://core-electronics.com.au/attachments/guides/Product-User-Guide-JY-MCU-Bluetooth-UART-R1-0.pdf
[11] https://www.raspberrypi.org/documentation/hardware/computemodule/RPI-CM-DATASHEET-V1_0.pdf
[12] https://developer.android.com/studio/index.html
[13] https://www.pennmedicine.org/for-patients-and-visitors/find-a-program-or-service/physical-medicine-and-rehabilitation/gait-and-biomechanics-laboratory