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 for improvement of a users gait. It can calculate step times, swing times, cadence, center of pressure, as well as give a visual representation of 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 of 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 gives a visual representation of a user’s step data, as well as parameter results. 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 does not affect a normal walking motion. The calculated voltage data is sent from the sensors to a microcontroller, which is located in the 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 communicates 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. The raw data is turned into a numerical representation of each parameter. The Android app’s graphical user interface then presents visual results of the calculated parameters. As a whole, the system alerts the user of any deviations from healthy averages, and shows the user a playback of their center of pressure. These results can help diagnose issues, as well as help the user find the proper footwear for their gait style.

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

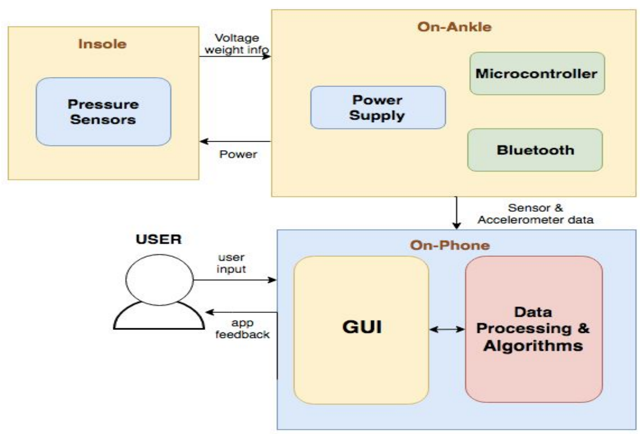

Fig. 1. Block Diagram.

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.

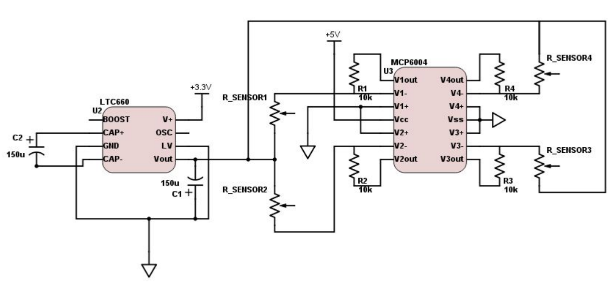

Fig 2. Final drive circuitry schematic

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 assembled on a printed circuit board and is completely portable and integrated within the on-ankle housing with the Arduino Uno and power bank. The full implemented housing and ankle design are shown below in Figure 4, with the insole in the user's shoe.

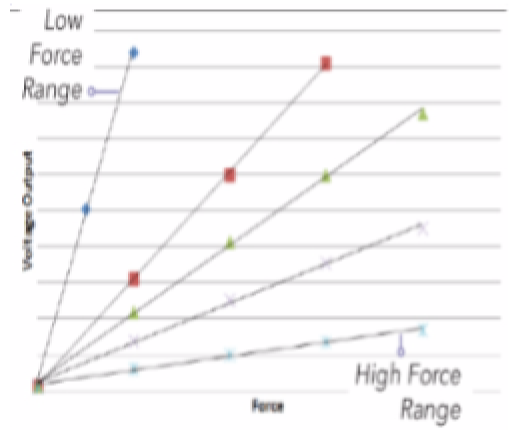

Fig. 3. Voltage output of the drive circuit as a function of the applied force. Note that the various plots are all depending on the chosen value of the fixed resistance value from the negative op amp terminal to the output, and can be used to select the sensitivity of the sensors.

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.

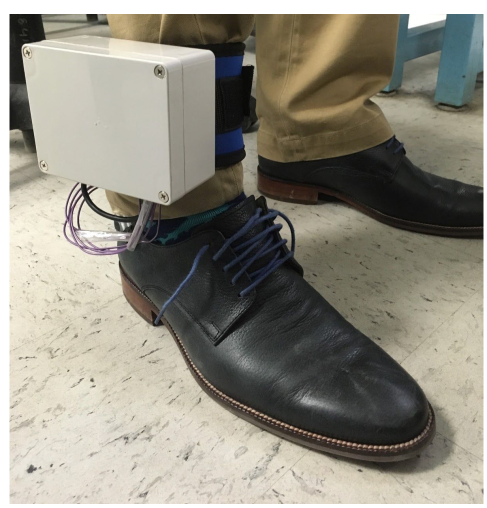

Figure 4: Implemented ankle housing and insole in use

C. Microcontroller

The microcontroller facilitated communication between the pressure sensors worn by the user in the insole and the Android application. The microcontroller read the analog information produced by the sensors and converted it to digital information usable by the Android application. Additionally, the microcontroller sent this information via Bluetooth connection to the Android application. Ultimately, the microcontroller would gather information and send it to the Android app upon request.

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 was extremely straightforward to read the analog signals produced by the sensors and their driving hardware. Further, the Arduino had dedicated voltage output pins and ground pins which simplify the process of interfacing the sensor drivers and the battery which directly connected to the Arduino. While the Arduino Uno did not have built in BlueTooth capability, the JY-MCU Arduino Bluetooth Wireless Serial Port Module [10] added simple and utilitarian BlueTooth capability to the Arduino's serial port connection. The fact that the Arduino used 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 utilized 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) = 83 mW. 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 pressure, and pronation 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 is instructed to begin walking at a self-selected pace. After taking a few 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 pressure (CoP). 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 CoP using the following equations:

$$CoP=(x,y)$$

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

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

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 CoP. Based off this CoP 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 graphs of individual parameters last 10 samples, as well as Center of Pressure playback over the most recent sample period . It is equipped with a menu screen to provide navigational functions.

After taking a sample, the user is navigated to a results page. There are two blocks for results; the first for cadence and the second is a comparison of step and swing times. The color of the boxes change based on the results. Green means the user was within a healthy range of the parameter, and red means they are outside the healthy range.

Clicking on a parameter results in a graphical display of the user's last 10 samples. The graph updates every time a new sample is taken and can give the user and idea whether or not they are progressing or possibly regressing.

There is another option called “Center of Pressure.” Using the calculations from the algorithms, and Android Studio’s plotting tools, our app provides a playback of a user’s CoP over their last recorded sample. There are two playback modes, fast and slow, to help to interpret this data. Interpreting the trends of the CoP can lead to a diagnosis of pronation style, so this is a very powerful part of our app.

Finally, we have a heat map option inside the application. Similar to the CoP option, it will playback the most recent sample. The heat map shows a representation of the pressures on the foot over time. While it does not offer much scientific data, it is a fun visual representation, and it is interesting to watch the pressure roll from the back of the foot towards the front. We have noticed that the heat map playback struggles on android devices with less powerful processors, as there is a lot of calculation required to display.

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, were essential. Online tutorials and code samples were good sources for learning.

Java programming knowledge, as well as mobile application building skills were helpful. The app was 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.

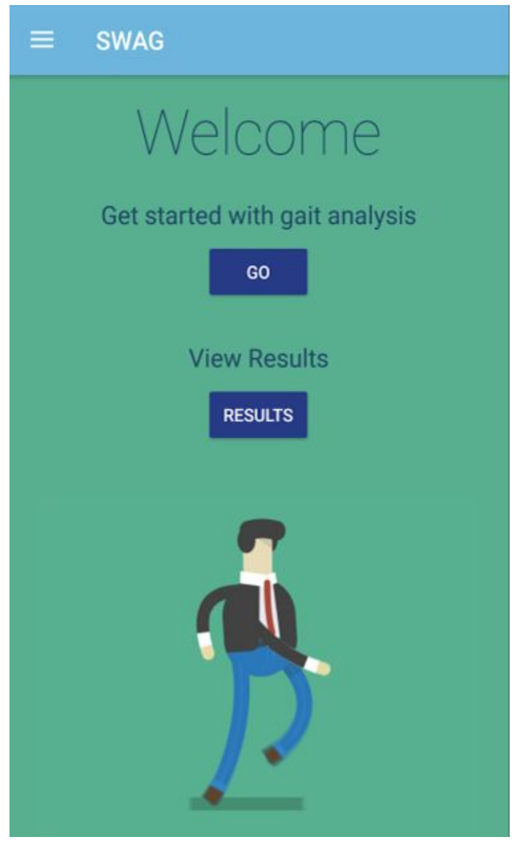

Figure 5: Home screen of the Android application.

III. Project Management

An open line of communication was most important to the completion of this project. Using a group messaging app, the group was able to constantly update each other as far as progress was concerned.

The team was comprised in a way that made splitting of responsibilities quite easy. Scott was the only CSE working on the project so he handled the majority of the application development. Alex is an EE and Math double major, and also had experience working with current gait analysis techniques, so he tackled the algorithm development. Eric, an EE, had previous experience with microcontroller coding as well as PCB design, so these responsibilities were given to him. Finally, Jack, an EE, has an expertise in analog electronics, so he was responsible for the power and drive circuitry design, as well as the design of the ankle housing.

Professor Aura Ganz kept the team on task by emailing and meeting weekly to check in on progress. Every week she suggested what the groups next milestone should be, and constantly checked in to make sure the group was making progress. She made it very clear to the group how much work would be required and always kept us on a path towards success.

IV. Conclusion

At the completion of the project, we accomplished a functioning end to end system with a very low rate of failure. The insole is easy to slide in and out of a shoe, and is robust enough to withstand all forces that would be applied. The ankle strap is comfortable and all the electronic components fit inside the plastic housing. The wires are not too long, but are not tight as to allow some room to maneuver the product.

Once the insole is inside the show and the ankle strap is attached, it is quite easy for a user to begin collecting data. Upon hitting the start button, the display switches to show a real time output of the pressure on each sensor at a given time. After taking a few steps, and hitting the stop button, the application switches to its results display. From here the menus are easy to navigate, and all results and instructions are clear and concise for easy user accessibility.

We have seen almost no signs of failure when taking a long enough sample using this product. We accomplished our goals of creating a quicker and cheaper alternative to current gait analysis techniques, and while are results are not quite as comprehensive as other technologies, they are accurate. When taking a smaller sample set (1-2 steps) there could possibly be some calculation errors, but as long as the user takes 3+ steps we have yielded great results.

Given more time, the inclusion of accelerometer and gyroscope data could add some incredibly useful calculations to this product, unfortunately it was outside of the scope of this project.

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, our advisor, who took the time to meet with us each week and kept us organized and on the path to success. We would like to thank our faculty evaluators, Professors Douglas Looze, and Patrick Kelly, for their constructive criticisms and feedback. Last but not least, we would like to thank Fran Caron and the entire SDP coordination team, without them, none of this would be possible.

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