Stride

John Higgins, EE, Joseph Menzie, CSE, Jarred Penney, EE, and Richard Hartnett, CSE

Abstract— **Individuals with Parkinson's Disease have difficulty moving routinely and, because of cost, may have limited access to medical care in the form of therapy. Stride is a low cost network of four sensor-based systems placed on body to analyze a user's gait and provide feedback. This product will be used in the home and will allow a patient to perform their gait training exercises outside of a clinical environment while still receiving the professional feedback needed. The design is made of four systems that collect and transmit lower body movement data to a central communication hub. This data is analyzed and real-time feedback is provided to the patient as well as long-term progress tracking through an Android application. The use of this product will allow a patient to work on their gait training exercises in their home as well as in their therapy sessions.**

I. Introduction

Individuals with Parkinson's Disease have an increased difficulty walking as the disease progresses over time. Although this disease is currently incurable, physical therapy has been proven to help slow the symptoms that come along with it [1]. The problem with this is, these therapy sessions are very expensive and insurance usually only covers two to three sessions a week. This means, a patient is limited in the number of times per week where they are able to interact with, and get feedback from, a medical professional. These limited sessions are often not enough therapy to effectively slow the symptoms that come along with this disease.

Parkinson's Disease (PD) is a disorder of the central nervous system that makes it difficult for an individual to move routinely. Worldwide there are about 10 million people living with PD and about 1 million of those people live in the United States [2]. Along with this, [2] estimates that a patient with PD spends about $2500 a year on medication and, if therapeutic surgery is needed, it can cost up to $100,000. The economic cost that comes along with this disease is often a roadblock for a patient trying to combat the PD symptoms.

There are several products right now that assist an individual with gait training. One of them, called Tekscan [3], uses a sensor-based insole to analyze an individual's gait. This product relays this information to an application where the patient can view their statistics. Another product, called XSENS shown in reference [4], is an array of wearable sensors that tracks a user's movements, relays the information to a computer application, and then displays the user's motion using a 3D figure. This product is more focused on full body movements of an individual rather than just their gait, but is still an effective tool when looking analyzing gait. There are two issues with these products. One, they are very expensive and not covered by insurance, and two, there is no real-time feedback. A patient only knows they performed incorrectly after the exercise is complete. This means they have to perform the exercise again to perfect it.

J. Higgins from Norwood, Ma (e-mail: jmhiggins@umass.edu).
J. Menzie, from Reading, Ma (e-mail: jmenzie@umass.edu).
J. Penney from Stoneham, Ma (e-mail: jjpenney@umass.edu).
R. Hartnett from Westport, MA (e-mail: rhartnett@umass.edu).

Our solution to this problem is Stride. Stride is a low cost array of sensors that monitor a patient's lower body movements and provide long term feedback, through an application, and real-time feedback, through error-indicating vibrations. This product monitors four different metrics, stride length, cadence, heel-to-toe motion, and number of freezes per exercise.

Requirement	Specification	Result
Provide real-time feedback	Feedback in less than 100ms	Feedback sent in <10ms *after every step is completed*
Provide accurate metric measurements	Within 10% error of actual stride length	Within 10% error 89% of the time
Lightweight product	Sensor systems (sensors, PCB, and power supply) less than 1 pound Waist Clip (Raspberry Pi and power supply) less than 1 pound	Shoe Sensor System: 14.7oz Ankle Sensor System: 11.3oz Waist Processing System: 10.2oz
Sufficient battery life	Battery life greater than 2 hours	Battery life of +10 hours

Table 1: List of system requirements and specifications.

Table 1 lists the system requirements and specifications for Stride. The main aspect of our product that sets it apart from other gait training products is the element of real-time feedback. The goal of this feedback is to allow a patient to know when they are performing incorrectly and then have the ability to correct their errors while still performing the exercise. This feedback is given within 10 ms of the step completing, giving the patient plenty of time to identify the vibration and focus on correcting the next step. Along with this, for the feedback to be accurate the measurements taken

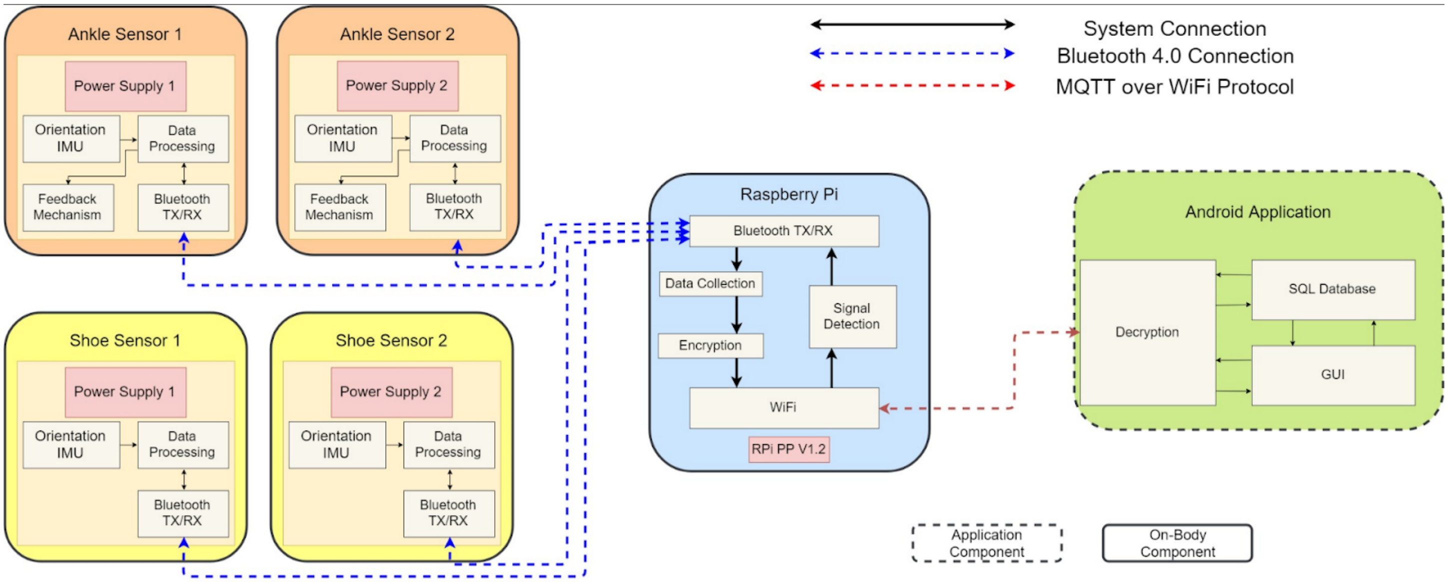

Figure 1: Block diagram for Stride. All the information from the sensors is relayed to the Raspberry Pi for analysis. Real-time feedback is provided by vibration components and long-term progress tracking is provided by the Android application.

and analyzed from the sensors need to be accurate. We spoke with Professor Boyer of the Kinesiology department at UMass and she mentioned that getting within 10% error for a product of this scale is a sufficient range [5]. The results of our product were measurements within 10% error 89% of the time. This was tested over a variety of heights and walking patterns to make sure to account for a general population.

Because of the symptoms of PD, for this product to be effective it needed to be lightweight and easy to put on. Our goal when developing this product was to keep each of the sensor systems fewer than one pound. After putting the product together we were able to meet this requirement. The shoe systems were 14.7 oz each including the shoe, the ankle systems were 11.3 oz each, and the waist clip with the Raspberry Pi was only 10.2 oz. By limiting the access weight we were able to make sure the added systems did not affect the gait of the individual. Along with this we made sure to make all straps velcro so they would be easy to put on and take off.

Our last requirement is that the product's battery life last more than two hours. Most therapy sessions last for two hours and therefore if the battery life is longer than two hours, it will be sufficient for a single session before needing to be recharged. We were able to surpass this requirement with ease. Our product has a battery life of 10+ hours ensuring it can be used for multiple sessions without needing to be charged.

II. Design

A. Overview

Stride will allow a patient to get around this financial roadblock by giving them the ability to perform their gait training exercises in the home, outside of the clinical environment, and still get the professional feedback they require. The block diagram shown in figure 1 shows the overall makeup of Stride. There will be four sensor systems located on the users lower half of the body. There will be one sensor system for each ankle and one for each foot. These systems will collect data and implement algorithms to assess the patient's gait. Along with this, after the data is processed in the sensor system, real-time feedback is triggered on the ankle systems during a workout. The systems will also transmit data wirelessly through Bluetooth to a Raspberry Pi, which will be located on the patient's hip. From there the data will be analyzed and sent over WIFI to the application backend for storage. Long-term feedback will be available through the Android Application.

A. Shoe Sensor System

The two shoe sensor blocks are designed in the same way and have the same functionality. Each block contains two additional orientation IMU Sensor blocks, a Bluetooth transmitter module, and a power supply. Each shoe has one *BNO055* [6] absolute orientation sensor mounted on the heel. The output data we use from the sensors is Euler angles, which give the orientation of the shoe in an angle from 0 to 360 degrees.. This data is relayed to the *hc-05* Bluetooth module, reference [7], and then transmitted to the Raspberry Pi for further analysis and for transmission of data to the application. In general, the *BNO055* needs to calibrate upon each new use in order to calculate accurate results. The sensor is fully calibrated when the accelerometer, gyroscope, and magnetometer are all fully calibrated. To allow an easier use for the user of our project, we calibrated the sensor once and imported the calibration offsets of the accelerometer, gyroscope, and magnetometer. Using this method, the sensor will be fully calibrated in under two seconds when the user turns the system on, and no special calibration methods are necessary for the user. In our workout, we program in for the user to wait five seconds before starting their workout.

A major improvement from MDR is the complete data transmission from the sensor to the Bluetooth transceiver to the Raspberry Pi. Initially, we attempted to transmit data from the sensor to the transceiver through I2C protocol, however I2C protocol and Bluetooth data transmission are not compatible. We discovered that the sensor is also able to communicate through UART by pulling the PS0 pin high, which then enabled the data to be received and transmitted by the Bluetooth module with the correct protocol. The Raspberry Pi is equipped with Bluetooth 4.0 so it is able to communicate with several *hc-05* modules simultaneously.

The data analysis is done on the *ATMega32u4*, which is on the PCB, mounted on the outside of the shoe. Using Euler angles, we are able to successfully measure heel-to-toe walking patterns and freezing. To measure heel-to-toe transfer we look at the Euler angle at heel-strike and toe-off, which is illustrated below in Figure 2. We define heel-to-toe transfer is the difference in angles at heel-strike and at toe-off.

$$\textit{Heel-to-Toe} = \textit{heelstrike} - \textit{toeoff} \quad (1)$$

Individuals with Parkinson's aim to increase their heel-to-toe transfer, as this corresponds to longer and more efficient strides. By tracking these values, an individual can set their baseline after a workout, and then track their progress against that baseline in future workouts.

Freezing is determined by an Euler angle staying the same for over a second. We allow a small threshold for the angle to change to account for noise and small variations that can occur as a result of weight shifts but not steps. Freezing is a recorded by the number of times one freezes in a workout. As the user progresses they can see if this number gets reduced over time.

B. Ankle Sensor System

The two ankle sensors are designed in the same way and have the same functionality. Each block contains one orientation IMU, a power supply, and Bluetooth transmitter. The same orientation IMU and Bluetooth module will be used on the ankle as were used on the shoe. The sensor programming and testing for the ankle-mounted orientation IMUs is the same as it is for the shoe mounted orientation IMUs, however, the data analysis is different. The ankle-mounted orientation IMUs will be responsible for determining stride length and cadence.

The data analysis on the ankle is achieved using Euler angles. First, for both stride length and cadence, we must determine when a step starts and when it stops. Our sensor is oriented on the ankle such that Euler angle in the "y" direction corresponds to 90 degrees when the leg is straight (perpendicular to the floor) and 0 degrees when the leg is parallel to the floor. Through experimentation, we determined that the angle of the ankle follows a sinusoidal-like curve as one walks. The range of angle values varies from person to person, and even from step to step for each individual person, however there is always the same pattern of angles when walking. When a step begins the individual bends their knee, causing the Euler angle to decrease from around 90 degrees to, for example, 30 degrees. Thus, by tracking the smallest angle and largest angle output from the sensor, we are able to determine when a step starts and stops.

To determine stride length, we moved away from the idea of using the double integral of acceleration during each step, as we found that there was far too much noise introduced by the sensor itself and as result of integrating. Instead, we used the smallest Euler angle found in each step, as the angular extent of the leg corresponds to the length of each stride [8]. Our initial stride length calculation is shown in formula (1)

$$S = L\cos(\Theta) \quad (2)$$

Where L is the length of the individual's leg from the hip trochanter to the ankle, and Θ is the angular extent of the swing phase at toe off, illustrated below. The user enter their leg length when creating an account, and the applications sends this information to the Raspberry Pi, which then sends it to the *ATMega32u4* to use in calculations.

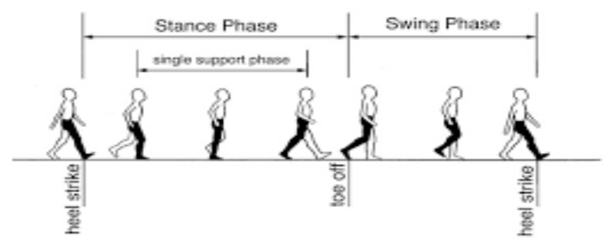

Figure 2: Gait pattern diagram. In our code, we consider a step starting at toe off and ending at heel strike. The Euler angle of the ankle at toe off is used in the formula (1) to measure stride length.

To test our system, we laid a long sheet of white paper on the floor, and taped a marker to each shoe, so that the marker only hit the paper when a step was complete. This way we could then measure the distance between dots and compare those distances to the stride lengths recorded on the application. We found our formula to work reasonably well, with maximum errors around 20%. To achieve better results, we recorded the stride lengths of 7 different people. We instructed the users to vary their stride lengths, from their normal walking pattern to the short shuffling steps that are characteristic of Parkinson's. We then used a regression to find the best formula that characterized the stride lengths, and determined the following:

$$S = L*4.0548e^{(-0.039\Theta)} \quad (3)$$

This equation yielded consistent and accurate results on all users and all gait patterns. Using the calculated stride length, we implemented vibration feedback on the ankle by comparing the stride length to an expected stride length. The expected stride length can be related to an average stride length based on one's height or the individual's average stride length based on their previous workout sessions, which will be stored in the application. The vibration will turn on when the percent difference between the expected length and the calculated length exceeds a certain threshold, which will be determined by the physical therapist; a stronger vibration will turn on if the percent difference between expected and calculated exceeds an even higher threshold.

To determine cadence, we simply record the number of steps taken and divide it by the time elapsed.

$$Cadence = steps/min \quad (4)$$

Since the workouts are time based, the time elapsed is already predetermined by the workout. Through auditory cueing, we aim to help the user achieve a certain cadence. Thus, when tracking progress, the user and therapist can compare their recorded cadence to the desired cadence.

C. Raspberry Pi

The Raspberry Pi is a tool that we use as our communications hub for our project. The device is responsible for four main tasks, receiving data from the on body sensors, performing the data analysis to provide real time feedback, relaying messages from the app to the devices, and transmitting it all to the AWS backend.

The first step in a workout is receiving the start message from the application. This is done using one of our two MQTT servers, this one in particular being our Cloud MQTT server. Once the device is turned on, the Raspberry Pi immediately subscribed to our specific URL and is waiting for a start message along with the patient specific information to be transmitted. Once this is received, the start signal for the workout along with the patient measurements are transmitted through Bluetooth to the sensor devices, which will described below.

The second step in the transmission part of the system is moving data from the on-body sensors to the Raspberry Pi. This is done using a Bluetooth Low Energy connection between each of the devices and the Pi. This is possible since the Raspberry Pi Model 3 B comes equipped with built in BLE capabilities [8]. The Pi3 B also allows for simultaneous communications without needing a separate adaptor, by attaching each specific Bluetooth connection to a RFCOMM serial port on the Pi. On the sensor side, the orientation IMU we chose did not come equipped with Bluetooth capabilities, so it needed to be connected to a transceiver. To transmit the data from our Atmega's we used an hc-05 Bluetooth transmitter module. This device enabled a serial TX/RX connection in both directions. Once the data has been transmitted, it can be processed by the system's analysis programs.

Once the data has been received through Bluetooth and analyzed by our formulas, the final task of the Raspberry Pi is to transmit this information to our AWS backend. The main goal of this step in our process is to provide secure and lossless delivery of our data to the correct destination.

In order to transmit the data to our backend, we needed to use the Raspberry Pi's wireless LAN component. After the device is connected to the local wireless network, in our case eduroam, there was still a need for a protocol to communicate with the server. MQTT is a publish/subscribe-based communication protocol that is used on top of the IP protocol [12]. The protocol allows the client to publish to specific 'Topics' that are located on a specific endpoint on the Internet. The AWS host provides the URL for the specified endpoint, and the topic that will be published/subscribe too will be specified by both the sender and receiver. Another aspect of this step in the transfer is the security of the transmission. When the MQTT client on the Raspberry Pi transmits it's message to the specified topic on the AWS endpoint, it also must provide it's security credentials. These were previously created for the device using AWS Iot. AWS IoT provides copies of the private and public keys as well as the CA certificate to the device. Without these the Raspberry Pi would be unable to publish to the specified endpoint.

Another aspect of the data transmission is that although MQTT protocol provides a way to connect to AWS Iot, the message being sent still needs to be in the proper format. AWS Iot uses JSON, or Javascript Object Notation, as the standard format for communications. JSON is ideal for tasks like these because it is can be used across multiple languages like C, Python, and Java [13]. The JSON format allows for different fields to be specified in the message, which are attributes such as user ID and specific data measurements. Once the message on the Raspberry Pi has been given the needed security credentials, put in JSON format, and addressed to the correct MQTT endpoint, it can be transmitted to AWS Iot.

Once the message from the MQTT client has been published properly, it still needs to be taken from that topic on the endpoint and moved to the DynamoDB database. The tool in AWS Iot called "Rules" allows for this task to be accomplished. A "Rule" can be created to tell the system to listen, or watch specific topics for incoming messages. The rule in the system will listen to our previously specified topic for transmitting data. When an incoming message is detected on the topic, the rule extracts the message part of the transmission, which is in JSON format. The JSON format is crucial in this step because the rule uses it's knowledge of the JSON format to navigate the message. The rule uses this ability to extract the userID fields and sessionID fields in the message for storage in the database. Using these two fields, the rule takes the information and stores it in the appropriate location in the database.

D. *Feedback Mechanism*

Real-time feedback will be used in our product to let patients know when their form is below threshold standards. This feedback will be given through vibrations on the ankle. All feedback analysis is implemented on the AVR processor, so there are no delays through Bluetooth connection. One of the major issues with PD patients is shuffling of the feet. If these movements of the patient are incorrect, a signal will be sent to the vibration module that will result in a minor vibration on the patient's ankle. This vibration on the ankle is only directed at the patients stride length. If the patient's stride length dips below a certain threshold, a vibration will be signaled. This will allow the patient to know when they have incorrect form and will allow them to modify this form before the next step is taken. This element of real-time feedback will stimulate neuro-training and will be used to provide the medical performance feedback usually given by the Physical Therapist.

The neuro-training in our feedback strategy was recommended to us by Professor Boyer in the Kinesiology department. Neuro-training is a muscle monitoring or muscle checking solution [5]. In other words, it involves training the brain to remember the right movements. This will be patient specific, as different patients have different gait patterns and different progress with our therapy. The vibrations involved in this feedback will have 3 settings, no vibration, slight vibration and full vibration respectively responding to form in the proper range, just out of the proper range, and very outside of the proper range. Even the highest vibration setting will be benign and safe for users to use.

Another aspect of neuro-training that will be integrated into our design is auditory cueing option. This technique was recommended to us by Debra Ellis, licensed therapist working in the Amherst area. Here, audio will be played in the form of sounds or rhythms from the app to assist to user by stimulating proper cadence [14]. Our vibrations will coincide with this to make it easier for the patients to identify and correct the problem with their form. All of this will be explained to the patient in the app prior to the workout.

E. *Software Application*

The software application is a tool for the patient to be able to track their long-term progress. In our product we currently have elements for real-time feedback to correct a user's form while performing an exercise, but in therapy it is important for a patient to be able to see their progress over a long period of time. The software application consists of two main aspects, an Android application and an Amazon Web Services (AWS) backend.

AWS allows for powerful backend operations. Services from data storage, to user authentication, to integration with IoT devices can all be incorporated into a single design. This power is what attracted us to the thought of having this as our backend.

One of the services employed by our design is Amazon Cognito. Amazon Cognito allows for user authentication using a login provider and a token granting service. As you can see in the image below, provided by reference [4], the initial login process involves three components outside of the user's device. The process works as follows. First the user enters their login information, which is verified by the Login Provider; in our case this is just an email and password. If correct, the Login Provider will return a ticket to present to Amazon Cognito. The device will then use this ticket to apply for a CredentialsForIdentity ticket. Before this is granted, Cognito has to verify information with the Login Provider. If the user is authenticated, Amazon Cognito returns the ticket. Using this ticket, the user applies to AWS STS. STS is a token granting service. These tokens allow for easy access to AWS services after the initial login. The ticket is then authenticated with Cognito, who authenticates it with the Login Provider, and if all check out the user is granted authentication tokens.

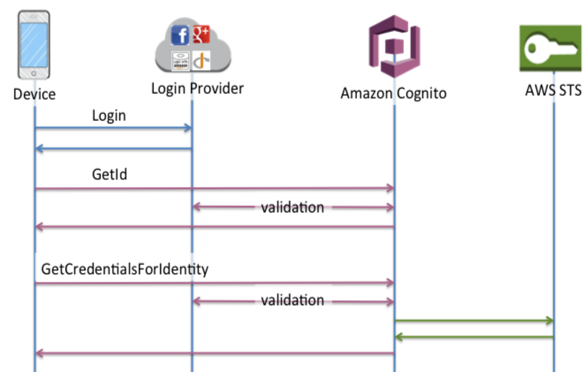

Figure 3: AWS authentication involves three different components. Once all three have authenticated the user, tokens are distributed allowing for access to other AWS services.

Now that the user has these tokens, they have easy access to all the services provided by amazon. One of those services which is employed in our application backend is a database, more specifically a DynamoDB Database service. DynamoDB is a NoSQL database that allows for quick storage and fast and efficient querying. This database is used to store a user's information for each session they participate in to track a patient's progress over time.

In this database there are several tables used to store different types of information. The first table is *User_Information* table used for login and user information. The second is *Therapist_Patient_List* which is used to keep track which therapist has which patient. The third is Patient_Workout_List which is used to keep track of workouts a patient has to do. These workouts are programmed by the therapist using the application. The fourth is Patient_Performance_Data which keeps track of performance data for individual sessions. The last is Error_Data which is where errors are written too. This database is checked after every session. The Raspberry Pi will store an error is one occurs during the session.

This information is extracted by the Android application and formatted into an easy to read display for the patient to track long-term progress. The application also allows for easy viewing for the therapist.

When the user logs in, they either login as a patient or as a therapist. If the user logs in as a therapist, the first things they have to do is select the patient they want to work with. This page also includes a button to add a patient. Here the therapist enters the username of the patient they wish to add and the database is parsed for that specific user. If that user exists and that user is a patient, they will be added to the list of patients for that therapist. From there they will have the option to edit their information, program a workout for the patient, or view the performance of the patient over the past sessions. If the therapist choses to program the program a workout they will have the ability to select the rate at which the user will walk in steps/min, the amount of time the patient has to perform the exercise, and lastly a goal stride length. This goal stride length can be two options, standardized or progressive. Standardized uses data from a general population to find what the appropriate stride length should be depending on the users leg length. The progressive option goes into the database and looks at the past couple sessions, takes and average of what the stride lengths were of those sessions, increases that average by 5% and sets that as the goal stride length. These two options will allow for an adjusted goal so the user can get more meaningful real-time feedback.

The user can also login as a patient. This is a slightly different process but with the same general idea. When the patient logs in they are presented with three options, edit their information, view programmed workouts, or view performance of past sessions. If the user selects view programmed workouts the app will bring up a screen containing a list of all the workouts the therapist has programmed for them. When the patient is ready they select the workout they want to perform and this brings them to a perform workout screen. They can click the start button when ready and the counter begins to count down from 5. During this count down information is sent to the Pi to indicate the type of workout that is about to be performed as well as a goal stride length for the user. This data is used by the sensor systems throughout the workout. Once this countdown finishes the patient is instructed to begin the exercise and continue until the timer runs out. Once the exercise is finished the patient is brought back to the main screen where they will have the ability to view their performance.

When the patient wishes to view their performance they are next instructed to select the metric they wish to look at, stride length, cadence, freezing, and heel-to-toe motion. Once one of these are selected they are brought to the screen similar to the one shown below. This screen shows a graph of the averages of that metric over the past couple sessions. This graph is meant to show long-term feedback for the patient. If the patient is struggling with getting a longer stride length, after a period of time the goal for that patient will be to see an increasing line. This page also gives a list of each session with the average for that metric and a goal for that metric. These specific sessions can be selected and the user has the ability to view a similar page showing measurements for individual sessions instead of the averages over past sessions. This patient-viewing page is the same page the therapist views when looking at the patient performance.

Figure 4: Visual representation of the Android application displaying the collected sensor data. This data is retrieved from the backend database.

III. Project Management

Richie, as team leader, has done more than expected to keep the group on track and advise on every major group decisions, even outside of his role. Along with this his main aspect of the project was the Android application and the systems backend. He worked with Android Studios to develop the application talked about above as well as set up the AWS backend including authentication and a NoSQL database.

Joe has done the majority of his work on tasks regarding the Raspberry Pi. After setting up the Raspberry Pi, he worked with Jarred on connecting and receiving data from the program Jarred created for the IMU sensors through a wired connection. He also worked with Richie to properly implement a data transmitting system from the Pi to the AWS database using an MQTT connection. Also he worked with Jack to select a Bluetooth transceiver to meet both the requirements of the PCB and the Raspberry Pi.

Jarred has been in charge of sensor programming and data analysis. He has developed a C program on the Raspberry Pi to control the IMUs and output data in m/s/s (acceleration) and degrees/s (angular velocity). He has tested the sensors using the "self-test" function of the sensors, as well as through simple movements and orientations of the sensor. He worked with Joe and Richie to move the data through the entire system, and also worked with Jack to find a Bluetooth receiver that could work with the sensors that we have chosen. Furthermore, Jarred has begun data analysis programming to convert values to performance metrics, but has not reached the testing phase yet.

Lastly, Jack has worked on the design to meet the needs of other group members roles and the project as a whole. As head of Mechanical Design, Jack worked with Jarred, head of data analysis in choosing a sensor that fits the needs of Stride. Together they chose the BNO-055. Jack also selected power supplies for the sensor systems and the Raspberry Pi. His main focus towards the end of our assignment was designing the PBC and mounting all elements on the body. Additionally, he assisted Jarred in data analysis algorithms and testing.

To summarize, our group has been meeting multiple times every week to inform each other what progress has been made. Recommendations and concerns of all team members are always addressed and meetings typically end on more than one group mate working together to complete the task(s) that have been discussed.

IV. Conclusion

The overall pipeline of our project is flawless. We were able to send data through Bluetooth to the Raspberry Pi and over our amazon web server to the android app. Data Analysis is implemented in every location. Values for our four main metrics, stride length, cadence, heel-to-toe, and freezing, were obtained by analysis on each AVR. Data is then collected and organized on the Raspberry Pi and structured and graphed on the app web server. Additionally, were able to send 'start workout' signals from the app to the sensor systems.

Our results were strong, with high correlation coefficients of the metric measured and the actual result. Our biggest hardship was the stride length algorithm. Limitations of the accuracy of our accelerometer restricted measurements we planned on using. For future progression, we have talked about a higher quality accelerometer and pressure sensors in the shoes for more accurate data.

Acknowledgment

Many thanks go to our advisor, Prof. Tessier, who has provided us with knowledgeable guidance and support.

REFERENCES

[1] Morris, Meg E. "Movement Disorders in People With Parkinson's Disease: A Model for Physical Therapy". *Physical Therapy*, vol. 80, pp.578-597, 2000. https://academic.oup.com/ptj/article/80/6/578/2842508

[2] Parkinson's Foundation. (2017). *Statistics*. [online] Available at: http://parkinson.org/Understanding-Parkinsons/Causes-and-Statistics/Statistics [Accessed 20 Dec. 2017].

[3] Tekscan. (2017). *Gait Analysis Systems*. [online] Available at: https://www.tekscan.com/gait-analysis-systems?utm_source=google&utm_medium=cpc&utm_term=gait+analysis&utm_content=eta2&utm_campaign=medical [Accessed 20 Dec. 2017].

[4] Xsens 3D motion tracking. (2017). *Gait analysis - Xsens 3D motion tracking*. [online] Available at: https://www.xsens.com/tags/gait-analysis/ [Accessed 20 Dec. 2017].

[5] Private conversation with Prof. Boyer of the Kinesiology Department of UMass Amherst. Date: 9-20-17.

[6] BNO055(2013). [Online]. Available:https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf [Accessed: 31-Jan-2018].

[7] HC-05 Product Specification v3.1. [Online]. Available: http://www.electronicaestudio.com/docs/istd016A.pdf [Accessed: 31-Jan-2018].

[8] S. T. Moore, H. G. MacDougall, J.-M. Gracies, H. S. Cohen, and W. G. Ondo, "Long-term monitoring of gait in Parkinson's disease," *Gait and Posture*, vol. 26, no. 2, pp. 200–207, Oct. 2006.

[9] Raspberry Pi 3 Model B. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ [Accessed: 31-Jan-2018].

[10] Suzuki, M., Mitoma, H. and Yoneyama, M. (2017). *Quantitative Analysis of Motor Status in Parkinson's Disease Using Wearable Devices: From Methodological Considerations to Problems in Clinical Applications*. [online] NCBI. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5451764/ [Accessed 20 Dec. 2017].

[11] Caron, F., Duflos, E., Pomorski, D. and Vanheeghe, P. (2017). *GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects*. [online] Available at: http://www.sciencedirect.com/science/article/pii/S156625350400065X [Accessed 20 Dec. 2017].

[12] Using Accelerometers to Estimate Position and Velocity. [Online]. Available: http://www.chrobotics.com/library/accel-position-velocity. [Accessed: 31-Jan-2018].

[13] HiveMQ. (2017). *Mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe* [online] Available: https://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe [Accessed December 20, 2017]

[14] JSON (2017). *Introducing JSON* [online]. Available at: https://www.json.org/ [Accessed December 20, 2017]

[15] Private conversation with Physical Therapist Debra Ellis. Date: December 2017. URL to her website: http://debellisphysicaltherapy.com/

[16] Docs.aws.amazon.com. (2017). *Authentication Flow - Amazon Cognito*. [online] Available at: http://docs.aws.amazon.com/cognito/latest/developerguide/authentication-flow.html [Accessed 20 Dec. 2017].