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 which 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.

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
Provide real-time feedback	Feedback given in less than 100ms
Provide accurate metric measurements	Within 10% error of Qualisys Oqus Motion Capture System measurements (in UMass Human Motion Lab)
Lightweight product	Sensor systems (sensors, PCB, and power supply) less than 1 pound Waist Clip (Raspberry Pi and power supply) less than 1 pound
Sufficient battery life	Battery life greater than 2 hours
User friendly mechanical design	User should be able to put on product easily with little to no added effort

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 needs to be given before the patient takes their next step. Along with this, for the feedback to be accurate the measurements taken 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 of the Qualisys Oqus Motion Capture System measurements, which is located in the UMass Human Motion Lab, is a sufficient range [5].

This product is going to be used on patients that have trouble moving routinely because of PD. This means the design needs to be easy to wear and be lightweight. Each sensor system

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).

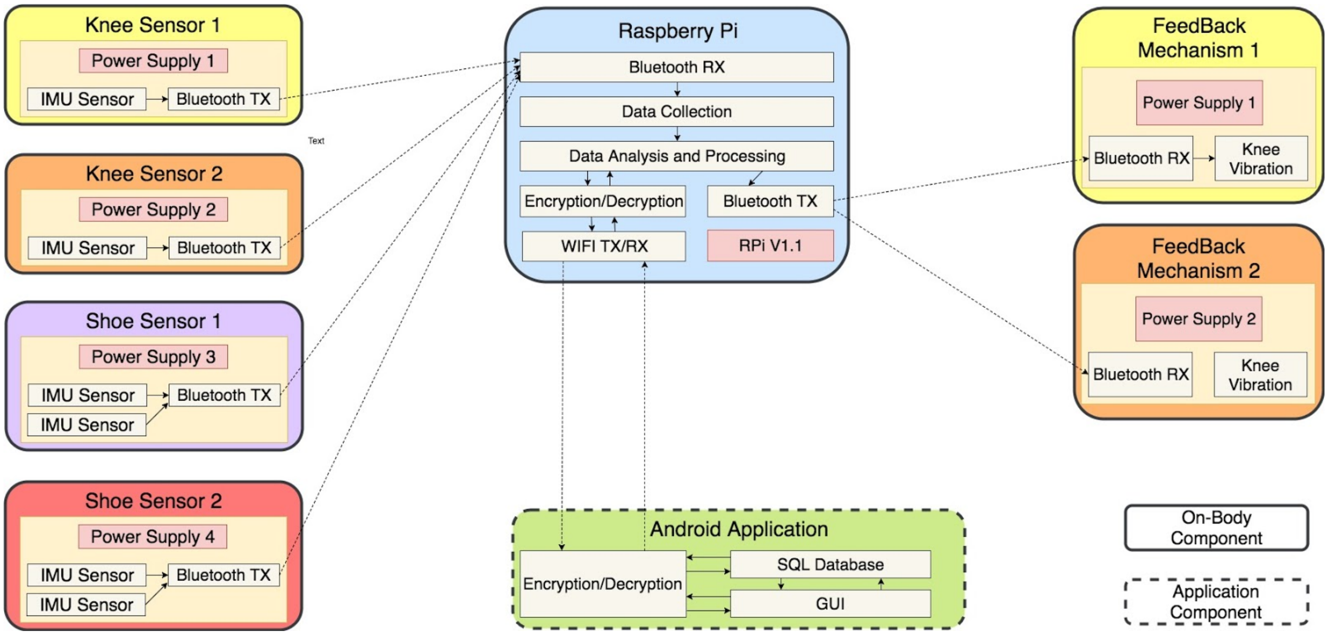

Figure 1: Block diagram for Stride. All the information from the sensors is relayed to the Raspberry Pi for analysis. Real-time feedback is provided by vibration components and long-term progress tracking is provided by the Android application.

should weight less than one pound. This sensor system includes, the sensor, PCB, and the power supply. Along with this, the waist clip, which holds the Raspberry Pi and the power supply, needs to be less than one pound. By limiting the weight of the product, the patient will not notice the added technology when performing the exercises.

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.

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 knee and for each foot. These systems will collect data and transmit it 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. Along with this, after the data is processed on the Raspberry Pi, real-time feedback is triggered using a signal sent over bluetooth to the feedback systems. 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 IMU Sensor blocks, a bluetooth transmitter module, and a power supply. Each shoe has two *BNO055* [6] inertial measurement units (IMUs), one mounted on the heel, and one mounted on the toe. The IMUs measure linear acceleration and angular velocity in the x, y, and z axes. This data is relayed to the *hc-05* bluetooth module, reference [7], and then transmitted to the Raspberry Pi for data processing and analysis.

Each IMU is programmed in C to set the registers in the sensors to their appropriate settings. For example, the IMU also contains a magnetometer, but this will be disabled for this project. Data rate, sensitivity levels, and input and output registers are additional examples of what can be modified when programming the IMU itself, as opposed to manipulating the data that the IMU outputs, which is done via C program on a Raspberry Pi 3 Model B [8]. Techniques used from programming and hardware courses, specifically Computer System Lab I and Hardware Organization and Design are used in building this block, as we must read from and write to 8 bit control and output registers to retrieve data.

To complete the building of the two shoe sensor blocks, we must learn how to receive and transmit the data from the IMUs to the Raspberry Pi via bluetooth. The bluetooth transceiver supports I2C protocol, thus transmitting the data via wires from the sensor to the correct bluetooth chip I/O pins should be relatively simple. The Raspberry Pi is equipped with bluetooth 4.0 so it will be able to communicate with the *hc-05* bluetooth module.

This block can be tested incrementally and then as full block integrated into the system. First, the sensor is wired directly to the Raspberry Pi GPIO pins using I2C protocol. To ensure a proper connection, we can run I2C commands from the Raspberry Pi terminals to see which output registers are associated with the accelerometer and gyroscope, and confirm this with the IMU datasheet. Next, the C code is executed from the Raspberry Pi terminal, and the outputs appear on the computer screen with acceleration and gyroscope data. We can then move the sensor around in different directions and see outputs change. The simplest test to make sure the values are correct is to keep the sensor flat on a desk and facing upwards and ensuring the acceleration in the z-axis is around 9.8 m/s^2, which is the acceleration due to gravity. Furthermore, the IMU has a self-test functionality, activated by setting the 'self-test' bit to logical 1, which applies an actuation force that simulates a definite acceleration and Coriolis force and will give a predefined output that signifies proper functionality.

Ultimately, the shoe mounted IMU's gather data on acceleration and angular velocity, and transmit this data to the Raspberry Pi, which handles the data analysis. In particular, the shoe IMU's will be responsible for determining moments of heel strike, toe strike, heel off, and toe off. This data will correspond to our performance metrics of heel-to-toe weight distribution and freezing, which will be available for the patient to track on the phone application.

B. Knee Sensor System

The two knee sensors are designed in the same way and have the same functionality. Each block contains one IMU, a power supply, and bluetooth transmitter. The same IMU and bluetooth module will be used on the knee as were used on the shoe, however a smaller power supply can be used since it only powers one sensor and transmitter. The sensor programming and testing for the knee mounted IMUs is the same as it is for the shoe mounted IMUs, however, the data analysis, which is described in the Raspberry Pi section, is much different. The knee mounted IMUs will be responsible for determining vertical displacement of the leg, which will correspond to stride length and 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 three main tasks, receiving data from the on body sensors, performing the data analysis to provide real time feedback, and transmitting it all to the AWS backend.

The first 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. On the sensor side, the IMU we chose did not come equipped with bluetooth capabilities, so it needed to be connected to a transceiver. For MDR, our goal was to use the *Adafruit Feather 32u4* to accomplish this task, but due to complications we were not able to accomplish this task on time. Moving forward, the sensor device will be paired with an *hc-05* bluetooth module and an *ATMega32u4* to transmit the data over the connection. Once the data has been transmitted, it can be processed by the system's analysis programs.

Once the Raspberry Pi receives the sensor data via bluetooth, it the next step is to analyze the data and convert it to the four performance metrics that the user and physical therapist will track over time. These metrics include, stride length, cadence, heel-to-toe weight transfer, and freezing. The data comes to the Raspberry Pi in units of mg/LSB (milli-G's per least significant bit) and mgauss/LSB, which we first convert to m/s^2 and Gauss so that we have meaningful values to work with.

Data analysis presents several challenges that we are working to overcome. The first obstacle is learning to properly handle "drift" of the gyroscope data and noise from the accelerometer. The gyroscope data from the IMU will output values that drift over time, this is an unavoidable property of gyroscopes, and thus steps must be taken in data analysis to correct for this drift. Since the entire system is designed to last for two hours, which is the length of a typical physical therapy session, the effects of drift will be significant, as gyroscopes typically drift around 0.114 deg/min. One way to solve the issue of drift is to reset the gyroscope to zero periodically [9]. By doing so, the effects of drift will be negligible as long as the gyroscope is reset after only short periods of time. The gyroscope can only be reset when it is stationary, thus the resetting would need to take place during a time when the user is standing still. Another way to compensate for drift is to fuse measurements from the gyroscope and accelerometer using a filter that favors values from the gyroscope for short periods of time and favoring the accelerometer values for longer periods of time since the accelerometer does not have drift. A Kalman Filter will be implemented to accomplish this, as it uses a series of previous measurements to produce an optimal estimate of the correct value for gyroscope data, and is used widely to estimate uncertain information in dynamic systems [10]. Ultimately a combination of resetting and filtering will be used for handling drift. To test our method of eliminating drift, we simply keep the program running for around two hours, and while also leaving the IMU in a constant orientation (i.e. flat on a table) and away from any magnetic interference. If the gyroscope outputs stay relatively constant, then we know that drift has been eliminated.

The ultimate goal of data analysis is to convert raw sensor values to stride length, cadence, heel-to-toe weight transfer, and freezing. These are the metrics that our system aims to improve in Parkinson's patients, stores long term to track progress, and, in the case of stride length, uses to signal our real-time vibration feedback. On the Raspberry Pi, we will process the data from the knee and shoe IMU's and output performance metrics using C programming. Using the data from the knee sensors, displacement can be measured by taking the double integral of acceleration and the integral of angular velocity. However, integrals introduce drift, which are low frequency deviations of values from the correct value. A small miscalculation of IMU orientation angle can result in large displacement errors over time. For example, an angle error of 0.5 degrees corresponds to about an 8.6 meter error in displacement after 10 seconds [9]. Thus, it is important to accurately calculate the orientation of the IMU using AHRS methods, and while also applying a high pass filter to eliminate drift [9] Since the sensor is on the knee, the displacement is in the vertical direction, so we then use the inverted pendulum model of walking, which translates vertical displacement into horizontal displacement (step length) [11]. Average cadence can be calculated easily by dividing the number of steps taken in a session by the time it took to complete the session. The shoe mounted IMU's will be used to find heel to toe walking motion and freezing. Heel-to-toe motion will be determined by recording moments of heel-strike, toe-strike, heel-off, and toe-off on each foot. If these moments occur in each foot in that sequence, then we know that the user is walking with the correct form. From there, the amount of force used in each aspect of the heel-to-toe motion will need to be determined using test cases.

The data analysis on the Raspberry Pi can first be tested by mounting the IMU's on to members of the team and measuring the values against manual measurements of stride length, cadence, heel-to-toe, and freezing that we emulate. Next, we will test the system on a user with Parkinson's Disease (we are speaking with physical therapists on this matter) and verify results again, while also recording data to be used for feedback. Feedback will respond to incorrect stride lengths, since those with Parkinson's suffer from a shortened stride length. After testing, we can determine what values of stride length trigger vibration feedback. Finally, we will test our whole system in the UMass Human Motion Lab, which has a gold standard motion capture system.

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 knee. The data collected by the Raspberry Pi is processed and analyzed. 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 which will result in a minor vibration on the patient's knee. This vibration on the knee 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 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 [15], 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.

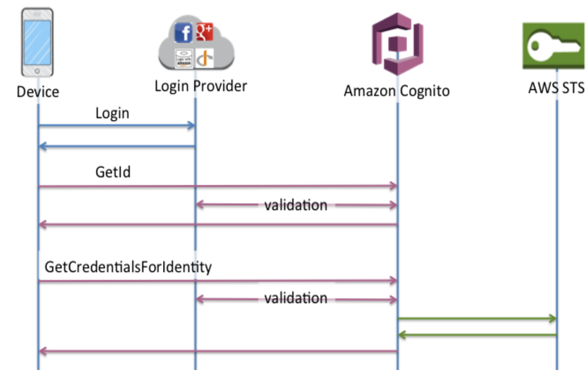

Figure 2: AWS authentication involves three different components. Once all three have authenticated the user, tokens are distributed allowing for access to other AWS services.

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 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 will be several tables to store several types of data. One table entry consists of a username, a session number, a measurement number, and all the data collected for that measurement by the sensors. This data includes the accelerometer reading, in the x, y, and z direction, and the gyroscope reading, in the x, y, and z direction. Along with this there will be separate tables to store information about an individual's stride length, cadence, freezing points, and heel-to-toe movements.

This information will be extracted by the Android application and formatted into an easy to read display for the patient to track long term progress and for a therapist to analyze their patients performance. As of right now, the application consists of three pages. The first, after login, allows a user to select which session they would like to view data from. From there, the application brings the user to a page displaying three possible options for sensor data to view, accelerometer, magnetometer, and gyroscope. When either of these are selected, the screen, shown below, displays all the measurements for that session and the values in the x, y, and z direction for that sensor.

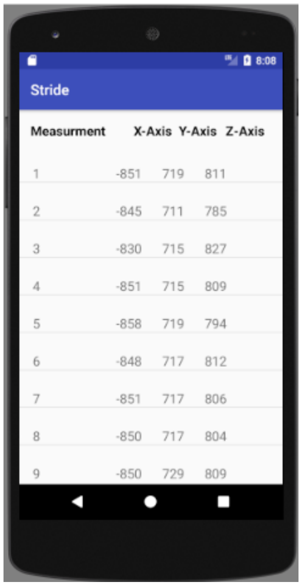

Figure 3: Visual representation of the Android application displaying the collected sensor data. This data is retrieved from the backend database.

III. Project Management

Deliverable	Status
Single IMU operating and recording data.	Yes: collects data using the accelerometer, gyroscope, and magnetometer
Bluetooth transmitter on a breadboard relaying the IMU data to Raspberry Pi.	No: breadboard design was altered and a different microprocessing transceiver was selected. New design implementation could not be met by MDR.
Routing the data through the Raspberry Pi and transmitting it over WIFI to AWS backend to be stored in a NoSQL database.	Yes: data from sensor is temporarily stored in text file which is then parsed and sent to database through Wifi
Android application retrieves data from database and displays it on a basic GUI.	Yes: application retrieves information from database and displays values in organized fashion

Table 2: MDR deliverables and statuses of each deliverable.

The above table, Table 2, shows the status of each of our MDR deliverables. In our presentation, we had single IMU operating and recording data, no bluetooth transmitter on a breadboard relaying the IMU data to Raspberry Pi, routing the data through the Raspberry Pi and transmitting it over WIFI to AWS backend to be stored in a NoSQL database, Android application retrieves data from database and displays it on a basic GUI. The table shows 3 out of 4 deliverables met. For the presentation to show this we had to hard wire our sensor directly to the Raspberry Pi.

Halfway done with SDP, Team Stride is a motivated, and hard working group. Members, in addition to there specific individual roles have worked with each other to conquer the challenging tasks.

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 an 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. After ordering our first microcontroller transceiver, we quickly learned that not all transceivers would pair with the Raspberry Pi. Working with Jared, Jack later found the Feather 32u4 Bluefruit LE that met the needs of easy wiring to the sensor on the breadboard and also easy for Joe to connect to the Raspberry Pi. He also adjusted the shoe design to replace the previous insole sensor which our group has determined to be unfeasible for this project due to expense and failure to get information from manufacturing companies. With Jared, they discussed what data needs to be collected to be able to still analyze the gait characteristics of a patient with parkinson's disease and formed our current shoe design.

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 groupmate working together to complete the task(s) that have been discussed.

Below is a list of our CDR Deliverables and Gantt Chart of what we plan on accomplishing. Notice how our roles listed

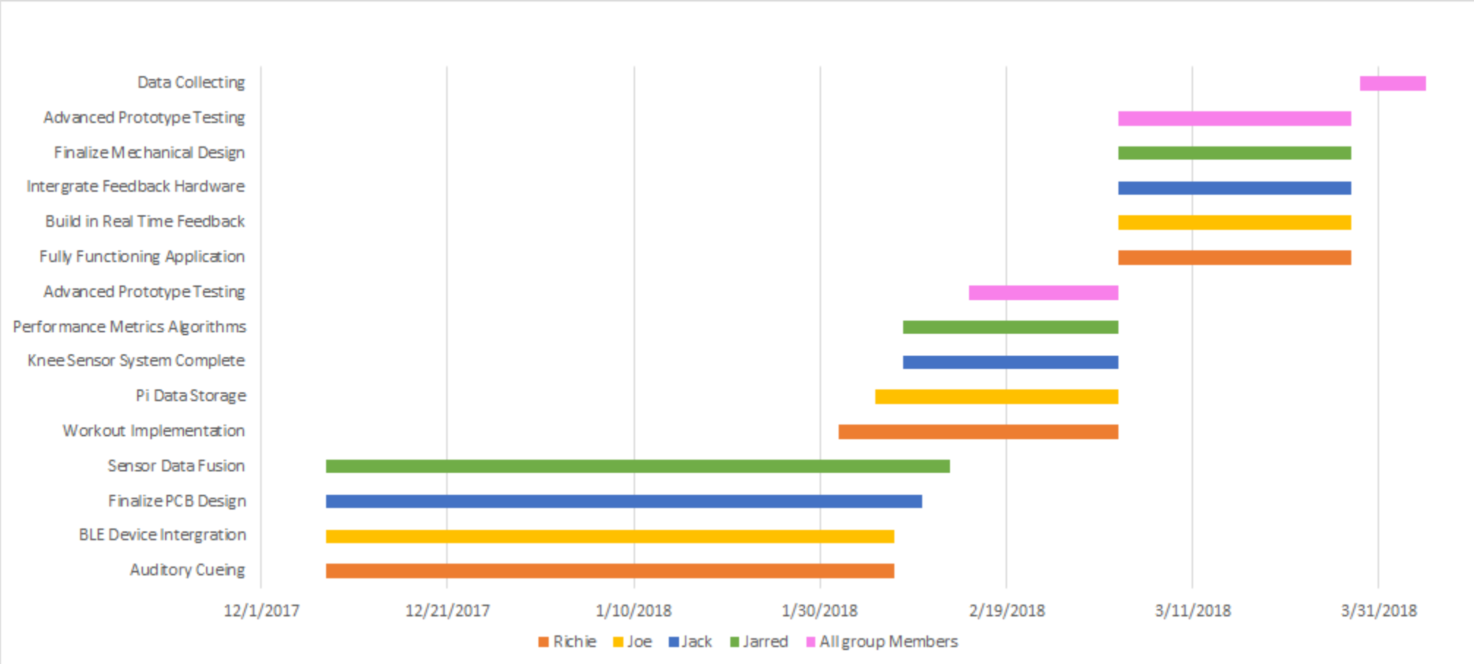

Figure 4: Gantt chart displaying the upcoming tasks along with deadline for each of the tasks. The last week before FDR will be designated for data collection as well as product polishing.

above directly correspond to our goals on how to move forward.

Proposed CDR Deliverables:

- Complete design of knee sleeve with functional IMU and PCB (no feedback)
- Functioning prototype of shoe system (w/o PCB implemented on shoe)
- Simultaneous bluetooth connection between both shoe and knee IMU systems and Raspberry Pi
- Completed data analysis of all useful data from knee sensor systems
- Shoe data passed through the pipeline of system
- Implement auditory cueing on Android Application and exercise programming

IV. Conclusion

The overall pipeline of our project is almost complete. The main idea of our MDR deliverables was to have data flowing from every aspect of our project. We successfully sent data through every connection of our design with the exception of the bluetooth connection of the breadboard to the Raspberry Pi. The change in the microprocessing bluetooth transceiver held a slight delay in our project schedule.

We plan on finishing the last deliverable right after final exams to start on the tasks for our next deadline. We will now attempt to accommodate the functions of the Feather 32u4 Bluefruit LE and meet our PCB requirement. We will use an Atmega32u4 processor, which is the same processor in the Feather already, and a hc-05 bluetooth module for connection to the Raspberry Pi. Both components will be mounted on PCB. We hope by using these components we can have an easy transition of breadboard-to-PCB regarding the bluetooth pairing and the feedback signaling.

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

[9] 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].

[10] 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].

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

[12] 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]

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

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

[15] 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].