Deliverables

**Introductions/overview** –

The Knee Habilitation Conversion projects goal is to turn the proof-of-concept algorithm into a more viable product, which is simple and accessible enough to be used by patients and clinicians. This will be achieved through the conversion of the pre-written Matlab code into C code, and the development of an Android based application for data accumulation and post processing. The software, which will be partially embedded and partially running on the smart phone application, will give the patient or clinician near real time results for knee-joint angle and gait analysis. The C programming language was chosen because of its time/space efficiency and high portability. The program could easily be ported to PC or another computer application without the need for extensive overhauling. Additionally, the Android platform was chosen because it has the largest market share of any mobile operating system. Leveraging this platform will allow the software to be accesses by a large amount of patients and clinicians.

**System Architecture** –

Once the smartphone application is launched for the first time, the patient will be prompted to fill out a profile that will be stored in memory. By saving the profile, the application will accumulate the patients exercise data over the course of his/her recovery. The patient will wear four wireless inertial measurement units (WIMUs), two on the left/right shank and two on the left/right thigh, using the onscreen guide as a reference. Each of these units will communicate with the smartphone via low power Bluetooth.

The patient will turn on the WIMUs and pair them to the smartphone. Once he/she has the units set up and connected to the smartphone, they will begin a new exercise session in the application. The application will ask the subject whether to compute gait analysis. At this point, the smartphone will send a start signal to the WIMUs. Packets of raw data will be sent to the smartphone at “random” time intervals. This can be done in two ways: relative to the smartphone internal clock (absolute timestamp) or at preprogrammed intervals from the WIMUs (We hope to be able to use the smartphone in order to randomize the time intervals). This raw data is used for orientation alignment of the shank WIMUs during post processing.

The units will begin calculating orientation angles and temporal gait parameters based on the sensor data. These calculations will be done in embedded C and the results will be transmitted to the smartphone via Bluetooth. These calculations will be stored on the WIMUs SD cards until they will be transferred to the smartphone. Once the exercise is complete, the patient will press the finish button on the application and the smartphone will send a stop signal to the WIMUs. A final packet of raw data will be sent immediately after the stop signal is received.

Once completed, the orientation and temporal gait calculations will be sent from the WIMUs to the smartphone. While most of the calculations are done in embedded C on the WIMUs, thosethat require information from more than one WIMU will be completed on the smartphone in the post processing phase. Included in this phase are the interpolation, resampling, and alignment functions.After this phase, the patient will be able to view their knee-joint angle and gait analysis results. They will be prompted to continue performing exercises, or end the session and save the results to memory.

An overview of the smartphone to WIMUs data flow is summarized in the diagram below. Note that the numerical ordering on the diagram corresponds to the order in which the program runs.

C:\Users\luke.jervis\Downloads\System Architecture (1).png

**Description of Application Functions** –

The most important aspect of the application is time/space efficiency. In order to deliver fast and accurate results, each function of the original Matlab code has been analyzed to find computational bottlenecks and areas of code that can be completed more efficiently in C. The processing workflow of the application will change with respect to the Matlab code due to the fact that only certain functions can be performed in embedded C. The flow charts below highlight these changes. Note that the functions running in embedded C are highlighted in yellow on the diagrams.Also, the “random” sample question in the C workflow is subject to change given our decision on whether to use the smartphone or WIMUs internal clocks. If we use the smartphone then true randomness can be achieved, but there will be a greater battery draw due to more Bluetooth signal transfers. Using the WIMUs clocks would require pre-determined time intervals but less battery draw.

C Workflow

Matlab Workflow

C:\Users\luke.jervis\Downloads\Untitled Diagram (4).pngC:\Users\luke.jervis\Downloads\C workflow (1).png

The WIMUs will collect raw data and preform orientation and temporal gait calculations in near real time using embedded C on the microcontrollers. These calculations will be sent to the smartphone application via low power Bluetooth at the end of the exercise. There will also be packets of raw data send at “random” time intervals from each of the WIMUs. These packets will be used in the post processing alignment functions.

The following presents a high level description of each Matlab functions purpose, whether it will be done embedded or on the smartphone, and comments on C translation and optimizations:

**Module Name:**

**data\_import\_export\_module (Embedded C)**

Matlab functions used: JCS\_conversion

This module imports the raw data from the sensors into an internal data structure that can be used for manipulation by the rest of the program**.** It also sends raw data to the smartphone at time intervals. At the end of the exercise the orientation and gait data stored on the SD card will be sent by this module to the smartphone application via Bluetooth.

This module is continuously running and monitors for the start and stop signal from the smartphone application in order to start/stop importing the raw data.

In order to conform to the International Society of Biomechanics (ISB) the data will be added to the internal data storage according to the JCS reference frame. This will allow for correct joint-knee angle analysis.

The implemented data structure resembles a vector-like array of double pointers that simulate rows in a matrix. These rows each hold pointer arrays, simulating columns. The data is stored through these pointers with double precision (64 bit). This data structure is dynamic; meaning rows and sections can be deleted and added without having to copy the contents of the entire structure, as Matlab does. This makes it much more efficient when preforming the numerous matrix manipulations that this program requires.

**Module Name:**

**Quaternions (Embedded C)**

Matlab functions used: GetOrientationQuaternions/GradientDescent/MyFilter/ComputeEulerAngles

This module will perform the quaternion fusion algorithm required to compute the knee-joint angles. This algorithm will combine quaternion based calculations from both the gyroscope and accelerometer in each WIMU to find its orientation. In the case of the accelerometer, a gradient descent algorithm is used to compute the direction of the gyroscope measurement error as a quaternion derivative.

**Module Name:**

**temporal\_instants\_module** **(Embedded C)**

Matlab functions used: temporal\_instants1

This module calculates the three most important temporal parameters for the gait analysis: heel strike, mid-stance (vertical shank), toe-off.

The current Matlab implementation requires a large amount of samples. This is suitable only for post processing. In order to complete this function in embedded C, it is necessary to revise the algorithm almost completely.

**Module Name:**

**interpolation\_module (Smartphone)**

Matlab functions used: Preprocessing2/lininterp1:

This module will perform a linear interpolation on the transmitted orientation data matrices. It will also run the data through various filtering techniques (one-dimensional median filter, zero phase digital filter) designed to reduce white noise in the sensor data.

Preforming linear interpolations is a necessary procedure because the WIMU’s sensors can record unreliable data samples due to inherent flaws and read/writing errors. The second linear interpolation is to add precise data points between rows in which the time stamps differ by more than a certain threshold.

One limitation when translating into C is that these filtering functions are pre written in Matlab and not in the C standard library. I will work with Marco S. to develop novel filtering techniques, or source outside libraries that will perform with the same or similar levels of accuracy as the Matlab functions.

**Module Name:**

**resampling\_module (Smartphone)**

Matlab functions used: Resampling1/resampling2

This module is designed to temporally align the WIMU’s in order to fix the sample number discrepancies that can occur, due to the sensors internal clock drift. This is done by comparing the relative time stamps of the WIMUs. A newly aligned matrix is returned for the orientation data from the WIMU’s.

In the first resampling function, the data from the thigh and shank of the same legs are aligned. The shorter of the two WIMU’s data matrices is used for the realigned matrices row dimension. The rows with corresponding relative time stamps closest in time are chosen to fill the new matrices.

In the second resampling function, a similar procedure is completed in order to temporally align the thigh/shank sensors on both legs to one another

**Module Name: horizontal\_alignment\_module (smartphone)**

Matlab functions used: horizontal\_alignment2/horizontal\_rotation

The WIMU’s on the shank and thigh, each have a local reference frame and orientation. In order to compute the knee-joint angle, a common reference frame is required. We assume that the horizontal and vertical axis on the thigh WIMU’s are in the correct common reference frame. Further, the sensors on the shanks have the correct vertical alignment. To achieve the correct common alignment, it is only necessary to rotate the horizontal (XY) plane on the skank WIMU’s. This is done using the X axis on the thigh WIMU as a reference point. This module will perform a horizontal alignment according to the Seel method in the literature.

In order to calculate the angle of rotation, samples of time stamp/gyroscopic data are needed from the thigh and shank WIMUs. It is important that these samples correspond in time with one another. To achieve this, the smartphone will send a signal to the WIMUs at “random” time intervals in order to receive raw data packets.

At the present moment, the C implementation of this function will conform to the Matlab implementation closely. Unfortunately, this function and its child function, horizontal\_rotation, presents the largest bottleneck in the code thus far. This is due to the fact that there are both constant-time nested loops (50,000 iterations) and sample size based loops. I plan on working with the Andrea and Salvatore to figure out a solution or compromise that will allow this module to run more quickly.

**Module Name: knee\_angle\_computation\_module (Smartphone)**

Matlab function used: KNEEHAB1

This module will calculate the final knee-joint angles: flexion-extension, varuus-valgus, and internal-external.

**Module Name: temporal\_intervals\_module(Smartphone)**

Matlab functions used: singleleg\_temporal\_intervals/combined\_temporal\_intervals

This module will calculate the temporal intervals: Gait Cycle Time (GCT), Swing phase (SW), Stance phase (ST), and Double-Support (DS).

**Module Name: stride\_length\_module** **(Smartphone)**

Matlab functions used: compute\_stridelength\_2

This module will compute the stride length (SL) and clearance. The angles of the leg and initial velocity are obtained using gyroscope data. Using the angles, the local acceleration is converted into the global frame. From the global frame, the stride length is computed by integrating the acceleration data twice and accounting for the initial velocity.

We are working on how to change some of the stride length algorithm in order to compute sections of the module in embedded C on the smartphone.

**Description of GUI:**

As was stated above, the GUI will be implemented using the Android mobile development platform. The goal of the GUI is to provide the patient/clinician with an easy to use application to view and save personal details and analysis data. The general user experience is outlined below:

* The user will see a welcome screen the first time he/she launches the application
* They will be prompted to complete a user profile that will include the personal details section of the KNEEHAB1 Matlab GUI
* This profile and the data associated with it will be saved on the application so that the user can complete repeated experiments without having to re-enter their personal data
* After completion, the user will be taken to the home screen of the application
* The home screen will have the following buttons:

**Start new session**: this will begin a new session of exercises

**View session data**: this will take the user to a new screen to view the saved analysis data from various sessions/exercises

**Change/Add Profile**: this will take the user to a new screen listing all of the saved profiles on the device, and allow then to alter or add profiles

**Update Profile**: this will take the user to a screen in which they can update their personal details

A more detailed explanation of the home buttons is outlined below:

**Start New Session:**

This will take the user to a set up screen, guiding them through the WIMU sensor set up and placement, Bluetooth connection, and exercise movements. The sensor set up will include a screen prompting the user to enter the test details and IMU’s sections of the KNEEHAB1 Matlab GUI. This will be used in preforming the post processing analysis for the specific session. It will then tell the user when to turn the sensors on/off via the microcontroller and allow them to view analysis data after the completion of one or more exercises. Once they review the analysis data for their exercise(s), they will be asked whether to save the session to the View Session Data section of the application. At this point they will be taken back to the home screen.

**View Session Data:**

This button will take the user to a screen containing a list of previously saved sessions for their profile. Upon selecting one of the sessions, the user will be brought to a screen containing the data for this session, which will resemble the excel files created in the Matlab code. They will also have the option of deleting the various sessions.

**Change/Add Profile:**

This button will take the user to a screen containing all of the saved profiles on the application. They are able to select another profile and the corresponding data will be available. From this screen they will also be able to set up new profiles following the same set up processe that was outlined above.

**Update Profile:**

This button will take the user to the same initial set up screen as described above. The text fields will contain the previously saved data, but will be available for modification. Once the user is finished updating their profile they can save the modifications and will be taken back to the home screen.

**Conclusion:**

The ultimate goal of the project is to develop a program that can deliver accurate results to the patient or clinician in an accessible, easy to use application. The use of real time computations on the WIMUs sensor data will make the program time efficient. Further, given that processing power on Android capable devices can vary, preforming part of the computations in embedded C will put less strain on devices and deliver comparable experiences to all users.