

December | 2011

Step Rate Monitor to be used during Gait Analysis

Carmen Coddington, Bryan Jepson, Joel Schmocker, Christa Wille

Client: Bryan Heiderscheit, PT, PhD, Orthopedics and Rehabilitation

Advisor: Mitchell Tyler, Biomedical Engineering, University of Wisconsin – Madison

December 14, 2011

Abstract

With an increase in the popularity of running, an increase in the occurrence of running related injuries has become evident. Although many risk factors have been identified, excessive knee joint loading has been recognized as one of the most common when predicting the occurrence of injury. A common outcome for altering joint loads during running is with an increased step rate (number of steps per minute). By achieving a reduction in joint loading, an injured runner may be enabled to continue running without aggravating symptoms, while receiving care for their injuries. Similarly, utilizing an increased step rate may prove beneficial following injury recovery as part of a progressive return to running. Thus, it is important to monitor step rate during a running analysis. We have created a design to monitor the vibrations that occur throughout the treadmill as a result of each step taken by the runner. A uniaxial accelerometer is used to detect small vibrations in the infrastructure of the treadmill. This signal is feed back to the computer where it is processed to identify step rate in real-time. The runner's step rate will be updated and displayed to the runner and clinician in real-time. The step rate monitor will eliminate the need for the clinician to manually count step rate, allowing them to focus more of their time with the runner. Furthermore, by providing the runner with useful visual feedback, the process of learning how to increase or decrease step rate will be simplified.

Table of Contents

Background.....	4
Motivation	4
Client Information.....	5
Problem Statement	6
Competition	6
Design Specifications	7
Sensor Alternatives.....	8
Accelerometer.....	8
Optical Sensor.....	8
Acoustic Sensor	8
Data Processing Alternatives.....	9
Java.....	9
Matlab.....	9
LabVlew	9
Design Matrices	9
Preliminary Data.....	11
Final designs.....	12
Data Processing.....	13
Filter	Error! Bookmark not defined.
Calibration.....	Error! Bookmark not defined.
Monitoring Step Rate.....	Error! Bookmark not defined.
Testing and Results.....	14
Placement of the Accelerometer.....	14
Altering Speeds	15
Orientation	16
Future Work.....	16
Real-Time Data Processing.....	16
Improve Signal Filtering.....	Error! Bookmark not defined.
Runner- Device Interface.....	16
Further Testing.....	19
Objectives	19
Appendix	20
A. Project Design Specifications	20
B. MatLab Code for Data Processing	23

BACKGROUND

MOTIVATION

With an increase in the popularity of running, an increase in the occurrence of running related injuries has become evident¹³. It is expected that approximately 56% of recreational runners will sustain a running-related injury each year¹⁴, with 42% of all injuries occurring at the knee¹². Although many risk factors have been identified, excessive knee joint loading has been recognized as one of the most common when predicting the occurrence of injury⁹.

In the interest of reducing loads to the lower extremity joints during the loading response (LR) of running, several popular strategies have been proposed including minimalist footwear and alterations in running form^{3,5,11}. A common outcome from these different strategies is an increased step rate (number of steps per minute). Heiderscheit et al., characterized the influence of step rate modification on lower extremity biomechanics during running. Kinematic changes that were observed as a result of an increase in step rate include a decrease in all of the following variables: step length, center of mass (COM) vertical excursion, horizontal distance from the COM and heel at initial contact (IC), foot inclination angle at IC, knee flexion angle at IC, peak knee flexion and step duration⁸ (Figure 1). Therefore running with an increased step rate will require a decrease in step length, thus decreasing the distance to the heel with respect to the COM at IC. As a result the foot inclination angle will also decrease, shifting the foot strike pattern from a heel strike to more of a mid-foot strike. In addition COM vertical excursion will also decrease, reducing the velocity at which the runner strikes the ground⁸.

Changes in kinematic variables may also be used to explain kinetic changes that occur with an increase in step rate. For example, with a decrease in COM vertical excursion the runner will strike the ground at a decreased vertical velocity. Therefore, a decrease in the peak vertical ground reaction force and the braking impulse is observed⁸ (Figure 2). A decrease in braking impulse is advantageous during running as the runner can devote a larger portion of energy expenditure towards the propulsive impulse instead. Furthermore, an increase in step rate is associated with a reduction in the mechanical energy absorbed during loading response (LR) in all lower extremity joints with the most significant

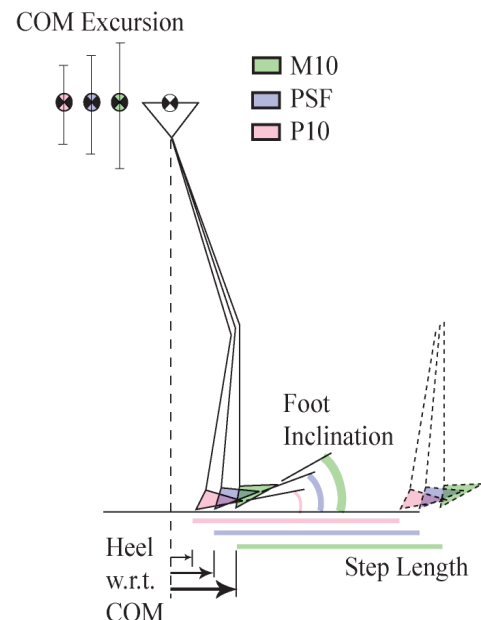


Figure 1. Kinematic changes that occur due to a modification of step rate, a comparison between preferred stride frequency (PSF) and 10% above (P10) and 10% below (M10) PFS. With an increase in step rate a decrease in stride length, foot inclination angle, center of mass (COM) vertical excursion, and the distance from heel to COM at initial contact will be observed⁸.

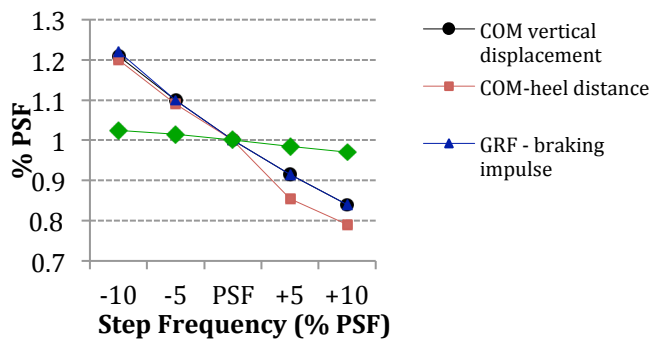


Figure 2. Biomechanical changes that occur due to a modification of step rate. It is likely that a decrease in center of mass (COM) vertical displacement and COM heel distance are two of the biggest contributing factors to a decrease in ground reaction forces (GRF), including braking impulse and the peak vertical GRF. All data are reported as a percentage of preferred stride frequency (PSF).⁸

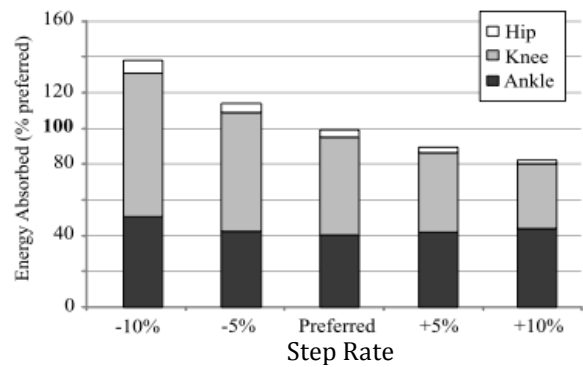


Figure 3. Changes in mechanical energy absorption during loading response with changes in step rate. All data are reported as a percentage of the preferred step rate condition.⁸

reduction occurring at the knee⁸ (Figure 3). Thus, adopting a step rate greater than one’s preferred may prove beneficial in reducing the risk of developing a running-related injury or facilitating recovery from an existing injury^{1,4,6}. The reduced energy absorption at the hip and knee when running with an increased step rate may prove useful as an adjunct to current rehabilitation strategies for running injuries involving these joints and associated tissues. That is, injured runners could be instructed using a metronome to increase their step rate while maintaining the same forward velocity. The associated reduction in loading may enable injured individuals to continue running without aggravating symptoms, while receiving care for their injuries. Similarly, utilizing an increased step rate may prove beneficial following injury recovery as part of a progressive return to running.

Due to the significant impact that step rate has on running mechanics, it is crucial for clinicians to identify the step rate of a patient who is seeking care for a running related injury. A typical visit to the University of Wisconsin’s Runners’ Clinic consists of a physical assessment to identify any structural or strength and flexibility deficits. Next the patient will run on a treadmill while the clinician conducts a video analysis to determine any asymmetries or imperfections in the individual’s running mechanics that may be associated with the patient’s symptoms. It is during this portion of the visit that step rate plays an important role in the analysis.

CLIENT INFORMATION

Our design project this semester has been proposed by Dr. Bryan Heiderscheit, a professor for the Doctor of Physical Therapy program at the University of Wisconsin- Madison. Dr. Heiderscheit is the co-director of the University of Wisconsin Neuromuscular Biomechanics Laboratory and the director of the UW Runners’ Clinic. His research interests are focused on the analysis of human movement with an emphasis is on the underlying mechanics and how they relate to injury. The majority of his patients are seeking assistance in the rehabilitation of a running related injury.

PROBLEM STATEMENT

Our proposed design project is to create a device that will identify a runner's step rate as they are running on a treadmill. Step rate feedback will be provided to the patient and the clinician in real time to assist in the gait analysis. It is intended that this device will be used in clinical settings, such as the UW Runners' Clinic. Future adaptations of our design will allow for portability and versatility of implementation in other clinics.

COMPETITION

Currently step rate is visually identified by the clinician. Dr. Heiderscheit must count the number of steps that are taken by the patient over a 30 second time interval. Although it may seem like a relatively short amount of time, it becomes a significant drawback during the visit. It is difficult for the clinician to manually identify a patient's step rate as they must refrain from talking with the patient while they are counting steps. In addition it is often inaccurate because steps may be miscounted, requiring the clinician to recount. Therefore, having a device to automatically monitor and update a patient's step rate would be ideal, saving the clinician valuable time and increasing accuracy.

There are a few different devices currently available on the market that are capable of identifying step rate, including pedometers and force instrumented treadmills. Pedometers can identify the number of steps an individual takes using an accelerometer. There are a select few pedometers that can identify step rate that are available on the market. The most commonly used pedometer to identify step rate is manufactured by Garmin.⁷ This company has created a multicomponent device consisting of a footpod that is worn on the individual's shoelaces and a watch that is used to display the step rate. Although this would accurately identify step rate there are several complications with these devices as well. Again it is time consuming to outfit the patient with the components necessary to identify step rate with a pedometer. In addition step rate is displayed on the watch worn by the patient, preventing the clinician from easily viewing the patient's step rate. Furthermore, with four different clinicians treating patients in the UW Runners' Clinic, it may become cumbersome to keep track of one device, providing all clinicians with a pedometer may become expensive.

One of the most accurate methods for identifying step rate is with the use of a force-instrumented treadmill. From this device, the vertical ground reaction forces can be monitored and recorded to determine the number of steps taken by an individual. The greatest obstacle in using an instrumented treadmill in the clinical setting is the cost. Although an instrumented treadmill is present in the lab where patients are seen in the UW Runners' Clinic, it is not practical to use it in the clinical setting, as it is not cost effective to operate an instrumented treadmill for clinical purposes only. Limitations of the mechanical components of the instrumented treadmill in the UW Neuromuscular Lab prevent it from functioning properly when used for extended periods of time. When used for research purposes, data collections often last only 5-10 minutes, the maximum amount of time the treadmill can be safely and reliably operated. However, a gait analysis for a patient at the Runners' Clinic could last up to 30

minutes, compromising the patient’s safety. Another complication of using an instrumented treadmill to identify step rate is the fact that data collected from the force plates of the treadmill is currently post-processed and analyzed to output the runner’s step rate. However, one criterion for our project is to provide real-time feedback for the patient and clinician, illuminating the possibility of using the force instrumented treadmill for our project.

Due to the limitations of the devices currently available on the market, we have created a solution that will effectively and efficiently identify the step rate of an individual while running on a treadmill. It is our intention that this device will be used in various runners’ clinics including the UW Runners’ Clinic, to assist in the analysis of a runner’s biomechanics.

DESIGN SPECIFICATIONS

Our design must meet several parameters. First it must be compatible with the treadmill (created by Standard Industries) in Dr. Heiderscheit’s lab. Our device must not compromise the infrastructure of the treadmill. In addition, it must not interfere with the runner on the treadmill in any way. It must accurately identify the step rate of an individual, regardless of the runner’s position on the treadmill. Furthermore, the client has requested that the runner’s step rate feedback is updated frequently. The identified step rate will be displayed in real time so that it is clear to the patient and clinician how many steps the individual is taking per minute. The design flow can be seen in Figure 4.

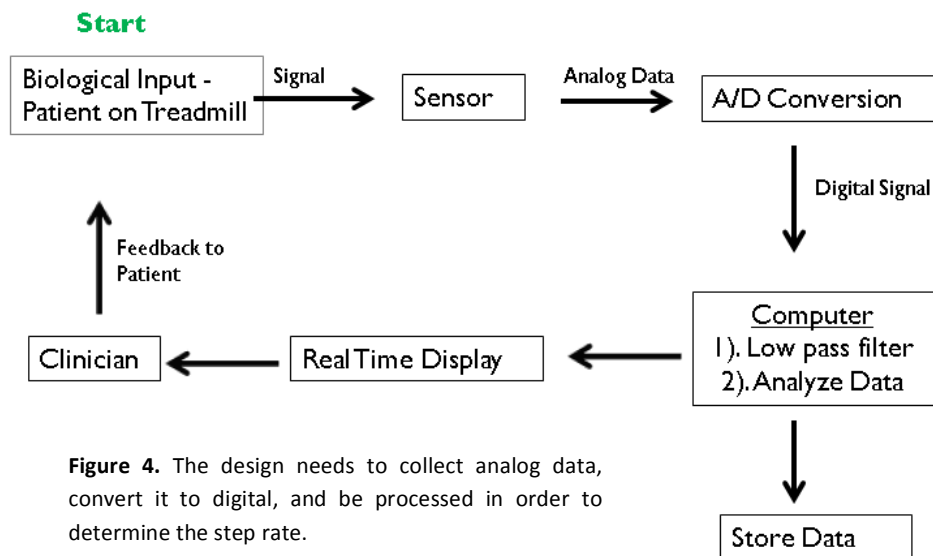


Figure 4. The design needs to collect analog data, convert it to digital, and be processed in order to determine the step rate.

Overall, the design will utilize a biologically relevant signal from a patient on the treadmill. There are several potentially relevant, detectable signals including vibrations of the treadmill, optical detection of leg or foot placement on the treadmill, and the acoustic information of footsteps on the treadmill. These analog signals will be sent through A/D conversion and to a computer system for further processing. Extraneous signal noise will be filtered out leaving the biologically relevant portion of the signal to be used in step rate calculations. This step rate data will be displayed and updated in real time to the clinician and

patient to provide necessary feedback. This design flow requires the choice of the sensor type and data processing software from several options. These sensor and data processing alternatives will be discussed in the following sections.

SENSOR ALTERNATIVES

Our design process began with the consideration of various methods of detecting footsteps on a treadmill. Detection modes we have compared include the use of an accelerometer, optical sensor, and acoustic sensor.

ACCELEROMETER

The vibrations of the treadmill incurred from each footstep of the runner can be detected by an accelerometer. By securely affixing an accelerometer to the treadmill, the sensor will experience the treadmill's vibrations resulting from each footstep. A properly calibrated accelerometer will allow the conversion of the biologically relevant vibrations of the treadmill to a relative output voltage that can be processed and used to compute step rate. This method of step detection will introduce several new design variables including sensor placement on the treadmill and the method of attachment. Accelerometer placement will influence the magnitude and relevance of the signal detected.

An accelerometer can also be superficially attached to the runner's anterior tibia to measure other relevant gait cycle accelerations that could be used to calculate step rate. Although this sensor placement will likely detect less noise in the signal, taking time to accurately place the sensor on each patient may be an inefficient use of clinical time.

OPTICAL SENSOR

Another method proposed to identify step rate is the use of an optical sensor system. An optical beam spanning the width of the treadmill will be broken each time a runner steps through the beam's path. This binary detection of foot placement could then be used to calculate the runner's step rate. This design will require mounting a laser-beam emitter and receiver onto the upper side of the treadmill and will provide signal consistency for various patients. One drawback of this design is its dependency on the positioning of the runner on the treadmill. If the runner strays forward or backward on the treadmill, the beam will be broken at different points of the gait cycle and would likely provide an inaccurate representation of step rate.

ACOUSTIC SENSOR

Our final design alternative is the use of a microphone to detect the audio signal of each step. By attaching a microphone near the runner's position of initial foot contact on the treadmill, the sound of each footstep can be recorded. This signal would then be processed and used to calculate step rate. Obvious drawbacks of this design include the presence of extraneous audio signals in the environment. Also, the variation of step force that depends on each runner's size, gait, and speed will affect the sound energy level detected, adding variability

to the signal. In order to reliably use this audio signal to calculate step rate, our processing techniques will need to separate the frequencies of interest from external broadband noise.

DATA PROCESSING ALTERNATIVES

Data received from any of the previously described sensor modalities will need to be processed and used to calculate step rate. Processing will include signal filtering as well as algorithms to calculate step rate from the signal data. We have considered several data processing alternatives including Java, Matlab, and LabVIEW.

JAVA

Using JAVA for our data processing software provides programming flexibility and many data presentation alternatives. However, with only limited pre-written code, most of our program will likely need to be written from the ground up. In addition, our analysis program will need to be optimized for the processing power and efficiency necessary for handling live data. Our literature searches have revealed a previously created Java platform, IU Sense, designed at the International University in Germany in 2003². This platform has been designed for processing input from multiple accelerometers but would need to be adapted to our application. Furthermore, IU Sense's display performance has been reported to be limited due to the limitations of Java's display software packages².

MATLAB

Matlab offers many built in features and functionalities for data processing that will improve the design of our software. Moreover, Matlab will have the necessary processing power to handle large data sets in a short amount of time. Drawbacks to using Matlab for data processing include its limited data presentation capabilities. More importantly, real-time data processing in Matlab is not as widely used as other software programs, thus it is not as widely developed for multiple applications.

LABVIEW

National Instrument's LabVIEW will provide even more built in data processing functionalities than Matlab, including various filtering options. LabVIEW has also been designed with live data acquisition and processing in mind and should be easily interfaced with Dr. Heiderscheit's current data acquisition system. Furthermore, LabVIEW offers many data presentation options that will be suitable for live data feedback for the subject and clinician.

DESIGN MATRICES

Our sensor alternatives have been ranked based on sensitivity, signal to noise ratio, feasibility, cost, and reliability. Based on our client's design requirements we have weighted sensitivity, signal to noise ratio, and reliability the highest in our design matrix. Choosing a design based on these parameters will ensure a clear and biologically relevant signal. As seen in Table 1, the accelerometer design has scored high in regards to signal-to-noise ratio and sensitivity scores. The sound sensor and optical sensor designs have scored low in signal-to-

noise ratio and feasibility categories, respectively. Based on the design matrix results and our client’s preference, we will pursue the accelerometer sensor design alternative.

	Weight (%)	Accelerometer	Sound	Optical
Sensitivity	20	16	16	16
Signal:Noise	40	34	16	38
Feasibility	15	13	10	9
Cost	5	4.5	4.5	2
Reliability	20	12	5	9
Total	100	79.5	51.5	74

Table 1. Sensor design matrix

After choosing to use the accelerometer detection modality, we considered possible accelerometer attachment locations, including attachment to the subject’s anterior tibia and attachment underneath the treadmill. Sensor preparation time is the most important parameter in the design matrix since clinics strive for time efficiency (Table 2). Any time-consuming sensor attachment process would be an ineffective use of clinical time. Other important factors include the signal-to-noise ratio and biological relevance of the signal. Tibial attachment of the accelerometer will more directly measure the characteristics of the step cycle than attachment to the treadmill. However, inconsistent placement on the tibia from patient to patient will introduce variation into the signal as well as signal noise that will complicate data processing. On the other hand, accelerometer attachment to the treadmill, although a less direct measurement of step cycle, will provide consistency in the measured signal from patient to patient. Placement of the accelerometer underneath the treadmill may provide a less direct measurement of the biological signal but will require the least preparation time. Based on our client’s emphasis on preparation time, we will attach the accelerometer to the underside of the treadmill.

	Weight (%)	Tibia	Under Treadmill
Signal:Noise	25	20	15
Preparation Time	40	25	40
Biologically Relevant Signal	35	30	25
Total	100	75	80

Table 2. Accelerometer location design matrix

The most important aspects of data processing for our application are the software’s real-time processing and data presentation capabilities. Available built-in functionality is also a beneficial aspect of data processing choice. LabVIEW excels in these categories unlike Java and Matlab and is therefore the clear choice for our data processing needs. Table 3 shows the results of our software evaluation.

	Weight (%)	LabVIEW	Java	Matlab
Real-time Processing	40	35	25	20
Data Presentation	30	27	25	10
Built-in Functionality	20	15	10	10
Flexibility	10	8	10	5
Total	100	85	70	45

Table 3. Data processing design matrix

PRELIMINARY DATA

Preliminary testing of sensor placement on the treadmill has confirmed our design choices. Placement of the sensor on a runner’s anterior tibia gives a clear signal and relatively high signal-to-noise ratio (Figure 5). However, as previously emphasized, this placement method will require an inefficient use of clinical time. We hoped to use the fidelity and relevance of this tibial acceleration measurement as a baseline for comparison to data recorded from an accelerometer placed underneath the treadmill.

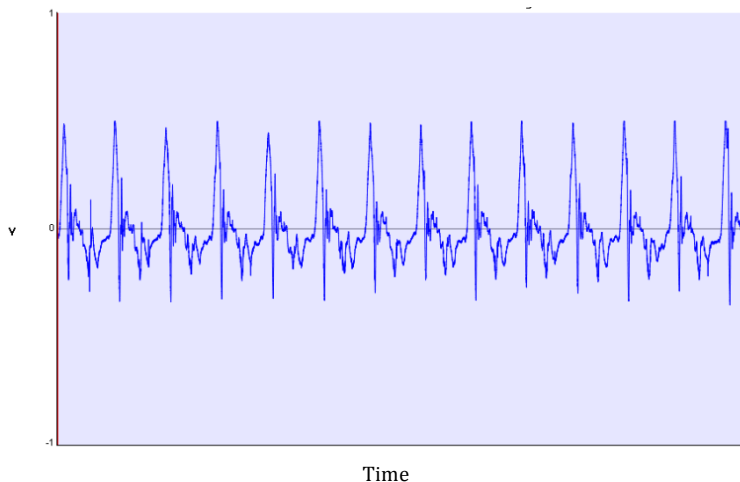


Figure 5. Placement of the accelerometer on the subject’s anterior tibia while running.

Next, the accelerometer was fastened underneath the treadmill directly below the mid-span region of initial foot contact to a lengthwise support beam. Data from this accelerometer

placement can be seen in Figures 6 and 7. The signal to noise ratio appears highest at lower speeds, as seen in the data from a walking subject in Figure 6. As expected logically, vibration amplitude, temporal duration of reverberating vibrations, and noise all increase with the speed of the runner (Figure 7). This result can be attributed to the increase in step contact force associated with higher runner speeds.¹⁰ Although the undesired reverberating vibrations and noise profile increase with runner speed, the desired foot contact signal remains salient and effective for step rate detection.

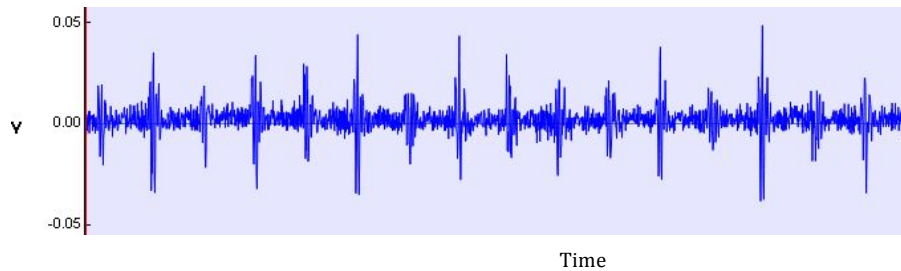


Figure 6. Placement of the accelerometer underneath the treadmill while the subject is walking.

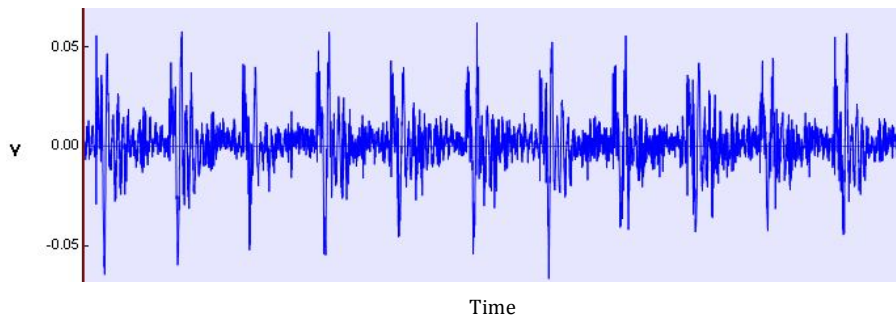


Figure 7. Placement of the accelerometer underneath the treadmill while the subject is running.

FINAL DESIGN

Although LabVIEW was the initial choice for the data processing method, several software compatibility issues have caused the shift to Matlab. The real time data collected by LabVIEW had a signal drift that could not allow for an accurate step rate calculation. With the use of Matlab, the design group was able to correctly post process data with a known step rate. This was done by the use of a low pass filter followed by the calculation process described below.

DATA PROCESSING

CALIBRATION

Once a patient has stepped onto the treadmill and began to run, the clinician will click the “Calibrate” function. This function acquires a small data sample (5-10 seconds) for the calculation of the threshold and time delay parameters. After collection of this preliminary data set, the program sends the absolute value of the signal through a low-pass butterworth filter (corner frequency = 100 Hz). This filter will attenuate any unwanted noise that could result from accelerometer resonance. After the data is filtered, the program determines the maximum and average voltages. With these values calculated, the threshold according to:

$$\text{Threshold} = \text{Average Voltage} + (\text{Max Voltage} - \text{Average Voltage}) * 0.12$$

After the threshold voltage is determined, this same data set is evaluated to determine which data points are greater than the threshold voltage, in order to find all of the data peaks. These data peaks, however, may either represent initial foot contact or simply reverberating vibrations from foot contact. In order to determine which peaks correspond to initial foot contact, an algorithm is used to calculate a time delay. This time delay allows all voltage peaks above the threshold after initial foot contact to be ignored if within the time delay window. This algorithm calculates the time gaps between voltage peaks, which are used in the time delay calculation. As seen in Figure 8, the time gaps between initial foot contact and reverberating vibrations are much smaller than the gap between the last reverberating vibration and the next foot contact. Therefore, the algorithm uses the minimum and maximum gap times to calculate an appropriate time delay window according to:

$$\text{Time Delay} = \text{Min time gap} + (\text{Max time gap} - \text{Min time gap}) * 0.7$$

These calibrated threshold and time delay values are saved into the GUI and passed into the Monitor function when the clinician is ready to monitor the patient’s step rate (see Appendix B for Matlab code).

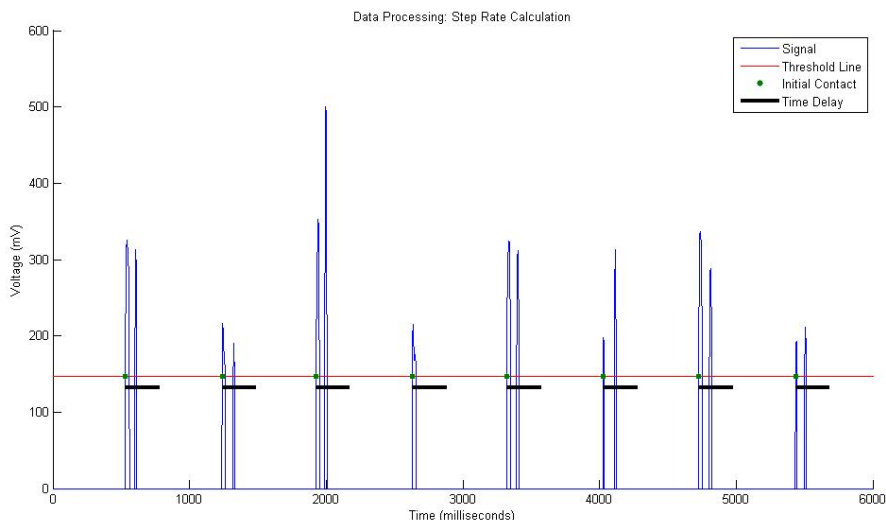


Figure 8: Step Rate Calculation. The time delay is calculated and implemented in order to prevent multiple “steps” from being counted due to a single foot strike.

MONITORING STEP RATE

After calibrating the threshold and time delay parameters, the clinician is able to click the “Monitor Step Rate” button on the user interface. This function continuously reads in three second data sets to be processed and used to calculate step rate. Therefore, the clinician and patient receive a step rate update every 3 seconds. Each three second data sample is processed in the same way. First, the absolute value of the data is filtered through a low-pass butterworth filter (corner frequency = 100 Hz). Next, all data points less than the calibrated threshold are set to 0. The algorithm then determines which data points potentially correspond to initial foot contact by finding the points with a value greater than the threshold whose preceding value equals 0. The data is then stepped through to determine the first step, after which the data peaks are ignored until after the calibrated time delay window. The timestamp of each step is recorded in this manner. After processing the entire 3 second data set, the code calculates the step rate according to:

$$\text{Step Rate} = \frac{\text{Number of steps} - 1}{\text{Time between first and last step (seconds)}} * 60 \frac{\text{seconds}}{\text{minute}}$$

The graphical display is updated with every new step rate calculation or every three seconds. The monitor function is designed to pass the time delay window over to the next data sample. This is important for data sets in which a step occurs at the very end of the three second window. The time delay is passed on so that the reverberating vibrations that occur at the start of the next data set are not mistaken for initial foot contact but will instead be ignored.

TESTING AND RESULTS

The design specifications of the device were determined by various tests performed in Dr. Heiderscheit’s lab with the clinical treadmill. Tests were performed to decide upon the placement of the accelerometer, assure that the signal is strong enough for different speeds, and determine the best orientation of the accelerometer.

PLACEMENT OF THE ACCELEROMETER

To determine the best placement for the accelerometer on the treadmill, we performed the following test. The accelerometer was placed in three different locations on the treadmill. All three were along the treadmill’s middle support beam in the front, middle, and back. A runner ran while the accelerometer was attached at the three different locations and data was recorded. The runner kept a constant step rate using a metronome and ran at a 4.95 meters/second pace in each trial so that accelerometer placement was the only changing variable. The results can be seen in Figure 9. Attachment near the front resulted in the best signal because that is the area that the foot most often strikes.

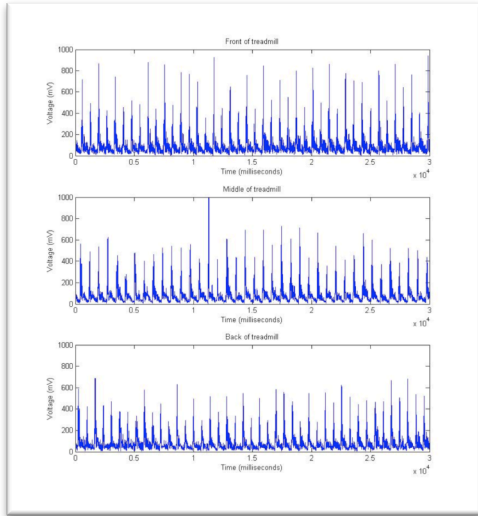


Figure 9. Acquired signal from attachment of the accelerometer on the front, middle, and back of the treadmill's middle support beam. Attachment on the front resulted in the best signal.

ALTERING SPEEDS

Various magnitudes of signal are produced when a runner alters their speed while keeping a constant step rate. The following test was performed to confirm that the relevant signal would be strong enough at different speeds. A runner ran at three different speeds: 2.68 meters/second, 3.35 meters/second, and 4.95 meters/second. The accelerometer was kept in the same location and the runner kept a constant step rate using a metronome in each trial so that the speed was the only changing variable. The results from the test can be seen in Figure 10. The test concluded that although the altering speeds resulted in different magnitudes of signal, the magnitudes were still great enough to determine a step.

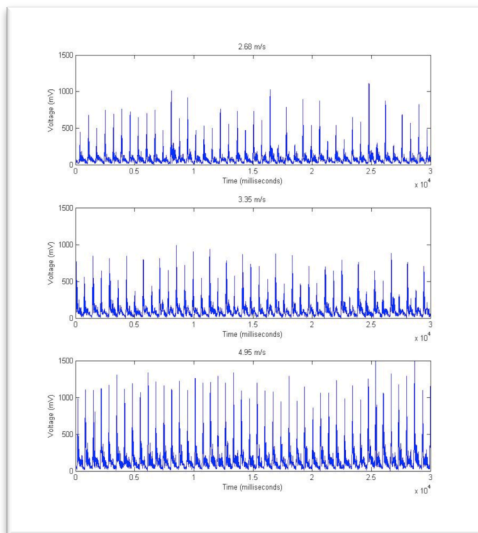


Figure 10. Comparison of the signal at different speeds, while maintaining a constant step rate. Relevant signal was acquired at all speeds.

ORIENTATION

Through logic, we determined that the accelerometer should be attached perpendicular to the plane of the belt on the treadmill. To confirm this, we tested the accelerometer in the direction perpendicular and parallel to the plane of the belt. All other variables were kept constant. The resulting data can be seen in Figure 11. Our predetermined notions were correct. With the accelerometer attached in the direction parallel to the plane of the running belt, virtually no biologically relevant data can be retrieved.

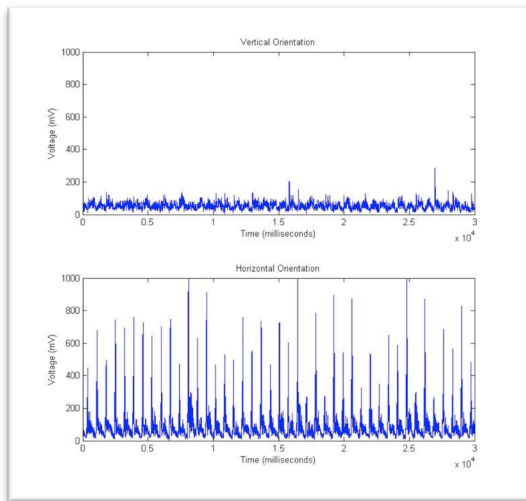


Figure 11. Comparison of the signal with the accelerometer attached parallel and perpendicular to the plane of the running belt. The most biologically relevant signal was collected with the latter.

FUTURE WORK

The focus of next semester will be improving the accuracy and usability of our device through acquiring accelerometer data in real-time, optimizing signal filtering, creating a visual display for the runner, and performing further testing. In this section, the future development and testing of each component will be discussed.

REAL-TIME DATA PROCESSING

The first step in moving forward with the design is to properly acquire the signal from the accelerometer in real-time. To do this, the MatLab program on the computer needs to be updated to the most current version. The required toolbox is already installed, but licensing issues still need to be worked out. Currently, the signal is collected using the clinic's existing data acquisition system, Cortex, and post processed using the code in Appendix. This code will also work for the real-time data; it is just a matter of making the correct connection with MatLab.

IMPROVE SIGNAL FILTERING

Future improvements to the signal filtering process have the potential to greatly simplify the step rate calculation algorithm. For example Figure 12 shows the result of a series of more complex filtering steps. The top subplot of the figure shows the unfiltered accelerometer signal,

and the bottom subplot is the resulting filtered signal. If optimized, this filtering sequence could resolve each step into a single voltage plateau which would eliminate the need for a time delay to handle multiple voltage peaks per step.

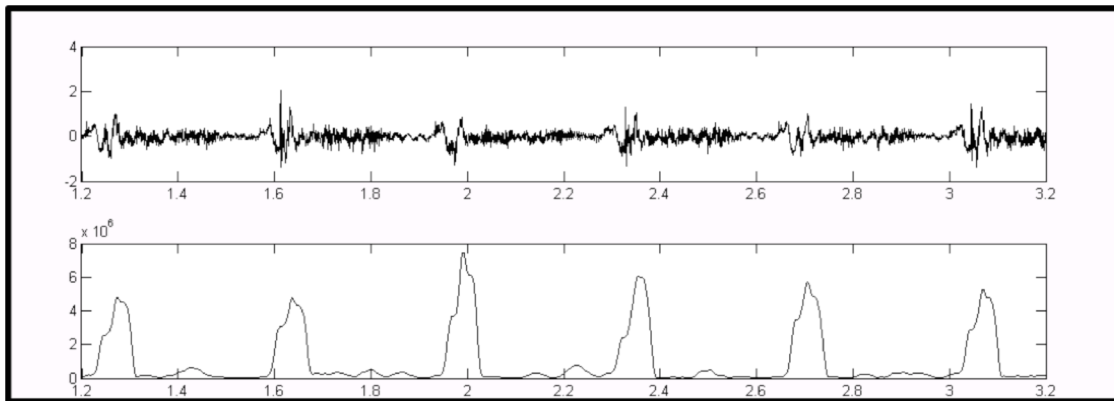


Figure 12. Comparison of the resulting and final signal.

The signal manipulation process begins with a low pass filter to attenuate the unwanted noise resulting from resonating accelerometer vibrations. The resulting data after the low pass filter is shown in Figure 13.

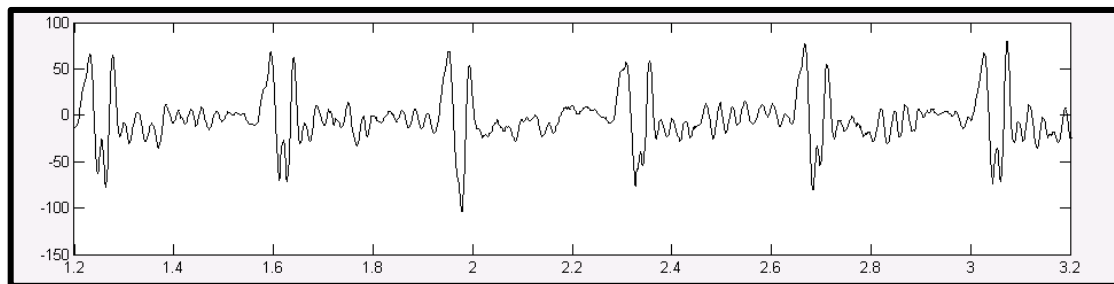


Figure 13. Resulting data after the low pass filter

The next step of the filtering processes is a DC blocking filter which is followed by signal integration. Integration is used to further eliminate noise and accentuate the initial foot strike signal. In order to effectively use the integration, however, the constant DC voltage must be blocked using a DC filter. The resulting signal is shown in Figure 14.

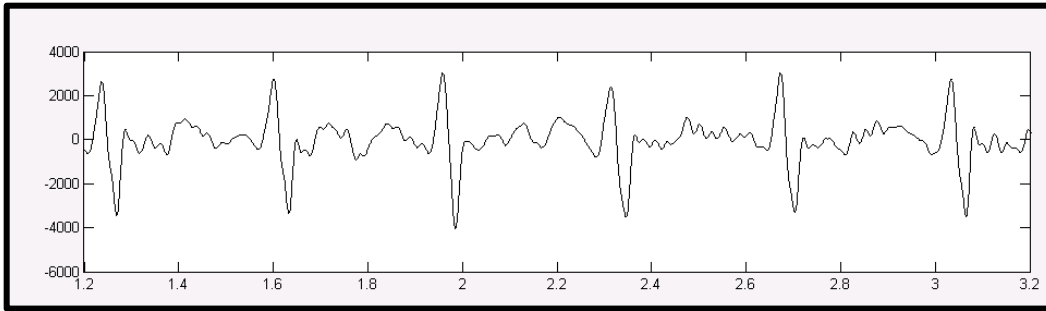


Figure 14. The resulting signal after integration

To better accentuate the peak of each foot strike, the signal is squared, resulting in the graph represented in Figure 15. Note in the figure that the squaring function produces double peaks for each step as a result of the squared negative portion of the signal.

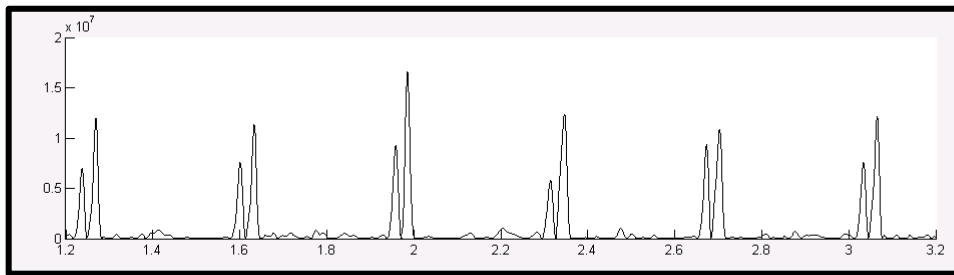


Figure 15. Result of the signal squaring function

Finally, to merge the double peaks into one, an averaging function was used. Each data point was averaged with the previous 70 data points. This value was chosen on a trial and error basis until a single peak was present. Figure 16 shows the result of this averaging and of the final signal.

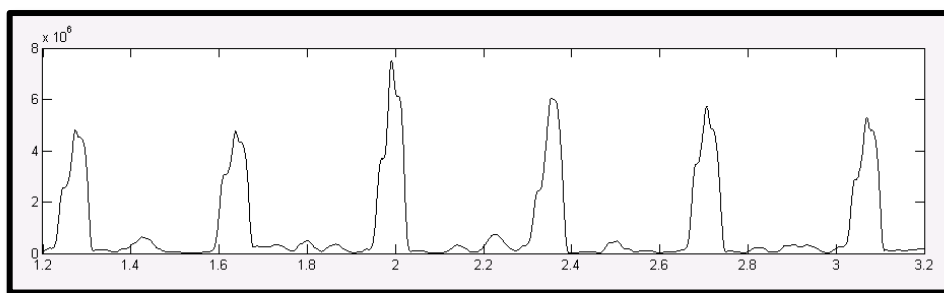


Figure 16. Final signal

RUNNER- DEVICE INTERFACE

After the signal to noise ratio is optimized with the improved filtering and the device is working properly for each individual, a visual platform will be created to provide the runner with

useful feedback. The visual display is currently only providing visual feedback for the clinician with step rate in the form of a raw number. To many runners, the term 'step rate' may not bear much significance as it can be a difficult concept to understand when first introduced. Telling the runner to increase or decrease their step rate from a raw number will therefore be difficult. Instead, giving the runner visual feedback in the form a speedometer and displaying a "green zone" with limits representing the values of step rate to stay within will give them better visualization. Displaying this information to the runner as they are being taught to alter their step rate and stride length will be useful because the runner can easily identify if they need to increase or decrease their step rate.

FURTHER TESTING

Finally, further testing will be done with the device. First, the device needs to be tested with diverse individuals. With each individual, the signal will undoubtedly vary. The signal magnitude recorded when a 300 pound line-backer is running will differ from a 100 pound long distance runner. There are many factors to consider including weight, stride length, step rate, height, etc. The device will be tested with individuals that have diverse body types and running styles to assure that the program is universal. We will have to analyze how the signal differs and determine if our code is robust enough to provide the correct step rate value for each individual.

Secondly, the final product will be marketable and will be able to be implemented in various running clinics. For this to be the case, the device will be tested on different clinical treadmills to assure that it can be easily step up in a spectrum of clinics.

OBJECTIVES

The creation of a step rate monitoring system for running analysis will improve the overall clinical experience. The step rate monitor will eliminate the need for the clinician to manually count step rate, allowing them to focus more of their time with the runner. Furthermore, by providing the runner with useful visual feedback, the process of learning how to increase or decrease step rate will be simplified. For this to be a possibility, the system will be easy to use, as well

REFERENCES

1. Buff, H., L. Jones, and D. Hungerford. Experimental determination of forces transmitted through the patello-femoral joint. *J Biomech.* 21:17-23, 1998.
2. Caracas, A, EA Heinz, P Robbel, A Singh, F Walter, P Lukowicz. "Real-time sensor processing with graphical data display in Java." *Signal Processing.* 2003. 14-17; 62-65.
3. Corbett, J., S. Vance, M. Lomax, and M. Barwood. Measurement frequency influences the rating of perceived exertion during sub-maximal treadmill running. *Eur J Appl Physiol.* 106:311-313, 2009.
4. de Leva, P. Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of Biomechanics.* 29:1223-1230, 1996.

5. Derrick, T.R., J. Hamill, and G.E. Caldwell. Energy absorption of impacts during running at various stride lengths. *Medicine and science in sports and exercise*. 30:128, 1998.
6. Dierks, T.A., K.T. Manal, J. Hamill, and I.S. Davis. Proximal and distal influences on hip and knee kinematics in runners with patellofemoral pain during a prolonged run. *J Orthop Sports Phys Ther*. 38:448-456, 2008.
7. Garmin. <http://www.garmin.com/us/> 11 December, 2011.
8. Heiderscheit BC, Chumanov ES, Michalski MP, Wille CM, Ryan MR. Effects of Step Rate Modification on Running Mechanics. *MSSE*. 2001. 43:2:296-302.
9. Messier SP, Legault C, Schoenlank CR, Newman JJ, Martin DF, DeVita P. Risk factors and mechanisms of knee injury in runners. *Med Sci Sports Exerc*. 2008 Nov;40(11):1873-9.
10. Nilsson J, Thorstensson A. Ground reaction forces at different speeds of human walking and running. *Acta Physiol Scand*. 1989. 136(2):217-27.
11. Souza, R.B., and C.M. Powers. Differences in hip kinematics, muscle strength, and muscle activation between subjects with and without patellofemoral pain. *J Orthop Sports Phys Ther*. 39:12-19, 2009.
12. Taunton JE, Ryan MB, Clement DB, McKenzie DC, Lloyd-Smith DR, Zumbo BD. A retrospective case-control analysis of 2002 running injuries. *Br J Sports Med*. 2002;36:95-101.
13. USA Track and Field Road Running Information Center. State of the sport report [online]. Available from URL: <http://runningusa.org/node/57770#58008> [Accessed 2010 Nov 28].
14. Walker PS, Rovick JS, Robertson DD. The effects of knee brace hinge design and placement on joint mechanics. *J Biomech*. 1988;21:965-74.

APPENDIX

A. PROJECT DESIGN SPECIFICATIONS

Project #44- Step rate monitor for treadmill

December 14th, 2011

Team: Carmen Coddington, Joel Schmocker, Bryan Jepson, Christa Wille

Client: Dr. Bryan Heiderscheit

Advisor: Professor Mitch Tyler

Function:

Our proposed design project is to create a device that will identify a runner's step rate as they are running on a treadmill. This device is intended to be used in clinical settings, such as runner's clinics. Additional capabilities of our device will include quantifying the relative magnitude of the ground reaction force. This information can be used as real-time visual

feedback for patients as they are being taught to alter their stride to minimize ground reaction forces while running.

Client Requirements:

- Real time identification of runner's step rate while running on a treadmill
- Quantify relative magnitude of ground reaction forces while running on treadmill
- Must not interfere with patient's running mechanics
- Securely mounted to treadmill
- Visually appealing
 - Device should be hidden from view on the internal structure of the treadmill
 - Simple, easily understood display of step rate
- User friendly software that can be used by multiple clinicians

Design Requirements:

1) Physical and Operational Characteristics

a) *Performance requirements*

- i. Accurately measure step rate
- ii. Display real-time visual feedback to the clinician and patient
- iii. Easily operated by multiple clinicians

b) *Safety*

- i. Non-distracting visual display
- ii. Components should not detract from the safety features of the treadmill
- iii. Device attachment should not compromise the durability of the treadmill
- iv. Should not interfere with patient's running mechanics

c) *Accuracy and Reliability*

- i. Must accurately measure step rate within 2 steps/minute
- ii. Accurately relate resultant vibration magnitudes in the treadmill to ground reaction forces
- iii. Function should not be compromised after multiple uses and patients

d) *Life in Service*

- i. Match or exceed the life of a treadmill
- ii. 10 years

e) *Shelf Life*

- i. Not applicable

f) *Operating Environment*

- i. Clinical gait analysis setting
- ii. Biomechanics research lab
- ii. Dry environment

g) *Ergonomics*

- i. Easily maintained
- ii. Device must not interfere with runner
- iii. Display must not interfere with safety of the runner or cause the runner to alter his/her mechanics to view visual display

h) *Size*

- i. Must be contained within treadmill cover
- ii. 3 x 3 x 3 inches maximum

i) *Weight*

- i. Sensor-attachment system should be light enough for appropriate sensitivity to treadmill vibrations

j) *Materials*

- i. Computer with data processing capabilities (Matlab)
- ii. Display screen (TV or computer monitor)
- iii. Treadmill
- iv. Accelerometer
- v. Power supply for accelerometer
- vi. Data acquisition system

k) *Aesthetics*

- i. Accelerometer hidden from view
- ii. Visually pleasing display

2) Production Characteristics

a) *Quantity*

- i. One complete system

b) *Target Product Cost*

- i. \$200

3) Miscellaneous

a) *Customer*

- i. Runner's Clinics
- ii. Home users

- iii. Fitness centers
 - b) *Patient-related concerns*
 - i. Must not interfere with patients' running mechanics
 - c) *Competition*
 - i. Pedometers
 - a) Garmin systems, Olympus
 - ii. Force-plate instrumented treadmill

B. MATLAB CODE FOR DATA PROCESSING

```
function varargout = steprate(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @steprate_OpeningFcn, ...
                  'gui_OutputFcn',  @steprate_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before steprate is made visible.
function steprate_OpeningFcn(hObject, eventdata, handles, varargin)
% Choose default command line output for steprate
handles.output = hObject;

fileList_CreateFcn2(hObject, eventdata, handles);

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = steprate_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in monitor.
function monitor_Callback(hObject, eventdata, handles)
set(handles.calibrate,'Enable','off');
set(handles.monitor,'String','stop');
set(handles.monitor,'BackgroundColor','r');

t=timer('timerfcn','myWait(0)','StartDelay',3);    %timer

%get parameters
sampleRate = 2000;
threshold = str2double(get(handles.threshold,'String'));
timeDelay = str2double(get(handles.timedelay,'String'));
indexDelay = timeDelay*2000/1000;
```

```

fileList = get(handles.fileList, 'String');
fileName = fileList(get(handles.fileList, 'Value'));
fileName = fileName{1};
dirPath = 'I:\Win\Desktop\bme design\data';
cd(dirPath);
dataFull = load([dirPath '\' fileName]);
if size(dataFull,2) == 26
    dataFull = dataFull(:,26);
else
    dataFull = dataFull(:,2);
end

dataFull = abs(dataFull);

[B,A]=butter(4, .1, 'low');
dataFull = filtfilt(B, A, dataFull);

n = length(dataFull)/2000/3;

nextData = 0;

for i=1:n
    if get(handles.monitor, 'Value') == 0
        break
    end

    start(t);
    wait(t);

    a = (i*3000)-2999;
    b = a+2999;
    data = dataFull(a:b);
    data(data<threshold)=0;

    stepIndices=zeros(1);
    for i=2:length(data)
        if i>= nextData && data(i)~=0 && data(i-1)==0
            stepIndices(end+1)=i;
            nextData = i+indexDelay;
        end
    end

    nextData = nextData - 2999;

    %calculate step rate
    totalIndices = stepIndices(end) - stepIndices(2);
    totalTime = totalIndices/sampleRate;
    stepRate = (length(stepIndices)-2)/totalTime * 60;
    %Subtract 2 from length(stepIndices) because the first index is 0 and the
    %last indicates the end of the last complete step

    %update display
    set(handles.steprate, 'String', num2str(stepRate));
    guidata(hObject, handles);
end
%update display
set(handles.calibrate, 'Enable', 'on');
set(handles.monitor, 'String', 'monitor step rate')
set(handles.monitor, 'Value', 0)
set(handles.monitor, 'BackgroundColor', 'g');
guidata(hObject, handles);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

% --- Executes on button press in calibrate.
function calibrate_Callback(hObject, eventdata, handles)
set(handles.monitor, 'Enable', 'off');

fileList = get(handles.fileList, 'String');
fileName = fileList(get(handles.fileList, 'Value'));
fileName = fileName{1};
dirPath = 'I:\Win\Desktop\bme design\data';
cd(dirPath);
data = load([dirPath '\' fileName]);

%data = load('run_167_3.txt');
if size(data,2) == 26
    data = data(:,26);
else
    data = data(:,2);
end

data = data(1:20000);
data = abs(data);

[B,A]=butter(4,.1,'low');
data = filtfilt(B, A, data);

%determine and set threshold voltage
maxVoltage = max(data);
find(data==maxVoltage);
positiveData = data(data>0);
avgPositiveVoltage = mean(positiveData);
threshold = avgPositiveVoltage + (maxVoltage-avgPositiveVoltage)*0.12;

%determine time delay
thrData = data;
thrData(thrData<threshold) = 0;
potStepIndices = zeros(1);
for i=2:length(thrData)
    if thrData(i)~=0 && thrData(i-1)==0
        potStepIndices(end+1)=i;
    end
end
potStepIndices = potStepIndices(potStepIndices~=0);

gaps = zeros(1,length(potStepIndices)-1);
for i=1:length(potStepIndices)-1
    gaps(i) = potStepIndices(i+1) - potStepIndices(i);
end

maxGap = max(gaps);
minGap = min(gaps);
indexDelay = minGap + (maxGap+minGap)*.7;
timeDelay = indexDelay/2000*1000;    %index converted to seconds then
milliseconds

%update display
set(handles.maxvoltage, 'String', num2str(maxVoltage));
set(handles.threshold, 'String', num2str(threshold));
set(handles.timedelay, 'String', num2str(timeDelay));

set(handles.monitor, 'Enable', 'on');
set(handles.monitor, 'BackgroundColor', 'g');
guidata(hObject, handles);

% --- Executes on selection change in fileList.
function fileList_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function fileList_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function fileList_CreateFcn2(hObject, eventdata, handles)
dirPath = 'I:\Win\Desktop\bme design\data';
cd (dirPath)
dirData = dir(dirPath);
dirName = {dirData.name};
dirName = dirName(3:length(dirName));

[sorted_names,sorted_index] = sortrows({dirData.name}');
handles.file_names = sorted_names;
handles.is_dir = [dirData.isdir];
handles.sorted_index = sorted_index;

guidata(hObject,handles)

set(handles.fileList,'String',handles.file_names,...
    'Value',1)

```