

PAPER • OPEN ACCESS

Gait rehabilitation monitor

To cite this article: P Leite *et al* 2019 *J. Phys.: Conf. Ser.* **1379** 012071

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Gait rehabilitation monitor

P Leite^{1,2}, O Postolache^{1,2}, J M Dias Pereira^{1,3} and G Postolache⁴

¹ Instituto de Telecomunicações, Lisboa, Portugal

² ISCTE – Instituto Universitário de Lisboa, Lisboa, Portugal

³ Escola Superior de Tecnologia de Setúbal, Instituto Politécnico de Setúbal, Setúbal, Portugal

⁴ CBIOS Universidade Lusofona, Lisboa, Portugal

Email: opostolache@lx.it.pt

Abstract. This paper presents a simple wearable, non-intrusive affordable mobile framework that allows remote patient monitoring during gait rehabilitation by doctors and physiotherapists. The system includes a set of 2 Shimmer3 9DoF Inertial Measurement Units (IMUs), an Android smartphone and a developed app for collecting, primary processing of data and for persistence of data in a remote PostgreSQL database, which is available in a remote server and where further data processing is performed. This framework provides gait features classifier by invoking an implemented REST API available in the remote server. Low computational load algorithms based on Euler angles and filtered signals were developed and used for the classification and identification of several gait disturbances. These algorithms include the alignment of IMUs sensors data by means of a common temporal reference as well as heel strike and stride detection algorithms. After segmentation of the remotely collected signals for gait strides identification relevant features were extracted to feed, train and test a classifier for prediction of gait abnormalities using supervised machine learning type and Extremely Randomized Trees method.

1. Introduction

Gait analysis support recognition of normal or pathological patterns of walking, for medical diagnosis and treatment [1], physical therapy [2] and sports training [3]. The traditional scales used to analyze gait parameters in clinical conditions include variable degree of subjectivism during assessment. Progress in new technologies has given rise to devices and techniques which allow objective evaluation of different gait parameters, resulting in more efficient measurement and providing specialists with a large amount of reliable information on patients' gaits. A simple and accurate technological system would be helpful to increase efficiency and affordability of motor rehabilitation. Gait analysis using wearable wireless sensors can be an economical, convenient and effective way to provide diagnostic and clinical information for various health-related issues [4]. In section II of the paper several related works are presented. Section III summarizes the proposed method and system. Section IV describes the results and discussion and Section V the conclusions.

2. Related Works

Several gait analysis systems have been proposed over the years. Alcaraz et. al [5] proposed a system using an IMU Shimmer2R and an Android application for mobile gait performance evaluation and



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

feedback. In this system the extraction of important features from accelerometer raw data in both time and frequency domain was performed for gait classification, and each peak within a data window was processed to distinguish between normal and abnormal gait. A method for measuring human gait posture using wearable sensors was proposed by Takeda et al. [6]. The acquisition of the signals was realized with a tri-axial acceleration sensor and three gyro sensors aligned on three axes. These were placed at abdomen level and at lower limbs (both thighs, both shanks and both feet) to measure acceleration and angular velocity during walking [6]. Bonnet and Jallon presented a technique for recognition of different gait conditions from a body-worn sensor data. A sensor located at subject's shank was a combination of a 3-D accelerometer and a 3-D magnetometer [6]. Charry et al. [7] work have shown that inertial sensors have good performance comparable to video-based systems. In Nukala, et al. [4] work a custom designed low-cost wireless gait analysis based on inertial measurement unit (IMU) was used to collect the gait data for four patients diagnosed with balance disorders and from three normal subjects. Each subject has executed Dynamic Gait Index (DGI) tests while wearing the custom wireless gait analysis sensor (WGAS) [4]. Previous researches have shown that machine learning methodologies are effective for classification of different activities from sensor data [8].

3. System Description

The implemented measurement system for gait classification is a distributed computation system, which includes embedded sensing device that are Bluetooth compatible, edge computation platform and user GUI expressed by an Android OS device and by advanced processing and classification in a remote server. IMU as part of sensing device allow acquisition of the signals for gait monitoring. The signals from measurement channels are primary processed and Wi-Fi transmitted by the Android OS Platform that also assures the Internet connectivity. An Android app was developed for smartphone that invokes a REST API in a remote server. Java and Python programming languages in a server support data processing and classification using machine-learning. The developed system includes two Shimmer3 devices and a database. In figure 1 the system architecture including two IMU units, an Android OS computation platform (smartphone) and a remote server is presented.

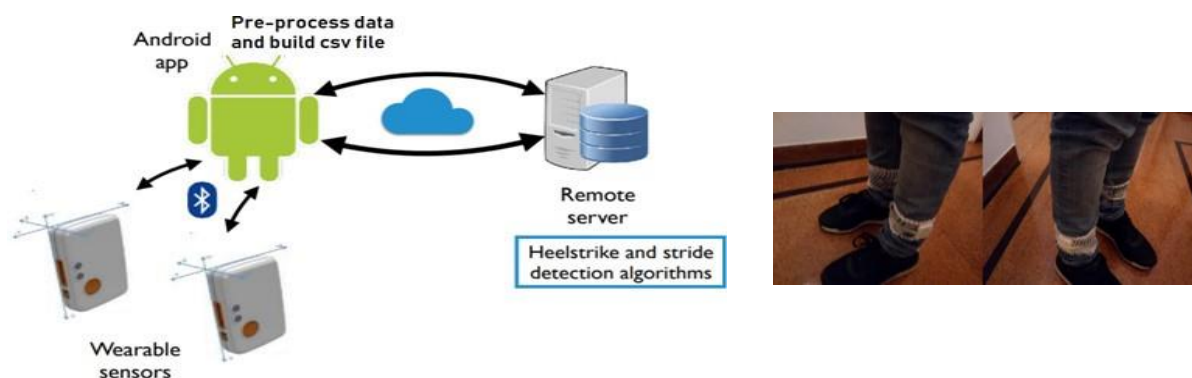


Figure 1. Gait Rehabilitation Monitoring System Block Diagram.

The software that runs in the Android OS device performs several tasks related to setup, data processing and packing, data transmission and some self-test functions that monitor the proper working of the system. Complex functions that require higher data processing capabilities are performed remotely by computation units as Laptop or remote server.

3.1. System Description - Hardware – IMUs

The hardware of the system includes 2 Shimmer3 IMUs. Shimmer3 is a small wireless sensing device, well suited for wearable applications. The integrated kinematic sensors, the storage and low-power standards-based communication capabilities of Shimmer3 IMUs are proper for development of motion capture applications, long-term data acquisition and real-time monitoring. 9DoF Calibration Application from Shimmer [9] was used to calibrate Shimmer3 devices.

3.2. System Description - Software

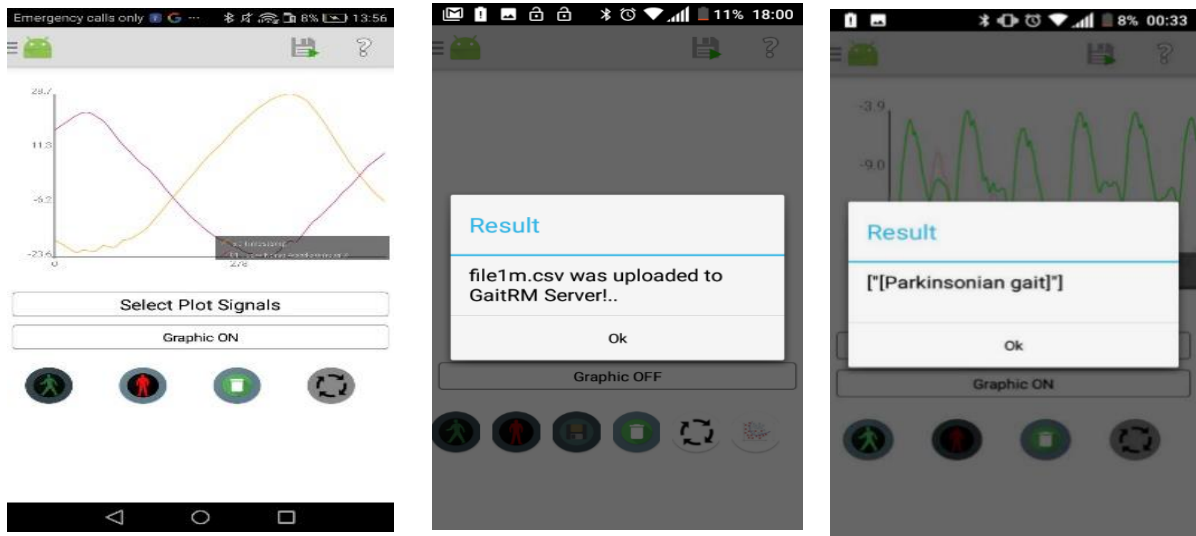
The APP for Android OS was developed using Multi Shimmer Template and the Shimmer drivers that provide Bluetooth connectivity to Shimmer3 devices. The *ShimmerDriver* and *ShimmerAndroidInstrumentDriver* that hold *ShimmerObject* (threads running connections between app and devices), *ShimmerBluetooth* (streaming over Bluetooth), *ObjectCluster* (that represents data from devices) and *Shimmer* (that represents a device) Java classes were used. Multiple Shimmer devices can be connected simultaneously to a device with Android OS, using multiple instances of the Shimmer Class object [10]. Shimmer drivers manage the connection with the app and data streaming over Bluetooth. Interactions with the remote server are executed using SSH2 implemented via JSch, Java Secure Channel and port forwarding and SFTP. Android *Fragments* were developed to implement the APP functionalities: *LoginFragment* and *RegisterFragment* for login and registration over SSH2. *WalkFragment* and *TrainFragment* have handlers to manage threads running the multiple Bluetooth connections with Shimmer3 devices, and to receive messages sent from the devices carrying the gait information. The received messages are synchronized using a common System timestamp reference, pre-processed and sent in a .csv to the remote server over SFTP. These two fragments layouts have the option to plot signals and YY Euler angles. *TrainFragment* has an *AlertDialog* to choose the classification label associated with each gait type and *WalkFragment* has an *AlertDialog* to show the test results. Figure 2 depicts the APP functionalities: Plot of the signals during the session, upload the data file on the server, and the display of classification results. Train classifier feature enables the APP to collect data and train a classifier model. The algorithm for gait classification is described in figure 3 and figure 4. Smartphone connects to the system devices using Shimmer drivers and Bluetooth, collecting tri-axial data from both devices that integrate accelerometers, gyroscopes and magnetometers. Data from the two Shimmer3 devices are pre-processed by Android device and transformed by the APP into a .csv file containing IMU Bluetooth address, device sampling-rate, gait classification labels, collected raw data, aligned timestamps and Euler angles. IMU signals alignment algorithm was developed and implemented. A common referential time was considered for IMUs and the obtained data from both devices was aligned to this common reference. A common temporal reference was considered as the instant at which the samples are received by the Android Smartphone [11] and the timestamp of arrival of each sample in the smartphone is collected.

Each IMU sample is associated with a time stamp according to the following relation:

$$ct = ws - sat \quad (1)$$

where ct is current timestamp; ws is start walking timestamp; sat is sample timestamp arrival in the smartphone.

Samples with timestamp prior to the moment gait starts are ignored. Euler angles are a convenient set of generalized coordinates for describing the orientation of a body-fixed coordinate frame [12]. In this application the Euler angles (yaw, pitch, and roll) were used to describe the legs movement based on the data in quaternion [13] format delivered by IMU Shimmer3 devices attached to the legs. On the APPs level a conversion is made from quaternion to Euler angles that are stored in .csv file that is sent to the system remote server by SFTP for off-line processing. Additionally, digital filtering has been considered. Thus a 4th order Butterworth bandpass digital filter characterized by 0.7 Hz low-cut off and 35 Hz high-cut off frequencies has been applied to the accelerometer's signals [7] to increase the signal-to-noise ratio which will help gait classification.

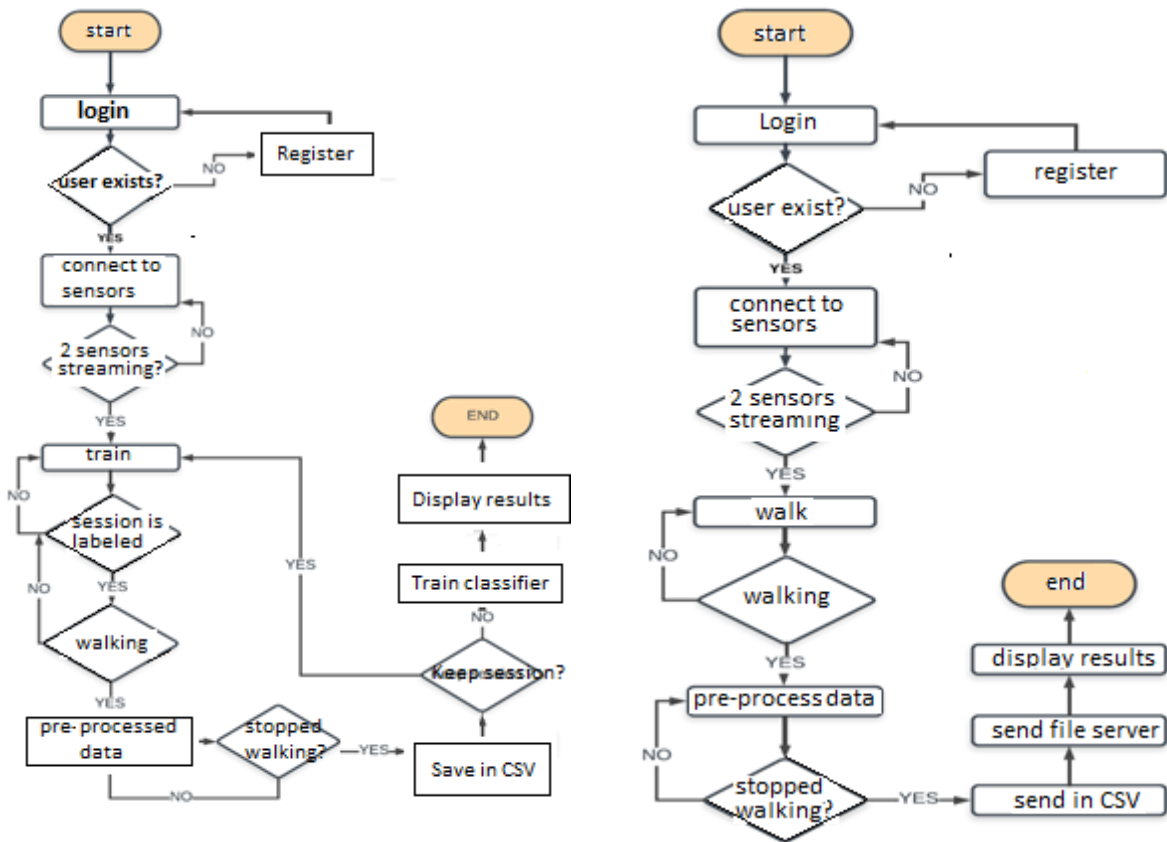


(a)

(b)

(c)

Figure 2. APP functionalities: (a) plot of the IMU signals; (b) upload the data as .csv file; (c) display classification results.



(a)

(b)

Figure 3. Algorithms flowchart: (a) gait classifier training flow; (b) gait session flow.

Additionally, 1st order Butterworth low-pass digital filter characterized by 10 Hz cut-off frequency has been applied to magnetometer signals and obtained Euler angles. These play a very important role in the detection of the heel strikes that occur during the gait cycles, which are useful to help segmenting the signal for stride identification and features extraction for training the classifier model. Euler angle from YY axis was used in this algorithm. The YY axis was chosen in this calculation considering the calibration procedure. A heel strike detection algorithm includes the following steps: i) calculation of the mean value of the discrete YY axis Euler signal; ii) determination of the signal indexes where values are below the mean value of the signals; iii) heel strikes detection, by calculating the maximum value between the lower and upper limits of the signal samples in noisy signals or in signals with many peaks, and elimination of peaks that may be misidentified as heel strikes (if two identified samples below the mean have more than two samples above the mean - initial adaptive distance = two samples, then there might exist a heel strike between them; at each iteration through samples determined in the second step, is performed the distance adjustment if the number of samples is above, in between or greater than current distance:

$$d = [d/2 + \text{abs}(l - n)]/2 \quad (2)$$

where d is distance, l is last sample, n is next sample; iv) storing the identified heel strikes in a buffer; v) calculation of the mean difference between heel strikes identified in the fourth step; vi) iteration through the heel strikes (indexes) to get the segments of the original discrete YY axis Euler signal between those heel strikes; vii) calculation of the heel strikes index for the original signal. After identifying the heel strikes, the strides can easily be obtained from the signal segments. Figure 4 shows the segments obtained from the acquired signal at each heel strike for the right and left legs. Each segment displayed in figure corresponds to a gait cycle.

A modified strides detection algorithm was developed and includes: i) calculation of the mean distance between the identified heel strikes of a YY Euler angle signal; ii) iteration through the heel strikes and extraction of the signals segments from the original signal contained in-between; iii) storage in a key-value list of the timestamp associated to the start of each segment and the segment itself; iv) elimination of all signal segments which length do not fit in the interval based on the mean segments distance: $[0.7 * \text{mean segments distance}; 1.3 * \text{mean segments distance}]$. By considering the segments of one of the two Euler angle signals, iteration through the segments and for each segment iteration through the key-value list of the other signal (other leg) that maps the start of an identified segment with a timestamp. If there is a segment timestamp that fits in the temporal window of the second signal extracted segment, then the current segment is accepted and the two segments from the different devices are considered as a gait stride, otherwise the segment is excluded.

Extremely Randomized Trees method was used for feature extraction. Following features were considered: *in time domain* - standard deviation, maximum, minimum, skew, kurtosis, displacement, interquantil range, mean absolute deviation and area; *in frequency domain* - Discrete Fast Fourier Transform (DFFT) (standard deviation of DFFT coefficients), Power Spectral Density – Welch (PSD) (standard deviation of PSD) and Discrete Wavelet Transform (Daubechies db4) (standard deviation of the approximation and detail coefficients); *in time-frequency domain*: Discrete Wavelet Transform (Daubechies db4) (standard deviation of the approximation and detail coefficients). A set of extracted features was used for training and other for testing the classification model.

4. Results and Discussions

The implemented gait assessment system was tested in laboratory with 4 volunteers [average and range age 38.8 years (33-45); average weight 78.8 Kg (67-87)] who performed normal and abnormal gait. A walking distance of nearly 40 meters for each gait was used to train the classifier, and of 10 meters to test the classification algorithm.

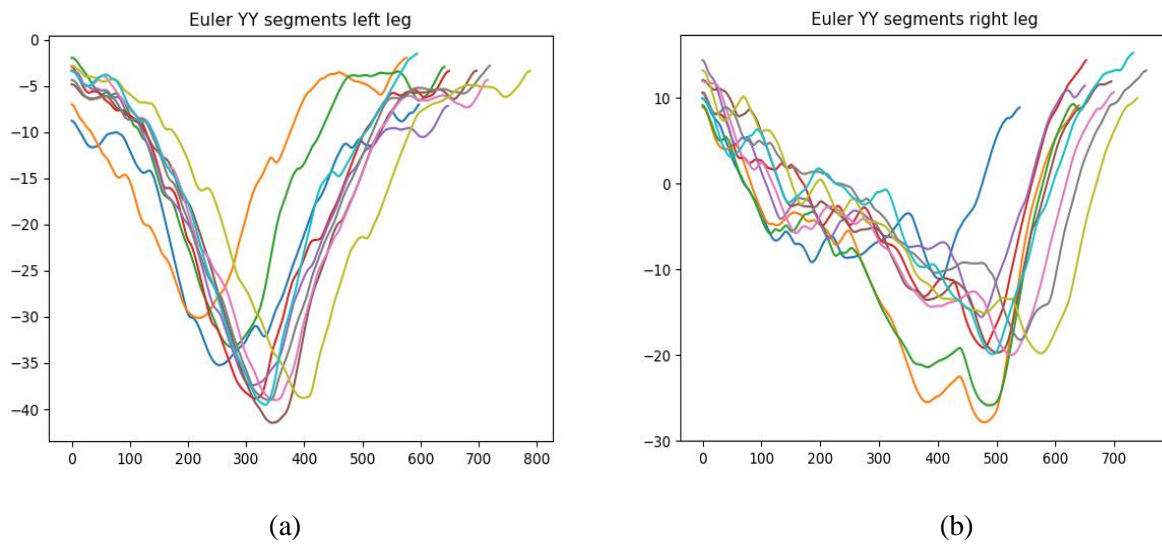


Figure 4. Euler YY angle segments for left (a) and right legs (b).

Table 1. Classifier Scores.

| Gait | Precision | Recall | F1 |
|------------------|-----------|--------|-------|
| normal | 1.000 | 0.951 | 0.974 |
| v. slow | 0.948 | 0.981 | 0.963 |
| parkinsonian | 1.000 | 1.000 | 1.000 |
| hemipl. right | 0.916 | 0.889 | 0.900 |
| equinus | 0.948 | 0.983 | 0.964 |
| glut. max. right | 0.972 | 0.935 | 0.952 |
| glut. max. left | 1.000 | 0.914 | 0.954 |
| running | 0.944 | 1.000 | 0.969 |
| hemipl. left | 0.972 | 1.000 | 0.985 |

The experiments were carried out by using 2 Shimmer3 devices ($f_s=51.2\text{Hz}$) and an Android Smartphone Alcatel U5 (Quad-core 1.3 GHz, 8GB ROM + 1 GB RAM, Android 7.0 Nougat). The type of the tested gaits and the labels for training the classifier were: normal gait (label1), very slow gait (label 2), parkinsonian gait (label3), hemiplegic gait – right leg (label 4), equinus gait-jump knee (label 5), gluteus maximus - right leg (label 6), gluteus maximus - left leg (label 7), running gait (label 8), hemiplegic gait – left leg (label 9). The classifier was trained 6 times and its average scores Precision, Recall and F1 at stride level are illustrated in table 1. The data from 4 volunteers was used for training and testing the classifier. The data from walking tests for 10 meters was used to test the capacity of the system for correctly identify the gait features. For this test results the most frequent identified stride label was considered as the identified gait type. Volunteers 2 and 4 and had 9 gait types 100 % correctly identified. Volunteer 1 had one misidentified gait: gluteus maximus - right leg as gluteus maximus - left leg and Volunteer 3 had one misidentified gait: gluteus maximus - left leg as gluteus maximus - right leg. Although the precision of gait recognition was high, only the results from algorithm testing with a larger sample size would allow the comparison with existing results from other research studies.

5. Conclusions

In this work, a simple wearable, non-intrusive affordable mobile framework that allows remote patient monitoring during gait rehabilitation is described. The remote functionality provided by the app offers

a tool for the health institutions for monitoring rehabilitation process. Heel strike and stride detection algorithms based on Euler angles and filtered signals disturbances including the alignment of IMUs signal data by means of a common temporal reference were developed and used for the classification and identification of several gait pathologies. More data from patients with pathologic and abnormal gait is intended to be included in the classifier in future work, to increase the sensitivity of the algorithm to predict the evolution of rehabilitation by considering the patient's gait session historic.

Acknowledgement

The work was supported by Fundação para a Ciência e Tecnologia, Project TailorPhy - PTDC/DTP-DES/6776/2014, Project UID/EEA/50008/2019 and Instituto de Telecomunicações.

References

- [1] Lee C-Y and Lee J-J 2002 *Int. J. of Human-friendly Welfare Robotic Systems* <http://koasas.kaist.ac.kr/handle/10203/8430>
- [2] Kong W *et al.* 2013 *Proc. IEEE Int. Conf. on Robotics and Biomimetics ROBIO (December 12-14, 2013, Shenzhen, China)*
- [3] Muro-De-La-Herran A, Garcia-Zapirain B and Mendez-Zorrilla A 2014 *Sensors (Basel, Switzerland)* **14** 2 3362-94
- [4] Nukala B T, Nakana T, Rodriguez A, Tsay J, Lopez J, Nguyen TQ, Zupancic S and Lie DYC 2016 *Biosensors (Basel, Switzerland)* **6** 4 58
- [5] Alcaraz JC, Moghaddamnia S and Peissig J 2015 *Proc. 17th IEEE Int. Conf. on Networking, Application Services HealthCom (October 14 – 17, 2015, Boston, MA, USA)*
- [6] Takeda R, Tadano S, Todoh M and Yoshinari S 2009 *Proc. 13th Int. Conf. on Biomedical Engineering CBME 2008 (December 3-6, 2008, Singapore)* (Berlin, Heidelberg: Springer) pp 1069-72
- [7] Charry E, Lai D T, Begg R K and Palaniswami M 2009 *Proc. Conf. Proc. IEEE Engineering in Medicine and Biology Society (3-6 Sept. 2009, Minneapolis, USA)* 4824, 4826-4827
- [8] Berke EC, Atasoy I, Acici K and Ogul H 2016 *Procedia Comput. Sci.* **98** 522-27
- [9] Shimmer 2017 *9DoF Calibration User Manual* https://www.shimmersensing.com/images/uploads/docs/Shimmer_9DOF_Calibration_User_Manual_rev2.10a.pdf
- [10] Shimmer 2016 *Shimmer Java/Android API User Manual* https://www.shimmersensing.com/images/uploads/docs/Shimmer_Java_Android_API_User_Manual_rev2.11.pdf
- [11] Shimmer 2017 *MSS Android User Guide* https://www.shimmersensing.com/images/uploads/docs/MSS_Android_User_Manual_v2.8.pdf
- [12] Knight C S 2008 *It's All in the Wrist: a Quantitative Characterization of Human Wrist Control* Massachusetts Institute of Technology Thesis <http://hdl.handle.net/1721.1/45623>
- [13] Vidakovic J Z, Lazarevic M P, Kurgic V M, Dancuo Z Z and Ferenc G Z 2014 *FME Transactions* **42** 3 189-199