

<b>Noname manuscript No.</b> (will be inserted by the editor)
--

---

# A smartphone-based system for detecting hand tremors in unconstrained environments

Iván García-Magariño · Carlos Medrano · Inmaculada Plaza · Bárbara Oliván

Received: date / Accepted: date

**Abstract** The detection of tremors can be crucial for the early diagnosis and proper treatment of some disorders such as Parkinson's disease. A smartphone-based application has been developed for detecting hand tremors. This application runs in background and distinguishes hand tremors from common daily activities. This application can facilitate the continuous monitoring of patients or the early detection of this symptom. The evaluation analyzes 1770 accelerometer samples with cross-validation for assessing the ability of the system for processing unknown data, obtaining a sensitivity of 95.8% and a specificity of 99.5%. It also analyzes continuous data for some volunteers for several days, which corroborated its high performance.

**Keywords** accelerometer · mobile application · pervasive healthcare · tremor detection · wearable sensors

## 1 Introduction

Tremors are one of the most common movement disorders. Tremors are defined as the rhythmic and involuntary oscillation of a body part, and are usually

---

I. García-Magariño

Department of Computer Science and Engineering of Systems, University of Zaragoza, Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain.

Tel.: +34 978645348

Fax: +34 978618104

E-mail: [ivangmg@unizar.es](mailto:ivangmg@unizar.es)

C. Medrano, I. Plaza

Department of Electronic Engineering and Communications, University of Zaragoza, Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain.

B. Oliván

Department of Psychology and Sociology, University of Zaragoza, Facultad de Ciencias Sociales y del Trabajo, c/ Violante de Hungría 23, 50009 Zaragoza, Spain.

IIS Health Research Institute of Aragón, Avda. San Juan Bosco 13, 50009 Zaragoza, Spain.

caused by muscle innervations that imply repetitive contractions [17]. From the dynamic point of view, this disorder can be classified as resting or action tremors [22]. Resting tremors occur when the body part is relaxed and supported by gravity. Action tremors occur when some muscle is voluntarily contracted, and can be classified as either postural, isometric or kinetic. The first kind happens when the body part is held against gravity, while the second one occurs when the muscle is contracted against stationary objects. Kinetic tremors occur when the muscle is contracted for actually moving the body part. The tremors can also be categorized by their causes [17,22], which can be several diseases. The most frequent causes are the Parkinson's Disease (PD) and the Essential Tremor (ET).

Due to the diseases that affect a considerable amount of the population and have the symptom of tremors, its detection is relevant in medicine. The methods for tremor detection can be categorized between conventional clinical methods and laboratory measurement methods [17]. The conventional clinical methods are based on the observation by a doctor. There are several validated scales such as the Unified PD Rating Scale (UPDRS) and Tremor Rating Scale (TRS), respectively for PD and ET. The laboratory measurement methods can be classified as accelerometry, electromyography (EMG), spiralometry, and analysis of spike trains. Accelerometry uses accelerometers attached to the body parts that suffer tremors. Generally, these tests require hospital or specialized equipment, and users may not take these tests until they are already aware of their tremors.

The goal of the current work is to detect and register tremors with smartphones in unconstrained environments. The system is transparent for users, who are not required to perform any specific movement or posture. Users only need to utilize their smartphones in their daily lives without restrictions. The implementation of this system is called *Hand Trembling detector App* (HTrembApp). This app can be installed and run as a background service permanently. It detects involuntary hand tremors, and distinguishes these from activities of daily life (ADL). The application both collects and processes the data. It notifies the user when tremors are detected. To avoid battery depletion, computation is based on simple rules. To the best of authors' knowledge, there is not any tremor detection system that allows users to perform movements without restrictions and uses smartphones at the same time.

The current tremor detector system has been evaluated with a database of tremors and ADL generated by a group of volunteers. In this manner, the current work assesses the sensitivity, the specificity and the accuracy of the detector. The ability of the detector to properly classify unknown data has been determined by means of a cross-validation strategy. It also reports the analysis of continuous data of three subjects.

The implications of this work are not only the substitution of a laboratory accelerometer. The presented application could have several utilities. One of these would be the continuous monitoring of hand tremors, which provides useful information for tracking the evolution of some diseases such as PD and ET. Another utility would be the early detection of hand tremors, so that

patients can get early treatments for these diseases. Recent studies show that early detection of PD may help in selecting the appropriate treatment [15]. Finally, this application could also be useful for monitoring the individual responses to therapies.

The remainder of this article is organized as follows. The next section introduces some related works highlighting the gap of the literature that the current work covers. Section 3 describes the design of the tremor detector system, and section 4 introduces its implementation. Section 5 presents the evaluation of the current detector with the group of volunteers. Finally, section 6 indicates the conclusions, and depicts some future lines of research.

## 2 Related work

The existing solutions for tremor detection use different kinds of inputs such as EMG, spike trains, sub-cortical data, and accelerometers, although more extensive information is provided for the latter. Section 2.1 introduces the tests that use any input different from accelerometers, which usually take place in clinics, hospitals or health centers. Section 2.2 mentions the related approaches that are based on accelerometers. Section 2.3 specifically indicates some tremor detection systems that use smartphones. Finally, section 2.4 discusses the main gap of the literature that is covered with the current approach.

### 2.1 Detection of tremors using non-accelerometer signals

To begin with, some works use EMG signals for detecting tremors. For instance, De Marchis et al. [9] presented an algorithm that is based on the calculation of a second order moment function of a moving time interval in EMG signals. It detected more tremors than the alternatives even with low signal-to-noise ratios.

Other works use sensors that are specific from the neuroscience field. For example, Kim et al. [16] introduced a method for the continuous tracking of the frequency and amplitude of tremors from spike trains. Their method applies a novel smoother that uses the Kalman filter. It can detect statistically significant tremors in a motorization with discontinuous tremors. Its main novelty was to detect tremors by means of neuroscience techniques (in particular analysis of spike trains). In addition, Camara et al. [4] presented a system that classifies different kinds of tremors within the PD disorder. The high impact of their work relies on the possibility of determining different kinds of treatment in the future for the different variations of PD regarding their classification of tremors. Their system processed sub-cortical data dealing directly with brain signals. They applied clustering techniques for performing the classification of tremors.

Some works perform detections of tremors by combining inputs from EMG signals and accelerometers data. For instance, Hossen et al. [13] presented an

algorithm for distinguishing whether certain tremors are caused by PD or ET. These two disorders have tremors that may not be distinguished by simple visual observation of a doctor, but the former disorder has greater consequences with more disabilities than the latter. There are medical tests that distinguish these diseases, but these tests are intrusive (e.g. injection of a radioactivity-labeled dopamine transporter ligand), and require expensive medical equipment and a considerable amount of time. The algorithm of Hossen et al. receives input from a light piezoelectric accelerometer (i.e. about 2 grams) and an EMG signal. Their application was trained with several patients of PD and several patients of ET. Their recordings were performed by asking patients to have a very specific posture that consisted of hands and fingers actively to a 0Å° position with the resting forearm. They achieved a success ratio of 85% in discriminating PD from ET in a different set of patients.

Some studies highlight the importance of detecting tremors not only in ad-hoc postures or movements, but also in ADL. Cole et al. [6] proposed the use of a device in each limb, which includes a measure of acceleration and surface EMG (sEMG). Normal activities were considered, such as walking, sitting and standing. Eight features were extracted from 1 s time windows, 5 from acceleration and 3 from sEMG. A subset of them fed either a Dynamical Neural Network or a Dynamical Support Vector Machine. Both systems had some trouble to detect tremor in the presence of certain movement states (e.g. walking).

Nevertheless, most of these approaches usually require that patients get to a hospital, a clinic, a health center or a similar place to perform the tests, since all these require specific medical equipment. Their tests were designed to be applied in controlled environments with the patients in a specific posture or situation. Only in [6] other movements were considered, but with a full set of sensors and with a fusion of sEMG and accelerometry.

## 2.2 Accelerometer-based detectors for tremors

Accelerometry is quite popular in tremor detection due to the ease of its application. Simple visual inspection of accelerometer traces can give some information, like in [3] where the postural tremor traces of PD patients are classified in three groups: sinusoidal, increased and aperiodic. However, the most common way to analyze signals is in frequency domain. Barroso et al. [2] calculated several parameters from the frequency spectrum: peak frequency and its amplitude, power around the peak, total power and power split into low (1-4 Hz) and high (4-7 Hz) frequency bands. The accelerometer was attached to the wrist and measurements were taken with the arm extended along the body. Except the peak frequency, all of them were able to fully discriminate between 10 normal subjects and 10 people with PD. Geman et al. [11] considered the wavelet transform to better characterize tremor. Data was recorded with the Wiimote of Nintendo in three groups of volunteers: 28 PD patients, 24 people “suspicious of PD” and 30 with normal tremor (healthy). In particular, they

observed strong similarities between the PD tremor signals and suspect PD tremor signals. However, the results depended on the type of mother wavelet and its scale.

Besides detecting tremor as a dichotomy problem, its severity can also be assessed. Hoff et al. [12] used a continuous ambulatory multichannel accelerometer to assess resting tremor in patients with PD. Duration and intensity of tremor were correlated with the clinical score for resting tremor (Spearman's rank correlation: 0.66-0.77). In [18] an automated method for both resting and action or postural tremor assessment was proposed using a set of accelerometers mounted on up to six patient's body segments. Action and posture recognition was based on a set of features that measure relative and absolute orientation, as well as the low frequency energy of the signals. Tremor severity was assessed with features in the frequency domain: dominant frequency, amplitude of the dominant frequency and the energy of the harmonic motion. Two Hidden Markov Models were used, one for action/posture recognition, and the other one for severity classification. The volunteers followed a set of directed activities (sitting, walking, finger to nose movement etc.). The system quantifies tremor severity with 87% accuracy, discriminates resting from postural tremor and tremor from other PD motor symptoms. D'Addona et al. [7] used the Wiimote for detecting tremors resulting from a previous sport exercise. Their system provided a quantitative assessment of these tremors. Their resulting measurement values were especially relevant for sports that need ability. These tremors were usually related to the fatigue. In their test, the subjects held the Wiimote for 90 seconds in a specific posture. They outperformed similar methods that used spiralometry and laser-pointing systems.

Van Someren et al. [20] proposed some test with ADL, besides some ad-hoc postures. Volunteers wore in their wrist a commercial available actigraph, the Actiwatch. Data acquisition took place during neurological examination, including walking, standing, sitting, arm rest, arm extension and arm movements. Five descriptors are calculated in a period-amplitude analysis of 2 s time windows: the maximal period length (MaxPer), the minimal period length (MinPer), the number of upward zero crossings (ZC), the peak amplitude (AMP) and the difference between the minimal and maximal period length (PerDiff). The distributions of these descriptors showed some differences between patients and control, but no single variable could discern tremor from other movement. Thus, the authors proposed a set of heuristic rules to calculate a summed likeliness, which provided an accurate classification of 71% of tremor time, and a false positive classification of 0.5% in controls. The authors suggested a possible extension to daily living in the future.

Different from most previous studies, our work targets tremor detection in unconstrained activities. Although in [20] similar ideas are presented, in the present work a wide range of activities is tested as well as some continuous data of three volunteers. In addition, the use of smartphones for continuous monitoring can reach a larger population since these devices are very popular. However, smartphones pose new challenges because there are more movements

that can be confused with or interfere the detection of tremors, like writing on the phone, and because it has the additional difficulty that it is not known whether the smartphone is actually held by the user.

### 2.3 Detection of tremors with smartphones

Since smartphones include accelerometers, some researchers have started using these devices for measuring tremors. For instance, Joundi et al. [14] interpreted the output of a general-purpose smartphone application (i.e. iSeismo) in an iPhone. They concluded that their results almost matched the precision of the clinical tests with expensive equipment. ISeismo is a general-purpose application that had been implemented by developers different from the authors of that work. The iSeismo application just shows the accelerometers variations, and the authors interpreted this graphical output. The work of Saifee et al. [19] also used an smartphone as measurement technology with successful results although the sample was reduced (only four patients). Carignan et al. [5] compared the results obtained with a smartphone with another measurement instrument based on accelerometers. Their work showed a high correlation between both outputs, and they concluded that the smartphones are appropriate for quantifying the pathological tremors. In this context, the work of Daneault et al. [8] is very complete. In their work, the smartphone application not only acquired the data, but also processed them in Blackberry phones. They found that the processing is possible, and the results were highly correlated with the software developed for PC, except for finding the frequency peak. In addition, they compared their results with a laboratory accelerometer. They found difficulties for measuring tremors with low amplitude in phones. However, if the amplitude of tremors surpassed 1 mm, all the frequency features were highly correlated between laboratory accelerometers and phones ( $r > 0.88$ ). Finally, they proved that the results with their smartphone application were highly correlated with the clinical scales, although this correlation was lower for kinetic tremors due to the difficulty of separating the information related to the voluntary movements. Therefore, smartphones can properly acquire and process data related to tremors providing useful information for the clinical evaluation.

Moreover, Arora et al. [1] presented a method for discriminating subjects with PD from the control subjects. Their work was based in a diagnosis with a smartphone application that assesses voice, posture, gait, finger tapping and response time. The subjects had to perform the five tasks concerning these features four times a day for a month. Their test also provided the modified motor proportion of the UPDRS.

Furthermore, Woods et al. [21] presented a mobile application that classifies patients with tremors between PD and ET. Their application receives input from the tremors of the subjects distinguishing the attention and distraction intervals. Their application could properly classify patients between PD and

ET, since these disorders usually imply different patterns of tremors in relation to attention and distraction periods.

However, none of the previous studies based on smartphones used them in everyday life.

## 2.4 Discussion

In most of the aforementioned works about tremor detection, the tests required that the subjects had a specific posture and/or mental state without doing any other activity. To the best of authors' knowledge, the current work is the first one that detects hand tremors distinguishing these from other movements corresponding to ADL with a smartphone. Hence, the current approach is the first one that monitors people during their daily lives with only a smartphone app for detecting tremors.

## 3 Design of the tremor detector system

The current detector has been designed to properly distinguish tremors from the common movements in ADL. Section 3.1 firstly introduces some features that characterize oscillatory movements. The current detector uses these features as parameters for the calibration and internal functioning of the detector. Section 3.2 presents the algorithm that detects hand tremors based on these parameters. Finally, section 3.3 discusses why the current approach filters the common tremor-like ADL.

### 3.1 Detector parameters

The current tremor detector has several parameters that determine which kinds of shakes will be detected. These parameters also implicitly determine which kinds of shaking or other movements are discarded. In this way, the detector can warn about hand tremors that are common in some patients of some diseases such as PD and ET. By contrast, it discards other movements such as walking with the mobile device, going upstairs, going downstairs, jogging, standing up, sitting down and the vibration of the device. These parameters follow:

- *Number of shakes*: It represents the number of shakes that must be detected before considering a movement to be a tremor. A shake is considered to take place each time the device changes its acceleration from one direction to another. Usually, each movement is counted as two of these shakes (i.e. acceleration and deceleration). This parameter allows the detector to discard movements that do not include a sequence with enough shakes such as standing up or sitting down.

- *Still acceleration*: This is the total acceleration (i.e. considering the three direction components X, Y and Z of the device) when the device is still. Ideally, it should be gravity but it can be slightly different from phone to phone due to sensor's bias. This feature is more related to the device than the movement. In order to make the current approach hardware independent, the app can automatically calibrate it. In particular, the user only needs to press a calibration button when the device is still without entering any specific value.
- *Maximum alpha ( $\alpha$ )*. This parameter establishes a threshold of maximum inclination of the device. In particular,  $\alpha$  refers to the angle between the Z axis of the device (perpendicular to the screen surface) and a vertical line. In this manner, the detector only detects hand tremors when the screen is face-up and nearly horizontal. Users commonly use the smartphone in this kind of positions. This allows the detector to discard movements such as carrying the device in the pocket with a complete different orientation of the device. This parameter also allows users to calibrate how much horizontal the device needs to be for detecting tremors.
- *Minimum and maximum intensity*: This parameter is composed of two thresholds about the intensity of a movement. The intensity is assumed to be the difference between the current acceleration and the still acceleration. This parameter discriminates whether a movement is considered a shake. The minimum intensity allows the detector to ignore slight movements different from hand trembling, such as writing on the smartphone. The maximum intensity makes the detector discard stronger movements different from hand trembling, like jogging for example.
- *Logging Intensity*: This establishes a threshold of minimum intensity of movements for reporting this intensity in the logs. This parameter does not actually influence the detections of hand trembling, but it does the logs. Low values of this parameter allow the detectors to provide very detailed logs, while high values make the detector output summarized logs easier to be tracked in the mobile device screen for example.
- *Minimum and maximum interval*: This parameter delimits the time interval from one shake to another to be considered trembling. It is composed of two thresholds. Specifically, the maximum interval discards several movements that are not related to each other. It also implicitly determines the minimum frequency of shakes. For example, some movements such as going for a walk have lower frequencies than tremors. In addition, the minimum interval discards repetitive movements with very high frequencies such as the vibration of the device.

### 3.2 Algorithm for detecting hand tremors

The application has a listener of sensor events specifically for the accelerometer changes. Figure 1 shows the dataflow diagram of the algorithm that is executed each time it receives an accelerometer event change. This diagram uses some



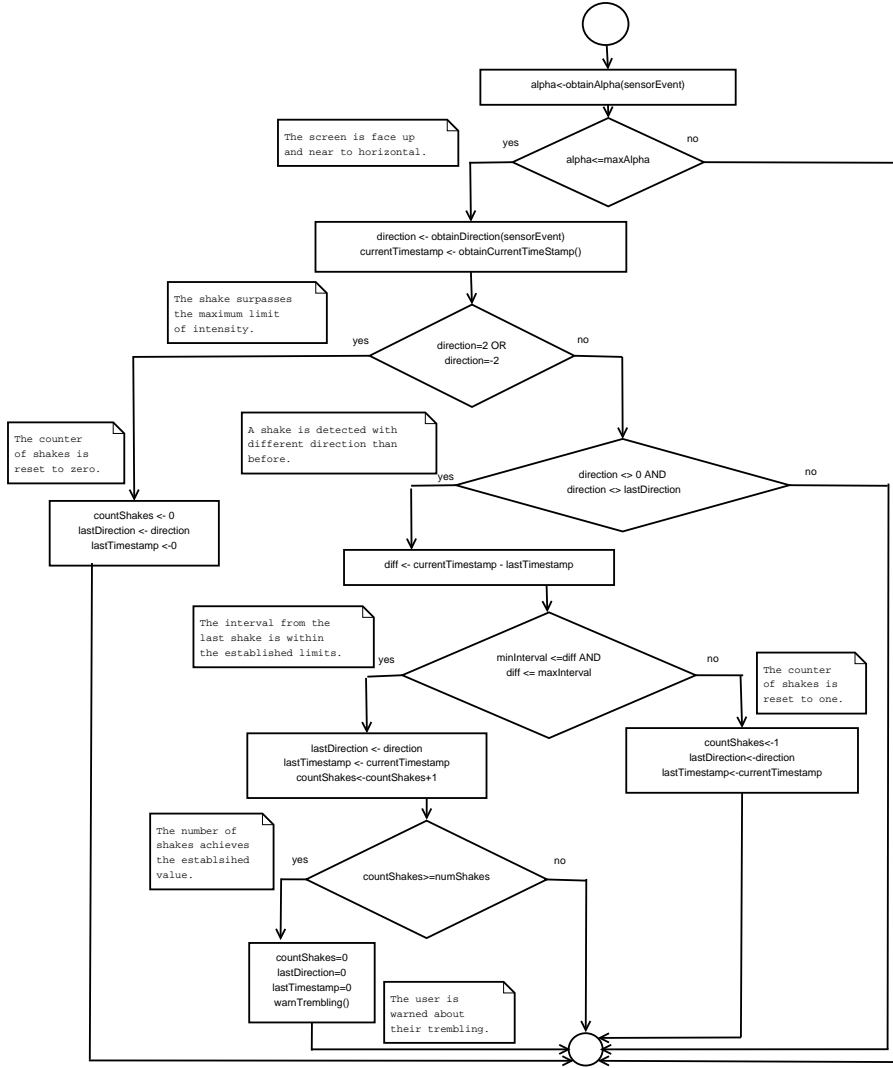


Fig. 1: Dataflow diagram that is executed when the accelerator sensor perceives a change.

of the detector parameters that were introduced in section 3.1, with properties with similar names (i.e. “numShakes”, “maxAlpha”, “minInterval” and “maxInterval”). The algorithm also uses some variable properties, which are initiated with some initial values when starting the detector. These properties may change in each execution of the method algorithm, and are conserved from one invocation of this algorithm to the next invocation. Table 1 describes the meanings of these variable properties, and indicates their initial values when starting the detector. The algorithm has the following steps:

Table 1: Variable properties of the detector

Property	Meaning	Initial value
“countShakes”	It represents the counter of shakes within the same trembling. A tremor is detected when this counter reaches the number of shakes determined in the parameter <i>Number of shakes</i> .	0
“lastTimestamp”	It stores the timestamp of the last shake. In this manner, the detector can check that the interval from the last shake is within the minimum interval and maximum interval parameters.	0
“lastDirection”	It indicates the direction of the last shake. It represents as 1 the positive shake that had total acceleration values higher than the still acceleration parameter. It represents the negative movements with -1, when the total acceleration values are below the still acceleration parameter. It uses other values such as 0 when there was not any shake that could not be considered part of the possible current trembling.	0

1. *Check the screen is face-up*: This step guarantees that the remaining of the algorithm is only executed if the screen is face up and near to horizontal. This is performed by obtaining the  $\alpha$  angle of the device inclination, and checking that this value is less than or equal to the maximum alpha calibration parameter. The calculation of  $\alpha$  is performed with the “ObtainAlpha” method that will be introduced later in this section.
2. *Discard strong movements*: This step checks whether the intensity surpasses the maximum limit of intensity. In this case, the detector discards the movement, resets the counter of shakes to zero, and finishes the management of the current event. If the intensity does not surpass the maximum limit, the algorithm goes forward to the next step. The probation of intensities of shakes is performed by means of the result provided by the “ObtainDirection” method that will be described later in this section.
3. *Alternative directions*: This step checks (1) that the device is actually having a shake with more intensity than the minimum limit, and (2) that this shake has a different direction from the previous registered shake. The acceleration event can be raised many times per second, and consequently this algorithm is invoked several times during the same shake. This algorithm distinguishes as a different shake when the direction has actually changed from the previous shake.
4. *Check frequency*: Shake frequency is implicitly checked by measuring the interval from the current shake and the last shake, by calculating the difference of the current timestamp (obtained from the system) and the timestamp of the last shake (previously stored). This step checks whether this interval is within the minimum interval and maximum interval calibration parameters. In case it is not, the current shake is considered as the first one in a possible tremor, ignoring all the previous shakes up to the present. In case the interval is within the limits, the algorithm goes forward to the next step.
5. *Count shakes*: The algorithm counts the shakes that have been admitted in the previous steps (i.e. alternative shakes with the appropriate intensities and frequency). In the handling of each event, it increases the counter of shakes. It checks whether the counter has arrived to the number of shakes

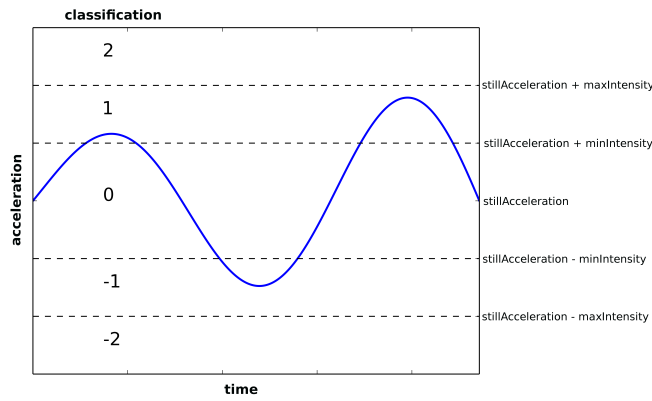


Fig. 2: Classification for obtaining the direction of a movement.

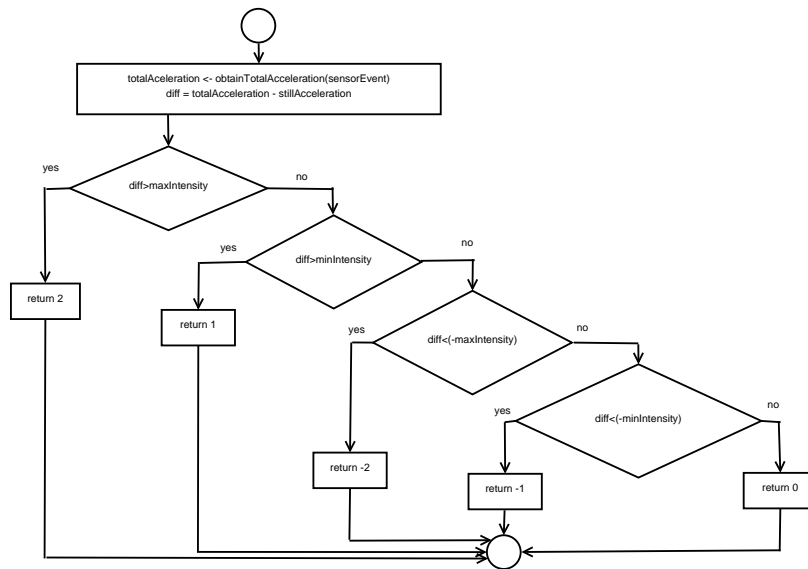


Fig. 3: Dataflow diagram for obtaining the direction of a movement.

established in the detector parameter. If so, the user is notified of its hand trembling in the mobile device, and all the variables properties are reset to their initial values for detecting future tremors.

The aforementioned algorithm (previously introduced in Figure 1) invokes the ObtainDirection method that discriminates the direction of the movement in five possible categories: excessive positive (i.e. 2), positive (i.e. 1), excessive negative (i.e. -2), negative (i.e. -1) or neutral (i.e. 0). Figure 2 graphically illustrates the categories of this classification with a repetitive acceleration

function that represents a tremor. Figure 3 shows the dataflow diagram of this classification method. This classification considers the intensity as the difference of the total acceleration from the still acceleration, denoted with the “diff” variable. This classification uses the minimum intensity and maximum intensity parameters (referred as “minIntensity” and “maxIntensity”), so that the detector only considers movements with intensities between the corresponding intervals. In particular, if the total acceleration of the device is greater than the still acceleration parameter with the intensity between the limits, the movement is considered to have a shake with positive direction. If the total acceleration is lower than the still acceleration with the intensity within the limits, the direction is classified as negative. If the movement has a total acceleration near the still acceleration considering the minimum intensity threshold, the direction is classified as neutral. Finally, if the difference from the still acceleration surpasses the maximum intensity parameter, this method classifies the movement as excessive positive or excessive negative regarding the sign of the difference.

As mentioned before, the direction of the movement is calculated from the total acceleration of the device. This total acceleration is calculated from the 3-axis accelerometer values (referred as “ $a_x$ ”, “ $a_y$ ” and “ $a_z$ ”), and is also known as acceleration magnitude. In particular, the total acceleration of the device is calculated as:

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

Finally, the ObtainAlpha method calculates the angle  $\alpha$  of the Z-axis of the device with a vertical line. The following formula calculates this angle in radians:

$$\alpha = \text{acos}(\min(1, a_z/a_s)) \quad (2)$$

where  $a_s$  is the still acceleration parameter. Then, it converts this angle to degrees, so that it can be compared with the maximum alpha calibration parameter that may be introduced by the user.

### 3.3 Detection of real tremors and filtering of tremor-like ADL

The tremors produce alternate peaks in the accelerometer trace. However, there are many ADL that also produce alternate peaks. Figure 4 presents the accelerometer traces for the tremors of a patient and some examples of ADL with alternate peaks.

Specifically, Figure 4a shows the example of a real tremor in a patient. This tremor was detected by the current approach. One can observe that the accelerometer magnitude has at least seven alternate peaks. These are separated with interval durations within the corresponding interval limits. The intensities of these peaks are also within the corresponding intensity limits. The intensity limits are represented with lines in this chart and in the one representing writing on the smartphone.

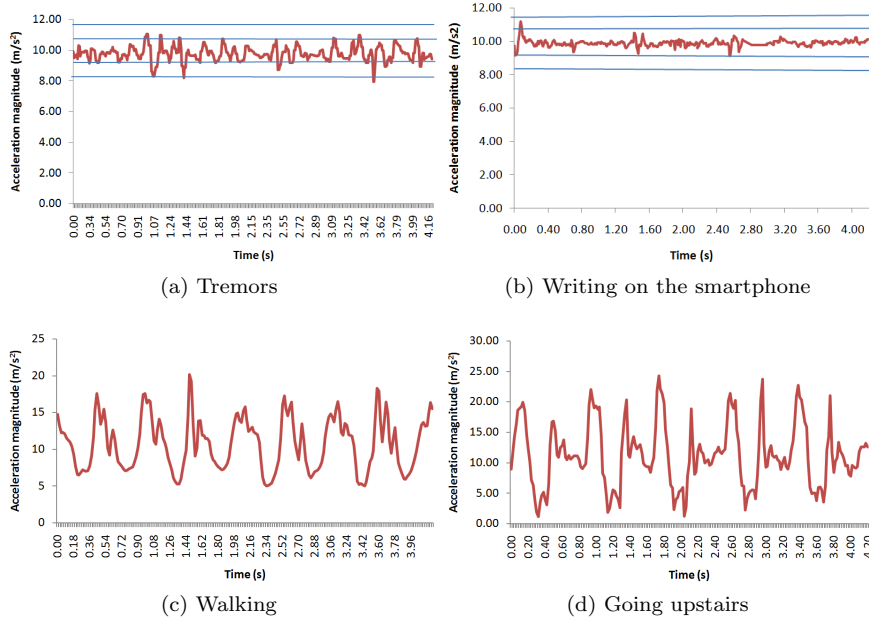


Fig. 4: Accelerometer traces of tremors and other ADL.

Figures 4b, 4c and 4d present respectively the accelerometer traces of the ADL of writing on the smartphone, walking and going upstairs. One can observe that, although all of these traces have alternate peaks, the current work distinguishes these traces from tremors by properly calibrating the detector parameters with a training set. In the case of writing on the smartphone, the peak amplitude is lower than in tremors. However, in the case of walking and going upstairs, the peak amplitude is greater than in tremors. Hence, the system can properly perform the classification by adjusting the parameters called minimum and maximum intensity. Many other ADL have been considered in the detector training as it will be indicated in section 5.

#### 4 Implementation

The current approach has been implemented with HTrembApp. This application can be executed in any mobile device (e.g. smartphones and tablets) whose operative system is Android 2.3 or above. Once the HTrembApp has been installed and started, the detector can be directly visualized as a foreground User Interface (UI), but it also runs as a background service when the user leaves the application. In this manner, users are not required to check HTrembApp and can carry on their daily routine.

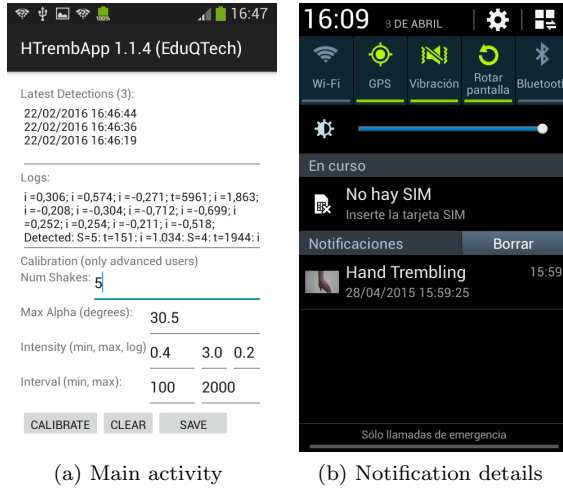


Fig. 5: UI of the app.

The main activity of the UI is shown in Figure 5a. As one can observe, it has the “Latest Detections” label with the number of recent detections between parentheses. Below, it has a scrollable list with the dates and times of the last tremor detections. In case the list is longer than the available space, the user can scroll the list. The user can also reset the list of detections to an empty list, by pressing the “Clear” button.

This UI also has a scrollable area for showing some abbreviated logs of the application. These are especially useful for calibrating the application. Below, there are several number-fields so that the user could calibrate the application by changing the detector parameters. When the user presses the “Calibrate” button, the detector starts using the new parameters. The user must press this button when the device is still, since the app automatically calibrates the still acceleration based on this assumption. This calibration can be based on performing certain real movements with the device, observing whether the application detects tremors, and observing the generated logs. The calibration process is only recommended for advanced users.

The logs report the information that is collected when the device is face-up and near to horizontal according to the corresponding detector parameter. The logs show the intensities of the detector that are greater than or equal to the logging intensity parameter. These intensities are outputted with the variable letter “i” (in  $m/s^2$ ). Each time a shake is detected, the elapsed time from the previous one is shown in milliseconds with the variable letter “t”. Each time the counter of shakes increases, this counter is printed out with the letter “S”. After a new calibration, the word “calibrated” appears alongside all the detector parameters. In the logs, the “Detected” word refers to a hand tremor detection. The logs have a reverse chronological order, so that the latest logs

are firstly shown. These logs let advanced users to know for example which parameters should be calibrated for detecting or discarding certain movements of the device.

The application also allows the user to save all the results to a text file, including the latest detections and the logs. This operation is performed by pressing the “save” button and then this information is saved in the “HTrembApp” folder in the device root using the date and time as the file name with the format “dd-mm-yyyy\_hh-mm-ss.txt”.

Each time the application detects a hand tremor, the user is notified. In case the application is running in foreground, the application changes its background color to red for one second. Otherwise, when the application is running in background, a notification of the tremor is inserted in the top notifications bar with an icon representing a hand. This notification can be consulted by the user, and it is titled “Hand Trembling”. Its message contains the time and date of the detection of the hand trembling. Figure 5b shows an example of a user consulting the details of a hand trembling notification. If another tremor is detected, this notification is updated with the date and time of the last detection. The user can click this notification, so that the application is brought to foreground with all the detections including the last one notified.

The background mode is especially useful for detecting involuntary hand tremors that might be not perceived in controlled tests. In earlier stages of some diseases such as PD, patients may voluntarily force themselves to have the hand still in a controlled test, while they have this symptom in some ADL such as consulting their smartphones or writing on it.

The application was initially tested with a Samsung Galaxy Trend Plus smartphone, which has a 1.2 GHz Dual core processor and a built-in accelerometer (range  $\pm 2g$ , sampling frequency 100 Hz).

Moreover, an off-line version of the detector was developed for performing the calibration and evaluation from a database of accelerometers records. This application receives input from text files with accelerometer and time data, and was developed as a standalone Java application.

## 5 Performance evaluation

The presented tremor detection system is evaluated by considering four possible situations evaluated in time windows: *True Positive (TP)*, the user had a tremor while holding the device and the system properly detected the tremor; *False Positive (FP)*, the system detected a tremor, but instead it was a normal ADL; *True Negative (TN)*, the system does not warn about any tremor detection in an ADL of a person without tremors; and *False Negative (FN)*, the system did not detected any tremor in a person with tremors while holding the device. The current evaluation considers time windows of three minutes.

The system reliability was measured in a similar way as other event detectors such as fall detectors. The *sensitivity* is calculated as  $TP/(TP+FN)$ , and measures the performance in detecting tremors. The *specificity* is obtained by

$TN/(TN+FP)$ , and assesses the ability of exclusively detecting tremors. The *accuracy* is expressed with the formula  $(TP+TN)/(TP+TN+FP+FN)$ , and represents the ratio of true results provided by the considered detector. In some cases, this paper also presents the percentage of FP (FP%), i.e.  $FP/(TN+FP)$ , and the percentage of FN (FN%), i.e.  $FN/(TP+FN)$ .

### 5.1 Cross-validation

In order to train the trembling detector, the current work used a database with the accelerometer traces of tremors and ADL generated by two groups of volunteers. The first group included 11 people with tremors, while the second group included 10 people without tremors. In the first group, there were five people who actually suffered from the symptom of resting and/or postural tremors. They were 58.2 years old in average, and three of them were women. The first group also had six people who simulated resting and/or postural tremors. They were 34.5 years old in average, and one of them was a woman. Finally, in the second group, none of the participants had tremors. Their average age was 31.4 years old, and three of them were women. The study protocol was approved by the Ethical Committee for Clinical Research of Aragon (CE-ICA). All subjects received oral and written information about the study, and written informed consent was obtained from them.

The database has 62 samples of people with tremors while holding the device. In these samples, they were reading and writing on the smartphone. The database also has 1708 samples of people without tremors performing ADL such as reading on the smartphone, writing on it, walking, standing up, sitting down on a hard chair, sitting down on soft surface such as a sofa, getting into a car, getting out of it, lifting the smartphone, picking up something from the floor, squatting, lying on a bed, getting up from bed, going upstairs, going downstairs, jogging, jumping, taking the smartphone out of the pocket for talking, putting it on the pocket, leaving a handbag with it on a table or lifting the handbag from the table. During some ADL, the smartphone was worn on a pocket or in a handbag. However, the smartphone was held in hand before and after each recorded activity. The number of samples of ADL is greater than the number of samples of tremors, since this work considers a high number of ADL that could be confused with tremors.

In order to assess the validity of the current system for processing unknown data, this work applied the *k-fold stratified cross-validation* method. In this method, the data is divided in  $k$  subsets in a balanced way. The detector system is calibrated and validated  $k$  times. Each time the system is calibrated with  $k-1$  subsets, and then it is validated with the remaining subset. The current experimentation has selected the value of ten for  $k$ , since the ten-fold stratified cross-validation is widely accepted as appropriate for limited sets of data.

The appropriate calibration of detector parameters requires training the system with a representative amount of data. Each parameter influences both the sensitivity and specificity of the system. In particular, Figure 6 presents the



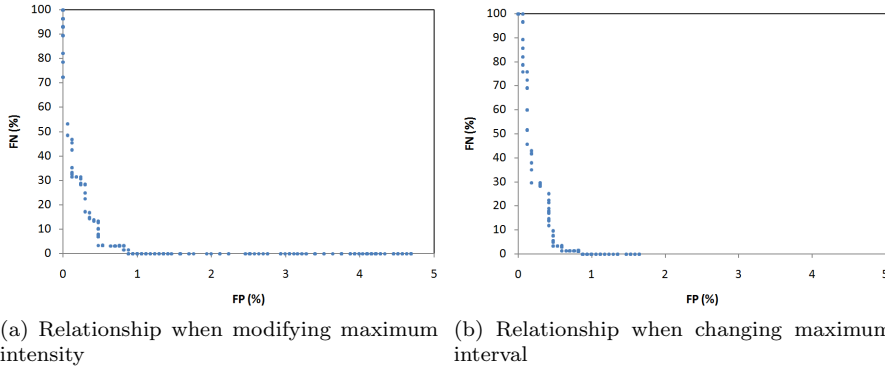


Fig. 6: The relationship between FN and FP when changing each parameter, while the other parameters are fixed to the values of Table 2.

Table 2: Parameters of the trembling detector.

Parameter	Value
Number of shakes	6.90
Maximum $\alpha$ (degrees)	30.54°
Minimum Intensity ( $m/s^2$ )	0.72
Maximum Intensity ( $m/s^2$ )	1.75
Minimum Interval (ms)	126.70
Maximum Interval (ms)	1381.20

relationship of the percentage of FN and the percentage of FP when varying the most relevant parameters. In particular, Figure 6a shows this relationship when changing the parameter of maximum intensity while the other parameters are fixed. Specifically, these are fixed to the values that will be introduced later in Table 2. Figure 6b shows this relation when altering the parameter about the maximum interval.

The detector system was calibrated with the White Box Calibration method [10] for each step of the cross-validation. Table 2 determines the average values of the parameters of the detector after the calibrations, considering all the steps of the cross-validation.

Table 3 indicates the sensitivity, specificity and accuracy, considering the average results in the cross-validation runs. Although 4.17% of time windows of people trembling were ignored, it is worth mentioning that the system detected tremors in all the people that actually had or simulated these. For the present tremor detector system, the specificity can be considered more important than the sensitivity, because (1) users may not want to be warned unnecessarily, (2) detecting a particular tremor is not as urgent as other events such as falls for example, and (3) users with the trembling symptom periodically have tremors, and it is irrelevant that the system misses a few of these.

In the group with tremors, this work analyzed the correlation between the kind of volunteer (i.e. either having real tremors or simulating these) with the

Table 3: Sensibility, specificity and accuracy of the trembling detector.

Index	Value (%)
Sensitivity	95.83
Specificity	99.51
Accuracy	99.41

Table 4: Results considering several groups of ADL

Activity	FP(%)	FN(%)
Reading / Writing on the phone	0.00	3.39
Pick the phone up and speaking with it	0.00	
Picking / leaving a bag with the phone	0.00	
Getting to bed / getting up from it	1.57	
Getting in/out of a car	0.61	
Standing up / sitting down	0.46	
Jogging	0.79	
Jumping	0.90	
Picking something from the floor / squatting	0.00	
Going upstairs / downstairs	0.45	
Walking	0.00	

results of the detector (i.e. their numbers of FN). The Point-Biserial Correlation analysis was applied as a particular case of the Pearson's correlation analysis, obtaining a Pearson correlation of 0.043 and a significance (sig.) (2-tailed) value of 0.900. Thus, there is not any statistically significant correlation about this.

In order to analyze the possible reasons of the FP, Table 4 presents the percentages of FP distinguishing between several groups of ADL. Some groups of ADL are always properly discarded such as (1) reading/ writing on the phone, (2) answering and speaking with it, (3) picking the phone up and speaking with it, (4) picking something from the floor/ squatting, and (5) walking. Conversely, getting to bed/ getting up from it is the only group of activities that obtains FP over 1%. The reason may be that getting to a mattress with springs may produce movements similar to trembling. The remaining activities obtain FP% in the (0% - 1%) interval. These can be either related to (a) sitting in a soft surface (e.g. inside a car or in a sofa.) or (b) doing a physical exercise (e.g. jogging, jumping and going upstairs/downstairs). The former can be caused by the bouncing movements after sitting. The latter can be caused by some fatigue. In fact, it is well-know that physical exercise can cause some tremors in some cases [7]. This table only presents the percentage of FN when reading and writing of the phone, since the system is aimed at detecting tremors in these particular situations. Finally, it is worth mentioning that this work also checked that the vibration of the smartphone was not detected as a tremor with a 0.00% of FP. However, this datum was omitted in the table for not strictly being an ADL.

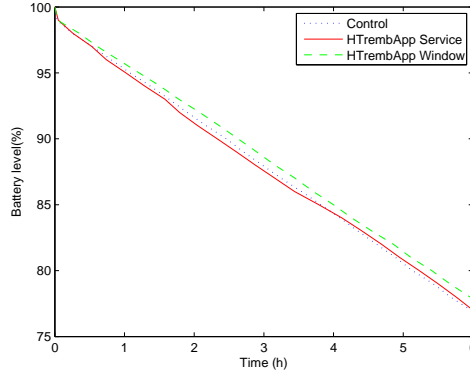


Fig. 7: Analysis of the battery consumption

## 5.2 Analysis of continuous data

Besides the experiments in a controlled environment, three volunteers collaborated in acquiring continuous data information. These volunteers were asked to follow their daily routines using their smartphones while running the presented mobile application. Two volunteers did not have tremors, while the other one simulated postural tremors. In total, the experimentation analyzes 118 h of the volunteers.

In particular the application monitored 88 h of the volunteers without tremors. This time is distributed into 40 h working in indoors jobs, 30 h being at home, 12 h outdoors, 2 h driving a car, and 4 h in a gym. In all this time, the volunteers regularly did activities such as walking, sitting down, lying, using several applications in the phone (Internet browsers, WhatsApp, email, camera, notes, alarms), activities at home (cooking meals, brushing their teeth, housekeeping), practicing some sports (running, spinning and lifting dumbbells) and other ADL.

Moreover, the application analyzed 30 h of the volunteer with tremors. This time is distributed into 8 h working in an indoors job, 18 h being at home, 3 h being outdoors and one hour driving a car. In all this time the volunteer did regular activities, very similar as in the previous case except that no sport activities were done.

The applications did not detect any tremor in all the time of the volunteers without tremors. Thus, the application worked properly, obtaining the maximum specificity value in this continuous data experiment. In addition, the application detected 34 tremors in the volunteer that actually had these. Thus, the application also worked properly for warning a person with tremors about these, and consequently the application also has a high sensitivity.

Finally, this work performs a brief analysis of the battery consumption of the application. Figure 7 shows the consumption of the smartphone battery in three experiments that only differ in the state of the presented application. In

the control experiment, HTrembApp was not installed. In the second experiment, HTrembApp was installed and run as a background service, completely transparent to the user. In the last experiment, HTrembApp was running with an open window. The remaining conditions were similar. In all the experiments, the smartphone was connected to Internet with WiFi, and had the same apps running: a web browser, a file browser, and the BatteryLog app. All the logs of the battery were recorded with this last app. As one can observe in Figure 7, the battery consumption was similar in the three experiments. Hence, according to these experiments, the repercussion of the HTrembApp execution on battery consumption is not notorious.

## 6 Conclusions and future work

This paper presents a smartphone-based application that detects tremors of their users during their ADL. To the best of the authors' knowledge, this application is the first one that distinguishes tremors from other movements during ADL with a smartphone. This allows users to detect this symptom at the earliest stages. User can even install the application before presenting the symptom and it will run as a background service without any required explicit testing action. Then, the application will notify them when they present the symptom if this happens. The early detection of this symptom can be crucial for the proper treatment of diseases such as PD and ET, as the literature reveals. Other utilities of this application can be to continuously track the evolution of certain diseases or to get feedback about the individual responses to a therapy.

The performance of the tremor detector has been evaluated with some healthy people and some people with tremors (real and simulated). The experimentation applied cross-validation with ten subsets to assess the ability of the current approach for properly classifying unknown data. The detector shows a high sensitivity for detecting tremors, and a high specificity when analyzing ADL without tremors. These results were also corroborated by analyzing continuous data of the daily routines of several volunteers.

The application is planned to be enhanced to emit reports based on the number of detections during a week (or a particular number of days). This report will include a final conclusion about whether the patient has the symptom of tremors. Users will be explicitly notified only if the result of this report is positive. This application will also be able to share these reports with a group of caregivers. The future evaluation will determine if these reports properly classify users.

Furthermore, the application will be extended to collect feedback of users about the correct and wrong classifications. The information will be used to automatically re-calibrate the detector for properly classifying similar situations. This feedback will be also automatically sent to the authors, so that the current approach can be improved taking this feedback into account.

**Acknowledgements** This work has been done in the context of the project “Soluciones tecnológicas para facilitar la práctica de mindfulness: caminando hacia mHealth” with reference TEC2013-50049-EXP and supported by the Spanish Ministry of Economy and Competitiveness. In addition, we acknowledge the “Fondo Social Europeo” and the “Departamento de Industria e Innovación del Gobierno de Aragón” for their joint support with grant number Ref-T81.

## References

1. Arora, S., Venkataraman, V., Zhan, A., Donohue, S., Biglan, K., Dorsey, E., Little, M.: Detecting and monitoring the symptoms of Parkinson’s disease using smartphones: A pilot study. *Parkinsonism & related disorders* **21**(6), 650–653 (2015)
2. Barroso, M.C., Esteves, G.P., Nunes, T.P., Silva, L.M., Faria, A.C., Melo, P.L.: A telemedicine instrument for remote evaluation of tremor: design and initial applications in fatigue and patients with Parkinson’s Disease. *BioMedical Engineering OnLine* **10**(1), 14 (2011)
3. Boika, A.: Postural tremor in Parkinson’s disease (PD): Types of accelerometer traces. In: XVII WFN Congress on Parkinson’s Disease and Related Disorders, *Parkinsonism & Related Disorders*, vol. 13, p. S69. Elsevier (2007)
4. Camara, C., Isasi, P., Warwick, K., Ruiz, V., Aziz, T., Stein, J., Bakštein, E.: Resting tremor classification and detection in Parkinson’s disease patients. *Biomedical Signal Processing and Control* **16**, 88–97 (2015)
5. Carignan, B., Daneault, J., Codere, C., Sadikot, A., Duval, C.: Identifying different pathological tremor characteristics with a smart phone. In: *Movement Disorders*, vol. 27, pp. S407–S408. Wiley-Blackwell 111 River ST, Hoboken 07030-5774, NJ USA (2012)
6. Cole, B.T., Ozdemir, P., Nawab, S.H.: Dynamic svm detection of tremor and dyskinesia during unscripted and unconstrained activities. In: *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 4927–4930. IEEE (2012)
7. D’Addona, V., Evangelista, M., Viggiano, D.: A new method for quantitative tremor assessment in sports. *Sport-Orthopädie-Sport-Traumatologie-Sports Orthopaedics and Traumatology* **30**(1), 54–59 (2014)
8. Daneault, J.F., Carignan, B., Codère, C.É., Sadikot, A.F., Duval, C.: Using a smart phone as a standalone platform for detection and monitoring of pathological tremors. *Front Hum Neurosci* **6**(357), 1–12 (2013)
9. De Marchis, C., Schmid, M., Conforto, S.: An optimized method for tremor detection and temporal tracking through repeated second order moment calculations on the surface EMG signal. *Medical engineering & physics* **34**(9), 1268–1277 (2012)
10. Fehler, M., Klügl, F., Puppe, F.: Techniques for analysis and calibration of multi-agent simulations. In: *Engineering Societies in the Agents World V, Lecture Notes in Computer Science*, vol. 3451, pp. 305–321. Springer (2005)
11. Geman, O., Zamfir, C.: Using wavelet for early detection of pathological tremor. In: *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 1723–1727. IEEE (2012)
12. Hoff, J.I., Wagemans, E.A., van Hilten, B.J.: Ambulatory objective assessment of tremor in Parkinson’s disease. *Clinical neuropharmacology* **24**(5), 280–283 (2001)
13. Hossen, A., Muthuraman, M., Raethjen, J., Deuschl, G., Heute, U.: Discrimination of Parkinsonian tremor from essential tremor by implementation of a wavelet-based soft-decision technique on EMG and accelerometer signals. *Biomedical Signal Processing and Control* **5**(3), 181–188 (2010)
14. Joundi, R.A., Brittain, J.S., Jenkinson, N., Green, A.L., Aziz, T.: Rapid tremor frequency assessment with the iphone accelerometer. *Parkinsonism & related disorders* **17**(4), 288–290 (2011)
15. Khatib, S., Finberg, J., Artoul, F., Lavner, Y., Mahmood, S., Tisch, U., Haick, H., Aluf, Y., Vaya, J.: Analysis of volatile organic compounds in rats with dopaminergic lesion: Possible application for early detection of Parkinson’s disease. *Neurochemistry international* **76**, 82–90 (2014)

16. Kim, S., McNames, J.: Detecting and tracking tremor in spike trains using the rectangular model based extended kalman smoother. *Journal of neuroscience methods* **188**(1), 97–104 (2010)
17. Mansur, P.H.G., Cury, L.K.P., Andrade, A.O., Pereira, A.A., Miotto, G.A.A., Soares, A.B., Naves, E.L.: A review on techniques for tremor recording and quantification. *Critical Reviews<sup>TM</sup> in Biomedical Engineering* **35**(5) (2007)
18. Rigas, G., Tzallas, A.T., Tsipouras, M.G., Bougia, P., Tripoliti, E.E., Baga, D., Fotiadis, D.I., Tsouli, S.G., Konitsiotis, S.: Assessment of tremor activity in the Parkinson's disease using a set of wearable sensors. *Information Technology in Biomedicine, IEEE Transactions on* **16**(3), 478–487 (2012)
19. Saifee, T., Kassavetis, P., Drougkas, L., Roussos, G., Pareés, I., Schwingenschuh, P., Katschnig, P., Bhatia, K., Rothwell, J., Edwards, M.: Diagnosis of psychogenic tremor using a smartphone. In: *Movement Disorders*, vol. 27, pp. S419–S420. Wiley-Blackwell 111 River ST, Hoboken 07030-5774, NJ USA (2012)
20. Van Someren, E.J., Pticek, M.D., Speelman, J.D., Schuurman, P.R., Esselink, R., Swaab, D.F.: New actigraph for long-term tremor recording. *Movement Disorders* **21**(8), 1136–1143 (2006)
21. Woods, A.M., Nowostawski, M., Franz, E.A., Purvis, M.: Parkinson's disease and essential tremor classification on mobile device. *Pervasive and Mobile Computing* **13**, 1–12 (2014)
22. Wyne, K.T.: A comprehensive review of tremor. *Journal of the American Academy of Physician Assistants* **18**(12), 43–50 (2005)