Hindawi Publishing Corporation
Journal of Sensors
Volume 2013, Article ID 254629, 11 pages
http://dx.doi.org/10.1155/2013/254629

Research Article

Supervised Expert System for Wearable MEMS Accelerometer-Based Fall Detector

Gabriele Rescio, Alessandro Leone, and Pietro Siciliano

Institute for Microelectronics and Microsystems, Italian National Research Council (CNR), Via Monteroni, c/o Campus Università del Salento, Palazzina A3, 73100 Lecce, Italy

Correspondence should be addressed to Gabriele Rescio; gabriele.rescio@le.imm.cnr.it

Received 8 February 2013; Revised 10 June 2013; Accepted 11 June 2013

Academic Editor: Andrea Cusano

Falling is one of the main causes of trauma, disability, and death among older people. Inertial sensors-based devices are able to detect falls in controlled environments. Often this kind of solution presents poor performances in real conditions. The aim of this work is the development of a computationally low-cost algorithm for feature extraction and the implementation of a machine-learning scheme for people fall detection, by using a triaxial MEMS wearable wireless accelerometer. The proposed approach allows to generalize the detection of fall events in several practical conditions. It appears invariant to the age, weight, height of people, and to the relative positioning area (even in the upper part of the waist), overcoming the drawbacks of well-known threshold-based approaches in which several parameters need to be manually estimated according to the specific features of the end user. In order to limit the workload, the specific study on posture analysis has been avoided, and a polynomial kernel function is used while maintaining high performances in terms of specificity and sensitivity. The supervised clustering step is achieved by implementing an one-class support vector machine classifier in a stand-alone PC.

1. Introduction

The problem of falls in the elderly has become a health care priority due to the related high social and economic costs [1]. In fact the European population aged 65 years or more, which may be in need of assistance is increasing. This trend asks care-holders institutions to employ more efficient and optimized methods in order to be able to grant the required service at lower costs. The consequences of falls in the elderly may lead to psychological trauma, physical injuries, hospitalization, and even death in the worst scenario [2–5]. The main reason that pushed for the development of the presented system is to allow noncompletely self-sufficient people (e.g., older people) to live safely in their own houses as long as possible. This is important not only for aspects of health regarding assisted people, but also for the consequent social advantages. The European community issued and funded various projects and consortia. The mission focuses on several purposes, all addressed to older people, varying from the assistance in case of need, to the prevention of dangerous or unhealthy situations. The purpose of the work described in this paper is to focus on people fall detection.

Many solutions have been proposed in the detection and prevention of falls, and some excellent review studies were presented [1, 6]. Basically, fall-detection solutions can be classified in three main classes: wearable devices, ambient devices, and camera-based devices. The first approach requires that the elderly holds some kind of devices (e.g., an assistive cane) or wears sensors like accelerometers and/or gyroscopes to detect the motion of the body. In particular, recent miniaturization and cost reduction of MEMS accelerometers and the availability of reliable wireless communication technologies enabled the realization of affordable wearable monitoring systems that can be worn by people performing their normal daily activities [7–10]. For these reasons, in the last few years, the use of portable devices in the health monitoring of chronic patients has increased considerably. However, these devices have some drawbacks: they are prone to be forgotten, worn in a wrong body position, or accidentally damaged. Regarding fall detection, with respect to vision or acoustic sensors, the accelerometer module has the advantage of not having to be set up and installed in all rooms of the "smart home," as it is required for instance for 3D video trackers or acoustic scene analyzers. On the other hand,

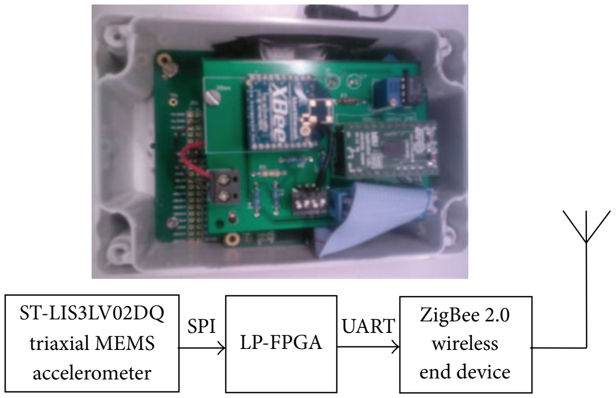


Figure 1: Picture and block diagram of the wireless accelerometer-based device.

the camera-based approach is a nonintrusive solution, since the sensor is a camera installed on a wall and is able to detect falls, switching off when the monitored person is no longer alone [11]. On the other hand, camera-based monitoring solutions suffer when large occluding objects (i.e., furniture) obstruct camera's viewing. Another significant drawback of any camera-based solution is the violation of monitored person's privacy.

In this paper a fall detector through a wearable triaxial MEMS accelerometer is presented. The proposed solution overcomes the limitation of well-known threshold-based approaches [12–17] for which the accelerometer-based device need, to be tuned in an appropriate way for each installation (i.e., the parameters setup may be different for different people). For these reasons, a machine-learning scheme [18] is used, showing high generalization capabilities in the fall detection discrimination process. The expert system uses robust features extracted taking into account important constraints and requirements of mobile solutions (workload). The extracted features are (quasi-)invariant both to specific characteristics of the mounting setup (device on chest, on waist, and on abdomen) and specific characteristics of the end users in terms of age, weight, height, and gender.

2. Materials and Methods

2.1. Triaxial Accelerometer Sensor System. The hardware used is a wearable device composed by commercial discrete circuits, according to the design proposed in [19]. The picture and the logical block diagram are shown in Figure 1. The system integrates an ST LIS3LV02DL tri-axial MEMS accelerometer with digital output, an FPGA for computing functionalities, and a ZigBee module for wireless communication up to 30 m. The power consumption is about 190 mW in streaming mode and 9 mW in idle.

The wearable device can operate in streaming (raw data are sent via ZigBee to an external computing platform for data analysis with a 10 Hz frequency) or in standalone modality, by running the threshold-based fall detection implementation on the on-board FPGA (the power consumption is limited since the ZigBee module is activated just when a fall event occurs). The LIS3LV02DL MEMS accelerometer is DC coupled, and it responds up to 0 Hz, with 16-bits resolution and a full scale in the range ±$2g$. Data can be transmitted in hexadecimal format. The sensor measures both static and dynamic accelerations along the 3 axes and allows one to receive information on the 3D spatial relative position (compared to the Earth gravity vector) of the person who wears it.

In stationary conditions, assuming a particular axis, the component of the acceleration (amplitude A, rif. (1)) is defined according to the value of the sine of the angle α between the considered axis and the horizontal plane, which is perpendicular to the Earth gravity component (g):

$$A = g \sin(\alpha). \tag{1}$$

In this way, if the accelerometer relative orientation is known, the resulting data can be used to determinate the angle of the user posture respect to the vertical direction.

2.2. Simulated Falls and ADL Tasks. In order to analyse the waveforms along each axis in the presence of falls and other kind of events (Activities of Daily Living, ADLs), a data collection has been defined in controlled (simulated) conditions by involving 11 healthy male and 2 healthy female volunteers. The simulated falls were performed by using a crash mat (height 2 cm) and knee/elbow pad protectors, meeting safety and ethical requirements. The range of actors age was 39.3 ± 12.3 years, weighing 73.7 ± 13.4 kg, and a height of 1.76 ± 0.1 m. 450 actions were simulated in which 250 were falls compliant to the specifications proposed by Noury et al. [20]. The following falls have been simulated for study:

(1) backward falls ending in the lying position,

(2) backward falls with recovery,

(3) forward falls ending in lying flat,

(4) forward falls with recovery,

(5) lateral falls.

During the data collection the wearable device was placed with an elastic band in a different positioning area on the upper part of the torso (on the chest, on the waist, and on the abdomen). Some ADLs were simulated in order to evaluate the ability of fall-detection algorithms to discriminate falls from ADLs. The simulated ADL tasks belong to the following categories:

(1) walking,

(2) sitting down on a chair and then standing up,

(3) lying down on a mat (height 33 cm) and then standing up,

(4) lying down on a mat (height 2 cm) and then standing up,

(5) kneeling on a mat (height 2 cm) and then standing up.

Each actor performed more than 15 simulated ADLs for a total of 205.

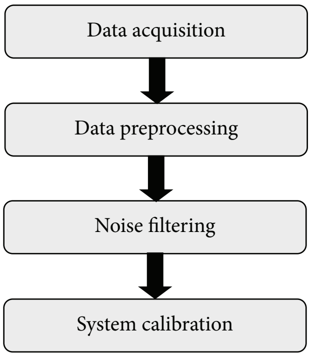


Figure 2: Logical framework overview.

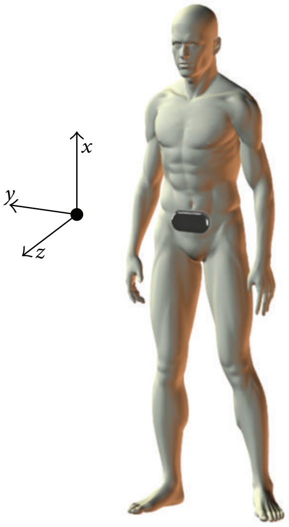


Figure 3: Mounting position and calibration.

Table 1: Acceleration values along the three axes during the initial position of Figure 3.

	A_{x0}^{ref}	A_{y0}^{ref}	A_{z0}^{ref}
Initial position	−1	0	0

2.3. Preprocessing and System Calibration. The first four computational steps (Figure 2) deal with data acquisition, data preprocessing, noise filtering, and system calibration.

The acceleration data on three axes (A_x, A_y, and A_z) is read out from the device worn by a user during the data collection. Data was stored into a portable computer and converted into gravitational units to represent acceleration data in the range $\pm 2g$, in order to make it possible to extract the angle α (described previously) and for not having orders of magnitude too different in the features (during the training of the classifier used to detect the fall events). The samples coming from the device are filtered out by a low pass 8 order, 8 Hz cut-off FIR (Finite Impulse Response) filter to reduce the noise due to electronic components, environment, and human tremor.

In order to correctly handle preprocessed data, a calibration procedure was accomplished by recovering the initial conditions after the device mounting. During this step the correct placing of hardware is verified by checking if two acceleration axes are orthogonal to the Earth gravity g (Figure 3): the acceleration values measured on the two orthogonal components must be close to zero.

The calibration procedure is composed by the following steps:

(i) the user wears the device in a standing position for 10 seconds;

(ii) the calibration routine calculates the average of the acceleration on each axis over this period. These are the initial acceleration values A_{x0}, A_{y0}, and A_{z0};

(iii) if A_{x0}, A_{y0}, and A_{z0} will be close to those expected, they will be considered as references in the fall detector algorithm, otherwise a routine to compensate the sensed misplacement will be enabled.

With respect to Figure 3, the acceleration along x-component will be close to −1 and almost zero for the others according to (1). The expected values of acceleration on three axes A_{x0}, A_{y0}, and A_{z0} are also reported in Table 1.

In practice, it is difficult to locate the device exactly in the right position, and the measured acceleration values will differ slightly from those expected. The calibration procedure is concluded if the following conditions are satisfied at the same time (and the values A_{x0}, A_{y0}, and A_{z0} will be recorded and used as references in the feature extraction phase):

$$\left|A_{x0} - A_{x0}^{\text{ref}}\right| < 0.3,$$
$$\left|A_{y0} - A_{y0}^{\text{ref}}\right| < 0.3, \tag{2}$$
$$\left|A_{z0} - A_{z0}^{\text{ref}}\right| < 0.3.$$

Since the values (A_{x0}^{ref}, A_{y0}^{ref}, and A_{z0}^{ref}) are (−1, 0, and 0);

$$\left|A_{x0} + 1\right| < 0.3, \qquad \left|A_{y0}\right| < 0.3, \qquad \left|A_{z0}\right| < 0.3. \tag{3}$$

Otherwise a routine to compensate the sensed misplacement is enabled, and the angles displacements of the sensor axes (αA_{x0}, αA_{y0}, and αA_{z0}) are calculated using the following trigonometric equation:

$$\alpha A_{x0} = \arctan\left(\frac{A_{x0}}{\sqrt{\left(A_{y0}\right)^2 + \left(A_{z0}\right)^2}}\right),$$
$$\alpha A_{y0} = \arctan\left(\frac{A_{y0}}{\sqrt{\left(A_{x0}\right)^2 + \left(A_{z0}\right)^2}}\right), \tag{4}$$
$$\alpha A_{z0} = \arctan\left(\frac{A_{z0}}{\sqrt{\left(A_{x0}\right)^2 + \left(A_{y0}\right)^2}}\right).$$

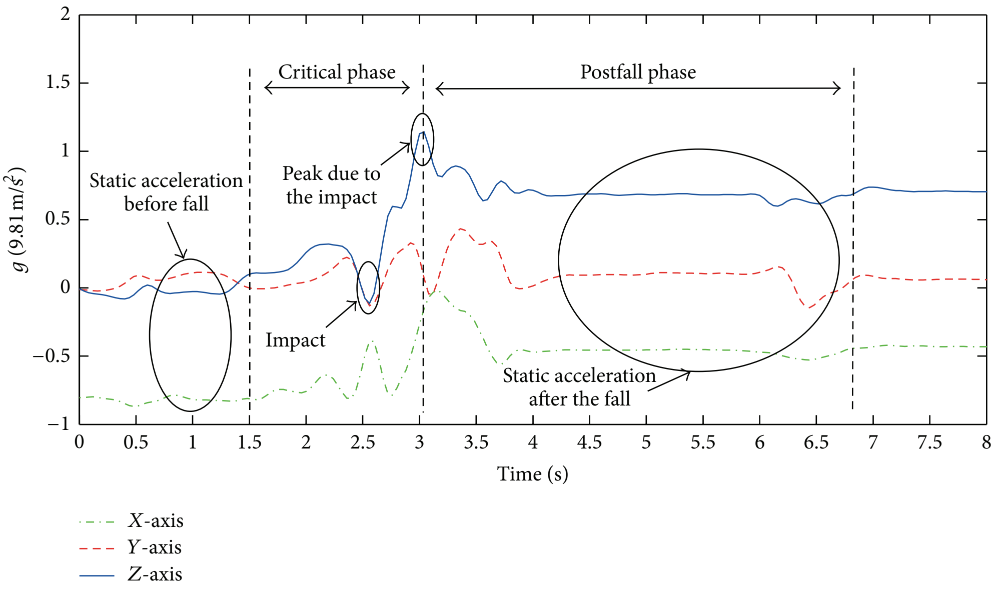


Figure 4: Typical acceleration waveform along each axis for a forward fall.

These values are stored and will be used for the correction of the misplacement during the feature extraction phase, described below.

2.4. Feature Extraction. Robust features are extracted in the time domain in a 5 sec sliding window by considering both quick and relevant acceleration changing along each axis (due to the fall) and by the change in position registered after the fall. The aim is to produce robust features taking into account all the information for the distinction of falls from other events. It is also important that such features have a low dependence on both the position of the sensor (whether it is placed on the waist, on the chest, or on the abdomen) and the human body characteristics of the user. Moreover, the computational cost must be limited for integration on embedded computing solutions.

The waveform in Figure 4 represents a typical acceleration signal of a forward fall on three axes and all phases are indicated according to the taxonomy proposed in [16]. For the features extraction process, both critical and postfall phases are of interest. In the former, the shock is measured due to the impact toward the ground, and a dynamic acceleration changing is registered. In the latter (the body is already lying on the floor) the static acceleration value records a great change due to the new position of the individual with respect to the calibration phase.

The 5 sec sliding window considered for features extraction is split in three parts as follow:

(1) from 0 sec to 0.5 sec: the maximum and minimum values of the acceleration are found, and the amplitude dynamics ΔA is calculated. It is proportional to the shock strength;

(2) from 0.5 sec to 2 sec: the system is in a transitory regime (samples are discarded);

(3) from 2 sec to 5 sec: the static-averaged acceleration value $\overline{A}$ is calculated to evaluate the new user position. Acceleration data is averaged to filter out tremors and the little movements of the end user.

During the system calibration phase it is verified that the device is worn correctly (see previous section), and the position changing is calculated as the difference between the value of the 3D-static acceleration after the fall (in the third part of the sliding window) and the one stored in the calibration phase. This difference, called Changing Position Offset (CPO), has a value in the range of 0 to 2, and it is proportional to the user displacement. In this way, for feature extraction, a study of posture was not made: only the relative varying posture analysis was considered causing a computational cost reduction and improving the robustness of the setup. The feature vector is made up of three parameters (one for each axis), coming from the modulation of the CPO and the dynamic acceleration peak, due to the impact of the fall.

If the routine to compensate the device misplacement is enabled, for the CPO calculation, the axes angles of the sensor in the initial condition need to be taken into account (αA_{x0}, αA_{y0}, and αA_{z0}, stored during the calibration phase). Using (1), the CPO value for x-axis is obtained as

$$\mathrm{CPO}_x = \left|\sin\left(\alpha\overline{A}_x - \alpha A_{x0}\right)\right|, \quad (5)$$

where $\alpha\overline{A}_x$ is the x-axis angle of the sensor in the third part of the sliding window; it is calculated considering the static averaged acceleration $\overline{A}_x$ and the same trigonometric equation used in (4) as follows:

$$\alpha\overline{A}_x = \arctan\left(\frac{\overline{A}_x}{\sqrt{\left(\overline{A}_y\right)^2 + \left(\overline{A}_z\right)^2}}\right). \quad (6)$$

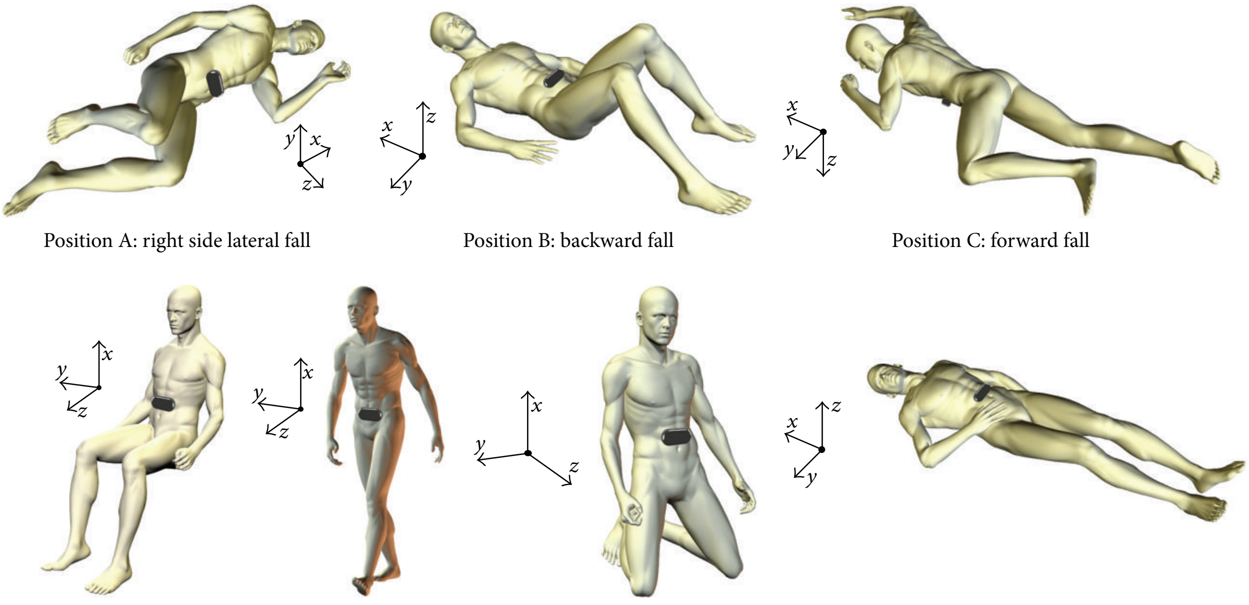
```
Let $A_0$ initial acceleration value measured after the calibration phase
Let $\alpha A_0$ initial axis angle of the sensor measured after the calibration phase

$\Delta A$ → shock dynamics due to the impact calculated in the first part of the sliding window
$\overline{A}$ → Static averaged acceleration in the third part of the sliding window

If compensation device misplacement procedure is enabled
    $\alpha \overline{A}$ is calculated → axis angle of the sensor in the third part of the sliding window
    $\mathrm{CPO} = \left|\sin\left(\alpha\overline{A} - \alpha A_0\right)\right|$ → Changing Position Offset
else
    $\mathrm{CPO} = \left|\overline{A} - A_0\right|$ → Changing Position Offset
End if

Let $\Delta A_x, \Delta A_y, \Delta A_z, \mathrm{CPO}_x, \mathrm{CPO}_y, \mathrm{CPO}_z$, $\Delta A$ and CPO calculated for each axis
$[X, Y, Z] = [\Delta A_x \cdot \mathrm{CPO}_x, \Delta A_y \cdot \mathrm{CPO}_y, \Delta A_z \cdot \mathrm{CPO}_z]$ → feature vector input for SVM
```

Pseudocode 1: Feature extraction pseudocode.



Position A: right side lateral fall

Position B: backward fall

Position C: forward fall

Position D: sitting

Position E: walking

Position F: kneeling

Position G: lying

Figure 5: Example of different spatial configurations of the device during falls and ADLs.

The same procedure must be done for the two other axes.

Thanks to this routine, the change in the angle position of the user along the three axes is measured. In this way the information coming from the modulation of the CPO remains highly selective for fall detection recognition even if the device is placed incorrectly and the features remain nearly the same. In this way the system could work efficiently reducing the problems of device positioning at the expense of an increase in the computational costs.

The process of feature extraction, described previous, is summarized in the pseudocode shown in Pseudocode 1.

For the feature extraction, It makes sense to consider the acceleration signal on each axis singularly, because a fall event leads to a change in the value of the static acceleration in at least two of the three acceleration axes (due to the orientation change of sensing axes). This is evident comparing the positions A, B, and C in Figure 5 (postfall phase of the lateral, backward, and forward falls, resp., are depicted) with respect to the initial position (see Figure 3). On the other hand, when a sitting event (D in Figure 5) occurs, the change in static acceleration can be neglected with respect to that of a fall; it is possible to do the same considerations for kneeling and walking (E and F). Instead, the axes orientation changes for a lying event (position G), but the acceleration peak produced is slower and lower than a fall, and through the product between the value of the acceleration peak and CPO it is also possible to discriminate this ADL.

Table 2 shows the nominal values of acceleration along the three axes A_x, A_y, and A_z and the value of CPO corresponding to the positions A–G of Figure 5.

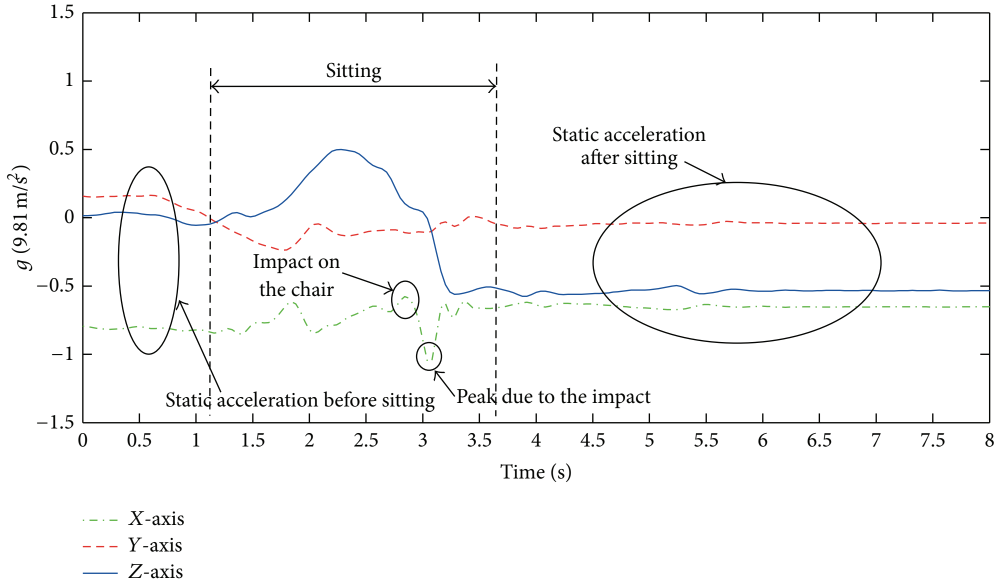


Figure 6: Example acceleration waveforms for a sitting event.

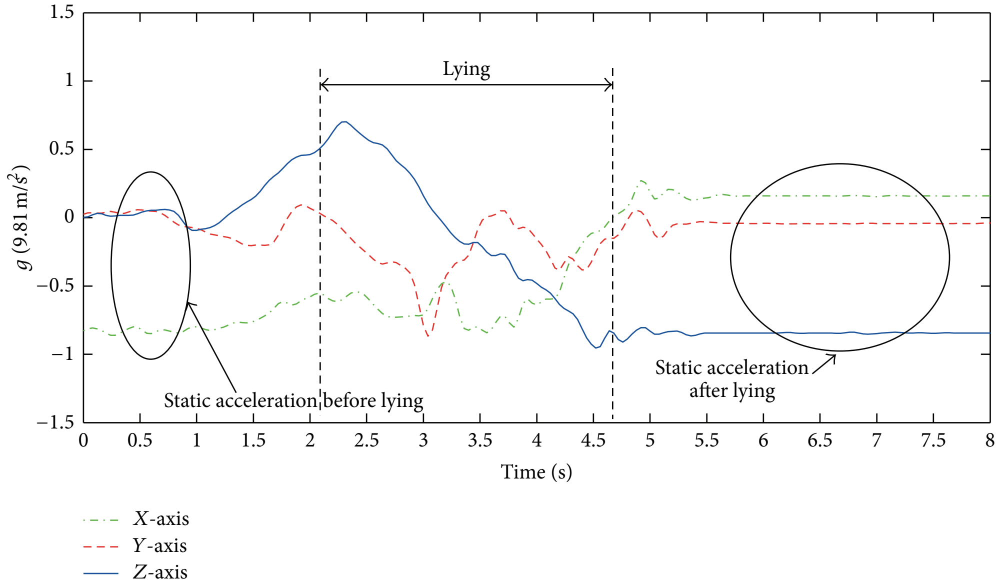


Figure 7: Example acceleration waveforms for a lying event.

Table 2: CPO coefficients by varying spatial position.

	A_x	A_y	A_z	CPO
Initial condition	−1	0	0	
Position A	0	−1	0	(1, 1, 0)
Position B	0	0	−1	(1, 0, 1)
Position C	0	0	1	(1, 0, 1)
Position D	−1	0	0	(0, 0, 0)
Position E	−1	0	0	(0, 0, 0)
Position F	−1	0	0	(0, 0, 0)
Position G	0	0	−1	(1, 0, 1)

When the orientation of acceleration does not change, the values of CPO are (close to) zero, and the elements of the features also become (close to) zero. In Figures 6 and 7 the waveforms of sitting and lying events are reported to make a comparison with that of the fall (shown in Figure 4), when the device is worn on the abdomen.

The difference among the features of the fall of Figure 4 and the sitting and lying events of Figures 6 and 7 is apparent in Figures 8, 9, and 10.

It was actually verified that all the consideration made thus so far remains valid whether the device is placed on the

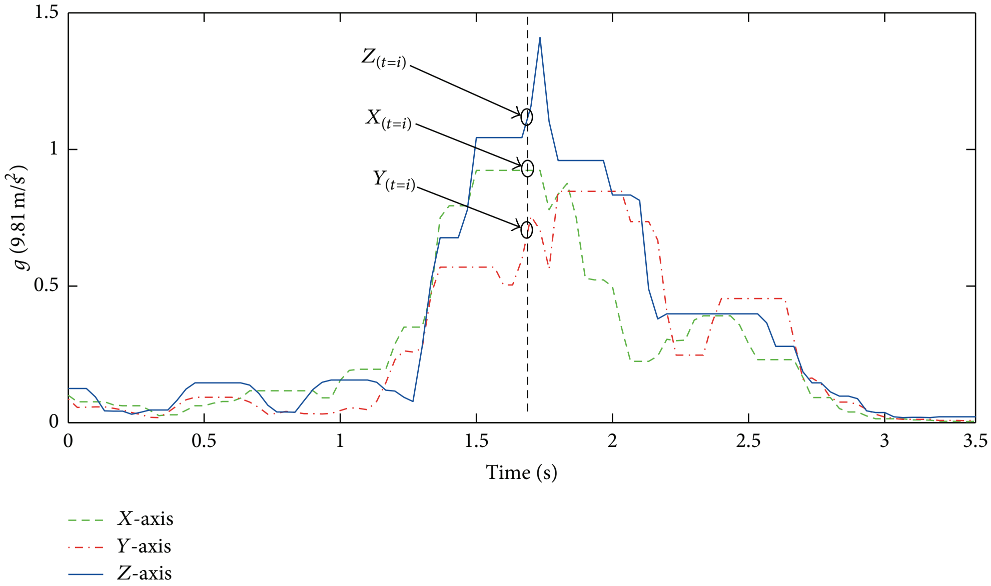


FIGURE 8: Features extracted of the forward fall in Figure 4.

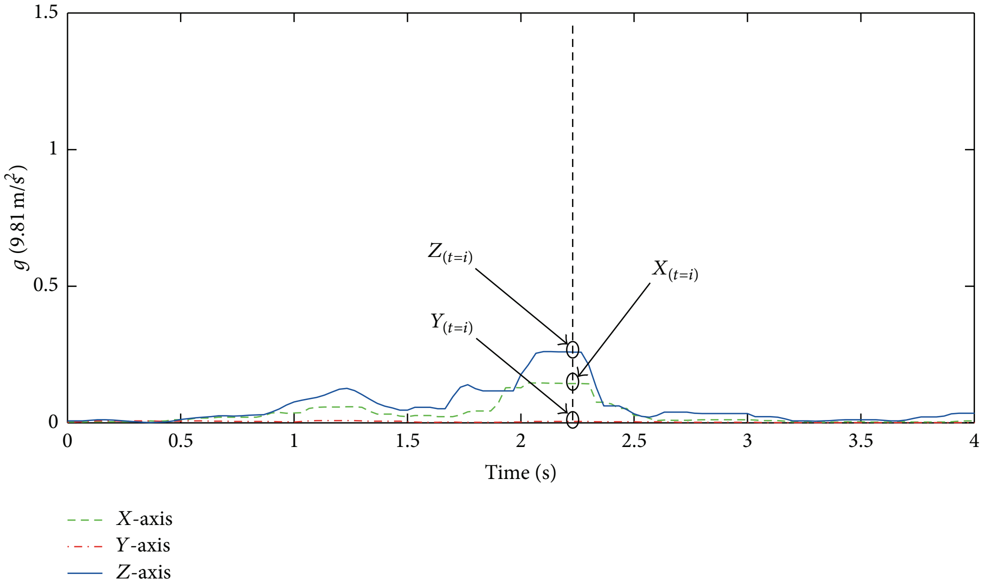


FIGURE 9: Features extracted of the sitting event in Figure 6.

waist or on the chest. In Figures 11 and 12 the features of a sequence of falls and daily events, when the device is worn on the waist and on the chest, are shown. So it is also evident that the features obtained discriminate the falls from ADLs when the device is placed in other area of the torso.

It is important to highlight that the measurements of the actions as shown in Figures 11 and 12 have been carried out by simulating critical behaviours, that is, sitting with a strong impact on the chair; bending down, lying down, and standing up quickly; moderate backward fall.

2.5. Fall Detection Algorithm. Once features are extracted, the fall events are detected by a one-class support vector machine (OC-SVM). SVM is a robust classification tool (in the presence of outliers too) with a good generalization ability. Furthermore it is less computationally intensive than other algorithms like neural networks [21].

One class SVM divides all samples into an objective field and a nonobjective field and then nonlinearly maps those sample into a higher dimensional features space with some efficient operators, called kernel function. The target is to find

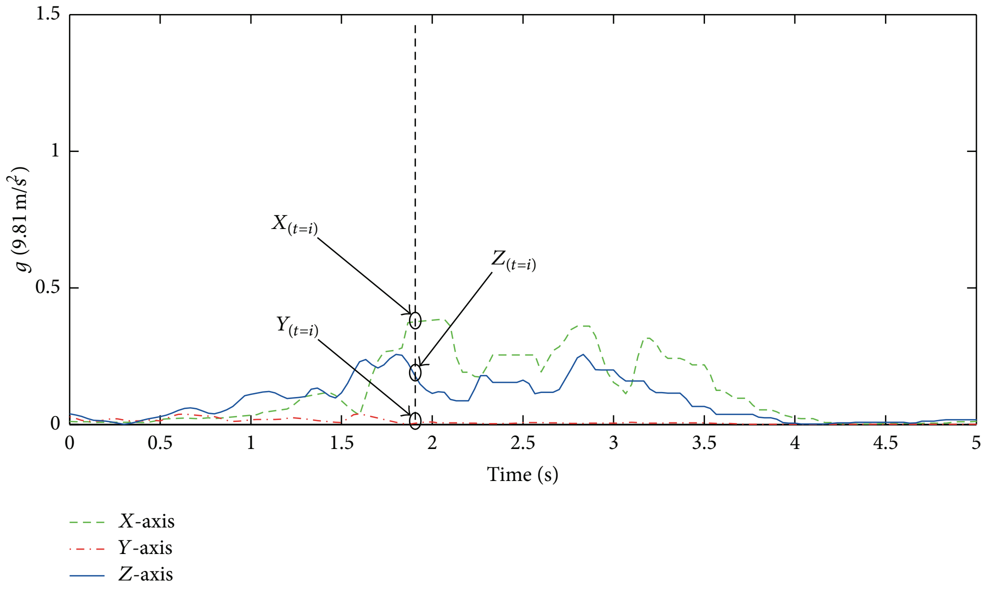


Figure 10: Features extracted of the lying event in Figure 7.

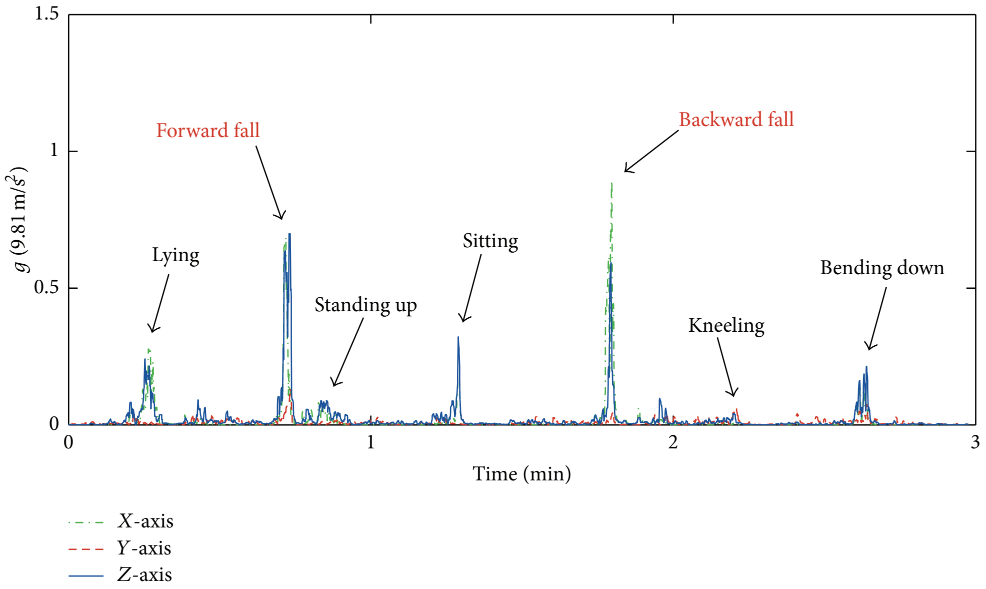


Figure 11: Features extracted of falls and ADLs simulated wearing the device on the waist.

a sphere that contains most of the normal data such that the corresponding radius R can be minimized as follows:

$$\begin{aligned} \min \quad & R^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{s.t.} \quad & \|a - v_i\|^2 \le R^2 + \xi_i, \quad \xi_i \ge 0, \end{aligned} \tag{7}$$

where a is the centre of the sphere and v_i a positive sample set. The slack variables ξ_i allow some data points to lie outside the sphere and the parameter C controls the trade-off between the volume of the sphere and the number of errors. The objective function is

$$\begin{aligned} \max \quad & \sum_{i=1}^{n}\alpha_i \langle v_i, v_i\rangle - \sum_{i,j=1}^{n}\alpha_i\alpha_j \langle v_i, v_j\rangle, \\ \text{s.t.} \quad & 0 \le \alpha_i \le C, \quad \sum_{i=1}^{n}\alpha_i = 1. \end{aligned} \tag{8}$$

The original data points are first mapped into a feature space, because some data are not spherically distributed in the input

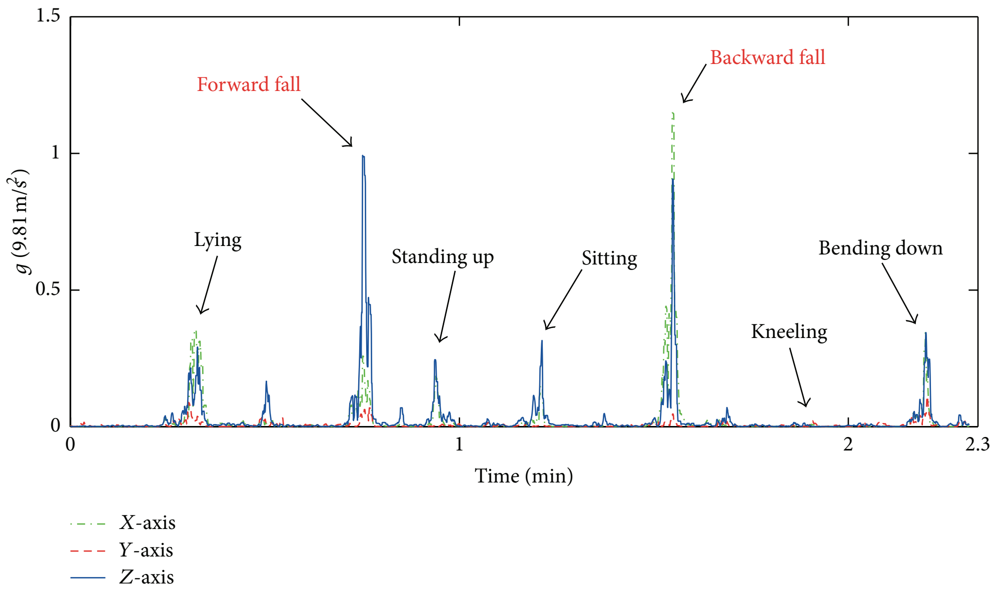


FIGURE 12: Features extracted of falls and ADLs simulated wearing the device on the chest.

TABLE 3: Types of tested kernels.

Kernel	Function
Linear	$K(v_i, v_j) = 1 + v_i^T v_j$
Polynomial	$K(v_i, v_j) = (1 + v_i^T v_j)^p$
Gaussian Radial Basis Function	$K(v_i, v_j) = \exp\left(-\gamma\left\|v_i - v_j\right\|^2\right)$
Sigmoid	$K(v_i, v_j) = \tanh\left(k v_i^T v_j - \delta\right)$

space. For this reason the kernel function $k(\cdot,\cdot)$ is used as follows:

$$\max \quad \sum_{i=1}^{n} \alpha_i k\left(v_i, v_i\right) - \sum_{i,j=1}^{n} \alpha_i \alpha_j k\left(v_i, v_j\right). \qquad (9)$$

The common types of kernel found in the literature are tested (in Table 3 the functions are shown). They were analyzed using the MATLAB tool STPRtool [22], changing the values of their parameters.

In the literature the Gaussian Radial Basis Function (GRBF) is often used in fall detectors [23–25]. For the features extracted, GRBF and polynomial kernel functions give the best results in terms of performance, even the polynomial kernel shows a lower computational cost (relative number of support vectors and relative execution time are considered). The comparison with other kernel functions will be shown in the following section.

The parameters of kernel were chosen to detect as many falls as possible, and a simple post-processing step has been implemented in order to reduce false negative events.

2.6. Postprocessing. In choosing the parameters of the tested kernel functions, the value of sensitivity was given priority. In this way some nonfall events are detected from SVM. To reduce this problem a simple filtering by voting was added: it detects an event such as a fall if the alarm coming from the SVM classifier remains high a given time t. With this approach the spikes or temporary anomalies are filtered too. The optimum time t for postprocessing is obtained from a parametric analysis. The results are shown in the next paragraph.

3. Results

This section will show a discussion about the measurements of the experimental data processed in MATLAB. To validate the implemented algorithm, the dataset previously described has been considered.

Since the output of the fall detection step is binary, there are four possible results.

(i) *True Positive (TP)*: a fall happens, and the algorithm detects it.

(ii) *False Positive (FP)*: a fall does not occur, and the algorithm reveals a fall.

(iii) *True Negative (TN)*: a daily event is performed, and the algorithm does not detect it.

(iv) *False Negative (FN)*: a fall occurs, but the algorithm does not detect it.

The performance of the system, with respect to both different kernel functions of SVM and their parameters, can be evaluated considering the following metrics:

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \qquad (10)$$

Table 4: Performance by varying the kernel functions.

	Sensitivity (%)	Specificity (%)	Relative support vectors	Relative execution time
Linear ($C = 10$)	98.6	52.3	0.51x	0.2x
Linear ($C = 100$)	97.2	46.4	0.44x	0.2x
Polynomial ($C = 20$, $P = 2$)	97.5	81.25	0.94x	0.8x
Polynomial ($C = 2.8$, $P = 3$)	97.7	95.8	1x	1x
GRBF ($C = 2.6$, $\gamma = 2$)	98.1	83.1	1.1x	1.3x
GRBF ($C = 2$, $\gamma = 3$)	97.4	95.2	1.3x	1.5x
Sigmoid ($C = 1$, $k = 1$, $\delta = 2$)	98.8	36	0.3x	4x
Sigmoid ($C = 100$, $k = 5$, $\delta = 2$)	99.1	42	0.4x	4.1x

For completeness the results obtained with postprocessing elaboration have been added. The algorithm is tested when the device is placed on the waist, abdomen, or chest.

The one class SVM classifier has been trained by using about 40 falls and 50 ADLs belonging to a large dataset in which more than 250 falls and 200 daily events were performed. The remaining 210 falls and 150 ADLs have been used for testing. Table 4 shows the specificity and the sensitivity for the kernel functions already described. The complexity of the various kernel functions can be studied by considering the relative number of vectors required and also the relative execution time [26, 27]. The kernel with the most significant values for the parameters is considered.

From Table 4 the polynomial kernel and GRBF kernel give better results. Their capacity to detect a fall is higher (more than 95% for sensitivity and specificity) than others. GRBF (with $\sigma = 3$ and $C = 2$) and the polynomial (with $P = 3$ and $C = 2.8$) provide similar values of specificity and sensitivity, but the last one works faster, and its number of vector is slightly lower, thereby it was chosen for this work. Misclassifications are for falls presenting slow dynamics or falls with partial recovery. It is important to see the result of the linear function: its workload is very low (less support vectors number and very low execution time), but it gives many false positive, and its value of the specificity is low. The function of GRBF is the most common used in the literature for the detection of falls, due to its high ability to separate the classes, even in the presence of many outliers (at the expense of the convergence time). In this work it has been also possible to obtain a high accuracy with the polynomial function thanks to features extracted: they have a high degree of intrinsic separability.

The implemented OC-SVM shows improvements in the specificity and sensitivity with respect to a threshold-based approach, and this can be verified in comparison with both the frameworks presented in [17, 28], choosing the algorithms where the same parameters of fall are used (the impact detection and posture monitoring). For the comparison of the fall detection algorithms were used the same hardware (shown in Figure 1), benchmark dataset and training/test sets described above. The values of thresholds were tuned according to the falls presented in the dataset used to train the OC-SVM scheme. The best results for threshold based algorithm were obtained with Bourke's algorithm and are compared with the proposed OC-SVM in Table 5: the latter seems more efficient, demonstrating its higher capacity to generalize, detecting a bigger number of falls with low impact magnitude than threshold-based algorithm. However, the computational cost of thresholdbased is lower.

Table 5: Comparison of the proposed OC-SVM method and threshold-based algorithm.

	Sensitivity (%)	Specificity (%)
OC-SVM	97.7	94.8
Threshold based	89.2	85.7

Table 6: Performance after filtering by voting.

Polynomial ($C = 2.8$, $P = 3$)	Sensitivity (%)	Specificity (%)
$t = 0.4$ s	97.7	94.8
$t = 0.8$ s	96.2	97.2
$t = 1.2$ s	86.4	98.1

Due to (a) the reduction in the number of the features, (b) the low computational cost of extracted features, and (c) the used kernel, the overall system workload implemented is compatible with an integration in embedded low-power solutions (DSP, FPGA, and microcontroller).

A reduction of false positive (and so an improving of specificity value), for the polynomial chosen, has been measured with filtering by voting with a temporal window of prefixed dimension (0.8 sec); see Table 6. Finally it was verified that the values of sensitivity and specificity remain almost the same even if the axes of the sensor are placed differently with respect to that shown in Figure 3 (the values of testing dataset have been changed simulating same misplacements of the device).

4. Conclusions

The proposed supervised scheme overcomes the limitation of well-known threshold-based approaches in which a heuristic choice of the parameters is accomplished. High performance in controlled conditions (events simulated) in terms of sensitivity and specificity was obtained using only the 20% of dataset for training. Performance metrics of different kernels in one class SVM are compared: best results are obtained with polynomial function and Gaussian Radial Basis Function. The polynomial kernel is used in order to limit the computational workload. A study of posture was not made, but only posture changing analysis was used, limiting the computational cost. Through the calibration step, the approach allows to generalize the detection of fall events leading invariance to physical characteristic of the end users. Future work will be devoted to validate the solution in real conditions, test the methodology with a large set of different MEMS accelerometers, and port the framework on embedded mobile solutions as FPGA or DSP.

References

[1] N. Noury, P. Rumeau, A. K. Bourke, G. ÓLaighin, and J. E. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM*, vol. 29, no. 6, pp. 340–349, 2008.

[2] M. C. Chung, K. J. McKee, C. Austin et al., "Posttraumatic stress disorder in older people after a fall," *International Journal of Geriatric Psychiatry*, vol. 24, no. 9, pp. 955–964, 2009.

[3] S. Sadigh, A. Reimers, R. Andersson, and L. Laflamme, "Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a Swedish municipality," *Journal of Community Health*, vol. 29, no. 2, pp. 129–140, 2004.

[4] A. Shumway-Cook, A. M. Ciol, J. Hoffman, J. B. Dudgeon, K. Yorkston, and L. Chan, "Falls in the medicare population: incidence, associated factors, and on health Care," *Physical Therapy*, vol. 89, no. 4, pp. 324–332, 2009.

[5] S. Elliott, J. Painter, and S. Hudson, "Living alone and fall risk factors in community-dwelling middle age and older adults," *Journal of Community Health*, vol. 34, no. 4, pp. 301–310, 2009.

[6] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *Proceedings of the 10th IEEE International Conference on E-Health Networking, Applications and Service (HEALTHCOM '08)*, pp. 42–47, Singapore, July 2008.

[7] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.

[8] J. Y. Hwang, J. M. Kang, Y. W. Jang, and H. C. Kim, "Development of novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly," in *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '04)*, pp. 2204–2207, San Francisco, Calif, USA, September 2004.

[9] H. J. Luinge and P. H. Veltink, "Inclination measurement of human movement using a 3-D accelerometer with autocalibration," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 1, pp. 112–121, 2004.

[10] G. Anania, A. Tognetti, N. Carbonaro et al., "Development of a novel algorithm for human fall detection using wearable sensors," in *Proceedings of the IEEE Sensors (SENSORS '08)*, pp. 1336–1339, Lecce, Italy, October 2008.

[11] A. Leone, G. Diraco, and P. Siciliano, "Detecting falls with 3D range camera in ambient assisted living applications: a preliminary study," *Medical Engineering and Physics*, vol. 33, no. 6, pp. 770–781, 2011.

[12] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS '05)*, pp. 3551–3554, Shanghai, China, September 2005.

[13] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & Posture*, vol. 28, no. 2, pp. 285–291, 2008.

[14] A. K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[15] A. K. Bourke, P. Van de Ven, A. E. Chaya, G. M. Olaighin, and J. Nelson, "Testing of a long-term fall detection system incorporated into a custom vest for the elderly," in *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '08)*, pp. 2844–2847, Vancouver, Canada, August 2008.

[16] A. K. Bourke, P. van de Ven, M. Gamble et al., "Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities," *Journal of Biomechanics*, vol. 43, no. 15, pp. 3051–3057, 2010.

[17] F. Bagalà, C. Becker, A. Cappello et al., "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS One*, vol. 7, Article ID e37062, 2012.

[18] S. -H. Liu and W. -C. Cheng, "Fall detection with the support vector machine during scripted and continuous unscripted activities," *Sensor*, vol. 12, no. 9, pp. 12301–12316, 2012.

[19] A. Leone, G. Diraco, C. Distante et al., "A multi-sensor approach for people fall detection in home environment," in *Proceedings of the Workshop on Multi-Camera and Multi-Modal Sensor Fusion Algorithms and Applications (M2SFA2 '08)*, pp. 1–12, 2008.

[20] N. Noury, A. Fleury, P. Rumeau et al., "Fall detection—principles and methods," in *Proceedings of the 29th Annual International Conference of IEEE-EMBS, Engineering in Medicine and Biology Society (EMBC '07)*, pp. 1663–1666, Lyon, France, August 2007.

[21] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *Journal of Machine Learning Research*, vol. 2, no. 1, pp. 139–154, 2001.

[22] V. Franc and V. Hlavac, "Statistical Pattern Recognition Toolbox (STPRtool)".

[23] T. Zhang, J. Wang, L. Xu, and P. Liu, "Fall detection by wearable sensor and one-class SVM algorithm," *Lecture Notes in Control and Information Sciences*, vol. 345, pp. 858–863, 2006.

[24] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.

[25] H. Cheng, H. Luo, and F. Zhao, "A fall detection algorithm based on pattern recognition and human posture analysis," in *Proceedings of the IET International Conference on Communication Technology and Application (ICCTA '11)*, vol. 2011, pp. 853–857, Beijing, China.

[26] R. Sangeetha and B. Kalpana, "A comparative study and choice of an appropriate kernel for support vector machines," in *Information and Communication Technologies*, V. V. Das and R. Vijaykumar, Eds., vol. 101 of *Communications in Computer and Information Science*, pp. 549–553, Springer, Heidelberg, Germany, 2010.

[27] R. Sangeetha and B. Kalpana, "Performance evaluation of kernels in multiclass support vector machines," *International Journal of Soft Computing and Engineering*, vol. 1, no. 5, pp. 138–145, 2011.

[28] M. Grassi, A. Lombardi, G. Rescio et al., "A hardware-software framework for high-reliability people fall detection," in *Proceedings of the IEEE Sensors (SENSORS '08)*, pp. 1328–1331, Lecce, Italy, October 2009.

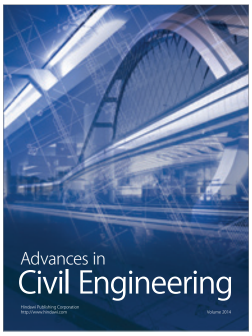
Journal of
Engineering

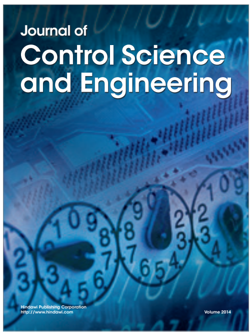
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

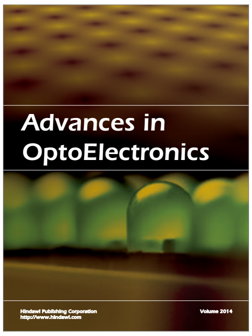
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

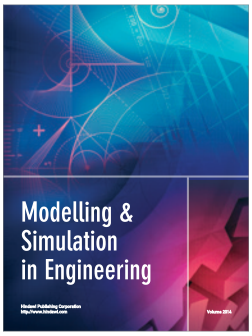
Journal of
Robotics

Journal of
Electrical and Computer Engineering

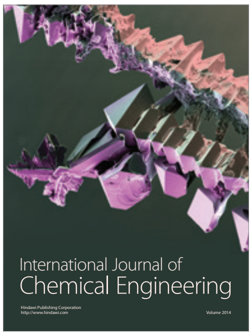

Advances in
OptoElectronics

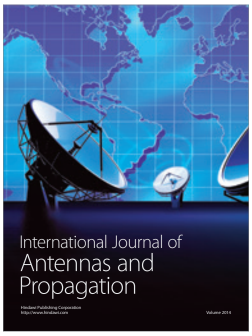
VLSI Design

International Journal of
Navigation and Observation

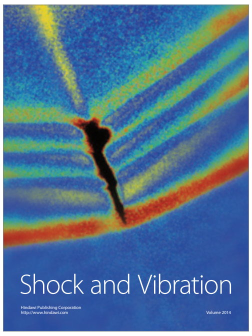

Modelling & Simulation in Engineering

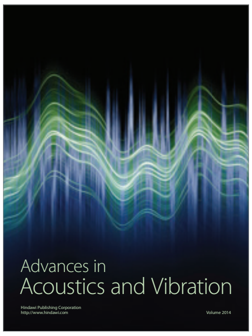
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration