FALL DETECTOR BASED ON NEURAL NETWORKS

Rubén Blasco, Roberto Casas, Álvaro Marco, Victorián Coarasa, Yolanda Garrido and Jorge L. Falcó
Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, María de Luna 1, Zaragoza, Spain

Keywords: Fall detector, neural networks, ZigBee, wearable sensors, pattern recognition.

Abstract: Falls are one of the biggest concerns of elderly people. This paper addresses a fall detection system which uses an accelerometer to collect body accelerations, ZigBee to send relevant data when a fall might have happened and a neural network to recognize fall patterns. This method presents improved performance compared to traditional basic-threshold systems. Main advantage is that fall detection ratio is higher on neural network based systems. Another important issue is the high immunity to events not being falls, but with similar patterns (e.g. sitting in a sofa abruptly), usually confused with real falls. Minimization of these occurrences has big influence on the confidence the user has on the system.

1 INTRODUCTION

Aging of population is a well-known problem in developed countries. Nowadays, elderly people (+65 years old) represents in Spain more than 16% of the population (Eurostat, 2007). Falls are one of the major fears of the elderly and their relatives. Indeed, some authors estimate the amount of falls of people aged over 75 to be at least 30 percent per year (Sixsmith and Johnson, 2004). In the end, people's concern about falls and whether there will be someone there to help them in case of an emergency, prevent them to age at home (Rodriguez et al, 2005). As a result, people have to move to residences, usually causing negative effect in their health and happiness and resulting in high costs to the individual, their family or the Social Welfare System.

Fortunately, many initiatives are going on in order to increase the time people can stay at home. We will further see many fall detection systems enabling people to receive quickly assistance even when they are not able to request the assistance by themselves (e.g. immobilized or unconscious). Also, combination of these systems with telemedicine allows closer monitoring or collaboration of various experts in the diagnoses (Tunstall web, 2007).

Various methods have been described in order to detect falls in the elderly. Those based in a sensing infrastructure - infrared cameras (Alwan, et al., 2006), vision systems (Williams et al., 1998) or smart floors (Williams et al., 1998) - can be hardly used in many cases. We find wearable systems more appropriated in real scenarios because people refuse to have cameras everywhere in their homes and systems are much more expensive.

Inertial elements are mostly used for mobile monitoring, but still the perfect detector does not exist. Main reason is the difficulty in modelling a fall, it can happen in many different ways; it will not always be the typical big impact followed by inactivity and horizontality. Williams et al. use a shock sensor and a tilt switch to measure the inclination after the impact (Williams et al., 1998). Doughty et al. also use two sensors to perform the same two-stage-analysis (Doughty et al, 2000), which moreover is concreted in a commercial gadget from Tunstall (Nait-Charif and McKenna, 2004). Noury refines the procedure using an accelerometer to detect the shock, also a tilt switch, and adding a vibration sensor to estimate the physiological activity (Noury, 2002). Of course, the more variables measured, the more accurate the detection can be, but also the more complicated and expensive the hardware will be. Many actual works propose just using accelerometers to carry out the full detection (Noury, 2002; Degen et al, 2005; Chen et al., 2005). Main reasons are their low power consumption, reduced cost and versatility detecting different events -shocks, inclination and activity-. The devices presented in these works perform satisfactory fall detection: more of 80% of falls are correctly detected (Noury, 2002).

Blasco R., Casas R., Marco Á., Coarasa V., Garrido Y. and L. Falcó J. (2008).
FALL DETECTOR BASED ON NEURAL NETWORKS.
In *Proceedings of the First International Conference on Bio-inspired Systems and Signal Processing*, pages 540-545
DOI: 10.5220/0001066205400545

2 SYSTEM DESCRIPTION

Users must find fall detection systems trust-worthy and efficient in order to use them. Systems which detect all falls but generate many false alarms make users unconfident about it. Moreover, if we consider the difficulty of distinguishing between some kind of falls and ordinary movements in elderly people's life, threshold systems (those that generate an alert when acceleration rise above a fixed value) become not be reliable enough (Noury, 2002).

Figure 1.a. shows tri-axial acceleration when a person has a sideward fall. On the other hand, Figure 1.b. shows accelerations when a person sits down on a sofa abruptly. Both figures were obtained with a device which measures triaxial accelerations, hanged around the neck.

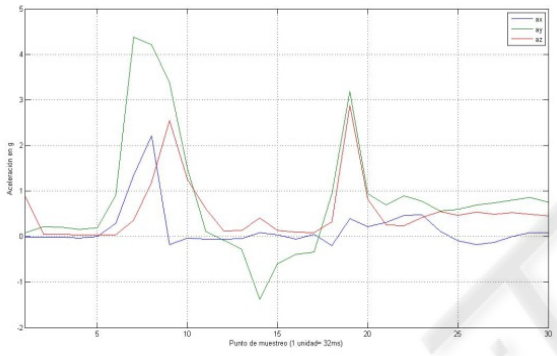

Figure 1a: Acceleration in three axes in a sideward fall.

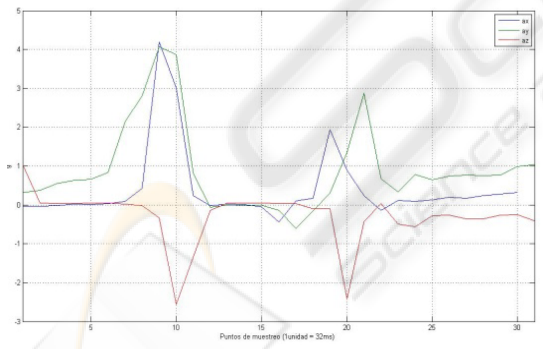

Figure 1b: Acceleration in three axes while sitting abruptly in a sofa.

As we can see, both figures have similar acceleration peaks being also the shapes pretty similar.

Our solution aims to distinguish falls from movements that have similar acceleration patterns not being falls using neural networks; that is to say, separate occurrences into true and false falls.

2.1 Blocks Diagram

The fall detector consists of a mobile device with an inertial sensor which is able to communicate through a ZigBee network. The system also needs a computer that analyzes data using a neural network. Figure 2 shows the portable device blocks: battery, sensor, microcontroller (µC), interface and Zigbee transceiver. Reduced size and low power consumption had been considered in the design process of every block.

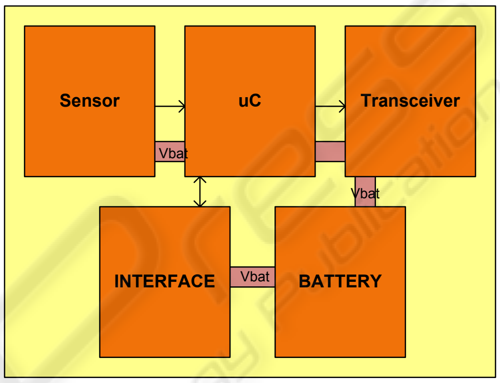

Figure 2: Blocks of the portable device.

The chosen sensor is MMA7260Q Freescale accelerometer because of its wide input voltage range (2,2 V - 3,6 V), current consumption (typically less than 500 µA and 3 µA in sleep mode) and reduced size (6x6x1,45mm). It also has three analog outputs that give the acceleration value in axis X, Y and Z. Its sensitivity is configured digitally into some ranges (1,5 g; 2 g; 4 g or 6 g). As some falls are above 4 g, our application uses the maximum range (6 g) and minimum sensitivity (200 mV/g).

The chosen microcontroller is Microchip's PIC16F688. It has eight A/D channels that can be configured to 10 bits. As well as working within a wide voltage supply range (2 V - 5,5 V), it also has very low current consumption (800 µA in active mode and 1 nA while sleeping).

Regarding communications, we discarded the development of a proprietary network for interoperability reasons. Other standard wireless protocols such as Bluetooth or WiFi consume too much energy as they are intended for higher data rates. We decided to use ZigBee because its adequate data rate (250 kbps), security (128 bits AES encryption), low latency (30 ms to join and 15 ms to access the network) and energy efficiency. Its interoperability with other potential applications (home control and automation), future projection of the protocol, and its consequent cost reduction were other strategic reasons behind our decision (Geer, 2005).

The chosen ZigBee chip is ETRX2 from Telegesis. This is a ZigBee module on the 2,4 GHz ISM band based upon the Ember's EM250 chip. We used the development environment proposed by Ember to develop a ZigBee-compliant network following mesh topology (ZigBee Alliance, 2007). The chip consumes 30 mA when receiving or transmitting data and 10 µA in sleep mode. As we will use the radio exceptionally, just when are reasonable indications about a fall (when a threshold is exceeded), average power consumption due to communication is reduced.

The user interface consists of a single button and a buzzer for user interaction. Figure 3 shows the mobile device prototype. Its size, including battery, is 58x36x16 mm and it weights 30 gr.

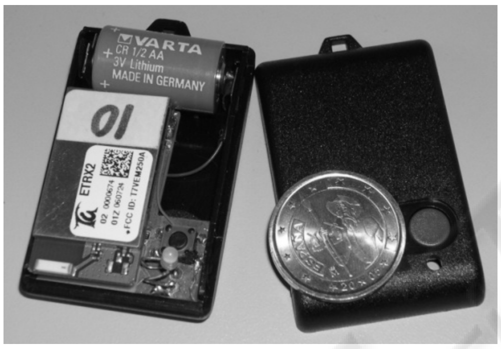

Figure 3: Mobile device prototype.

In order to make the device useful is extremely important to keep it on working long time using the same set of batteries. That is why we gave preference to power-conservative and size of batteries among other designing requirements like transmission rate or processing time. Precise battery life estimation is very difficult because it will depende on the number of false alarms generated; every time the threshold is exceeded it sends data via ZigBee. Anycase, with the battery used (3 V, 1000 mA·h), it can last for several months daily sending several false falls to analize.

2.2 Software

As we said before, we designed a neural net to detect falls also aiming to minimize the number of false-falls compared to simple threshold based detectors.

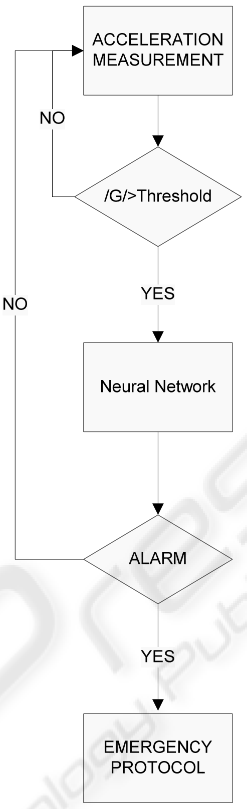

Figure 4: System's simplified flow diagram.

In our case, we use an acceleration threshold to get the "acceleration pattern" of the possible fall to be studied. Every 32 ms the device stores the current acceleration measurements. It keeps a buffer with the last 5 samples ($t_1 \approx 160$ ms). If the threshold is exceeded, a possible fall might have happened. Then we gather 25 samples more ($t_2 \approx 800$ ms) and all the data (960 ms) is sent via ZigBee to the PC. As we will see in section 3.2, those times and the threshold have been empirically set through acceleration pattern analysis of many falls and false-falls.

The "window time" ($t_w=t_1+t_2$) represents the time that the neural net analyzes the data in order to relation the detected event to a true fall.

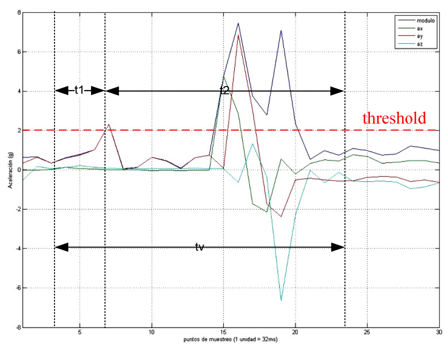

Figure 5: Window time.

In case the neural network detects a real fall, the PC asks the mobile device to buzz for one minute. During this time, the user can cancel the fall situation pressing the button; the user is okay and does not need assistance. In other case, an alert is sent to an assistance center asking for help.

3 NEURAL NETWORK DESIGN

We have chosen MLP (MultiLayer Perceptron) architecture because is the best neural network for pattern classification (Del Hoyo Alonso, 2003). MLPs are feedforward neural networks trained with the standard backpropagation algorithm. They are networks that learn how to transform input data into a desired response. As they are supervised, they require a set of known patterns with known responses to get trained. With one or two hidden layers, they can approximate virtually any input-output map. They have been shown to approximate the performance of optimal statistical classifiers in difficult problems (Neurosolution web, 2007).

Every acceleration point, within window time, is considered as input data to train neural network to distinguish between true and false falls (figures 1a and 1b). That is to say, if an event is represented by 30 samples for each axis (X, Y and Z), the number of inputs will be 90 (30x3). Consequently, it is the same as we give the net the whole graph to compare and classify.

We have decided to train the net with one hidden layer. To check if our choice is convenient or not, we have designed a test bench with different numbers of neurons, studying the absolute error in each case. To accelerate the training, we have chosen a bipolar sigmoid activation function for neurons of the hidden layer. The activation function of the output neuron is unipolar sigmoid so the output looks like a binary signal (1 = TRUE FALL; 0 = FALSE FALL).

The suitable number of neurons of the hidden layer is obtained doing simulations of different neural nets. Finally, we choose the one which produces the minimum absolute error. To reduce the number of simulations and to get patterns from the inputs able to generalize the results, we have defined a requirement: the number of inputs is greater than the number of neurons of the hidden layer.

3.1 Input Data Harvesting

Ten people of different ages, weight, height and sex imitated the movements of elderly people to create a data base of falls.

Table 1: Volunteers' characteristics.

Age range	25-40 years
Weight range	44-105 kg
Height range	1.58-1.90 m

To get the data as close to reality as possible, the volunteers had the acceleration detector hanged around the neck. Volunteers were asked to simulate true and false falls situations.

TRUE falls:

Every volunteer falls down 10 times on a straw mat. The fall intensity changed (rough and soft) and the way of falling down too (side, front, backwards), hitting the ground with their back, hip, knees, etc.

FALSE falls:

Every volunteer flings himself down 5 times on the center and 5 times on the side of a sofa.
Every volunteer stumbles and hits a wall without falling down 5 times.
Every volunteer walks around for 2 minutes doing normal movements like sitting up and down in chairs, picking up things, etc.

During the test, the fall detector continuously samples the three acceleration axes each 32 ms sending them to a PC working as a data logger. In the end, we get a file with all the acceleration samples in axis X, Y and Z for every volunteer. The resulting data base consists of 99 samples of true falls (we had one error while collecting data) and 150 of false falls.

3.2 Input Data Analysis

First of all, data analysis has determined the window time length. After studying all the falls, we decided that an event could be represented with 30 samples (tv = 960 ms; t1 = 160 ms; t2 = 800 ms). This means that the microcontroller has to store always in memory the last five samples to send, in case the acceleration threshold is exceeded, the event to the PC to be analyzed.

With the window time selected, the number of inputs to the neuronal network is set to 90. In order to reduce the number of network entries -and consequently the network size- we have done a PCA (Principal Component Analysis). This method lies in referencing input data to a new origin and coordinate base.

In the new reference, the main components are chosen to be those with the maximum variance among samples (those with the highest covariance).

Therefore, if we take the samples representing more than 95% of covariance, the number of input will be reduced without losing significant information. This leads to suppose that the greater is the variance of an input, the more information it gives.

The acceleration threshold was decided experimentally. At first, guided by most of the bibliography (Chen et al., 2005), we chose a 3 g value. Then 97 out of 99 true falls and 121 out of 150 false falls surpassed the selected threshold.

Missing true falls is far worse than over-detecting false falls, thus we reduced the threshold to 2 g to prevent losing any fall. As expected, we got all the falls, but the number of false falls which surpassed the threshold, increased to 241 because even normal movements triggered the detection process.

After using PCA analysis with the 340 events (99 falls plus 241 false-falls), the number of inputs was reduced from 90 to 55, keeping the 95% of the covariance of the original data.

3.3 Network Performance

The network was trained used Levenberg-Marquardt algorithm (Neural-toolbox in Matlab).

We trained different MLP architectures 55xMx1 (being M the number of neurons in the hidden layer, 5≤M≤35). We repeated this process ten times in order to ensure the network design and its performance. Each test randomly selected 80% of the events for training and 20% for validating. That is to say, from the whole 340 events (99 falls plus 241 false-falls), the validation group had 20 true falls and 48 events that could be confused with falls. In the end, a neural net with 22 hidden neurons was able to classify falls correctly.

When interpreting the neural net output give precedence to the fall detection. Thus, we decided that if the output is above or equal to 0.3, a fall is detected. On the other hand, if the output is below 0.3, the analyzed event was not a true fall.

In table 2 we can see the network performance for the ten tests.

Table 2: Validation group detection results.

	Network fall detection / Fall events	Network fall detection / False-fall events
Test 1	20 / 20	0 / 48
Test 2	20 / 20	0 / 48
Test 3	20 / 20	1 / 48
Test 4	20 / 20	0 / 48
Test 5	18 / 20	1 / 48
Test 6	20 / 20	1 / 48
Test 7	16 / 20	1 / 48
Test 8	18 / 20	0 / 48
Test 9	17 / 20	1 / 48
Test 10	15 / 20	0 / 48

We can see how the network is able to detect 92% of all the falls and filter up 99% of the events that can be confused with falls.

In figure 6, we show the network output we got for the validating group in third test.

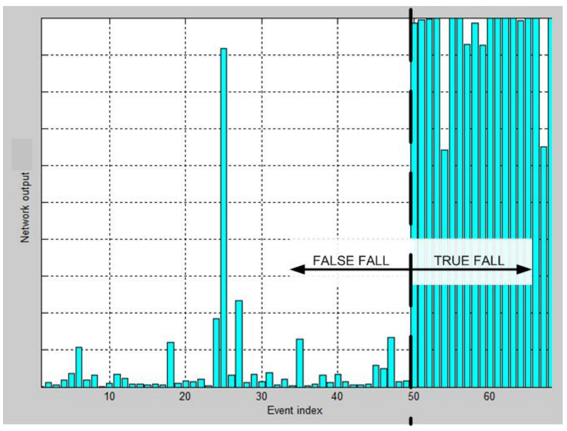

Figure 6: Network output for the third validation group.

4 CONCLUSIONS

The final results using MLP neural networks for fall detection have been quite satisfactory. The application classifies correctly 92% of the validation group falls, better performance than other detection methods: 80% in (Chen et al., 2005). Moreover, the number of false alarms is drastically reduced to 1%, which leads to enhance users trust on the fall detector. Nevertheless, a more extensive study with more users being also elderly has to be developed in order to gather more data and confirm the results.

Although the portable device can run for months with the same battery, the system needs a computer to analyze all the data. In order to reduce costs, it is possible to analyze the pattern remotely. As the amount of exchanged data is reduced, it could be sent via ADSL (if the person is at home), GPRS or even SMS to a service center. Anyhow our application gets better performance than others embedded in a microcontroller but a higher cost and complexity. To overcome this, we are currently minimizing the neural network size so it can run in a microcontroller or FPGA.

ACKNOWLEDGEMENTS

This work was supported by the Spanish MCYT under project Ambiennet (TIN2006-15617-C03-02) and by the EU under projects MonAmi (IST-5-0535147) and EasyLine+ .

REFERENCES

Alwan, M., et al., 2006. ‘A Smart and Passive Floor-Vibration Based Fall Detector for Elderly’. Proc. The 2nd IEEE International Conference on Information & Communication Technologies: from Theory to Applications ICTTA’06, Damascus, Syria, April 2006.

Chen, J., et al., 2005. ‘Wearable Sensors for Reliable Fall Detection’. Proc. 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS 2005, September 2005, pp. 3551–3554.

Degen, T., et al, 2005. ‘SPEEDY: a fall detector in a wrist watch’. Proc 7th IEEE International Symposium on Wearable Computers, October 2005, pp. 184–187.

Del Hoyo Alonso, R. 2003. ‘Supervised classification with Associative SOM.’ IWANN 2003: 334-341

Doughty, K., et al, 2000. ‘The design of a practical and reliable fall detector for community and institutional telecare’, Journal of Telemedicine and Telecare, February 2000, 6, (1), pp. 150–154.

Eurostat web: http://epp.eurostat.ec.europa.eu/ last visited 2007

Geer, D., 2005. ‘Users make a Beeline for ZigBee sensor technology’. IEEE Computer Magazine, 38, pp. 16–19.

Hansen, T.R., et al., 2005. ‘Using smart sensors and a camera phone to detect and verify the fall of elderly persons’. Proc. European Medical & Biological Engineering Conference (EMBEC 2005), November 2005.

Nait-Charif, H., and McKenna, S. J., 2004. ‘Activity Summarisation and Fall Detection in a Supportive Home Environment’. Proc. of the 17th international Conference on Pattern Recognition, (Icpr'04), Washington DC, August 2004, 4, pp. 323–326.

Neurosolution web.
http://www.nd.com/definitions/mlp.htm last visited 2007.

Noury, N., 2002. ‘A Smart Sensor for the remote follow up of activity and fall detection of the elderly’. Proc. 2nd Annual International Special Topic Conference on Microtechnologies in Medicine & Biology, IEEE-MMB2002, Madison-USA, May 2002, pp. 314–317.

Rodriguez J.M., et al, 2005. ‘Portable System for Patient Monitoring With Wireless Technologies’. The European Journal for the Information Professional, October 2005, 5, (5), pp. 46–52.

Sixsmith, A., and Johnson, N., 2004. ‘A smart sensor to detect the falls of the elderly’, IEEE Pervasive Computing, April-June 2004, 3, (2), pp. 42–47.

Tunstall web. http://www.tunstall.co.uk last visited 2007.

Williams, G., et al., 1998. ‘A smart fall and activity monitor for telecare applications’. Proc. of the 20th IEEE Annual International Conference of the Engineering in Medicine and Biology Society, October 1998, 3, pp.1151–1154.

ZigBee Alliance: ‘ZigBee Standard’, 2007