Computer Applications in Electrical Engineering
Vol. 14 *2016*
DOI 10.21008/j.1508-4248.2016.0027

Burglary detection based on accelometric data using selected signal processing algorithms

Bogdan Fabiański, Krzysztof Nowopolski, Bartłomiej Wicher
Poznań University of Technology
60–965 Poznań, ul. Piotrowo 3a, e–mail: {bogdan.fabianski, krzysztof.nowopolski, barlomiej.wicher}@put.poznan.pl

The paper presents two approaches to the problem of burglary detection. The first one utilizes direct signal processing, while the other – artificial neural network (ANN). Both algorithms are compared in real operating conditions. The implementation of the algorithms was performed in a portable, battery operating devices that can be easily attached to the door. For direct comparison, two identical devices including several MEMS accelerometers and 32 bit microcontroller have been used – each with one algorithm implemented. The goal of using artificial neural network algorithm was to improve the performance of the burglary detection system in comparison to classical direct signal processing. The structure of ANN and required pre – processing of the input data, is presented and discussed as well. The article also describes the research system required to collecting the data for ANN training and to directly compare both algorithms. Finally, the results of behavior of the classification methods in real actual conditions is discussed.

KEYWORDS: MEMS, accelerometer sensor, data streaming, DSP, low–power MCU, alarm system, artificial neural network

1. Research system

1.1. General overview

The investigated research system is separated into several circuits, connected together by communication links. PC data client is present in the system as well, enabling data acquisition for future analysis. System architecture is outlined in Figure 1. Sensor boards S–1 and S–2 are equipped with two and three MEMS accelerometers, respectively ([1, 2]). The data collected by these boards are provided through the RS–485 link to the local data server board (DS). This board is a bridge to the Ethernet connection with mentioned PC client. The selected RS–485 and Ethernet interfaces were selected due to possibility of long distance data transmission.

The PC client was running a C# application responsible for logging information received from DS and its presentation on a chart. The application developed in Microsoft Visual Studio enables also storing received data in CSV file format, that can be easily imported to Matlab environment so as to be analyzed in more sophisticated way.

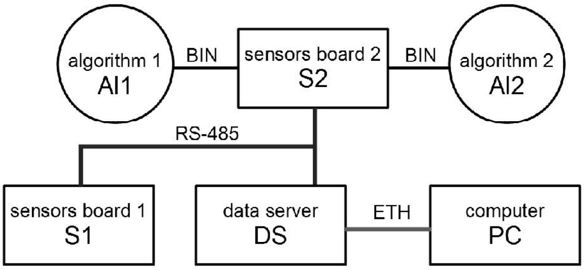


Fig. 1. System general structure overview

Since the sensor boards S1 and S2 are responsible only for accelerometers data acquisition, the data processing algorithms detecting burglary had to be implemented in separate devices. These boards are depicted in Figure 1 as Al–1 and Al–2. The assumption of the research was development of such burglary detection algorithms, so that can be implemented in a low–power real–time system based on a common microcontroller. Therefore, the system has to enable not only storing data received from data server DS and it analysis in Matlab, but also evaluation and verification of algorithms run in embedded systems. During the research system operation, Al–1 and Al–2 boards were running two different detection algorithms. The results of detection were provided by a simple binary interface to one of sensor boards, which transferred this information through RS–485 protocol to data sever (DS) and, further, to the PC (over the Ethernet link). The most powerful advantage of the system in the possibility of simultaneous acquisition of accelerometric data from several different MEMS devices and evaluation of the embedded detection algorithms. What is more, results of operation of these algorithms may be compared with their computer version, run in Matlab environment. Such a platform stands out against other laboratory setups due its great versatility [4].

1.2. Data source boards (DS–1, DS–2)

The data source boards DS–1 and DS–2 are presented in detail in Figure 2. Both of them are equipped with the same ultra–low–power microcontroller unit from STM32L0 family [3]. In a fact, these microcontroller units have only two tasks: acquisition signals from MEMS devices and sending these data to DS subsystem. Localization of sensor in specific boards is presented in Table 1.1. The S–2 boards is connected with two burglary detection boards by a binary link – this information is collected and sent through RS–485 protocol in the same manner as data from MEMS devices.

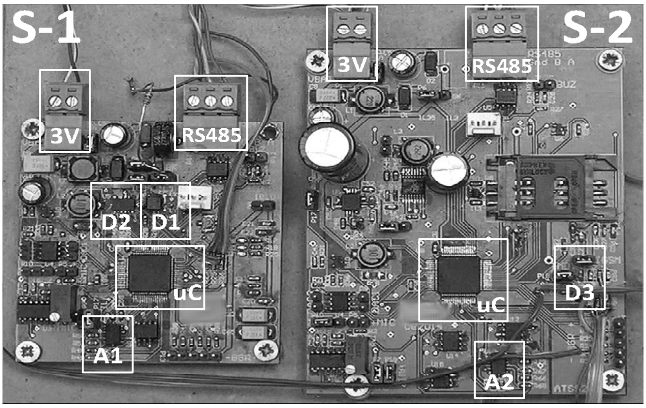


Fig. 2. Signal boards S–1 and S–2 in test bench. Labels: 'uC' – STM32L0 microcontroller, 'A1' – LIS352AX, 'A2' – MMA7361LC, 'D1' – LIS3DH, 'D2' – LIS35DE, 'D3' – MMA8451Q, '3V' – power connectors, 'RS485' – communication connectors

Table 1.1. MEMS sensor interfaces and localization in the research system

MEMS	LIS352AX	MMA7361LC	LIS3DH	LIS35DE	MMA8451Q
Interface	analog	analog	SPI	SPI	I2C
Board	S–1	S–2	S–1	S–1	S–2

1.3. Signal processing algorithm Boards (Al–1, Al–2)

The signal processing boards Al–1 and Al–2 ran two different algorithms of burglary detection, described in detail in sections 2.2 and 2.3. The boards are equipped with STM32L0 microcontroller, the same as S–1 and S–2 subsystems. When two different algorithms process the same signals simultaneously, an objective quality comparison is possible. In the presented system, there is no impact of place, time, and scenario.

It is worth to emphasize, that the test structure presented in this paper may be also utilized for automatic algorithm parameters adaptation. The automatic system of algorithms parameters optimization may be a superior method based e.g. on fuzzy logic or artificial neural networks training.

1.4. Data server and PC application

In the Figure 3, a complete laboratory setup with Ethernet server is presented. As mentioned before, DS (see Figure 1) is the RS485↔Ethernet interface translator. The RS485 frames are clustered to bigger packages (over 1 kB) and sent via Ethernet to PC system where are received by the dedicated application. Because the RS485 communication is fully synchronized by DS and data contains time information, Ethernet transmission delays have no impact on the further analysis, so also on data integration [6].

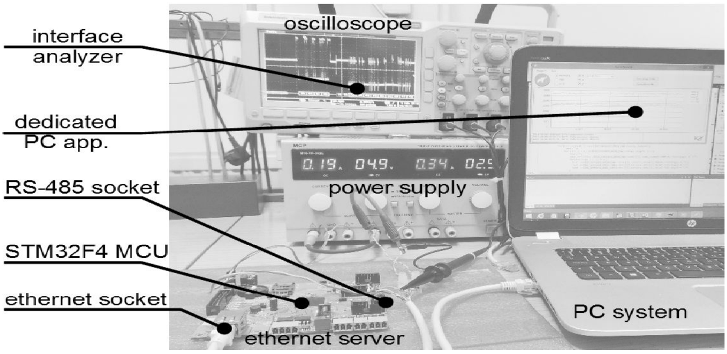


Fig. 3. Laboratory stand with Ethernet data server

The data server is a custom made design powered by STM32F4 microcontroller, with much more computational power than STM32L0, and is equipped with required Ethernet MAC peripheral, which the STM32L0 is devoid of. The server is described in more details in [3]. Tektronix DPO3054 oscilloscope is provided with embedded interface module.

2. Algorithms description

2.1. Methodology

As it has been already mentioned, MEMS accelerometer signals were acquired from S–1 and S–2 signal boards, then transferred to data server by RS–485 link and finally sent to PC system by Ethernet. The data were then saved in CSV format. The collected files were separated into different types of scenarios and the desired classification results were assigned to each file. This input data set was utilized during

The experimental tests were based on real–time computing by microcontroller mounted directly into the Al–1 and Al–2 devices. In both systems, sample time was set as 2 ms.

Besides of the unavoidable measurement noise, the signal drift is one of commonly known disadvantages of MEMS accelerometers [7]. In order to remove the bias of the measurement, which could initiate a false detection, at the system start, an average of several tens of samples was calculated, which was then treated as the signal bias. Averaging was necessary to reduce the impact of noise and signal drift.

2.2. Direct signal processing approach based on application phenomena (DDSP algorithm)

Let us assume, that M_k is the measured data input vector aggregating *x, y, z* accelerometer axes signal values in *k*–th sample of the algorithm execution:

$$M_k = [x_k, y_k, z_k] \tag{1}$$

In every algorithm step, values of dynamic D_k change in all axes and static S_k change in *x* axis of the accelerator are computed according to equations (2) and (3).

$$D_k = M_k - M_{k-1} \tag{2}$$

$$S_k = x_k - x_0 \tag{3}$$

The value of M_0 is acceleration value measured at the algorithm start:

$$M_0 = [x_0, y_0, z_0]$$

As it has been already mentioned in section 2.1, this value is an average of first measurements. It is worth to indicate, that in the first attempt of the authors, D_k and S_k values were computed as a difference squares, which deprecated low amplitude signals and made the system less sensitive. As it may be easily noticed, D_k is an equivalent of jerk (1st time derivative of acceleration).

Three accumulators are used in the algorithm: *AD* (aggregate dynamic), *AXS* (aggregate X axis static) and *TN* (threshold *N*). In each iteration, these values are computed as follows:

$$AD_k = AD_{k-1} + D_k \tag{4}$$

$$AXS_k = AXS_{k-1} + S_k \tag{5}$$

$$TN_k = TN_{k-1} + 1 \tag{6}$$

The abovementioned equations are computed only when a condition $D_k > L0L$ is fulfilled. The *L0L* value is a constant algorithm parameter, responsible for separation of the desired signal from noise values. In general, burglary event detection takes place, when *AD*, *AXS* and/or *TN* accumulators exceed appropriate limits. In the algorithm, two stages of detection are distinguished: warning (suspected activity was recognized) and alarm (certain access violation). Meaning of accumulators is following. *AD* accumulator is responsible for registration of changes occurring in axes. *AXS* accumulator, which is incremented by static difference S_k (3) is used to detect centrifugal acceleration, well reflected in S_k values when doors are being opened. *TN* counter is responsible for sensitivity to the signals of very low amplitude with a good separation from the noise (attempts to interfere in the lock of the door).

There are two conditions that cause warnings: $AD > L1L$ and $TN > L0N$. Alarm is reached also in two ways: one is condition $AD > L2L$ and the second way $AXS > L1XL$, where $L1XL = L1L/2$. The second condition is an equivalent of door opening detection which need to cause alarm state. The values of all of the mentioned parameters are collected and presented in the Table 2.1.

Equations (4)–(6) show only the accumulating phase. If left alone, it is only the matter of time when appropriate warning or alarm levels is exceeded (e.g. noise, which unexpectedly cross the *L0L* value, will be also integrated). The mechanism of subtraction is done in time domain with some constants assumed and it is done physically in timer peripheral interrupt execution procedure of the MCU (1000 Hz tick). Global accumulator *AD* value is decremented in 500 ms period according to:

$$AD_i = AD_{i-1} - ADD \tag{7}$$

AXS value (5) is decremented in the time interrupt at the same period:

$$AXS_i = AXS_{i-1} - AXSD \tag{8}$$

Corresponding with equations (7, 8), *TN* value is subtracted in *TI* in period of 100 ms in following way:

$$TN_i = TN_{i-1} - TND \tag{9}$$

The values of decrementation parameters are provided in Table 2.1 as well.

Table 2.1. DDSP algorithm parameters

Parameter	*L0L*	*L1L*	*L0N*	*L1XL*	*L2L*	*ADD*	*AXSD*	*TND*
Value	15	40 000	320	13 333	150 000	5 000	5 000	4

In the Figure 4, algorithms of warning and alarm excitation, which correspond with the nomenclature and equations (1)–(9) are presented. The additional description is required for *WaitForAXSElapsed*() function. This function is added to make possible of omitting warning state when centrifugal force is detected – then it means that the door is opened and alarm should be released – it is very likely, that when opening the door, only pressing the handle will give the accumulator the warning level.

The system of accumulators decrementing in time domain is explained in the left part of Figure 4. It is important, that corresponding values are decremented only when are bigger than zero (bottom limit) and incremented only when less than trigger level (top limit).

2.3. Neural Classifier (NC algorithm)

The second approach to the issue of burglary detection is based on application of artificial neural network. The classifying network was designed and trained off–line on the basis of previously collected samples, for the sensor placed in different conditions, e.g. hitting in the door, scratching, knocking etc.

The stored waveforms for these situations were assigned to a value representing the level of danger: 0 – normal conditions, 1 – warning, 2 – certain attempt of invasion. This assignment was done manually, on the basis of expert knowledge of the system designer.

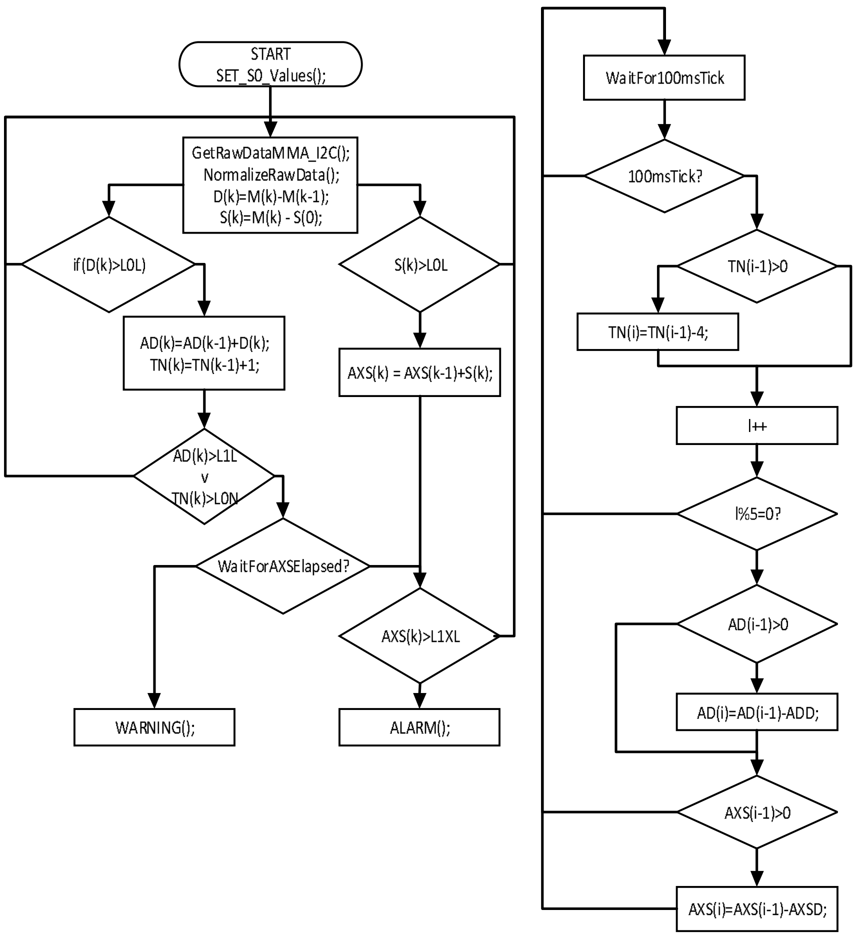


Fig. 4. Block diagrams of the main algorithm: left – incrementation of accumulators, right – decrementation

The next step was selection of proper input signals of the artificial neural network. Since the described classification based on single sample of acceleration measurements in three axes is impossible and can be done only if a longer horizon of samples is taken into consideration. Therefore, there were two possible approaches: prepare a network with a number of inputs big enough to provide the network with all of the samples from the assumed time horizon or calculate analytical factors, that may be treated as a kind of signal descriptors.

It was decided to apply the second solution, since the number of network inputs was smaller in this case. The following inputs of the neural classifier were selected: signal variance (10), difference between maximal and minimal sample value (11), discrete integral of the signal (12).

$$I_k^1 = \sum_{i=k-H+1}^{k} (x_i - x_{\mathrm{avg}})^2, \quad x_{\mathrm{avg}} = \frac{1}{H} \sum_{i=k-H+1}^{H} x_i \tag{10}$$

$$I_k^2 = \max(\{x_{k-H+1}, \dots, x_k\}) - \min(\{x_{k-H+1}, \dots, x_k\}) \tag{11}$$

$$I_k^3 = T_{\mathrm{s}} \cdot \sum_{i=k-H+1}^{H} x_i \tag{12}$$

All of the indicators were calculated for each of the axes separately in the horizon H of the last 50 samples. It shall be emphasized, that the values of these factors were calculate for unbiased signal values, which were additionally preprocessed by a first order IIR (infinite impulse response) discrete filter, described by transfer function (13) with time constant $T = 40$ ms.

$$G(z) = \frac{z \cdot \Delta t}{T \cdot z - T + \Delta t \cdot z} \tag{13}$$

The mentioned nine factors were used as the neural network inputs $x_{\{1..9\}}$, which layout is presented in figure 5. These values were normalized and delivered to first (hidden) layer of the network, that consisted of 20 neurons with sigmoidal activation function:

$$y_n^{(1)} = \frac{2}{1 + \exp\left(-2 \cdot \left(\sum_{i=1}^{9} a_i^{(1)} x_i + b_n^{(1)}\right)\right)} - 1 \tag{14}$$

where $y^{(1)}{}_n$ is the output value of the n–th neuron in the first layer, x_i and a_i are the i–th input and its scaling factor, b_n is the neuron bias. The second layer consisted of only one neuron $y^{(2)}$ with linear output:

$$y^{(2)} = \sum_{i=1}^{20} a_i^{(2)} y_i^{(1)} + b^{(2)} \tag{15}$$

The network was trained off–line using the Neural Network Pattern Recognition Tool, provided by MathWorks, according to the scaled conjugate gradient backpropagation algorithm (SCG) [8]. The main advantages of this training method are: memory complexity linearly related with total network weight number and an order of magnitude faster convergence than classical backpropagation algorithm [9].

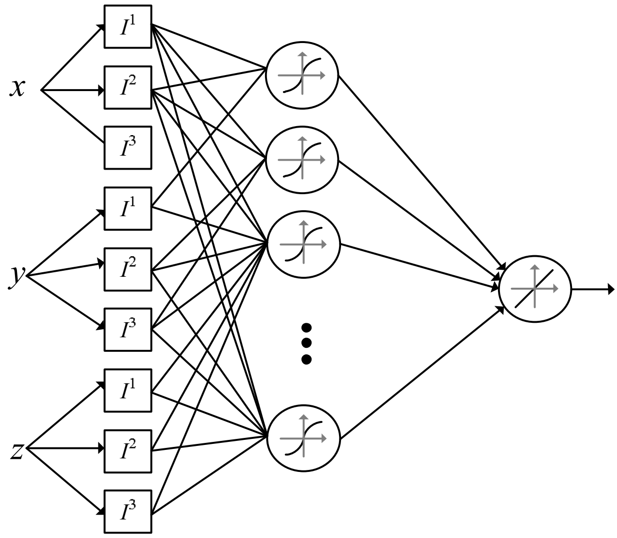


Fig. 5. Structure of applied neural network: 9 inputs defined according to (10)–(12), 20 neurons in the hidden layer, one linear output neuron

During implementation of the NC algorithm in the embedded system, it turned out, that the microcontroller unit is not efficient enough to process all data with frequency equal to the sampling of 500 Hz. Therefore, the neural classifier algorithm step was limited to Δt = 6 ms. The observed computational complexity is related with two facts. Firstly, the applied MCU does not possess floating–point coprocessor, therefore calculations of sigmoidal activation function appeared especially time consuming. Secondly, the ANN input factors (variance, dynamics and integral) have to be computed in every algorithm sample for three vectors of 50 samples. Such tasks are a kind of challenge for a very low–power microprocessor system.

3. Experimental verification

3.1. Methodology

The data set required for conducting simulations are obtained from MMA8541 accelerometer located on data source board (Figure 2). This sensor offers best signal to noise ratio from all of the tested devices [4]. The data source boards are mounted rigidly on internal door. Source data for off–line algorithm evaluation are grabbed by the dedicated streaming system. Together with the sensor signals, algorithm outputs are transferred, what gives the possibility of summary the results of the experimental system operation in a reliable way.

3.2. Simulation models

The simulation versions of the detection algorithms were design so as to enable simple translation of these models to C code for the embedded system. The utilized simulation software was Matlab, and the detection algorithms were defined as M–files scripts. This approach made the preparation of the algorithms much more convenient, due to obvious advantages of the scripting language simplicity. What is more, the Matlab scripting language is relatively similar to the language C, since the most of the basic algorithmic structures are similar.

The DDSP algorithm described in subsection 2.2 was developed in its simulation version directly, i.e. made of basic script directives, while the neural network classifier was generated by the Neural Network Pattern Recognition Matlab tool. However, the automatically generated structure was defined as a clear M–file including functions specific to the mentioned toolbox, but these commands may be relatively conveniently translated to corresponding C structures.

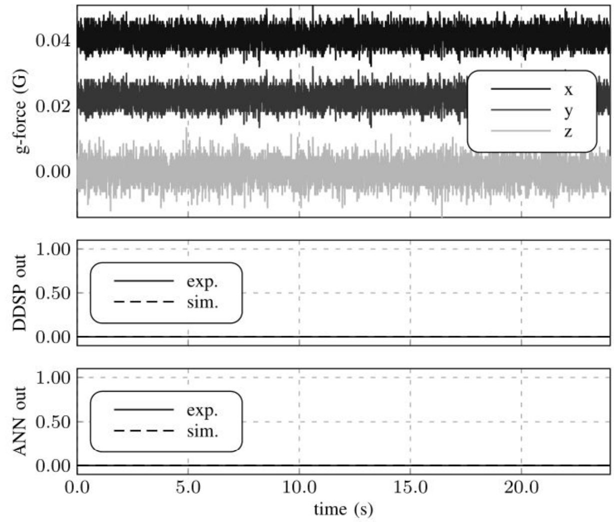
3.3. Experimental results

Results of simulation and experimental verification of the discussed algorithms are presented in Figures 6–9. In Figure 6, possible dynamics of accelerometric signals is also depicted – silence is contrasted with very rapid opening the door. As it is clearly visible, both algorithm remained zero output and no false alarm was excited in the first case. Unlikely, in second scenario, fast and appropriate response of both of the algorithms may be observed (DDSP approach revealed slightly faster response).

Algorithms responses to opening the door and knocking are presented in Figures 7 and 8, respectively. Some divergent may be observed between simulation and embedded implementations of the algorithms. It may explained by the fact, that the Al–1 and Al–2 boards possessed their own accelerometers, different than the MEMS devices attached in the S–1 and S–2 boards. Hence, reading from accelerometers located in slightly different part of the test bench has to vary, even if it is the same device model. However, the general tendencies of responses are remained in simulation and embedded versions of the algorithms.

The most important conclusions are that the system is stable during silence and there is a difference between operation in door lock (warning) and door opening (unconditional alarm). The convergence of simulation with experiment is easy to be noticed, but better when DDSP is used. Overall positive detection rate is a little better with DDSP (Al–2) algorithm (0,83 to 0,67), but – as mentioned before – more advanced research and further evaluation are necessary.

a)



b)

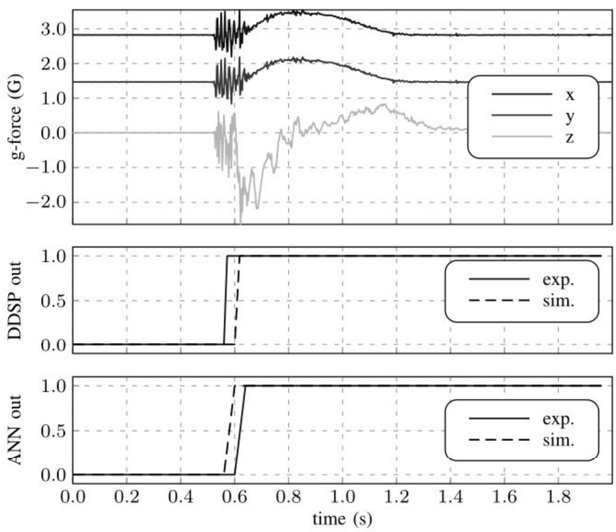


Fig. 6. Waveforms of registered accelerometric signals and results of detection algorithms operations: a) silence, b) rapid opening the door

a)

b)

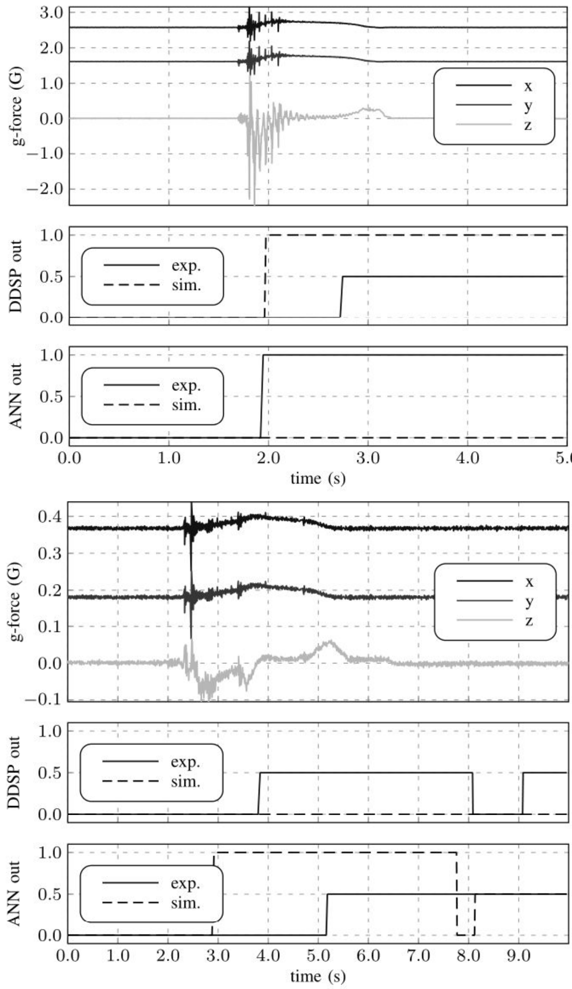


Fig. 7. Waveforms of registered accelerometric signals and results of detection algorithms operations during opening the door: a) normal, b) very slow opening

Table. 3.1. Summary of simulation and experimental test runs at different scenarios

Algorithm name	Detection rate [%]		
	Intrusion	**Warning**	**False det.**
Direct DSP: **experiment** (simulation)	**100**(100)	**66**(66)	**0**(0)
Neural classifier: **experiment** (simulation)	**100**(100)	**33**(100)	**0**(0)

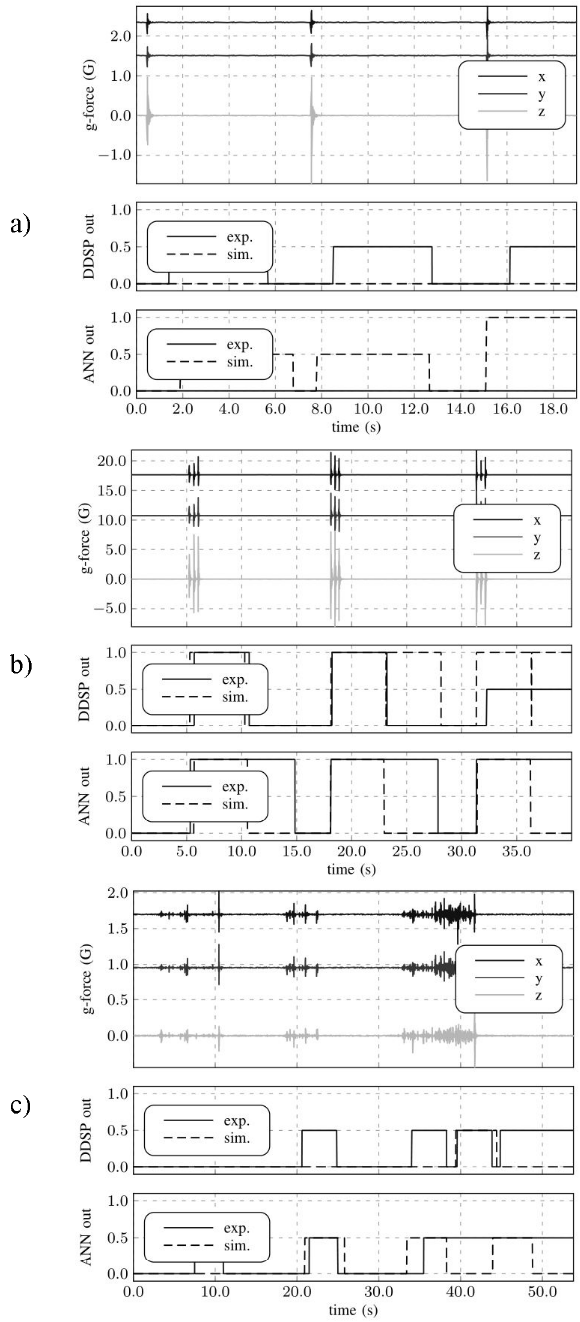


Fig. 8. Waveforms of registered accelerometric signals and results of detection algorithms operations during knocking the door: a) normal, b) very loud, c) manipulation near to the lock

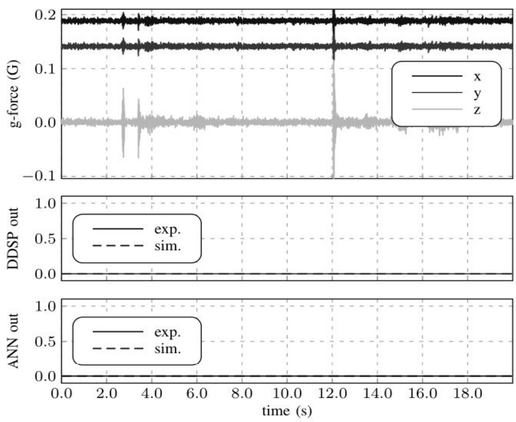


Fig. 9. Waveforms of registered accelerometric signals and results of detection algorithms operations during silent behaviour – gentle scratching

4. Summary

The system source of the MEMS accelerometer signals was presented in the article. Two different algorithms were described – the first was called direct digital signal processing approach (DDSP), which was based on application phenomena, and the second was a neural classifier.

Algorithms were implemented both in the simulation environment (Matlab/Simulink) and in the embedded (experimental) systems, in the way the alarm state from each system could be compiled with the accelerometer signals. That gave the possibility for the real–time data acquisition and then analyzing it offline in order to improve the effectiveness of the algorithms.

The simulation results summary Table 3.1 shows the high quality of used classic (DDSP) approach in the application of source signal classifier (door intrusion detection). In almost all cases (scenarios) warning or alarm state is correctly classified (positive detection factor 0.83 in DDSP and 0.67 in NC). Experimental verification confirms the conclusions from the simulations.

It is planned to made the system in both DDSP and NC structure self–adapting. The new algorithm will be based on fuzzy logic, since such an algorithm has a generalizing features which are important in the desired application and rules table in FLA system will base on the general knowledge what makes implementation more realistic.

References

[1] Finkbeiner S., "MEMS for automotive and consumer electronics," in *Solid–State Device Research Conference (ESSDERC), 2013 Proceedings of the European*, 2013, pp. 9–14.

[2] Perlmutter M. and Robin L., "High–performance, low cost inertial MEMS: A market in motion!," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, 2012, pp. 225–229.
[3] "STM32L052R8 Ultra–low power ARM Cortex–M0+ MCU with 64 Kbytes Flash, 32 MHz CPU, USB – STMicroelectronics," 09–Feb–2016. [Online]. Available:http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1817/LN184
[4] Fabiański B., Nowopolski K., and Wicher B., "The System of Streaming and Analysis of Signals from MEMS Accelerometers," *Poznan University of Technology Academic Journals*, Vol. 87, 2016, pp. 267–278, ISSN 1897–0737.
[5] Awasthi S. and Joshi A., "MEMS accelerometer based system for motion analysis," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 762–767.
[6] Fabianski B., "Embedded System of Critical Information Management," *Poznan University of Technology Academic Journals*, vol. 76, pp. 143–149, 2013.
[7] Chen Y.–Y., Chang S.–J., Huang C.–Y., and Hsinag C. Y., "An Elimination Design for drift rate effects of MEMS–Based Inertial Devices," in *Microsystems, Packaging, Assembly Circuits Technology Conference, 2008. IMPACT 2008. 3rd International*, 2008, pp. 153–155.
[8] Møller M. F., "A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.
[9] Rumelhart D. E., Hinton G. E., and Williams R. J., "Learning internal representations by error propagation," DTIC Document, 1985.

(Received: 5. 10. 2016, revised: 8. 11. 2016)