

Bogdan FABIAŃSKI*
Krzysztof NOWOPOLSKI*
Bartłomiej WICHER*

SELECTED ALGORITHMS OF MEMS ACCELEROMETERS SIGNAL PROCESSING IN BURGLARY DETECTOR APPLICATION

In the paper, implementations and results of operation of artificial neural network applied as a burglary classifier are presented in comparison to solution with a direct digital signal processing (DSP) approach. The neural network operates in a mobile access control device, that may be easily attached to a door. The device is an integrated system, equipped with several sensors based on microelectromechanical systems (MEMS) technology. Due to limited effectiveness of simple, conditional logic algorithms on acquired signal samples, a more sophisticated approaches are investigated. Data acquisition during imitation of various burglary scenarios and further processing of the recorded signals are described in the paper. Selection of the neural network structure and pre-processing methods of sensor signals are presented as well. The direct DSP algorithm based on the application of the properties of application phenomena is shown in the same way. Finally, results of selected algorithms implementation in a low-power 32-bit microcontroller system are presented. Limitation of the platform responsiveness in the real-time conditions and comparison of used classification methods are discussed in the paper conclusions.

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

1. SYSTEM ARCHITECTURE

1.1. General description

The system consist of several independent boards, which create two networks, based on RS485 and Ethernet (Figure 1). The main sensor boards S1 and S2 include five different microelectromechanical systems (MEMS) sensors ([1, 2]). These boards are integrated through RS485 network, where also data server (DS, Figure 1) unit is present. DS is in a role of protocol translator from

* Poznan University of Technology.

RS485, which is possible to implement in embedded system (STM32L0 [3] microcontroller which is S1/S2 core do not serve an Ethernet peripheral), to Ethernet, which is easy to use in PC environment. Both interfaces have the advantage of long distance transmission.

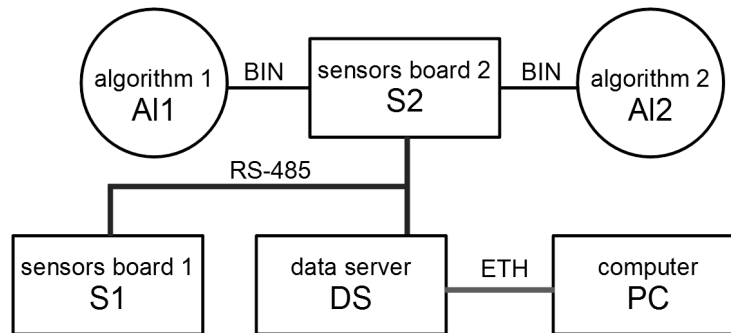


Fig. 1. System general structure overview

A dedicated PC application was written in C# language with use of Microsoft Visual Studio IDE. The application makes the system much more efficient in the way of captured data analysis due to its intuitive user interface and built-in functions (e.g. data conversion into format compatible with Matlab environment).

Algorithm boards (AI1/AI2 in the Figure 1) are important for the article main subject. They are independent, stand-alone, battery supplied devices with only one MEMS sensor installed per board. This way, two different algorithms may be evaluated at one time, with the same input signals. The results of real-time algorithms operation are sent by binary interface (described as BIN in the figure 1) and added to the RS-485/Ethernet data stream, containing signals registered by MEMS sensors on S1/S2 boards, without any additional AI1/AI2 system load. This system, described more widely in [4], is much more sophisticated than other test setups [5].

1.2. Data source boards

In the Figure 2 two boards, S1 and S2, are presented. These boards are the source of the reference signals. S1/S2 are driven by STM32L0 [3] ultra-low-power microcontroller unit (88 μ A/MHz in run mode, 1.65 V to 3.6 V power supply). Onboard interfaces give the possibility of data capture from several different sensors equipped with different interfaces: analogue (LIS352AX pointed on the Figure 2 as A1 and MMA7361LC as A2), SPI (LIS3DH as D1 sensor, LIS35DE as D2 sensor) and I2C (MMA8451Q as D3 sensor).

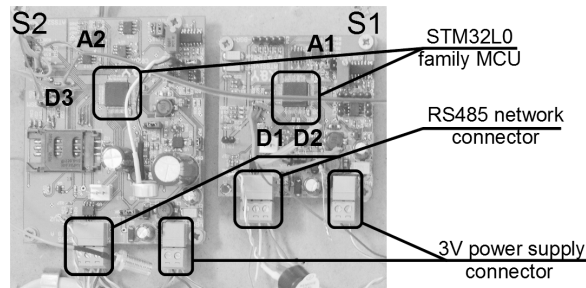


Fig. 2. MEMS signal source boards

1.3. Experimental classification units AI

The Figure 3 presents independent signal analyzer units connected via binary interface to the S2 board (these binary signals are added to the RS485 MEMS data stream and the reason to select S2 rather than S1 is that S1 has more data to send, what assures equilibrium of the amount of data sent by each board). Simultaneous operation of AI1 and AI2 boards, running two different algorithms in the real-time, makes the objective quality comparison of the intrusion detection algorithms possible – there is no impact of place, time, and scenario.

Such a structure may be also used in the automatic algorithm parameters adaptation in the test cycle (with the use of fuzzy logic, based on the theory of artificial neural networks training, or other analytical methods).

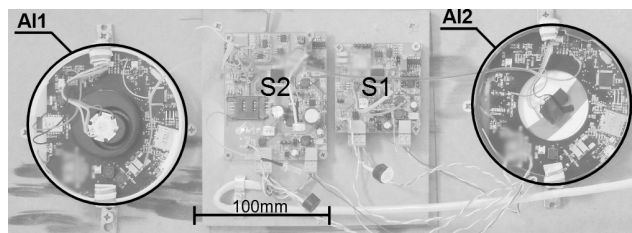


Fig. 3. Independent signal analyser units

1.4. Data server and PC application

In the Figure 4, 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].

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

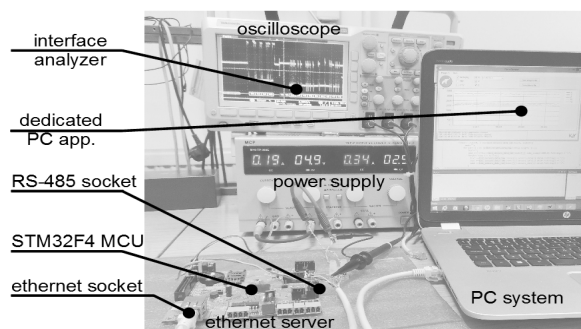


Fig. 4. Laboratory stand with Ethernet data server

2. ALGORITHMS DESCRIPTION

2.1. Methodology

The input signal of the algorithms is a raw reading from MEMS accelerometer (I2C interface based sensor), which contains samples from X, Y and Z axis (the axes are constituting orthogonal system). Data for simulation were grabbed online from S1/S2 boards in several scenarios and then saved for use in the Matlab environment. Experimental tests base on real-time computing by microcontroller mounted directly into the AI device. In both systems, sample time was set as 2 ms. At the system start, an average of several tens of samples is calculated, which is then treated as signal bias. Averaging is necessary to reduce the impact of the unavoidable measurement noise and signal drift, characteristic for MEMS [7].

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

At the beginning of the algorithm description, we assume M as measured data input vector of X, Y, Z accelerometer axes signal values:

$$M = [x, y, z] \quad (1)$$

and two equations that are solved on each sample time:

$$D_k = M_k - M_{k-1}, \quad (2)$$

$$S_k = M_k - M_0, \quad (3)$$

where: k is sample number, D_k – dynamic change of the acceleration, S_k – static change of the acceleration, M_0 – acceleration value at the start point. In the first attempt, D_k value and so S_k were computed as a difference square, which deprecates the low amplitude signals and makes the system less sensitive. D_k is an equivalent of jerk (time derivative of acceleration).

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

$$AD_k = AD_{k-1} + D_k \quad (4)$$

$$AXS_k = AXS_{k-1} + S_k \quad (5)$$

$$TN_k = TN_{k-1} + 1 \quad (6)$$

The above equations are computed only when the condition $D > LOL$ is fulfilled, where LOL is a constant algorithm parameters, responsible for separation of the desired signal from noise values.

Equations (4) and (5) 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 LOL value, will be also integrated). 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 (6) gives sensitivity to the signals of very low amplitude with a good separation from the noise (attempts to interfere in the lock of the door). Operation (6) is made only when condition $D > LOL$ (noise level is broken in the sample time period) is true.

The mechanism of subtraction is made 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:

$$AD_i = AD_{i-1} - ADD, \quad (7)$$

where: ADD is a constant varies on the $L2L$ value (level of alarm). $ADD = L2L/30$ (which means 15s time to reduce of accumulator AD to zero from the $L2L$ value).

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

$$AXS_i = AXS_{i-1} - AXSD, \quad (8)$$

where $AXSD = ADD$ in the analysed algorithms.

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

$$TN_i = TN_{i-1} - TND \quad (9)$$

$L1L$ value is the level of warning. There are two conditions that cause warnings: $AD > L1L$ and $TN > L0N$. Alarm is reached also in two ways: one condition $AD > L2L$ and the second way $AXS > L1XL$, where $L1XL = L1L/2$. Second condition is an equivalent of door opening detection which need to cause alarm state.

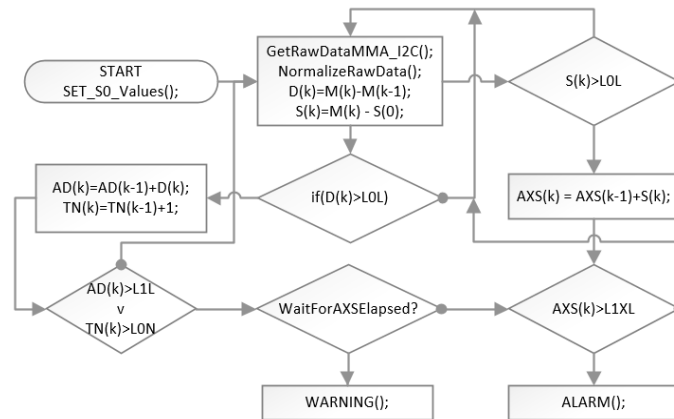


Fig. 5. Block diagram of main algorithm (incrementation of accumulators)

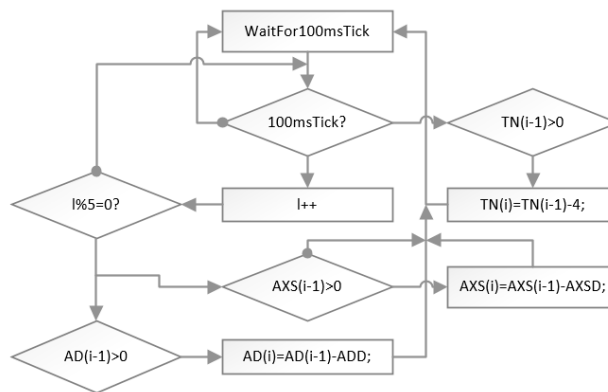


Fig. 6. Block diagram of main algorithm (decrementation of accumulators)

In the Figures 5 and 6 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 (from Figure 5). 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 engaged – 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 Figure 5. 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).

The parameters values used in the simulations and experiment are as follows: $L0L = 15$; $L1L = 40000$; $L0N = 320$; $L2L = 150000$; $L1XL = L1L/3$; $ADD = L2L/30$; $AXSD = ADD$. These values are set experimentally to fulfil the required effects on testing scenario.

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 amount 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.

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 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, difference between maximal and minimal sample value, discrete integral of the signal. All of the signals were calculated for each of the axes separately in the horizon 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. The determination of the bias values was done at the system startup for each accelerometer axis separately, by calculation of average of first 50 samples.

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

$$y_n^{(1)} = \frac{2}{1 + \exp\left(-2 \cdot \left(\sum_{i=1}^9 a_i^{(1)} x_i + b_n^{(1)}\right)\right)} - 1, \quad (10)$$

where $y_n^{(1)}$ 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:

$$y^{(2)} = \sum_{i=1}^{20} a_i^{(2)} y_i^{(1)} + b^{(2)} \quad (11)$$

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].

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 6 ms. The observed computational complexity is related with two fact. Firstly, the applied MCU does not possess floating-point coprocessor, therefore calculations of sigmoidal activation function appear 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 (Figures 2 and 3). 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 DDSF algorithm described in sub section 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

The data set are grouped by four different scenarios that typically occur in real conditions: no excitation (relative silence), rapidly opened door (Figure 7), subtle picking the door lock, strong knocking (Figure 8). These scenarios are quite clear (there is need to make more sophisticated tests in the future in larger scale), but shows that algorithms work well in many cases. 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 notice, but better when DDSF is used. Overall positive detection rate is a little better with DDSF (AI2) algorithm (0,83 to 0,67), but – as mentioned before – more advanced research and further evaluation are necessary.

Table. 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)

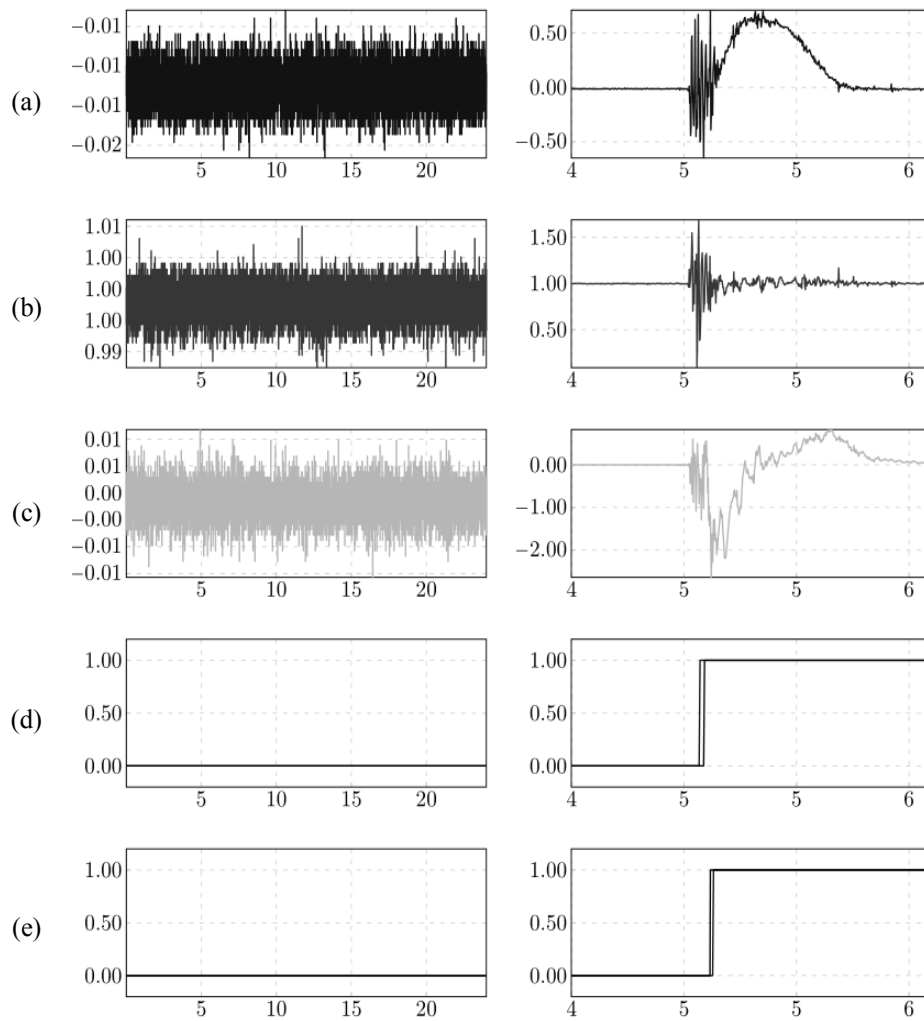


Fig. 7. Simulation and experimental algorithms execution results paired with source data: MEMS sensor signal of (a) X-axis, (b) Y-axis, (c) Z-axis, (d) DDSP and (e) neural classifier detection results (black – simulation, blue – experiment); left section scenario: no excitation (relative silence), right section: rapidly opened door

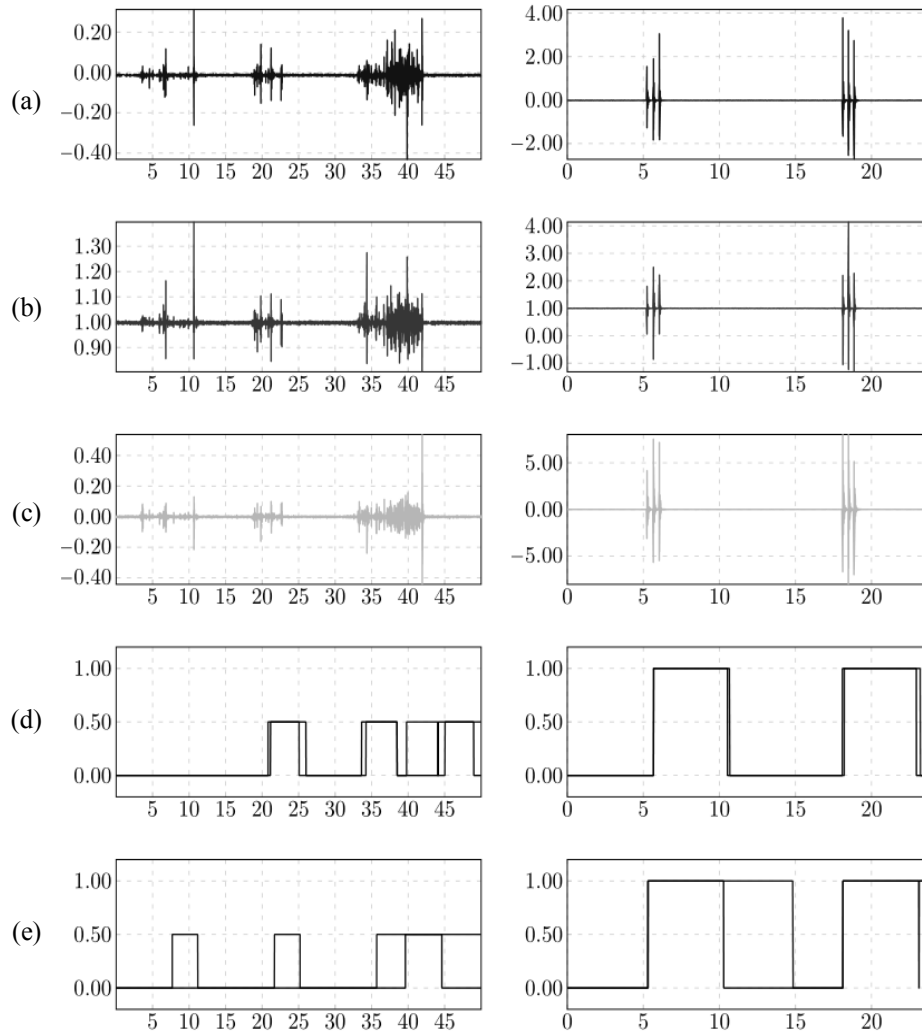


Fig. 8. Simulation and experimental algorithms execution results paired with source data: MEMS sensor signal of (a) X-axis, (b) Y-axis, (c) Z-axis, (d) DDSP and (e) neural classifier detection results (black – simulation, blue – experiment); left: picking the door lock, right: strong knocking

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 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 to implement.

REFERENCES

- [1] S. Finkbeiner, "MEMS for automotive and consumer electronics," in *Solid-State Device Research Conference (ESSDERC), 2013 Proceedings of the European*, 2013, pp. 9–14.
- [2] M. Perlmutter and L. Robin, "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] B. Fabiański, K. Nowopolski, and B. Wicher, "The System of Streaming and Analysis of Signals from MEMS Accelerometers," *Poznan University of Technology Academic Journals*, 2016.
- [5] S. Awasthi and A. Joshi, "MEMS accelerometer based system for motion analysis," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 762–767.
- [6] B. Fabianski, "Embedded System of Critical Information Management," *Poznan University of Technology Academic Journals*, pp. 143–149, 2013.
- [7] Y. Y. Chen, S.-J. Chang, C.-Y. Huang, and C. Y. Hsinag, "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] M. F. Møller, "Moller, M.F.: A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning. Neural Networks 6, 525-533," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, 1985.

(Received: 18. 02. 2016, revised: 5. 03. 2016)