DOI 10.15276/imms.v12.no4.269
УДК 681.518

Informatics and Mathematical Methods in Simulation
Vol. 12 (2022), No. 4, pp. 269-279

ALGORITHM FOR AUTOMATIC DAMAGE IDENTIFICATION USING LOW-COST STRUCTURAL HEALTH MONITORING SYSTEMS

A.V. Basko, O.A. Ponomarova, Y.O. Prokopchuk

Prydniprovska State Academy of Civil Engineering and Architecture, Chernyshevskoho str., 24a, Dnipro, 49600; Ukraine; basko.artem@pgasa.dp.ua; pricmech@ukr.net; itk3@ukr.net

The analysis of existing structural health monitoring (SHM) systems and damage identification algorithms showed that monitoring systems with low-cost elements are currently used rarely, and damage identification algorithms for such systems are practically non-existent. Based on a previous analysis of existing SHM, as well as further research on feature extraction, which is mainly carried out using already recorded information for data processing. In addition, for the application of algorithms for identification, detection and assessment of possible damage to the object of monitoring. Therefore, it can be concluded that the use of existing SHM systems has a number of disadvantages related to the complexity of implementation and adjustment, as well as economic impracticality. Some studies are aimed at reducing the cost of the sensor node by using a cheaper elemental base. But data from the entire sensor network is still transmitted to the main data collection server. All this requires significant computing power to process such a large amount of data. It should be noted that with the increase of the sensor network, the speed of data transmission in it decreases. The proposed study presents the development of a modern method of automatic damage identification with the possibility of automatic assessment of the level of damage. In addition, the developed algorithm is used to obtain results in real time. Also, the algorithm is conceptually aimed at low-cost SHM systems, with the possibility of local computation directly on the sensor node. This will make it possible to compress a large amount of data as much as possible and transmit only useful information about the state of the object. The article describes in detail the algorithm for automatic data processing in real time, and provides an implementation in the C++ programming language. The results of the experiment using the proposed algorithm to check the efficiency and reliability of the work are also presented. The developed method can be used as an alternative to traditional methods, especially for low-cost SHM systems.

Keywords: damage identification algorithm, dynamic characterization, low-cost SHM system, structural response, wireless sensor network, structural health monitoring.

Introduction

Modern research in the field of structural health monitoring (SHM), as well as actually working systems installed on various architectural objects, such as bridges, buildings, wind turbines, nuclear power plants, showing the seriousness of their application [1-3]. Due to the complexity of the structures, and as a result of the use of various architectural forms and materials, influence of the environment, the life of the object may decrease and endanger nature and man. Therefore, such systems help to understand how structures react under operating [4-5]. This allows to prevent the negative consequences, caused by destructions and apply preventive procedures for periodic inspection and repair of possible damage.

A lot of data processing algorithms presented in a considerable amount research can be: the least mean squares (LMS) algorithm, the recursive least squares (RLS) algorithm and the Kalman filtering (KF) algorithm [6]. Also, one of the popular ones is the use of the generalized least square (GLS) method [7], and an algorithm based on the principle of fuzzy logic [8].

One of the popular solutions in signal processing is the use of machine learning technologies. Some methods have already shown their advantages in tasks related to text recognition, image identification, and text translation.

Bayesian analysis and decision tree are complex and rarely used due to lack of flexibility and high requirement of processing power [9]. The use of such machine learning methods as Artificial Neural Network (ANN) and Support Vector Machine (SVM) is described [10].

Different algorithms of damage recognition based on machine learning technology was described in detail. In addition, a number of characteristics were identified that a sensitive element must have for a successful process of damage identification [11].

The construction of the Low-Cost Wireless sensor has been described and the choice of components in the study carefully justified. In addition, signal processing for damage identification was presented [12].

An approach to interpreting both static and dynamic data was presented in the study. The main emphasis was placed on identifying daily and seasonal conditions, as well as possible trends associated with critical failures. The considered approaches were tested on the data set from the "Two Towers" of Bologna [13].

Among the wide variety of algorithms that researchers offer, we can distinguish the main requirements for their implementation:

- The need for precise synchronization of data from all sensors
- It is necessary to have a data set in an intact state of the monitored object.
- The impossibility of obtaining the result of calculations in real time
- The need to post-process and analyze data in MATLAB or other specialized software, which requires a qualified specialist.
- A large number of sensor nodes leads to an overload of the sensor network and it may not have time to transmit data.
- Large amount of memory for the accumulation of measurements.

For modern sensor networks, it is necessary to build systems based on dynamic calculation, local filtering, data compression, damage thresholds, and all this in automatic mode. All this will not only simplify the identification process, but also eliminate the human factor, increase the speed of the system's response to disturbing influences. Obviously, this will save money on the design of the sensor node and sensor network, as well as reduce the demands on the computational properties of the sensor node.

Data compression is one of the important requirements in modern web-based automatic damage identification. Many different data compression methods have been proposed in the scientific community [14].

Content statement of the problem. The purpose of the work is to develop an algorithm for automatic identification of damage to improve safety buildings and structures using cheap and low-power sensor nodes. The essence of the proposed automatic identification is the automatic adjustment of the algorithm using step-by-step vibration assessment for signal processing.

In addition, the work proposes an approach that allows you to automatically determine the levels of possible damage for their identification in real time.

To achieve the set goal, the following tasks are set before the work:

- analyze innovative approaches and algorithms for damage identification;
- to develop a simple and effective automatic algorithm for damage identification;
- using the MVS (Microsoft Visual Studio) environment and the C++ programming language, implement the proposed damage identification algorithm;
- determine the requirements that a sensor node must meet to implement an automatic algorithm;

▪ perform an experiment using the proposed algorithm and demonstrate the effectiveness of the algorithm;

In addition, this algorithm is very convenient to use when working with ADC values, since it is not tied to the data scale. Also, integer data is more preferred for microcontrollers because math operations on them are faster than on floating point numbers.

Algorithm of data processing. The proposed algorithm belongs to unsupervised algorithms, has advantages related to the speed of calculations and does not require initial data and long-term collection of initial data. Comparing all machine learning algorithms, it can be noted that unsupervised learning algorithms will have lower accuracy compared to supervised learning.

It is assumed that the proposed algorithm will work locally on the sensor node, and this is also the novelty of the study. And only data about the damaged state is transmitted to the central server, where the server can localize the problem.

Before considering the principle of operation of the proposed algorithm, it is necessary to determine a number of minimum technical characteristics that the sensor node must meet:

- microcontroller based on 32-bit architecture or higher;
- flash memory size not less than 512 Kbytes;
- support for one of the wireless protocols Zigbee, Zigbee Pro or LoRaWAN;
- acceleration measurement sensor sensitivity ± 4g or less
- built-in ADC with a data resolution of at least 16 bits per range and a sampling frequency of at least 200 Hz.

The developed approach can be divided into several main stages that has been described below. On the first stage to get the value of the current step S_i between two values, as the difference between the current and the previous value, as shown in equation (1).

$$S_i = |x_i - x_{i-1}|, \tag{1}$$

where x_i – is the current value and x_{i-1} – is the previous value of vibration.

Secondly, we should evaluate how our steps change. To estimate changes of steps we should follow next conditions, if $S_{i-1} > S_i$ we should follow equation (2), otherwise (3).

$$SE_i = \frac{S_i}{S_{i-1}} \cdot G, \tag{2}$$

$$SE_i = \frac{S_{i-1}}{S_i} \cdot G, \tag{3}$$

where SE_i – current step evaluation, S_i – current step, S_{i-1} – previous step, G – smoothing coefficient, the initial value was set to 0.5.

The third stage is estimating the smoothing coefficient GE, allows you to automatically evaluate the work of the smoothing coefficient. The estimation is performed by the one-dimensional measure G_koeff, which is set as a constant (it is recommended to set value from 0.1 to 0.3). For more noisy signal lower value, and higher value otherwise. So, we need to evaluate every N times, so if $S_{i-1} \leq$ G_koeff then use equation (4) otherwise use equation (5).

$$Temp\_GE = Temp\_GE + 1 \tag{4}$$

$$Temp\_GE = Temp\_GE \tag{5}$$

The GE smoothing coefficient is estimated every N values. Where N can be chosen to be F_{adc}, where F_{adc} is the sampling rate of the data. To calculate an estimated value of the smoothing coefficient, the equation (6) is used. When N values are reached, the temporary counter Temp_GE must be reset to zero.

$$GE = \frac{Temp\_GE}{N} \tag{6}$$

One of the important part of the algorithm is correction of the smoothing coefficient G. When the estimated value GE was got, we can adjust smoothing coefficient value. First, we need to define a constant GE_Limit value, it is value to which will tend to our estimated coefficient GE.

GE_Limit is chosen in the range from 0.7 to 0.9. Than the closer the value is to 1, the gain factor will tend to smooth out very much.

First, should to check if GE = GE_Limit, then the smoothing coefficient G will not change, otherwise we will check the following condition. If (|GE_Limit-GE|>0.1), the smoothing coefficient will be corrected the smoothing coefficient will be corrected in accordance with equation (7), else with equation (8).

$$G = G - \left(0.1 \cdot \frac{GE\_Limit - GE}{|GE\_Limit - GE|}\right) \tag{7}$$

$$G = G - \left(0.01 \cdot \frac{GE\_Limit - GE}{|GE\_Limit - GE|}\right) \tag{8}$$

At the final stage, the corrected value can be calculated using equation (9). It should be noted that the algorithm needs to make some certain number of periods to self-correct the smoothing coefficient. The number of periods will depend on the form of the data signal, it usually takes less than 10 periods.

$$XF_i = x_i \cdot SE_i + XF_{i-1} \cdot (1 - SE_i) \tag{9}$$

Automatic levels of damage identification. One of the advantages of the proposed method is the ability to automatically define several levels of damage identification. It should be noted that these levels will be located between the average and maximum values. In addition, damage levels are automatically retrained with changes in the input data.

For example, the first level will be more sensitive to small changes, and subsequent levels are more likely to detect critical damage.

In most of the presented systems, frequently some fixed threshold is chosen, which is greater than the arithmetic means by 5%. In the presented algorithm, the threshold metric is multidimensional and depends on the nature and rate of data change.

Further, the calculation of three levels of damage identification will be carried out. To start automatic learning damage levels, first of all need to wait a certain time. This must be done in order to train the coefficients of smoothing algorithm. Depending on the selected sample rate F_{adc}, this time might be chosen from 1 second to 1 minute. To initialize the first value of each of the levels, the algorithm chooses current value from the received data set, as shown in the equations (10-12), for each of the levels, respectively.

$$L1_i = x_i \tag{10}$$

$$L2_i = x_i \tag{11}$$

$$L3_i = x_i \tag{12}$$

Since the levels are located between the average and maximum values, so there is no any reasons to make many levels, as they will tend to the border of the maximum values.

The first level L1 is calculated relative to the average value XA of the received data. Thus, the new value updates the average value. To do this, we apply a filter that smoothes the data quite hard, as shown in the equation (13).

$$XA_i = x_i \cdot 0.01 + XA_{i-1} \cdot (1 - 0.01) \tag{13}$$

For the first level, the condition $x_{i-1} > XA_i$ must be met, if it is not met, then go to equation (15). Then we get the temporary value of the level, if $x_i<x_{i-1}<x_{i-2}$ then use equation (14), otherwise equation (15).

$$L1_{temp_i} = x_{i-1} \tag{14}$$

$$L1\_temp_i = L1\_temp_{i-1} \tag{15}$$

When the new temporary value of the level is received, then we can get the value of the current level using the following equation (16):

$$L1_i = L1\_temp_i \cdot k + L1_{i-1} \cdot (1-k) \qquad (16)$$

where k is the smoothing factor, it affects the level response speed, the choice of which depends on the sampling rate and data noise. To begin with, it is enough to choose $k=1/F_{adc}$.

The second level is calculated relative to the first level. For the second level, the condition $x_{i-1} > L1_i$ must be met, if it is not met, then go to equation (18). Then we get the temporary value of the level, if $x_i<x_{i-1}<x_{i-2}$ then use equation (17), otherwise equation (18).

$$L2_{temp_i} = x_{i-1} \qquad (17)$$

$$L2\_temp_i = L2\_temp_{i-1} \qquad (18)$$

When the new temporary value of the level is received, then we can get the value of the current level using the following equation (19):

$$L2_i = L2\_temp_i \cdot k + L2_{i-1} \cdot (1-k) \qquad (19)$$

Similarly, to the second level, the third level is calculated relative to the second level. For the second level, the condition $x_{i-1} > L2_i$ must be met, if it is not met, then we go to equation (21). Then we get the temporary value of the level, if $x_i<x_{i-1}<x_{i-2}$ then we use equation (20), otherwise equation (21).

$$L3_{temp_i} = x_{i-1} \qquad (20)$$

$$L3\_temp_i = L3\_temp_{i-1} \qquad (21)$$

When the new temporary value of the level is received, then we can get the value of the current level using the following equation (22):

$$L3_i = L3\_temp_i \cdot k + L3_{i-1} \cdot (1-k) \qquad (22)$$

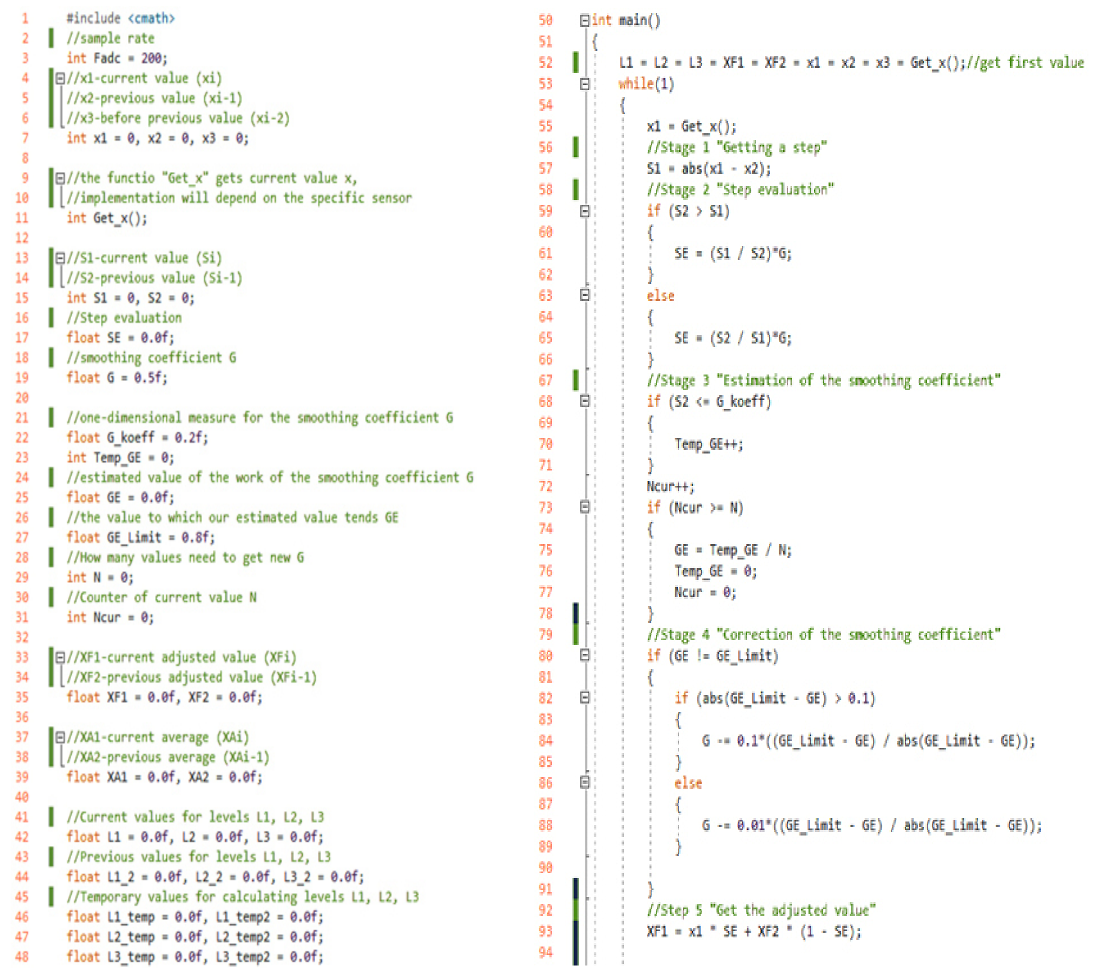
Thus, for multilevel damage identification, it is sufficient to evaluate the excess of one of the levels by comparing the corrected value of XF_i and the levels $L1_i$, $L2_i$ and $L3_i$, which is displayed in equations (23-25).

$$XF_i > L1_i \qquad (23)$$

$$XF_i > L2_i \qquad (24)$$

$$XF_i > L3_i \qquad (25)$$

Program code of proposed algorithm. For ease of understanding the operation of the calculation algorithm, the code was written in the C ++ programming language, which is shown in Fig. 1-2. The code below shows the implementation of calculations for data coming from one axis of the accelerometer. For complex structures where three-axis accelerometers are used, and maybe even more than one accelerometer, it will be rational to use an object-oriented programming approach.

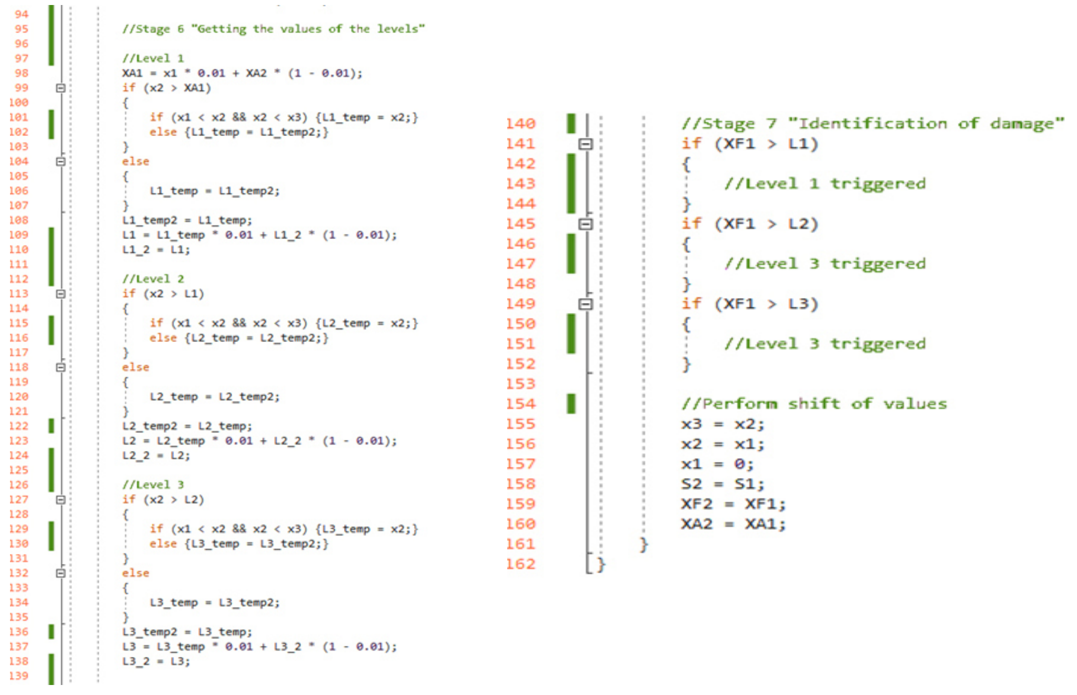


```
#include <cmath>
//sample rate
int Fadc = 200;
//x1-current value (xi)
//x2-previous value (xi-1)
//x3-before previous value (xi-2)
int x1 = 0, x2 = 0, x3 = 0;

//the functio "Get_x" gets current value x,
//implementation will depend on the specific sensor
int Get_x();

//S1-current value (Si)
//S2-previous value (Si-1)
int S1 = 0, S2 = 0;
//Step evaluation
float SE = 0.0f;
//smoothing coefficient G
float G = 0.5f;

//one-dimensional measure for the smoothing coefficient G
float G_koeff = 0.2f;
int Temp_GE = 0;
//estimated value of the work of the smoothing coefficient G
float GE = 0.0f;
//the value to which our estimated value tends GE
float GE_Limit = 0.8f;
//How many values need to get new G
int N = 0;
//Counter of current value N
int Ncur = 0;

//XF1-current adjusted value (XFi)
//XF2-previous adjusted value (XFi-1)
float XF1 = 0.0f, XF2 = 0.0f;

//XA1-current average (XAi)
//XA2-previous average (XAi-1)
float XA1 = 0.0f, XA2 = 0.0f;

//Current values for levels L1, L2, L3
float L1 = 0.0f, L2 = 0.0f, L3 = 0.0f;
//Previous values for levels L1, L2, L3
float L1_2 = 0.0f, L2_2 = 0.0f, L3_2 = 0.0f;
//Temporary values for calculating levels L1, L2, L3
float L1_temp = 0.0f, L1_temp2 = 0.0f;
float L2_temp = 0.0f, L2_temp2 = 0.0f;
float L3_temp = 0.0f, L3_temp2 = 0.0f;
```

```
int main()
{
    L1 = L2 = L3 = XF1 = XF2 = x1 = x2 = x3 = Get_x();//get first value
    while(1)
    {
        x1 = Get_x();
        //Stage 1 "Getting a step"
        S1 = abs(x1 - x2);
        //Stage 2 "Step evaluation"
        if (S2 > S1)
        {
            SE = (S1 / S2)*G;
        }
        else
        {
            SE = (S2 / S1)*G;
        }
        //Stage 3 "Estimation of the smoothing coefficient"
        if (S2 <= G_koeff)
        {
            Temp_GE++;
        }
        Ncur++;
        if (Ncur >= N)
        {
            GE = Temp_GE / N;
            Temp_GE = 0;
            Ncur = 0;
        }
        //Stage 4 "Correction of the smoothing coefficient"
        if (GE != GE_Limit)
        {
            if (abs(GE_Limit - GE) > 0.1)
            {
                G -= 0.1*((GE_Limit - GE) / abs(GE_Limit - GE));
            }
            else
            {
                G -= 0.01*((GE_Limit - GE) / abs(GE_Limit - GE));
            }

        }
        //Step 5 "Get the adjusted value"
        XF1 = x1 * SE + XF2 * (1 - SE);
```

a b

Fig. 1. Variable initialization (a) and main block of program (b)



```
        //Stage 6 "Getting the values of the levels"

        //Level 1
        XA1 = x1 * 0.01 + XA2 * (1 - 0.01);
        if (x2 > XA1)
        {
            if (x1 < x2 && x2 < x3) {L1_temp = x2;}
            else {L1_temp = L1_temp2;}
        }
        else
        {
            L1_temp = L1_temp2;
        }
        L1_temp2 = L1_temp;
        L1 = L1_temp * 0.01 + L1_2 * (1 - 0.01);
        L1_2 = L1;

        //Level 2
        if (x2 > L1)
        {
            if (x1 < x2 && x2 < x3) {L2_temp = x2;}
            else {L2_temp = L2_temp2;}
        }
        else
        {
            L2_temp = L2_temp2;
        }
        L2_temp2 = L2_temp;
        L2 = L2_temp * 0.01 + L2_2 * (1 - 0.01);
        L2_2 = L2;

        //Level 3
        if (x2 > L2)
        {
            if (x1 < x2 && x2 < x3) {L3_temp = x2;}
            else {L3_temp = L3_temp2;}
        }
        else
        {
            L3_temp = L3_temp2;
        }
        L3_temp2 = L3_temp;
        L3 = L3_temp * 0.01 + L3_2 * (1 - 0.01);
        L3_2 = L3;
```

```
        //Stage 7 "Identification of damage"
        if (XF1 > L1)
        {
            //Level 1 triggered
        }
        if (XF1 > L2)
        {
            //Level 3 triggered
        }
        if (XF1 > L3)
        {
            //Level 3 triggered
        }

        //Perform shift of values
        x3 = x2;
        x2 = x1;
        x1 = 0;
        S2 = S1;
        XF2 = XF1;
        XA2 = XA1;
    }
}
```

a b

Fig. 2. Calculation of damage identification levels (a) and damage identification (b)

Experiments and results. In order to test and evaluate the quality of the proposed automatic damage identification algorithm, a set of acceleration data from the Hardanger Bridge [15] has been used. In this case, the data set was selected with a sampling rate of 200 Hz. It is important that there are no hardware filters for this sample. The data has the following time line from 0 to 50 seconds, the data is taken from the bridge in a calm state, and from 50 to 80 seconds, a transition is made to the storm wind data, which is shown in Fig. 3.

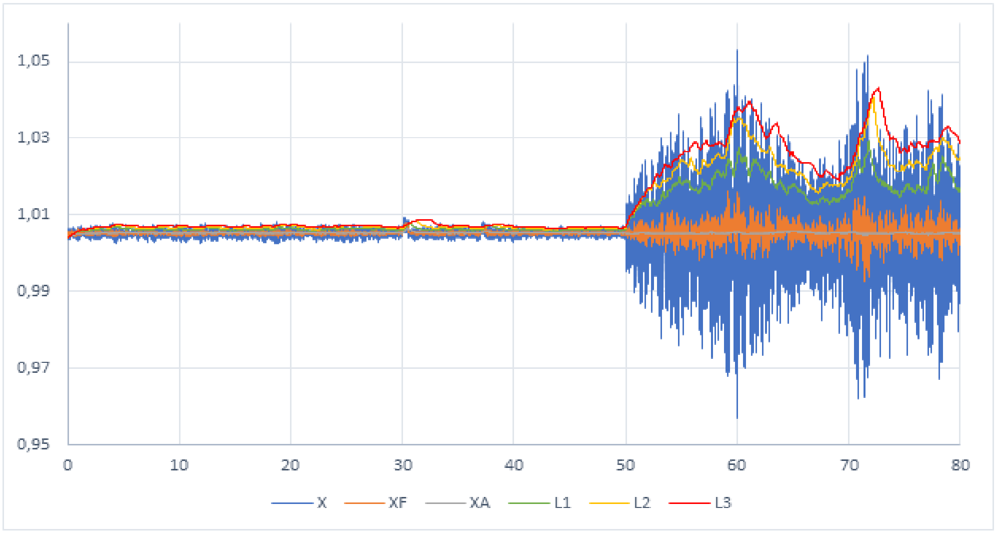


Fig. 3. Full set of training data

To understand the operation of the proposed algorithm, the test data will be displayed on a larger scale, while maintaining the time scale as shown in Fig. (4-7). In the first period of time, when the algorithm is not trained, we can create an increasing graph, which indicates the learning process, which can be seen in Fig. 4.

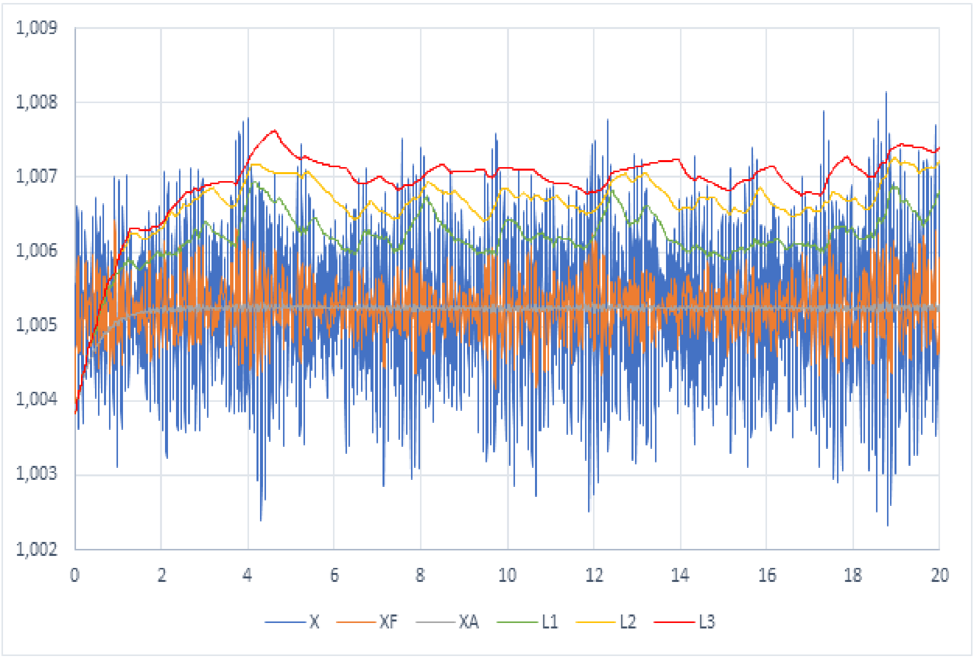


Fig. 4. Learning process of data

In Fig. 5, small disturbing signals are artificially created in order to show the sensitivity of the algorithm even to small signal changes with a pulse time of 500ms.

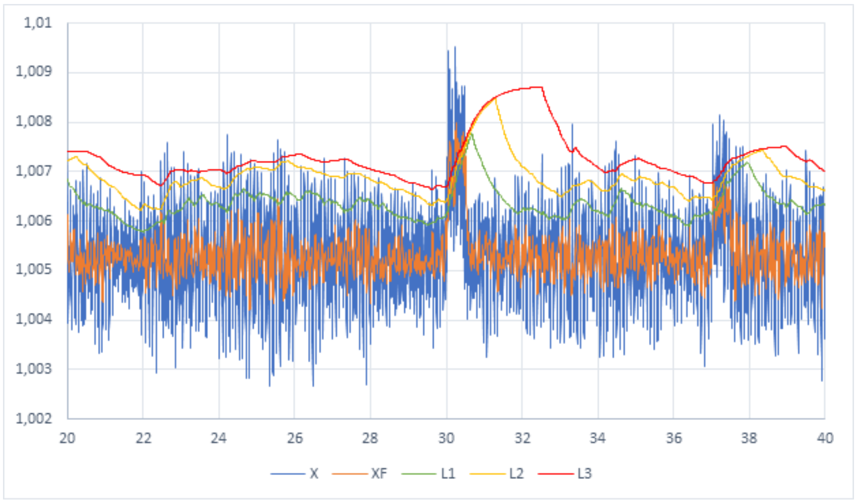


Fig. 5. Reaction on artificial disturbing signals

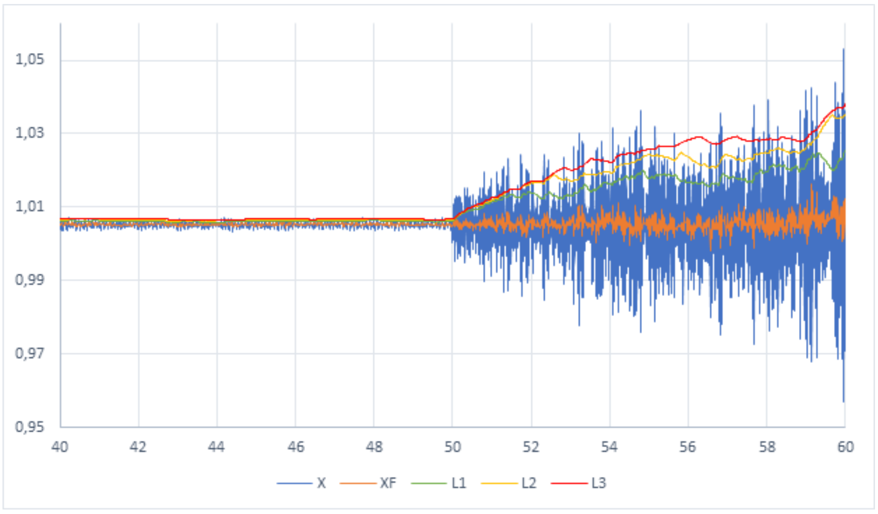


Fig. 6. Transition from quiet to stormy condition

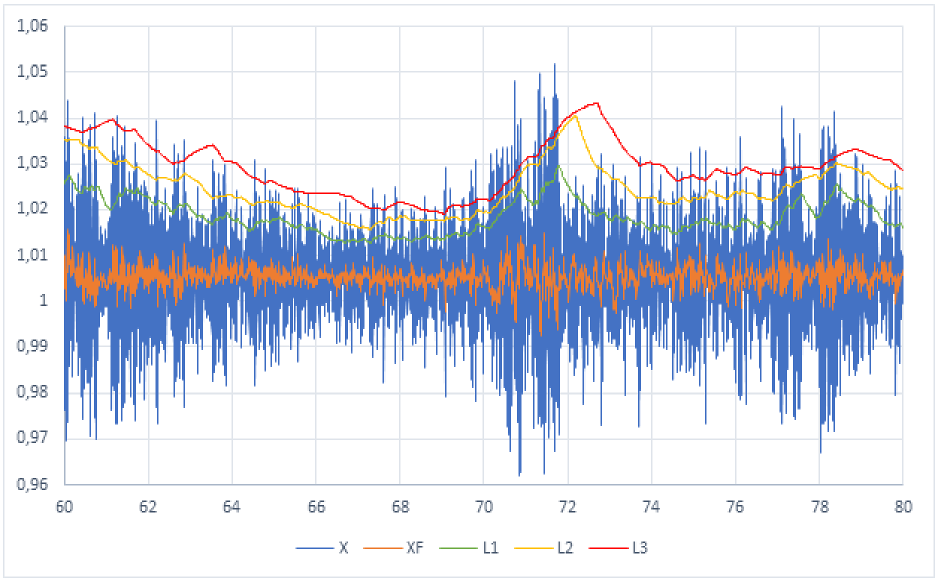


Fig. 7. Data from the bridge in a storm

The key feature of the proposed algorithm is the automatic calculation of damage identification levels. As can be seen from Fig. 8, the filtered data overlaps with all detected identification levels, which may indicate a high probability of critical damage.

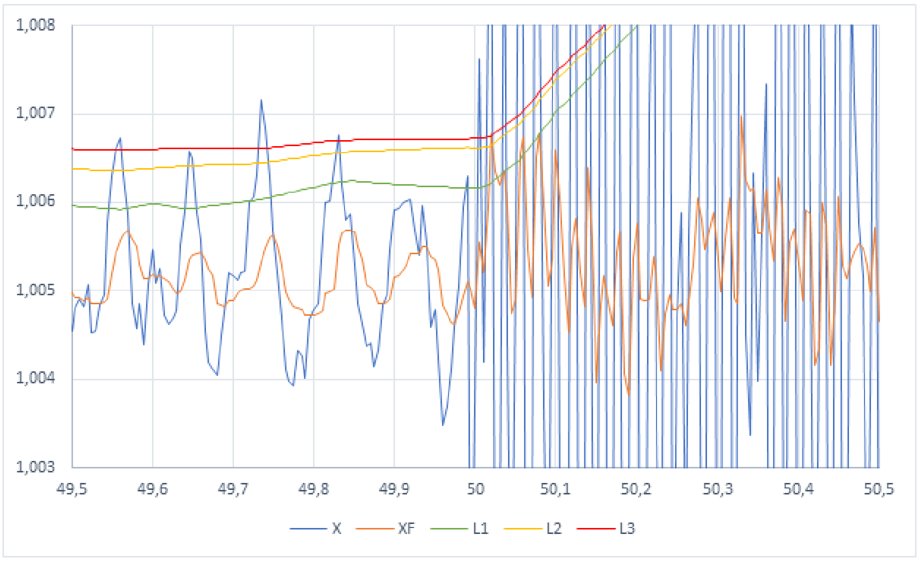


Fig. 8. Damage identification stage

Conclusion

An innovative solution in SHM, certainly was the use of wireless technologies. But existing systems are doing their best to improve by reducing their cost in order to popularize such method of monitoring. A decrease a cost of sensors by using a cheaper element base imposes restrictions on the use of data processing algorithms.

In the present study, a brief overview of the most commonly used data processing methods was made. The algorithm proposed in the study is a first step in the field of automatic identification of pressures. The described method assumes that the sensor will be automatically calibrated and adjusted without the participation of a trained specialist and a person.

Since this algorithm is the first attempt in the field of automatic identification, the next stages of development will be aimed at applying the algorithm to real systems and its subsequent improvement.

References

1. Rice J. A., Mechitov K., Sim S. H., Nagayama T., Jang S., Kim R., Spencer B. F. J., Agha G., Fujino Y. Flexible smart sensor framework for autonomous structural health monitoring. *Smart Structures and Systems*. 2010. Vol. 6. P. 423–438, DOI: https://www.doi.org/10.12989/sss.2010.6.5_6.423
2. Araujo A., García-Palacios J., Blesa J., Tirado-Andrés F., Romero E., Samartín A., Nieto-Taladriz O. Wireless measurement system for structural health monitoring with high time-synchronization accuracy. *IEEE Transactions on Instrumentation and Measurement*. 2012. Vol. 61, Issue. 3. P. 801-810. DOI: https://www.doi.org/10.1109/TIM.2011.2170889
3. Fu Y., Hoang T., Mechitov K., Kim J., Zhang D., Spencer B. Sudden event monitoring of civil infrastructure using demand-based wireless smart sensors. *Sensors*. 2018. Vol. 18, Issue. 12. P. 1-17. DOI: https://www.doi.org/10.3390/s18124480
4. Hailing F., Khodaei Z.S., Ferri Aliabadi M.H. An event-triggered energy-efficient wireless structural health monitoring system for impact detection in composite airframes. *IEEE Internet of Things Journal*. 2019. Vol. 6, Issue. 1. P. 1183-1192. DOI: https://www.doi.org/10.1109/JIOT.2018.2867722

5. Wang P., Yan Y., Gui Y.T., Bouzid O., Ding Z. Investigation of wireless sensor networks for structural health monitoring. *Journal of Sensors.* 2012. Vol. 2012. P. 1-7. DOI: https://www.doi.org/10.1155/2012/156329
6. Kluga A., Kluga J. Dynamic Data Processing with Kaiman Filter. *Elektronika Ir Elektrotechnika*. 2011. Vol. 111, Issue. 5. P. 33–36. DOI: https://doi.org/10.5755/j01.eee.111.5.351
7. Yang Z., Gu Z., Zhang W. Dynamic data processing method based on generalized least square method. *Journal of Physics: Conference Series*. 2021. Vol. 2108, Issue. 1. P. 1–6. DOI: https://doi.org/10.1088/1742-6596/2108/1/012057
8. Gorski J., Dziendzikowski M., Dworakowski Z. Recommendation System for Signal Processing in SHM. *Artificial Intelligence and Soft Computing.* 2021. Vol. 12854. P. 328–337. DOI: https://doi.org/10.1007/978-3-030-87986-0_29
9. Gordan M., Razak H.A., Ismail Z., Ghaedi K. Recent Developments in Damage Identification of Structures Using Data Mining. *Latin American Journal of Solids and Structures.* 2017. Vol. 14, Issue. 13. P. 2373–2401. DOI: https://doi.org/10.1590/1679-78254378
10. Finotti R. P., Cury A. A., Barbosa F. D. S. An SHM approach using machine learning and statistical indicators extracted from raw dynamic measurements. *Latin American Journal of Solids and Structures*. 2019. Vol. 16, Issue. 2. P. 1–17 DOI: https://doi.org/10.1590/1679-78254942
11. Figueiredo E., Santos A. Machine Learning Algorithms for Damage Detection. *Vibration-Based Techniques for Damage Detection and Localization in Engineering Structures*. 2018. P. 1–39. DOI: https://doi.org/10.1142/9781786344977_0001
12. Caballero-Russi D., Ortiz A. R., Guzman A., Canchila C. Design and Validation of a Low-Cost Structural Health Monitoring System for Dynamic Characterization of Structures. *Applied Sciences.* 2022. Vol. 12, Issue. 6. P. 1–28. DOI: https://doi.org/10.3390/app12062807
13. Baraccani S., Palermo M., Azzara R. M., Gasparini G., Silvestri S., Trombetti T. Structural Interpretation of Data from Static and Dynamic Structural Health Monitoring of Monumental Buildings. *Key Engineering Materials*. 2017. Vol. 747. P. 431–439. DOI: https://doi.org/10.4028/www.scientific.net/kem.747.431
14. Aime J. O. O., Jean P. N., Guy-Germain A., Pierre E. Compression of Vibration Data by the Walsh-Hadamard Transform. *Journal of Engineering and Applied Sciences*. 2020. Vol. 15, Issue. 10. P. 2256–2260.
15. Fenerci A., Kvale K., Petersen Y., Ronnquist A., Oiseth O. Data Set from Long-Term Wind and Acceleration Monitoring of the Hardanger Bridge. *Journal of Structural Engineering*. 2021. Vol. 147, Issue. 5. P. 1–13. DOI: https://doi.org/10.1061/(asce)st.1943-541x.0002997

АЛГОРИТМ АВТОМАТИЧНОЇ ІДЕНТИФІКАЦІЇ ПОШКОДЖЕНЬ З ВИКОРИСТАННЯМ НЕДОРОГИХ СИСТЕМ МОНІТОРИНГУ СТАНУ КОНСТРУКЦІЙ

А.В. Басько, О.А. Пономарьова, Ю.О. Прокопчук

Придніпровська державна академія будівництва та архітектури, Chernyshevskoho str., 24a, Dnipro, 49600; Ukraine; basko.artem@pgasa.dp.ua; pricmech@ukr.net; itk3@ukr.net

Аналіз існуючих систем моніторингу стану конструкцій (SHM) та алгоритмів ідентифікації пошкоджень показав, що системи моніторингу з використанням маловартісних елементів на сьогодні використовується дуже мало, а алгоритми ідентифікації пошкоджень для таких систем практично відсутні. Ґрунтуючись на першочерговому аналізі існуючих SHM, а також на подальших дослідженнях щодо виділення ознак, які в основному здійснюється з використанням уже записаної інформації для обробки даних. Крім того для застосування алгоритмів ідентифікації, виявлення та оцінки можливих збитків об’єкту моніторинг. Отже можна зробити висновок, що використання існуючих систем SHM, має ряд недоліків пов'язаних зі складністю імплементації та налаштування, а також з економічною недоцільністю. Деякі дослідження спрямовані на зниження вартості сенсорного вузла за рахунок використання більш дешевої елементної бази. Але дані з усієї сенсорної мережі все одно передаються на головний сервер збору даних. Все це вимагає значної обчислювальної потужності для обробки такої великої кількості даних. Слід зауважити, що зі збільшенням сенсорної мережі швидкість передачі даних у ній знижується. Запропоноване дослідження представляє розробку сучасного методу автоматичної ідентифікації пошкоджень із можливістю автоматичної оцінки рівня пошкоджень. Крім того, розроблений алгоритм використовується для отримання результатів у реальному часі. Також алгоритм концептуально націлений на недорогі системи SHM, з можливістю локального обчислення безпосередньо на сенсорному вузлі. Це дозволить максимально стиснути великий об’єм даних і передавати лише корисну інформацію про стан об’єкта. У статті детально описано алгоритм автоматичної обробки даних в режимі реального часу, наведено реалізацію на мові програмування C++. Також представлені результати експерименту із застосуванням запропонованого алгоритму для перевірки ефективності та надійності роботи. Розроблений метод може бути використаний як альтернатива традиційним методам, особливо для недорогих систем SHM.

Ключові слова: алгоритм ідентифікації пошкоджень, динамічна характеристика, недорогі системи SHM, структурна реакція, бездротова сенсорна мережа, структурний моніторинг стану.