STRUCTURAL HEALTH MONITORING FOR DANGEROUS VIBRATIONS FOR TOWERS IN SMALL CAPACITY WIND TURBINES

FLORIN GABRIEL POPESCU[1], RAZVAN SLUSARIUC[2], FLORIN MUREȘAN-GRECU[3], SEBASTIAN DANIEL ROȘCA[4]

Abstract: Vibration monitoring is an essential element in assessing the structural health of modern engineering systems. This paper presents a methodology for acquiring and processing vibration signals using MEMS sensors integrated into a monitoring system designed for structures subjected to dynamic loads. The measured signals are processed by digital filtering, in particular using a Butterworth filter, to reduce noise and highlight the relevant spectral components. The results obtained demonstrate the system's ability to identify variations in vibration amplitude and frequency, correlated with the occurrence of non-compliant operating conditions. The proposed methodology provides support for the implementation of predictive maintenance strategies and contributes to increasing the safety and reliability of monitored structures.

Key words: renewables, wind turbine, vibrations, sensor, damage.

1. INTRODUCTION

The oscillations of a wind turbine tower, known in the literature as **"tower sway"**, represent the dynamic phenomenon whereby the tower structure undergoes lateral and longitudinal displacements under the action of aerodynamic, gravitational and mechanical loads generated during operation. These oscillations are characteristic of a flexible system subjected to time-varying excitations and can be described by classical models of vibrations of structures with low rigidity [1], [3], [7], [10].

From a mechanical point of view, the wind turbine tower behaves like a **beam embedded at the base**, with distributed mass and a system of rigidities that depend on its geometry, material properties and rotor inertia. The oscillations are the result of the complex interaction between:

1. **The aerodynamic loads of the blades**, predominant at frequencies *1P* (one blade pass per rotation) and *3P* (the interaction of the three blades);

[1] *Ph.D., Associate Prof. Eng., University of Petroșani, floringabriel82@yahoo.com*
[2] *Ph.D., Lecturer. Eng., University of Petroșani, razvansir@yahoo.com*
[3] *Ph.D. Student Eng., University of Petroșani, flomavon2002@yahoo.com*
[4] *Ph.D., Assistant Prof. Eng., University of Petroșani, sebastianrosca91@gmail.com*

2. **Centrifugal and gyroscopic forces**, which introduce additional components to changes in wind direction or rotor speed variations;
3. **The interaction between the tower, nacelle and rotor**, which form a dynamically coupled system;
4. **Atmospheric turbulence**, which causes random and nonlinear excitations.

The tower sway phenomenon occurs predominantly at low frequencies (generally between **0.1 and 1 Hz**, depending on the height and stiffness of the tower), these frequencies being specific to the fundamental vibration modes of the structure [21]. The magnitude of the oscillations depends on the wind speed, the mass-stiffness ratio, and the mass of the rotor at the top of the tower, which acts as an amplifier of dynamic effects [4], [14], [17], [20], [22].

Monitoring and controlling tower oscillations is essential for:

- **assessing the** long-term **structural integrity** of the turbine,
- **preventing resonance phenomena**, which can induce dangerous increases in vibration amplitude,
- **optimising turbine control strategies** (pitch control, yaw control, load reduction),
- **early detection of structural defects** such as material fatigue, cracks or weld degradation.

2. MEMS ACCELEROMETERS

MEMS accelerometers are motion transducers in a single IC device package. The typical construction uses a pair of capacitors with a silicon micromass with metal plates in the middle. Very thin regions of silicon suspend the mass in the middle. Changes in the position of the mass result in changes in the capacitance of the device, which translate into a voltage signal proportional to the acceleration of the suspended mass [2], [6], [9], [13].

MEMS devices require a power supply to operate, and some MEMS accelerometers have a built-in digitiser to eliminate unnecessary noise and the need to match sensors and recorders [19]. As shown in Figure 1, the frequency response of a MEMS accelerometer is like a low-pass filter with a limit at the resonance frequency.

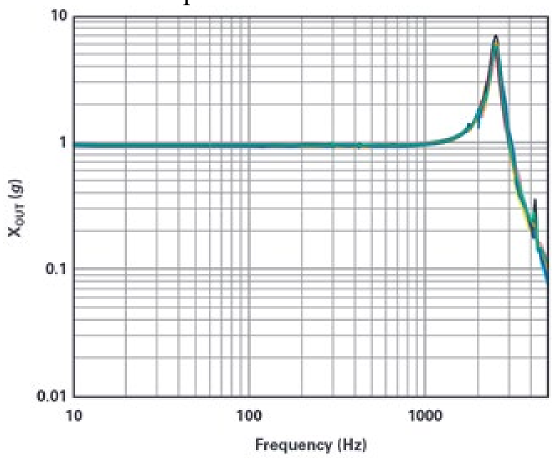

Fig.1. Frequency response of a MEMS accelerometer (ADXL354) on its X-axis.

Due to compensation displacements, MEMS accelerometers perform better at higher frequencies up to their resonance frequency.

3. THE ADXL 345 SENSOR

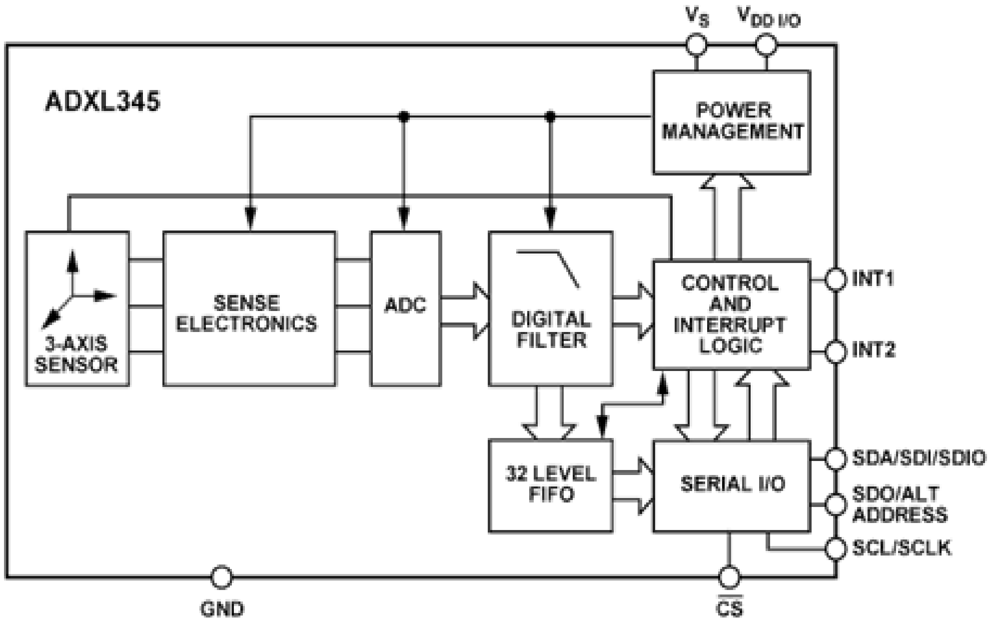

Fig.2. Block diagram of the ADXL 345 sensor chosen for the project

The ADXL335 sensor is a three-axis (X, Y and Z) analogue accelerometer capable of measuring accelerations in a typical range of ±3 g, making it suitable for vibration monitoring applications in structures and equipment. Its integration with an Arduino platform allows for the creation of a simple, flexible and low-cost data acquisition system [23], [26]. The analogue signals provided by the three outputs of the sensor are connected to the analogue inputs of the Arduino board, where they are converted into digital values using the internal analogue-to-digital converter (ADC).

For vibration monitoring, the acquired data is sampled at a frequency high enough to capture the relevant dynamic components and is then processed using digital filtering algorithms, such as the Butterworth filter, to reduce noise and eliminate unwanted components. The processed values can be transmitted to a computer or stored locally for spectral analysis and identification of dominant frequencies [5], [8], [12], [16].

The main advantage of this solution is its simplicity of implementation and the possibility of expansion with additional modules (wireless communication, SD card storage), which makes it suitable for continuous monitoring and preventive diagnostics of systems subject to vibrations [11], [15], [21].

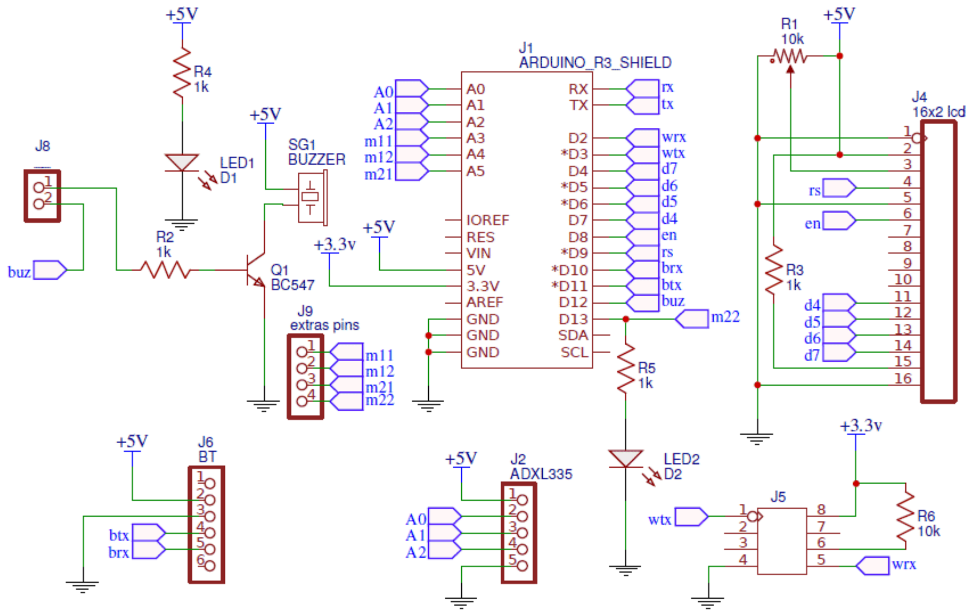

Fig.3. Electronic circuit of the earthquake detection system using the ADXL 345 MEMS sensor

3.1. Source code used in the microcontroller

After connecting the microcontroller to the computer, upload the following code:

```
//   - X-axis (Analogue Input 0)
//   - Y-axis (Analogue Input 1)
//   - Z-axis (Analogue Input 2)
//   - Vcc (+5V)
//   - GND (GND)

const int xPin = 0; // X-axis pin
const int yPin = 1; // Y-axis pin
const int zPin = 2; // Z-axis pin

void setup() {
           Initialise the serial port with a baud rate of 9600.
  Serial.begin(9600);
}

void loop() {
           Read the analogue inputs from the MEMS sensor.
  int xValue = analogRead(xPin);
  int yValue = analogueRead(yPin);
  int zValue = analogRead(zPin);
```

Convert the analogue input values to G-forces.

```
  float xg = (xValue * 5.0 / 1023.0 - 1.5) / 0.3;
  float yg = (yValue * 5.0 / 1023.0 - 1.5) / 0.3;
  float zg = (zValue * 5.0 / 1023.0 - 1.5) / 0.3;
```

The forces are displayed on the screen via the serial port.

```
  Serial.print("X-axis: ");
  Serial.print(xg);
  Serial.print(" g, Y-axis: ");
  Serial.print(yg);
  Serial.print(" g, Z-axis: ");
  Serial.print(zg);
  Serial.println(" g");
```

A delay of 100 milliseconds before reading the sensor again.

```
  delay(100);
}
```

This code reads the analogue input values from the X, Y, and Z pins of the ADXL335 sensor, converts them to g-force values, and then prints them to the Arduino IDE serial monitor. The analogueRead() function reads the analogue input values, and the conversion formula is taken from the ADXL335 sensor datasheet.

The Serial.print() function is used to print the g-force values to the serial monitor, and the delay() function is used to pause the program for 100 milliseconds before reading the sensor again [24], [25].

3.2. Real-time graph obtained from sensor data

To create the graph of the forces involved, the following code is used:

```
ADXL335 Accelerometer Sensor  – Real-time graph example
//   - X-axis (Analogue Input 0)
//   - Y-axis (Analogue Input 1)
//   - Z-axis (Analogue Input 2)
//   - Vcc (+5V)
//   - GND (GND)

const int xPin = 0; // X-axis pin
const int yPin = 1; // Y-axis pin
const int zPin = 2; // Z-axis pin

void setup() {
  // Initialise serial communication
  Serial.begin(9600);
}
```

```
void loop() {
  // Read the analogue input values from the ADXL335 sensor
  int xValue = analogRead(xPin);
  int yValue = analogueRead(yPin);
  int zValue = analogueRead(zPin);

  // Convert the analogue input values to g-force (using the formula from the datasheet)
  float xg = (xValue * 5.0 / 1023.0 - 1.5) / 0.3;
  float yg = (yValue * 5.0 / 1023.0 - 1.5) / 0.3;
  float zg = (zValue * 5.0 / 1023.0 - 1.5) / 0.3;
```

The G-force values are printed on the Serial Monitor port.

```
  Serial.print(xg);
  Serial.print(",");
  Serial.print(yg);
  Serial.print(",");
  Serial.println(zg);
```

Plotting the G-forces graph on the Serial Plotter.

```
  Serial.flush(); // Wait for the Serial Plotter to catch up
}
```

This code is similar to the previous one, but instead of printing the g-force values on the serial monitor, it sends them to the serial plotter. The Serial.print() function is used to send the g-force values to the serial plotter in the format "x,y,z", where x, y, and z are the g-force values for the X, Y, and Z axes, respectively.

The Serial.flush() function is used to wait for the Serial Plotter to catch up before sending the next set of values. To plot the graph in the Serial Plotter, select "Serial Plotter" from the "Tools" menu in the Arduino IDE or press "Ctrl+Shift+L" on your keyboard.

3.3. Transmitting and storing data in Matlab

To obtain serial data from Arduino to Matlab, follow these steps:

- Connect the Arduino board to your computer using a USB cable.
- Open the Arduino IDE platform and upload the code to the board you want to use to send data to Matlab. Open Matlab and create a new script or function file.
- Use the "serial" command to create a serial object in Matlab that will communicate with the Arduino board. For example:

```
s = serial('COM3', 'BaudRate', 9600);
```

This creates a serial object called "s" that will communicate with the Arduino board on COM3 at a baud rate of 9600.

The serial port is opened by calling the "fopen" function with the serial object as the input argument:

```
fopen(e);
```

The data is read from the serial port using the "fscanf" function. For example:

```
data = fscanf(s, '%f');
```

This reads a floating point number from the serial port and stores it in the variable "data".
Close the serial port when you are finished by calling the "fclose" function:

```
fclose(e);
```

You can also use the "fread" function to read raw binary data from the serial port if your Arduino code sends binary data.
The baud rate specified in Matlab must match the baud rate specified in the Arduino code, otherwise the data will not be received correctly [12], [18].

3.4. Data sent in real time in Matlab

To represent real-time data from an accelerometer sensor connected to an Arduino board in Matlab, follow these steps:
Connect the Arduino to your computer using a USB cable and upload the code that reads the data from the accelerometer sensor and sends it via serial communication.
Open Matlab and create a new script or function file. Use the "serial" command to create a serial object in Matlab that will communicate with the Arduino board. For example:

```
s = serial('COM3', 'BaudRate', 9600);
```

This creates a serial object called "s" that will communicate with the Arduino board on COM3 at a baud rate of 9600.

Open the serial port by calling the "fopen" function with the serial object as the input argument:

```
fopen(s);
```

Create an empty array to store the data:

```
data = [];
```

Create a while loop that reads data from the serial port and stores it in the "data" array until the desired number of data points is reached:

```
while numel(data) < num_points
line = fscanf(s, '%f,%f,%f');
line_data = strsplit(line, ',');
data(end+1,:) = [str2double(line_data{1}) str2double(line_data{2})
str2double(line_data{3})];
end
```

Digital processing is used to filter signals in the detection of vibrations in wind turbines, and two of these filters are Butterworth and Kalman.

The Butterworth filter filters a signal to remove unwanted components (usually noise) and preserve the useful signal as faithfully as possible. Its distinctive feature is that it provides a very smooth response in the band, without ripples, making it one of the most widely used filters in engineering.

The main purpose of the Butterworth filter is to obtain the cleanest possible signal without distorting important frequencies.

Key features

1. No ripple in the passband

The Butterworth filter is also called a "maximally flat filter" because its passband is perfectly smooth—there are no oscillations like in Chebyshev or Elliptic filters. The useful signal is preserved as faithfully as possible.

2. Effectively eliminates noise outside the useful band

It has a progressive attenuation slope that increases with the order of the filter. It can eliminate electrical noise, unwanted vibrations, interference, etc.

3. It is ideal when the shape of the signal is important. Butterworth has relatively little effect on the shape of the signal over time.

Implementation in Matlab for filtering the signal from Arduino:

```
fs = 100;      % sampling frequency (Hz)
fc = 5;        % cutoff frequency (Hz)
order = 4;     % filter order

[b, a] = butter(order, fc/(fs/2), 'low');

data_filt_butter = zeros(size(data));

for k = 1:3
   data_filt_butter(:, k) = filtfilt(b, a, data(:, k));
end

figure;
plot(data_filt_butter);
title("ADXL345 - Butterworth Filter");
xlabel("Samples");
```

```
ylabel("Acceleration (raw)");
legend("X", "Y", "Z");
grid on;
```

Tower sway analysis is performed using structural dynamics models, modal analysis techniques, as well as monitoring systems based on accelerometers, gyroscopes and advanced filtering and estimation algorithms (e.g. Kalman, Butterworth filters, FFT spectral analysis).

Tower sway refers to the slow, low-amplitude oscillations of a tall structure caused by:

- wind
- earthquakes
- dynamic loads
- structural resonance

The typical frequencies for such vibrations are low:

- **0.1 – 5 Hz** (depending on height and stiffness)

The accelerometer measures the total acceleration:

$$a(t)=a_{structure}(t)+a_{noise}(t) \tag{1}$$

where noise can come from:

- local vibrations
- electromagnetic interference
- electronic noise from the sensor

For *tower sway* detection, we are only interested in low-frequency components. The following are used: Butterworth low-pass filter or band-pass filter centred on the natural frequency of the structure.

Using a filter has advantages

- Maximum flat response in the passband
- No ripple (does not distort the actual vibration amplitude)
- Good numerical stability
- Simple real-time implementation

The amplitude response of an nth-order analogue Butterworth filter is defined by the relationship:

$$\left|H(j\omega)\right| = \frac{1}{\sqrt{1+\left(\frac{\omega}{\omega_c}\right)^{2n}}} \tag{2}$$

where:

- ω is the signal frequency,
- ω_c is the cutoff frequency,
- n is the order of the filter.

At the cutoff frequency, the amplitude decreases to from its maximum value, corresponding to an attenuation of −3 dB$\frac{1}{\sqrt{2}}$. The order of the filter determines the attenuation slope in the stop band: the higher the order, the steeper the transition between the pass band and the stop band (the slope is approximately 20n dB/decade).

The poles of the Butterworth filter are evenly distributed on a semicircle in the complex plane, which ensures stability and a smooth response characteristic.

To create a system for monitoring the oscillations of an experimental wind tower, an ADXL 345 sensor was used together with an ATMEL 328 microcontroller.

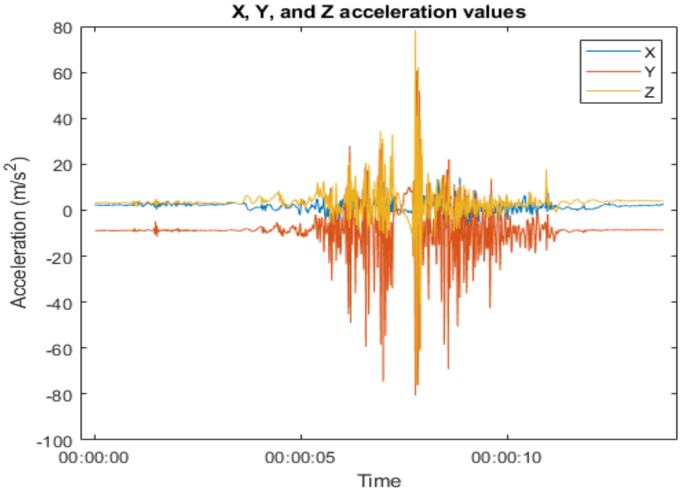

Fig.4. Saving data in Matlab via the serial port

This reads the data from the serial port and stores it in the "data" array until the "num_points" data points have been collected. The "fscanf" function reads a line of data from the serial port, which is then split into three numbers using the "strsplit" function. The three numbers are then converted to doubles and stored in the "data" array.

Ploting the data using the "plot3" function:

plot3(data(:,1), data(:,2), data(:,3), '.', 'Color', [1 0 0]);

This plots the data in a 3D diagram using the x, y, and z coordinates from the "data" array. The "." argument specifies that each data point should be represented as a point, and the "Colour" argument specifies that the colour should be red ([1 0 0]). You can use different colours for each axis by changing the colour argument.

4. CONCLUSIONS

The analysis of non-compliant vibrations of the wind turbine tower, performed on the basis of signals acquired using a MEMS accelerometer, demonstrated that this category of sensors is a viable and economical solution for continuous monitoring of the dynamic behaviour of the structure, providing adequate accuracy for structural diagnostic applications. The implementation of the Butterworth filter in the signal processing chain allowed for the effective attenuation of high-frequency noise and the isolation of relevant components of structural vibrations, maintaining an almost flat amplitude response in the passband, favourable for modal analysis.

The spectral results obtained after filtering highlighted frequency components located in the vicinity of the tower's natural frequencies, indicating the possibility of non-compliant vibration states associated with resonance and aeroelastic coupling phenomena. Correlating the acceleration values with the turbine operating regime and wind intensity showed that dynamic stresses increase significantly during transition intervals, such as start-up, shutdown and sudden speed variations, these regimes representing critical moments for structural integrity.

The study confirms that the use of a monitoring system based on MEMS sensors, combined with Butterworth digital filtering, can form the basis of a Structural Health Monitoring system capable of early detection of deviations from normal structural behaviour. The proposed methodology facilitates the identification of structural degradation trends and supports the adoption of predictive maintenance strategies, thus contributing to increasing the tower's lifespan and reducing operating costs.

At the same time, certain limitations of the proposed solution are highlighted, related to the sensitivity of MEMS accelerometers to temperature variations, the need for periodic calibration and the low resolution at very low frequencies, characteristic of the tower's high-amplitude oscillations. In the future, the integration of multiple sensors, the use of advanced signal processing techniques (wavelet analysis, adaptive filters) and the application of artificial intelligence-based methods may lead to a significant improvement in the accuracy of non-compliant vibration detection and an expansion of the monitoring system's capabilities.

In conclusion, tower oscillations are a fundamental aspect of the dynamic behaviour of modern wind turbines, with direct implications for structural safety, energy performance and operational life.

REFERENCES

[1]. Arad S., Marcu M., Pasculescu D., Petrilean D.C., *Aspects of the electric arc furnace control,* Proceeding. of international symposium advanced engineering & applied management, Faculty of Engineering Hunedoara, 2010.

[2]. Balan D., Cârstea E., Neagu S., *Analysis of non-compliant vibrations in high-altitude wind turbine towers*, National Conference on Applied Mechanics, Cluj-Napoca, 2021.

[3]. Butcher E.A., *Nonlinear Dynamics of Wind Turbine Systems*, John Wiley & Sons, Hoboken, NJ, 2016.

[4]. Cruceru A.E., Popescu F.G., Fita D.N., Marcu M.D., Olteanu R.C., Schiopu A.M., Popescu G., *Chapter 4: Exploring the Dimensions of Energy Security in Relation to the National Power Grid Current Approaches in Engineering Research and Technology,* Vol. 10, Book Publisher International, India, pp.54–69, 2024.

[5]. Cruceru A.E., Popescu F.G., Fita D.N., Marcu M.D., Olteanu R.C., Schiopu A.M., G. Popescu, *Chapter 10: Electricity Storage: The Main Pillar of Energy Security Scientific Research, New Technologies and Applications,* Vol. 10, Book Publisher International, India, pp.145–172, 2024.

[6]. Fita D., Cruceru A., Popescu G., Draghila M., Lisu A., Radu Al., *Smart Power – Analiza Soft și Hard în Relațiile Internaționale*, Editura Risoprint Cluj Napoca, pp.162., 2025.

[7]. Fita N.D., Utu I., Marcu M.D., Pasculescu D., Obretenova M.I., Popescu F.G., Lazar T., Schiopu A.M., Muresan Grecu F., Cruceru E.A., *Global energy crisis and the risk* *of blackout: interdisciplinary analysis and perspectives on energy infrastructure and security*, Energies, Vol.18, Nr. 16, 2025.

[8]. Fîță N. D., Marcu M. D., Păsculescu D., Popescu F. G., Lazăr T., *Security risks assessment on the 400/275/25 kV Elvanfoot power substation from Scotland in order to ensure resilience and energy security*, In 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET), pp. 1-6, IEEE, 2023.

[9]. Ganguly S., *Finite Element Simulation of Tower Vibrations in Wind Turbines*, International Journal of Mechanical Sciences, Vol. 78, pp. 12-25, 2013.

[10]. Ionescu S., *Comparative Study of Tower vs. Blade Induced Vibrations in Wind Systems*, Energetica, Vol. 51, No. 2, 2022.

[11]. Kusiak A., *Wind Turbine Condition Monitoring Based on Vibration Analysis*, IEEE Transactions on Industrial Electronics, Vol. 59, No. 11, 2012.

[12]. Lazar T., Marcu M.D., Utu I., Popescu F.G., Pasculescu D., *Mașini electrice - culegere de probleme,* Editura Universitas, Petroșani, pp. 197, 2023.

[13]. Marcu M., Niculescu T., Popescu F., Slusariuc R., *Incremental conductance algorithm used in a programmable microcontroller as maximum power point tracker*, International Multidisciplinary Scientific GeoConference – SGEM, Albena, Bulgaria, 2017.

[14]. Marcu M., Popescu F., Pana L., Slusariuc I.R., *Modelling and simulation of solar radiation*, Applied Mechanics and Materials, Vol. 710, pp. 113–118, 2015.

[15]. Marcu M., Popescu FG, Slusariuc R., S Arad S., Handra AD., *Overview of control methods for induction motor drives,* Annals of the University of Petrosani, Electrical Engineering, 20, 2018.

[16]. Moaveni B., *Structural Dynamics and Earthquake Engineering*, Cambridge University Press, Cambridge, 2020.

[17]. Pasculescu D., Slusariuc I.R., Popescu F., Fîță N.D., Tatar A., Lazar T., *Modelling and simulation of lighting of a road with 2 strips per direction to EN 13201:2015 standard*, Annals of the University of Petrosani, Electrical Engineering, Vol. 24, pp. 65–74, 2022.

[18]. Popescu F.G., **Marcu M.D.,** *Metode moderne de analiză şi reducere a armonicilor de curent şi tensiune*, Editura Universitas, Petroşani, 2016.

[19]. Popescu F.G., Pasculescu D., Marcu M., Pasculescu V.M., Fita N.D., Tatar A., Lazar T., P*rinciples of effective energy management and power control system,* Annals of the University of Petrosani, Electrical Engineering, Vol.24, pp.111-118, 2022.

[20]. Popov V.V., *Dynamics of Rotating Machines*, Springer, Berlin, 2013.

[21]. Riurean S., Fîță N. D., Păsculescu D., Slușariuc R., *Securing photovoltaic systems as critical infrastructure: A multi-layered assessment of risk, safety, and cybersecurity*, Sustainability, 17(10):4397, 2025.

[22]. Rosca S.D., Riurean S., Popescu G., Lazar T., Popescu F.G., *Development of a BCI video game for mental state recognition*, Annals of the University of Petrosani, Electrical Engineering, VOL. 26, pp. 383-390, 2024.

[23]. Rotaru F., *Modelling and Identification of Wind Turbine Tower Vibration Modes*, International Conference on Mechanical Engineering, Bucharest, 2019.

[24]. Safta G.E., Fita D.N., Popescu F.G., Cruceru A.E., Olteanu R.C., Schiopu A.M., Popescu G., *Chapter 6: Improving the Quality of Electricity in Eastern Europe: Challenges and Solutions Scientific Research, Scientific Research, New Technologies and Applications,* Vol. 9, Book Publisher International, India, pp.132–149, 2024.

[25]. Staicu M., Păunescu C., *Contributions to Dynamic Analysis of Wind Turbine Towers under Stochastic Wind Loading*, AGIR Bulletin, Vol. 23, 2013.

[26]. Vlad L., Georgescu D., *Analysis of non-compliant vibrations induced by gusts on wind turbine towers*, Revista Energetică, Vol. 59, No. 4, 2020.