**Introduction:**

In rotating machinery, vibration is a key indicator of mechanical health. A sudden increase in vibration can point toward faults such as imbalance, misalignment, or bearing wear. This project aims to perform a comparative vibration analysis between a healthy and a faulty bench grinder using a low-cost inertial sensor (MPU6050) and ESP32 microcontroller. Data is analyzed using MATLAB, focusing on three sampling rates: 5 Hz, 10 Hz, and 20 Hz. The goal is to detect anomalies in vibration patterns using time-domain, filtered, and frequency-domain (FFT) methods.

**Objectives:**

Using an **MPU6050 accelerometer** and **ESP32**, we collected vibration data in three axes (X, Y, and Z), recorded it in Excel format, and processed the data using **MATLAB**. The key objectives included:

* To collect tri-axial vibration data using MPU6050 from both healthy and faulty bench grinders.
* To apply filtering and FFT to identify significant frequency-domain characteristics.
* To compare the vibration behavior across different sampling rates (5 Hz, 10 Hz, and 20 Hz).
* To detect and visualize differences in vibration amplitude and frequency between healthy and faulty machines.

**Experimental Setup:**

To perform condition monitoring through vibration analysis, we set up an experiment involving two bench grinders—one operating normally (healthy) and one exhibiting signs of mechanical faults (faulty). The setup focused on capturing, recording, and analyzing vibration data from both machines under real operating conditions.

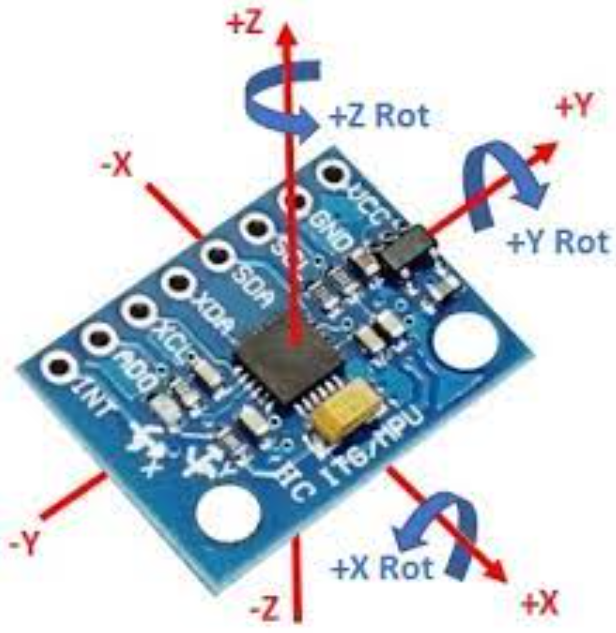
## **Equipment Used**

### **1. Bench Grinders**

* **Healthy Grinder**: Operates smoothly with expected vibration patterns.
* **Faulty Grinder**: Visibly vibrates more, especially under load, indicating potential mechanical defects such as:
  + Unbalanced rotor
  + Loose components
  + Worn bearings or misalignment

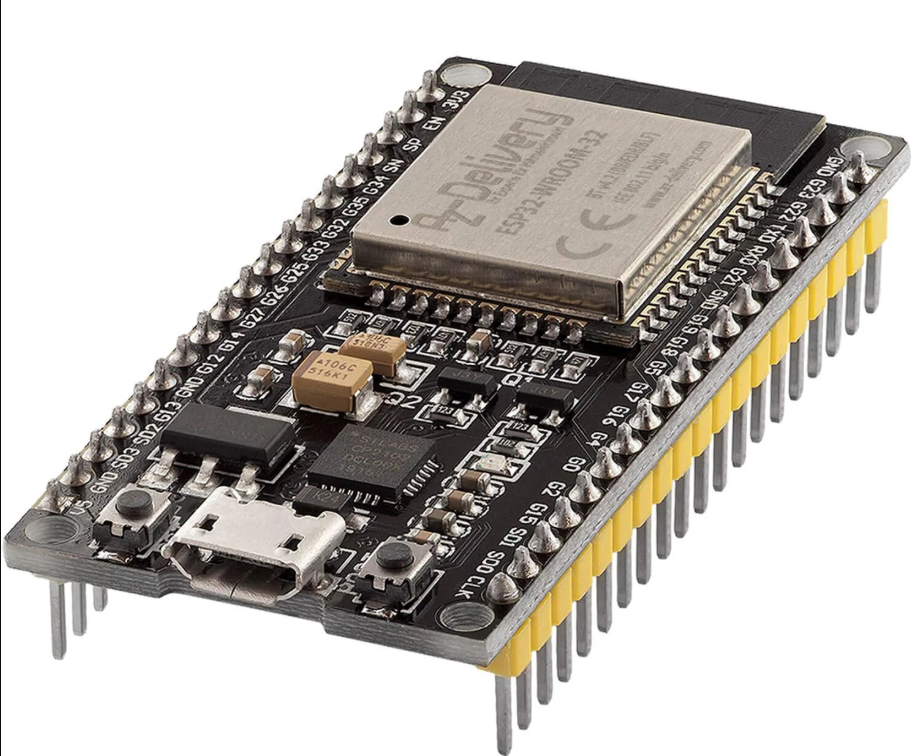
### **2. MPU6050**

The MPU6050 is a MEMS (Micro-Electro-Mechanical Systems) sensor that combines a 3-axis accelerometer and a 3-axis gyroscope into a single compact module. It is specifically designed to measure linear acceleration and angular velocity in three-dimensional space (X, Y, and Z axes). In this experiment, the accelerometer component of the MPU6050 was utilized to capture vibration data from the motors. The sensor outputs raw acceleration values in units of *g*, which represents gravitational acceleration. For communication and data transfer, the MPU6050 was interfaced with the ESP32 microcontroller using the I2C (Inter-Integrated Circuit) protocol, allowing efficient and synchronized data acquisition.



### **3. ESP32 Microcontroller:**

The ESP32 microcontroller played a central role in data acquisition and transmission for this experiment. It was responsible for reading real-time acceleration data from the MPU6050 sensor and either storing it locally or transmitting it to a computer via USB or Wi-Fi. The ESP32 also ensured that each data point was accurately time-stamped, which is essential for synchronized analysis and processing in MATLAB. Its processing capability and communication flexibility made it ideal for capturing and logging vibration data effectively.



### **4. Data Logging and Storage:**

Data logging was performed at three different sampling rates—5 Hz, 10 Hz, and 20 Hz—to observe how resolution impacts vibration analysis and fault detection. Each test ran for approximately 120 seconds, capturing acceleration data from all three axes (X, Y, Z) using the MPU6050 sensor. The ESP32 microcontroller recorded the data with precise time-stamps and stored the results in CSV format. These files included columns for time (in milliseconds) and acceleration values (in units of g), allowing for detailed post-processing in MATLAB.

### **5. MATLAB Software:**

MATLAB was utilized for post-processing and detailed analysis of the collected vibration data. The software enabled seamless importing of CSV files and facilitated signal conditioning through Butterworth low-pass filtering to remove noise. Fast Fourier Transform (FFT) was applied to shift the data into the frequency domain, allowing for fault identification based on frequency components. Additionally, MATLAB was used to generate both 2D and 3D visualizations of the raw, filtered, and frequency-transformed signals, aiding in the comparative evaluation of healthy and faulty motor behaviors across different sampling rates.

**Data Processing in MATLAB**

The collected data was processed using the following steps:

### **1. Import and Preprocessing**

* Data from Excel files was read using ***readtable ()*** function*.*
* Time was converted from milliseconds to seconds.
* All datasets were truncated to 120 seconds to ensure uniformity.

### **2. Signal Filtering**

* A 4th-order **Butterworth low-pass filter** was applied to remove high-frequency noise.
* The cutoff frequency was normalized based on each sampling rate:

|  |
| --- |
| *matlab*  *[b, a] = butter(4, Cutoff/(Fs/2));*  *Ax\_filt = filtfilt(b, a, Ax\_raw);* |

### **3. FFT Analysis**

* Fast Fourier Transform (FFT) was applied to filtered signals to identify dominant vibration frequencies.
* Spectral analysis was done for each axis (X, Y and Z).

|  |
| --- |
| *matlab*  *fft\_signal = abs(fft(filtered\_signal));* |

**Visualization**

* **3D plots** were used to visualize raw and filtered acceleration over time and sampling rate.
* FFT spectrum plots showed frequency components, highlighting dominant peaks.

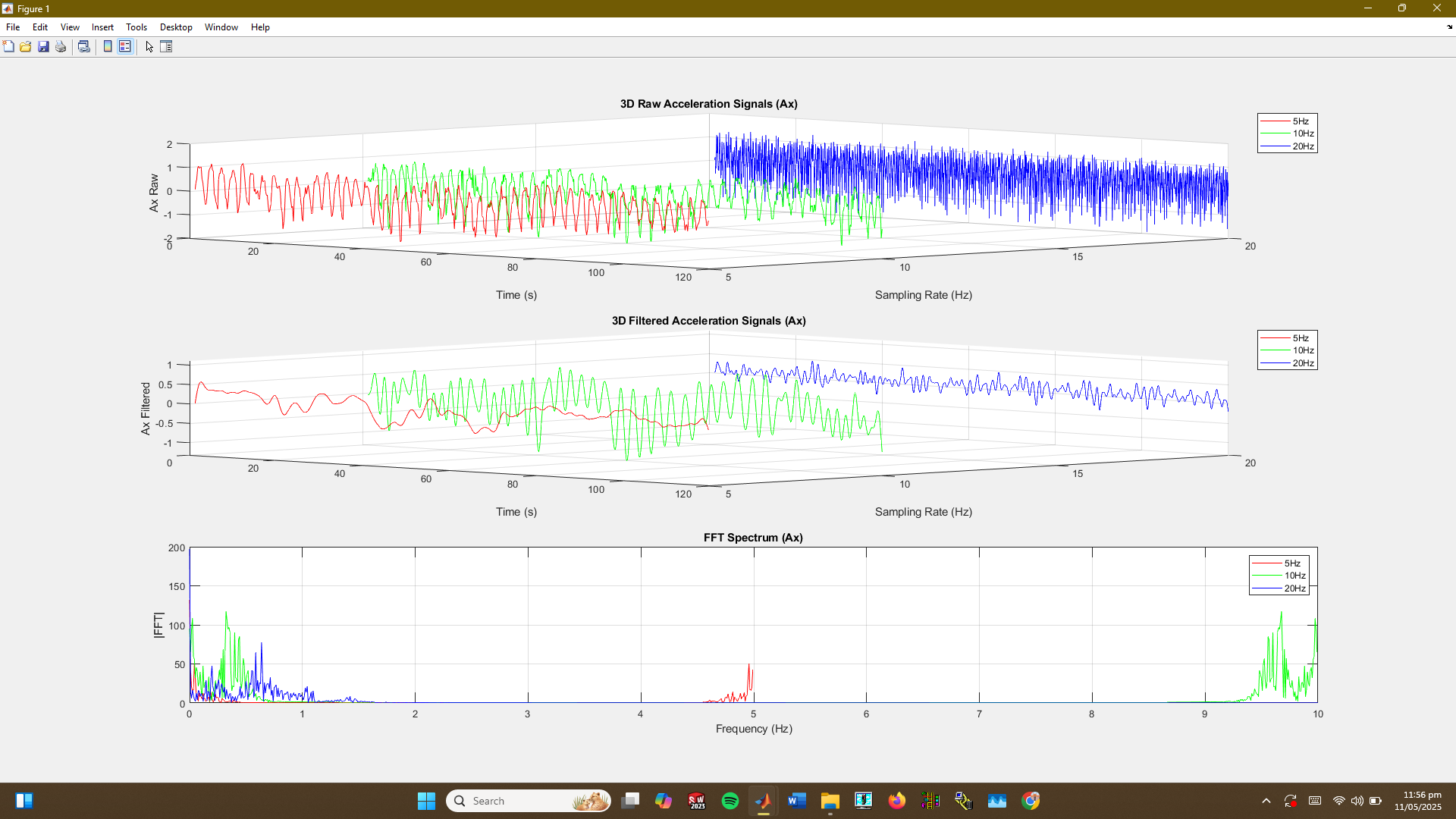
**Results and Analysis:**

### **1. Healthy Motor**

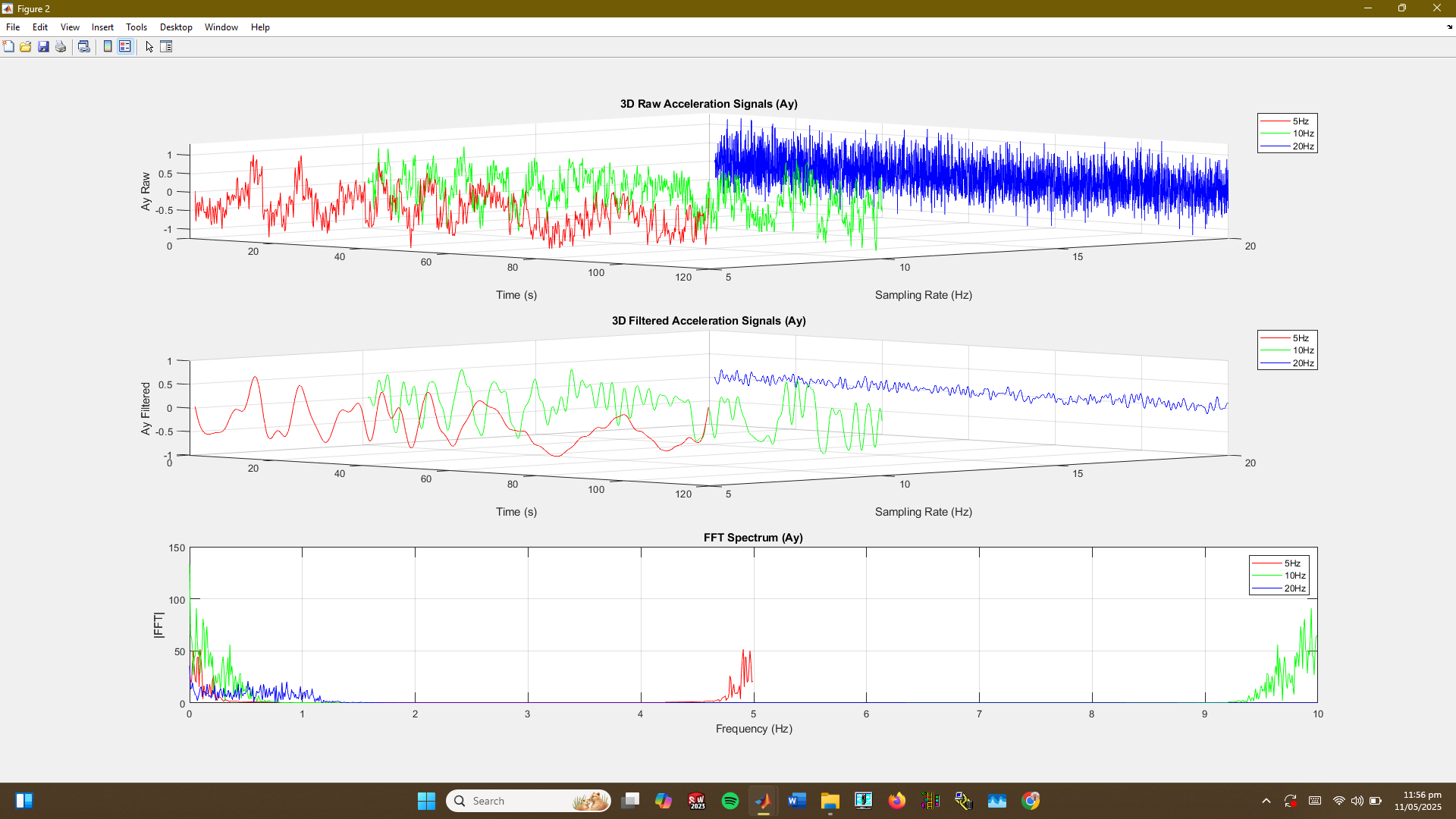
The healthy motor exhibited low and consistent vibrations across the X, Y, and Z axes, indicating normal operational behavior. In the frequency domain, FFT plots of the healthy motor revealed low-amplitude, periodic components corresponding to the motor’s base frequency and its harmonics. The application of low-pass filtering effectively removed minor high-frequency noise while preserving the essential vibration patterns necessary for analysis.

**Results:**

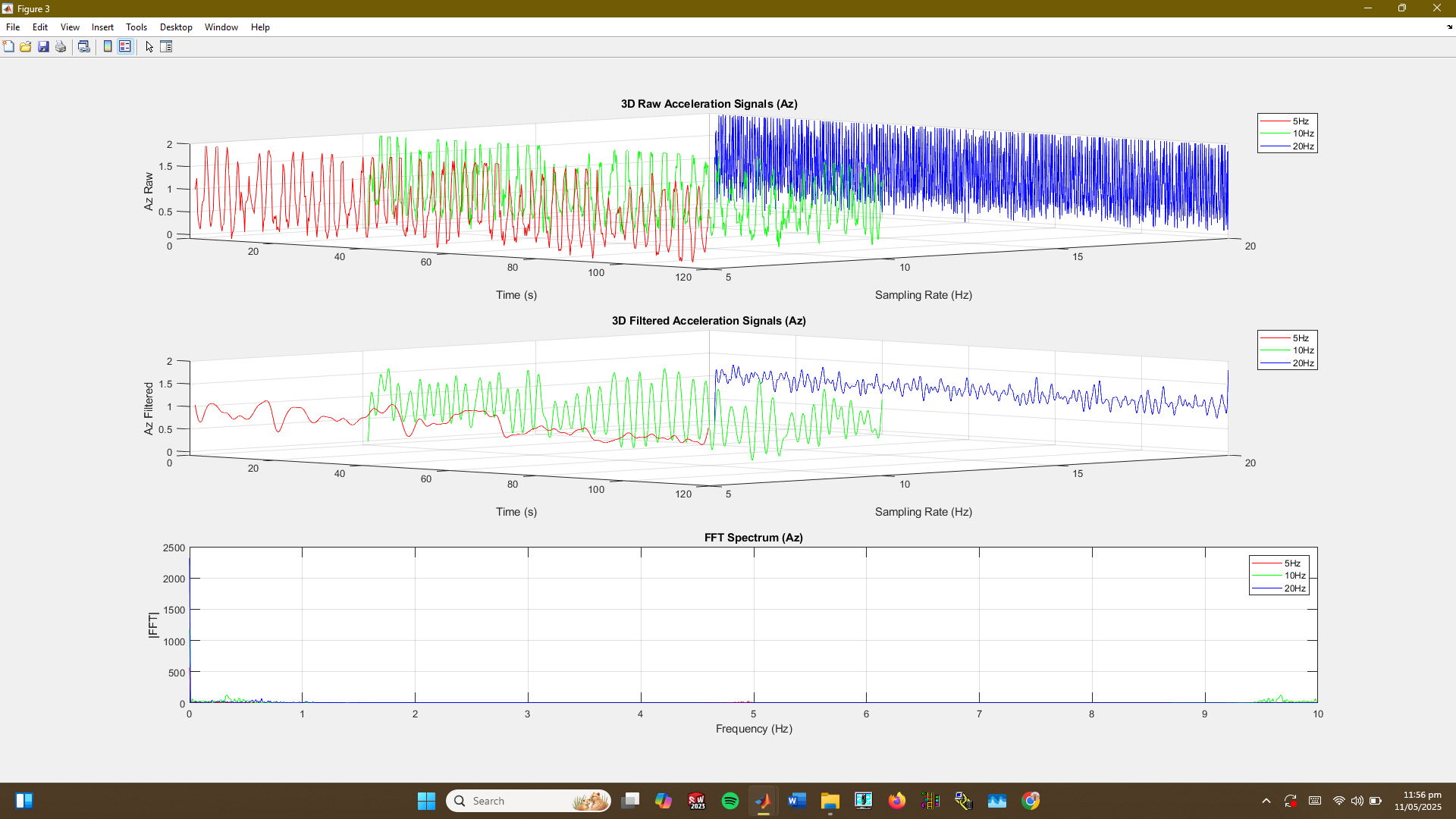
**Ax:**



**Ay:**



**Az:**

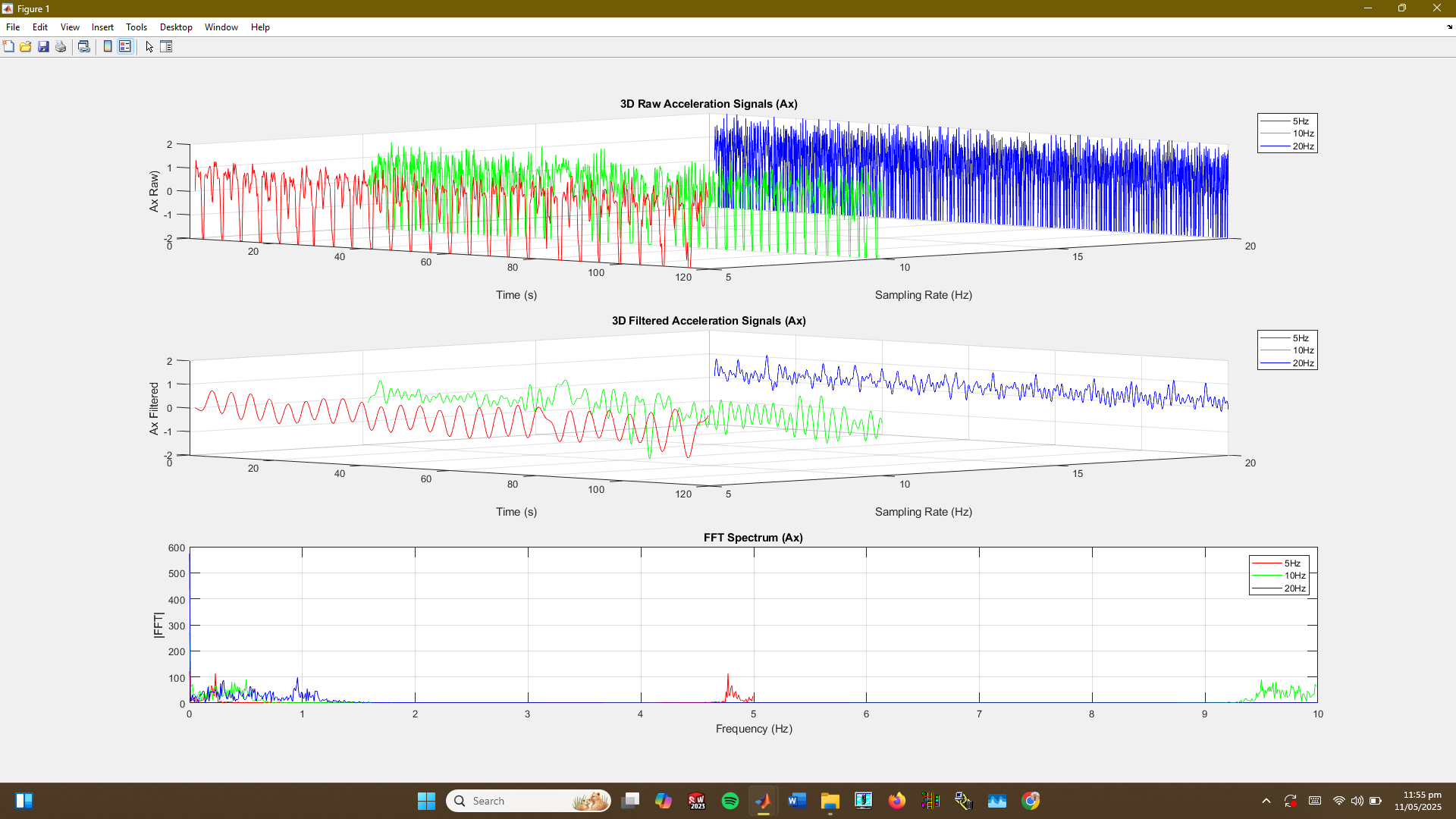


### **2. Faulty Motor**

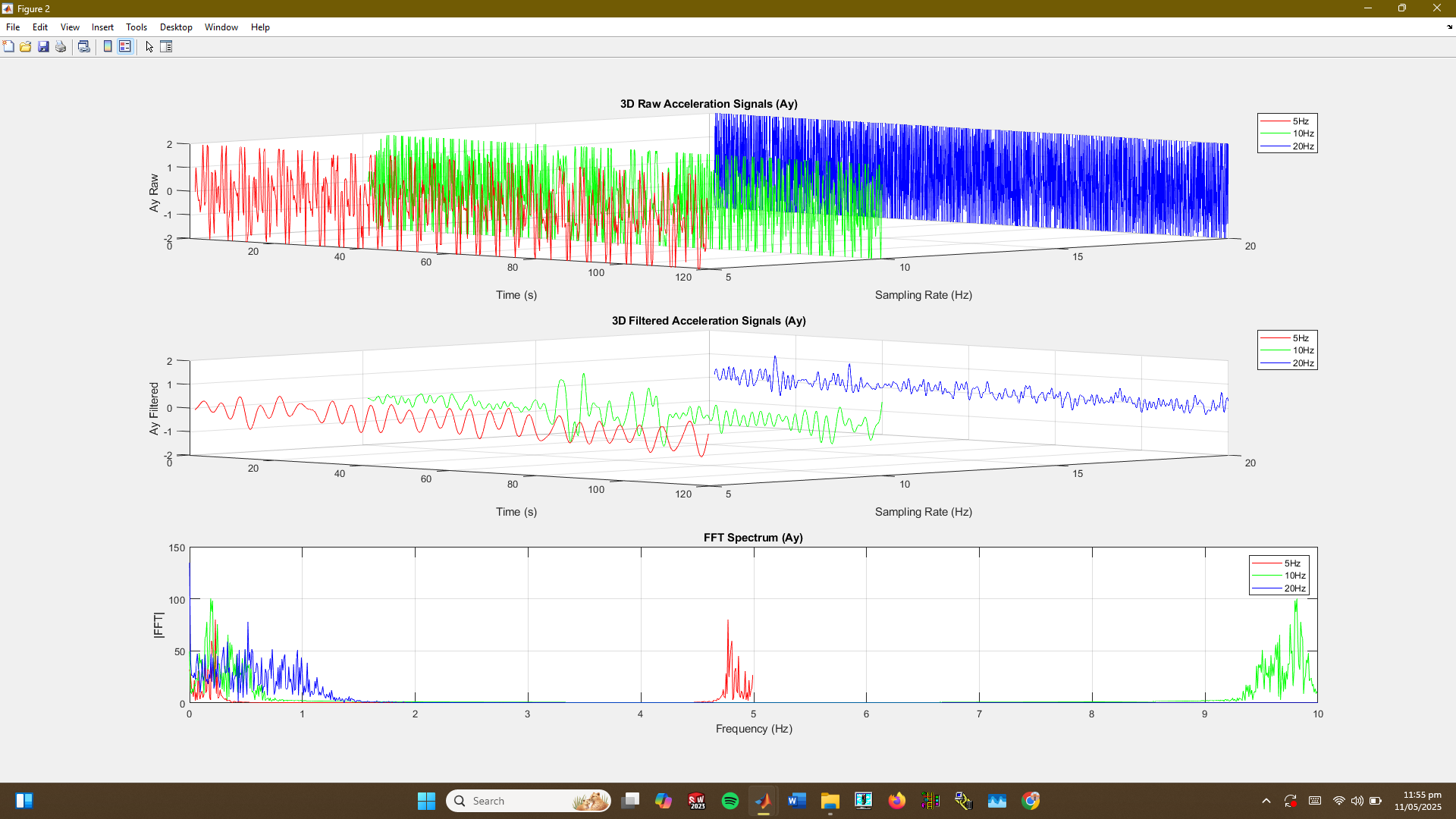
In contrast, the faulty motor data, particularly at 5 Hz, displayed irregular spikes and significantly increased vibration amplitudes, especially under load conditions. These anomalies were most pronounced in the X and Y axes, suggesting mechanical issues such as imbalance, shaft misalignment, or bearing defects. The time-domain signals appeared more chaotic, and the FFT spectrum showed stronger and broader frequency peaks compared to the healthy motor, further supporting the presence of faults.

**Results:**

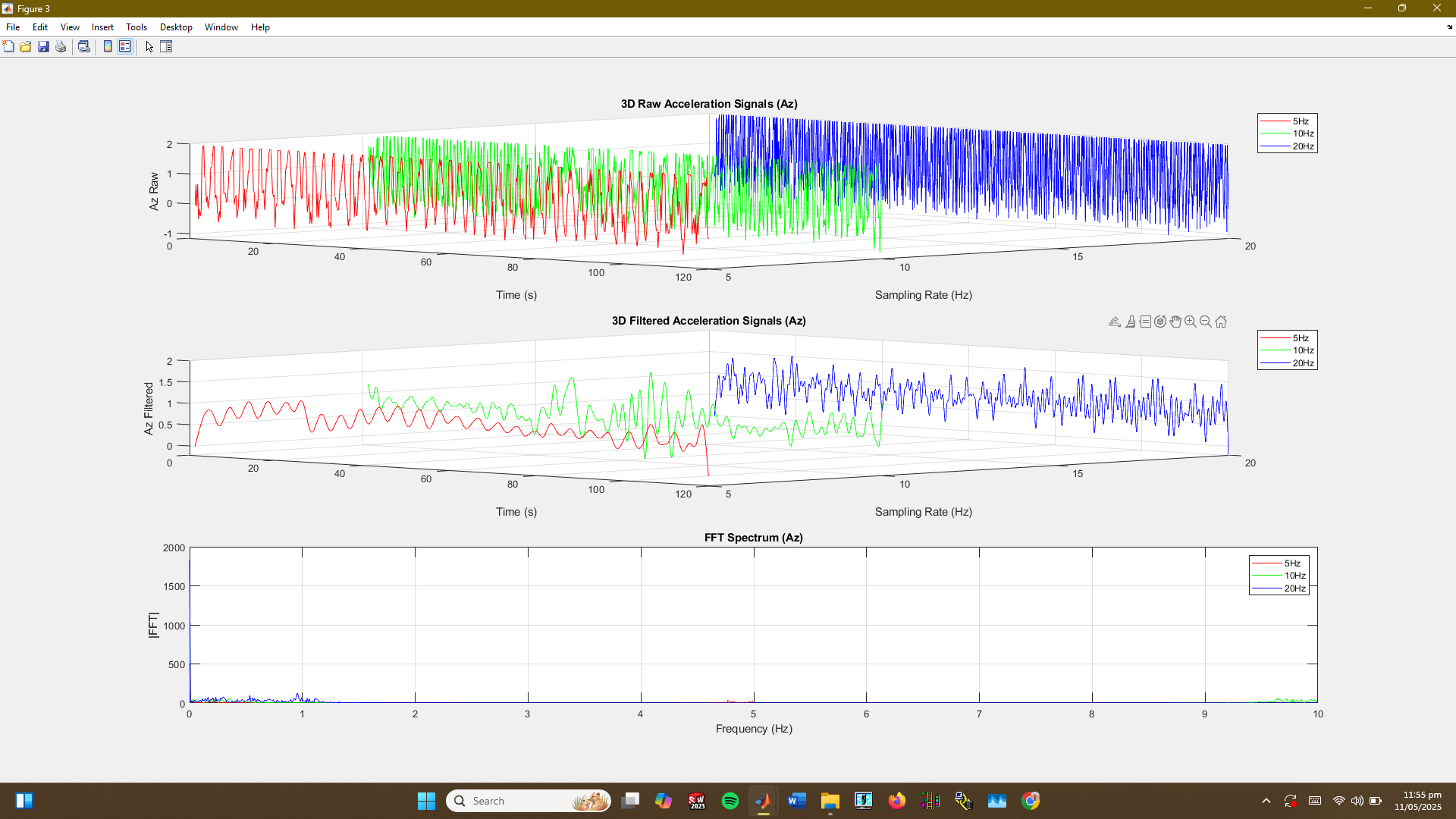
**Ax**



**Ay:**



**Az:**



### **3. Effect of Sampling Rate**

* **Higher sampling rates (10 Hz and 20 Hz)** captured finer vibration details.
* At **lower rates (5 Hz)**, the signal appeared more smoothed but still revealed major faults.
* FFT accuracy and resolution improved with increasing sampling rate, as seen in clearer spectral peaks at 20 Hz.

### **4. Key Observations**

The faulty motor showed higher vibration energy, especially under load, as confirmed by FFT plots indicating mechanical issues. Vibration analysis proved effective for early fault detection and condition monitoring.

|  |  |  |
| --- | --- | --- |
| Sampling rate |  | Faulty Motor |
| 5 Hz | Smooth, periodic low amplitude | Spikes, irregular high amplitude |
| 10 Hz | More detailed periodic behavior | Amplified, noisy signal with more peaks |
| 20 Hz | Fine structure and stable peaks | Clear dominant frequency due to fault |

**Conclusion:**

This project showed that vibration data from an MPU6050-ESP32 setup can effectively identify motor faults when processed in MATLAB. The faulty motor displayed abnormal behavior, and applying filtering and FFT analysis enhanced signal clarity. Higher sampling rates improved fault detection but also increased data size and processing requirements.