NANYANG POLYTECHNIC
EG431D Data Acquisition

Lesson 7

Course: Diploma in Engineering (Electronic and Digital Engineering)

Module: EG431D Data Acquisition

Title: Frequency Domain Analysis

Objective:

The objective is to provide a comprehensive understanding of frequency domain analysis by describing the process of analyzing both stationary and non-stationary signals, focusing on how frequency components are extracted and interpreted to reveal signal characteristics. It aims to explain the fundamental principles of frequency domain analysis, including the Fourier Transform and its variations like FFT and STFT, and their application in converting time-domain signals into frequency components. Furthermore, the objective includes demonstrating the practical use of frequency domain techniques, such as spectral analysis and filtering, in real-world applications like isolating frequency ranges in audio processing, identifying periodicity in biomedical signals, and diagnosing anomalies in non-stationary signals for machinery vibration analysis.

Learning Objectives:

- Describe the process of analyzing both stationary and non-stationary signals in the frequency domain, explaining how frequency components are extracted and interpreted to understand the characteristics of signals.
- Explain the fundamental rules of frequency domain analysis, such as Fourier Transform and its variations (e.g., FFT, STFT), and how these methods are applied to convert time-domain signals into their frequency components.
- Demonstrate how frequency domain analysis techniques including spectral analysis and filtering are used in real-world scenarios, such as detecting and isolating specific frequency ranges for audio processing, identifying periodicity in stationary signals for biomedical signal processing, and diagnosing anomalies in non-stationary signals for machine vibration analysis.

1. Introduction

Frequency domain analysis is a fundamental concept in signal processing, focusing on understanding the frequency components of a signal rather than its variation over time. Unlike the time-domain representation, where signals are analyzed based on amplitude or voltage variations as a function of time, frequency domain analysis decomposes a signal into its constituent sinusoidal components. This transformation enables engineers and scientists to identify the frequencies present in a signal, their amplitudes, and their phases. This representation is crucial for analyzing periodic, non-periodic, and complex signals in a variety of applications, including audio processing, communications, control systems, and vibration analysis.

The core mathematical tool for frequency domain analysis is the **Fourier Transform**, which transforms a time-domain signal $x(t)$ into a frequency-domain representation $X(f)$ as depicted in Figure 1A. The Fourier Transform operates under the principle that any complex signal can be represented as a **sum of sinusoids** at different frequencies, amplitudes, and phases. The **Discrete Fourier Transform** (DFT), a sampled version of the Fourier Transform, is widely used for analyzing discrete signals in digital systems. To make the computation of DFT more efficient, the **Fast Fourier Transform** (FFT) algorithm is often employed, reducing computational complexity and enabling real-time analysis for applications requiring high-speed processing.

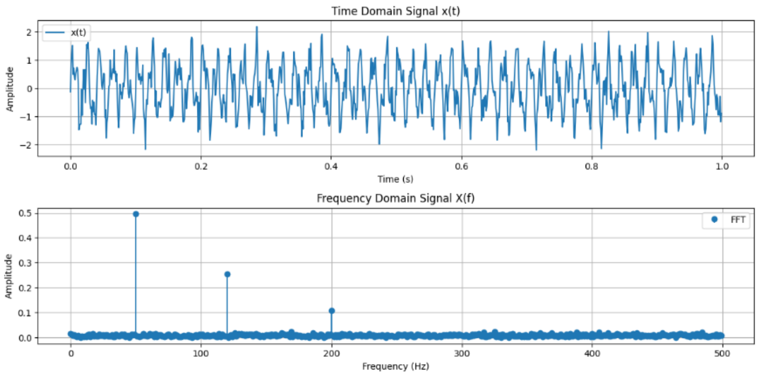

Figure 1A – A complex signal and its frequency domain equivalent

Frequency domain analysis provides several advantages, particularly in identifying signal characteristics that are difficult to discern in the time domain. For example, periodic signals such as audio tones or communication signals exhibit distinct peaks in the frequency spectrum at their fundamental and harmonic frequencies. This analysis is invaluable for tasks such as spectral filtering, where specific frequency components can be isolated or suppressed. Similarly, in communication systems, frequency domain analysis is essential for bandwidth allocation, modulation, and interference mitigation.

Another technical application of frequency domain analysis lies in system characterization and response analysis. Using tools like the **Bode plot** and **frequency response function**, engineers evaluate how systems respond to different frequency inputs, which is critical in designing control systems, amplifiers, and filters. The concept of **convolution** in the time domain, which represents the interaction of signals with a system's impulse response, becomes a straightforward multiplication operation in the frequency domain, simplifying complex calculations and enhancing computational efficiency.

Filtering is another domain where frequency domain analysis plays a critical role. Filters, such as low-pass, high-pass, band-pass, and band-stop, are designed and analyzed in the frequency domain to ensure they meet the desired specifications, such as cut-off frequencies, transition bandwidths, and stop-band attenuation. Digital filters, implemented using Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) methods, rely heavily on frequency domain representations for their design and verification.

Modern advancements in frequency domain analysis have extended its application into image and video processing, where transforms like the **2D Fourier Transform** and the **Discrete Cosine Transform** (DCT) are used for image compression, denoising, and feature extraction. Additionally, in biomedical engineering, frequency domain analysis is utilized to examine electrocardiogram (ECG) and electroencephalogram (EEG) signals, enabling the detection of anomalies or periodic patterns associated with physiological conditions.

2. Fast Fourier Transform

The Fast Fourier Transform (FFT) is a mathematical algorithm used to compute the Discrete Fourier Transform (DFT) of a signal efficiently. The DFT itself is a transformation that converts a discrete-time signal from the time domain into the frequency domain. Mathematically, for a sequence of N discrete samples $x[n]$, the DFT is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \tag{1}$$

$$k = 0, 1, 2, \dots, N-1$$

Where $x[n]$ is the discrete signal input, $X[k]$ is the frequency-domain output, N is the total number of samples, and $e^{-j\frac{2\pi}{N}kn}$ is a complex exponential function that forms the basis of the transformation.

The DFT has a computational complexity of $O(N^2)$, as it involves N summations, each requiring N multiplications. This makes it computationally expensive for large N. The FFT reduces this complexity to $O(NlogN)$ by decomposing the DFT computation into smaller, more manageable computations.

The FFT is based on the principle of divide-and-conquer. The algorithm works by splitting the DFT into smaller DFTs of even and odd-indexed samples. This decomposition is achieved by using the periodic and symmetry properties of the complex exponential function in (1) and can be re-written as:

$$X[k] = \sum_{m=0}^{\frac{N}{2}-1} x[2m] e^{-j\frac{2\pi}{N}k(2m)} + \sum_{m=0}^{\frac{N}{2}-1} x[2m+1] e^{-j\frac{2\pi}{N}k(2m+1)}$$

Where the first term represents the DFT of the even-indexed samples $x[2m]$, and the second term represents the DFT of the odd-indexed samples $x[2m+1]$.

Using the periodicity property of the exponential function, the DFT computation for the original sequence can be expressed as a combination of the smaller DFTs of the even and odd parts. This allows the algorithm to recursively compute the DFT, reducing redundant calculations.

Radix-2 Cooley-Tukey FFT Algorithm

The **radix-2 Cooley-Tukey FFT** is the most common implementation of the FFT. It assumes N, the number of input samples, is a power of 2 ($N = 2^m$). The algorithm recursively breaks the DFT into smaller DFTs until it reaches size 2, where the DFT can be computed directly using:

$$X[0] = x[0] + x[1], \qquad X[1] = x[0] - x[1]$$

At each stage, the results of the smaller DFTs are combined using the **twiddle factors**, which are the precomputed values of $e^{-j\frac{2\pi}{N}kn}$. The recursive nature of the algorithm and the reuse of these twiddle factors lead to a significant reduction in computational effort.

FFT Output Interpretation

The output $X[k]$ of the FFT represents the frequency components of the input signal where:

- The magnitude $|X[k]|$ gives the amplitude of the *k*-th frequency component.
- The phase $\angle X[k] = tan^{-1}\left(\frac{Im(X[k])}{Re(X[k])}\right)$ gives the phase shift of the component.

Practical Considerations

In practice, FFT requires padding the input signal to the nearest power of 2 if N is not already a power of 2. Additionally, the signal is often windowed (e.g., using Hamming or Hann windows) before applying the FFT to reduce spectral leakage. FFT outputs are normalized or scaled to interpret the amplitude correctly.

3. Short-Time Fourier Transform

The **Short-Time Fourier Transform** (STFT) is a tool in signal processing, offering a means to analyze signals with time-varying frequency content. Unlike the traditional Fourier Transform, which provides a static frequency spectrum, STFT provides **time-frequency localization** by segmenting the signal into smaller time windows. This capability makes it particularly suited for non-stationary signals where frequency characteristics change over time. The wide-ranging applicability of STFT spans multiple disciplines, from audio processing to biomedical engineering and mechanical diagnostics.

(A) Audio Signal Analysis and Processing

One of the prevalent applications of STFT is in audio signal analysis and processing. Audio signals, whether in speech or music, are inherently non-stationary, with varying frequency components over time. STFT is used in:

- **Speech Analysis and Synthesis**: STFT helps extract detailed features such as pitch, formants, and phonemes, which are crucial for speech synthesis, recognition, and coding. For instance, in automatic speech recognition systems, STFT-derived spectrograms are used as inputs for machine learning models.

- **Music Signal Processing**: In music analysis, STFT identifies the harmonic content and detects beats or rhythm patterns. It also aids in separating instruments in a polyphonic piece, allowing for applications such as karaoke systems or remixing.
- **Noise Reduction**: By analyzing the spectrogram of an audio signal, noise components can be isolated and suppressed, enhancing the clarity of speech or music.

(B) Biomedical Signal Analysis

Biomedical signals like electrocardiograms (ECG) and electroencephalograms (EEG) are vital indicators of human health, and their non-stationary nature makes them ideal candidates for STFT analysis.

- **EEG Signal Processing**: STFT helps identify abnormal brainwave activity, such as epileptic seizures, by highlighting changes in frequency patterns over time.
- **Heart Rate Variability (HRV)**: STFT is used to monitor frequency components of HRV, offering insights into autonomic nervous system functions, which are critical for stress analysis or cardiac health assessment.
- **Speech Disorder Analysis**: By analyzing vocal cord vibrations through time-frequency representations, STFT assists in diagnosing and treating speech impairments.

(C) Mechanical Vibration Analysis

STFT is widely employed in mechanical systems for fault detection and predictive maintenance. Machines generate vibrations with characteristic frequency patterns, which STFT can analyze in real time.

- **Condition Monitoring**: STFT identifies deviations in normal vibration frequencies caused by wear, misalignment, or imbalance in machinery components such as bearings and gears.
- **Rotational Machinery Diagnostics**: For rotating equipment, STFT tracks changes in rotational harmonics, pinpointing issues like shaft misalignment or gear tooth wear, which might otherwise lead to catastrophic failures.

(D) Seismic and Geophysical Applications

Geophysical signals, particularly seismic waves, are complex and non-stationary. STFT is instrumental in understanding these signals:

- **Earthquake Analysis**: STFT reveals the frequency content of seismic waves during an earthquake, aiding in seismic source characterization.
- **Resource Exploration**: In oil and gas exploration, STFT processes seismic reflection data to detect subsurface structures and assess resource potential.

(E) Communication Systems

In telecommunications, STFT is a cornerstone for analyzing modulated signals and spectrum usage:

- **Spectrum Analysis**: STFT identifies occupied frequency bands and detects interference in wireless communications, enhancing signal reliability.
- **Signal Decoding**: By analyzing modulated signals like amplitude modulation (AM) or frequency modulation (FM), STFT aids in decoding information transmitted over communication channels.

(F) Radar and Sonar Signal Processing

Radar and sonar systems rely on STFT for tracking objects and measuring their velocity:

- **Doppler Shift Analysis**: STFT extracts frequency shifts caused by moving objects, enabling precise speed and direction detection in radar systems.
- **Sonar Applications**: In underwater acoustics, STFT analyzes echoes to detect and classify underwater objects or obstacles.

(G) Image Processing

STFT extends its utility to image processing by analyzing localized frequency variations:

- **Texture Analysis**: In computer vision, STFT identifies textures and repetitive patterns within an image.
- **Edge Detection**: By detecting high-frequency content in localized regions, STFT assists in identifying edges or boundaries in images.

(H) Environmental Monitoring

TFT is a valuable tool for analyzing environmental signals such as noise pollution or underwater acoustics:

- **Noise Pollution Monitoring**: STFT measures the frequency content of environmental sounds, helping to evaluate and mitigate noise pollution.
- **Underwater Acoustics**: It is used to analyze soundscapes in marine environments, aiding in ecosystem monitoring and navigation.

4. Frequency Domain Analysis with Arduino

To perform frequency domain analysis with Arduino, libraries such as *arduinoFFT* offer optimized implementations of the **FFT algorithm**, enabling efficient computation of frequency spectra. The process starts with signal acquisition, where a sensor connected to an analog input pin records the signal (or to digital interface pins if a digital sensor is used). Arduino Board's ADC samples the analog signal at a fixed rate, ensuring that the Nyquist criterion is met - this rate must be at least twice the highest frequency component in the signal. The sampled data is then processed through the FFT algorithm, which decomposes the signal into its frequency components. The FFT outputs an array of magnitudes corresponding to

discrete frequency bins, which can be further analyzed to identify dominant frequency components.

Arduino-based frequency domain analysis has advanced applications across various fields. In vibration analysis, accelerometers connected to Arduino can measure and analyze vibrations to detect machinery faults or imbalances. In bio-signal processing, frequency-specific patterns in heartbeat or EEG signals can be identified. For spectrum monitoring, Arduino systems can be used to visualize and monitor radio or sound frequencies for educational or diagnostic purposes.

4.1 Audio Signal Analysis: Environmental Sound Analysis

Environmental sound analysis in the frequency domain involves capturing, processing, and analyzing ambient audio signals to extract meaningful insights about the acoustic environment. By leveraging Arduino's capabilities, environmental sounds such as traffic noise, machine hum, or natural ambient sounds can be decomposed into their frequency components to facilitate noise monitoring, sound classification, or even anomaly detection.

To implement such an analysis, a microphone module or sound sensor is connected to the Arduino board's analog input (covered in Lesson 2). This sensor captures the audio signals as varying voltage levels representing sound pressure. The Arduino samples these analog signals at a sufficient rate, such as 8 kHz, to ensure that the Nyquist criterion is satisfied for analyzing frequencies up to 4 kHz. These sampled data points represent the time-domain representation of the sound wave.

Using the Fast Fourier Transform (FFT), the time-domain data is transformed into the frequency domain. This transformation breaks down the complex sound wave into its constituent frequencies, revealing information about the **spectral content** of the environmental sound. For instance, dominant frequency peaks in the spectrum may indicate specific sound sources, such as machinery operating at a certain frequency or the pitch of bird chirps in a natural setting.

The implementation involves several key steps. First, the sound signal is sampled consistently, and a windowing function (e.g., Hamming or Hanning window) is applied to minimize **spectral leakage** during the FFT operation. The FFT is then executed using an available library such as ArduinoFFT, which computes the magnitude of the frequency components. The resulting **frequency spectrum** can be visualized in real time on a connected display or sent to a computer for further analysis.

For example, in a noise monitoring application, the system can identify and log frequency bands exceeding permissible limits, such as those in industrial environments or urban areas. Alternatively, in an ecological study, the frequency spectrum could be used to identify specific wildlife sounds, aiding in biodiversity assessments.

4.2 Vibration Analysis: Machine Condition Monitoring and Predictive Maintenance

Vibration analysis in the frequency domain is a technique for monitoring machine condition and enabling predictive maintenance. By analyzing the frequency content of vibration signals, faults such as **misalignment**, **imbalance**, **looseness**, or **bearing wear** can be detected early, preventing unexpected downtime and reducing maintenance costs. Arduino Board equipped with accelerometer sensors and capable of performing Fast Fourier Transform (FFT) operations, provides an efficient and cost-effective solution for this purpose.

To perform vibration analysis, a tri-axial accelerometer (covered in Lesson 2) is connected to the Arduino Board. The accelerometer measures vibrations along the X, Y, and Z axes, outputting real-time acceleration data in terms of gravitational force (g). These measurements are captured by the Arduino Board through its analog-to-digital converter (ADC) or digital interface port (e.g., SPI or I2C port if digital accelerometer is used). To ensure that the frequency range of interest is preserved, the sampling rate must be at least twice the highest frequency component of the vibration signal, adhering to the Nyquist-Shannon theorem.

The vibration signals, initially in the time domain, are converted to the frequency domain using FFT. Arduino libraries, such as ArduinoFFT, simplify the implementation of FFT. The FFT algorithm decomposes the time-domain signal into its frequency components, generating a spectrum that reveals the amplitude of vibrations at different frequencies. This frequency spectrum is crucial for identifying **characteristic fault frequencies** associated with specific machine components, such as rotating shafts or bearings.

Figure 4A illustrates an example of a machine vibration signal measured along a single axis, showing both time-domain and frequency-domain plots. In the time domain, the plot presents the raw signal, which includes multiple frequency components superimposed with noise. The frequency-domain plot, derived from the FFT, highlights the primary frequency components that contribute to the machine's vibrations, such as those from motor imbalance, loose parts, and bearing faults. These visualizations are crucial for diagnosing potential mechanical issues by identifying dominant frequencies and anomalies in the vibration data.

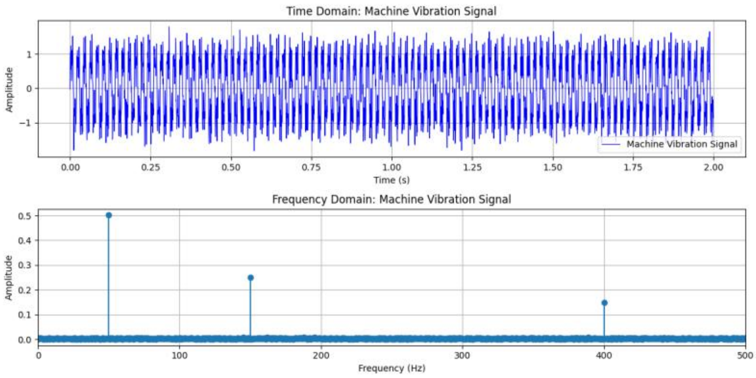

Figure 4A – Machine vibration signal and its frequency domain equivalent

The frequency spectrum obtained from FFT analysis can be compared with **baseline vibration signatures** to identify **anomalies**. For instance, an increase in amplitude at specific frequencies may indicate imbalance, while harmonic peaks may suggest misalignment. Additionally, broadband frequency content might signify looseness, and characteristic frequency bands can help diagnose bearing defects.

Using Arduino, these insights can be visualized in real-time on an LCD screen, logged for trend analysis, or transmitted to a cloud-based dashboard for remote monitoring. Alerts can be triggered when vibration levels exceed predefined thresholds, prompting timely maintenance actions.

The predictive maintenance aspect involves continuously monitoring vibration data and using historical trends to predict potential failures. By identifying gradual changes in the frequency spectrum, such as the appearance or growth of fault-specific frequencies, maintenance teams can address issues before they escalate into critical failures. This approach not only enhances machine reliability but also minimizes unnecessary maintenance activities, optimizing operational efficiency.

4.3 Heart Rate Variability Analysis

Heart Rate Variability (HRV) is a crucial metric for assessing general health and understanding the autonomic nervous system's function, reflecting the balance between the **sympathetic** and **parasympathetic** branches. By analyzing the variations in the intervals between successive heartbeats (**RR intervals**) in the frequency domain, valuable insights into an individual's physiological state, such as stress levels, relaxation, or overall cardiovascular health, can be obtained.

Figure 4B illustrates a HRV analysis plot for an individual with a heart rate of approximately 75 beats per minute (BPM). The time-domain plot demonstrates the fluctuation in RR intervals over time, which reflects the beat-to-beat variability in the heart rate. The corresponding frequency-domain plot highlights two dominant signal components: the Low-Frequency (LF) and High-Frequency (HF) bands. The LF component, centered around 0.1 Hz, represents the autonomic regulation of heart rate, influenced by both sympathetic and parasympathetic nervous system activity. The HF component, occurring near 0.25 Hz, is primarily associated with the parasympathetic nervous system, particularly respiratory sinus arrhythmia, which is the heart rate modulation linked to the respiratory cycle. Together, these components provide insight into the balance of autonomic nervous system functions and the physiological state of the individual.

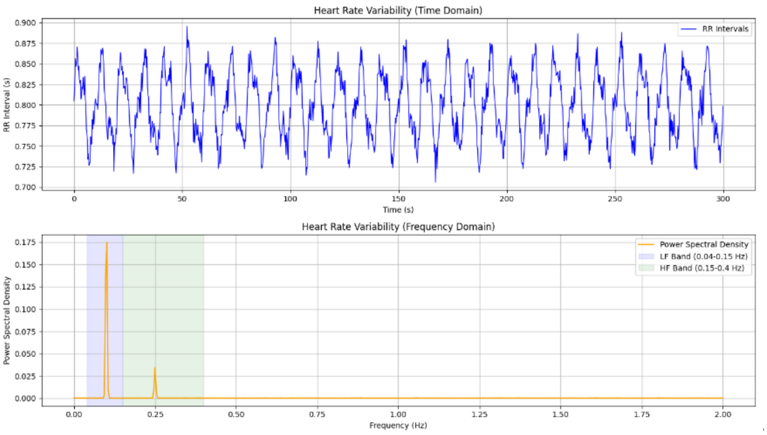

Figure 4B – HRV signal and its frequency domain equivalent

An HRV-based stress monitoring system is a tool designed to help users monitor and manage their stress levels, enhancing overall well-being. This system leverages the MAX30100 pulse oximeter sensor, which uses photoplethysmography (PPG) to detect changes in blood volume in the microvascular bed. Through this technology, the sensor captures heart rate and RR intervals (time difference between consecutive heartbeats), which are critical for HRV analysis, an established indicator of stress and autonomic nervous system function.

The system can be built using an Arduino Board, which acts as the central processing unit, interfacing with the MAX30100 sensor via the I2C protocol. This communication method ensures efficient and reliable data transmission between the sensor and the MCU. The Arduino Board collects the raw heart rate and RR interval data and preprocesses it by filtering out noise and artifacts using digital filtering techniques, such as low-pass filters, to ensure the integrity of the data.

Using the ArduinoFFT library, the time-domain RR intervals are transformed into the frequency domain to generate a power spectral density (PSD) plot.

The PSD is analyzed to extract the following key metrics:

- **Low-Frequency (LF) Band (0.04–0.15 Hz)**: Reflects a mix of sympathetic and parasympathetic activity, indicative of stress and general autonomic response.
- **High-Frequency (HF) Band (0.15–0.4 Hz)**: Associated with parasympathetic activity and relaxation, showing vagal tone.
- **LF/HF Ratio**: The balance between the LF and HF bands serves as an indicator of autonomic balance, with higher values pointing to stress dominance.

Consider a user who wants to monitor their stress levels during daily activities. The HRV system continuously tracks the RR intervals and computes the PSD in real-time. During a calm morning, the HF power is dominant, indicating relaxation. However, during a stressful work meeting, the LF power increases, and the LF/HF ratio rises, reflecting heightened sympathetic activity. These insights can guide the user to adopt stress-relief techniques such as deep breathing or meditation.

The system can also provide long-term health metrics, such as average daily LF/HF ratio trends, to highlight patterns that may require intervention. For example, consistently elevated LF power might suggest chronic stress, warranting lifestyle adjustments or professional consultation.

- The End -