Fourier Analysis for Harmonic Signals in Electrical Power Systems

A PROJECT REPORT

Submitted by

BL.EN.U4AIE21044----- D. Mohammad Abdulla

BL.EN.U4AIE21048------G. Tej Deep Reddy

BL.EN.U4AIE21050------ G. Mukesh Venkata sai

for the course 21MAT212-Mathematics for Intelligent Systems-4

Guided and Evaluated by

Dr. Sarada Jayan

Dept. of MATHS

AMRITA SCHOOL OF COMPUTING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

June – 2023

TABLE OF CONTENTS

ABSTRACT

Fourier analysis plays a crucial role in the analysis of harmonic signals in electrical power systems. Harmonics are unwanted electrical waveforms that result from the non-linear behavior of power system components, such as power electronic devices, and can lead to various issues, including increased losses, equipment overheating, and interference with sensitive equipment. Understanding and mitigating harmonics require a comprehensive understanding of their frequency content and characteristics, which can be effectively achieved through Fourier analysis.

This paper presents an overview of Fourier analysis techniques for harmonic signals in electrical power systems. The fundamental concepts of Fourier analysis, including Fourier series and Fourier transform, are discussed in the context of harmonic analysis. The paper also explores different methods for harmonic measurement and analysis, including spectral analysis, harmonic distortion analysis, and total harmonic distortion (THD) calculations. Furthermore, the importance of harmonic modeling and simulation is emphasized to assess the impact of harmonics on power system performance accurately.

The application of Fourier analysis in power quality assessment is discussed, highlighting its role in identifying and quantifying harmonic distortion in voltage and current waveforms. Moreover, the paper presents techniques for harmonic source identification, which is essential for effective harmonic mitigation strategies. The impact of harmonics on power system components, such as transformers, capacitors, and cables, is also addressed, emphasizing the significance of harmonic analysis in designing and operating power systems.

In conclusion, Fourier analysis provides a powerful tool for understanding and mitigating harmonics in electrical power systems. By accurately characterizing the frequency content of harmonic signals, power system engineers can identify the sources of harmonics, assess their impact on system performance, and devise effective strategies for harmonic mitigation.

INTRODUCTION

Electrical power systems are prone to the presence of harmonics due to the increasing use of non-linear loads, such as power electronic converters, in modern industrial, commercial, and residential environments. Harmonics are periodic waveforms with frequencies that are integer multiples of the fundamental frequency. These harmonics, if left unmitigated, can have detrimental effects on power system operation and quality.

Fourier analysis is a mathematical tool that allows the decomposition of complex waveforms into their constituent harmonics. By applying Fourier analysis techniques, power system engineers can accurately assess the frequency content, magnitude, and phase relationships of harmonic signals. This knowledge is crucial for understanding the impact of harmonics on power system performance and designing effective mitigation strategies.

The objective of this paper is to provide an overview of Fourier analysis techniques specifically tailored for harmonic analysis in electrical power systems. The paper will delve into the fundamentals of Fourier series and Fourier transform, explaining their applications in decomposing waveforms and extracting harmonic components. Various methods for measuring and quantifying harmonics will be explored, including spectral analysis, harmonic distortion analysis, and THD calculations.

Furthermore, the paper will discuss the importance of harmonic modeling and simulation for accurately assessing the impact of harmonics on power system components and overall system performance. The role of Fourier analysis in power quality assessment, harmonic source identification, and harmonic mitigation strategies will also be examined.

In summary, this paper aims to provide a comprehensive understanding of Fourier analysis in the context of harmonic signals in electrical power systems. By elucidating the theoretical foundations and practical applications of Fourier analysis, this work intends to facilitate the effective analysis and mitigation of harmonics, leading to improved power system performance and reliability.

Approach

The approach we have used for this project is Graph Theory, Probability Analysis for Reliability, Weibull Distribution, Fourier signals, for GUI we have used MATLAB.

Literature Survey

1. "Harmonic analysis and elimination in power systems: review of current methods and future trends" by Z. Antonić, D. Dujak, and M. G. Malović.
   - This paper provides a comprehensive review of various methods for harmonic analysis and elimination in power systems. It discusses the application of Fourier analysis and its limitations in handling non-stationary signals and time-varying harmonics. The paper also explores future trends and challenges in harmonic analysis.
2. "Advanced techniques for harmonic analysis in power systems" by S. Bhattacharya and K. D. Srivastava.
   - This paper discusses advanced techniques beyond the classical Fourier analysis for harmonic analysis in power systems. It explores techniques such as wavelet transform, time-frequency analysis, and higher-order spectra. The paper highlights the advantages and limitations of these techniques compared to Fourier analysis.
3. "Fourier analysis of power system harmonics using complex vector representation" by D. S. Wong, C. P. Yue, and M. L. Crow.
   - This paper presents a detailed analysis of power system harmonics using complex vector representation and Fourier analysis. It discusses the benefits of using complex vectors to analyze harmonics and presents case studies to demonstrate the application of Fourier analysis in power system harmonic analysis.
4. "Harmonic analysis of power systems with non-stationary signals using time-frequency methods" by A. Monti, F. Ponci, and F. Giulietti.
   - This paper focuses on the challenges of analyzing power system harmonics with non-stationary signals and explores the application of time-frequency methods, such as short-time Fourier transform and wavelet transform, for harmonic analysis. It discusses the advantages of time-frequency methods in capturing time-varying harmonics and presents case studies to illustrate their effectiveness.

Concepts Used

The programme extracts the time and signal values from a dataset after reading it from a CSV file. The sampling rate is then determined using the time values, and the signal values are subjected to an FFT.

The code starts by importing the required libraries: matplotlib.pyplot for charting, numpy for numerical computations, and pandas for data management.

Bringing in the Data:

The code reads data from a CSV file called "detect_dataset.csv" and stores it in a pandas DataFrame called "data" using the pd.read_csv() method.

The code then used Matplotlib to plot the signal's frequency-domain and time-domain representations. Sorting the FFT results reveals the prominent frequencies and their related magnitudes. The top five dominating frequencies are listed along with their magnitudes.**Key Concepts in Graph Theory:**

Explanation of code

1. Importing Libraries:
   - The code begins by importing the necessary libraries: pandas for data manipulation, numpy for numerical operations, and matplotlib.pyplot for plotting.
2. Loading the Data:
   - The code uses the pd.read_csv() function to read the data from a CSV file called 'detect_dataset.csv' and store it in a pandas DataFrame named data.
3. Extracting Time and Signal Values:
   - The code extracts the time and signal values from the DataFrame by accessing the specified columns (time_column and signal_column). It assigns these values to time_values and signal_values respectively.
4. Calculating Sampling Rate:
   - The sampling rate is calculated as the reciprocal of the time interval between consecutive time values. It is stored in the variable samplingRate.
5. Performing FFT:
   - The code applies the Fast Fourier Transform (FFT) algorithm to the signal values using the np.fft.fft() function. The result is stored in the fftSignal variable.

6. Generating Frequency Values:
    - The code generates the corresponding frequency values for the FFT result using the np.fft.fftfreq() function. It takes the length of the signal values (N) and the sampling rate as input parameters. The resulting frequencies are stored in the frequencies variable.
7. Plotting the Time Domain:
    - The code sets up a figure with two subplots using plt.subplot(). It then plots the time values against the signal values using plt.plot(). Additional formatting is done for the title, x-axis label, and y-axis label.
8. Plotting the Frequency Domain:
    - The code sets up the second subplot and plots the frequencies against the magnitudes of the FFT result using plt.stem(). This creates a stem plot to visualize the dominant frequencies. The title, x-axis label, and y-axis label are formatted accordingly.
9. Setting Frequency Range:
    - The code limits the x-axis range of the frequency plot to show frequencies up to half of the sampling rate. This is done using plt.xlim().
10. Adjusting Layout and Displaying the Plot:
    - The code adjusts the layout of the subplots using plt.tight_layout() to prevent overlapping. Finally, it displays the plot using plt.show().
11. Identifying Dominant Frequencies:
    - The code identifies the dominant frequencies and their corresponding magnitudes by sorting the absolute values of the FFT result using np.argsort(). The top 5 indices with the highest magnitudes are selected (dominant_indices).
12. Printing Dominant Frequencies and Magnitudes:
    - The code extracts the dominant frequencies and their magnitudes using the selected indices. It prints the dominant frequencies and their corresponding magnitudes using print() statements.

Fourier Signals: A mathematical method called Fourier analysis is used to break down periodic or non-periodic signals into their component frequencies. The early 19th-century French mathematician Jean-Baptiste Joseph Fourier is credited with developing the approach, which bears his name.

Implementation

First, some subsystems from the airplane are taken and connections for them are given as the inputs along with the sub-system working hours and reliability (taken randomly). Then using the digraph command in MATLAB a directed acyclic graph is generated.

1. The frequency analysis using the Fast Fourier Transform (FFT)
2. Ensure that you have a CSV file named 'detect_dataset.csv' with the appropriate data columns ('Ia' for time values and 'Va' for signal values). Place the file in the same directory as your Python script or notebook.
3. Make sure you have the necessary libraries installed: pandas, numpy, and matplotlib.
4. Copy and paste the code into your Python script or notebook.
5. Run the code.

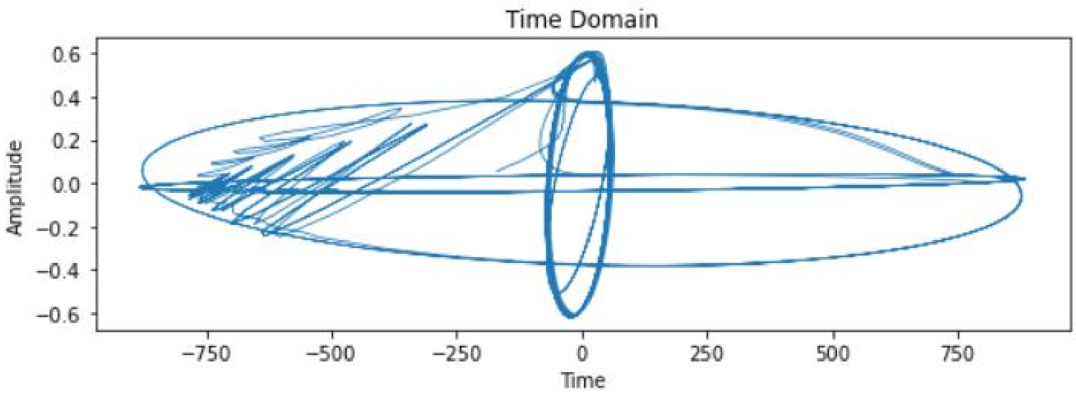
The CSV file, perform the FFT on the signal values, and plot the time domain and frequency domain representations of the signal. It will also identify the dominant frequencies and their magnitudes, which will be printed to the console.

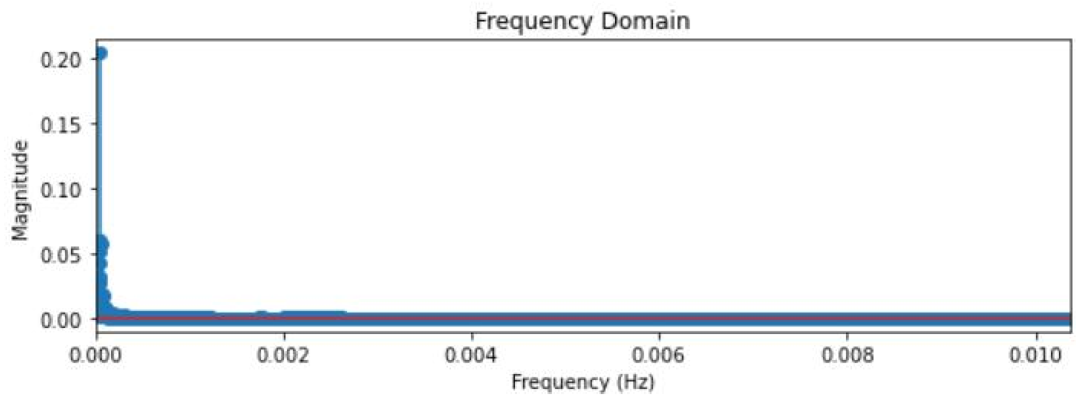
Code: Using Data set "detect_dataset.csv"

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('detect_dataset.csv')

time_column = 'Ia'
signal_column = 'Va'
time_values = data[time_column].values
signal_values = data[signal_column].values

samplingRate = 1 / (time_values[1] - time_values[0])
N = len(signal_values)
fftSignal = np.fft.fft(signal_values)
frequencies = np.fft.fftfreq(N, 1/samplingRate)

plt.figure(figsize=(8, 6))
plt.subplot(2, 1, 1)
plt.plot(time_values, signal_values, linewidth=0.8)
plt.title('Time Domain')
plt.xlabel('Time')
plt.ylabel('Amplitude')

plt.subplot(2, 1, 2)
plt.stem(frequencies, np.abs(fftSignal)/N, use_line_collection=True)
plt.title('Frequency Domain')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')

plt.xlim([0, samplingRate/2])

plt.tight_layout()
plt.show()
```

```
sorted_indices = np.argsort(np.abs(fftSignal))
dominant_indices = sorted_indices[-5:]
dominant_frequencies = frequencies[dominant_indices]
dominant_magnitudes = np.abs(fftSignal[dominant_indices]) / N

print('Dominant Frequencies:')
print(dominant_frequencies)
print('Corresponding Magnitudes:')
print(dominant_magnitudes)
```

OUTPUT






```
Dominant Frequencies:
[ 6.21884599e-05 -4.14589733e-05  4.14589733e-05 -5.18237166e-05
  5.18237166e-05]
Corresponding Magnitudes:
[0.05682713 0.05993828 0.05993828 0.20435762 0.20435762]
```

Without using Dataset : using value for better understanding

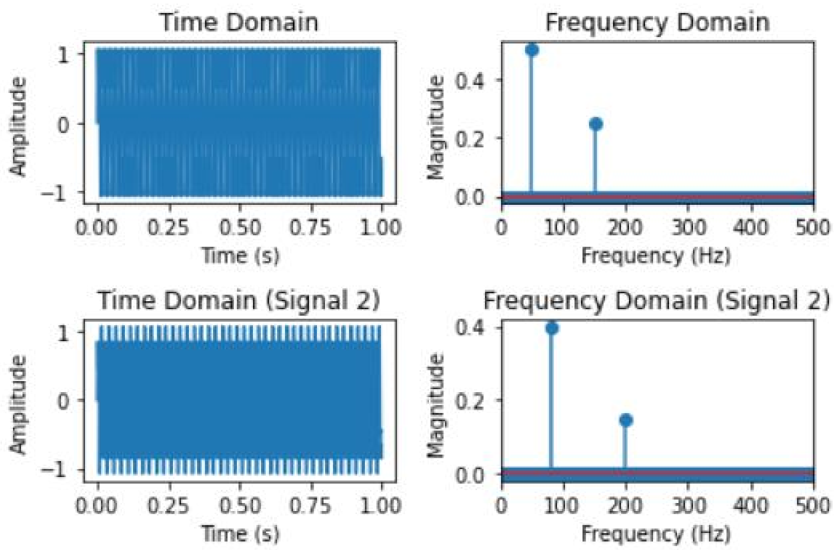
```
import numpy as np
import matplotlib.pyplot as plt

frequency = 50
amplitude = 1
harmonicFrequency = 150
harmonicAmplitude = 0.5
samplingRate = 1000
duration = 1

t = np.arange(0, duration, 1/samplingRate)
signal = amplitude*np.sin(2*np.pi*frequency*t) + harmonicAmplitude*np.sin(2*np.pi*harmonicFrequency*t)

N = len(signal)
fftSignal = np.fft.fft(signal)
frequencies = np.arange(0, N)*(samplingRate/N)

plt.subplot(2, 2, 1)
plt.plot(t, signal)
plt.title('Time Domain')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')

plt.subplot(2, 2, 2)
plt.stem(frequencies, np.abs(fftSignal)/N)
plt.title('Frequency Domain')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')

plt.xlim([0, samplingRate/2])

sortedIndices = np.argsort(np.abs(fftSignal)/N)[::-1]
dominantFrequencies = frequencies[sortedIndices[:5]]
dominantMagnitudes = np.abs(fftSignal[sortedIndices[:5]])/N
print('Dominant Frequencies:')
```

```
print(dominantFrequencies)
print('Corresponding Magnitudes:')
print(dominantMagnitudes)

frequency2 = 80
amplitude2 = 0.8
harmonicFrequency2 = 200
harmonicAmplitude2 = 0.3

signal2 = amplitude2*np.sin(2*np.pi*frequency2*t) + harmonicAmplitude2*np.sin(2*np.pi*harmonicFrequency2*t)

fftSignal2 = np.fft.fft(signal2)

plt.subplot(2, 2, 3)
plt.plot(t, signal2)
plt.title('Time Domain (Signal 2)')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')

plt.subplot(2, 2, 4)
plt.stem(frequencies, np.abs(fftSignal2)/N)
plt.title('Frequency Domain (Signal 2)')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')

plt.xlim([0, samplingRate/2])

sortedIndices2 = np.argsort(np.abs(fftSignal2)/N)[::-1]
dominantFrequencies2 = frequencies[sortedIndices2[:5]]
dominantMagnitudes2 = np.abs(fftSignal2[sortedIndices2[:5]])/N
print('Dominant Frequencies (Signal 2):')
print(dominantFrequencies2)
print('Corresponding Magnitudes (Signal 2):')
print(dominantMagnitudes2)
```

```
plt.tight_layout()
plt.show()
```

OUTPUT

```
Dominant Frequencies:
[ 50. 950. 850. 150. 486.]
Corresponding Magnitudes:
[5.00000000e-01 5.00000000e-01 2.50000000e-01 2.50000000e-01
 2.03458018e-15]
Dominant Frequencies (Signal 2):
[ 80. 920. 200. 800. 928.]
Corresponding Magnitudes (Signal 2):
[4.00000000e-01 4.00000000e-01 1.50000000e-01 1.50000000e-01
 1.98972442e-15]
```



Applications and Future Scope

The frequency analysis and FFT algorithm have numerous applications across various domains. Here are some common applications and potential future scope for further exploration:

1) Signal Processing: Frequency analysis is widely used in signal processing applications, such as audio and image processing. It helps in tasks like noise removal, filtering, compression, and feature extraction.

2) Audio and Music Analysis: FFT is commonly used in audio and music analysis to extract spectral features, identify musical notes, analyze harmonics, detect patterns, and classify audio signals.

3) Communication Systems: In telecommunications and wireless systems, frequency analysis is essential for tasks like modulation and demodulation, channel estimation, equalization, and spectrum sensing.

4) Vibrational Analysis: FFT is utilized for analyzing vibrations in mechanical systems. It helps in identifying natural frequencies, modes of vibration, and detecting faults or anomalies in machinery.

5) Power Systems: Frequency analysis is employed in power system analysis to identify harmonics, transient events, voltage fluctuations, and disturbances. It aids in power quality monitoring and fault detection.

6) Medical Signal Processing: FFT is used in analyzing medical signals such as electrocardiograms (ECG), electroencephalograms (EEG), and medical imaging. It assists in diagnosing conditions, detecting abnormalities, and monitoring patient health.

7) Machine Learning and Pattern Recognition: Frequency analysis techniques, combined with machine learning algorithms, can be used for pattern recognition, feature extraction, and classification tasks in various domains.

8) Future Scope: Advanced techniques like wavelet transforms, time-frequency analysis, and higher-order spectra offer further opportunities for exploring signal processing and analyzing signals with non-stationary properties.

9) Real-Time Applications: Implementing frequency analysis algorithms in real-time systems and embedded devices can enable applications like audio processing, sensor data analysis, and adaptive control systems.

10) Research and Innovation: Ongoing research focuses on developing more efficient and accurate algorithms for frequency analysis, exploring applications in emerging fields, and integrating frequency analysis with other analysis techniques for comprehensive signal processing.

These are just a few examples of the wide range of applications and potential future directions for frequency analysis and the FFT algorithm. The field continues to evolve with advancements in technology and interdisciplinary collaborations, offering exciting opportunities for exploration and innovation.

Conclusion

frequency analysis using the Fast Fourier Transform (FFT) algorithm is a powerful technique for analyzing signals in various domains. By transforming a signal from the time domain to the frequency domain, we can gain insights into the underlying frequencies and their magnitudes present in the signal.

Fourier analysis allows us to identify and quantify harmonic distortion, which occurs when non-sinusoidal components are present in the power system waveforms. It helps in assessing the quality of electrical power by measuring harmonic content and compliance with industry standards and regulations

Finally, the code generates and plots either a clean or distorted Fourier sine wave signal using the plotting features of MATLAB, depending on the system operation condition.

References

[1] Arrillaga, J.; Watson, N. R.; and Chen, S. Power System Quality Assessment. 2nd ed. University of Michigan: John Wiley & Sons, 2000; 2007. 300 p. DOI: 0471988650.

[2] Gonen T. Electrical Power Transmission System Engineering: Analysis and Design, 3rd ed. California State University, Sacramento: CRC Press; 2015. 1093 p. DOI: 1482232235.

[3] Dugui, W.; and Zheng, X. Harmonic model of power transformer. In: POWERCON '98. 1998 International Conference on, editor. Power System Technology, 1998. Proceedings; 18–21 Aug; Hangzhou, China. IEEE; 1998. DOI: 10.1109/ICPST.1998.729244.

[4] Greene, J. D.; and Gross, C. A. Nonlinear modelling of transformers. IEEE Transactions on Industry Applications. 1988; 24(3):434–438. DOI: 10.1109/28.2892