AGH
Akademia Górniczo-Hutnicza
im. Stanisława Staszica
w Krakowie

KRiM
Department of Robotics
and Mechatronics
krim.agh.edu.pl

Analysis and Implementation of Spectral Correlation Denisty and Spectral Coherence Density procedures

inż. Jakub Szczepaniak

June 2024

Mechatronic System Identification

Contents

Abstract

This report presents the implementation and analysis of Spectral Correlation Density (SCD) and coherence. Python functions were developed for both analyses, validated with test signals, and demonstrated to reveal key signal properties. The methods provide effective tools for signal analysis in mechatronic systems.

1 Introduction

1.1 Background and Motivation

Spectral analysis is crucial in mechatronic systems because it enables the identification and characterization of system dynamics, including resonant frequencies and vibration modes, which are essential for diagnosing faults and optimizing performance. By analyzing the frequency content of signals, engineers can detect and address issues such as imbalances, misalignment, and wear in mechanical components, leading to improved reliability and efficiency. Additionally, spectral analysis aids in the design of control systems by providing insights into system behavior and ensuring stability and precision in operation.

Spectral Correlation Density (SCD) and Spectral Coherence are specific parts of spectral analysis used in the analysis of cyclostationary signals. We need to analyze them because cyclostationary signals exhibit periodic variations in their statistical properties, which traditional spectral analysis methods cannot fully capture. By examining these signals, we can uncover hidden periodicities and correlations that provide deeper insights into system behavior, enhance fault detection, improve signal processing techniques, and optimize the performance of mechatronic systems.

1.2 Objectives

The main goals of the report are to understand SCD and Spectral Coherence Density and to implement functions for these analyses in Python. Python is chosen because it is an open-source, popular tool that can be integrated with MATLAB, facilitating future use in both of the most popular languages for signal analysis.

2 Theoretical Background

2.1 Basic Concepts of Cyclostationary Signals

Cyclostationarity refers to a characteristic of signals whose statistical properties, such as mean and autocorrelation, exhibit periodic variations over time. Unlike stationary signals, whose statistical properties are constant over time, cyclostationary signals have properties that vary cyclically. [1] This periodicity makes cyclostationary signals particularly relevant in various applications, including communications, mechanical systems, and radar signal processing.

The mean value of a cyclostationary signal is a periodic function of time.

$$E[x(t)] = E[x(t + T_0)]$$

where $E[x(t)]$ is the expectation or mean of the signal $x(t)$, and T_0 is the period.

The autocorrelation function of a cyclostationary signal is also periodic in time.

$$R_x(t, \tau) = E[x(t)x(t + \tau)] = R_x(t + T_0, \tau)$$

where $R_x(t, \tau)$ is the autocorrelation function, and τ is the time lag.

Cyclostationary signals exhibit correlations between different frequency components, which can be characterized using Spectral Correlation Density (SCD). This reveals periodicities in the frequency domain that are not apparent in traditional spectral analysis.

2.2 Examples of Cyclostationary Signals

Common examples of cyclostationary signals include:

- **Mechatronics**: Vibration signals from rotating machinery, where periodic rotations create cyclostationary patterns.
- **Communications**: Modulated signals such as AM, FM, and PSK, where the modulation process induces periodicities.

2.3 Spectral Analysis

Spectral analysis is a technique used to decompose a signal into its frequencies components, providing insights into its frequency content. Regular spectral analysis methods, such as Power Spectral Density (PSD), assume that the signal is stationary, meaning its statistical properties do not change over time. However, cyclostationary signals exhibit periodic variations in their statistical properties, making traditional PSD inadequate for capturing these variations. Consequently, specialized methods like SCD and Spectral Coherence are required to effectively analyze cyclostationary signals and reveal the periodicities.

2.4 Spectral Correlation Density (SCD)

Definition: The Spectral Correlation Density (SCD), sometimes also called the cyclic spectral density or spectral correlation function, is a function that describes the cross-spectral density of all pairs of frequency-shifted versions of a time-series. [2] The SCD function $S_x(f, \alpha)$ captures these correlations for all pairs of frequency-shifted versions of the signal. It provides a two-dimensional representation: one dimension for the actual frequency f and another for the cyclic frequency α.

Properties:

- SCD captures periodic correlations between frequency components.
- It can identify modulated components and their harmonics.
- SCD provides detailed insights into the structure of cyclostationary signals.

Applications:

- **Communications:** Detection and analysis of modulated signals (e.g., AM, FM, PSK).
- **Mechanical Systems:** Vibration analysis in rotating machinery to diagnose faults.
- **Radar and Sonar:** Feature extraction and target detection in received signals.

2.5 Spectral Coherence Density

Definition: Spectral coherence is a measure that quantifies the degree of correlation between two signals at different frequencies. It provides insight into the linear relationship between signals in the frequency domain. It can be used to determine the quality of response function. It also gives first intications of nonlinearity.

Properties:

- Spectral coherence ranges from 0 to 1, where 0 indicates no correlation and 1 indicates perfect correlation.
- It is a normalized measure, making it independent of the signal power.
- Spectral coherence can reveal phase and amplitude relationships between signals.

Applications:

- **Communications:** Analyzing the relationship between transmitted and received signals to detect interference and noise.
- **Neuroscience:** Studying the synchrony between different brain regions using EEG or MEG signals.
- **Mechanical Systems:** Monitoring and diagnosing faults in machinery by comparing vibration signals from different parts.

3 Methodology

3.1 Overview of the FFT Accumulation Method (FAM)

The FAM is an efficient technique for estimating the SCD. It was implemented based on work of Tomaž Šolc [3]. The method involves the following steps:

1. **Input Channelization:** The input signal is divided into overlapping segments using a sliding window approach. This segmentation allows for better frequency resolution and reduces spectral leakage.
2. **Windowing:** Each segment is multiplied by a window function, such as the Hamming window, to mitigate edge effects and reduce spectral leakage. The windowed segments are then normalized to maintain their energy.
3. **First FFT:** A Fast Fourier Transform (FFT) is applied to each windowed segment. This transforms the time-domain signal segments into the frequency domain, producing a series of frequency components for each segment.
4. **Complex Demodulation:** The frequency-shifted versions of the signal are generated by multiplying the FFT results by complex exponentials. This step shifts the frequency components to center them around zero frequency, which is crucial for analyzing cyclostationary signals.

5. **Conjugate Products and Second FFT:** For each pair of frequency components, the product of one component and the complex conjugate of the other is computed. These products are then averaged over all segments. A second FFT is applied to these averaged products to obtain the final SCD matrix.

6. **Frequency and Cyclic Frequency Vectors:** The normalized frequency and cyclic frequency vectors are generated to represent the axes of the SCD plot. The frequency vector is generated using the FFT frequency bins, and the cyclic frequency vector is derived similarly.

Advantages of FAM:

- **Computational Efficiency:** FAM leverages the efficiency of the FFT algorithm, making it suitable for real-time applications and large datasets.

- **Improved Frequency Resolution:** The use of overlapping segments and windowing enhances the frequency resolution and reduces spectral leakage, resulting in more accurate SCD estimates.

- **Detection of Cyclostationary Properties:** FAM is particularly effective for analyzing cyclostationary signals, as it captures periodic correlations in the frequency domain that are not detectable with traditional spectral analysis methods.

- **Flexibility:** FAM can be applied to a wide range of signals and is adaptable to different window lengths, overlap sizes, and FFT lengths, allowing for customization based on specific application requirements.

3.2 Coherence Algorithm Design

Coherence is a measure of the linear relationship between two signals as a function of frequency. It quantifies how well one signal can be predicted from another at different frequencies. The coherence calculation algorithm involves the following steps:

1. **Segmenting the Signals:** Both input signals are divided into overlapping segments using a sliding window approach. The overlap between segments typically ranges from 50% to 75%. This segmentation enhances frequency resolution and improves the statistical reliability of the coherence estimate.

2. **Windowing:** Each segment is multiplied by a window function, such as the Hamming or Hann window. Windowing reduces edge effects and spectral leakage, which can distort the coherence estimate. The window function is normalized to maintain the energy of the signal segments.

3. **Computing the FFT:** A Fast Fourier Transform (FFT) is applied to each windowed segment of both signals. This transforms the time-domain signal segments into the frequency domain, providing a series of frequency components for each segment.

4. **Calculating Cross-Spectral and Power Spectral Densities:**

    -

- **Cross-Spectral Density (CSD):** For each pair of corresponding segments from the two signals, the product of the FFT of one segment and the complex conjugate of the FFT of the other segment is computed. The average of these products across all segments provides the cross-spectral density.
- **Power Spectral Densities (PSD):** For each signal, the power spectral density is calculated by averaging the squared magnitude of the FFT across all segments.

5. **Calculating Coherence:** The coherence function is computed using the cross-spectral density and the power spectral densities. Mathematically, the coherence $C_{xy}(f)$ between two signals $x(t)$ and $y(t)$ at frequency f is given by:

$$C_{xy}(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)}$$

where $S_{xy}(f)$ is the cross-spectral density, and $S_{xx}(f)$ and $S_{yy}(f)$ are the power spectral densities of $x(t)$ and $y(t)$, respectively. The coherence values range from 0 to 1, where 0 indicates no linear relationship and 1 indicates a perfect linear relationship at that frequency.

Advantages of the Coherence Calculation Algorithm:

- **Identification of Frequency-Specific Relationships:** Coherence provides a frequency-dependent measure of the relationship between two signals, allowing for the identification of specific frequencies where the signals are strongly related.
- **Noise Reduction:** By averaging over multiple segments, the algorithm reduces the impact of noise and improves the reliability of the coherence estimate.
- **Applicability to Non-Stationary Signals:** The use of overlapping segments and windowing makes the algorithm suitable for analyzing non-stationary signals, where the statistical properties change over time.
- **Implementation Flexibility:** The algorithm can be easily implemented using standard signal processing libraries and is adaptable to different window lengths, overlap sizes, and FFT lengths, allowing for customization based on specific application requirements.

3.3 Implementation Details

The implementation is carried out in a Python programming environment, utilizing the following libraries: `numpy` for numerical computations, `scipy` for signal processing functions, and `matplotlib` for plotting results. The code is organized into three modules:

1. **signal_generation.py:** This module contains functions for generating various test signals. These functions are used to create input signals for testing the SCD and coherence calculations.
2. **analysis.py:** This module includes the implementations of the SCD and coherence functions. It also contains functions for displaying the results, such as plotting the SCD and coherence graphs.

3. **testing.py:** This script is used to perform all testing of the implementations. It calls functions from the `signal_generation.py` and `analysis.py` modules to generate test signals, compute SCD and coherence, and display the results.

This structured approach ensures modularity and ease of testing, allowing for efficient development and validation of the signal processing algorithms.

4 Results and Analysis

4.1 Test Signals

The test signals used in this implementation are designed to validate the accuracy and effectiveness of the Spectral Correlation Density (SCD) and coherence calculations.

- **Cyclostationary Signal:** This signal consists of sinusoidal components modulated by a low-frequency cosine wave. The generated signal includes frequencies at 50 Hz and 120 Hz, with modulation at 5 Hz.
- *Expected Outcome:* The SCD plot should reveal peaks at the fundamental frequencies (50 Hz and 120 Hz) and their sidebands (45 Hz, 55 Hz, 115 Hz, and 125 Hz), indicating the presence of cyclostationarity. See Figure 1 for a plot of the generated cyclostationary signal.
- **Coherence Test Signals:** Two signals are generated with a known linear relationship. The first signal is a pure sinusoid with added noise, and the second signal is a phase-shifted version of the first signal with added noise.
- *Expected Outcome:* The coherence plot should show high coherence (close to 1) at the frequency of the sinusoid, indicating a strong linear relationship between the two signals at that frequency. See Figure 2 for plots of the generated signals for coherence testing.

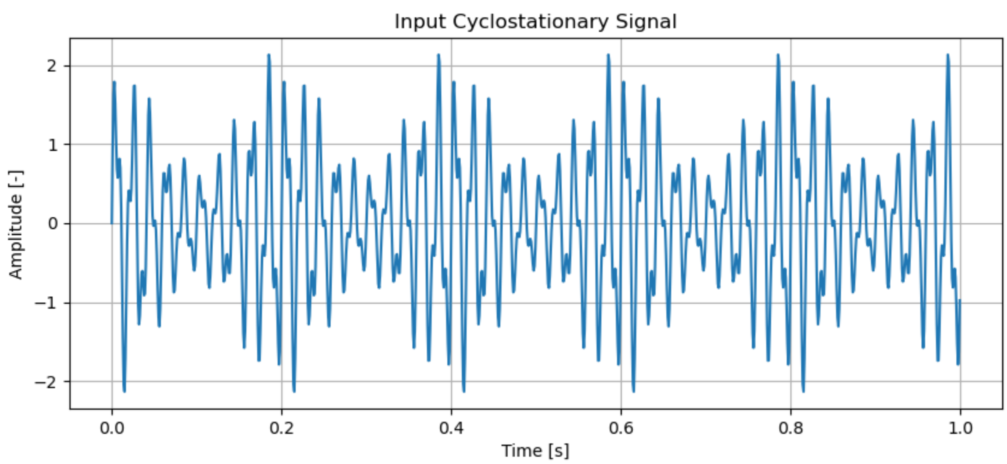

Figure 1: Generated cyclostationary signal.

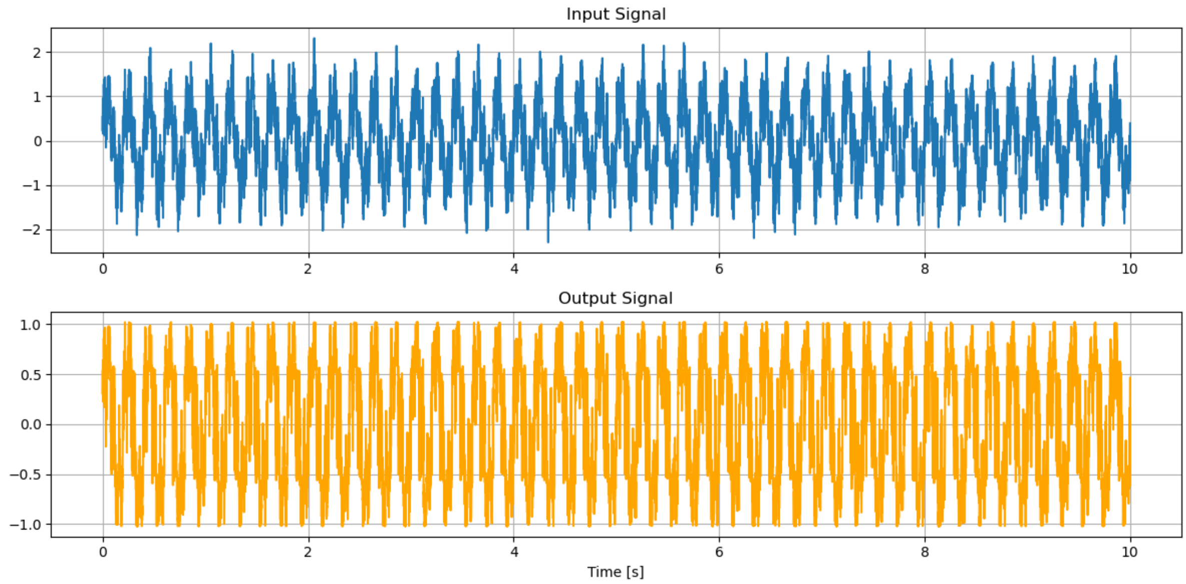

Figure 2: Generated signals for coherence testing.

4.2 SCD Results

The SCD graph (Figure 3) provides insights into the spectral properties of the cyclostationary signal generated. The graph has the frequency axis (y-axis) ranging from -1.5 to -0.5 and the cyclic frequency axis (x-axis) ranging from -1.0 to 0 . Color intensity indicates the magnitude of spectral correlation, with the scale shown on the right.

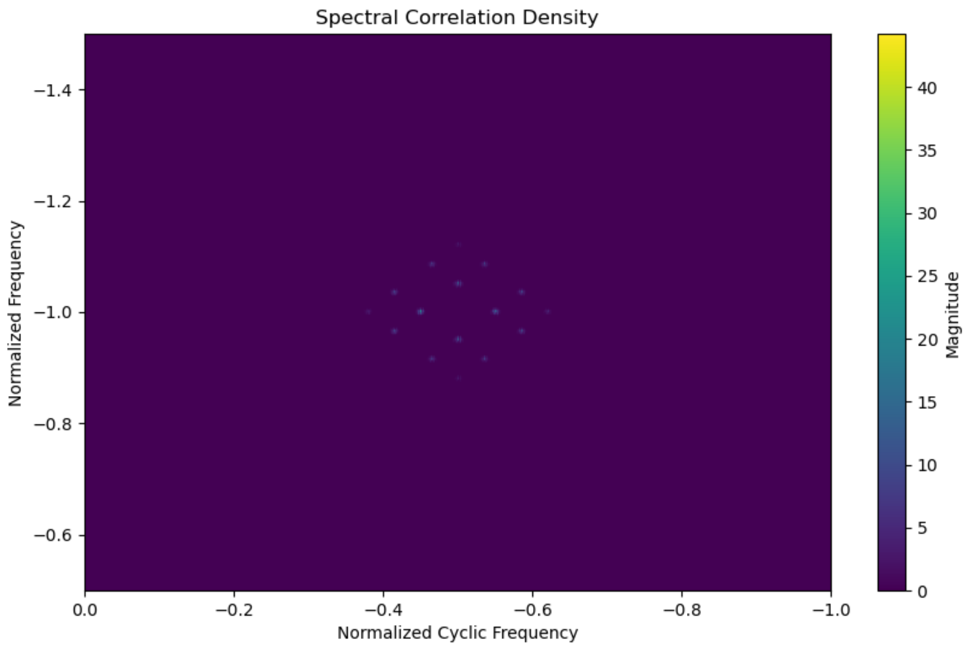

Figure 3: Results of performing SCD on signal mentioned in section 4.1.

The SCD graph shows several bright yellow peaks arranged in a grid-like pattern, indicating strong spectral correlations at specific frequency and cyclic frequency pairs. These peaks correspond to the frequencies of the sinusoids (50 Hz and 120 Hz) and their modulated components with a modulation frequency of 5 Hz. The peaks aligned along diagonal lines suggest periodic modulation at specific frequencies.

Key observations include the center peak near -1.0 Hz, -0.5 Hz representing the fundamental modulation frequency, and adjacent peaks indicating harmonics and combinations of carrier frequencies and the modulation frequency. The presence of these peaks signifies strong correlations characteristic of cyclostationary signals, with the grid pattern indicating multiple frequency components modulated by a common frequency.

4.3 Spectral Coherence Density Results

The figure 4 displays two plots representing the coherence between two signals as a function of frequency.

The top plot shows the coherence calculated using the `scipy.signal.coherence` function. The coherence is close to 1 at very low frequencies, indicating a strong linear relationship between the signals in this range. A noticeable dip in coherence occurs at a certain frequency, suggesting a weaker linear relationship at this frequency. After this dip, the coherence stabilizes and remains relatively high (around 0.6 to 0.8) across a broad range of frequencies, indicating a moderate to strong linear relationship.

The bottom plot depicts the coherence calculated using a manual implementation. The results are consistent with those obtained using `scipy.signal.coherence`.

Overall, both methods show similar coherence profiles, validating the manual implementation.

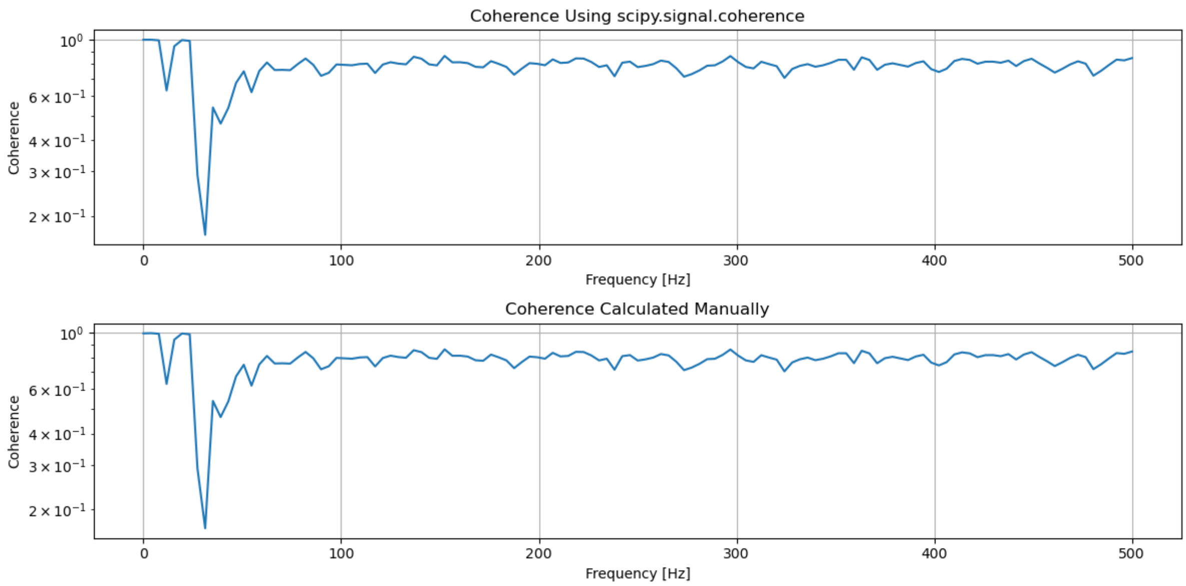

Figure 4: Results of performing coherence on signal mentioned in section 4.1.

5 Discussion

This report presents a comprehensive analysis of SCD and coherence for signal processing. We implemented functions for calculating SCD and coherence in Python, leveraging libraries such as `numpy` and `scipy`, and demonstrated their effectiveness through test signals. The SCD analysis revealed key periodicities and correlations within the frequency domain, essential for characterizing cyclostationary signals. Coherence analysis, highlighted the linear relationship between signals across different frequencies. The consistency between the manual and `scipy`-based coherence calculations validated the implementation. The modular code structure, with separate modules for signal generation, analysis, and testing, ensures flexibility and ease of use. Overall, the methods and results discussed provide valuable tools for signal analysis in mechatronic systems, enhancing our ability to diagnose and optimize system performance.

References

[1] Antonio Napolitano. 1 - characterization of stochastic processes. In Antonio Napolitano, editor, *Cyclostationary Processes and Time Series*, pages 3–35. Academic Press, 2020.

[2] W. Gardner. Measurement of spectral correlation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(5):1111–1123, 1986.

[3] Tomaž Šolc, Mihael Mohorčič, and Carolina Fortuna. A methodology for experimental evaluation of signal detection methods in spectrum sensing. *PLOS ONE*, 13(6), June 2018.

Appendices

Full Code Listings

Full code can be found at https://github.com/jakub-scpnk/mechatronics-system-identification-project