Minnesota State University, Mankato

Electrical and Computer Engineering Department

Lab Assignment 3 – Correlation and Autocorrelation Analysis of Continuous-Time Signals

Date of experiment: 02/04/2025

Date of Submission: 02/11/2025

Student name

Nathnael Minuta

TA name

Masamune Oso

EE 363-01 Communication Systems Laboratory

Dr. Yikai Li

Spring 2025

Contents

                                                                  Pages

Abstract ……………………………………………………………………………. 3

Purpose…………………………………………..………………………………… 3

Materials…………………………………………..………………………………  3

Procedure, Data Collection, and Analysis……….…………………   3

Conclusion…………………………………………..……………………………   5

Appendices…………………………………………..……………………………  6

References…………………………………………..……………………………   10

Correlation and Autocorrelation Analysis of Continuous-Time Signals

*Abstract*

In this lab, we explored the fundamentals of correlation and autocorrelation analysis for continuous-time signals using Python. We implemented functions to calculate the Pearson-type correlation coefficient between pairs of signals by approximating continuous integrals with discrete summations, and we applied these functions to compare numerical results with analytical predictions. Additionally, we computed the autocorrelation function of a composite sinusoidal signal through numerical integration to examine its periodic features and inherent time-lagged dependencies. By varying the sampling time step, we investigated its impact on the accuracy of numerical approximations. This exercise demonstrated the practical application of Python in digital signal processing, bridging theoretical concepts with computational implementations, and provided deeper insights into signal similarity and periodic behavior.

*Materials*

**Equipment Needed                                        Quantity.**

Laptop (Visual Studio Code)                                 1

**Methods (Procedure, Data Collection, and Analysis)**

**Part A (Task 1 (Correlation Coefficient))**

1. Develop a Python function correlation\_coefficient(x, y, dt) that computes the Pearson-type correlation coefficient by:

* Calculating the mean of each signal.
* Approximating the continuous integrals using discrete summation (i.e., summing products multiplied by the time step dt).

1. Collect data by sampling signals (e.g., constant signals and sinusoidal variations) over a defined interval.
2. Compute the correlation coefficients between a reference signal and several test signals, including self-correlation.
3. Analyze the effect of varying the time step dt by comparing numerical results with analytical predictions.

**Part B: Task 2 (Append the Parity Bit)**

1. Create a Python function compute\_autocorrelation(x, t, tau\_vals) that:

* Determines the time step dt from the time vector.
* Computes the autocorrelation Rx(τ) for each lag τ by numerically integrating the product x(t)⋅x(t−τ) over the overlapping intervals.

1. Collect data by generating a composite sinusoidal signal x(t)=sin(2π×5t)+0.5sin(4π×5t) over 0 to 2 seconds with a 1000 Hz sampling rate.
2. Calculate Rx(τ) for lags in the range [−0.5,0.5] seconds.
3. Analyze by plotting both the original signal and its autocorrelation function to observe periodic characteristics and verify the numerical integration.

*Results*

A black screen with yellow text

AI-generated content may be incorrect.

A graph of a graph

AI-generated content may be incorrect.

*Conclusion*

In this lab, we successfully implemented and analyzed the principles of correlation and autocorrelation for continuous-time signals using Python. Specifically, we developed functions to compute the Pearson-type correlation coefficient and to calculate the autocorrelation function via numerical integration. These functions demonstrated how discrete approximations can effectively capture signal similarities and inherent periodic features, even when working with sampled data. Challenges included accurately approximating continuous integrals with discrete summations and handling edge effects during signal shifts in the autocorrelation computation. By overcoming these challenges, we deepened our understanding of how Python can be applied to implement fundamental signal processing concepts and analyze the impact of sampling parameters on numerical accuracy. This exercise reinforced both our programming skills and our theoretical knowledge of signal analysis in digital systems.

*Appendices*

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

*A screenshot of a computer program

AI-generated content may be incorrect.*

*A screen shot of a computer program

AI-generated content may be incorrect.*

*A screen shot of a computer program

AI-generated content may be incorrect.*

*References*

1. <https://rodzah.files.wordpress.com/2011/07/how-to-write-lab-report.pdf>
2. EE343- LAB Assignment 3.pdf