Digital Signal Processing

Experiment Report



**Convolution**

Submitted by:

|  |  |
| --- | --- |
| Owais Rao | 22L-7638 |

Submitted to:

Dr. Muhammad Akmal

March 28, 2025

Department of Electrical Engineering

National University of Computer and Emerging Sciences, Lahore

## Introduction

1. **Problem Analysis**

## Design Requirements

## Possible Solutions

1. **Preliminary Design**

## Design Description

## Experimental Results

## Software Simulation

1. **Performance Analysis**
2. **Conclusion**

In digital signal processing, template matching is a crucial technique used to detect the presence of predefined patterns within a given signal. This experiment aims to explore the application of convolution operations for template matching, where known signals are identified within a combined signal.

Convolution is widely used in signal processing for pattern recognition, filtering, and feature extraction. By convolving a template signal with a composite signal, we can determine the locations where the template appears within the signal. This method is highly effective for detecting periodic or structured signals in various domains, such as speech processing, biomedical signal analysis, and communication systems.

In this experiment, we generate two sinusoidal signals, x1(n) and x2(n), in which fo is the fundamental frequency and Φ\Phi is the phase shift. These signals are concatenated to form a composite signal x(n). Using MATLAB, we perform convolution-based template matching to identify the presence and locations of x1(n) and x2(n) within x(n).

Additionally, the experiment examines the effect of noise and phase variations on template matching performance. Gaussian noise is introduced at different levels (2 dB and -2 dB) to assess the robustness of the method. By analyzing the results, we gain insights into the challenges of real-world signal detection and filtering techniques.

The experiment provides a hands-on approach to understanding convolution in signal processing and demonstrates its practical applications in detecting and extracting useful information from signals.

In many real-world applications, signals are often composed of multiple overlapping components, making it challenging to identify individual patterns within them. This experiment addresses the problem of detecting known sinusoidal signals within a combined signal using convolution-based template matching.

The core issue lies in accurately extracting predefined signal components, especially when noise is introduced. In practical scenarios, signals are rarely isolated and are often affected by external interferences, such as background noise in audio signals, sensor noise in biomedical applications, or channel noise in communication systems. Therefore, it is crucial to develop techniques that can reliably detect target signals even under noisy conditions.

To break down the problem further:

* We have two known signals, a sine wave x1(n) and a cosine wave x2(n), which are concatenated to form a composite signal x(n).
* The objective is to verify whether x1(n) and x2(n) exist within x(n) by using convolution.
* The experiment also introduces Gaussian noise at different levels (2 dB and -2 dB) to evaluate the effect of noise on detection accuracy.
* The phase of the signal is varied to analyze how phase shifts impact the convolution results.

A major challenge in this problem is distinguishing the desired signal components from noise and other irrelevant information. By applying convolution, we can measure the similarity between the composite signal and the known templates, helping us determine whether a match exists. However, convolution performance may degrade when noise is present, requiring additional filtering techniques to enhance accuracy.

This experiment aims to provide insights into signal processing techniques for template matching, noise resistance, and the impact of phase variations. Understanding these concepts is essential for applications such as speech recognition, radar signal processing, and biomedical signal analysis.

The design of this experiment involves generating and processing signals to perform template matching using convolution. The following requirements outline the key steps and constraints of the experiment:

1. **Generation of Sinusoidal Signals x1(n) and x2(n)**
   * Create two discrete-time sinusoidal signals:
     + - x1(n)=sin(2πf0t+Φ)
       - x2(n)=cos(2πf0t+Φ)
   * The fundamental frequency f0 determines the oscillation rate of the signals.
   * The phase shift Φ\Phi is introduced to study its effect on convolution-based template matching.
   * The signals are generated over a specified duration with a given sampling frequency fs.
2. **Combination of Signals into a Single Composite Signal x(n)**
   * The generated signals are concatenated to form a combined signal: x(n)=[x1(n) x2(n)]
   * This combined signal represents a scenario where multiple components exist within a signal, mimicking real-world applications such as audio signals containing multiple frequencies.
3. **Application of Template Matching Using Convolution**
   * Convolution is used to detect whether x1(n) and x2(n) exist within the composite signal x(n).
   * The convolution operation is performed as follows:
     + - conv\_x1(n)=x(n)∗x1(n)
       - conv\_x2(n)=x(n)∗x2(n)
   * This step helps in measuring the similarity between the composite signal and the reference signals.
4. **Analysis of Results Under Different Noise Conditions**
   * Gaussian noise is added to the composite signal to evaluate how noise impacts template matching.
   * Two different noise levels are considered:
     + - **2 dB Noise**: A moderate noise level is added, simulating a real-world noisy environment.
       - **-2 dB Noise**: A higher noise level is introduced to test the robustness of convolution-based detection.
   * The experiment analyzes how well convolution can identify the template signals in noisy conditions.

These requirements ensure that the experiment effectively demonstrates convolution-based template matching, the impact of noise on signal detection, and the role of phase variations in signal processing.

The problem of detecting known signals within a composite signal can be approached using several techniques. The primary objective is to accurately identify the presence of predefined signal patterns while ensuring robustness against noise. Below are three possible solutions that could be applied:

#### **1. Convolution-Based Template Matching**

* Convolution is a widely used technique in signal processing for detecting patterns. By convolving the composite signal x(n) with a known template x1(n) or x2(n), we can measure the similarity between the two signals.
* The convolution operation is defined as:

y(n)=x(n)∗h(n)=

where h(n) is the template signal.

* Peaks in the convolution result indicate regions where the template closely matches the composite signal.
* **Advantages:**
  + Simple to implement using MATLAB’s conv() function.
  + Effective in detecting signals with minimal noise.
* **Disadvantages:**
  + Performance can degrade in high-noise environments.
  + Sensitive to phase shifts, which can affect detection accuracy.

#### **2. Cross-Correlation for Pattern Recognition**

* Cross-correlation is another effective technique for template matching, often used in signal and image processing.
* It measures the similarity between two signals as a function of the time shift between them.
* Unlike convolution, cross-correlation is better suited for detecting time shifts between signals. It provides a direct measure of similarity and is less sensitive to phase variations.
* **Advantages:**
  + More robust to phase differences compared to convolution.
  + Can accurately determine shifts in signal occurrences.
* **Disadvantages:**
  + Requires additional computation compared to convolution.
  + Still affected by noise, though less than convolution.

### **Chosen Solution**

For this experiment, convolution-based template matching is used because it provides a straightforward way to detect known signals in a composite signal.

|  |
| --- |
| Generate sinusoidal signals x₁(n) and x₂(n) |
| ↓ |
| Combine them into a single composite signal x(n) |
| ↓ |
| Combine them into a single composite signal x(n) |
| ↓ |
| Introduce Gaussian noise at different levels |
| ↓ |
| Analyze the results and compare noise-free vs. noisy cases |

The experiment is designed to implement and analyze template matching using convolution in digital signal processing. A structured approach is followed to ensure systematic signal generation, processing, and evaluation. The key steps involved in the design are outlined below:

#### **1. Generate Signals** x1(n) **and** x2(n)

* + Two sinusoidal signals are generated using MATLAB:
* x1(n)=sin(2πf0n+Φ)
* x2(n)=cos(2πf0n+Φ)
  + The sampling frequency fsf\_sfs​ is selected to ensure proper signal representation.
  + The phase Φ\PhiΦ is varied to study its impact on template matching.

#### **2. Combine Signals into** x(n)x(n)x(n)

* The two signals x1(n)x\_1(n)x1​(n) and x2(n)x\_2(n)x2​(n) are concatenated to form a single composite signal x(n)x(n)x(n).
* This simulates a real-world scenario where multiple components exist in a signal.
* The composite signal is plotted for visualization.

#### **3. Detect Templates Using Convolution**

* The goal is to check if x1(n)x\_1(n)x1​(n) and x2(n)x\_2(n)x2​(n) can be identified within x(n)x(n)x(n).
* The convolution operation is applied as follows:

y1(n)=x(n)∗x1(n) & y2(n)=x(n)∗x2(n)

* Peaks in the output indicate the presence of x1(n)x\_1(n)x1​(n) and x2(n)x\_2(n)x2​(n) within x(n)x(n)x(n).
* MATLAB’s conv() function is used to perform the convolution.

#### **4. Analyze Performance With/Without Noise**

* The effect of noise on template detection is analyzed by adding Gaussian noise to the composite signal.
* Different noise levels (2 dB and -2 dB) are introduced to test the robustness of convolution-based detection.
* The results are compared for noise-free and noisy cases.

**TASK 1**

fs = 1000;

t = 0:1/fs:1;

fo = 5;

phi = pi/4;

x1 = sin(2\*pi\*fo\*t + phi);

x2 = cos(2\*pi\*fo\*t + phi);

x = [x1 x2];

conv\_x1 = conv(x, x1, 'same');

conv\_x2 = conv(x, x2, 'same');

subplot(3,1,1); plot(x); title('Concatenated Signal x(n)');

subplot(3,1,2); plot(conv\_x1); title('Convolution with x1(n)');

subplot(3,1,3); plot(conv\_x2); title('Convolution with x2(n)');

**TASK 2**

fs = 1000;

t = 0:1/fs:1;

fo = 5;

phi = pi/4;

x1 = sin(2\*pi\*fo\*t + phi);

x2 = cos(2\*pi\*fo\*t + phi);

x = [x1 x2];

x\_noisy\_2 = awgn(x, 2, 'measured');

x\_noisy\_neg2 = awgn(x, -2, 'measured');

conv\_x1\_2 = conv(x\_noisy\_2, x1, 'same');

conv\_x2\_2 = conv(x\_noisy\_2, x2, 'same');

conv\_x1\_neg2 = conv(x\_noisy\_neg2, x1, 'same');

conv\_x2\_neg2 = conv(x\_noisy\_neg2, x2, 'same');

subplot(2,3,1); plot(x\_noisy\_2); title('Signal with 2 dB Noise');

subplot(2,3,2); plot(conv\_x1\_2); title('Convolution with x1(n) (2 dB)');

subplot(2,3,3); plot(conv\_x2\_2); title('Convolution with x2(n) (2 dB)');

subplot(2,3,4); plot(x\_noisy\_neg2); title('Signal with -2 dB Noise');

subplot(2,3,5); plot(conv\_x1\_neg2); title('Convolution with x1(n) (-2 dB)');

subplot(2,3,6); plot(conv\_x2\_neg2); title('Convolution with x2(n) (-2 dB)');

**TASK 3**

fs = 1000;

t = 0:1/fs:1;

fo = 5;

x1\_no\_phi = sin(2\*pi\*fo\*t);

x2\_no\_phi = cos(2\*pi\*fo\*t);

x\_nophi = [x1\_no\_phi x2\_no\_phi];

conv\_x1\_nophi = conv(x\_nophi, x1\_no\_phi, 'same');

conv\_x2\_nophi = conv(x\_nophi, x2\_no\_phi, 'same');

subplot(3,1,1); plot(x\_nophi); title('Concatenated Signal x(n) Without Phase');

subplot(3,1,2); plot(conv\_x1\_nophi); title('Convolution with x1(n) Without Phase');

subplot(3,1,3); plot(conv\_x2\_nophi); title('Convolution with x2(n) Without Phase');

**TASK 4**

fs = 1000;

t = 0:1/fs:1;

fo = 5;

phi\_custom = pi/6;

x1\_custom = sin(2\*pi\*fo\*t + phi\_custom);

x2\_custom = cos(2\*pi\*fo\*t + phi\_custom);

x\_custom = [x1\_custom x2\_custom];

conv\_x1\_custom = conv(x\_custom, x1\_custom, 'same');

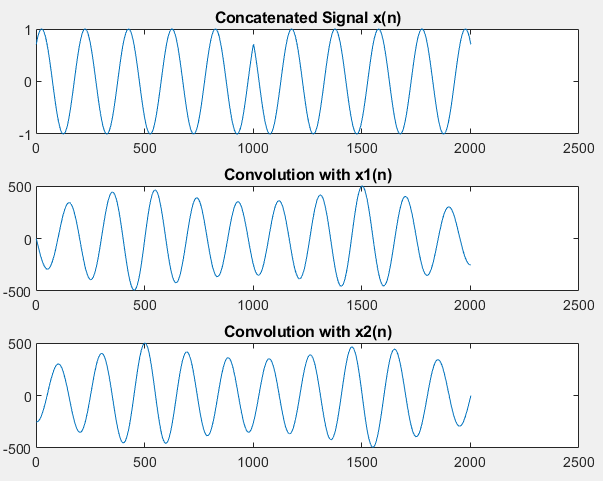
conv\_x2\_custom = conv(x\_custom, x2\_custom, 'same');

subplot(3,1,1); plot(x\_custom); title('Concatenated Signal x(n) With Custom Phase');

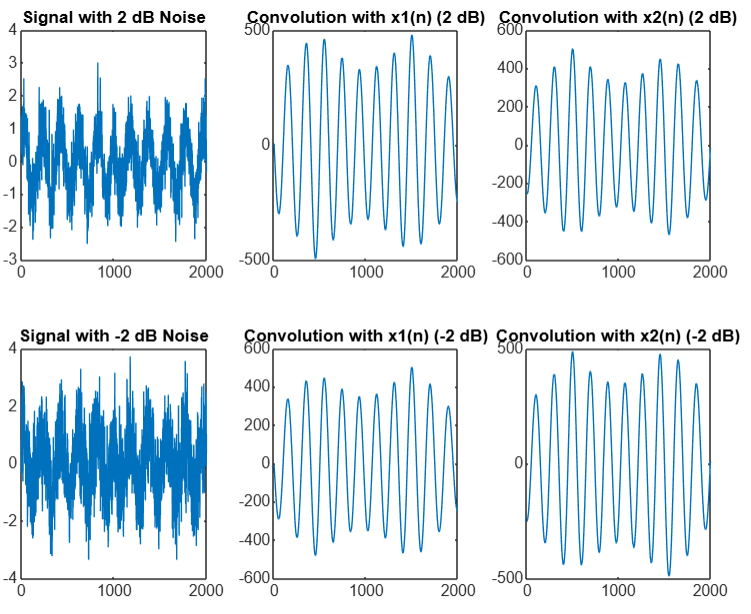
subplot(3,1,2); plot(conv\_x1\_custom); title('Convolution with x1(n) With Custom Phase');

subplot(3,1,3); plot(conv\_x2\_custom); title('Convolution with x2(n) With Custom Phase');

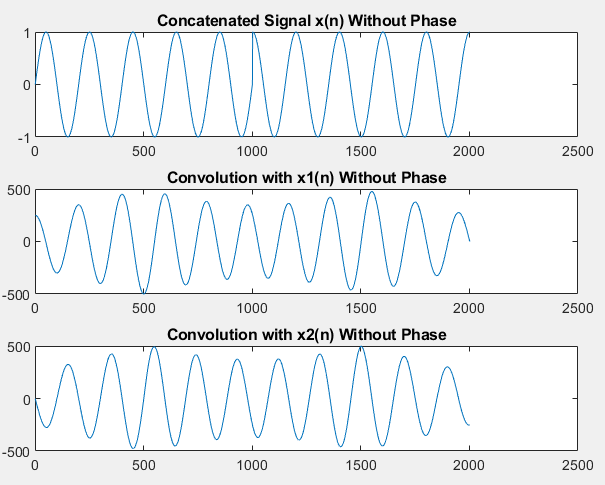
**TASK 1**

****

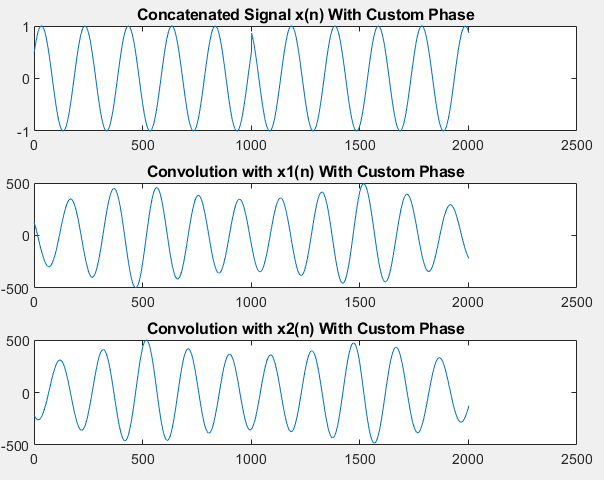
**TASK 2**



**TASK 3**



**TASK 4**



The performance of convolution-based template matching is evaluated under different conditions, including noise-free and noisy environments. The analysis focuses on how well the method can detect predefined signals and how its accuracy is affected by noise and phase shifts.

#### **1. Detection in Noise-Free Conditions**

* When no noise is added to the composite signal, convolution successfully detects both x1(n) and x2(n).
* The convolution outputs exhibit clear peaks at locations where the templates match the composite signal.
* This demonstrates that convolution is a reliable method for template matching in an ideal scenario.

#### **2. Performance Under Noise**

* Gaussian noise is introduced at two levels: **2 dB noise** (moderate) and **-2 dB noise** (high).
* In the presence of **2 dB noise**, the peaks in the convolution output are still distinguishable, but the signal detection is less clear due to interference.
* With **-2 dB noise**, the noise level is high, significantly distorting the signal and making detection difficult. Peaks become less prominent, and false positives may appear.
* This indicates that the convolution method is **sensitive to noise**, and additional filtering may be required to enhance detection accuracy in real-world applications.

#### **3. Impact of Phase Shifts**

* The experiment also examines the effect of phase variations on template matching.
* A shift in phase changes the alignment of the sinusoidal signals, affecting convolution results.
* If the template signal has a different phase than the signal in x(n), the convolution peaks may shift or reduce in magnitude.
* This suggests that convolution-based template matching is **highly dependent on phase alignment**, making it less robust in cases where the phase of the signal is unknown.

#### **4. Computational Efficiency**

* The convolution operation is computationally efficient and executes quickly in MATLAB.
* However, in cases with large datasets or real-time processing, cross-correlation might be a better alternative for improved robustness.

This experiment successfully demonstrates the application of **convolution-based template matching** for detecting predefined signal patterns within a composite signal. Through a structured approach, the signals were generated, combined, and analyzed using convolution to determine their presence.

Key findings from the experiment include:

* **Convolution effectively detects templates** in noise-free conditions, showing clear peaks when the templates match the composite signal.
* **Noise significantly impacts detection accuracy**, with higher noise levels making it difficult to distinguish signal components.
* **Phase shifts affect convolution results**, as the method relies on precise alignment between the template and the composite signal.
* **Computational efficiency is high**, making convolution a feasible method for simple template matching applications.

However, the study also highlights the **limitations of convolution-based template matching**, particularly in the presence of noise and phase variations. Future improvements could include **cross-correlation techniques for better phase invariance** and **noise filtering methods to enhance signal detection in noisy environments**.

This experiment provides valuable insights into the practical challenges of signal processing and reinforces the importance of choosing the right detection method based on noise levels and phase considerations.