

# REPORT

Course: Analog and Digital Electronic Circuits

Teacher: Prof. Dr. Hab. Vasyl Martsenyuk

Lab No. 1 and 2

Date: January 18, 2025

Topic: "Figures demonstrating Furier transform "

Variant: 8

Name: Daria Krecichwost

Computer Science (Second Degree)

Full-time studies, Semester 1

Group: B

# Signal Synthesis Using Inverse Discrete Fourier Transform (IDFT)

Daria Krecichwost

January 18, 2025

## Abstract

In this report, we discuss the process of synthesizing discrete-time signals from their Discrete Fourier Transform (DFT) coefficients using the Inverse Discrete Fourier Transform (IDFT). The IDFT is computed in matrix notation for different sequences, and the resulting time-domain signals are visualized.

## 1 Problem Statement

The task involves synthesizing discrete-time signals from their DFT coefficients using the Inverse Discrete Fourier Transform (IDFT) in matrix notation. Each signal is represented as a sequence of coefficients  $\mathbf{X}_\mu$ , which is transformed into the corresponding time-domain signal using the IDFT formula. The goal is to compute the IDFT for different sequences of varying lengths  $N$  and visualize the results.

## 2 Input Data

The input data consists of multiple sequences of coefficients  $\mathbf{X}_\mu$ , for which we need to calculate the corresponding time-domain signals using the IDFT. For example, one of the sequences for variant 8 is:

$$\mathbf{x}_\mu = [6, 2, 4, 4, 4, 5, 0, 0, 0, 0]$$

For each task, the sequence will vary, and we need to apply the IDFT transformation to reconstruct the time-domain signal.

## 3 Commands Used (or GUI)

### 3.1 a) Source Code

The following Python code was used to synthesize signals from the input DFT coefficients using IDFT:

```

import numpy as np
import matplotlib.pyplot as plt

# Function to generate the IDFT matrix
def idft_matrix(N):
    W_N = np.exp(-2j * np.pi * np.outer(np.arange(N), np.arange(N)) / N)
    return W_N

# Function to synthesize the signal using IDFT
def synthesize_signal(X_mu):
    N = len(X_mu)
    W_N = idft_matrix(N) # Compute the IDFT matrix
    x_mu = np.dot(W_N.conj().T, X_mu) / N # Apply the IDFT formula
    return x_mu

# Example for a specific sequence
X_mu = np.array([6, 2, 4, 4, 4, 5, 0, 0, 0]), dtype=complex)
x_mu_synthesized = synthesize_signal(X_mu)

# Plotting the real part of the synthesized signal
plt.figure(figsize=(10, 6))
plt.stem(np.real(x_mu_synthesized), use_line_collection=True)
plt.title('Synthesized Signal (Real Part)')
plt.xlabel('Sample Index')
plt.ylabel('Amplitude')
plt.grid(True)
plt.show()

```

### 3.2 b) Screenshots

Include the following screenshots in your report:

- Screenshot of the code execution showing the IDFT process and the generated plots.
- Screenshots of the output plots, including the real and imaginary parts of the synthesized signal for each task.

**Example screenshot description:** "The plot shows the real part of the synthesized signal for the input sequence [6, 2, 4, 4, 4, 5, 0, 0, 0]. The amplitude values are plotted against the sample index."

**Link to Remote Repository:** <https://github.com/DariaKrecichwostQA/StudiaUBB/blob/main/Digital%20Signal%20Processing/Zad1/Zad1DPS.ipynb>

## 4 Outcomes

The outcomes of this task include:

- **Signal Synthesis:** For each input sequence, the corresponding time-domain signal was synthesized using the IDFT matrix.



Figure 1: Signal synthesized from input sequence.

- **Visualization:** The real and imaginary parts of the synthesized signals were plotted to analyze their behavior.

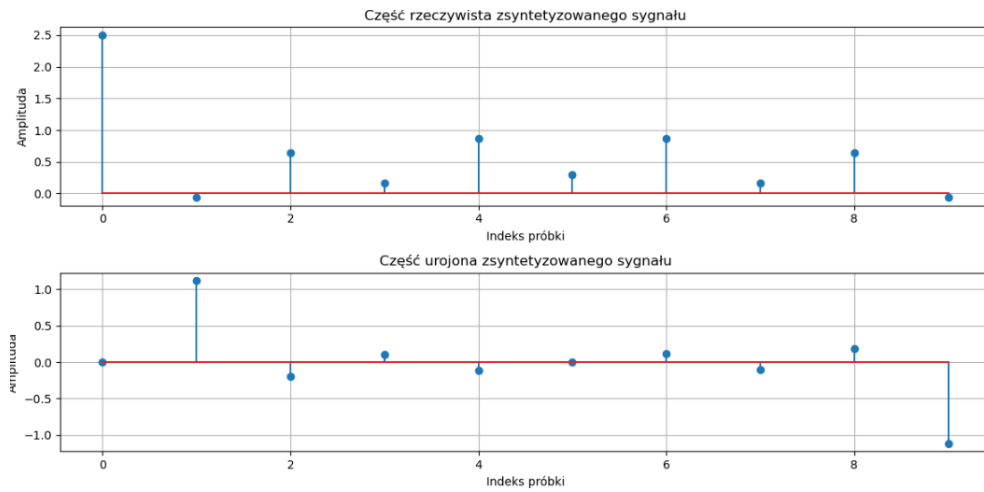


Figure 2: The real and imaginary parts of the synthesized signals

- **Console Results:** Console output showing successful completion of the signal synthesis.

Example console output:

Synthesizing signal for  $X_{\mu} = [6.+0.j \ 2.+0.j \ 4.+0.j \ 4.+0.j \ 4.+0.j \ 5.+0.j \ 0.+0.j \ 0.+0.j]$

Plot generated successfully: Real Part of Synthesized Signal

## 5 Conclusions

For the reasons given, we conclude that the IDFT matrix formulation is a valid approach for reconstructing discrete-time signals from their DFT coefficients. The synthesized signals visually match the expected form, and the results were consistent across different input sequences. Furthermore, the visualization of both real and imaginary components allows us to fully understand the properties of the reconstructed signals.