REPORT

Zajęcia: Digital Signal Processing

Teacher: prof. dr hab. Vasyl Martsenyuk

Lab 1

Date 28.09.2024

Topic: "Spectral Analysis of Deterministic Signals"

Variant 1

Imię Nazwisko: Mariusz Jagosz

Informatyka II stopień,

niestacjonarne,

1 semestr,

Gr.A

1. Problem statement: Synthesize a discrete-time signal by using the IDFT in matrix notation for different values of N. Show the matrices W and K. Plot the signal synthesized.

2. Input data:

xμ = [6, 2, 4, 3, 4, 5, 0, 0, 0, 0]T

3. Commands used (or GUI):

a) source code

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import inv
from numpy.fft import fft, ifft


N = 10
k = np.arange(N)
mu = np.arange(N)


K = np.outer(k, mu)
W = np.exp(+1j * 2*np.pi/N * K)
X_test = np.array([6, 2, 4, 3, 4, 5, 0, 0, 0, 0])
x_test = 1/N * np.matmul(W, X_test)

plt.stem(k, np.real(x_test), label='real',
         markerfmt='C0o', basefmt='C0:', linefmt='C0:')
plt.stem(k, np.imag(x_test), label='imag',
         markerfmt='C1o', basefmt='C1:', linefmt='C1:')

plt.plot(k, np.real(x_test), 'C0o-', lw=0.5)
plt.plot(k, np.imag(x_test), 'C1o-', lw=0.5)
plt.xlabel(r'sample $k$')
plt.ylabel(r'$x[k]$')
plt.legend()
plt.grid(True)

print(np.allclose(ifft(X_test), x_test))
print('DC is 1 as expected: ', np.mean(x_test))

True
DC is 1 as expected:  (0.6+8.881784197001253e-17j)
```

png

```
x_test2 = X_test[0] * W[:, 0] + X_test[1] * W[:, 1] + X_test[2] * W[:,
2] +X_test[3] * W[:, 3]  + X_test[4] * W[:, 4] +  X_test[5] * W[:, 5]
```

```
x_test2 *= 1/N
print(np.allclose(x_test, x_test2))
```

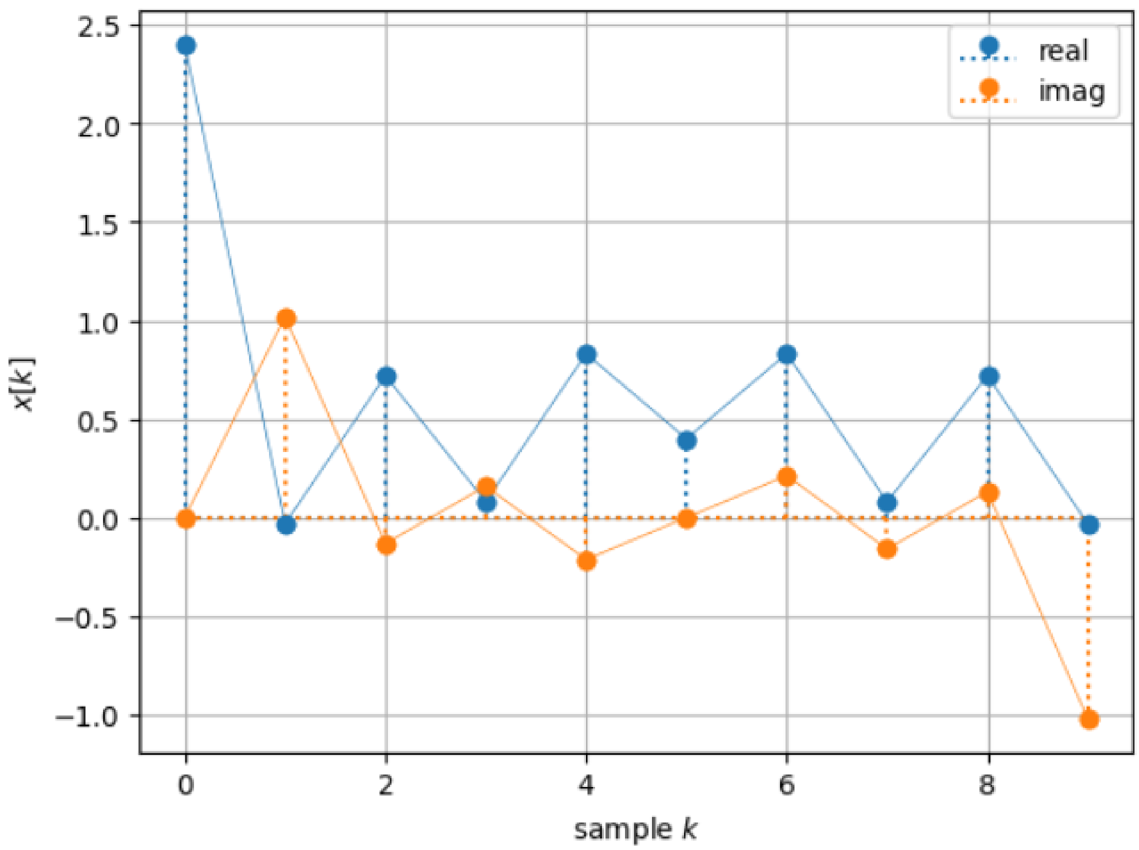
```
True
```

b) screenshots:

c) Link to the repo: [unibb/Digital signal processing/Task1 at main · mariuszjagosz/unibb (github.com)]()

4. Outcomes:

Plot:



Matrix K:

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 \\
0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 \\
0 & 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 \\
0 & 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 \\
0 & 6 & 12 & 18 & 24 & 30 & 36 & 42 & 48 & 54 \\
0 & 7 & 14 & 21 & 28 & 35 & 42 & 49 & 56 & 63 \\
0 & 8 & 16 & 24 & 32 & 40 & 48 & 56 & 64 & 72 \\
0 & 9 & 18 & 27 & 36 & 45 & 54 & 63 & 72 & 81
\end{pmatrix}
$$

Matrix W:

$$
\begin{pmatrix}
1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j & 1.000000+0.000000j \\
1.000000+0.000000j & 0.809017+0.587785j & 0.309017+0.951057j & -0.309017+0.951057j & -0.809017+0.587785j & -1.000000+0.000000j & -0.809017-0.587785j & -0.309017-0.951057j & 0.309017-0.951057j & 0.809017-0.587785j \\
1.000000+0.000000j & 0.309017+0.951057j & -0.809017+0.587785j & -0.809017-0.587785j & 0.309017-0.951057j & 1.000000-0.000000j & 0.309017+0.951057j & -0.809017+0.587785j & -0.809017-0.587785j & 0.309017-0.951057j \\
1.000000+0.000000j & -0.309017+0.951057j & -0.809017-0.587785j & 0.809017-0.587785j & 0.309017+0.951057j & -1.000000+0.000000j & 0.309017-0.951057j & 0.809017+0.587785j & -0.809017+0.587785j & -0.309017-0.951057j \\
1.000000+0.000000j & -0.809017+0.587785j & 0.309017-0.951057j & 0.309017+0.951057j & -0.809017-0.587785j & 1.000000-0.000000j & -0.809017+0.587785j & 0.309017-0.951057j & 0.309017+0.951057j & -0.809017-0.587785j \\
1.000000+0.000000j & -1.000000+0.000000j & 1.000000-0.000000j & -1.000000+0.000000j & 1.000000-0.000000j & -1.000000+0.000000j & 1.000000-0.000000j & -1.000000+0.000000j & 1.000000-0.000000j & -1.000000+0.000000j \\
1.000000+0.000000j & -0.809017-0.587785j & 0.309017+0.951057j & 0.309017-0.951057j & -0.809017+0.587785j & 1.000000-0.000000j & -0.809017-0.587785j & 0.309017+0.951057j & 0.309017-0.951057j & -0.809017+0.587785j \\
1.000000+0.000000j & -0.309017-0.951057j & -0.809017+0.587785j & 0.809017+0.587785j & 0.309017-0.951057j & -1.000000+0.000000j & 0.309017+0.951057j & 0.809017-0.587785j & -0.809017-0.587785j & -0.309017+0.951057j \\
1.000000+0.000000j & 0.309017-0.951057j & -0.809017-0.587785j & -0.809017+0.587785j & 0.309017+0.951057j & 1.000000-0.000000j & 0.309017-0.951057j & -0.809017-0.587785j & -0.809017+0.587785j & 0.309017+0.951057j \\
1.000000+0.000000j & 0.809017-0.587785j & 0.309017-0.951057j & -0.309017-0.951057j & -0.809017-0.587785j & -1.000000+0.000000j & -0.809017+0.587785j & -0.309017+0.951057j & 0.309017+0.951057j & 0.809017+0.587785j
\end{pmatrix}
$$

5. Conclusions:

In this report I've shown how I synthesized a discrete-time signal using the Inverse Discrete Fourier Transform (IDFT) in matrix notation for N = 10.

The W matrix was calculated according to the equation:

$$\mathbf{W} = \mathrm{e}^{+\mathrm{j}\frac{2\pi}{N}\odot\mathbf{K}}$$

While the K matrix was calculated based on this outer product:

$$\mathbf{K} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ N-1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 2 & \cdots & N-1 \end{bmatrix}$$

The synthesized signal was computed and plotted, showcasing both its real and imaginary components. The IDFT implementation was validated using the function *np.allclose* to see if the signal matched the *ifft* function. Overall, this exercise demonstrated the effectiveness of using matrix operations for IDFT and highlighted the importance of visualization in analyzing discrete-time signals.