jupyter dsp1i2 Last Checkpoint: 5 minutes ago

File Edit View Run Kernel Settings Help

Not Trusted

Markdown

JupyterLab Python 3 (ipykernel)

[69]:

```
import numpy as np
import matplotlib.pyplot as plt

# Define the signal
x_mu = np.array([6, 8, 2, 4, 3, 4, 5, 0, 0, 0], dtype=float)
N = len(x_mu)

# Create the index matrix K
k = np.arange(N)
mu = np.arange(N)
K = np.outer(k, mu)  # Compute K matrix

# Construct the Fourier matrix W for DFT
W = np.exp(-2j * np.pi * K / N)

# Compute DFT using the W matrix
X = np.dot(W, x_mu)

if N == 10:
    X_test = np.array([6, 8, 2, 4, 3, 4, 5, 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(np.fft.ifft(X_test), x_test))
    print('DC is 1 as expected: ', np.mean(x_test))

# Construct the inverse Fourier matrix W_inv for IDFT
W_inv = np.exp(2j * np.pi * K / N)

# Reconstruct the signal using IDFT
x_reconstructed = (1 / N) * np.dot(W_inv, X)
```

```
# Display the matrices K and W
print("Matrix K:\n", K)
print("\nMatrix W:\n", W.round(1))

# Display the reconstructed signal
print("\nReconstructed signal x (IDFT):\n", x_reconstructed)

# Plot the synthesized signal (real and imaginary parts)
plt.figure(figsize=(10, 6))
plt.stem(k, np.real(x_reconstructed), markerfmt='C0o', basefmt='C0:', linefmt='C0-', label='Reconstructed (Real part)')
plt.stem(k, np.imag(x_reconstructed), markerfmt='C1o', basefmt='C1:', linefmt='C1--', label='Reconstructed (Imag part)')
plt.title("Synthesized Signal using IDFT")
plt.xlabel("Sample index k")
plt.ylabel("Amplitude")
plt.legend()
plt.grid(True)
plt.show()
```

```
False
DC is 1 as expected:  (0.5999999999999999-5.551115123125783e-17j)
Matrix K:
 [[ 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]]

Matrix W:
 [[ 1. +0.j   1. +0.j   1. +0.j   1. +0.j   1. +0.j   1. +0.j   1. +0.j
   1. +0.j   1. +0.j   1. +0.j ]
 [ 1. +0.j   0.8-0.6j  0.3-1.j  -0.3-1.j  -0.8-0.6j -1. -0.j  -0.8+0.6j
  -0.3+1.j   0.3+1.j   0.8+0.6j]
 [ 1. +0.j   0.3-1.j  -0.8-0.6j -0.8+0.6j  0.3+1.j   1. +0.j   0.3-1.j
  -0.8-0.6j -0.8+0.6j  0.3+1.j ]
 [ 1. +0.j  -0.3-1.j  -0.8+0.6j  0.8+0.6j  0.3-1.j  -1. -0.j   0.3+1.j
   0.8-0.6j -0.8-0.6j -0.3+1.j ]
 [ 1. +0.j  -0.8-0.6j  0.3+1.j   0.3-1.j  -0.8+0.6j  1. +0.j  -0.8-0.6j
   0.3+1.j   0.3-1.j  -0.8+0.6j]
 [ 1. +0.j  -1. -0.j   1. +0.j  -1. -0.j   1. +0.j  -1. +0.j   1. +0.j
  -1. -0.j   1. +0.j  -1. +0.j ]
```