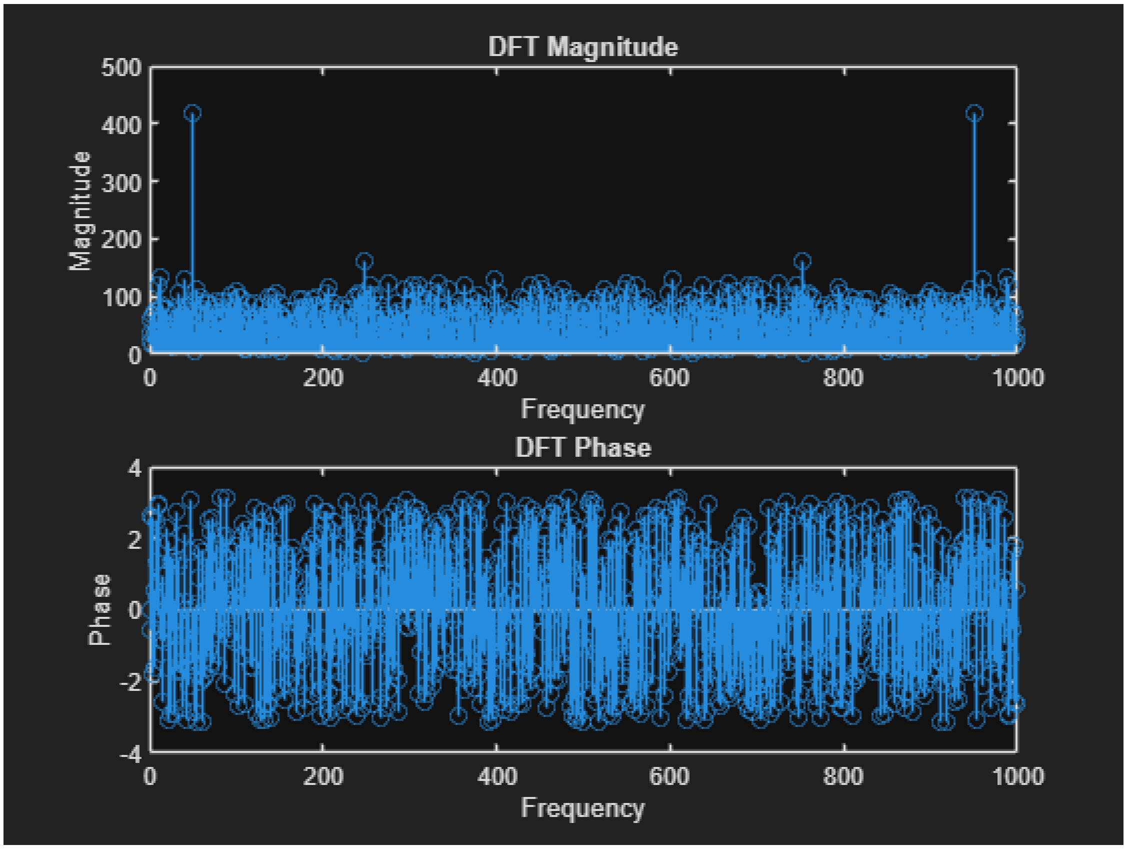
Table of Contents

Workspace Prep

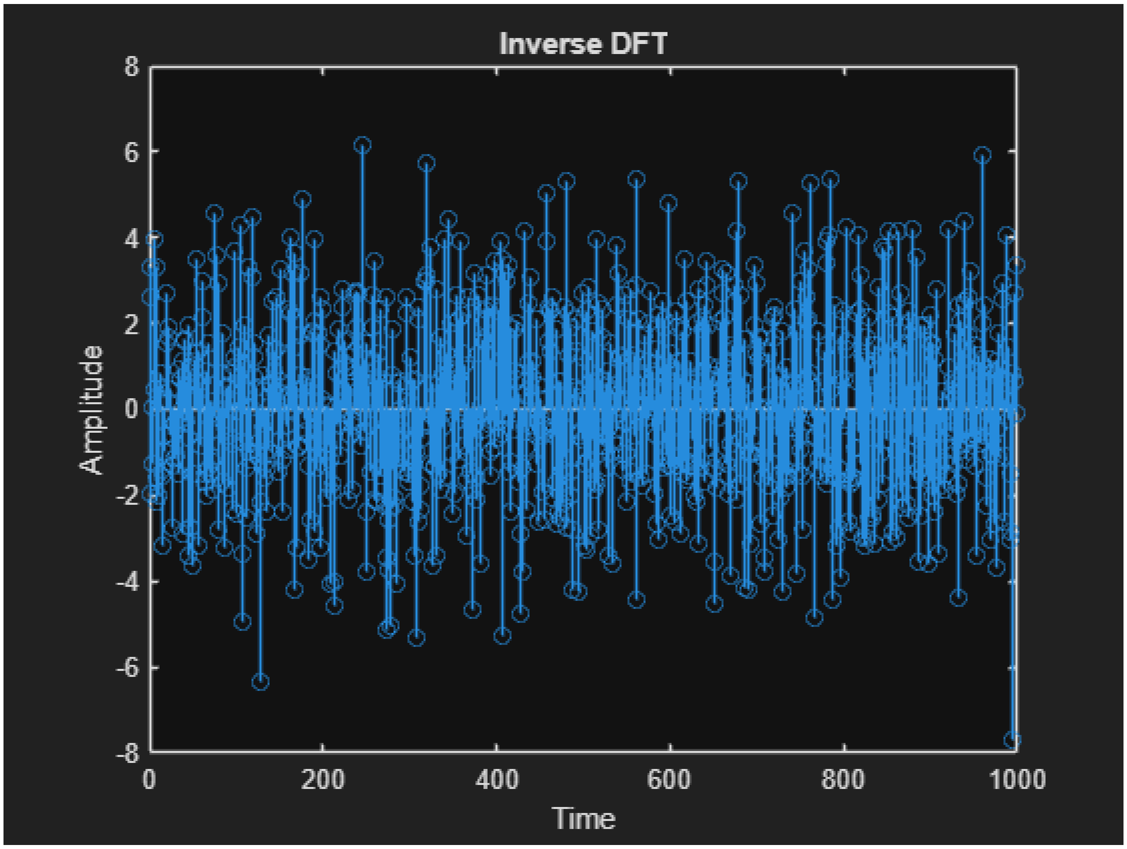
PART 1: DFT and IDFT Systems

DFT Function



IDFT Function

```
Warning: Using only the real component of complex data.
```



Published with MATLAB® R2023b

Table of Contents

```
%Roshan Jaiswal-Ferri
%Section - 03
%Aero 300 PreLab 9 - DFT and FFT and System Identification: 5/23/24
```

Workspace Prep

```
format long      %Allows for more accurate decimals
close all;       %Clears all
clear all;       %Clears Workspace
clc;             %Clears Command Window
```

PART 1: DFT and IDFT Systems

```
load('fourierData.mat');

%DFT on X
Xk = calcdft(X, L);

%Calculate size and phase of the DFT
szXk = abs(Xk);
phaseXk = angle(Xk);

%Frequency vector
k = 0:L-1;

%Plot magnitude and phase of the DFT
figure;
subplot(2,1,1);
stem(k, szXk);
title('DFT Magnitude');
xlabel('Frequency');
ylabel('Magnitude');

subplot(2,1,2);
stem(k, phaseXk);
title('DFT Phase');
xlabel('Frequency');
ylabel('Phase');

%IDFT on XDFT
xn = calcidft(Xk);
```

```
%Time vector
t = 0:L-1;

%Plot the IDFT result
figure;
stem(t, xn);
title('Inverse DFT');
xlabel('Time');
ylabel('Amplitude');
```

DFT Function

```
function [Xk] = calcdft(xn, N)
    L = length(xn);
    if N < L
        error('N must be greater than or equal to L!!');
    end
    x1 = [xn, zeros(1, N - L)];
    W = zeros(N, N);
    for k = 0:N-1
        for n = 0:N-1
            W(k+1, n+1) = exp(-1i * 2 * pi * n * k / N);
        end
    end
    Xk = W * x1.'; %keep x1 a col vector
end
```

IDFT Function

```
function [xn] = calcidft(Xk)
    N = length(Xk);
    IT = zeros(N, N);
    for k = 0:N-1
        for n = 0:N-1
            IT(k+1, n+1) = exp(1i * 2 * pi * n * k / N);
        end
    end
    xn = (IT * Xk) / N;
end
```

Published with MATLAB® R2023b