EXPERIMENT-6

Theme: Discrete Time Fourier Transform

Name: Anupama Kulshreshtha

Roll no.: EE22BTECH11009

Sample x(t) at appropriate sampling rate to get x[n] and compute DTFT of x[n] and plot |X(f)| vs f.

x(t) = Sin(2 π f1 t) + 2 Sin(2 π f2 t) + 1.5 Sin(2 π f3 t), where t= 0 to 1 sec, f1= 500 Hz, f2= 1KHz, f3 = 700 Hz.

Aim of the experiment: To compute DTFT of the given signal and plot it.

Theory of the DTFT:

The Discrete Time Fourier Transform of a signal x[n] is given by:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

Given signal,

$x(t) = \sin(2\pi f_1 t) + 2\sin(2\pi f_2 t) + 1.5\sin(2\pi f_3 t)$

Sampling Frequency selection:

According to Nyquist Theorem, sampling frequency f_S must be at least twice the highest frequency component in signal to avoid aliasing.

Hence, in this signal, $f_{max} = f_2 = 1$ kHz

- ⇨ Fs > 2 kHz

Here, we select fs = 7 kHz to get accurate results.

Implementation Steps:

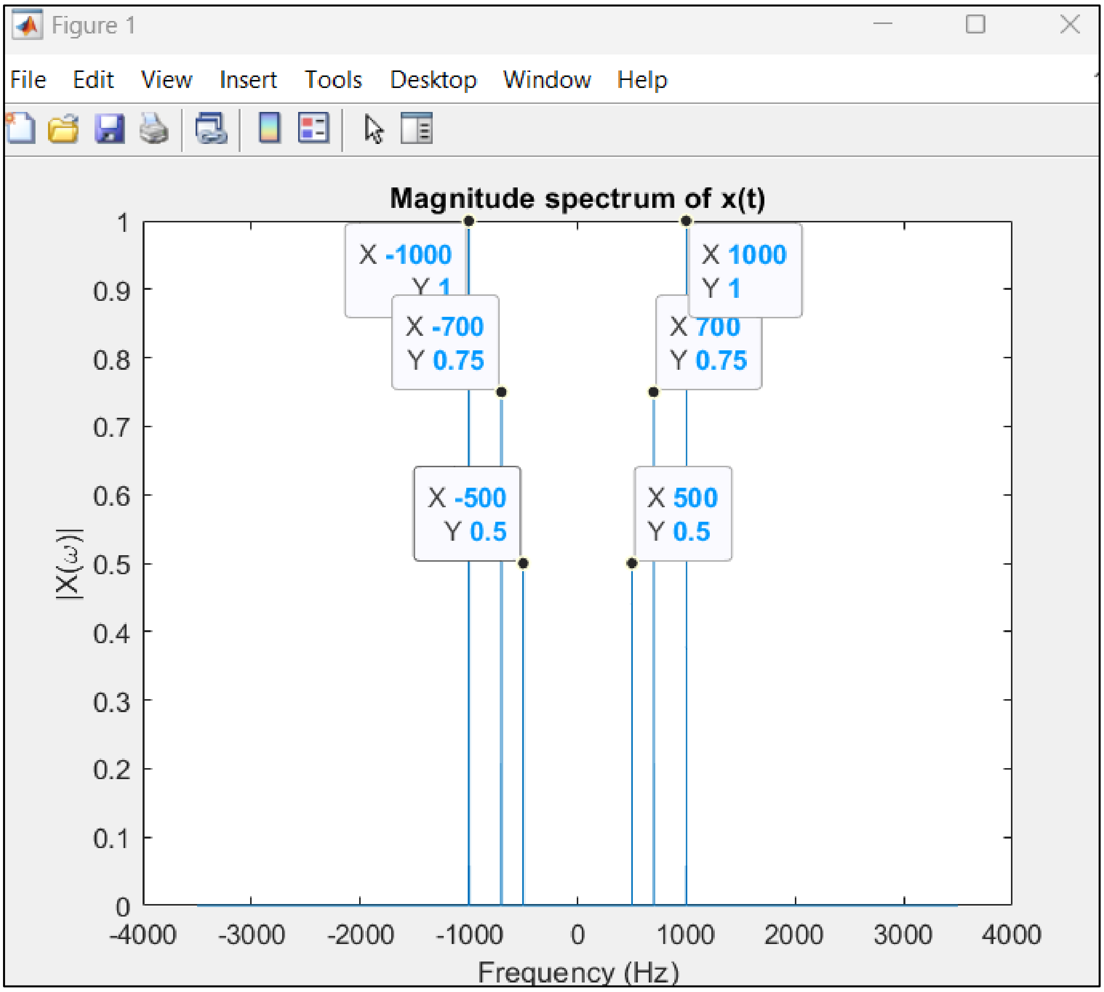
1. We code a function named dtft (x, omega), which takes the signal and its angular frequency as the input and returns the Discrete time fourier transform of the signal as the output.

2. The function runs a loop from 1 to length(omega) and sums the product of the signal and the exponential from 0 to N-1, where N represents the length of the signal.
3. In order to use this function to get the required DTFT of the given signal, we set the sampling frequency to 7 kHz, and define time t from 0 to 1, with intervals every (1/fs) second.
4. Omega to be given to the input of the coded function, is defined from -π to π, uniformly spaced taking the interval of length of t defined in the previous step, using the linspace function in MATLAB.
5. We then plot the output X, which is the DTFT of given signal x(t), vs the f in the x axis given by omega/(2π).
6. In order to get generalized results irrespective of sampling frequency, we scale the values in the axes by fs. We multiply f in x-axis with fs, and divide the absolute value of obtained X(f) by fs.

Code:

```
f1 = 500;
f2 = 1000;
f3 = 700;
fs = 7000;
t = 0:1/fs:1;

x = sin(2*pi*f1*t) + 2*sin(2*pi*f2*t) + 1.5*sin(2*pi*f3*t);
omega = linspace(-pi,pi,length(t));
f = omega/(2*pi);
X = dtft(x, omega);

plot(f*fs, abs(X)/fs);
xlabel('Frequency (Hz)');
ylabel('|X(\omega)|');
title('Magnitude spectrum of x(t)');

function X = dtft(x, omega)
    N = length(x);
    X = zeros(size(omega));

    for k = 1:length(omega)
        X(k) = sum(x.*exp(-1j*omega(k)*(0:N-1)));
    end
end
```

Output:



Observations and Conclusion:

1. The peaks in the magnitude plot correspond to the frequencies of the sinusoidal components present in the original continuous-time signal.
2. The magnitudes of these peaks represent the relative strengths of each frequency component in the output sampled signal.
3. Since the sampling frequency was chosen appropriately using the Nyquist theorem, there is no aliasing observed in the DTFT plot. All frequency components are accurately represented without any distortion caused by undersampling.
4. Therefore, this experiment enhanced our understanding of the DTFT, and helped to visualize the output using the generated plot.