Experiment-6

EE:2801 DSP-Lab

Indian Institute of Technology, Hyderabad

Jay Vikrant
EE22BTECH11025

I. Aim of the experiment

Sample $x(t)$ at an appropriate sampling rate to obtain $x(n)$ and compute the Discrete Time Fourier Transform (DTFT) of $x(n)$. Plot $|X(f)|$ versus f. Given signal:

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

where t ranges from 0 to 1 second, and the frequencies are $f_1 = 500$ Hz, $f_2 = 1000$ Hz, and $f_3 = 700$ Hz.

II. Introduction

The Discrete Time Fourier Transform (DTFT) is a mathematical tool used to analyze the frequency content of a discrete-time signal. In this simulation experiment, we aim to analyze the frequency spectrum of a given signal, $x(t)$, through the DTFT. The signal is composed of three sinusoidal components with different frequencies.

III. Theory of DTFT

The DTFT of a discrete-time signal $x[n]$ is given by the equation:

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

where ω is the angular frequency, and $X(e^{j\omega})$ represents the frequency response of the signal.

The DTFT provides a continuous frequency representation of the signal and is defined over the entire real axis. It is a periodic function with a period of 2π, and its values are complex numbers.

IV. MATLAB Simulation

The MATLAB code provided generates a signal $x(t)$ using three sinusoidal components with frequencies $f_1 = 500$ Hz, $f_2 = 1000$ Hz, and $f_3 = 700$ Hz. The signal is sampled with a sampling rate of 4000 Hz over the time interval $t = 0$ to 1 second.

The function `DTFT_func` calculates the DTFT of the signal using the given angular frequencies ω. The DTFT is computed using the formula:

$$X(e^{j\omega}) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot e^{-j\omega n}$$

where N is the length of the signal.

The resulting DTFT is then plotted in the frequency domain, representing the magnitude $|X(\omega)|$ against the frequency ω in Hz.

V. Code , Output and Plot of the simulations

```
function main()

fs = 4000;
f1 = 500;
f2 = 1000;
f3 = 700;

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 = -pi:(2*pi)/fs:pi;

X = DTFT_func(x, omega);

figure;
plot((omega*fs)/(2*pi), abs(X));
xlabel('frequency(Hz)')
ylabel('|X(w)|')
grid on;

end

function X = DTFT_func(x, omega)

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

The following got computed in Matlab,

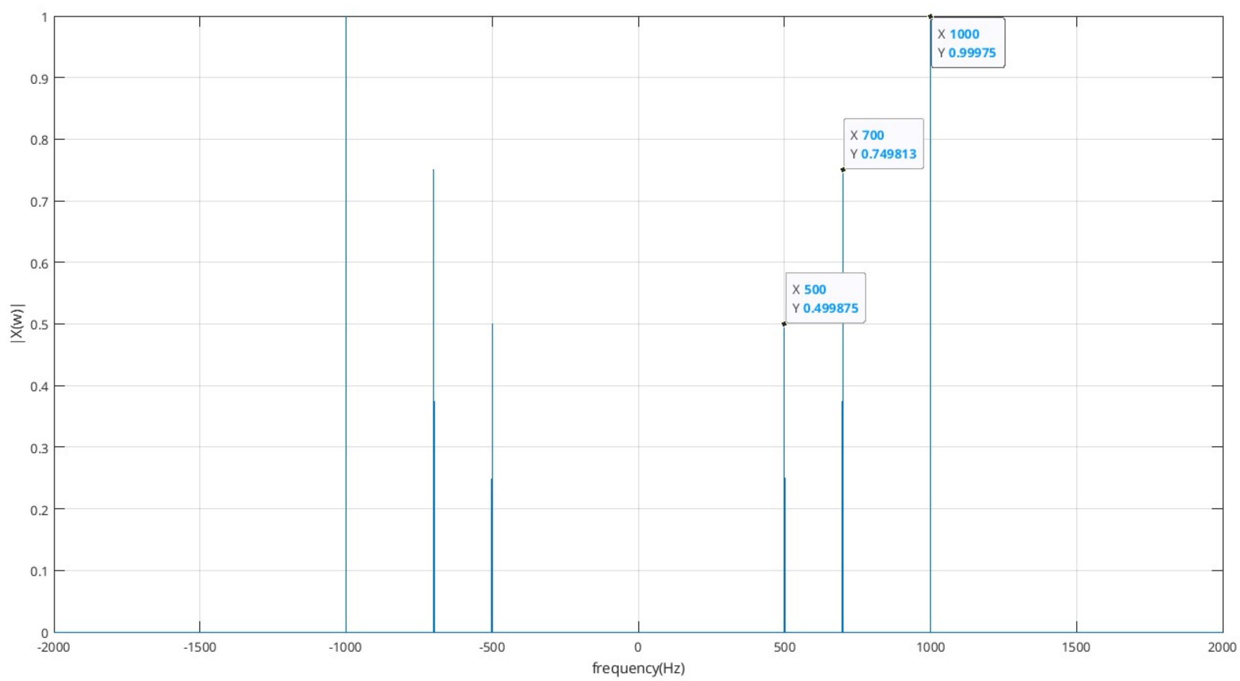

Fig. 0. $|X(w)|$ vs frequency plot

VI. Observations

The generated plot visually represents the frequency content of the signal $x(t)$. The peaks in the plot correspond to the frequencies of the sinusoidal components in the signal. The amplitude of each peak indicates the strength of the corresponding frequency component.

Observing the plot, we can identify peaks at frequencies $f_1 = 500$ Hz, $f_2 = 1000$ Hz, and $f_3 = 700$ Hz, confirming that the DTFT accurately captures the frequency components present in the original signal.

VII. Conclusion

In conclusion, the DTFT is a powerful tool for analyzing the frequency content of discrete-time signals. The MATLAB simulation successfully demonstrates the application of DTFT to a signal composed of multiple sinusoidal components. The resulting plot provides valuable insights into the frequency composition of the signal.

This experiment can be further extended to explore the effects of different sampling rates, signal durations, and signal types on the DTFT analysis.