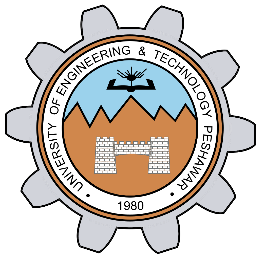
# **LAB REPORT # 07**



**CSE-402L Digital Signal Processing Lab**

# Submitted by: **Naveed ahmad**

Registration No.: **22PWCSE2165**

Class Section: **B**

“On my honor, as a student at the University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

# **Dr. Yasir Saleem Afridi**

January 7, 2025

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**COMPARING DOUBLE-SIDEBAND AND SINGLE-SIDEBAND AMPLITUDE-MODULATED SIGNALS USING MATLAB**

**Amplitude Modulation (AM)** is a technique where the amplitude of a carrier wave is varied in proportion to the message signal. In AM, two sidebands are generated: an upper sideband (USB) and a lower sideband (LSB).

1. Double-Sideband AM (DSB-AM): Both sidebands are transmitted.
2. Single-Sideband AM (SSB-AM): Only one of the sidebands (upper or lower) is transmitted, reducing bandwidth usage.

**TASKS**

1. Set the sampling frequency to 100 Hz and the carrier frequency to 10 Hz. Generate a time vector having a duration of 100 s.
2. Create a single-tone sinusoidal signal 1 Hz Signal.

x = sin(2\*pi\*t);

1. Carrier Signal

yc = sin(2\*pi\*fc\*t);

1. Plot the modulating Signal and Carrier Signal

fs = 100;

fc = 10;

t = (0:1/fs:100)';

x = sin(2 \* pi \* 1 \* t); % Modulating signal

yc = sin(2 \* pi \* fc \* t); % Carrier signal

figure;

plot(t, x, 'r', t, yc, 'b--');

xlabel('Time (s)');

ylabel('Amplitude');

legend('Original Signal', 'Carrier Signal');

title('Figure 1: Modulating and Carrier Signal');

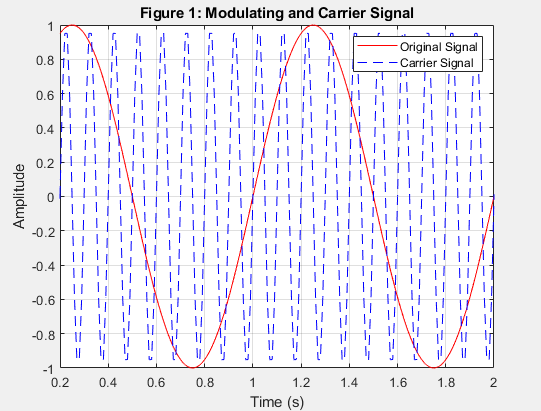
xlim([0.2, 2]);

ylim([-1, 1]);

xticks(0.2:0.2:2);

yticks(-1:0.2:1);

grid on;



The figure shows a modulating signal with a frequency of 1 Hz and a carrier signal of 10 Hz.

1. Modulate x using single- and double-sideband AM.

%Modulate signals

t = (0:1/fs:10)';

x = sin(2 \* pi \* 1 \* t); % Modulating signal

carrier\_cos = cos(2 \* pi \* fc \* t); % Cosine carrier

carrier\_sin = sin(2 \* pi \* fc \* t); % Sine carrier

% Double-Sideband Modulation (DSB)

ydouble = x .\* carrier\_cos;

% Hilbert Transform for SSB (Single Sideband Modulation)

x\_hilbert = imag(hilbert(x));

% Lower Sideband (LSB)

lowerSidebandSignal = x .\* carrier\_cos - x\_hilbert .\* carrier\_sin;

% Upper Sideband (USB)

upperSidebandSignal = x .\* carrier\_cos + x\_hilbert .\* carrier\_sin;

figure;

subplot(3,1,1);

plot(t, ydouble);

title('Double-Sideband AM');

xlabel('Time (s)');

ylabel('Amplitude');

subplot(3,1,2);

plot(t, lowerSidebandSignal);

title('Lower Sideband Signal (LSB)');

xlabel('Time (s)');

ylabel('Amplitude');

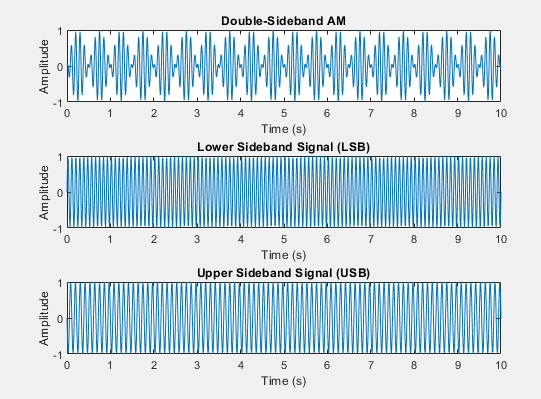
subplot(3,1,3);

plot(t, upperSidebandSignal);

title('Upper Sideband Signal (USB)');

xlabel('Time (s)');

ylabel('Amplitude');



* DSB modulation multiplies the baseband signal with the carrier signal.
* SSB modulation eliminates one of the sidebands (upper or lower) using the Hilbert transform.

1. Plot the DSB, USB, and LSB signals.

% Spectrum Analysis Using FFT

n = length(t); % Number of samples

f = (-n/2:n/2-1) \* (fs/n); % Frequency vector

% Compute FFT for each signal

fft\_double = fftshift(abs(fft(ydouble))/n);

fft\_lsb = fftshift(abs(fft(lowerSidebandSignal))/n);

fft\_usb = fftshift(abs(fft(upperSidebandSignal))/n);

% Plot Spectrum

figure;

subplot(3,1,1);

plot(f, fft\_double);

title('Spectrum of Double-Sideband Signal');

xlabel('Frequency (Hz)');

ylabel('Amplitude');

subplot(3,1,2);

plot(f, fft\_lsb);

title('Spectrum of Upper Sideband Signal');

xlabel('Frequency (Hz)');

ylabel('Amplitude');

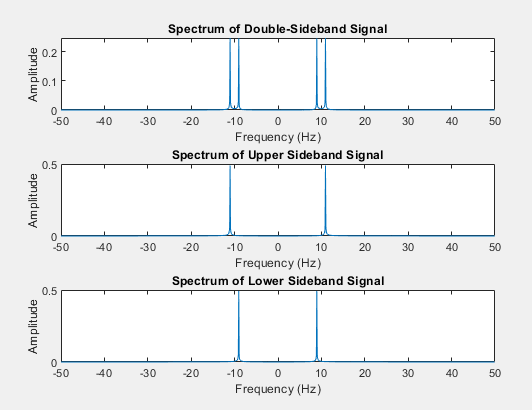
subplot(3,1,3);

plot(f, fft\_usb);

title('Spectrum of Lower Sideband Signal');

xlabel('Frequency (Hz)');

ylabel('Amplitude');



We’ve performed FFT on DSB, USB, and LSB signals and observed the spectrum of the signals.

Demodulate the lower and upper sideband signals

% Define a proper low-pass filter

lpFilt = designfilt('lowpassfir', 'PassbandFrequency', 2, ...

'StopbandFrequency', 5, 'PassbandRipple', 0.5, 'StopbandAttenuation', 60, ...

'SampleRate', fs);

% Demodulate Lower Sideband Signal (LSB)

demod\_lsb = lowerSidebandSignal .\* cos(2 \* pi \* fc \* t); % Multiply with carrier

s1 = filtfilt(lpFilt, demod\_lsb); % Apply low-pass filter

% Scale demodulated signal to match original amplitude

s1 = s1 / max(abs(s1)) \* max(abs(x));

% Demodulate Upper Sideband Signal (USB)

demod\_usb = upperSidebandSignal .\* cos(2 \* pi \* fc \* t); % Multiply with carrier

s2 = filtfilt(lpFilt, demod\_usb); % Apply low-pass filter

% Scale demodulated signal to match original amplitude

s2 = s2 / max(abs(s2)) \* max(abs(x));

The modulated signal is multiplied by the carrier and passed through a low-pass filter to get back the original message signal.

1. Compare processed signals with the original and verify reconstruction.

figure;

plot(t, x, 'k', t, s1, 'r:', t, s2, 'g-.'); % Plot original and demodulated signals

title('Comparison of Original and Demodulated Signals');

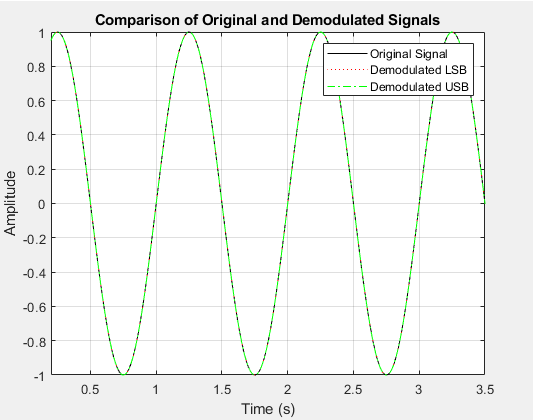
xlabel('Time (s)');

ylabel('Amplitude');

legend('Original Signal', 'Demodulated LSB', 'Demodulated USB');

xlim([0.2, 3.5]); % Adjust time range for better visualization

grid on;



Plotting the message and demodulated signals together shows that they closely match each other.

Answer the following Questions

1.plot the power density:

%Power Spectral Density using periodogram

[pxx, f\_psd] = periodogram(x, rectwin(length(x)), length(x), fs, 'power');

% Plot PSD

figure;

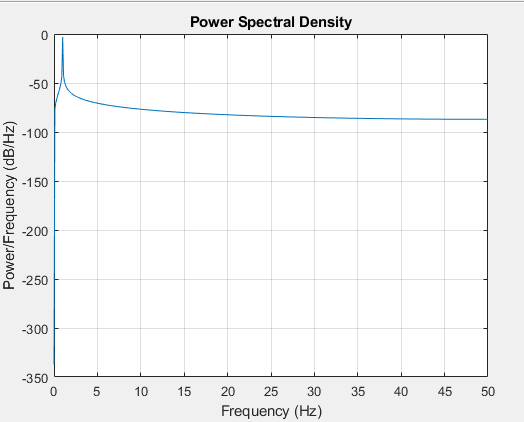
plot(f\_psd, 10\*log10(pxx)); % Convert power to dB scale

title('Power Spectral Density');

xlabel('Frequency (Hz)');

ylabel('Power/Frequency (dB/Hz)');

grid on;



The PSB represents the distribution of power across frequencies.

Plot the Spectrum. Choose Spectrum Unit as dBW, dBm, and Watts

%Plotting the Spectrum

n = length(x);

fft\_x = fft(x); % Perform FFT

f\_fft = (0:n-1) \* (fs/n); % Frequency axis

fft\_mag = abs(fft\_x) / n; % Normalize

% Plot

figure;

plot(f\_fft(1:n/2), 10\*log10(fft\_mag(1:n/2))); % One-sided spectrum

title('Signal Spectrum');

xlabel('Frequency (Hz)');

ylabel('Magnitude (dB)');

grid on;

% Spectrum in Watts

figure;

subplot(3,1,1);

plot(f\_fft(1:n/2), fft\_mag(1:n/2));

title('Spectrum in Watts');

xlabel('Frequency (Hz)');

ylabel('Amplitude (Watts)');

grid on;% Spectrum in dBm

spectrum\_dBm = 10 \* log10(fft\_mag(1:n/2).^2 \* 1000);

subplot(3,1,2);

plot(f\_fft(1:n/2), spectrum\_dBm);

title('Spectrum in dBm');

xlabel('Frequency (Hz)');

ylabel('Power (dBm)');

grid on;

% Spectrum in dBW

spectrum\_dBW = 10 \* log10(fft\_mag(1:n/2).^2);

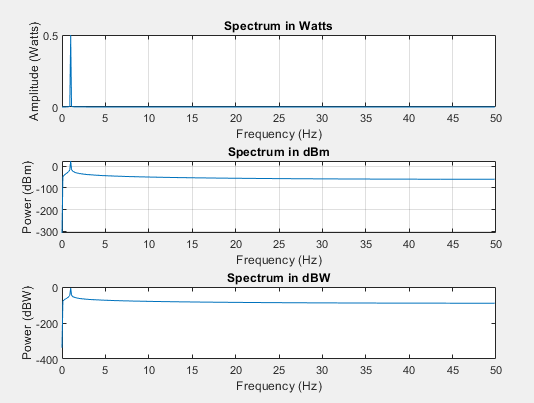
subplot(3,1,3);

plot(f\_fft(1:n/2), spectrum\_dBW);

title('Spectrum in dBW');

xlabel('Frequency (Hz)');

ylabel('Power (dBW)');



**CONCLUSION:**

In this lab, we explored modulation and demodulation techniques to understand the transmission and recovery of signals. Double-Sideband (DSB) and Single-Sideband (SSB) modulation methods were implemented, and their frequency spectra were analyzed. Demodulation techniques successfully reconstructed the original signal, verified by comparing it with the demodulated outputs.