## AM_FM SPECTRUM

```
close all
clear all
clc
Fs = 100; % Sampling frequency
t = (0:2*Fs+1)'/Fs;
Fc = 10; % Carrier frequency
x = sin(2*pi*2*t); % Message signal
Ac = 1;
% AM modulation (with carrier)
xam = (1 + x) .* cos(2*pi*Fc*t);
zam = abs(fft(xam));
zam = zam(1:length(zam)/2+1);
frqam = (0:length(zam)-1)*Fs/length(zam)/2;
% DSB-SC modulation (no carrier)
ydouble = x .* cos(2*pi*Fc*t);
zdouble = abs(fft(ydouble));
zdouble = zdouble(1:length(zdouble)/2+1);
frqdouble = (0:length(zdouble)-1)*Fs/length(zdouble)/2;
% SSB modulation (upper sideband using Hilbert Transform)
analytic_signal = hilbert(x);
ysingle = real(analytic_signal .* exp(1j*2*pi*Fc*t));
zsingle = abs(fft(ysingle));
zsingle = zsingle(1:length(zsingle)/2+1);
frqsingle = (0:length(zsingle)-1)*Fs/length(zsingle)/2;
% FM modulation
kf = 2*pi*10;
int_x = cumsum(x)/Fs;
xfm = cos(2*pi*Fc*t + kf*int_x);
zfm = abs(fft(xfm));
zfm = zfm(1:length(zfm)/2+1);
frqfm = (0:length(zfm)-1)*Fs/length(zfm)/2;
% Plot AM, DSB, SSB
figure;
subplot(3,1,1); plot(frqam, zam); title('Spectrum of AM signal');
subplot(3,1,2); plot(frqdouble, zdouble); title('Spectrum of DSB-SC
signal');
subplot(3,1,3); plot(frqsingle, zsingle); title('Spectrum of SSB
signal');
% Plot FM
figure;
plot(frqfm, zfm); title('Spectrum of FM signal');
```

---

## -SNR

```
clear all;
close all;
clc;
% Parameters
fm = 10;                    % Message frequency
fc = 250;                   % Carrier frequency
fs = 2*(fc + 2*fm)*10;      % Sampling frequency (oversampled)
t = 0:1/fs:(2/fm)-(1/fs);   % Time vector
```

```matlab
% Modulating signal
Vm = 1;
x = Vm * sin(2*pi*fm*t);
% Carrier signal
Vc = 1;
c = Vc * sin(2*pi*fc*t);
% AM Modulation (standard AM with carrier)
Sam = (1 + x) .* c;
% Normalize AM signal to unit power
Sam_norm = Sam / sqrt(mean(Sam.^2));
% Plot original signals (message, carrier, modulated)
figure;
subplot(3,1,1);
plot(t, x, 'b');
title('Message Signal');
xlabel('Time'); ylabel('Amplitude'); grid on;
subplot(3,1,2);
plot(t, c, 'r');
title('Carrier Signal');
xlabel('Time'); ylabel('Amplitude'); grid on;
subplot(3,1,3);
plot(t, Sam_norm, 'k');
title('AM Modulated Signal (Normalized, No Noise)');
xlabel('Time'); ylabel('Amplitude'); grid on;
% Loop over SNR values and plot noisy/modulated vs demodulated
snr_values = 0:10:50;
figure;
for i = 1:length(snr_values)
   SNR_dB = snr_values(i);
   % Add AWGN manually
   snr_linear = 10^(SNR_dB/10);
   noise_power = 1 / snr_linear;
   noise = sqrt(noise_power) * randn(size(Sam_norm));
   vrect = Sam_norm + noise;
   % Envelope detection (demodulation)
   y = abs(vrect);
   y = y - mean(y);    % Remove DC offset
   % Plot noisy modulated signal
   subplot(length(snr_values), 2, 2*i-1);
   plot(t, vrect);
   title(['AM + Noise (SNR = ' num2str(SNR_dB) ' dB)']);
   xlabel('Time'); ylabel('Amplitude');
   grid on;
   % Plot demodulated signal
   subplot(length(snr_values), 2, 2*i);
   plot(t, y);
   title(['Demodulated Signal (SNR = ' num2str(SNR_dB) ' dB)']);
   xlabel('Time'); ylabel('Amplitude');
   grid on;
end
```

# AWGN (Toolbox)

```
% Amplitude modulation and demodulation with AWGN and at the SNR=5,50db
% define modulating freq.
clear all;
close all;
clc;
fm = 10;
fc =250;
%sampling frequency fs>2*(fc+BW)
fs=2*(fc+2*fm)*10;
t=0:1/fs:(2/fm)-(1/fs);
%%modulating signal
Vm=1;
Wm=2*pi*fm*t;
x=sin(Wm);
x=Vm*x;
%% carrier signal
Vc=1;
c=Vc*sin(2*pi*fc*t);
%% Am Mdulation
Sam=ammod(x,fc,fs,0,Vm);
figure,
subplot(5,1,1);
plot(t,x);
title('Messge signal');
subplot(5,1,2);
plot(t,c);
title('carrier signal');
subplot(5,1,3);
plot(t,Sam);
title('Am modulaton ');
%% adding noise
noise_SNR=5;
vrect=awgn(Sam,noise_SNR);
subplot(5,1,4)
plot(t,vrect);
title(' Am modulation adding AWGN noise ');
%% Demodulation of AM signal
y=amdemod(vrect,fc,fs,0,Vc);
subplot(5,1,5);
plot(t,y);
title('Demodulated signal')
```

---

# Signals

```
t=-2*pi:0.1:2*2*pi
fm=0.1
s1=sin(2*pi*fm*t)
subplot(2,4,1)
plot(t,s1)
xlabel('time')
ylabel('amlitude')
title('sine wave')
hold on
```

```
s2=cos(2*pi*fm*t)
subplot(2,4,2)
plot(t,s2)
xlabel('time')
ylabel('amlitude')
title('cos wave')
hold on
unitstep=t>=0
subplot(2,4,3)
plot(t,unitstep)
xlabel('time')
ylabel('amlitude')
title('unit step')
hold on
ramp=t
subplot(2,4,4)
plot(t,ramp)
xlabel('time')
ylabel('amlitude')
title('ramp')
hold on
s3=square(t)
subplot(2,4,5)
plot(t,s3)
xlabel('time')
ylabel('amlitude')
title('square wave')
hold on
s4=sawtooth(t,0.5)
subplot(2,4,6)
plot(t,s4)
xlabel('time')
ylabel('amlitude')
title('sawtoth')
hold on
s5=sign(t)
subplot(2,4,7)
plot(t,s5)
xlabel('time')
ylabel('amlitude')
title('signum wave')
hold on
s6=exp(-t)
subplot(2,4,8)
plot(t,s6)
xlabel('time')
ylabel('amlitude')
title('exponentially decay')
t=0:0.01:2*2*pi
fm=5000
s1=sin(2*pi*fm*t)
s1dft=fft(s1)
%f1=fft(t,s1)
%subplot(2,4,1)
plot(abs(s1dft))
xlabel('frequency')
ylabel('amlitude')
title('frequeny spectrum of sine wave')
```

## AWGN FM

```
clear all;
close all;
clc;
% Parameters
fs = 100e3;                % Sampling frequency
t = 0:1/fs:0.1;            % Time vector (0.1 seconds)
fc = 20e3;                 % Carrier frequency
kf = 2*pi*5000;            % Frequency sensitivity
Am = 1;                    % Amplitude of message signal
% Message signal (sine wave)
fm = 200;                  % Message frequency
m = Am * sin(2*pi*fm*t); % Message signal
% Integrate message signal for FM modulation
int_m = cumsum(m) / fs;
% FM modulated signal
s_fm = cos(2*pi*fc*t + kf * int_m);
% AWGN channel
snr_dB = 100;               % SNR in dB
s_noisy = awgn(s_fm, snr_dB, 'measured');
% Demodulation using differentiation and envelope detection
% Differentiate the noisy signal
y_diff = [0 diff(unwrap(angle(hilbert(s_noisy))))] * fs / kf;
% Plotting
figure;
subplot(4,1,1);
plot(t, m, 'b');
title('Original Message Signal');
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
subplot(4,1,2);
plot(t, s_fm, 'k');
title('FM Modulated Signal');
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
subplot(4,1,3);
plot(t, s_noisy, 'r');
title(['FM Modulated + AWGN (SNR = ' num2str(snr_dB) ' dB)']);
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
subplot(4,1,4);
plot(t, y_diff, 'g');
title('Demodulated Signal from Noisy FM');
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
```

--------------------------------------------------------------------------------
-

## SNR FM

```
clear all;
close all;
clc;
% Parameters
fs = 100e3;                % Sampling frequency
t = 0:1/fs:0.1;            % Time vector (0.1 seconds)
fc = 20e3;                 % Carrier frequency
kf = 2*pi*5000;            % Frequency sensitivity
Am = 1;                    % Amplitude of message signal
% Message signal (sine wave)
fm = 200;                  % Message frequency
```

```matlab
m = Am * sin(2*pi*fm*t); % Message signal
% Integrate message signal for FM modulation
int_m = cumsum(m) / fs;
% FM modulated signal
s_fm = cos(2*pi*fc*t + kf * int_m);
% AWGN channel (manual implementation)
snr_dB = 20;              % SNR in dB
signal_power = mean(s_fm.^2);       % Signal power
noise_power = signal_power / (10^(snr_dB / 10)); % Noise power
noise = sqrt(noise_power) * randn(size(s_fm));  % Gaussian noise
s_noisy = s_fm + noise;  % Noisy signal
% Demodulation using phase extraction and filtering
% Extract the phase information
analytic_signal = hilbert(s_noisy);  % Compute the analytic signal
phase = angle(analytic_signal);  % Extract phase
% Differentiate the phase to recover the frequency deviation
y_diff = [0 diff(unwrap(phase))] * fs / (2 * pi * kf); % Differentiate and scale
% Low-pass filter to recover the original message signal
[b, a] = butter(4, fm / (fs / 2)); % 4th-order Butterworth LPF with cutoff near message frequency
y_filtered = filtfilt(b, a, y_diff); % Apply zero-phase filtering
% Plotting
figure;
subplot(4,1,1);
plot(t, m, 'b');
title('Original Message Signal');
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
subplot(4,1,2);
plot(t, s_fm, 'k');
title('FM Modulated Signal');
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
subplot(4,1,3);
plot(t, s_noisy, 'r');
title(['FM Modulated + AWGN (SNR = ' num2str(snr_dB) ' dB)']);
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
subplot(4,1,4);
plot(t, y_filtered, 'g');
title('Demodulated Signal from Noisy FM');
xlabel('Time (s)'); ylabel('Amplitude'); grid on;
```