



浙江工业大学

MATLAB & Communication Simulations

Lab Report3

Name 凌智城

Teacher 张昱

Class 通信工程 1803 班

Student ID 201806061211

Department 信息工程学院

Date: Apr. 18th, 2021

I. Task_1

A. Code

lab3_1.m

```
%% Generate random process.
```

```
clear;
```

```
clc;
```

```
%% Task(1)
```

```
% ①Generate a random process  $y[n]=x1[n]+\sin(x2[n])$ .
```

```
N = 1000; % The maximum value of n,sequence length
```

```
M = 50; % Select a sequence of length 101 =  $M*2+1$ 
```

```
Y = zeros(1,N);
```

```
Ryav = zeros(1,2*M+1); % 950:1050
```

```
Syav = zeros(1,2*M+1); % 950:1050
```

```
% Take the ensemble average over one hundred realizations.
```

```
for i = 1:100
```

```
    X1 = randn(1,1000); % Use 'randn(x,y)' to generate x*y matrix
```

```
    X2 = randn(1,1000);
```

```
    for n = 1:N
```

```
        Y(n) = X1(n)+sin(X2(n));
```

```
    end
```

```
    Ry = xcorr(Y,'unbiased'); % Autocorrelation of {Yn}
```

```
    Ry = Ry(950:1050); % Original Ry, m is -999 to 999( $2*N-1$ )
```

```
    Sy = fftshift(abs(fft(Ry))); % Power spectrum of {Yn}
```

```
    Ryav=Ryav+Ry; % 100 times
```

```
    Syav=Syav+Sy;
```

```
end
```

```
Ryav=Ryav/100; % Time-average
```

```
Syav=Syav/100;
```

```
%% Task(2)
```

```
% ① the mean and variance of  $y[n]$ .
```

```
% Use 'mean(Y)' and 'var(Y)' to calculate the mean and variance  
of  $y[n]$ .
```

```
Myav = mean(Y)
```

```
Vyav = var(Y)
```

```

%% Task(3)
% ① the spectrum and auto-correlation of y[n].
figure(1)
subplot(121)
plot(-M:M,Ryav);
title('Auto-correction of y[n]')
xlabel('Time')
subplot(122)
plot(-0.5:0.01:0.5,Syav);
title('Spectrum of y[n]')
xlabel('Time')

```

B. Figure

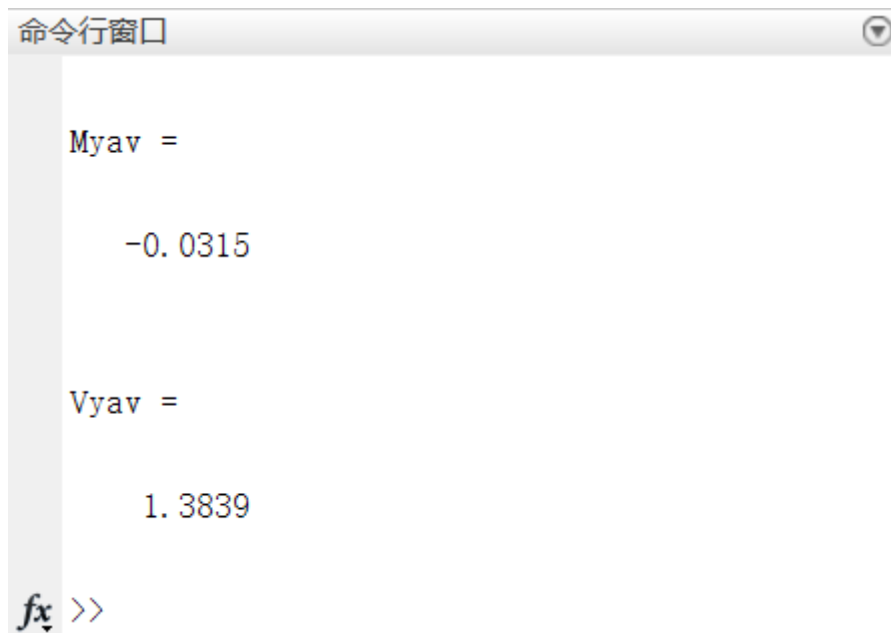


Fig.1 The mean and variance of $y[n]$.

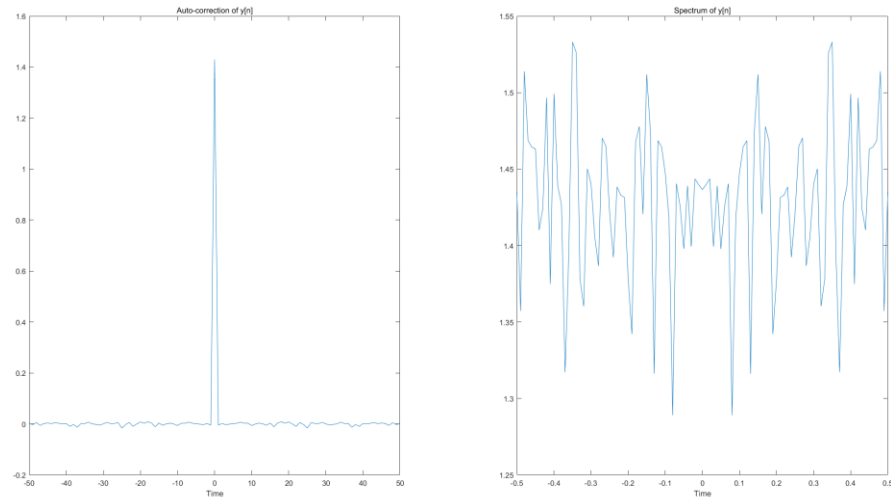


Fig.2 The spectrum and auto-correlation of $y[n]$.

C. Discussion

In this task, I use 'randn(x,y)' to generate 'x*y' matrix. In order to calculate the autocorrelation and power spectrum of $y[n]$ in the time-average, I take the ensemble average over one hundred realizations, and then take their average. The length of $x1[n]$, $x2[n]$ and $y[n]$ are all N in length.

As we all know, the length of 'xcorr(y)' is ' $2*N-1$ '. For original R_y , m is -999 to 999, so I selected (950:1050) of the intervals. After a hundred stacking, use ' $R_{yav}=R_{yav}/100$ ' and ' $S_{yav}=S_{yav}/100$ ' to calculate the time-average. I also use 'mean(Y)' and 'var(Y)' to calculate the mean and variance of $y[n]$.

To judge rodman progress is ergodic or stationary, I compare the time average and statistical average and find that the expectation of $y[n]$ is a constant while its autocorrelation only depends on the time difference. Also, the expectation equals the time-average. So, we can say the random process will go through all the possible states. The conclusion is that the rodman progress $y[n]$ is ergodic and stationary.

II. Task_2

A. Code

lab3_2.m

```
%% Amplitude modulation. Sampling rate Fs=1500Hz.

clear;
clc;

%% Task(1)

% Initialization.
t0 = 1;           % Signal duration
Fs = 1500;        % Sampling rate
Ts = 1/Fs;        % Sampling interval
Fc = 20;          % Carrier rate
t = 0:Ts:t0;      % Time vector
N = length(t);    % Length of t
N0 = 0.0005;      % Power spectral density of noise

% Message signal and modulated signal(DSB_AM and USSB_AM).
m = sin(2*pi*t);  % Message signal
c = cos(2*pi*Fc*t); % Carrier signal
Ac_DSB_AM = 1;
Ac_USSB_AM = sqrt(2);

% DSB_AM modulated signal.
u_dsb = Ac_DSB_AM*m.*c;
% SSB_AM modulated signal.
u_ussb = 1/2*Ac_USSB_AM*real(hilbert(m).*exp(1i*2*pi*Fc*t));
% "equal to " u_ussb = 1/2*Ac_USSB_AM*cos(2*pi*(Fc+1)*t).

% Fourier transform(Time domain to frequency domain).
[~,M] = T2F(t,m);
[~,U_DSB] = T2F(t,u_dsb);
[df1,U_USSB] = T2F(t,u_ussb);

% ① Calculate the power of the modulated signal and message
signal.
m_power = sum(abs(m).^2)/N
dsb_power = sum(abs(u_dsb).^2)/N
```

```

ussb_power = sum(abs(u_ussb).^2)/N

% ② Plot the modulated signal and the spectrum.
figure(1)
subplot(2,2,1)
plot(t,u_dsb)
title('DSB Signal')
xlabel('Time')
subplot(2,2,2)
plot(t,u_ussb)
title('USSB Signal')
xlabel('Time')
subplot(2,2,3)
plot(df1,abs(U_DSB))
title('Spectrum of the Modulated DSB Signal')
xlabel('Frequency')
subplot(2,2,4)
plot(df1,abs(U_USSB))
title('Spectrum of the Modulated USSB Signal')
xlabel('Frequency')

%% Task(2)

% Demodulate the above signals.
% Coherent carrier mixing.
y_dsb = u_dsb.*c;
y_ussb = u_ussb.*c;

% Fourier transform(Time domain to frequency domain).
[~,Y_DSB]=T2F(t,y_dsb);
[df1,Y_USSB]=T2F(t,y_ussb);

% Initialization of lowpass filter.
H_DSB = zeros(1,length(df1));
H_USSB = zeros(1,length(df1));
for i=1:length(df1)
    if abs(df1(i))<2
        H_DSB(i)=2*Ac_DSB_AM;
        H_USSB(i)=2*Ac_USSB_AM;
    end
end

% Lowpass filter filtering(Spectrum of the filter output).
DEM_DSB=H_DSB.*Y_DSB;

```

```

DEM_USSB=H_USSB.*Y_USSB;

% Fourier transform(Frequency domain to time domain).
[~,dem_w_dsb]=F2T(df1,DEM_DSB);
[dt1,dem_w_ussb]=F2T(df1,DEM_USSB);
dem_w_dsb=dem_w_dsb(1:length(t));
dem_w_ussb=dem_w_ussb(1:length(t));

% ① The pass band of the low-pass filter for DSB_AM and USSB_AM.
passband_dsb = 2*1
passband_ussb = 2*1
% ② Plot the demodulated signal and the spectrum.
figure(2)
subplot(2,2,1)
plot(dt1,real(dem_w_dsb))
title('Demodulated DSB signal')
xlabel('Time')
subplot(2,2,2)
plot(dt1,real(dem_w_ussb))
title('Demodulated USSB signal')
xlabel('Time')
subplot(2,2,3)
plot(df1,(abs(DEM_DSB)))
title('Spectrum of the Demodulated DSB signal')
xlabel('Frequency')
subplot(2,2,4)
plot(df1,(abs(DEM_USSB)))
title('Spectrum of the Demodulated USSB signal')
xlabel('Frequency')

%% Task(3)

% Add the thermal noise.
noise = randn(1,length(t))*sqrt(N0/2*Fs);
u_n_dsb = u_dsb + noise;
u_n_ussb = u_ussb + noise;

% Coherent carrier mixing.
y_n_dsb = u_n_dsb.*c;
y_n_ussb = u_n_ussb.*c;

% Fourier transform(Time domain to frequency domain).
[~,Y_N_DSB]=T2F(t,y_n_dsb);
[df1,Y_N_USSB]=T2F(t,y_n_ussb);

```

```

% Lowpass filter filtering(Spectrum of the filter output).
% Filters have been initialized already.
DEM_N_DSB=H_DSB.*Y_N_DSB;
DEM_N_USSB=H_USSB.*Y_N_USSB;

% Fourier transform(Frequency domain to time domain).
[~,dem_n_dsb]=F2T(df1,DEM_N_DSB);
[dt1,dem_n_ussb]=F2T(df1,DEM_N_USSB);

% ①Plot the modulated signal and demodulated signal(time domain).
figure(3)
subplot(2,2,1)
plot(t,u_n_dsb)
title('Modulated DSB signal with noise')
xlabel('Time')
subplot(2,2,2)
plot(t,u_n_ussb)
title('Modulated USSB signal with noise')
xlabel('Time')
subplot(2,2,3)
plot(dt1,real(dem_n_dsb))
title('Demodulated DSB signal with noise')
xlabel('Time')
subplot(2,2,4)
plot(dt1,real(dem_n_ussb))
title('Demodulated USSB signal with noise')
xlabel('Time')

%% Task(4)

% What is the disadvantage
% if we apply a low-pass filter with a band much wider than
necessary?

% Initialization of lowpass filter much wider than necessary.
H_W_DSB = zeros(1,length(df1));
H_W_USSB = zeros(1,length(df1));
for i=1:length(df1)
    if abs(df1(i))<200      % 200 >> 2
        H_W_DSB(i)=2*Ac_DSB_AM;
        H_W_USSB(i)=2*Ac_USSB_AM;
    end
end
end

```



```

% Lowpass filter filtering(Spectrum of the filter output).
DEM_W_DSB=H_W_DSB.*Y_DSB;
DEM_W_USSB=H_W_USSB.*Y_USSB;

% Fourier transform(Frequency domain to time domain).
[~,dem_w_dsb]=F2T(df1,DEM_W_DSB);
[dt1,dem_w_ussb]=F2T(df1,DEM_W_USSB);
dem_w_dsb=dem_w_dsb(1:length(t));
dem_w_ussb=dem_w_ussb(1:length(t));

% The demodulated signal and the spectrum.
figure(4)
subplot(2,2,1)
plot(dt1,real(dem_w_dsb))
title('Demodulated DSB signal with a much wider band')
xlabel('Time')
subplot(2,2,2)
plot(dt1,real(dem_w_ussb))
title('Demodulated USSB signal with a much wider band')
xlabel('Time')
subplot(2,2,3)
plot(df1,(abs(DEM_DSB)))
title('Spectrum of the Demodulated DSB signal with a much wider band')
xlabel('Frequency')
subplot(2,2,4)
plot(df1,(abs(DEM_USSB)))
title('Spectrum of the Demodulated USSB signal with a much wider band')
xlabel('Frequency')

```

B. Figure

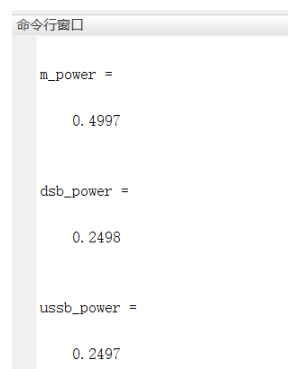


Fig.3 The power of the modulated signal and message signal.

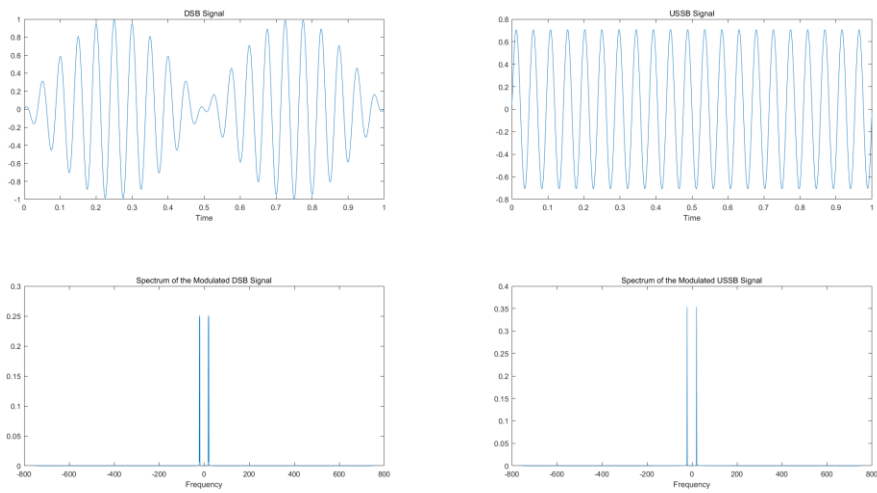


Fig.4 The modulated signal and the spectrum.

```
passband_dsb =  
    2  
  
passband_usb =  
    2  
  
fx >>
```

Fig.5 The pass band of the low-pass filter for DSB_AM and USB_AM.

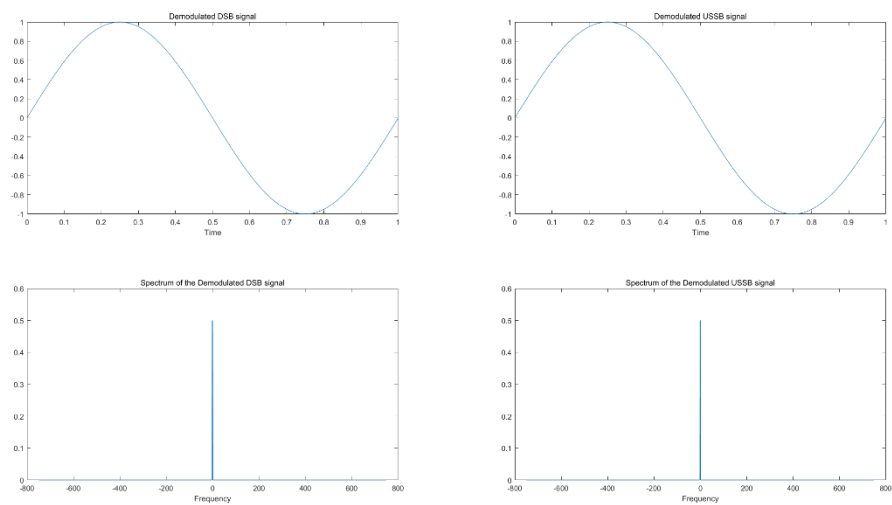


Fig.6 The demodulated signal and the spectrum.

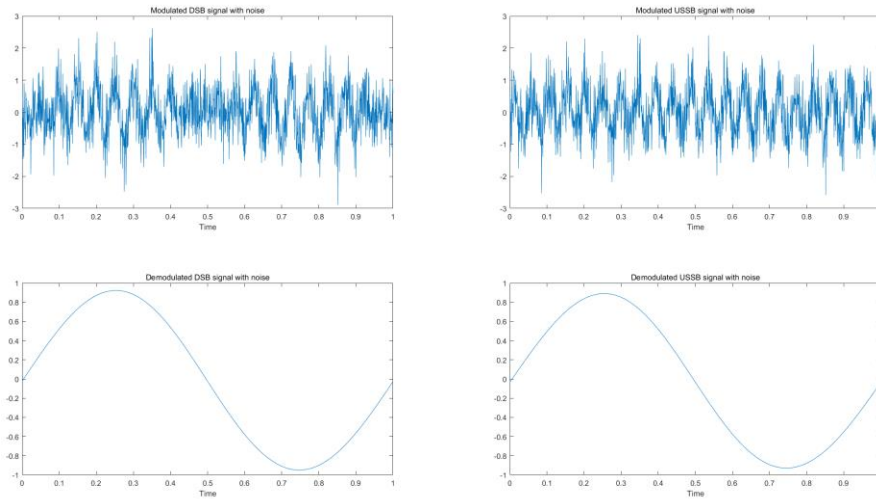


Fig.7 The modulated signal and demodulated signal with noise.

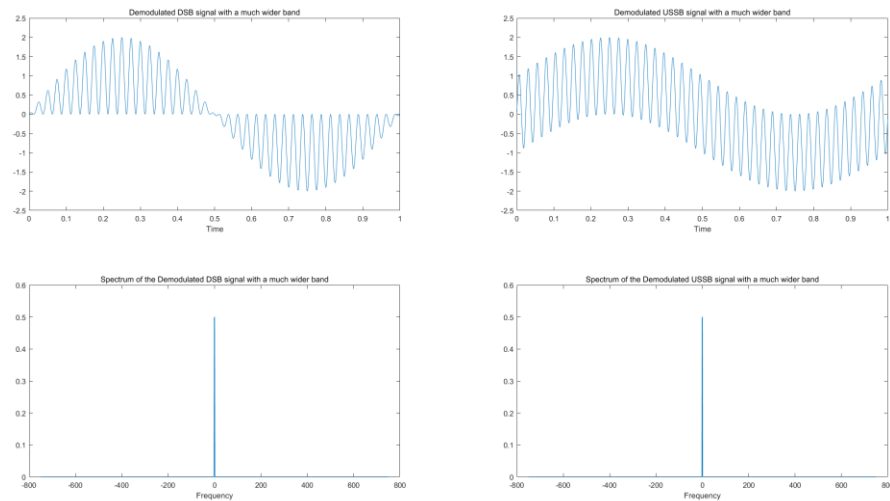


Fig.8 The demodulated signal and the spectrum(much wider band).

C. Discussion

In this task, signal duration is $t_0=1$, message signal $m=\sin(2\pi t)$, carrier signal $c=\cos(2\pi f_c t)$, there are two modulation methods DSB-AM and SSB-AM. In the SSB-AM, we use the upper sideband modulation, USSB-AM.

The modulated signals:

DSB-AM signal :

$$s(t) = A_c m(t) \cos(2\pi f_c t) \quad (1.1)$$

USSB-AM signal:

$$u(t) = \frac{A_c}{2} m(t) \cos(2\pi f_c t) - \frac{A_c}{2} \hat{m}(t) \sin(2\pi f_c t) \quad (1.2)$$

One thing to note is that the A_c of DSB-AM and USSB-AM are different. The former is 1 and the latter is $\sqrt{2}$. Use 'sum(abs(m).^2)/N' to calculate their power (see fig.3). The modulated signal and the spectrum (see fig.4)

Then demodulating the above signals, I use coherent carrier to demodulate them and then pass them through a low-pass filter with different gains.

The demodulated signals:

DSB-AM signal:

$$y(t) = A_c m(t) \cos(2\pi f_c t) \cos(2\pi f_c t) = \frac{A_c}{2} m(t) + \frac{A_c}{2} m(t) \cos(4\pi f_c t) \quad (1.3)$$

USSB-AM signal:

$$\begin{aligned} y(t) &= \frac{A_c}{2} m(t) \cos^2(2\pi f_c t) - \frac{A_c}{2} \hat{m}(t) \sin(2\pi f_c t) \cos(2\pi f_c t) \\ &= \frac{A_c}{4} m(t) + \frac{A_c}{4} \cos(4\pi f_c t) - \frac{A_c}{4} \hat{m}(t) \sin(4\pi f_c t) \end{aligned} \quad (1.4)$$

In order to recover the original signal $m(t)$, one thing is to use a low-pass filter to filter out unwanted components, and the other thing is to choose right gain. To divide the A_c , for the DSB-AM, what needs to be kept is $\frac{A_c}{2} m(t)$, and then the gain is obviously

$\frac{2}{A_c} = 2$; For the USSB-AM, what needs to be kept is $\frac{A_c}{4} m(t)$, and then the

gain is obviously $\frac{4}{A_c} = 2\sqrt{2}$. As the band of original signal $m(t)$ is $f=1$, so

I set the passband of the low-pass filter to 2Hz (see fig.5). The cut-off frequency is 2HZ, which is twice the maximum frequency of the original signal, then the original signal is recovered well in both modulation

methods (see fig.6).

In fact, for the noise, all of the frequency in array of (f_m, f_c) are ok, although the amplitude of the demodulated signal will change due to the constant gain after mixing noise, the original signal is restored almost without distortion (see fig.7).

If we apply a low-pass filter with a band much wider than necessary, when filtering, it will mix in the high frequency components with the frequency spectrum nearby. The high frequency component is just the continuation of the baseband component at high frequency, if it is not too wide, may we can barely recover the original signal. If the cut-off frequency is set too wide, it may cause signal aliasing and amplitude changes.