



به نام خدا  
دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده مهندسی برق و کامپیوتر



# اصول سیستم‌های مخابراتی

پاییز 1400

استاد: دکتر مریم صباغیان

تمرین کامپیوتری شماره 2

محمد مهدی عبدالحسینی

810 198 434



Communication Systems

## فهرست مطالب

1	بخش اول: مدولاسیون دامنه از نوع متعارف .....
5	بخش دوم: مدولاسیون های DSB و SSB .....
17	بخش سوم: آشنایی با تبدیل هیلبرت و آشکار ساز پوش .....

## بخش اول : مدولاسیون دامنه از نوع متعارف

در این قسمت میخواهیم تابعی بنویسیم که سیگنال پیام  $x_m(t)$  ، دامنه موج حامل  $A_c$  ، اندیس مدولاسیون  $\mu$  و فرکانس موج حامل  $f_c$  را بگیرد و سیگنال مدوله شده را برگرداند. البته برای سادگی کار  $t$  را هم به عنوان ورودی تابع در نظر گرفته‌ام.

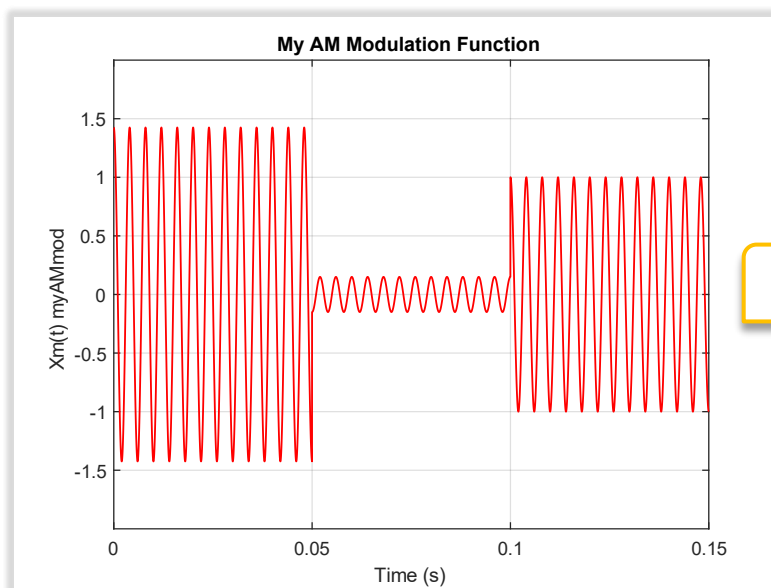
```
function Xc = my_AMmod(x, Ac, mu, fc, t)
    n = max(abs(x));
    Xc = Ac*(1 + mu*x/n).*cos(2*pi*fc*t);
end
```

سیگنال پیام بصورت زیر تعریف شده است:

$$x_m(t) = \begin{cases} 1 & 0 \leq t < \frac{t_0}{3} \\ -2 & \frac{t_0}{3} \leq t < \frac{2t_0}{3} \\ 0 & o.w. \end{cases}$$

```
fs_part1 = 100000; % 100kHz
t0_part1 = 0.15;
ts_part1 = 1/fs_part1;
t = 0:ts_part1:t0_part1 - ts_part1;
Xmt = zeros([1,t0_part1*fs_part1]);
Xmt(1, 1:(t0_part1*fs_part1/3 - 1)) = 1;
Xmt(1, (t0_part1*fs_part1/3):(t0_part1*fs_part1*2/3 - 1)) = -2;
```

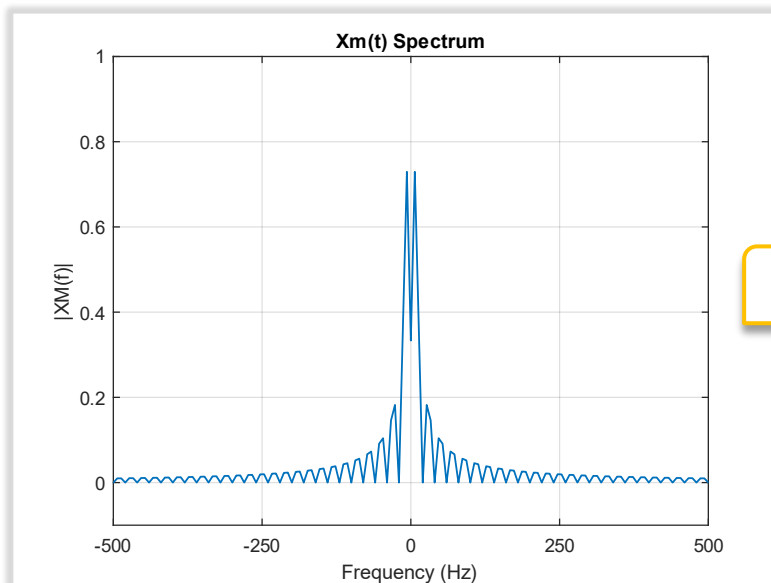
در ادامه قصد داریم سیگنال پیام  $x_m(t)$  را با فرکانس  $f_c = 250Hz$  ،  $\mu = 0.85$  و  $t_0 = 0.15$  مدوله کنیم و سیگنال مدوله شده را رسم کنیم. (شکل 1)



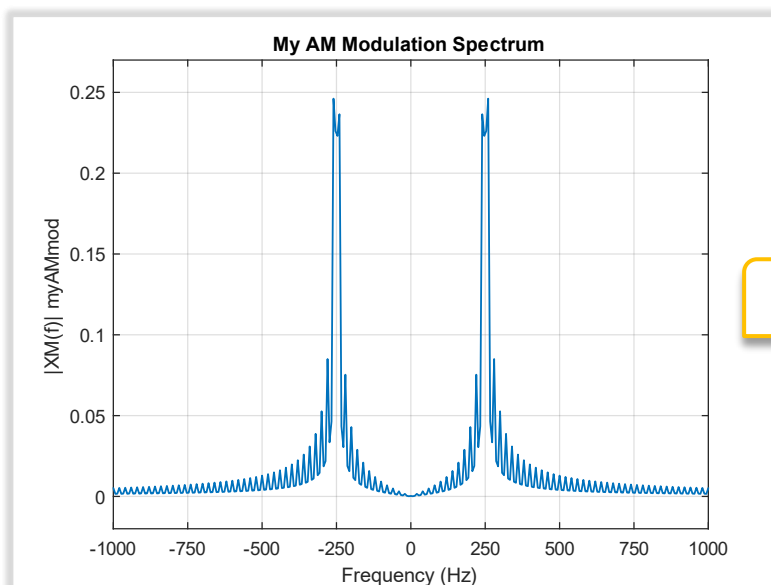
شکل 1

```
Ac_part1 = 1;
mu_part1 = 0.85;
fc_part1 = 250;
Xm_myAMmod = my_AMmod(Xmt, Ac_part1, mu_part1, fc_part1, t);
plot(t, Xm_myAMmod, 'r', 'linewidth', 1);
title('My AM Modulation Function');
xlabel('Time (s)');
ylabel('Xm(t) myAMmod');
set(gca, 'xtick', 0 : 0.05 : 0.15, 'xlim', [0, 0.15]);
set(gca, 'ytick', -1.5 : 0.5 : 1.5, 'ylim', [-2, 2]);
grid on;
```

در این قسمت میخواهیم طیف مربوط به سیگنال پیام  $x_m(t)$  (شکل 2) و سیگنال مدوله شده (شکل 3) را رسم کنیم.



شکل 2



شکل 3

در قسمت بعدی قصد داریم توان و بازدهی مدولاسیون را برای سیگنال مدوله شده بدست آوریم.

با استفاده از روابط تئوری توان سیگنال پیام را میتوان بصورت زیر محاسبه کرد:

$$s_x = \langle x^2(t) \rangle = \frac{1}{t_0} \int_0^{t_0} |x(t)|^2 dt = \frac{1}{0.15} \left( \frac{1^2}{2^2} \times 0.03 + \frac{(-2)^2}{2^2} \times 0.05 \right) = \frac{5}{12}$$

\* دقت شود که ابتدا باید سیگنال پیام، نرمالیزه شود و برای اینکار کفایت سیگنال را بر ماکزیمم اندازه آن تقسیم کنیم، و سپس پس از محاسبه توان، آن را در  $\mu^2$  ضرب میکنیم.

$$\mu^2 \langle x^2(t) \rangle = \mu^2 \times \frac{5}{12} = 0.301$$

در ادامه به محاسبه توان سیگنال مدوله شده میپردازیم:

$$S_T = \frac{A_c^2}{2} (1 + 2\mu \langle x(t) \rangle + \mu^2 \langle x^2(t) \rangle) = \frac{1}{2} \left( 1 + 2 \times 0.85 \times \frac{-1}{6} + 0.85 \times \frac{5}{12} \right) = 0.5088$$

بازدهی مدولاسیون به صورت زیر محاسبه میشود:

$$\text{Modulation Efficiency} = \frac{0.301}{0.5088} = 0.5917 = 59.17\%$$

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
Sx_Xmt_in = 0;
l_Xmt_in = length(Xmt_in);
for i = 1:length(Xmt_in)
    Sx_Xmt_in = Sx_Xmt_in + 1/(l_Xmt_in) * Xmt_in(1, i)^2;
end
Sx_Xmt_in
St_Xm_myAMmod = 0;
l_Xm_myAMmod = length(Xm_myAMmod);
for i = 1:length(Xm_myAMmod)
    St_Xm_myAMmod = St_Xm_myAMmod + 1/(l_Xm_myAMmod) * Xm_myAMmod(1, i)^2;
end
St_Xm_myAMmod
Modulation_Efficiency = Sx_Xmt_in / St_Xm_myAMmod
```

خروجی قطعه کد بالا بصورت زیر خواهد بود:

```
Sx_Xmt_in = 0.3010

St_Xm_myAMmod = 0.5088

Modulation_Efficiency = 0.5917
```

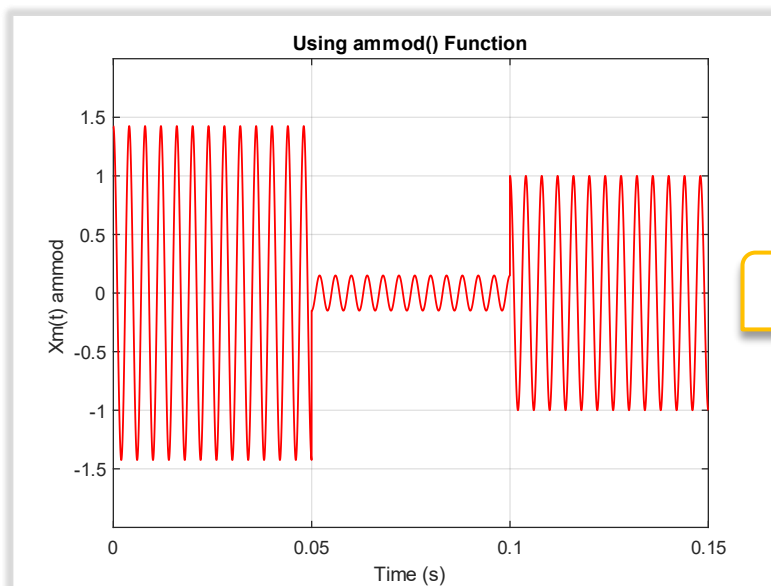
در این قسمت میخواهیم این بار سیگنال پیام را با استفاده از تابع ammod مدوله کنیم و برای اینکار

کفایت ورودی تابع را بصورت زیر در نظر بگیریم:

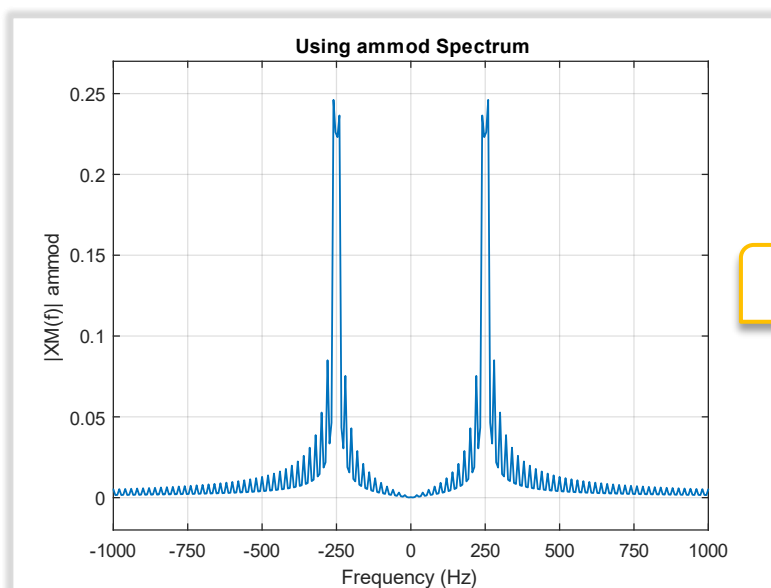
$$x_{in}(t) = \frac{\mu x_m(t)}{\max\{|x_m(t)|\}}$$

```
Xmt_in = mu_part1 * Xmt / max(abs(Xmt));
Xm_ammod = ammod(Xmt_in, fc_part1, fs_part1, 0, 1);
```

در ادامه به رسم آن در حوزه زمان (شکل 4) و فرکانس (شکل 5) میپردازیم:



شکل 4



شکل 5

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
plot(t, Xm_ammod, 'r', 'linewidth', 1);
title('Using ammod() Function');
xlabel('Time (s)');
ylabel('X_m(t) ammod');
set(gca, 'xtick', 0 : 0.05 : 0.15, 'xlim', [0, 0.15]);
set(gca, 'ytick', -1.5 : 0.5 : 1.5, 'ylim', [-2, 2]);
grid on;
```

```
Xmf_ammod = fftshift(fft(Xm_ammod));
l_Xmf_ammod = length(Xmf_ammod);
abs_Xmf_ammod = abs(Xmf_ammod/l_Xmf_ammod);
f3_part1 = -fs_part1/2 : fs_part1/l_Xmf_ammod : fs_part1/2 -
fs_part1/l_Xmf_ammod;
plot(f3_part1, abs_Xmf_ammod, 'linewidth', 1);
title('Using ammod Spectrum');
xlabel('Frequency (Hz)');
ylabel('|XM(f)| ammod');
set(gca,'xtick', -1000 : 250 : 1000, 'xlim', [-1000,1000]);
set(gca,'ytick', 0 : 0.05 : 0.25, 'ylim', [-0.02,0.27]);
grid on;
```

### بخش دوم : مدولاسیون های DSB و SSB

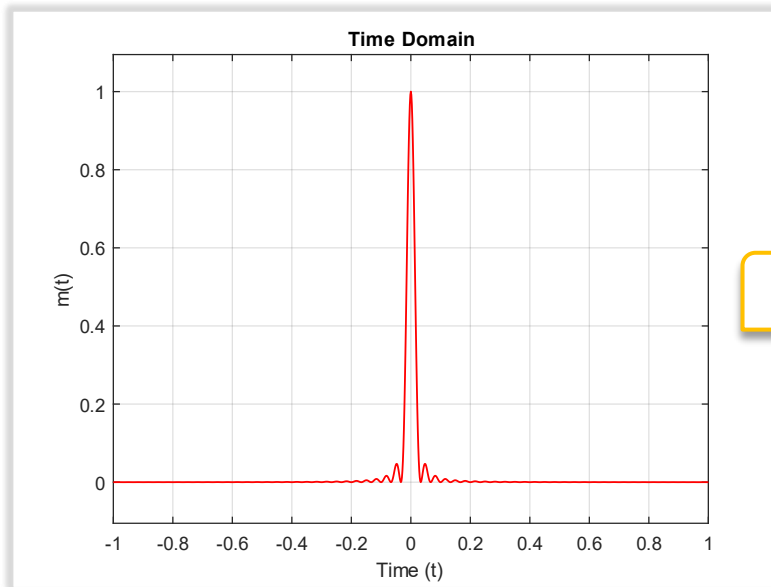
سیگنال پیام  $m(t)$  بصورت زیر تعریف شده است. بازه زمانی سیگنال را  $[-5, 5]$  و فرکانس نمونه برداری را 600Hz در نظر بگیرید.

$$m(t) = \text{sinc}^2(30t)$$

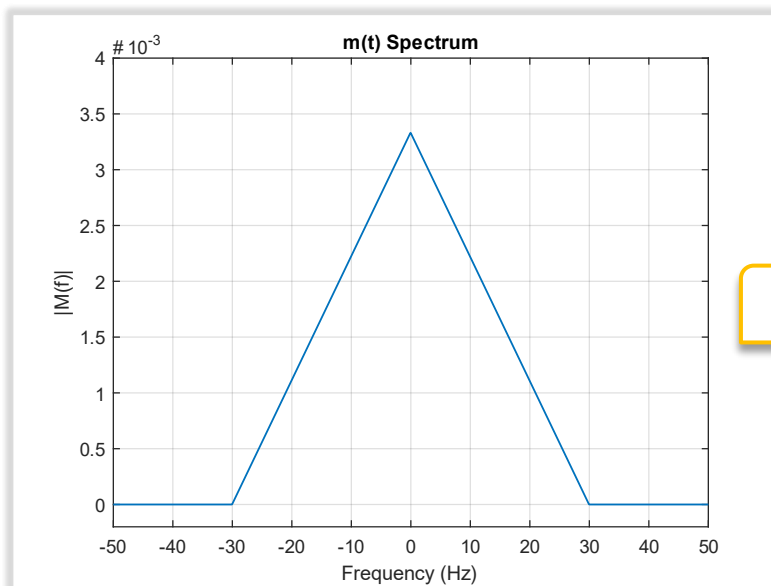
در قسمت اول میخواهیم سیگنال پیام را در حوزه زمان برحسب ثانیه (شکل 6) و تبدیل فوریه آن را برحسب هرتز (شکل 7) رسم کنیم.

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
fs_part2 = 600;
ts_part2 = 1/fs_part2;
t_part2 = -5:ts_part2:5;
mt = sinc(30*t_part2).^2;
plot(t_part2, mt, 'r', 'linewidth', 1);
title('Time Domain');
xlabel('Time (t)');
ylabel('m(t)');
set(gca,'xtick', -1 : 0.2 : 1, 'xlim', [-1,1]);
set(gca,'ytick', 0 : 0.2 : 1.6, 'ylim', [-0.1,1.1]);
grid on;
Mf = fftshift(fft(mt));
l_Mf = length(Mf);
abs_Mf = abs(Mf/l_Mf);
f1_part2 = -fs_part2/2 : fs_part2/l_Mf : fs_part2/2 - fs_part2/l_Mf;
plot(f1_part2, abs_Mf, 'linewidth', 1);
title('m(t) Spectrum');
xlabel('Frequency (Hz)');
ylabel('|M(f)|');
set(gca,'xtick', -50 : 10 : 50, 'xlim', [-50,50]);
set(gca,'ytick', -0.5*10^-3 : 0.5*10^-3 : 4*10^-3, 'ylim', [-0.2*10^-3,4*10^-3]);
grid on;
```



شکل 6



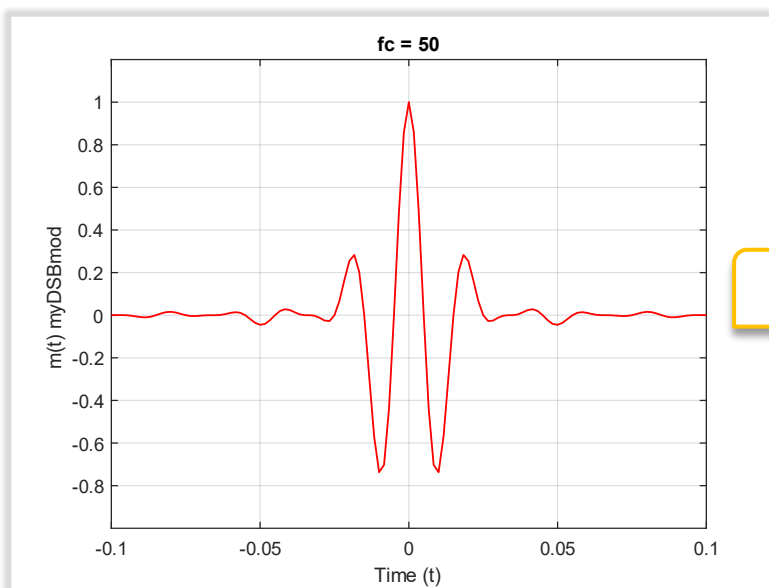
شکل 7

در این قسمت میخواهیم تابعی بنویسیم که سیگنال پیام  $m(t)$ ، دامنه موج حامل  $A_c$ ، اندیس مدولاسیون  $\mu$  و فرکانس موج حامل  $f_c$  را بگیرد و سیگنال مدوله شده DSB را برگرداند. البته برای سادگی کار  $t$  را هم به عنوان ورودی تابع در نظر گرفته‌ام.

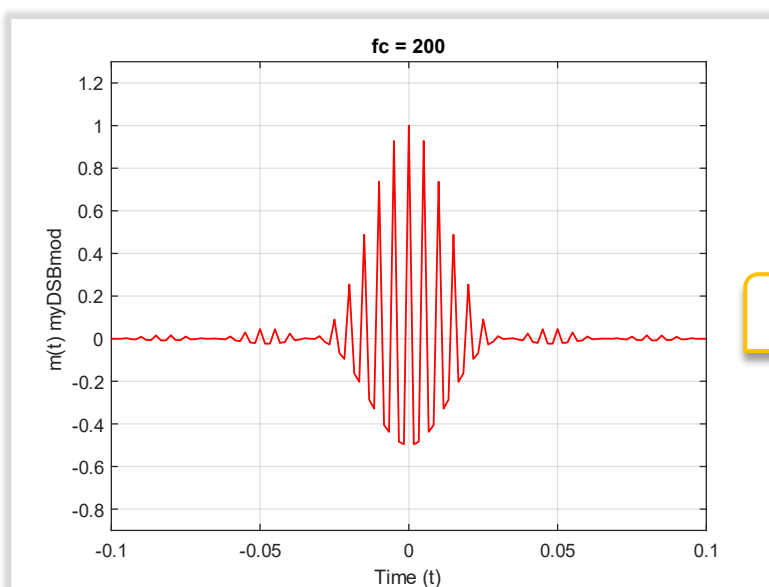
```
function Xc = my_DSBmod(x, Ac, fc, t)
    Xc = Ac*x.*cos(2*pi*fc*t);
end
```

در ادامه با استفاده از تابع قسمت قبل، سیگنال پیام  $m(t)$  را با فرکانس‌های  $f_c = \{50, 200, 600, 1200\}$  مدوله کرده و رسم میکنیم.

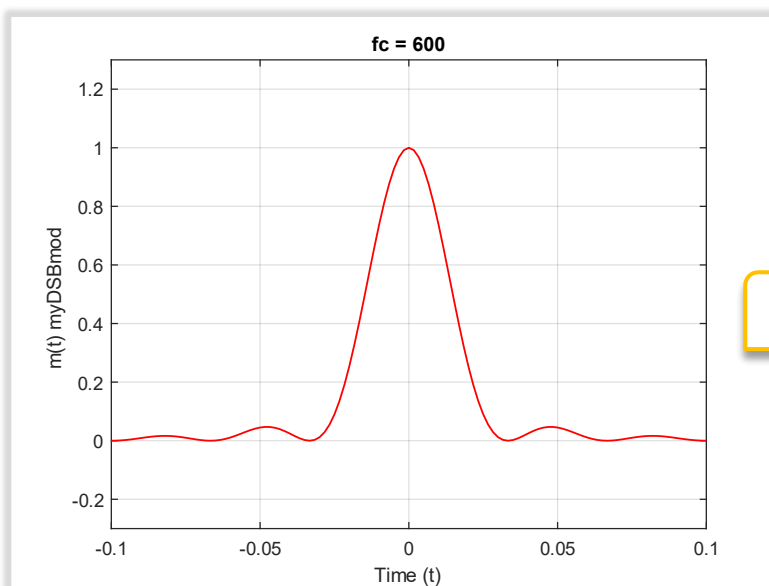




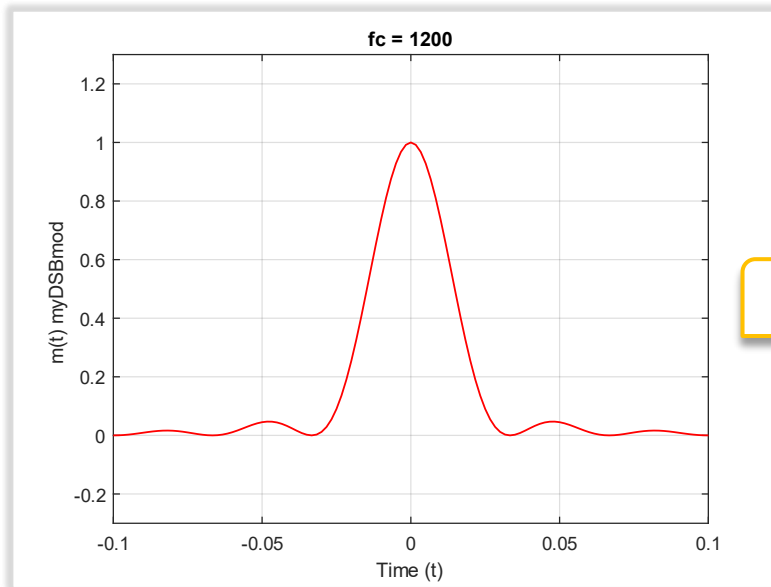
شکل 8



شکل 9



شکل 10



شکل 11

\* همانطور که انتظار داشتیم سیگنال مدوله شده با فرکانس موج حامل بزرگتر از نصف فرکانس نمونه‌برداری، صحیح نمیباشد.

$$\max\{f_c\} = \frac{f_s}{2}$$

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
Ac_part2 = 1;
fc_part2 = 50;
mt_DSBmod = my_DSBmod(mt, Ac_part2, fc_part2, t_part2);
plot(t_part2, mt_DSBmod, 'r', 'linewidth', 1);
title('fc = 50');
xlabel('Time (t)');
ylabel('m(t) myDSBmod');
set(gca, 'xtick', -0.2 : 0.05 : 0.2, 'xlim', [-0.1, 0.1]);
set(gca, 'ytick', -0.8 : 0.2 : 1, 'ylim', [-1, 1.2]);
grid on;

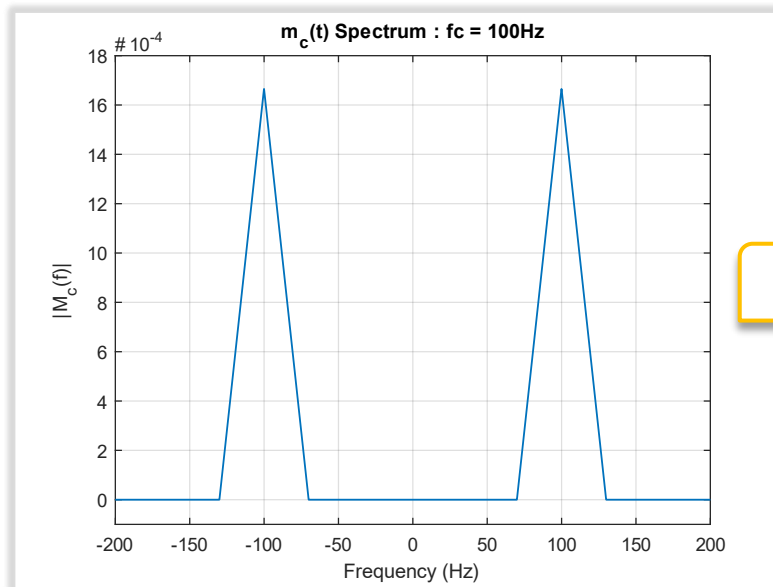
fc_part2 = 200;
mt_DSBmod = my_DSBmod(mt, Ac_part2, fc_part2, t_part2);
plot(t_part2, mt_DSBmod, 'r', 'linewidth', 1);
title('fc = 200');
xlabel('Time (t)');
ylabel('m(t) myDSBmod');
set(gca, 'xtick', -0.2 : 0.05 : 0.2, 'xlim', [-0.1, 0.1]);
set(gca, 'ytick', -1.2 : 0.2 : 1.6, 'ylim', [-0.9, 1.3]);
grid on;

fc_part2 = 600;
mt_DSBmod = my_DSBmod(mt, Ac_part2, fc_part2, t_part2);
plot(t_part2, mt_DSBmod, 'r', 'linewidth', 1);
title('fc = 600');
xlabel('Time (t)');
```

```
ylabel('m(t) myDSBmod');
set(gca,'xtick', -0.2 : 0.05 : 0.2, 'xlim', [-0.1,0.1]);
set(gca,'ytick', -1.2 : 0.2 : 1.6, 'ylim', [-0.3,1.3]);
grid on;

fc_part2 = 1200;
mt_DSBmod = my_DSBmod(mt, Ac_part2, fc_part2, t_part2);
plot(t_part2, mt_DSBmod, 'r', 'linewidth', 1);
title('fc = 1200');
xlabel('Time (t)');
ylabel('m(t) myDSBmod');
set(gca,'xtick', -0.2 : 0.05 : 0.2, 'xlim', [-0.1,0.1]);
set(gca,'ytick', -1.2 : 0.2 : 1.6, 'ylim', [-0.3,1.3]);
grid on;
```

در این قسمت می‌خواهیم سیگنال پیام را با  $f_c = 100\text{Hz}$  و  $A_c = 1$  مدوله کنیم و تبدیل فوریه سیگنال خروجی را برحسب هرتز رسم نماییم. (شکل 12)



شکل 12

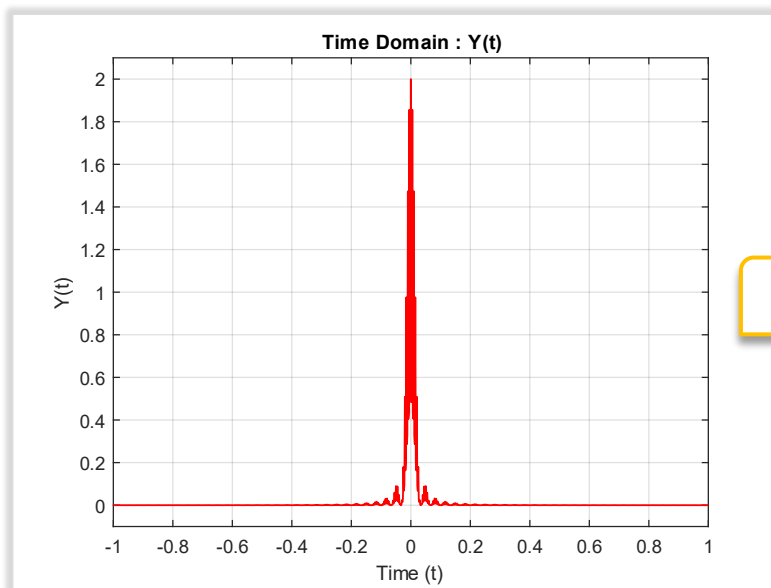
قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
fc_part2 = 100;
mt_DSBmod = my_DSBmod(mt, Ac_part2, fc_part2, t_part2);
Mf_DSBmod = fftshift(fft(mt_DSBmod));
l_Mf_DSBmod = length(Mf_DSBmod);
abs_Mf_DSBmod = abs(Mf_DSBmod/l_Mf_DSBmod);
f2_part2 = -fs_part2/2 : fs_part2/l_Mf_DSBmod : fs_part2/2 - fs_part2/l_Mf_DSBmod;
plot(f2_part2, abs_Mf_DSBmod, 'linewidth', 1);
title('m_c(t) Spectrum : fc = 100Hz');
xlabel('Frequency (Hz)');
ylabel('|M_c(f)|');
set(gca,'xtick', -200 : 50 : 200, 'xlim', [-200,200]);
set(gca,'ytick', -0.2*10^-3 : 0.2*10^-3 : 2*10^-3, 'ylim', [-0.1*10^-3,1.8*10^-3]);
grid on;
```

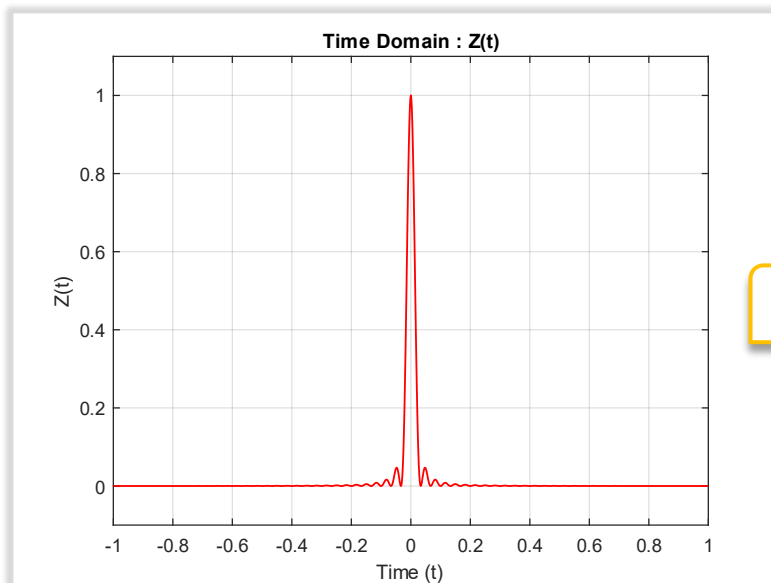
در این قسمت تابعی مینویسیم که سیگنال مدوله شده  $x_c(t)$ ، دامنه موج حامل  $A_c$ ، فرکانس موج حامل  $f_c$  را بگیرد و سیگنال پیام را استخراج کند.

```
function [Zt, Yt] = my_DSBdemod(Xc, Ac, fc, t)
    Yt = 2*Ac*Xc.*cos(2*pi*fc*t);
    Zt = lowpass(Yt,50,600);
end
```

در ادامه می‌خواهیم با استفاده از تابع قسمت قبل، سیگنال پیام را از سیگنال مدوله شده بازیابی کنیم. ابتدا سیگنال‌های  $Y(t)$  و  $Z(t)$  را در حوزه زمان رسم می‌کنیم. (شکل 13 و 14)



شکل 13



شکل 14

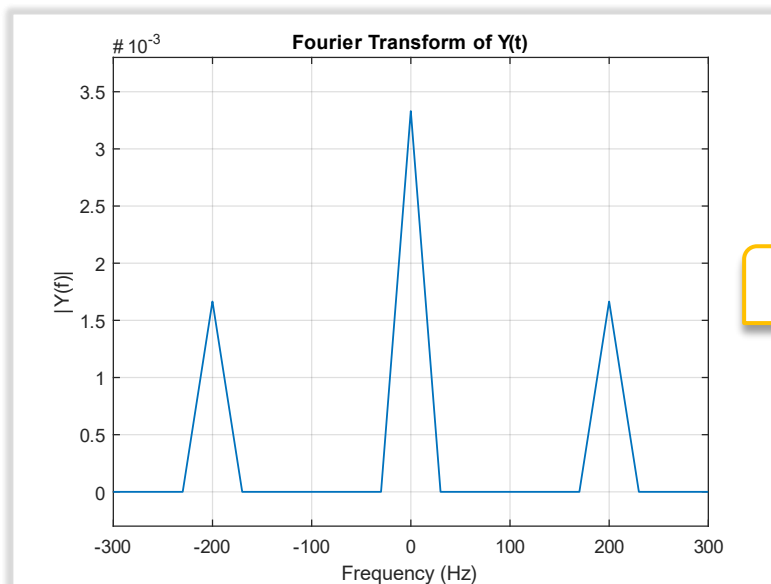
قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
[Zt, Yt] = my_DSBdemod(mt_DSBmod, Ac_part2, fc_part2, t_part2);
```

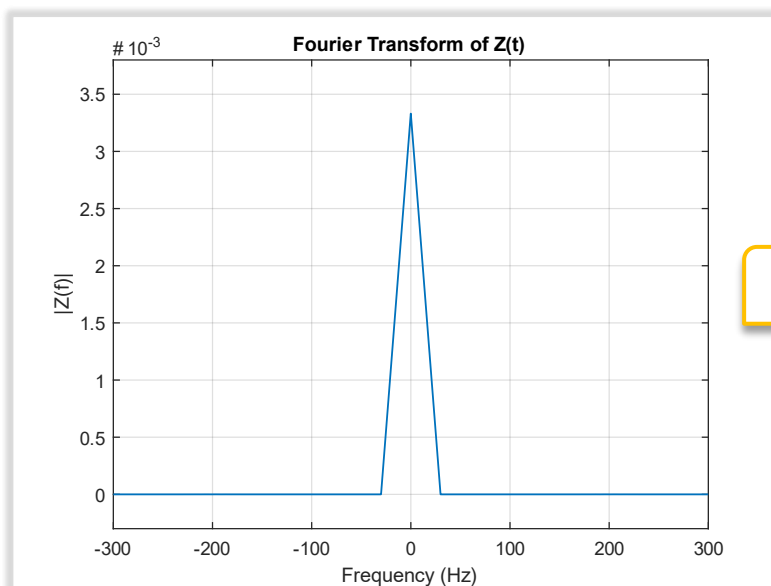
```
plot(t_part2, Yt, 'r', 'linewidth', 1);
title('Time Domain : Y(t)');
xlabel('Time (t)');
ylabel('Y(t)');
set(gca,'xtick', -1 : 0.2 : 1, 'xlim', [-1,1]);
set(gca,'ytick', 0 : 0.2 : 2.6, 'ylim', [-0.1,2.1]);
grid on;

plot(t_part2, Zt, 'r', 'linewidth', 1);
title('Time Domain : Z(t)');
xlabel('Time (t)');
ylabel('Z(t)');
set(gca,'xtick', -1 : 0.2 : 1, 'xlim', [-1,1]);
set(gca,'ytick', 0 : 0.2 : 2.6, 'ylim', [-0.1,1.1]);
grid on;
```

در ادامه سیگنال های  $Y(f)$  و  $Z(f)$  را در حوزه فرکانس رسم میکنیم. (شکل 15 و 16)



شکل 15



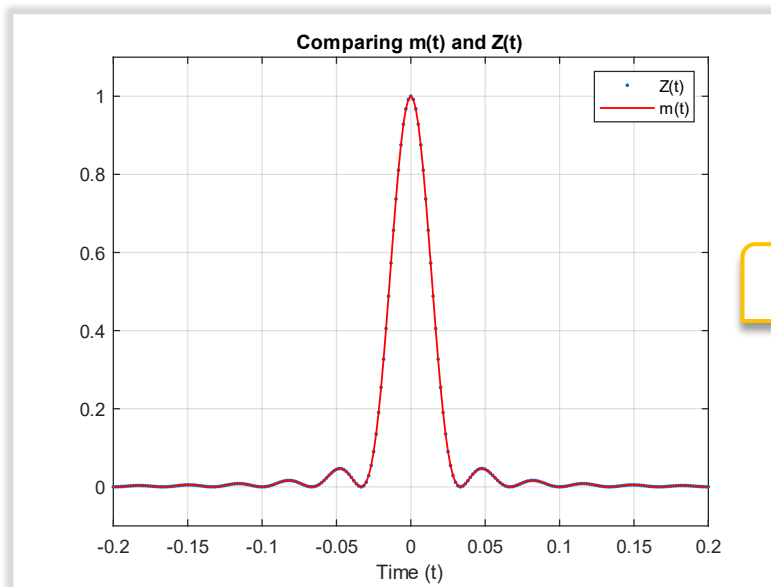
شکل 16

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
Yf = fftshift(fft(Yt));
l_Yf = length(Yf);
abs_Yf = abs(Yf/l_Yf);
f3_part2 = -fs_part2/2 : fs_part2/l_Yf : fs_part2/2 - fs_part2/l_Yf;
plot(f3_part2, abs_Yf, 'linewidth', 1);
title('Fourier Transform of Y(t)');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
set(gca, 'xtick', -300 : 100 : 300, 'xlim', [-300,300]);
set(gca, 'ytick', -0.5*10^-3 : 0.5*10^-3 : 3.5*10^-3, 'ylim', [-0.3*10^-3,3.8*10^-3]);
grid on;

Zf = fftshift(fft(Zt));
l_Zf = length(Zf);
abs_Zf = abs(Zf/l_Zf);
f4_part2 = -fs_part2/2 : fs_part2/l_Zf : fs_part2/2 - fs_part2/l_Zf;
plot(f4_part2, abs_Zf, 'linewidth', 1);
title('Fourier Transform of Z(t)');
xlabel('Frequency (Hz)');
ylabel('|Z(f)|');
set(gca, 'xtick', -300 : 100 : 300, 'xlim', [-300,300]);
set(gca, 'ytick', -0.5*10^-3 : 0.5*10^-3 : 3.5*10^-3, 'ylim', [-0.3*10^-3,3.8*10^-3]);
grid on;
```

در ادامه سیگنال های  $m(t)$  و  $Z(t)$  را در یک نمودار در حوزه زمان رسم میکنیم. (شکل 17)



شکل 17

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
plot(t_part2, Zt, '.', t_part2, mt, 'r', 'linewidth', 1);
title('Comparing m(t) and Z(t)');
legend('Z(t)', 'm(t)');
```

```
xlabel('Time (t)');
set(gca,'xtick', -0.2 : 0.05 : 0.2, 'xlim', [-0.2,0.2]);
set(gca,'ytick', 0 : 0.2 : 2.6, 'ylim', [-0.1,1.1]);
grid on;
```

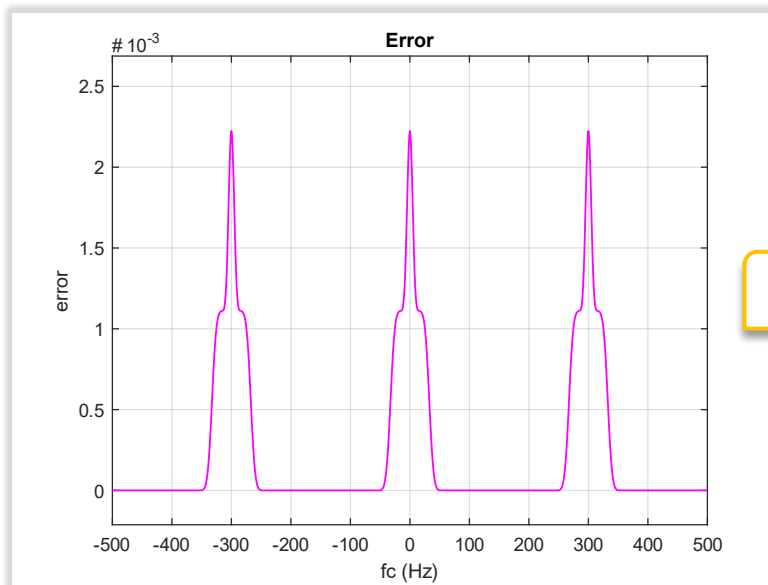
اختلاف دو سیگنال  $m(t)$  و  $Z(t)$  با استفاده از معیار میانگین مجذور خطا بصورت زیر بدست می آید:

```
error_myDSBdemod = immse(mt,Zt)
error_myDSBdemod = 2.4959e-10
```

در این قسمت میخواهیم نمودار خطای بدست آمده در قسمت قبل را نسبت به فرکانس موج حامل  $f_c = [-500,500]$  رسم کنیم (شکل 18) و بهترین مقدار برای فرکانس حامل را تعیین کنیم.

```
fcx_part2 = -500:500;
error = zeros(1, length(fcx_part2));
for fci = fcx_part2
    mti_DSBmod = my_DSBmod(mt, Ac_part2, fci, t_part2);
    [Zti, Yti] = my_DSBdemod(mti_DSBmod, Ac_part2, fci, t_part2);
    error(1, fci+501) = immse(mt,Zti);
end

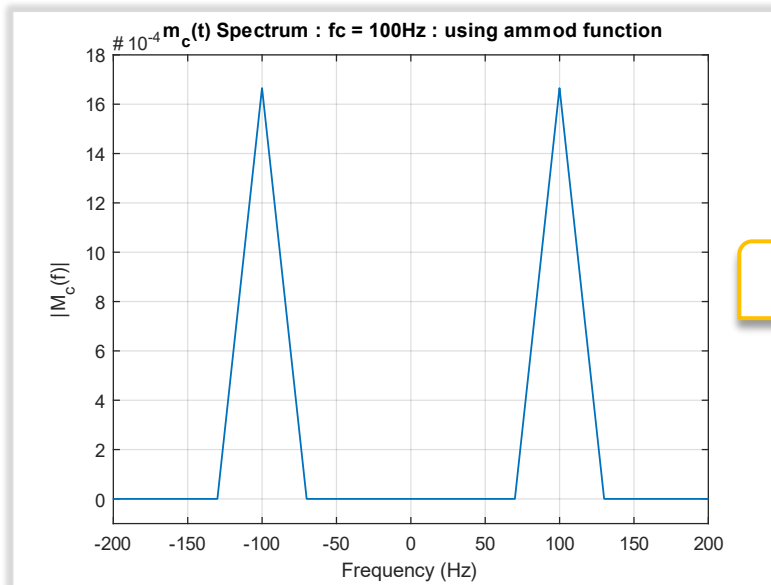
plot(fcx_part2, error, 'm', 'linewidth', 1);
title('Error');
xlabel('fc (Hz)');
ylabel('error');
set(gca,'xtick', -500 : 100 : 500, 'xlim', [-500,500]);
set(gca,'ytick', -0.5*10^-3 : 0.5*10^-3 : 2.5*10^-3, 'ylim', [-0.2*10^-3,2.7*10^-3]);
grid on;
```



شکل 18

همانطور که از نمودار شکل 18 مشخص است، بهترین بازه برای انتخاب  $f_c$ ،  $[100,200]$  میباشد.

در این قسمت میخواهیم سیگنال پیام را با  $f_c = 100\text{Hz}$  و  $A_c = 1$  را با استفاده از تابع `ammod` در متلب مدوله کنیم و تبدیل فوریه سیگنال خروجی را بر حسب هرتز رسم نماییم. (شکل 19)



شکل 19

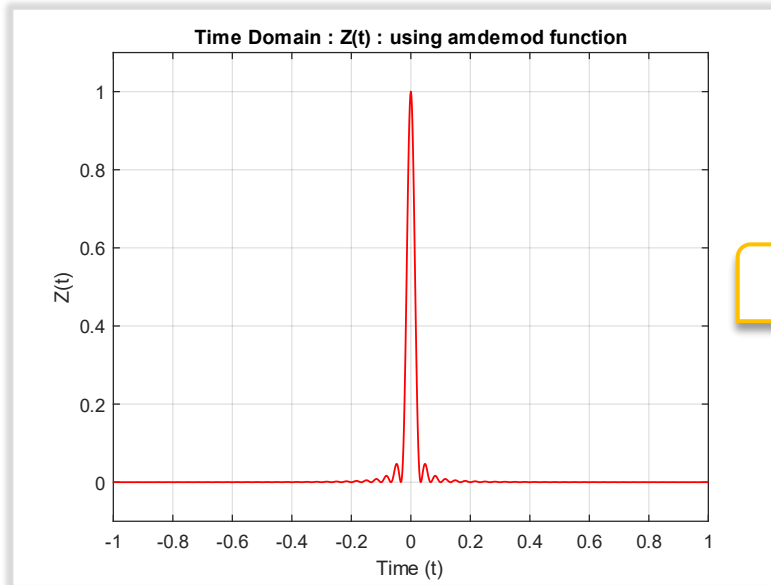
قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
mt_ammod = ammod(mt,fc_part2,fs_part2);
Mf_ammod = fftshift(fft(mt_ammod));
l_Mf_ammod = length(Mf_ammod);
abs_Mf_ammod = abs(Mf_ammod/l_Mf_ammod);
f5_part2 = -fs_part2/2 : fs_part2/l_Mf_ammod : fs_part2/2 -
fs_part2/l_Mf_ammod;

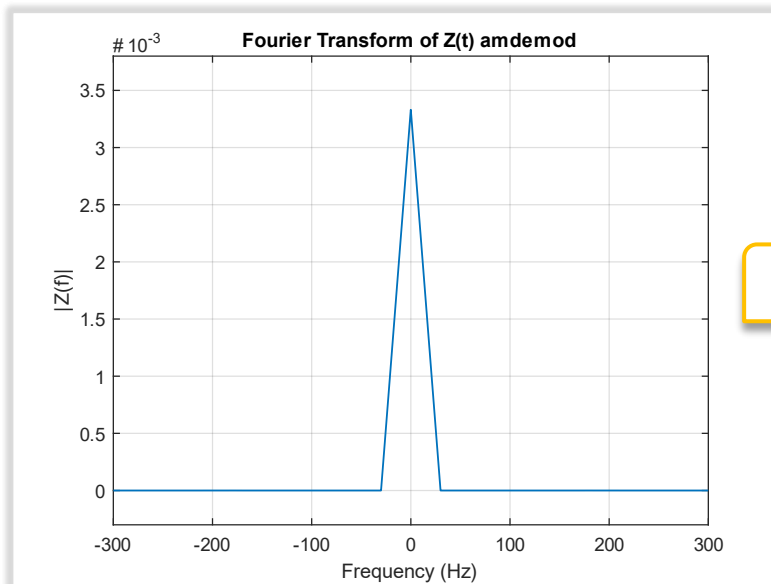
plot(f5_part2, abs_Mf_ammod, 'linewidth', 1);
title('m_c(t) Spectrum : fc = 100Hz : using ammod function');
xlabel('Frequency (Hz)');
ylabel('|M_c(f)|');
set(gca,'xtick', -200 : 50 : 200, 'xlim', [-200,200]);
set(gca,'ytick', -0.2*10^-3 : 0.2*10^-3 : 2*10^-3, 'ylim', [-0.1*10^-3,1.8*10^-3]);
grid on;
```

در ادامه میخواهیم با استفاده از تابع `amdmod`، سیگنال پیام را از سیگنال مدوله شده بازیابی کنیم. و در نهایت سیگنال بازیابی شده را در حوزه زمان (شکل 20) و فرکانس (شکل 21) رسم میکنیم.





شکل 20



شکل 21

همانطور که از نمودارهای شکل 20 و شکل 21 مشخص است، سیگنال اصلی به خوبی بازیابی شده است.

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
Zt_amdemod = amdemod(mt_ammod, fc_part2, fs_part2);

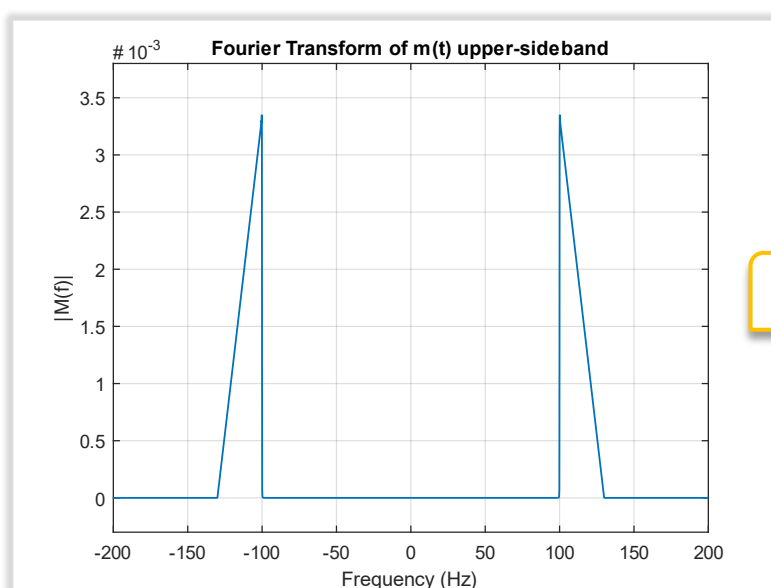
plot(t_part2, Zt_amdemod, 'r', 'linewidth', 1);
title('Time Domain : Z(t) : using amdemod function');
xlabel('Time (t)');
ylabel('Z(t)');
set(gca, 'xtick', -1 : 0.2 : 1, 'xlim', [-1,1]);
set(gca, 'ytick', 0 : 0.2 : 2.6, 'ylim', [-0.1,1.1]);
grid on;

Zf_amdemod = fftshift(fft(Zt_amdemod));
l_Zf_amdemod = length(Zf_amdemod);
```

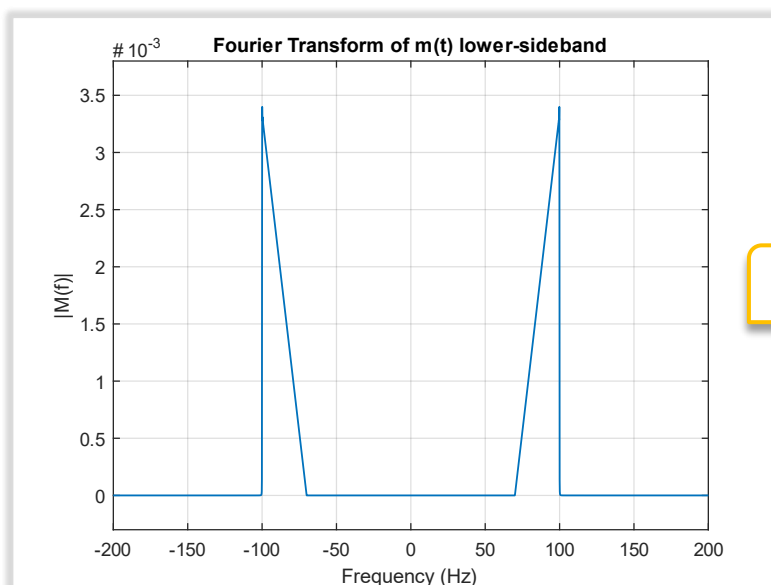
```
abs_Zf_amdemod = abs(Zf_amdemod/l_Zf_amdemod);
f6_part2 = -fs_part2/2 : fs_part2/l_Zf_amdemod : fs_part2/2 -
fs_part2/l_Zf_amdemod;

plot(f6_part2, abs_Zf_amdemod, 'linewidth', 1);
title('Fourier Transform of Z(t) amdemod');
xlabel('Frequency (Hz)');
ylabel('|Z(f)|');
set(gca,'xtick', -300 : 100 : 300, 'xlim', [-300,300]);
set(gca,'ytick', -0.5*10^-3 : 0.5*10^-3 : 3.5*10^-3, 'ylim', [-0.3*10^-3,3.8*10^-3]);
grid on;
```

در این قسمت میخواهیم سیگنال پیام را با  $f_c = 100\text{Hz}$  و  $A_c = 1$  را با استفاده از تابع `ssbmod` در متلب به دو شکل `lower-sideband` (شکل 22) و `upper-sideband` (شکل 23) مدوله کنیم و تبدیل فوریه سیگنال خروجی را برحسب هرتز رسم نماییم.



شکل 23



شکل 22

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

```
mt_ssbmod_lower = ssbmod(mt, fc_part2, fs_part2);
Mf_ssbmod_lower = fftshift(fft(mt_ssbmod_lower));
l_Mf_ssbmod_lower = length(Mf_ssbmod_lower);
abs_Mf_ssbmod_lower = abs(Mf_ssbmod_lower/l_Mf_ssbmod_lower);
f7_part2 = -fs_part2/2 : fs_part2/l_Mf_ssbmod_lower : fs_part2/2 -
fs_part2/l_Mf_ssbmod_lower;

plot(f7_part2, abs_Mf_ssbmod_lower, 'linewidth', 1);
title('Fourier Transform of m(t) lower-sideband');
xlabel('Frequency (Hz)');
ylabel('|M(f)|');
set(gca,'xtick', -200 : 50 : 200, 'xlim', [-200,200]);
set(gca,'ytick', -0.5*10^-3 : 0.5*10^-3 : 3.5*10^-3, 'ylim', [-0.3*10^-
3,3.8*10^-3]);
grid on;

mt_ssbmod_upper = ssbmod(mt, fc_part2, fs_part2, 0, 'upper');
Mf_ssbmod_upper = fftshift(fft(mt_ssbmod_upper));
l_Mf_ssbmod_upper = length(Mf_ssbmod_upper);
abs_Mf_ssbmod_upper = abs(Mf_ssbmod_upper/l_Mf_ssbmod_upper);
f8_part2 = -fs_part2/2 : fs_part2/l_Mf_ssbmod_upper : fs_part2/2 -
fs_part2/l_Mf_ssbmod_upper;

plot(f8_part2, abs_Mf_ssbmod_upper, 'linewidth', 1);
title('Fourier Transform of m(t) upper-sideband');
xlabel('Frequency (Hz)');
ylabel('|M(f)|');
set(gca,'xtick', -200 : 50 : 200, 'xlim', [-200,200]);
set(gca,'ytick', -0.5*10^-3 : 0.5*10^-3 : 3.5*10^-3, 'ylim', [-0.3*10^-
3,3.8*10^-3]);
grid on;
```

### بخش سوم : آشنایی با تبدیل هیلبرت و آشکارساز پوش

تبدیل هیلبرت گسسته بصورت زیر قابل محاسبه است:

$$H\{x[n]\} = \hat{x}[n] = \sum_{i=-\infty}^{\infty} h[i]x[n-i]$$

$$h[n] = \begin{cases} \frac{2}{\pi n} & n \text{ odd} \\ 0 & n \text{ even} \end{cases}$$

سیگنال متناوب پیام بصورت زیر تعریف شده است.

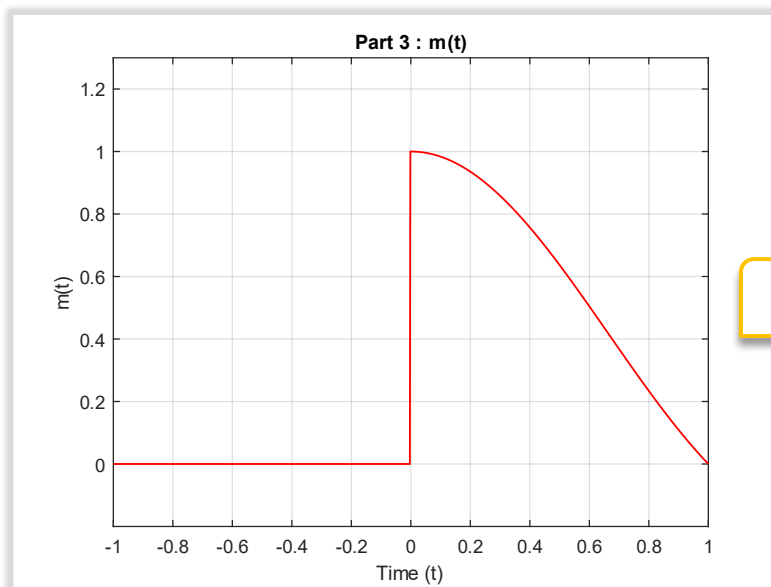
$$m(t) = \begin{cases} \text{sinc}(t) & 0 \leq t \leq 1 \\ 0 & -1 \leq t \leq 0 \end{cases}$$

با استفاده از تابع hilbert، تبدیل هیلبرت سیگنال پیام را بدست می آوریم.

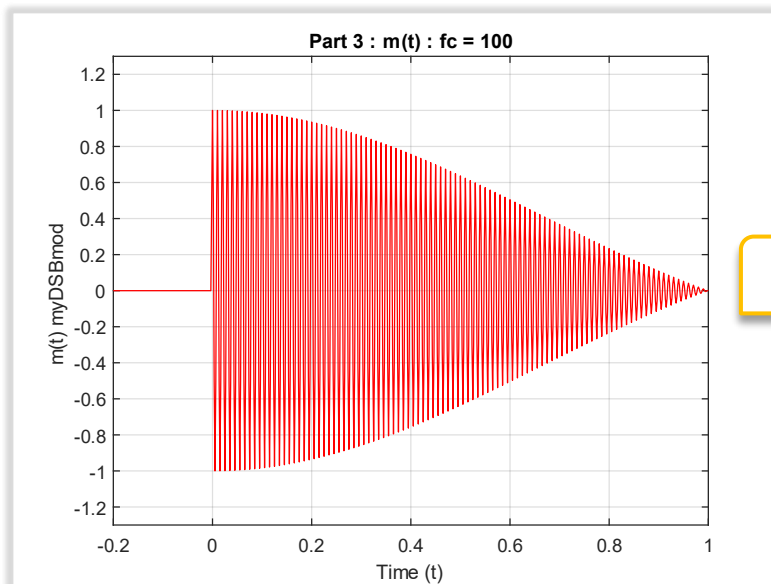
```
fs_part3 = 600;
ts_part3 = 1/fs_part3;
t_part3 = -1:ts_part3:1;
mt_part3 = sinc(t_part3);
mt_part3(1, 1: fs_part3 - 1) = 0;
```

```
hmt_part3 = hilbert(mt_part3);
```

در ادامه می‌خواهیم سیگنال پیام (شکل 24) را با  $f_c = 100\text{Hz}$  و  $A_c = 1$  بصورت DSB مدوله کنیم و سیگنال خروجی را در حوزه زمان رسم نماییم. (شکل 25)



شکل 24



شکل 25

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

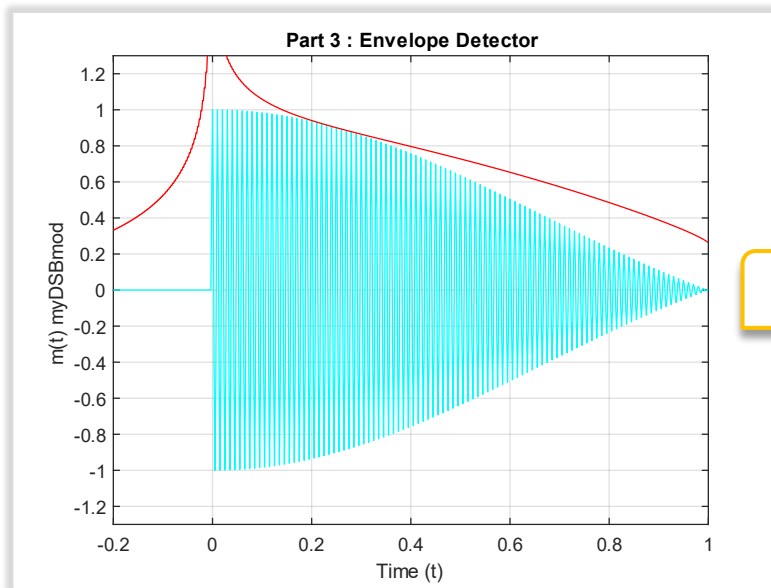
```
plot(t_part3,mt_part3, 'r', 'linewidth', 1);
title('Part 3 : m(t)');
xlabel('Time (t)');
```

```
ylabel('m(t)');
set(gca,'xtick', -1 : 0.2 : 1, 'xlim', [-1,1]);
set(gca,'ytick', 0 : 0.2 : 1.6, 'ylim', [-0.2,1.3]);
grid on;

Ac_part3 = 1;
fc_part3 = 100;
mt_part3_DSBmod = my_DSBmod(mt_part3, Ac_part3, fc_part3, t_part3);
plot(t_part3,mt_part3_DSBmod, 'r');
title('Part 3 : m(t) : fc = 100');
xlabel('Time (t)');
ylabel('m(t) myDSBmod');
set(gca,'xtick', -1 : 0.2 : 1, 'xlim', [-0.2,1]);
set(gca,'ytick', -1.2 : 0.2 : 1.6, 'ylim', [-1.3,1.3]);
grid on;
```

در این قسمت میخواهیم با استفاده از آشکارساز پوش، سیگنال پیام را بازیابی کنیم. مشخصاً این سیگنال قابل بازیابی است اما این که چطور میشود آنرا بازیابی کرد، جای بحث دارد.

با بررسی نمودار شکل 26 میتوان گفت بازیابی به درستی صورت نگرفته است. اما این پایان کار نیست!



شکل 26

قطعه کد زده شده برای این قسمت بصورت زیر میباشد:

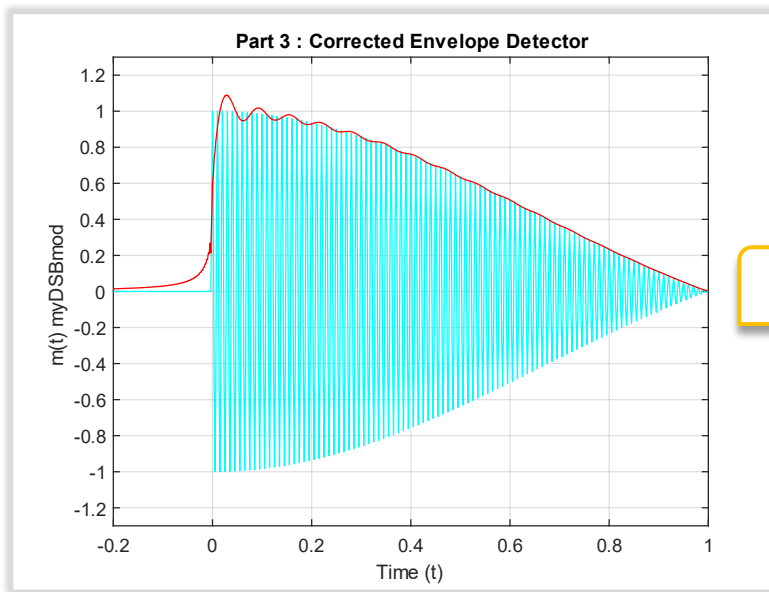
```
plot(t_part3,mt_part3_DSBmod,'c',t_part3,abs(hmt_part3),'r');
title('Part 3 : Envelope Detector');
xlabel('Time (t)');
ylabel('m(t) myDSBmod');
set(gca,'xtick', -1 : 0.2 : 1, 'xlim', [-0.2,1]);
set(gca,'ytick', -1.2 : 0.2 : 1.6, 'ylim', [-1.3,1.3]);
grid on;
```

ما به شما تبدیل هیلبرت اصلاح شده را معرفی میکنیم که بصورت زیر تعریف میشود:

```
hmt_part3_corrected = hilbert(mt_part3.*sin(fc_part3*t_part3));
```

با بررسی نمودار شکل 27 میتوان گفت بازایی سیگنال پیام به خوبی انجام شده است.

```
plot(t_part3,mt_part3_DSBmod,'c',t_part3,abs(hmt_part3_corrected),'r');
title('Part 3 : Corrected Envelope Detector');
xlabel('Time (t)');
ylabel('m(t) myDSBmod');
set(gca,'xtick', -1 : 0.2 : 1, 'xlim', [-0.2,1]);
set(gca,'ytick', -1.2 : 0.2 : 1.6, 'ylim', [-1.3,1.3]);
grid on;
```



شکل 27