

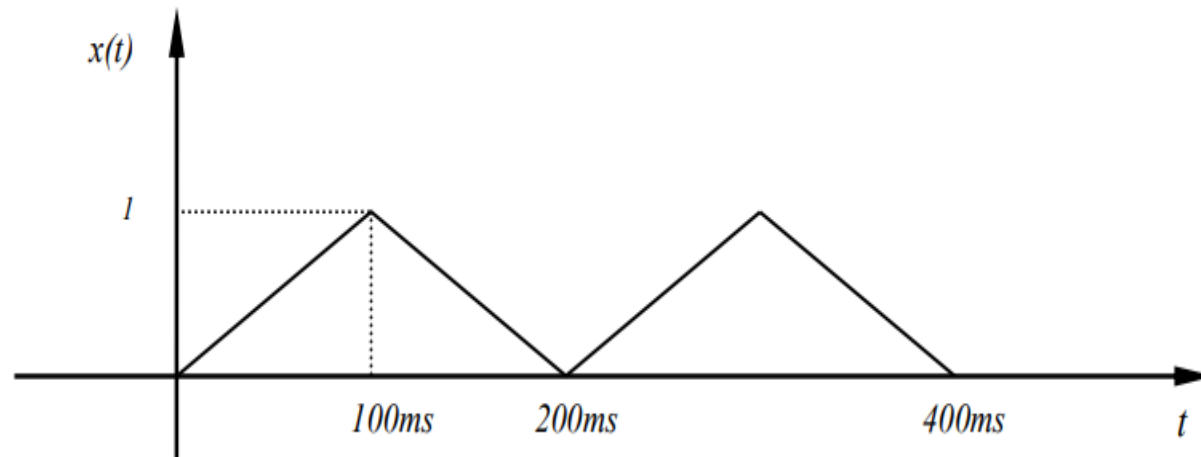
# Communication 1 project

Nima Jahanbازfard

400113020

Dr.Kazemi

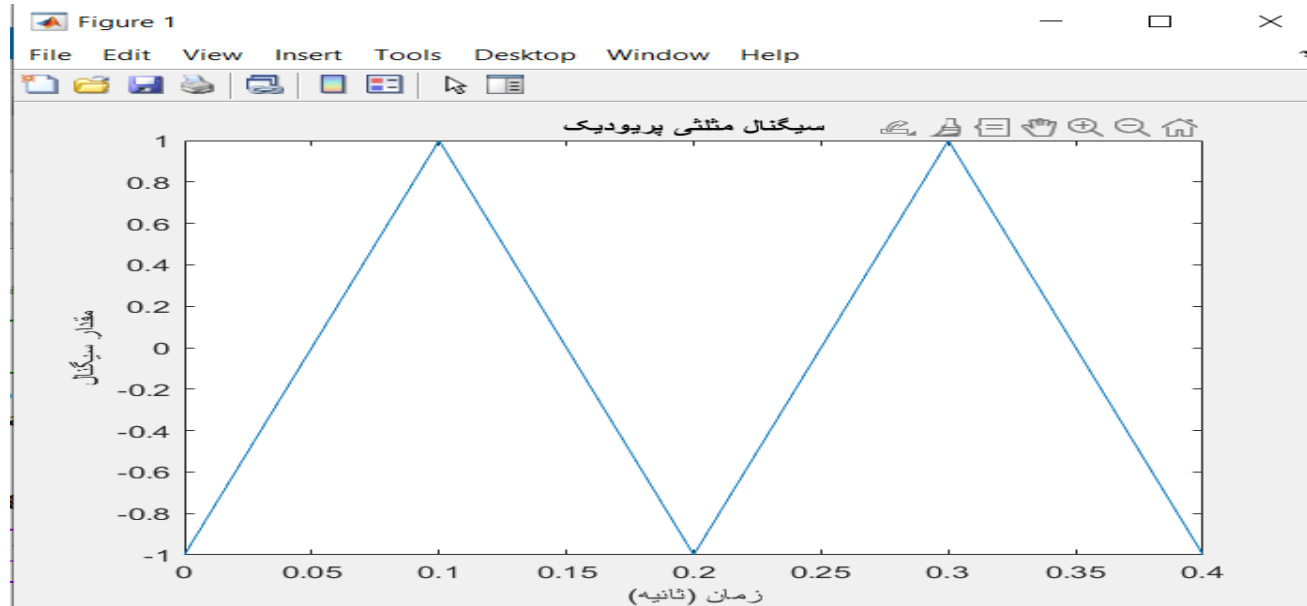
1- یک سیگنال پیام مثالی به صورت زیر شبیه سازی کنید:



```

1  % پارامترها
2  Fs = 1000; % نرخ نمونه برداری (1kHz)
3  f = 5; % فرکانس سیگنال (5Hz)
4
5  % ایجاد سیگنال مثلثی پریودیک
6  t = 0:1/Fs:0.4; % بازه زمانی
7  triangularSignal = sawtooth(2*pi*f*t, 0.5);
8
9  % نمایش سیگنال
10 plot(t, triangularSignal);
11 title('سیگنال مثلثی پریودیک');
12 xlabel('زمان (ثانیه)');
13 ylabel('مقدار سیگنال');

```



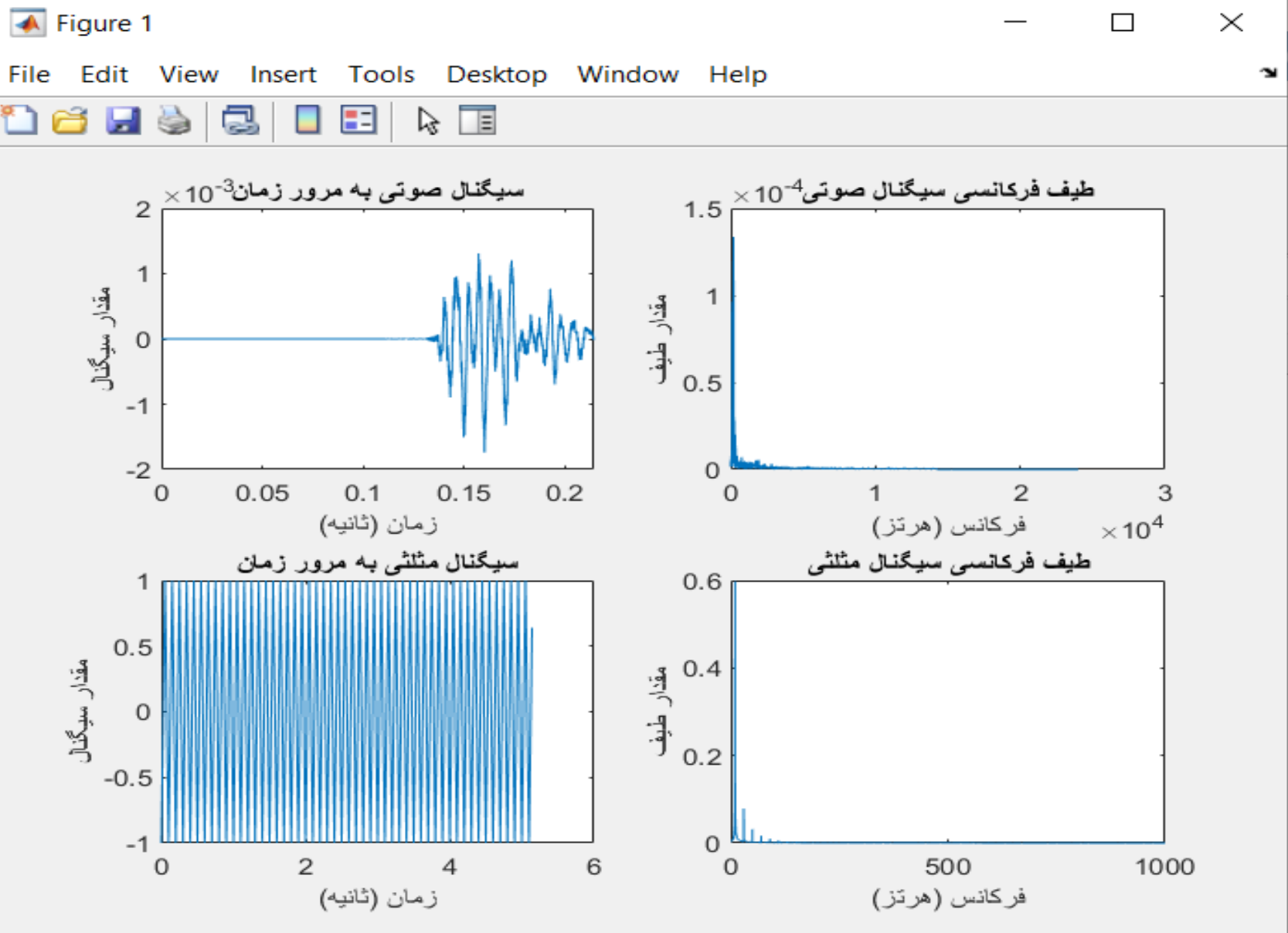
2- با استفاده از میکروفن صدای خود را در حالیکه خودتان را معرفی میکنید ضبط کنید(ذکر نام و شماره دانشجویی) و فایل مربوطه را به عنوان ورودی بخشهای بعد استفاده کنید.  
نمودار سیگنال بر حسب زمان و طیف آن بر حسب فرکانس) با استفاده از دستور FFT و را برای دو سیگنال فوق رسم کنیم فمر کنید فرکانس ناونه برداری برابر 500KHz باشد.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7
8 % پارامترها برای سیگنال مثلثی
9 duration = length(audioData) / Fs_audio;
10 Fs_triangle = 2e3; % (2 کیلوهرتز) نرخ نمونه برداری برای سیگنال مثلثی
11 T_triangle = 1/Fs_triangle; % دوره سیگنال مثلثی
12
13 % ایجاد سیگنال مثلثی
14 t_triangle = 0:T_triangle:duration;
15 triangularSignal = sawtooth(2*pi*10*t_triangle, 0.5); % یک سیگنال مثلثی با فرکانس 10 هرتز
16
17 % تطبیق طول دو سیگنال
18 audioData = audioData(1:min(length(audioData), length(t_triangle)));
19 triangularSignal = triangularSignal(1:min(length(audioData), length(t_triangle)));
20
21 % نمودار سیگنال صوتی به مرور زمان
22 figure;
23 subplot(2, 2, 1);
24 plot((0:length(audioData)-1) / Fs_audio, audioData);
25 title('سیگنال صوتی به مرور زمان');
26 xlabel('زمان (ثانیه)');
```

```

27 ylabel('مقدار سیگنال');
28
29 % نمودار طیف فرکانسی سیگنال صوتی
30 subplot(2, 2, 2);
31 N_audio = length(audioData);
32 frequencies_audio = linspace(0, Fs_audio/2, N_audio/2);
33 Y_audio = fft(audioData);
34 magnitude_audio = 2/N_audio * abs(Y_audio(1:N_audio/2));
35 plot(frequencies_audio, magnitude_audio);
36 title('طیف فرکانسی سیگنال صوتی');
37 xlabel('فرکانس (هرتز)');
38 ylabel('مقدار طیف');
39
40 % نمودار سیگنال مثلثی به مرور زمان
41 subplot(2, 2, 3);
42 plot(t_triangle, triangularSignal);
43 title('سیگنال مثلثی به مرور زمان');
44 xlabel('زمان (ثانیه)');
45 ylabel('مقدار سیگنال');
46
47 % نمودار طیف فرکانسی سیگنال مثلثی
48 subplot(2, 2, 4);
49 N_triangle = length(triangularSignal);
50 frequencies_triangle = linspace(0, Fs_triangle/2, N_triangle/2);
51 Y_triangle = fft(triangularSignal);
52 magnitude_triangle = 2/N_triangle * abs(Y_triangle(1:N_triangle/2));
53 plot(frequencies_triangle, magnitude_triangle);
54 title('طیف فرکانسی سیگنال مثلثی');
55 xlabel('فرکانس (هرتز)');
56 ylabel('مقدار طیف');
57

```



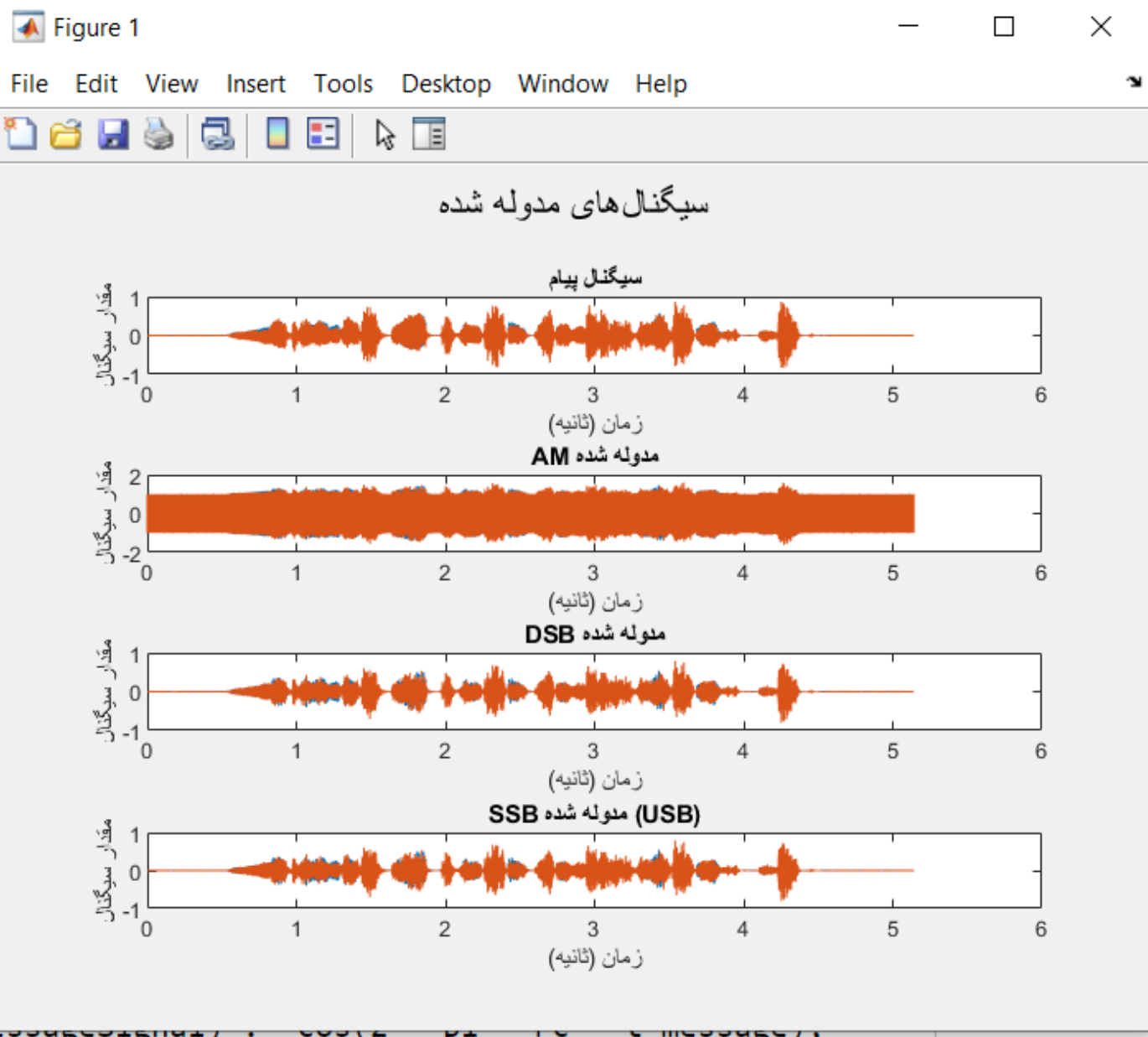
3- با پیاده سازی یکی از ساختارهای مدولاتور بیان شده در کلاس در محیط پایتون و یا MATLAB، سیگنالهای پیام بخش شماره 1 را به روشهای (AM  $\mu=0.85$ )، DSB و SSB مدوله کنید و شکل سیگنال مدوله شده را رسم نمایید.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 % پارامترها
8 Fc = 50e3; % فرکانس حامل (50KHz)
9 miu = 0.85; % ضریب مدولاسیون
10
11 % تولید سیگنال پیام
12 t_message = (0:length(audioData)-1) / Fs_audio ;
13 messageSignal = audioData';
14
15 % AM مدولاسیون
16 amSignal = (1 + miu * messageSignal) .* cos(2 * pi * Fc * t_message);
17 % DSB مدولاسیون
18 dsbSignal = messageSignal .* cos(2 * pi * Fc * t_message);
19
20 % SSB (USB) مدولاسیون
21 ssbSignal_usb = hilbert(messageSignal) .* exp(1i * 2 * pi * Fc * t_message);
22
23 % نمایش سیگنال ها
24 figure;
25
26 subplot(4, 1, 1);
27 plot(t_message, messageSignal);
28 title('سیگنال پیام');
29 xlabel('زمان (ثانیه)');
```

```
30 title('سیگنال پیام');
31 xlabel('زمان (ثانیه)');
32 ylabel('مقدار سیگنال');
33
34 subplot(4, 1, 2);
35 plot(t_message, amSignal);
36 title('AM مدوله شده');
37 xlabel('زمان (ثانیه)');
38 ylabel('مقدار سیگنال');
39
40 subplot(4, 1, 3);
41 plot(t_message, dsbSignal);
42 title('DSB مدوله شده');
43 xlabel('زمان (ثانیه)');
44 ylabel('مقدار سیگنال');
45
46 subplot(4, 1, 4);
47 plot(t_message, real(ssbSignal_usb)); % USB نمایش بخش حقیقی
48 title('SSB (USB) مدوله شده');
49 xlabel('زمان (ثانیه)');
50 ylabel('مقدار سیگنال');
51
52 sgtitle('سیگنال‌های مدوله شده');
53
```

---





3.1- در این بخش پس از مدوله کردن سیگنال صوت(am) با استفاده از دستورات لازم صوت سیگنال مدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 % پارامترها
8 Fc = 50e3; % فرکانس حامل (50KHz)
9 miu = 0.85; % ضریب مدولاسیون
10
11 % تولید سیگنال پیام
12 t_message = (0:length(audioData)-1) / Fs_audio ;
13 messageSignal = audioData';
14
15 % AM مدولاسیون
16 amSignal = (1 + miu * messageSignal) .* cos(2 * pi * Fc * t_message);
17 %AM انجام مدولاسیون
18 player=audioplayer(amSignal,Fs_audio);
19 play(player);
```

3.2- در این بخش پس از مدوله کردن سیگنال صوت (dsb) با استفاده از دستورات لازم صوت سیگنال مدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 % پارامترها
8 Fc = 50e3; % فرکانس حامل (50KHz)
9
10 % تولید سیگنال پیام
11 t_message = (0:length(audioData)-1) / Fs_audio ;
12 messageSignal = audioData';
13
14 % مدولاسیون DSB
15 dsbSignal = messageSignal .* cos(2 * pi * Fc * t_message);
16
17 %پخش کردن صدا بعد از انجام مدولاسیون dsb
18 player=audioplayer(dsbSignal,Fs_audio);
19 play(player);
```

3.3- در این بخش پس از مدوله کردن سیگنال صوت(ssbu) با استفاده از دستورات لازم صوت سیگنال مدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 % پارامترها
8 Fc = 50e3; % فرکانس حامل (50KHz)
9
10 % تولید سیگنال پیام
11 t_message = (0:length(audioData)-1) / Fs_audio ;
12 messageSignal = audioData';
13
14 % SSB (USB) مدولاسیون
15 ssbSignal_usb = hilbert(messageSignal) .* exp(1i * 2 * pi * Fc * t_message);
16
17 %ssbu پخش کردن صدا بعد از انجام مدولاسیون
18 player=audioplayer(ssbSignal_usb ,Fs_audio);
19 play(player);|
```

4- با پیاده سازی یکی از ساختارهای دمدوالتور بیان شده در کالس برای مدولاسیون مربوطه، سیگنال اطلاعات مدوله شده در بخش قبل را دمدوله کنید و با سیگنال اطلاعات اصلی مقایسه کنید.

```
1 clear
2
3 % نام فایل صوتی خود را وارد کنید
4 audioFilename = 'E:\mm\20240123_144908.m4a';
5
6 % خواندن فایل صوتی
7 [audioData, Fs_audio] = audioread(audioFilename);
8
9 % پارامترها
10 Fc = 50e3; % فرکانس حامل (50KHz)
11 miu = 0.85; % ضریب مدولاسیون
12
13 % تولید سیگنال پیام
14 t_message = (0:length(audioData)-1) / 1000 ;
15 messageSignal = audioData';
16
17 % AM مدولاسیون
18 amSignal = (1 + miu * messageSignal) .* cos(2 * pi * Fc * t_message);
19
20 % DSB مدولاسیون
21 dsbSignal = messageSignal .* cos(2 * pi * Fc * t_message);
22 |
23 % SSB (USB) مدولاسیون
24 ssbSignal_usb = hilbert(messageSignal) .* exp(1i * 2 * pi * Fc * t_message);
25
26 % AM دمدولاسیون
27 demodulatedAM = abs(amSignal) - 1;
28
29 % DSB دمدولاسیون
```

```
30 demodulatedDSB = dsbSignal .* cos(2 * pi * Fc * t_message);
31
32 % دمدولاسیون SSB (USB)
33 demodulatedSSB_usb = real(ssbSignal_usb);
34
35 % نمایش سیگنال‌های دمدوله شده
36 figure;
37 subplot(3, 1, 1);
38 plot(t_message, demodulatedAM);
39 title('AM دمدوله شده');
40 xlabel('زمان (ثانیه)');
41 ylabel('مقدار سیگنال');
42
43 subplot(3, 1, 2);
44 plot(t_message, demodulatedDSB);
45 title('DSB دمدوله شده');
46 xlabel('زمان (ثانیه)');
47 ylabel('مقدار سیگنال');
48
49 subplot(3, 1, 3);
50 plot(t_message, demodulatedSSB_usb);
51 title('SSB (USB) دمدوله شده');
52 xlabel('زمان (ثانیه)');
53 ylabel('مقدار سیگنال');
54
55 sgtitle('سیگنال‌های دمدوله شده');
56
57 % مقایسه سیگنال اصلی و دمدوله شده
58 figure;
```

```
59
60 subplot(3, 1, 1);
61 plot(t_message, messageSignal);
62 title('سیگنال پیام اصلی');
63 xlabel('زمان (ثانیه)');
64 ylabel('مقدار سیگنال');
65
66 subplot(3, 1, 2);
67 plot(t_message, demodulatedAM);
68 title('AM دمدوله شده');
69 xlabel('زمان (ثانیه)');
70 ylabel('مقدار سیگنال');
71
72 subplot(3, 1, 3);
73 plot(t_message, demodulatedDSB);
74 title('DSB دمدوله شده');
75 xlabel('زمان (ثانیه)');
76 ylabel('مقدار سیگنال');
77 sgtitle('مقایسه سیگنال اصلی و دمدوله شده');
```

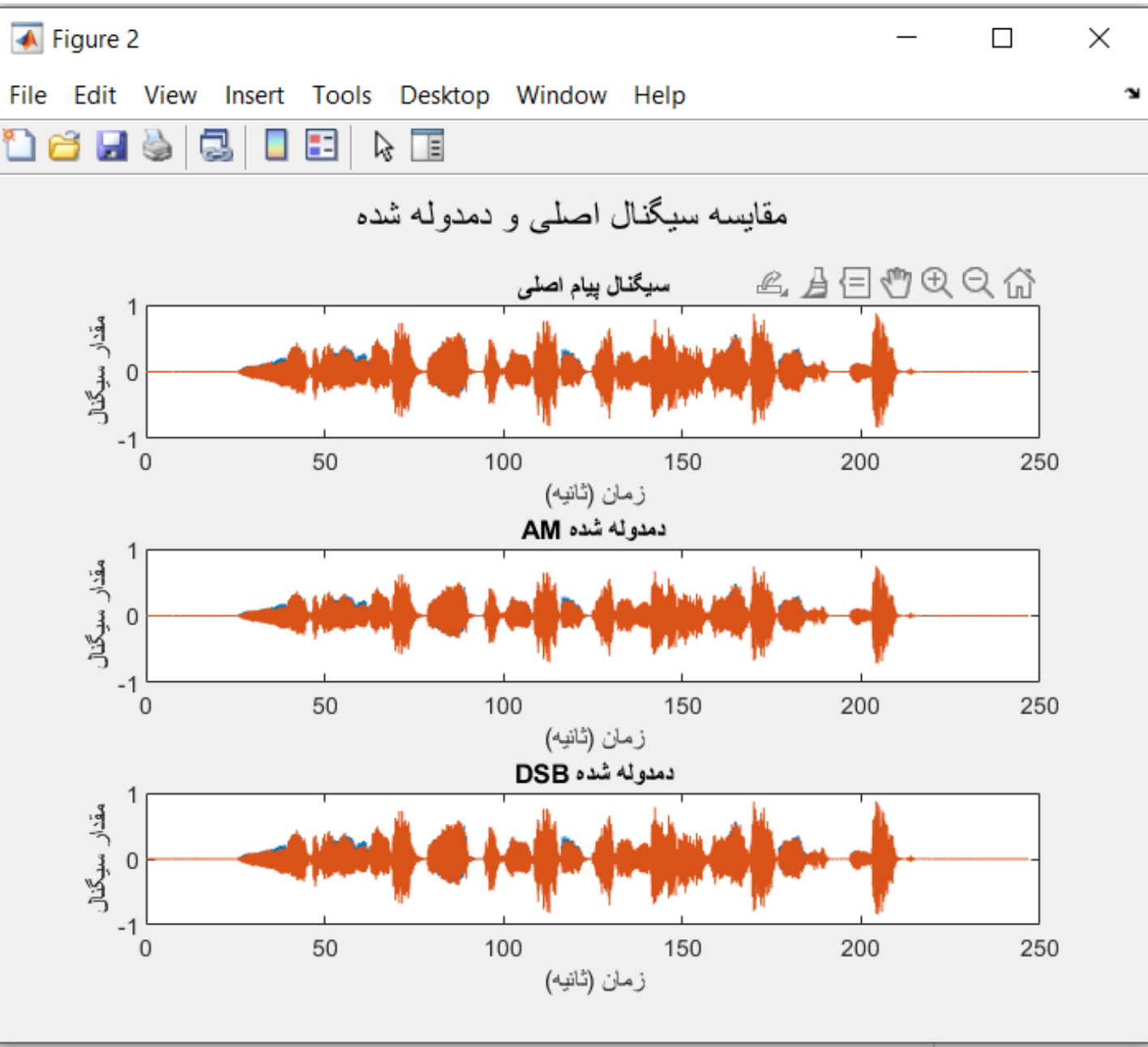




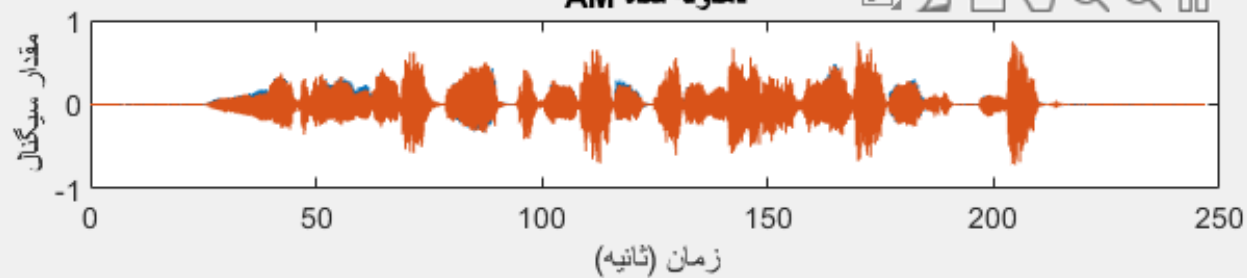
Figure 1

File Edit View Insert Tools Desktop Window Help

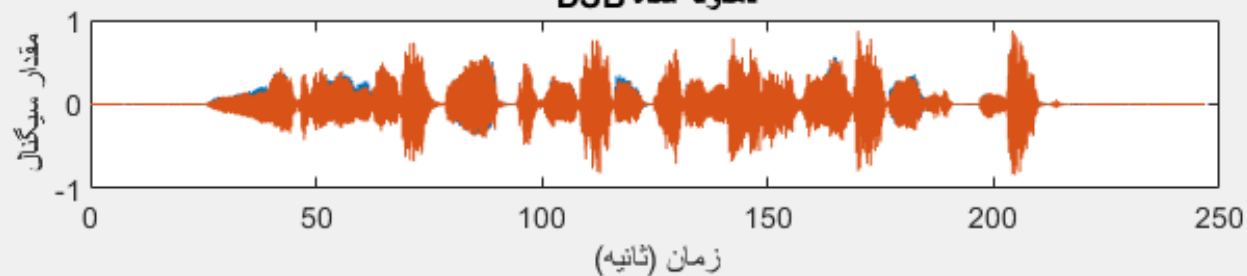


## سیگنال‌های دمدوله شده

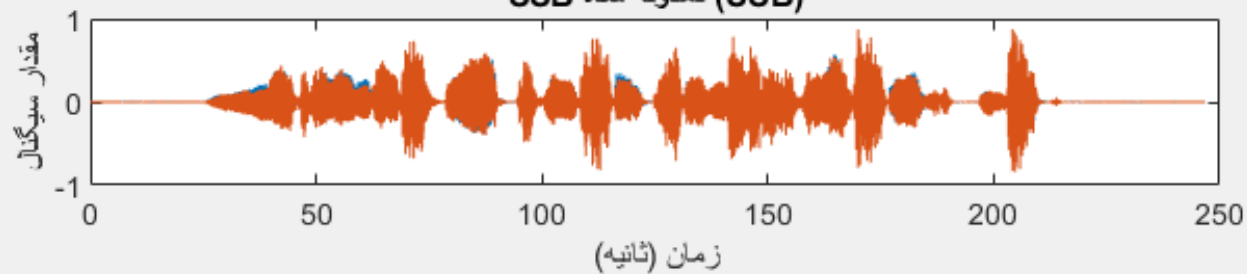
دمدوله شده AM



دمدوله شده DSB



دمدوله شده SSB (USB)



4.1- در این بخش پس از دمدوله کردن سیگنال صوت(am) با استفاده از دستورات لازم صوت سیگنال دمدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 % پارامترها
8 Fc = 50e3; % فرکانس حامل (50KHz)
9 miu = 0.85; % ضریب مدولاسیون
10
11 % تولید سیگنال پیام
12 t_message = (0:length(audioData)-1) / Fs_audio ;
13 messageSignal = audioData';
14
15 % AM مدولاسیون
16 amSignal = (1 + miu * messageSignal) .* cos(2 * pi * Fc * t_message);
17
18 % AM دمدولاسیون
19 demodulatedAM = abs(amSignal) - 1;
20
21 % پخش صوت پس از دمدولاسیون AM
22 player=audioplayer(demodulatedAM,Fs_audio);
23 play(player);
```

4.2- در این بخش پس از دمدوله کردن سیگنال صوت (dsb) با استفاده از دستورات لازم صوت سیگنال دمدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 % پارامترها
8 Fc = 50e3; % فرکانس حامل (50KHz)
9
10 % تولید سیگنال پیام
11 t_message = (0:length(audioData)-1) / Fs_audio ;
12 messageSignal = audioData';
13
14 % DSB مدولاسیون
15 dsbSignal = messageSignal .* cos(2 * pi * Fc * t_message);
16
17 % DSB دمدولاسیون
18 demodulatedDSB = dsbSignal .* cos(2 * pi * Fc * t_message);
19
20 % پخش کردن صدا بعد از انجام دمدولاسیون dsb
21 player=audioplayer(demodulatedDSB,Fs_audio);
22 play(player);|
```

4.3- در این بخش پس از دمدوله کردن سیگنال صوت(ssbu) با استفاده از دستورات لازم صوت سیگنال دمدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 % پارامترها
8 Fc = 50e3; % فرکانس حامل (50KHz)
9
10 % تولید سیگنال پیام
11 t_message = (0:length(audioData)-1) / Fs_audio ;
12 messageSignal = audioData';
13
14 % SSB (USB) مدولاسیون
15 ssbSignal_usb = hilbert(messageSignal) .* exp(1i * 2 * pi * Fc * t_message);
16
17 % SSB (USB) دمدولاسیون
18 demodulatedSSB_usb = real(ssbSignal_usb);
19
20 % پخش کردن صدا بعد از انجام دمدولاسیون ssb
21 player=audioplayer(demodulatedSSB_usb ,Fs_audio);
22 play(player);
```

5- انجام مدولاسیون fm برای بخش های قبل با  $f = 2\text{KHz}$  و  $f_c = 50\text{KHz}$

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 t=(0:length(audioData)-1)/Fs_audio;
8 % تنظیمات
9 delta_f = 2e3; % (حداکثر تغییر فرکانس)
10 Fc = 50e3; % فرکانس حامل
11
12 % مدولاسیون FM
13 fmSignal = zeros(size(audioData));
14 phase = 0;
15
16 for i = 1:length(audioData)
17     delta_phi = 2 * pi * delta_f * audioData(i) / Fs_audio;% محاسبه fs,delta_f تغییر فاز به نسبت مقدار نقطه متناظر در audiodata
18     phase = phase + delta_phi;% تجمع فازهای تغییر یافته
19     fmSignal(i) = cos(2 * pi * Fc * t(i) + phase);% با افزودن کسینوس سیگنال مدوله شده اف ام ساخته میشود
20 end
21
22 % FM نمایش سیگنال مدوله شده
23 figure;
24 subplot(2, 1, 1);
25 plot(t, fmSignal);
26 title('مدوله شده FM');
27 xlabel('زمان (ثانیه)');
28 ylabel('مقدار سیگنال');
```

```

30 % دمدولاسیون FM
31 demodulatedFM = zeros(size(fmSignal));
32 phase = 0;
33
34 for i = 2:length(fmSignal)
35     % شروع می‌کنیم تا از اندیس 1 استفاده نکنیم i=2 از
36     % تا از اندیس‌های قبلی در محاسبات استفاده نشود
37     delta_phi = angle(fmSignal(i)) - angle(fmSignal(i-1)); % محاسبه فاز یا زاویه خود سیگنال با سیگنال قبلیش
38     demodulatedFM(i) = delta_phi / (2 * pi * t(i));
39 end
40
41 % FM نمایش سیگنال دمدوله شده
42 subplot(2, 1, 2);
43 plot(t, demodulatedFM);
44 title('دمدوله شده FM');
45 xlabel('زمان (ثانیه)');
46 ylabel('مقدار سیگنال');
47
48 % FM نمایش مقایسه سیگنال اصلی و دمدوله شده
49 figure;
50 subplot(2, 1, 1);
51 plot(t, audioData, 'g', t, demodulatedFM, 'r--');
52 title('دمدوله شده FM مقایسه سیگنال اصلی و');
53 legend('سیگنال اصلی', 'FM شده');
54 xlabel('زمان (ثانیه)');
55 ylabel('مقدار سیگنال');
56

```

Figure 2

File Edit View Insert Tools Desktop Window Help

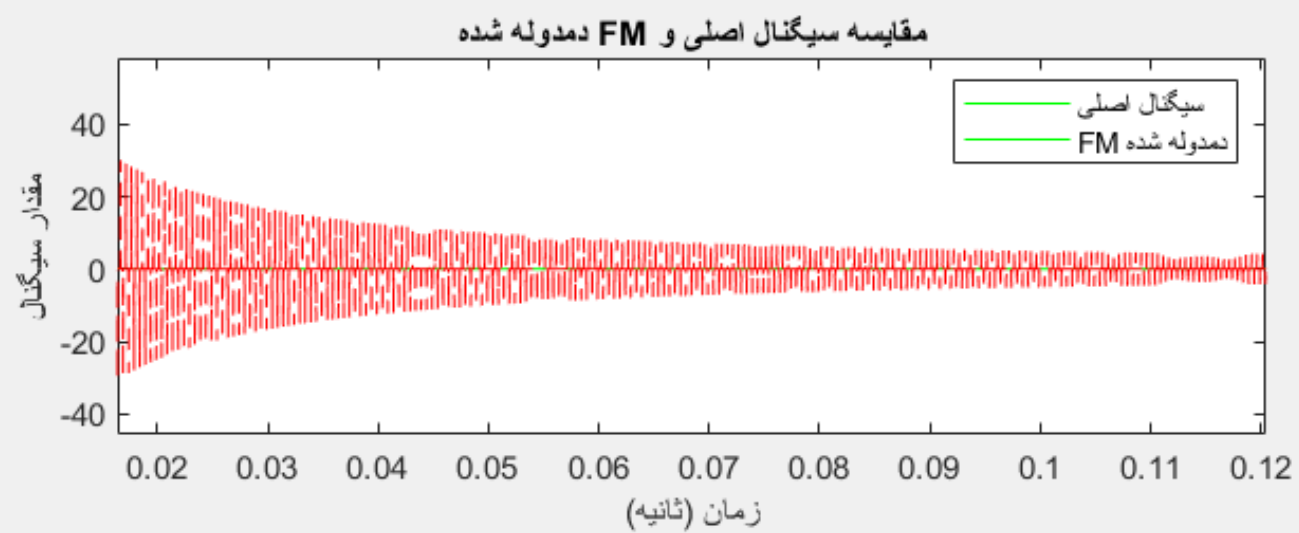
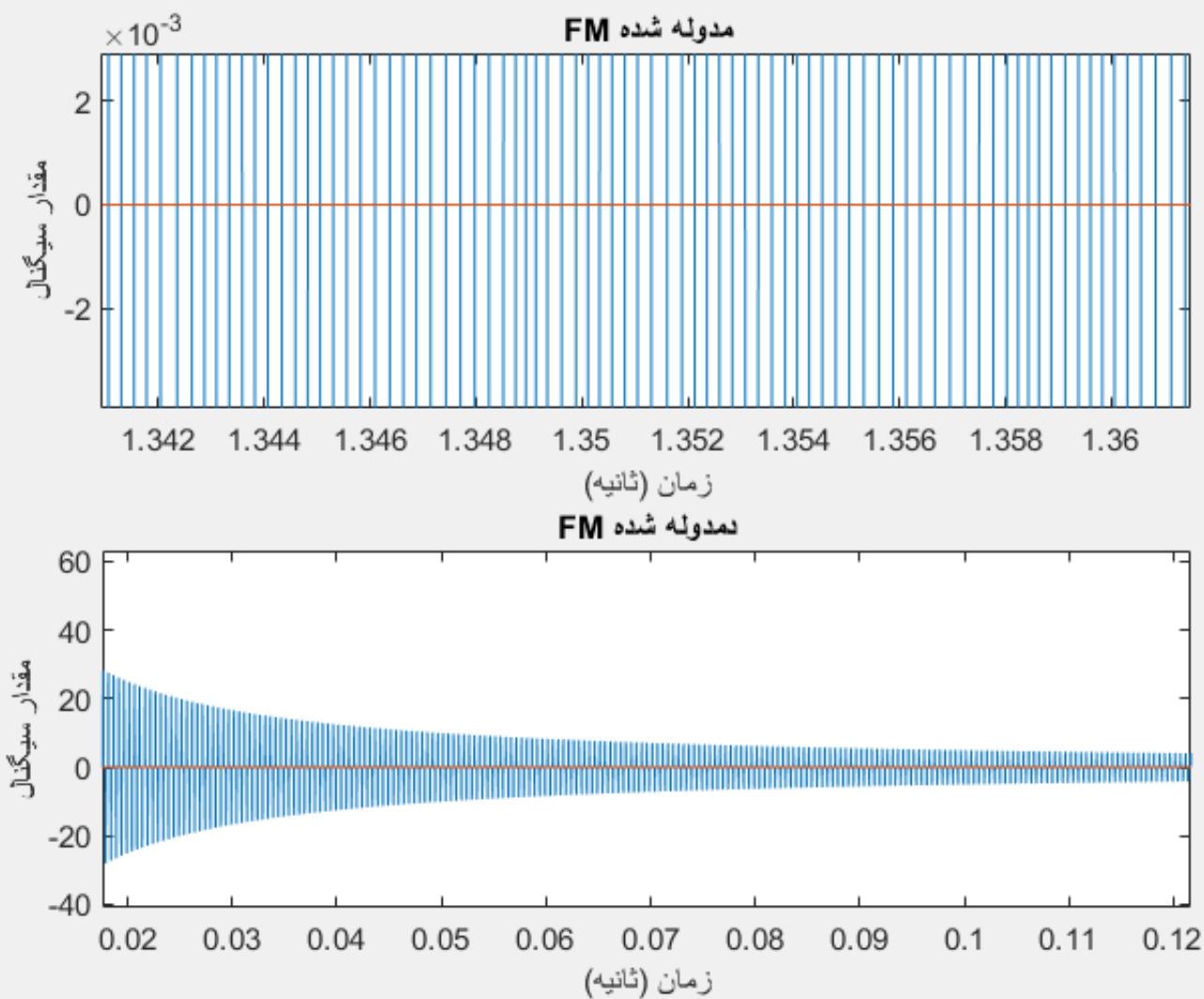


Figure 1

File Edit View Insert Tools Desktop Window Help





5.1- در این بخش پس از مدوله کردن سیگنال صوت (FM) با استفاده از دستورات لازم صوت سیگنال مدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 t=(0:length(audioData)-1)/Fs_audio;
8 % تنظیمات
9 delta_f = 2e3; % (حداکثر تغییر فرکانس)
10 Fc = 50e3; % فرکانس حامل
11
12 % مدولاسیون FM
13 fmSignal = zeros(size(audioData));
14 phase = 0;
15
16 for i = 1:length(audioData)
17     delta_phi = 2 * pi * delta_f * audioData(i) / Fs_audio;% محاسبه مقدار تغییر فاز در fs,delta_f
18     phase = phase + delta_phi;% تجمیع فازهای تغییر یافته
19     fmSignal(i) = cos(2 * pi * Fc * t(i) + phase);% با افزودن کسینوس سیگنال مدوله شده اف ام ساخته میشود
20 end
21
22 %پخش کردن صدا بعد از انجام مدولاسیون FM
23 player=audioplayer(fmSignal,Fs_audio);
24 play(player);
```

5.2- در این بخش پس از دمدوله کردن سیگنال صوت (FM) با استفاده از دستورات لازم صوت سیگنال دمدوله شده پخش می شود.

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 t=(0:length(audioData)-1)/Fs_audio;
8 % تنظیمات
9 delta_f = 2e3; % (حداکثر تغییر فرکانس)
10 Fc = 50e3; % فرکانس حامل
11
12 % FM مدولاسیون
13 fmSignal = zeros(size(audioData));
14 phase = 0;
15
16 for i = 1:length(audioData)
17     delta_phi = 2 * pi * delta_f * audioData(i) / Fs_audio;% محاسبه مقدار تغییر فاز در fs,delta_f
18     phase = phase + delta_phi;% جمع فازهای تغییر یافته
19     fmSignal(i) = cos(2 * pi * Fc * t(i) + phase);% با افزودن کسینوس سیگنال مدوله شده اف ام ساخته میشود
20 end
21
22 demodulatedFM = zeros(size(fmSignal));
23 phase = 0;
24
25 for i = 2:length(fmSignal)
26     % شروع می‌کنیم تا از اندیس 1 استفاده نکنیم i=2 از
27     % تا از اندیس‌های قبلی در محاسبات استفاده نشود
28     delta_phi = angle(fmSignal(i)) - angle(fmSignal(i-1));% محاسبه فاز یا زاویه خود سیگنال یا سیگنال قبلیش
29     demodulatedFM(i) = delta_phi / (2 * pi * t(i));
30 end
31
32 %FM مدولاسیون انجام
33 player=audioplayer(demodulatedFM,Fs_audio);
34 play(player);
```

6- فرض کنید که سیگنال مدوله شده از یک کانال با اعوجاج غیر خطی ( $y=x^3+0.5x^2$ ) عبور می‌کنند. نشان دهید که مدولاسیون FM نسبت به مدولاسیونهای AM و DSB در مقابل این اعوجاج غیر خطی مصونیت بیشتری دارند

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 miu = 0.85; % ضریب مدولاسیون
8 Fc = 50e3; % فرکانس حامل
9
10 t=(0:length(audioData)-1)/Fs_audio;
11 audioSignal = audioData';
12 % تابع اعوجاج غیرخطی
13 nonlinearChannel = @(x) x.^3 + 0.5*x.^2;
14
15 % AM مدولاسیون
16 amSignal =(1+miu*audioSignal) .* cos(2*pi*Fc*t);
17
18 % DSB مدولاسیون
19 dsbSignal = audioSignal .* cos(2*pi*Fc*t);
20
21 % FM مدولاسیون
22 delta_f = 20; % (حداکثر تغییر فرکانس)
23 fmSignal = zeros(size(audioSignal));
24 phase = 0;
25
26 for i = 1:length(audioSignal)
27     delta_phi = 2 * pi * delta_f * audioSignal(i) / Fs_audio;
28     phase = phase + delta_phi;
29     fmSignal(i) = cos(2 * pi * Fc * t(i) + phase);
30 end
31
32 % اعمال اعوجاج غیرخطی بر روی سیگنال‌ها
33 amSignalDistorted = nonlinearChannel(amSignal);
34 dsbSignalDistorted = nonlinearChannel(dsbSignal);
35 fmSignalDistorted = nonlinearChannel(fmSignal);
36
37 % نمایش سیگنال‌های مدوله شده
38 figure;
39 subplot(3, 1, 1);
```

```

40 plot(t, `amSignal`);
41 title('مدوله شده AM');
42
43 subplot(3, 1, 2);
44 plot(t, dsbSignal);
45 title('مدوله شده DSB');
46
47 subplot(3, 1, 3);
48 plot(t, fmSignal);
49 title('مدوله شده FM');
50
51 % نمایش سیگنال‌های مدوله شده و اعمال اعوجاج
52 figure;
53 subplot(3, 2, 1);
54 plot(t, amSignalDistorted);
55 title('مدوله شده و اعوجاج AM');
56
57 subplot(3, 2, 3);
58 plot(t, dsbSignalDistorted);
59 title('مدوله شده و اعوجاج DSB');
60
61 subplot(3, 2, 5);
62 plot(t, fmSignalDistorted);
63 title('مدوله شده و اعوجاج FM');
64
65 % دمدولاسیون
66 amDemodulated = abs(amSignalDistorted) - 1;
67 dsbDemodulated = dsbSignalDistorted .* cos(2*pi*Fc*t);
68 fmDemodulated = zeros(size(fmSignalDistorted));
69 phase = 0;
70
71 for i = 2:length(fmSignalDistorted)
72     delta_phi = angle(fmSignalDistorted(i)) - angle(fmSignalDistorted(i-1));
73     fmDemodulated(i) = delta_phi / (2 * pi * t(i));
74 end
75
76 % نمایش سیگنال‌های دمدوله شده
77 subplot(3, 2, 2);

```

```

78 plot(t, amDemodulated);
79 title('AM دمدوله شده');
80
81 subplot(3, 2, 4);
82 plot(t, dsbDemodulated);
83 title('DSB دمدوله شده');
84
85 subplot(3, 2, 6);
86 plot(t, fmDemodulated);
87 title('FM دمدوله شده');
88
89 % مقایسه سیگنال اصلی و دمدوله شده
90 figure;
91 subplot(3, 1, 1);
92 plot(t, audioSignal, 'g', t, amDemodulated, 'r--');
93 title('مقایسه سیگنال اصلی و دمدوله شده AM');
94 legend('سیگنال اصلی', 'AM دمدوله شده');
95
96 subplot(3, 1, 2);
97 plot(t, audioSignal, 'g', t, dsbDemodulated, 'r--');
98 title('مقایسه سیگنال اصلی و دمدوله شده DSB');
99 legend('سیگنال اصلی', 'DSB دمدوله شده');
100
101 subplot(3, 1, 3);
102 plot(t, audioSignal, 'g', t, fmDemodulated, 'r--');
103 title('مقایسه سیگنال اصلی و دمدوله شده FM');
104 legend('سیگنال اصلی', 'FM دمدوله شده');
105

```

Figure 1

File Edit View Insert Tools Desktop Window Help

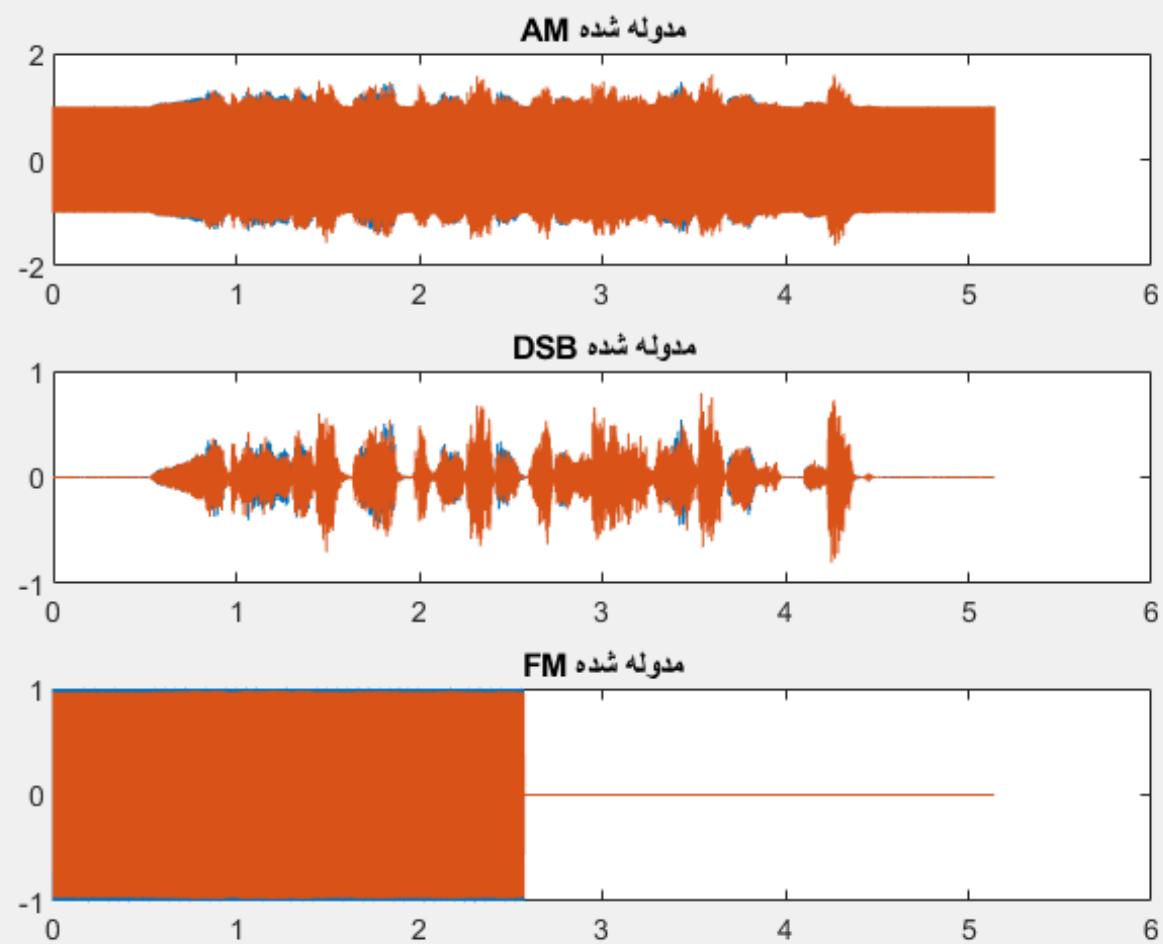


Figure 2

File Edit View Insert Tools Desktop Window Help

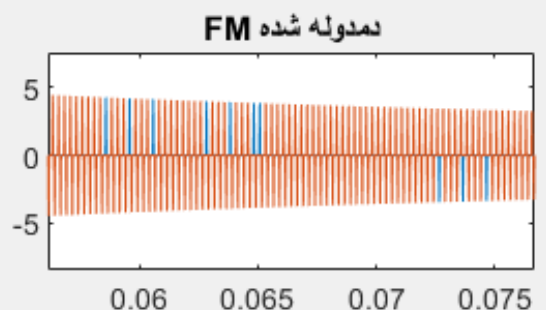
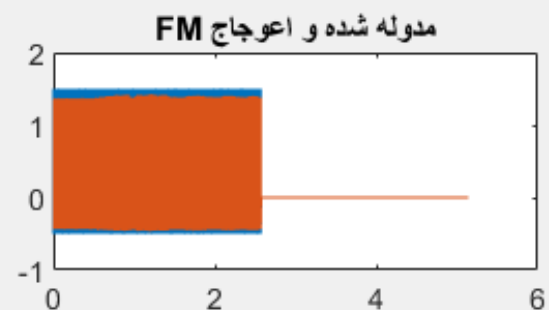
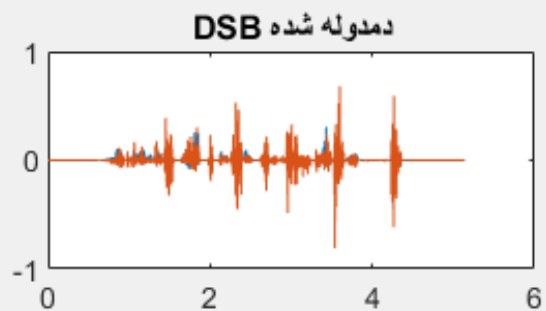
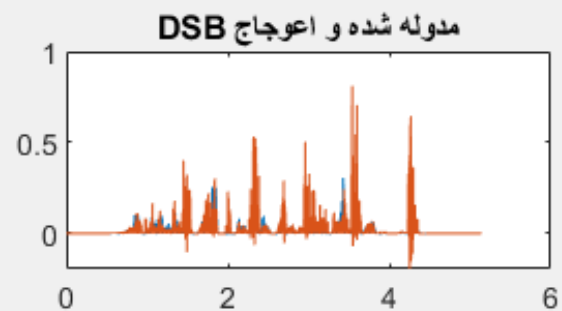
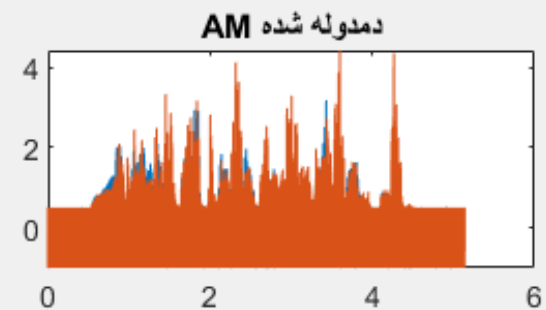
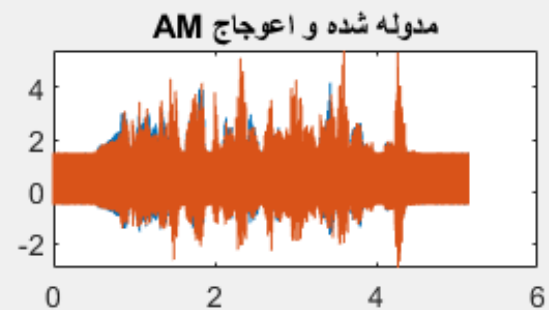
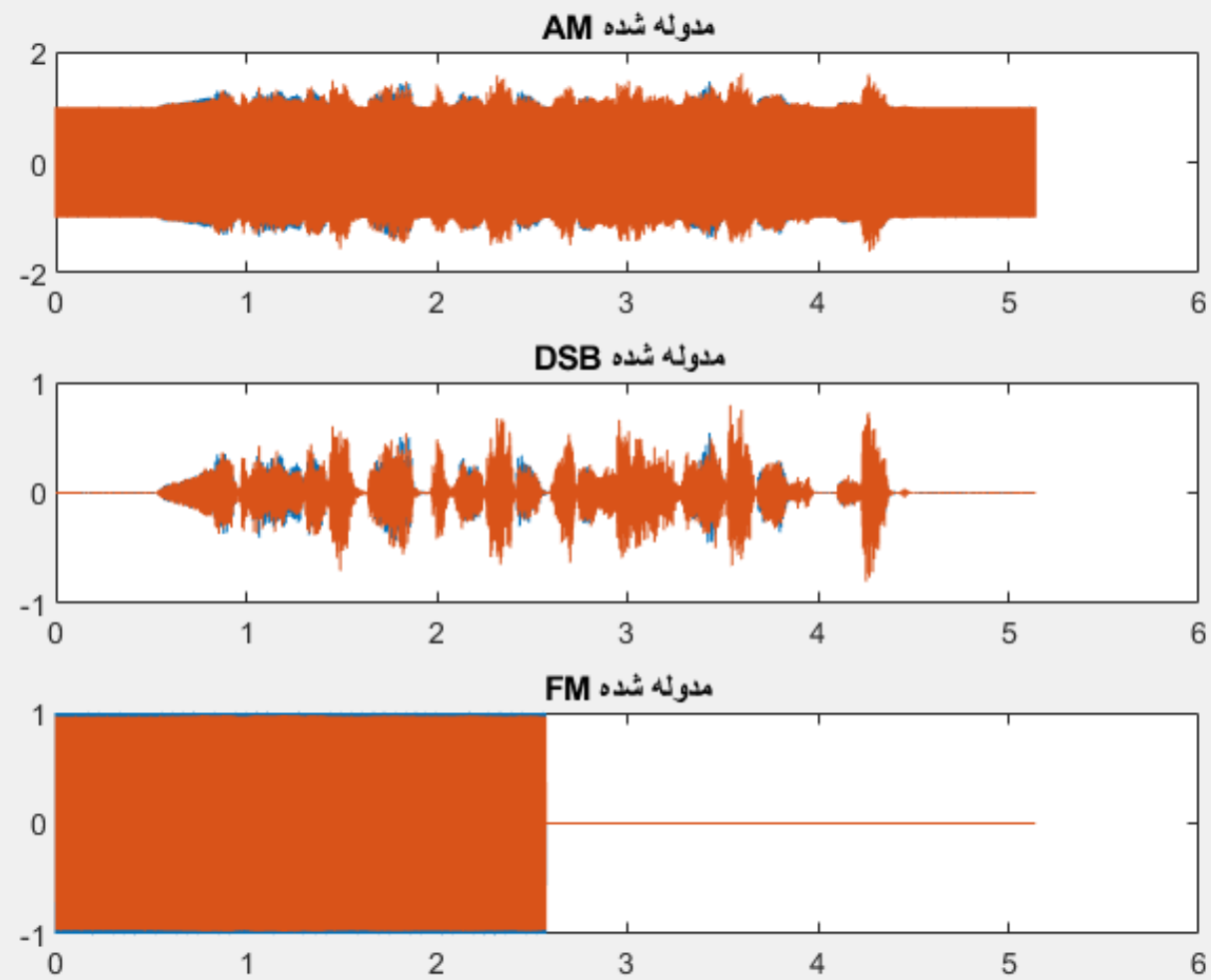


Figure 1

File Edit View Insert Tools Desktop Window Help





## 6.1- صوت نهایی بعد از انجام دمدولاسیون am روی سیگنال صوت مدوله شده (am) پس از اعوجاج

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7 miu = 0.85; % ضریب مدولاسیون
8 Fc = 50e3; % فرکانس حامل
9
10 t=(0:length(audioData)-1)/Fs_audio;
11 audioSignal = audioData';
12 % تابع اعوجاج غیرخطی
13 nonlinearChannel = @(x) x.^3 + 0.5*x.^2;
14
15 % AM مدولاسیون
16 amSignal =(1+miu*audioSignal) .* cos(2*pi*Fc*t);
17
18 % اعمال اعوجاج غیرخطی بر روی سیگنال‌ها
19 amSignalDistorted = nonlinearChannel(amSignal);
20
21 % دمدولاسیون
22 amDemodulated = abs(amSignalDistorted) - 1;
23
24 % پخش صوت پس از دمدولاسیون AM
25 player=audioplayer(amDemodulated,Fs_audio);
26 play(player);
```

## 6.2- صوت نهایی بعد از انجام دمدولاسیون dsb روی سیگنال صوت مدوله شده (dsb) پس از اعوجاج

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7
8 Fc = 50e3; % فرکانس حامل
9 |
10 t=(0:length(audioData)-1)/Fs_audio;
11 audioSignal = audioData';
12 % تابع اعوجاج غیرخطی
13 nonlinearChannel = @(x) x.^3 + 0.5*x.^2;
14
15 % DSB مدولاسیون
16 dsbSignal = audioSignal .* cos(2*pi*Fc*t);
17
18 % اعمال اعوجاج غیرخطی بر روی سیگنال‌ها
19 dsbSignalDistorted = nonlinearChannel(dsbSignal);
20 % DSB دمدولاسیون سیگنال اعوجج یافته
21 dsbDemodulated = dsbSignalDistorted .* cos(2*pi*Fc*t);
22
23 % پخش کردن صدا بعد از انجام دمدولاسیون DSB
24 player=audioplayer(dsbDemodulated,Fs_audio);
25 play(player);
```

### 6.3- صوت نهایی بعد از انجام دمدولاسیون fm روی سیگنال صوت مدوله شده (fm) پس از اعوجاج

```
1 clear
2 % نام فایل صوتی خود را وارد کنید
3 audioFilename = 'E:\mm\20240123_144908.m4a';
4
5 % خواندن فایل صوتی
6 [audioData, Fs_audio] = audioread(audioFilename);
7
8 Fc = 50e3; % فرکانس حامل
9
10 t=(0:length(audioData)-1)/Fs_audio;
11 audioSignal = audioData';
12 % تابع اعوجاج غیرخطی
13 nonlinearChannel = @(x) x.^3 + 0.5*x.^2;
14
15 % FM مدولاسیون
16 delta_f = 20; % (حداکثر تغییر فرکانس)
17 fmSignal = zeros(size(audioSignal));
18 phase = 0;
19
20 for i = 1:length(audioSignal)
21     delta_phi = 2 * pi * delta_f * audioSignal(i) / Fs_audio;
22     phase = phase + delta_phi;
23     fmSignal(i) = cos(2 * pi * Fc * t(i) + phase);
24 end
25
26 % اعمال اعوجاج غیرخطی بر روی سیگنال‌ها
27 fmSignalDistorted = nonlinearChannel(fmSignal);
28
29 fmDemodulated = zeros(size(fmSignalDistorted));
30 phase = 0;
31
32 for i = 2:length(fmSignalDistorted)
33     delta_phi = angle(fmSignalDistorted(i)) - angle(fmSignalDistorted(i-1));
34     fmDemodulated(i) = delta_phi / (2 * pi * t(i));
35 end
36 %FM کردن صدا بعد از انجام دمدولاسیون
37 player=audioplayer(fmDemodulated,Fs_audio);
38 play(player);
```

## 7- اثر نویز جمع شونده در مدولاسیون

```
1 % Define the parameters
2 filename = 'E:\mm\20240123_144908.m4a'; % Replace with the path to your audio file
3 [x, fs] = audioread(filename); % Read the audio file
4
5 % Define the noise power
6 P_noise = 10^(-5); % Noise power in W
7
8 % Generate the noise
9 n = sqrt(P_noise) * randn(size(x)); % Noise samples with specified power
10
11 % Add the noise to the audio signal
12 y = x + n;
13
14 % Calculate the SNR
15 SNR = snr(x, n);
16
17 % Calculate the spectrum of the original signal
18 X = fft(x);
19
20 % Calculate the spectrum of the signal with noise
21 Y = fft(y);
22
23 % Plot the spectrum of the original signal
24 figure;
25 frequencies = linspace(0, fs, length(X));
26 plot(frequencies, abs(X));
27 title('طیف فرکانسی سیگنال پیام بدون نویز');
28 xlabel('فرکانس (هرتز)');
29
30 % Plot the spectrum of the signal with noise
31 figure;
32 plot(frequencies, abs(Y));
33 title('طیف فرکانسی سیگنال پیام با نویز');
34 xlabel('فرکانس (هرتز)');
35
36 % Display the SNR
37 disp(['SNR: ' num2str(SNR) ' dB']);
38
```

