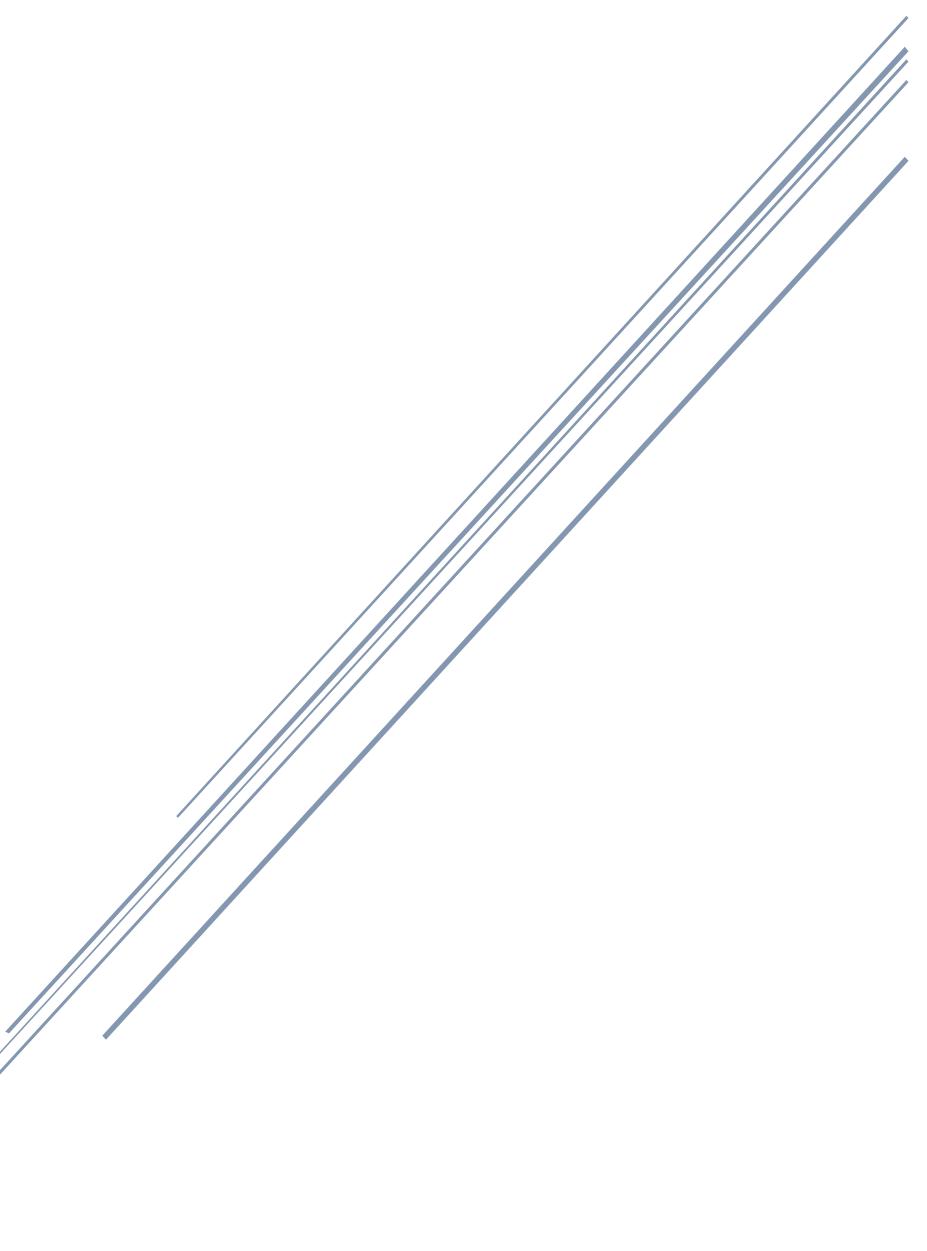


پروژه اصول سیستم های مخابراتی

ترم ۱۴۰۲-۲



سید علیرضا میرعبدیینی

۱۴۰۲۱۴۱۴۰

فهرست مطالب

۱. برسی اثر نویز، شیفت فاز و فیلتر بر سیگنال صوتی ۲
۲. طیف سیگنال صوتی(تکه ای از آواز غزل رومی) ۳
۳. نویز سفید ب.
۴. جز تربیعی نویز ت.
۵. ترکیب نویز و شیفت فاز ث.
۶. فیلتر قبل از جبران فاز ح.
۷. فیلتر پس از جبران فاز خ.
۸. بخش دوم - سیمولینک ۲
۹. مدولاتور AM (Am.slx) ۸
۱۰. دتکتور AM (AmDet.slx) ب.

۱. بررسی اثر نویز، شیفت فاز و فیلتر بر سیگنال صوتی

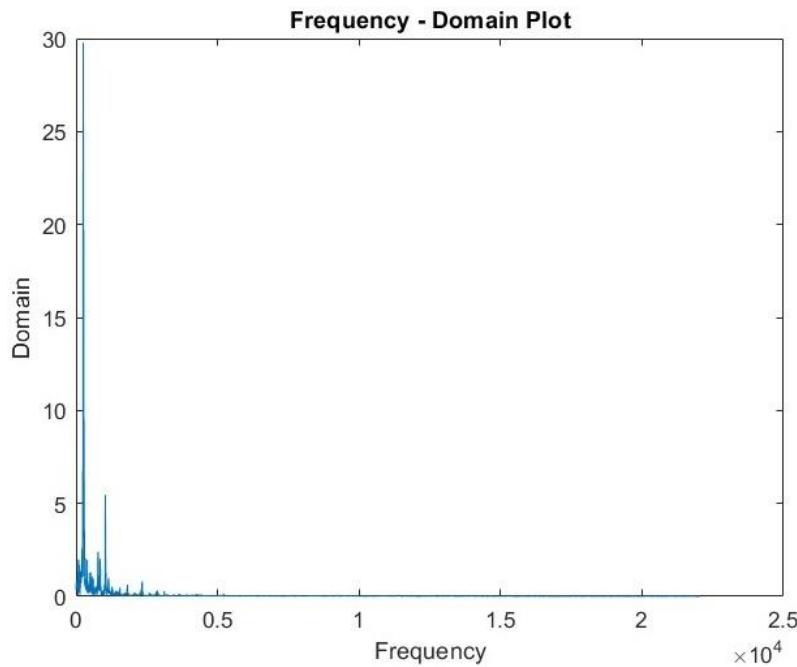
أ. طیف سیگنال صوتی(تکه ای از آواز غزل رومی)

ابتدا فایل صوتی را به عنوان ورودی درون استراکت `message` ذخیره کرده سپس تبدیل فوریه داده صوتی را گرفته و اندازه آن را بر حسب فرکانس رسم میکنیم.

```
close all;
clc;
clear;

message = importdata("Avaz on Rumi Sonnet.mp3");
F = linspace(0,message.fs,nfft);
msg_ft_data = fft(message.data,nfft);
msg_absft_data = abs(msg_ft_data);

figure;
plot(F(1:nfft/2), msg_absft_data(1:nfft/2));
xlabel("Frequency");
ylabel("Domain");
title("Frequency - Domain Plot")
```



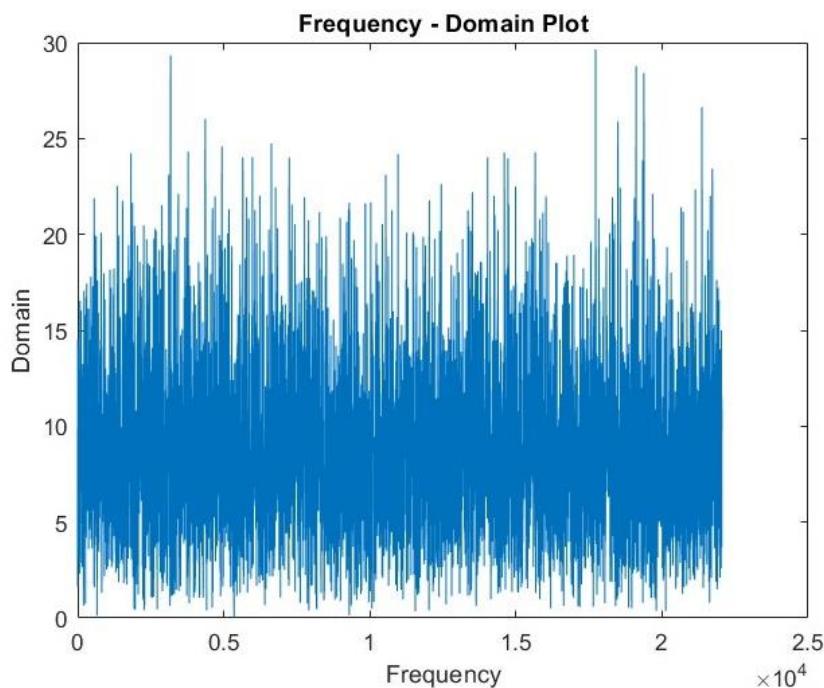
شکل ۱: نمودار طیف سیگنال ورودی

با توجه ب خروجی، صوت باند پایه است.

ب. نویز سفید

با استفاده از `wgn` تابع گوسی با مشخصات داده شده درست کرده و تبدیل فوریه آن را گرفته و اندازه آن را حساب کرده و طیف آن را رسم میکنیم.

```
white_noise = wgn(length(message.data), 2, -20);
white_noise_ft = fft(white_noise, nfft);
white_noise_ftabs = abs(white_noise_ft);
figure;
plot(F(1:nfft/2), white_noise_ftabs(1:nfft/2));
xlabel("Frequency");
ylabel("Domain");
title("Frequency - Domain Plot")
```



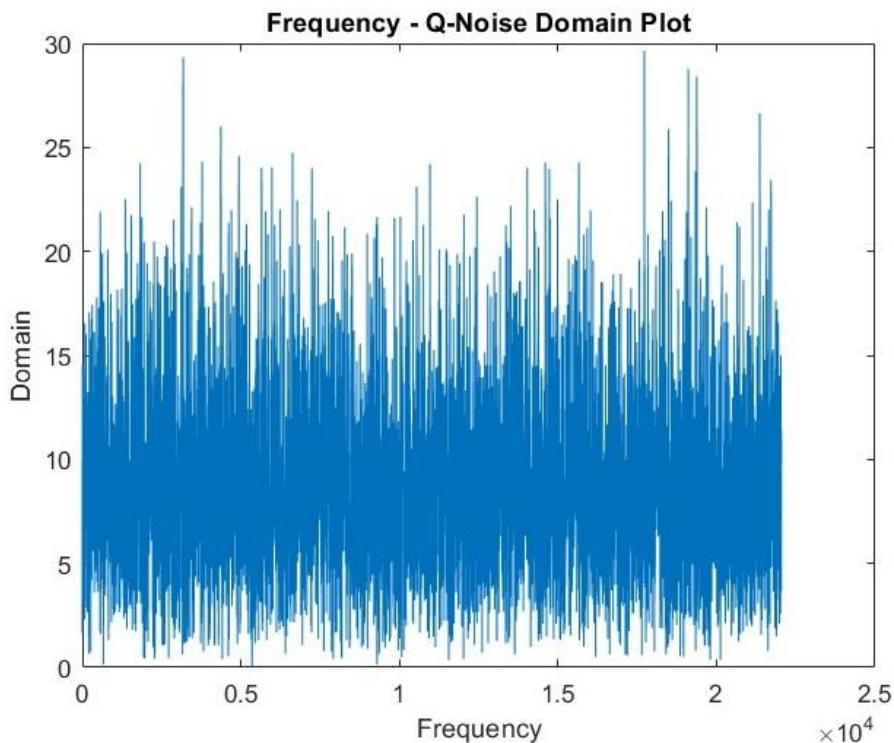
شکل ۲: طیف نویز سفید

با توجه به نمودار خروجی میبینیم که همه فرکانس ها را داراست.

ت. جز تربیعی نویز

با استفاده از تبدیل هیلبرت و آن که بخش موهومی آن همان بخش تربیعی نویز می‌شود، اسکریپت زیر را می‌نویسیم.

```
white_noise_Qd = imag(hilbert(white_noise,nfft));
figure;
plot(F(1:nfft/2),white_noise_Qd(1:nfft/2));
plot(F(1:nfft/2), white_noise_ftabs(1:nfft/2));
xlabel("Frequency");
ylabel("Domain");
title("Frequency - Q-Noise Domain Plot")
```



شکل ۳: نمودار طیف بخش تربیعی نویز

ث. ترکیب نویز و شیفت فاز

به طبع صدای نویز سفید قابل حس هست اما شیفت اما نمی‌شود.
صدای شیفت بدون نویز را هم با هم گوش دادم ولی تغییری حس نمی‌شد
با این حال با کنار هم قراردادن شان، تغییر کاملاً دیده می‌شود)

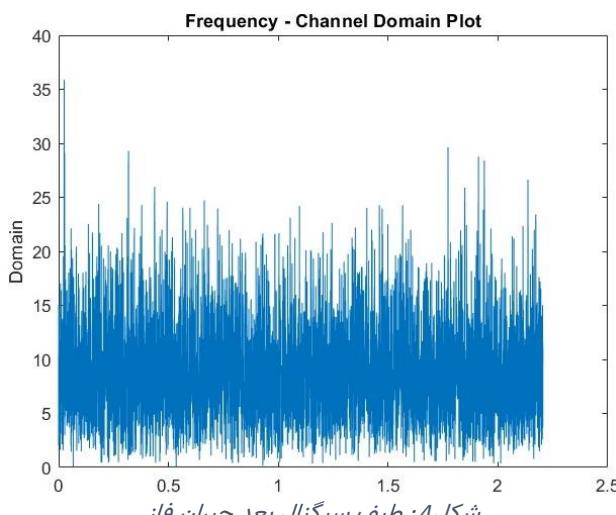
```
signal_ft_shifted = msg_ft_data .* exp(4i*pi/9) ;
signal_ft_shifted_WithNoise = signal_ft_shifted + white_noise_ft;
signal_ft_WithNoise = msg_ft_data + white_noise_ft;
signal_shifted = real(ifft(signal_ft_WithNoise));
audiowrite('shifted_audio_with_noise.wav', signal_shifted, message.fs);
فایل های
```

دققت شود shifted_audio.wav / shifted_audio_with_noise.wav / Compare_1&2

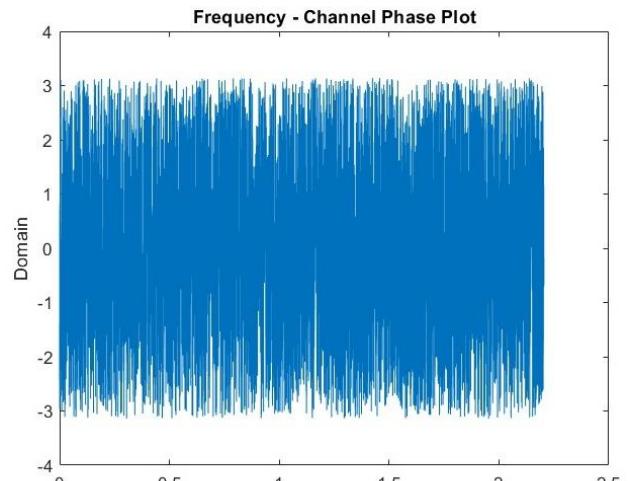
ج. جبران فاز

با استفاده از اسکریپت زیر انجام می‌گیرید.

```
channel_ft = signal_ft_WithNoise .* exp(-4i*pi/9) ;
channel_phase = angle(channel_ft);
channel_abs = abs(channel_ft);
figure;
plot(F(1:nfft/2), channel_abs(1:nfft/2));
xlabel("Frequency");
ylabel("Domain");
title("Frequency - Channel Domain Plot");
figure;
plot(F(1:nfft/2), channel_phase(1:nfft/2));
xlabel("Frequency");
ylabel("Domain");
title("Frequency - Channel Phase Plot");
audiowrite('Channel.wav', real(ifft(channel_ft)), message.fs);
```



شكل 4: طیف فاز سیگنال بعد از جبران فاز

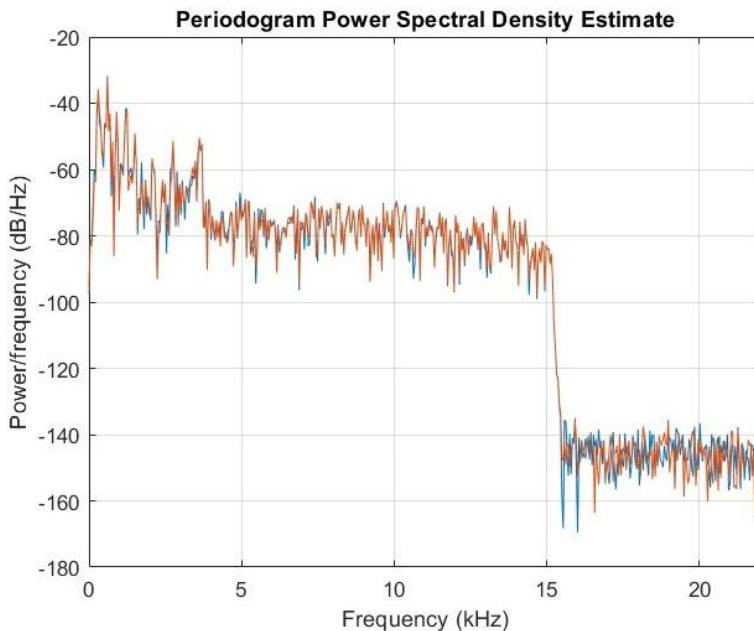


شكل 5: طیف فاز سیگنال بعد از جبران فاز

ح. فیلتر قبل از جبران فاز

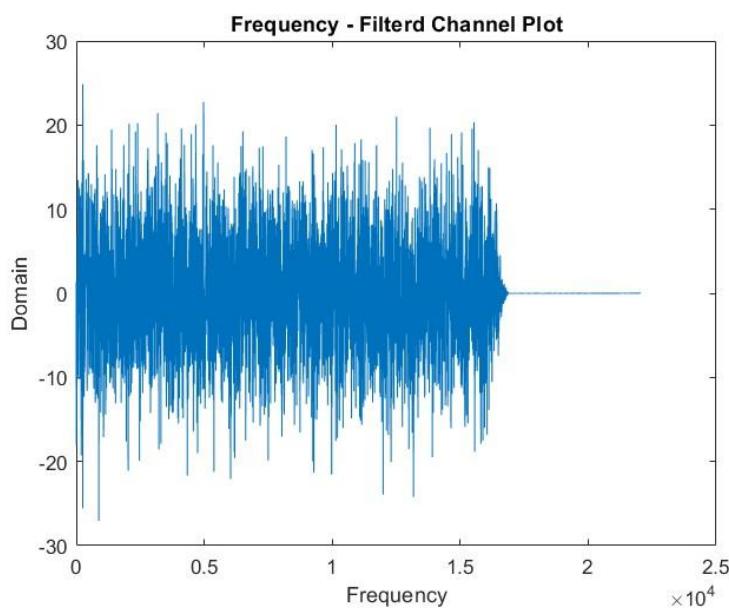
با استفاده از کد زیر دقیقتر به طیف ورودی می‌پردازیم.

```
plot(psd(spectrum.periodogram,message.data,'Fs',message.fs,'NFFT',nfft));
```



شکل ۶: طیف دقیقتر سیگنال ورودی

همان طور که دیده می‌شود ۱۶ کیلوهرتز اول مهم است بنابراین با استفاده از کد زیر سیگنال را فیلتر کرده و در `Channel_Filterd.wav` ذخیره می‌کنیم.



شکل ۷: سیگنال اول فیلتر شده و سپس جبران فاز شده

```

Filterd = lowpass(ifft(signal_ft_WithNoise), 16000, 44100);
Filterd_ft = fft(Filterd,nfft);
Filterd_channel = Filterd_ft .* exp(-4i*pi/9);
figure;
plot(F(1:nfft/2), Filterd_channel(1:nfft/2));
xlabel("Frequency");
ylabel("Domain");
title("Frequency - Filterd Channel Plot");
audiowrite('Channel_Filterd.wav', real(ifft(Filterd_channel)), message.fs);
میبینیم که نویز صدا کمتر شده است.

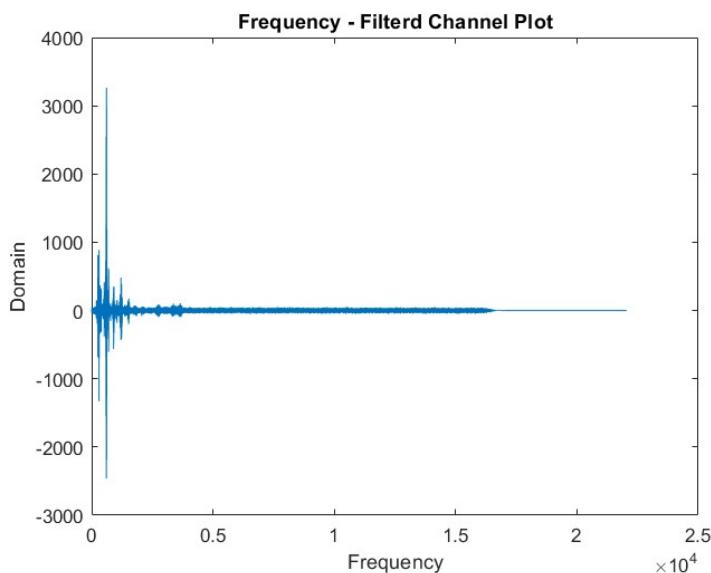
```

خ. فیلتر پس از جبران فاز
توضیحات آن به شکل بالا است.

```

Filterd_After = lowpass(real(ifft(channel_ft)), 16000, 44100);
Filterd_After_ft = fft(Filterd_After,nfft);
figure;
plot(F(1:nfft/2), Filterd_After_ft(1:nfft/2));
xlabel("Frequency");
ylabel("Domain");
title("Frequency - Filterd Channel Plot");
audiowrite('Channel_After_Filterd.wav', real(ifft(Filterd_channel)),
message.fs);

```



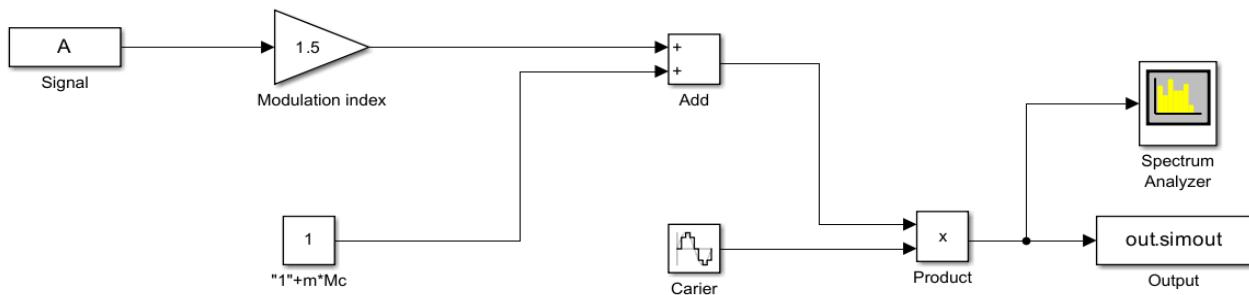
شکل ۸ : طیف سیگنال خواسته شده

صوت در فایل Channel_After_Filterd.wav ذخیره شده است.
بنظر روش قبلی خروجی بهتری را داشته است.

۲. بخش دوم - سیمولینک

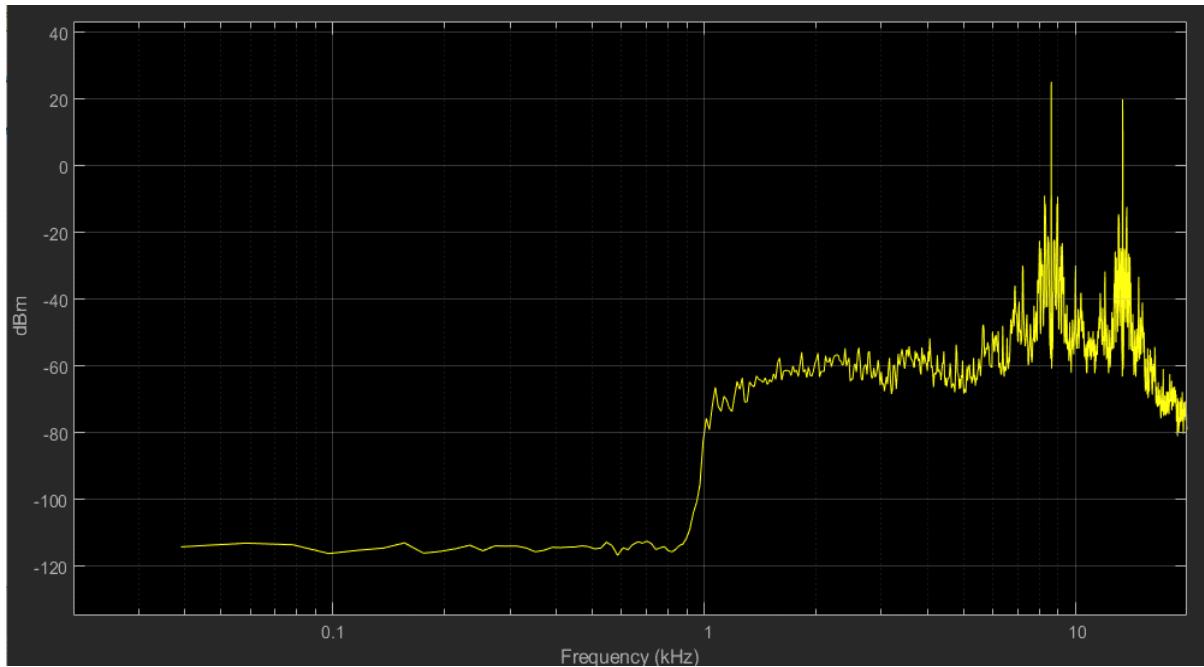
أ. مدولاتور AM (Am.slx)

با توجه به موارد گفته شده و $\mu = 0.2 + \frac{24}{200} = 0.32$ در نتیجه بلوک دیاگرام سیمولینک به صورت زیر میشود.

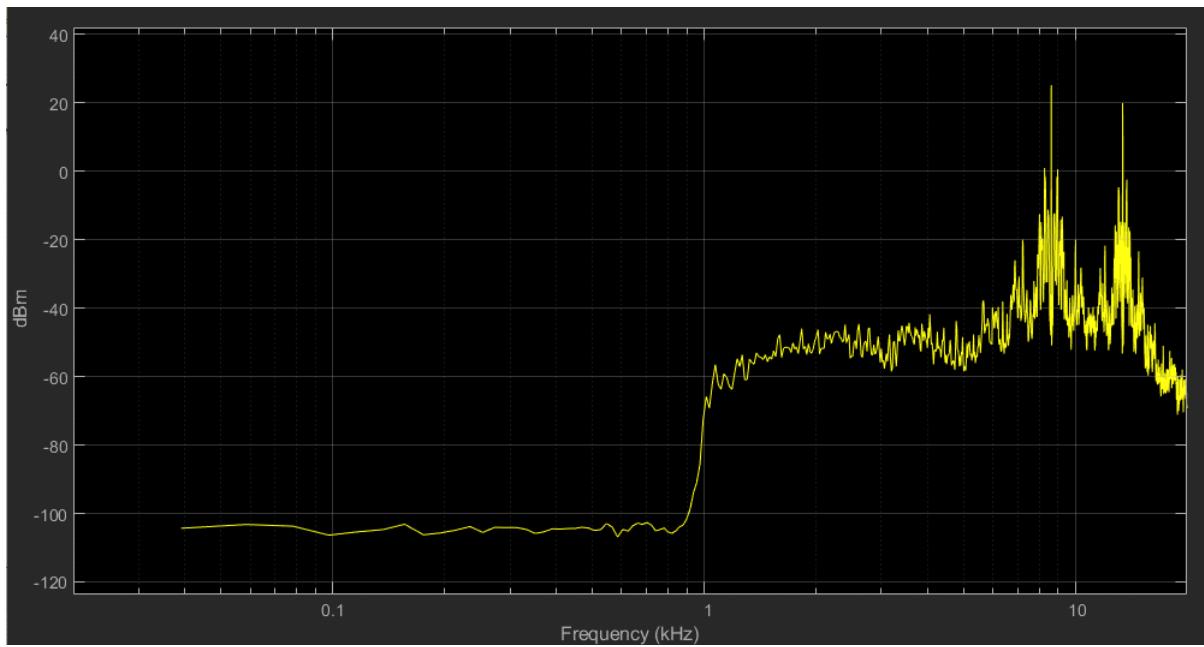


شکل ۶: بلوک دیاگرام مدولاتور AM

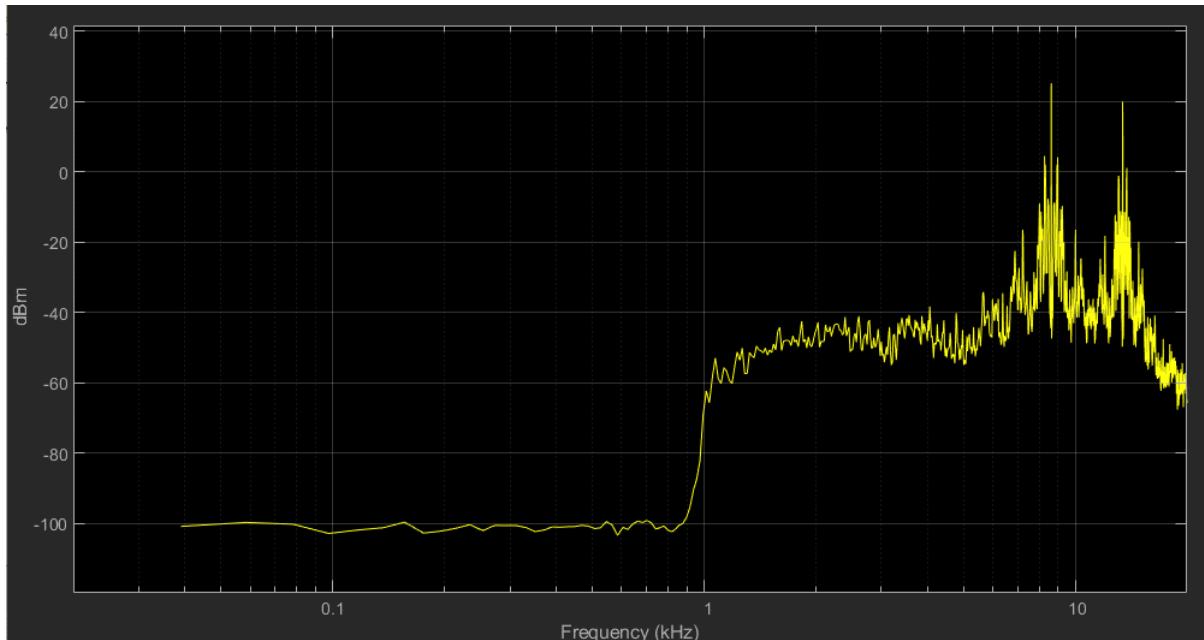
که با استفاده از اسکریپت A.m در فایل ها استراکت A ساخته میشود.



شکل ۷: طیف سیگنال مدوله با $m=0.32$



شکل ۱۰: طیف سیگنال مدوله با $m=1$



شکل ۱۱: طیف سیگنال مدوله با $m=1.5$

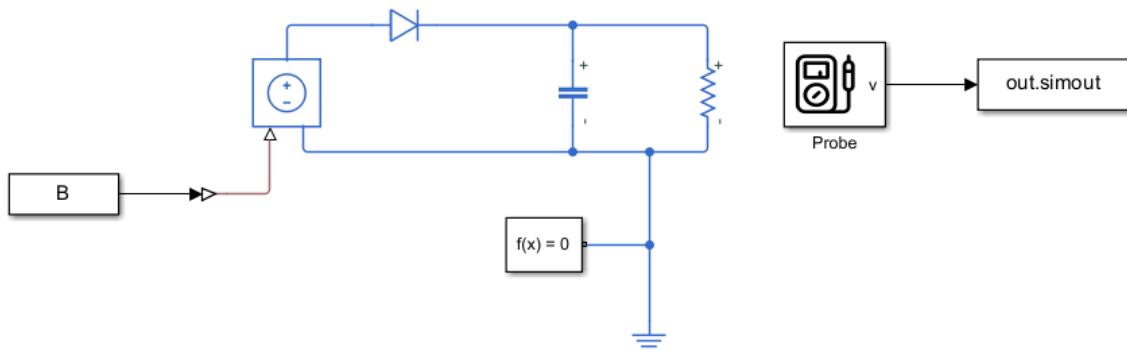
صدای ایجاد شده صدای کاملا غیر قابل فهمی است که صرفا ضرب های صدا در آن مشخص است.

صدا ها در فایل های Am_150.wav / Am_100.wav / Am_032.wav

که به ترتیب برای $m=1.5, 1.00, 0.32$ است.

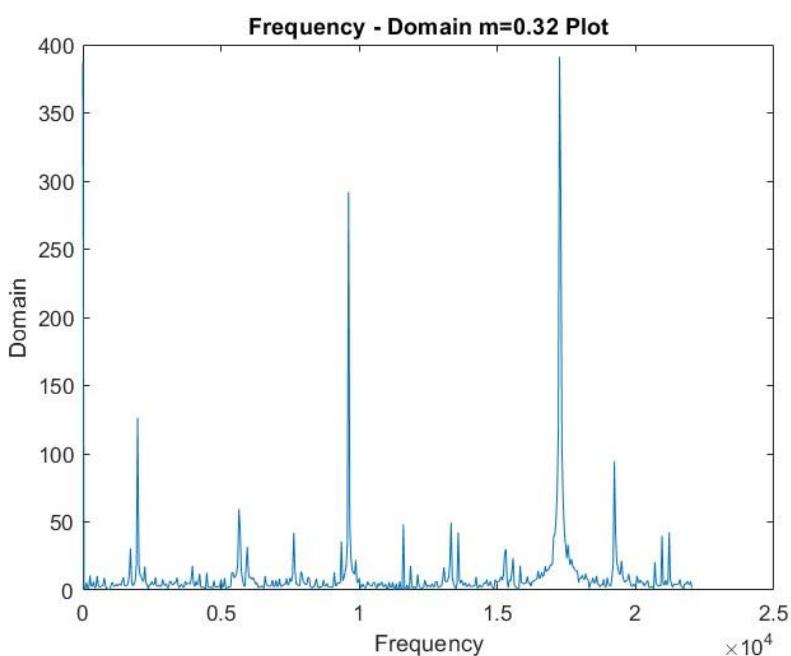
ب. دتکتور AM (AmDet.slx)

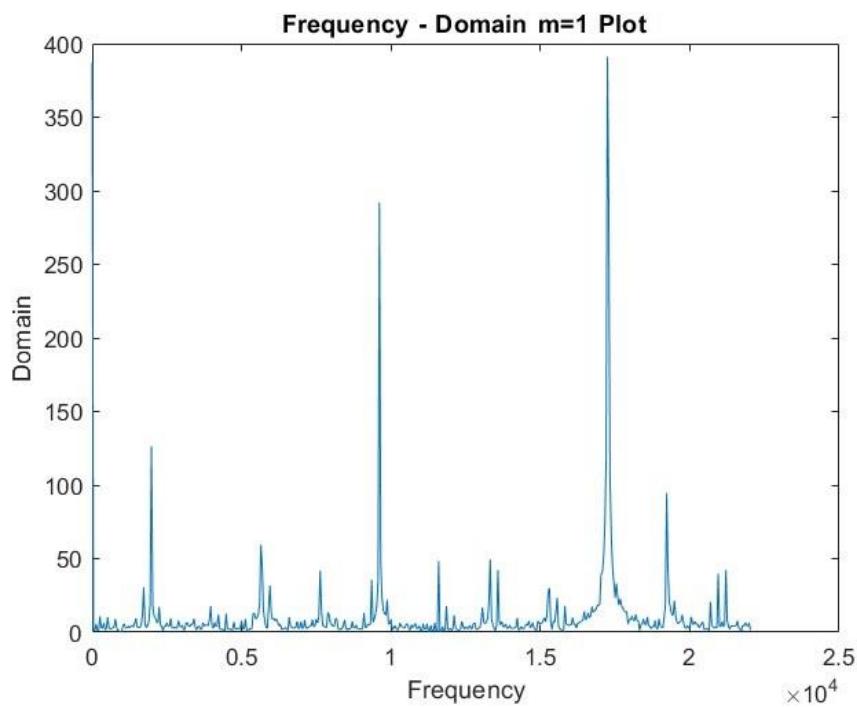
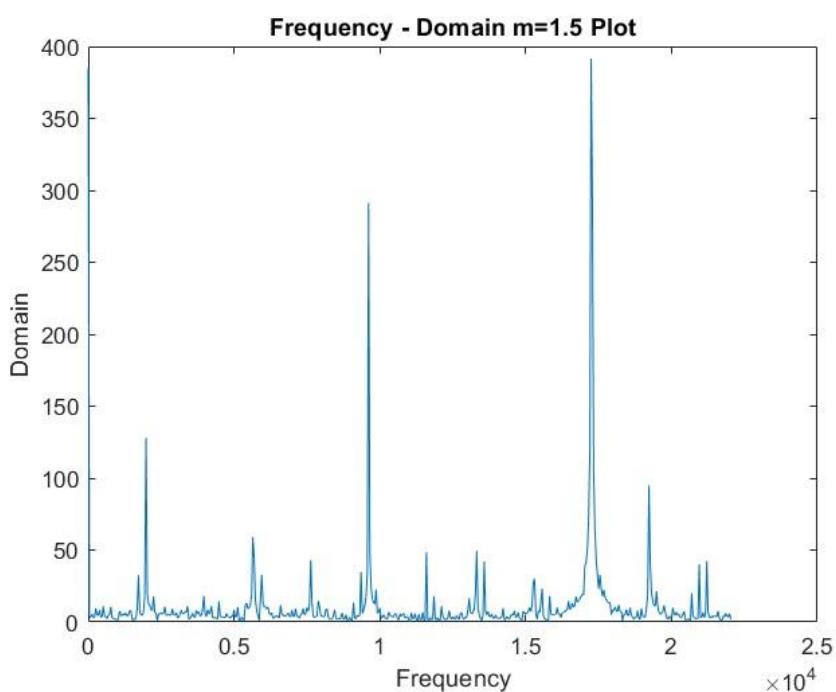
با توجه به مدار داده شده و با استفاده از SimScape بلوک دیاگرام زیر را درست میکنیم.



شکل ۱۲: بلوک دیاگرام دتکتور AM

که B با استفاده از اسکریپت B.m بدست می آید.

شکل ۱۳: طیف خروجی به ازای $m=0.32$

شکل ۱۴: طیف خروجی به ازای $m=1$ شکل ۱۵: طیف خروجی به ازای $m=1.5$

با توجه به نمودار ها می بینیم که سیگنال خروجی باند پایه شده و یک سری نویز به همراه خودش دارد که با یک فیلتر پایین گذر کیفیت بهتری پیدا میکند که همین طور هم می شود. Det_F_100.wav/Det_F_150.wav/Det_F_032.wav خروجی بعد فیلتر Det_150.wav/Det_100.wav/Det_032.wav و خروجی قبل فیلتر هستند.

می بینیم که هر چه قدر ضریب مودولاسون بیشتر باشد قدر صدای خروجی نسبت به نویز ایجاد شده بیشتر است.