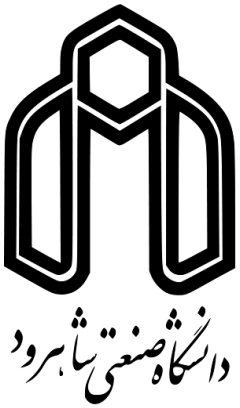
بسم الله الرحمن الرحیم



آزمایش ۶

پروژه نهایی درس

نام استاد : جناب دکتر مقیمی

نام دانشجو : محمد توزنده جانی

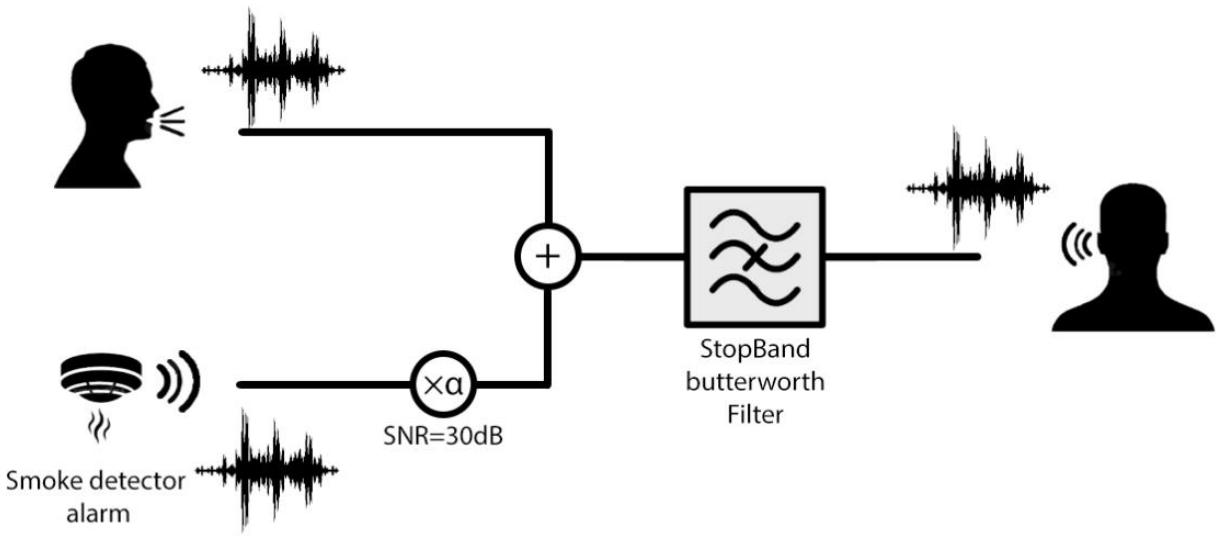
۹۷۲۰۷۸۳

تاریخ ارسال :

۱۴۰۱/۰۳/۱۱

نیمسال دوم ۱۴۰۰-۱۴۰۱

فهرست مطالب



۱) یک سیگنال صوتی ضبط شده از صدای خود که در آن شماره دانشجویی خود را بیان می‌کنید، بسازید، سپس این فایل صوتی را با فایل صوتی نویز پیوست شده (صدای آژیر دستگاه تشخیص دود) جمع کنید به نحوی که نسبت سیگنال به نویز در حدود 30^{db} باشد. قابل توجه است که باید فایل ضبط شده با مشخصه‌های سیگنال صوتی پیوست تطابق داشته باشد. (فایل‌های صوتی ممکن است بصورت استریو ضبط شده باشند که با فرخوانی آن در متلب بصورت ماتریس n*2 بارگذاری میشود برای این تمرین از اطلاعات یک کانل (یک ستون داده) استفاده کنید) سپس بهترین فیلتر باتورثی که میتواند صدای آژیر را از صوت شما حذف کند را طراحی کنید و بکار بگیرید. نتیجه را در حوزه زمان و فرکانس بررسی کنید.

ذخیره سازی سیگنال ها :

- ابتدا با دستور audioread صوت مربوط به صدای آژیر دستگاه تشخیص دود را به عنوان سیگنال نویز و صوت معرفی خود را به عنوان سیگنال داده اصلی در متلب ذخیره می کنیم:

```
clc;
clear;
close all;

%% Create  Noise & Data Signal-----------------------------------------------
[sig,fs]=audioread('1-MyName.mp3');              %signal --> Data Signal
[noise,fsn]=audioread('2-Smoke_Detector.mp3');   %noise --> Smoke_Detector
```

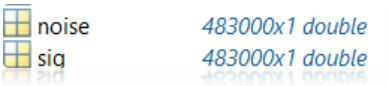
یکسان سازی ابعاد ماتریس سیگنال ها :

- ماتریس های هر سیگنال دارای دو ستون می باشد که ستون دوم مربوط به حالت stereo است.
- جهت حذف داده های اضافی بلااستفاده ، با دستور زیر هر دو سیگنال صوتی را از stereo به mono تبدیل می کنیم. (ستون دوم هر ماتریس را حذف می کنیم):
-

```
%% Convert Stereo to Mono ---------------------------------------------
sig=sig(:,1);
noise=noise(:,1);
```

یکسان سازی ابعاد ماتریس سیگنال ها:

- در ادامه با دستور زیر ابعاد دو ماتریس را یکسان می کنیم تا مشکلی جهت جمع کردن دو سیگنال با هم نداشته باشیم :




```
%% Equalize the dimensions of the matrices -----------------------------
sig(483001:length(sig),:)=[];
noise(483001:length(noise),:)=[];
```

جمع دوسیگنال پیام و نویز :

- می دانیم نسبت سیگنال به نویز برابر 30dB (دسی بل) می باشد ، به عبارت دیگر در جمع دو سیگنال ضریبی مانند α در کنار نویز ظاهر می شود :

$$d = x + \alpha n$$

- حال رابطه این ضریب را با SNR بدست می آوریم :

$$SNR = \frac{Psig}{Pnoise} = \frac{\frac{1}{N}\sum_{k=1}^{N} S_k^2}{\frac{1}{N}\sum_{k=1}^{N} (\alpha n_k)^2}$$

- حال به سادگی ضریب α بدست می آید و با توجه به این ضریب سیگنال d که جمع دو سیگنال داده و نویز می باشد نیز بدست می آید.

$$\alpha = \sqrt{\frac{\sum_{k=1}^{N} S_k^2}{SNR \sum_{k=1}^{N} (n_k)^2}}$$

- کد های مربوط به جمع دو سیگنال با توجه به روابط بالا :

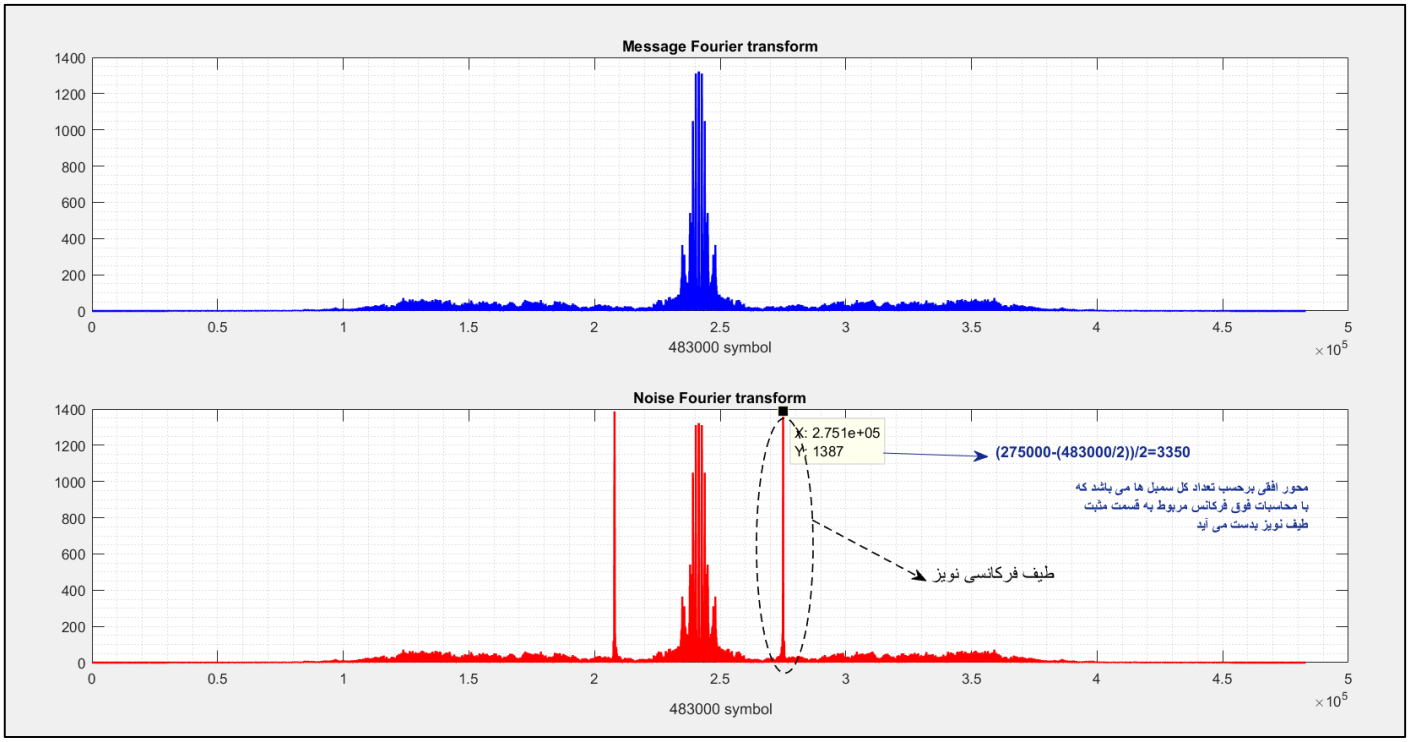
```
%% Adding Noise to Signal ----------------------------------------------------
% Compute Sout = signal + noise such that SNR = Ps/Pn
%s: Input signal
%SNR: Desired signal-to-noise ratio
%d: Output signal
SNR=30;       %in dB
Es = sum(sig(:).^2);
En = sum(noise(:).^2);
alpha = sqrt(Es/(SNR*En));
Sout = sig+(alpha*noise);
audiowrite('3-Combined_signal(SNR=30).wav',Sout,fs);   %Save Combined Signal
```

- با دستور audiowrite سیگنال جمع شده را با نام دلخواه در پوشه ی متلب و با فرکانس سیگنال داده ذخیره می نماییم .
- با پخش فایل صوتی ایجاد شده ، صدای خود(سیگنال پیام) و صدای آژیر دستگاه(نویز) شنیده می شود.
- حال با طراحی فیلتر باترورث می خواهیم نویز را تا از سیگنال اصلی حذف کنیم.

بررسی در حوزه ی فرکانس:

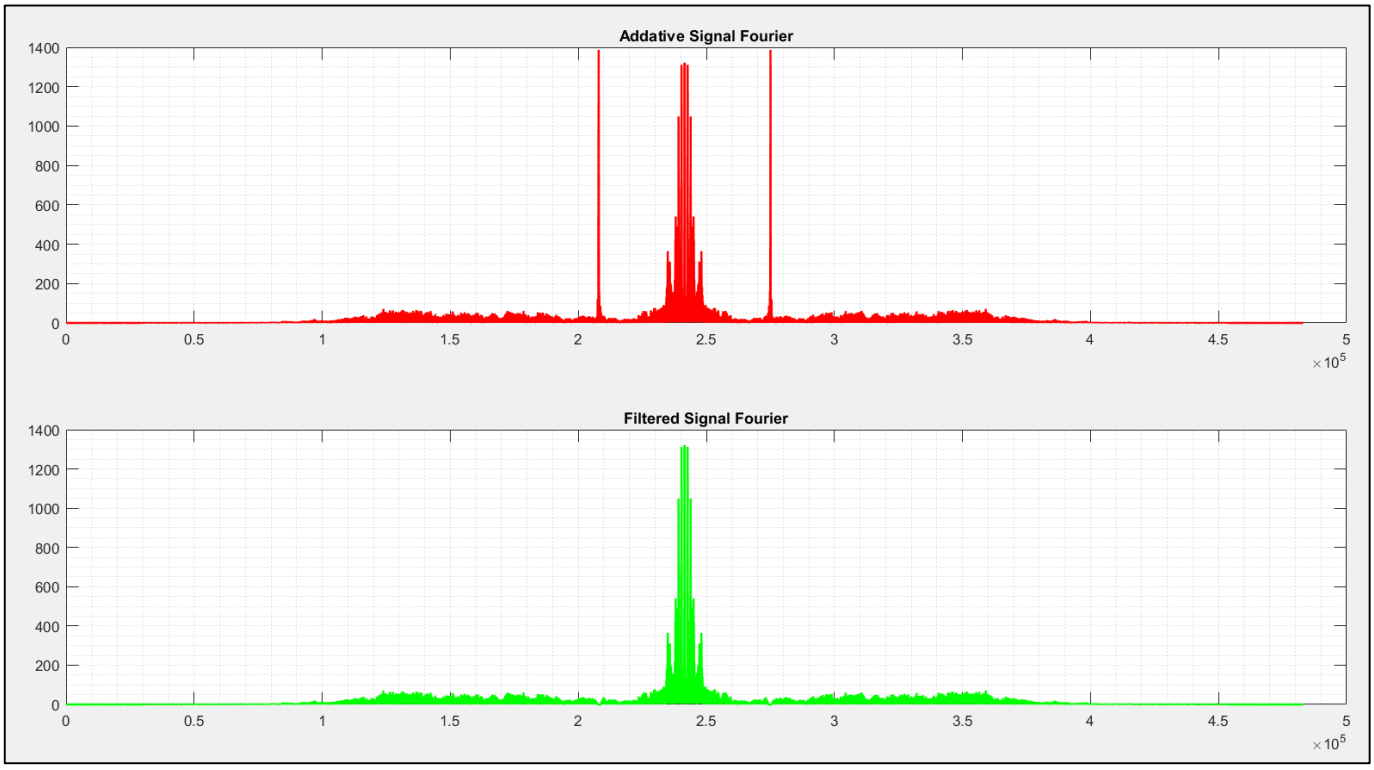
- با دستور fft تبدیل فوریه سیگنال جمع شده با نویز و سیگنال پیام را بدست می آوریم با مقایسه این دو، محدوده ی طیف فرکانسی نویز مشخص است :

```
%% Frequency Analyze ----------------------------------------
figure
subplot(2,1,1)
a0=fft(sig);
b0=fftshift(a0);
plot(abs(b0),'b','linewidth',1.5);
title('Message Fourier transform')
xlabel('483000 symbol')

grid minor

subplot(2,1,2)
a1=fft(Sout);
b1=fftshift(a1);
plot(abs(b1),'r','linewidth',1.5);
title('Noise Fourier transform')
xlabel('48000symbol')
grid minor
```



شکل۱: طیف فرکانسی سیگنال پیام و سیگنال جمع شده

- حال پس از ساخت سیگنال فیلتر شده تبدیل فوریه آن را در بخش CommandWindow گرفته و با سیگنال جمع شده مقایسه می کنیم : در شکل زیر مشخص است طیف فرکانسی مربوط به نویز تاحد مطلوبی حذف شده است :

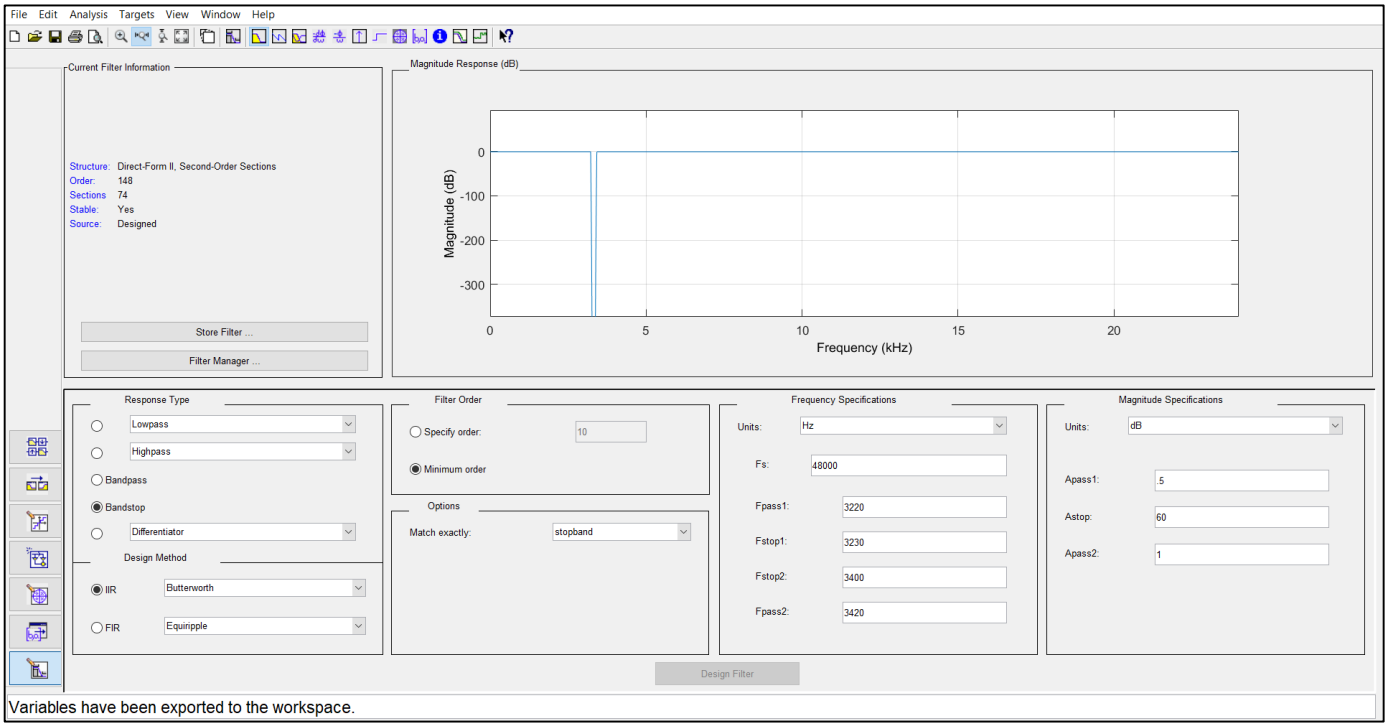


شکل۲: طیف فرکانسی سیگنال جمع شده و سیگنال فیلتر شده

طراحی فیلتر باتروورث:

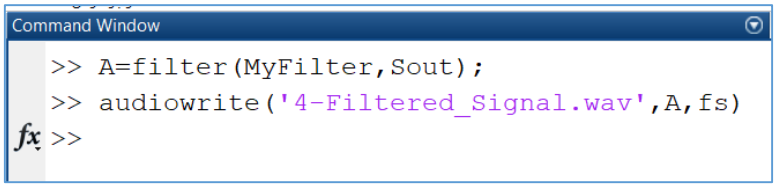
- با توجه به تحلیل در حوزه ی فرکانس در قسمت قبل ، از یک فیلتر باتروورث IIR میانگذر (BandStop) برای طراحی استفاده می کنیم.
- یک فیلتر IIR علاوه بر صفر ، قطب نیز دارد.
- اولویت ، طراحی فیلتر هایی با مرتبه پایین تر می باشد زیرا هرچه مرتبه فیلتر پایین تر باشد ، سرعت پردازش اطلاعات توسط فیلتر بالاتر است.
- در پنجره ی FilterDesigner پارامتر های مربوط به فیلتر قرار داده شده است :
- Astop سطح باند توقف بر حسب dB می باشد که برابر ۶۰ قرار می دهیم ، هرچه این مقدار بزرگتر باشد ، درجه فیلتر بالاتر است .
- ❋ در تحلیل فرکانسی مشاهده کردیم در فرکانس حدود 3325Hz طیف فرکانسی نویز وجود داشت بنابراین در پنجره frequency specifications محدوده ی باند توقف را در محدوده این فرکانس
- به صورت زیر در نظر می گیریم؛ تا سیگنال نویز حذف گردد:

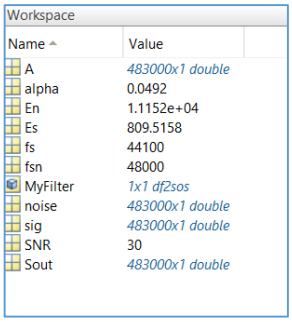
$$fs = 48000$$
$$fpass1 = 3220 \qquad fpass2 = 3420$$
$$fstop1 = 3230 \qquad fstop2 = 3400$$



شکل ۳: پنجره FilterDesigner و پارامتر های طراحی فیلتر باتروورث

- پس از وارد کردن پاراکتر ها از منوی File قسمت Export فیلتر را با نام دلخواه در Workspace ذخیره کرده و با وارد کردن دستور های زیر در CommandWindow سیگنال فیلتر شده خروجی را بدست می آوریم و آن را به عنوان یک فایل صوتی جدید در پوشه متلب ذخیره می نماییم و نتیجه کار یعنی سیگنال خروجی فیلتر را می شنویم :



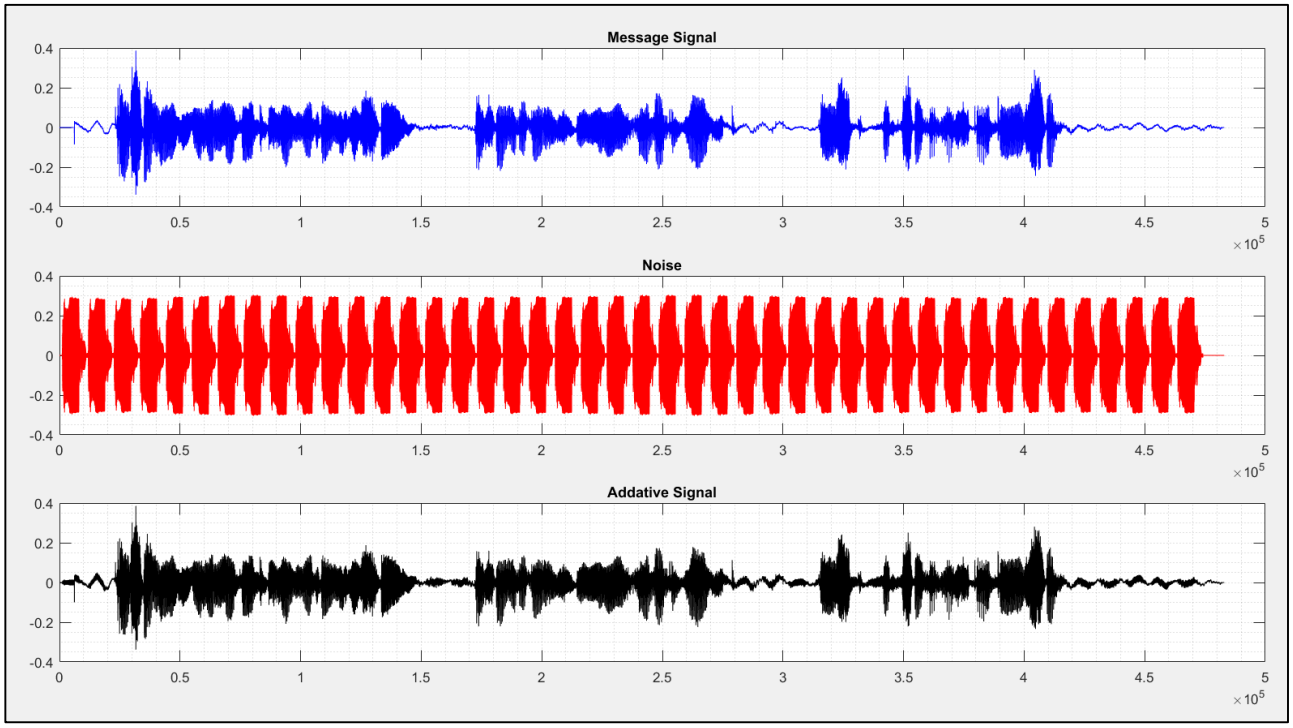


شکل ۴ : workspace متلب

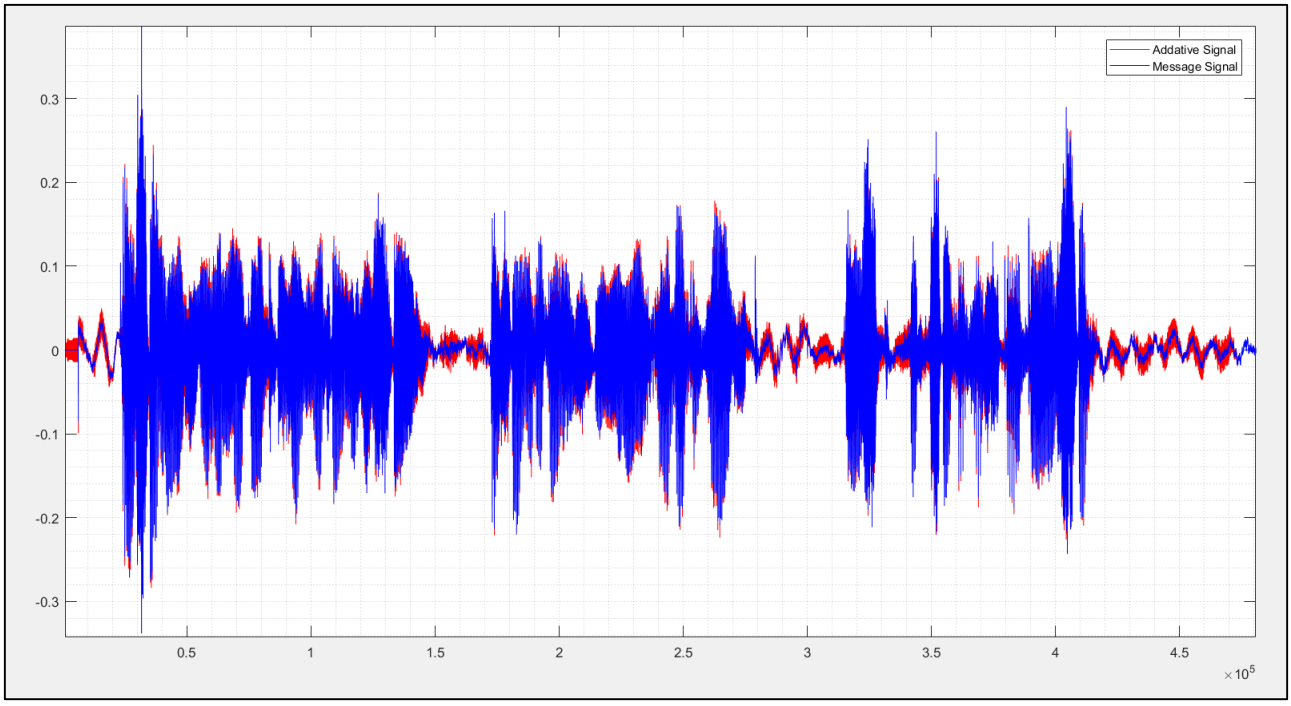
بررسی در حوزه ی زمان:

- در این بخش ، سیگنال های پیام ، نویز و سیگنال جمع را در حوزه ی زمان رسم و بایکدیگر مقایسه می کنیم:
- شکل موج آبی رنگ سیگنال پیام می باشد.
- شکل موج قرمز رنگ سیگنال نویز(صدای آژیر دستگاه) می باشد.
- شکل موج سیاه رنگ ، حاصل جمع دو سیگنال پیام و نویز است.

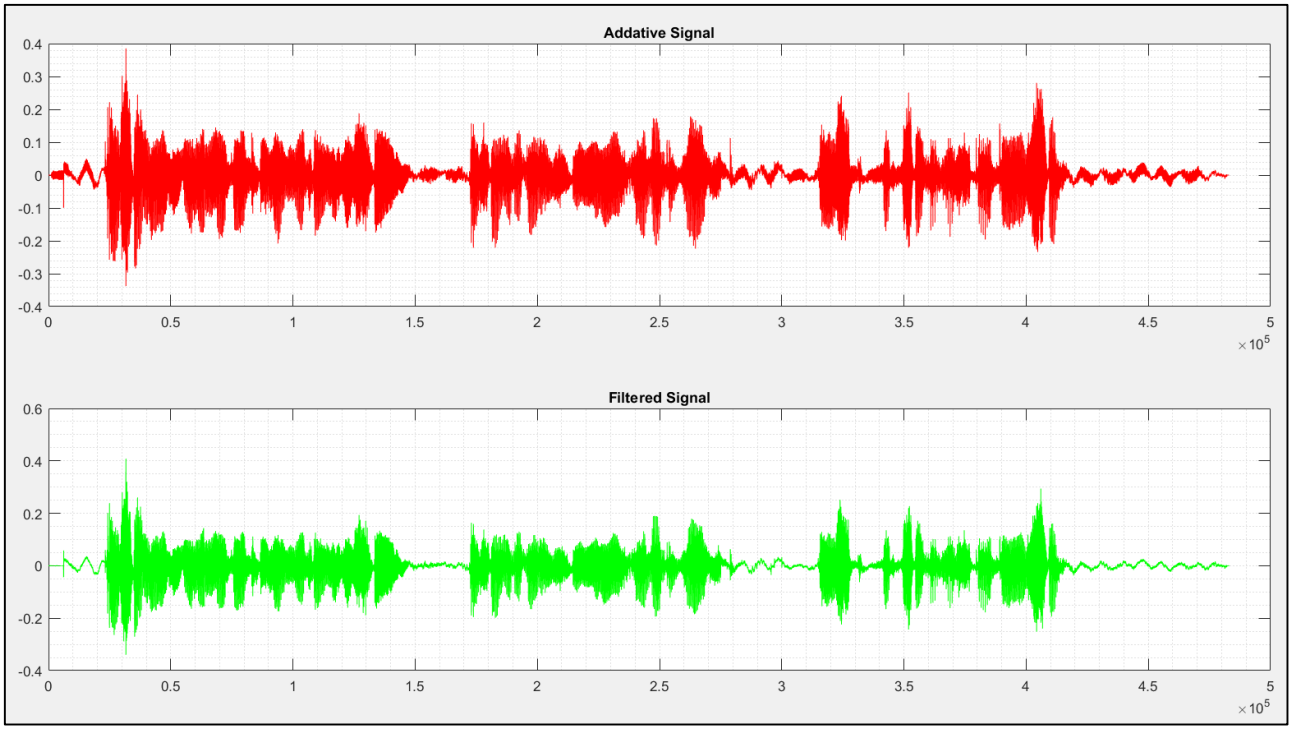
```
%% Time Analyze-----------------------------------------------------
figure
subplot(3,1,1)
plot(sig,'b')
title('Message Signal')
grid minor
subplot(3,1,2)
plot(noise,'r')
title('Noise')
grid minor
subplot(3,1,3)
plot(Sout,'k')
title('Addative Signal')
grid minor

figure
plot(Sout,'r')
hold on
plot(sig,'b')
grid minor
legend('Addative Signal','Message Signal')
```



شکل ۵ : سیگنال پیام - سیگنال نویز(صدای آژیر دستگاه) – سیگنال جمع شده در حوزه ی زمان



شکل ۵ : نمایش اثر نویز بر سیگنال پیام (قسمت های قرمز رنگ ، افزایش دامنه سیگنال پیام ناشی از افزوده شدن نویز به آن می باشد).



شکل ۵ : سیگنال قرمز ورودی فیلتر (سیگنال جمع) - سیگنال سبز خروجی فیلتر (سیگنال فیلتر شده)

* جهت ذخیره سیگنال فیلتر شده در WorkSpace و پخش صوت آن در نرم افزار متلب پس از Export کردن فیلتر از پنجره FilterDesigner دستورات زیر در CommandWindow اعمال شود؛

```
>> A=filter(MyFilter,Sout);
>> audiowrite('4-Filtered_Signal.wav',A,fs2)
>> sound(A,fs2)
```

لینک های پخش فایل های صوتی :

برای پخش روی نوشته های زیر کلیک کنید؛

۱- پخش سیگنال پیام

۲- پخش سیگنال نویز

۳- پخش سیگنال جمع شده (سیگنال پیام همراه با نویز)

۴-پخش سیگنال فیلتر شده (خروجی فیلتر باتروورث - نتیجه کار)