بسم الله الرحمن الرحیم

درس سیستم های مخابراتی

استاد حمید بهروزی

تمرین سری ۳ متلب

امیرحسین رستمی ۹۶۱۰۱۶۳۵

گزارش تمرین سری ۳

دانشگاه صنعتی شریف

سوال اول:

قسمت الف:

از آنجاکه ما به دنبال بازیابی پوش سیگنال هستیم اگر به طریقی بتوانیم حاصل جبری بلوک دیاگرام حاصل را به صورت کانولوشن سیگنال اصلی با تعدادی ضربه بیابیم و سپس شرط فیلترینگ را جهت بازیابی کامل سیگنال اصلی را اعمال کنیم(نگه داشتن ضربه با شیفت صفر و فیلترینگ باقی ضربه ها) را وارد مساله کنیم داریم که مساله حل می گردد.

توجه کنید که ایده بالا از اینکه سیگنال ورودی ما به فیلتر به صورت قدرمطلق پوش(که طبق شرایط مساله خود پوش سیگنال است) ضرب در قدرمطلق کسینوس که خود رابطه ای با کسینوس(و در حوزه ی فرکانس با ضربه ها!) دارد به دست آمده است!

در اولین وهله بسط فوریه | cos(x) | را بررسی می کنیم:

حال به محاسبه ی بسط سری فوریه ی | cos(x) | می پردازیم:

$$|\cos(x)| = \begin{cases} \cos(x) & \text{if } 0 \le x \le \pi/2 \\ -\cos(x) & \text{if } \pi/2 \le x \le \pi, \end{cases}$$

we have that

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} |\cos(x)|\cos(nx)\,dx = \frac{2}{\pi}\int_0^{\pi}|\cos(x)|\cos(nx)\,dx \\ &= \frac{2}{\pi}\int_0^{\pi/2}|\cos(x)|\cos(nx)\,dx + \frac{2}{\pi}\int_{\pi/2}^{\pi}|\cos(x)|\cos(nx)\,dx \\ &= \frac{2}{\pi}\int_0^{\pi/2}\cos(x)\cos(nx)\,dx - \frac{2}{\pi}\int_{\pi/2}^{\pi}\cos(x)\cos(nx)\,dx. \\ a_1 &= \frac{2}{\pi}\int_0^{\pi/2}\cos^2(x)\,dx - \frac{2}{\pi}\int_{\pi/2}^{\pi}\cos^2(x)\,dx = 0. \end{aligned}$$

Using the following trigonometric identity, with $a = x, b = nx$,

$$\cos(a)\cos(b) = \frac{\cos(a+b)+\cos(a-b)}{2},$$

we find

$$\begin{aligned} a_{2m} &= \frac{4}{\pi\,(1-4m^2)}\cos(\frac{2m\pi}{2}) = \frac{4}{\pi\,(1-4m^2)}(-1)^m \\ a_{2m+1} &= \frac{4}{\pi(1-4(2m+1)^2)}\cos(\frac{(2m+1)\pi}{2}) = 0, \qquad m = 1, 2, 3, \ldots. \end{aligned}$$

The expansion of $|\cos(x)|$ into a trigonometric Fourier series in the interval $[-\pi, \pi]$ is thus

$$|\cos x| = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n\cos(nx) + b_n\sin(nx)) = \frac{2}{\pi} + \frac{4}{\pi}\sum_{m=1}^{\infty}\frac{(-1)^m}{1-4m^2}\cos(2mx)$$

حال همانطور که در عبارت بالا ملاحظه می کنید بسط سری فوریه ی | cos(x) | به دست آمده است.حال برویم سراغ تبدیل فوریه سیگنال داریم که تبدیل فوریه ی سری نوشته شده عبارت است از:

توجه کنید که ضریب بسط فوریه ی یک سیگنال با منقبض یا منبسط شده ی آن در حوزه ی زمان برابر است به اصطلاح ویژگی زیر را سری فوریه دارد:

$f(x) \rightarrow a_k$

$f(\varphi x) \rightarrow a_k$

حال از انجا که ما | cos(2piFct) | داریم که که در بسط فوریه ی آن فاصله ی ضربه های فرکانس از یکدیگر برابر fc خواهد بود(چون ضریب فرکانس کسینوسی های بسط برابر fc است) حال با توجه به رابطه ی دست آمده در بالا و ویژگی ذکر شده در بالا داریم که در نهایت تبدیل فوریه ی بسط سری فوریه ی | cos(2piFct) | برابر با عبارت زیر خواهد بود:

$$FourierTransform : \left(1 + \mu X(f)\right) Conv \left(\frac{2}{\pi} Delta(f) + \sum \frac{4}{\pi} Delta(f + kf_c)\right)$$

Note k ≠ fc.

حال باید فیلتر را به طریقی قرار بدهیم که در نهایت فقط Delta(f) از عبور کند و اثری از ترم های دیگر در خروجی فیلتر نباشد در تا در نهایت بتوانیم سیگنال پوش را بازیابی کنیم. حال اگر فرض کنیم پهنای باند سیگنال اصلی BW باشد داریم که در نهایت پهنای باند فیلتر باید در شرط زیر صدق کند:

$$\boldsymbol{Signal_{BW} \leq LPF_{BW} \leq f_c - Signal_{BW}}$$

$$\boldsymbol{And\ LPF_a = \frac{1}{\frac{2}{\pi}} = \frac{\pi}{2}}$$

حال پس از اعمال فیلتر بالا روی خروجی قدرمطلق کننده داریم که فقط Delta(f) عبور می کند و سیگنال پوش پس از فیلتر پایین گذر احیا می گردد.

قسمت ب:

همانطور که در قسمت قبل نشان دادیم که بلوک دیاگرام قسمت اول پوش سیگنال را احیا می کند حال برای حل این قسمت ابتدا بررسی کنیم به بلوگ دیاگرام قسمت الف چه چیزی وارد می شود و طبق آن قضاوت کنیم: داریم که خروجی مشتق گیرنده عبارت زیر است

$$\frac{d}{dt}\left(Acos(2\pi f\ ct\ +\ 2\pi f_\Delta \int x(t)dt)\right)\ \rightarrow\ -2A\pi(fc\ +\ f_\Delta x(t))sin(2\pi f\ ct\ +\ 2\pi f_\Delta \int x(t)dt)$$

لذا داریم که پوش سیگنال برابر مقدار زیر است:

توجه کنید که ممکن است بگویید که عبارت بالا به صورت ضرب پوش در سینوس است ولی خب سینوس همان کسینوس است منتها با اخلاف فاز π/2 که تغییری در محاسبات ما ایجاد نمی کند.

$-2A\pi(fc \ + \ f_\Delta x(t))$

لذا داریم که سیگنال پوش ما به صورت جمع مقدار DC + ضریبی از سیگنال اصلی است حال داریم که فرض کنیم DC را به طریقی فیلتر می کنیم و درنهایت خواهیم داشت که:

$$-2A\pi\big(fc \ + \ f_\Delta x(t)\big) \rightarrow DC\ block \rightarrow \ -2A\pi f_\Delta x(t) \rightarrow LPF \rightarrow x(t)$$

$$so\ in\ amplitude\ of\ LPF\ is : A_{LPF} * \frac{2}{\pi} * (2A\pi f_\Delta) = 1 \dashrightarrow A_{LPF} = \frac{1}{4Af_\Delta}$$

حال می خواهیم پهنای باند سیگنال FM خود را محاسبه کنیم به کمک روابط موجود در کتب و جزوه ی استاد داریم که در نهایت پهنای باند سیگنال FM عبارت است از مقدار زیر:

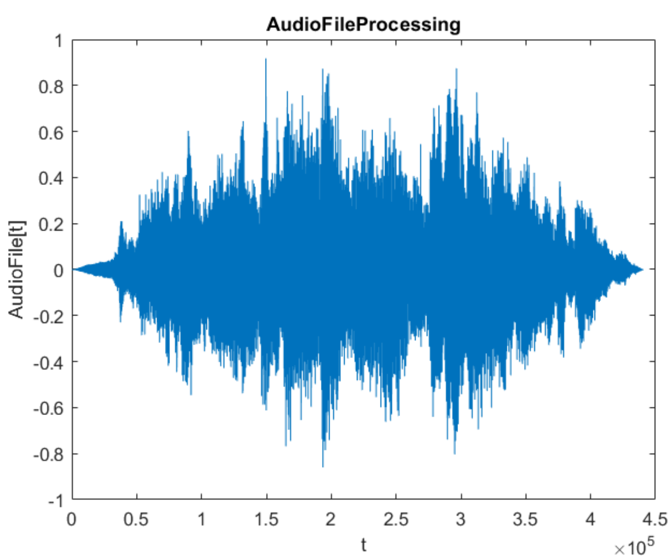
$$BW = 2(\beta + 1)f_m \dashrightarrow BW = 2\left(\frac{f_\Delta}{f_m} * \max\big(x(t)\big) + 1\right) * fm = 2(f_\Delta * \max\big(x(t)\big) + f_m)$$

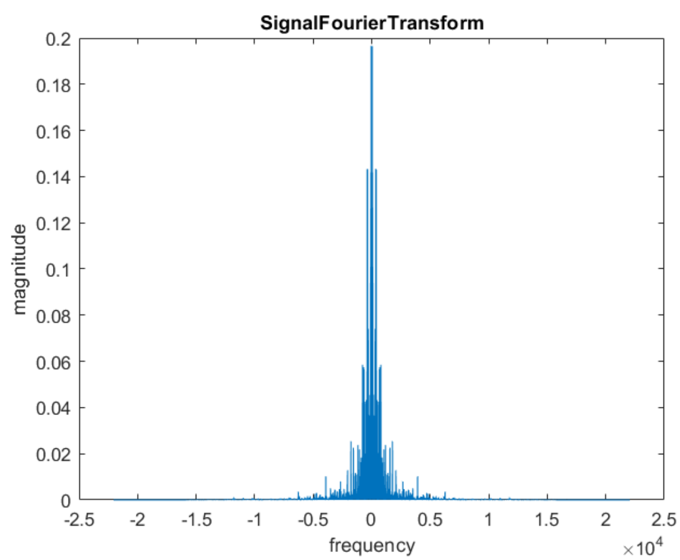
از آنجا که مقدار ماکزیمم سیگنال x(t) رو نداریم و مقداریر f_m, f_Δ در صورت سوال داده نشده اند داریم که در نهایی ترین عبارت برای پهنای باند سیگنال عبارت است از رابطه ی بالا.

قسمت پ:

ا.

ابتدا سیگنال را در حوزه ی زمان و فوریه رسم می کنیم:





در ابتدا برای منظور پهنای باند سوال،این موضوع که منظور شخص طراح همانند تمرین سری قبل 99 درصد انرژی کل، را در نظر میگیریم و محاسبات را انجام می دهیم:

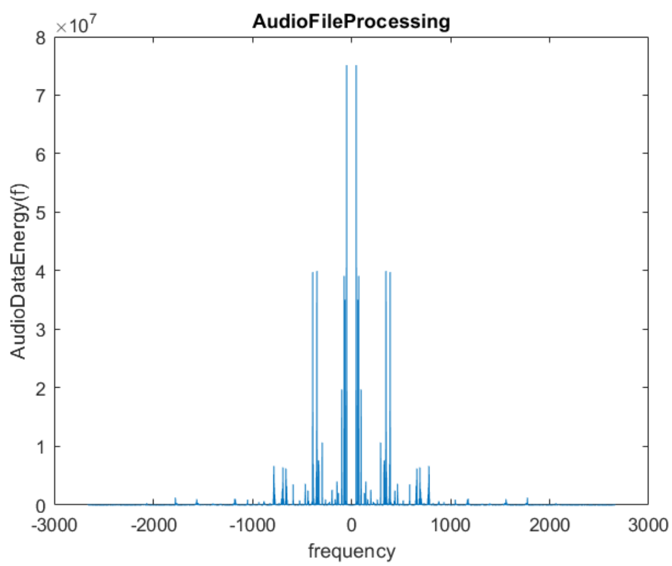
حال میخواهیم با تعریف ارائه شده پهنای باند سیگنال را پیدا کنیم، ابتدا برای این امر از سیگنال با دستور FFT تبدیل فوریه میگیریم و یک محود برای فرکانس درست میکنیم که از صفر تا fs را به تعداد نقاط فوریه تقسیم میکند. اندازه تبدیل فوریه را تقسیم بر fs کرده و آنرا در مزدوج مختلط خود ضرب میکنیم تا چگالی توان به دست بیاید ولی از انجا که در حوزه گسسته هستیم و جمع این چگالی طیفی ها باید برابر انرژی کل شود و در محاسبه آن ما انتگرال میگیریم دوباره مانند استدلال بخش های قبل باید یکبار دیگر تقسیم بر fs آنرا بکنیم.

ما طیف دوطرفه در اختیار داریم برای جمع هر فرکانس در پهنای باند باید به طور متقارن جلو برویم برای همین از انجا که می دانیم سیگنال ورودی حقیقی و طیف آن زوج میباشد یک طیف یک طرفه درست میکنیم که از ابتدا تا وسط طیف قبلی میباشد حال که به نوعی در مرکز آرایه که برابر است با مصداق فرکانس صفر تبدیل فوریه ی دوطرف است قرار میگیریم(البته ما در تابع انرژی قرار میگیریم که برابر است با ضرب تبدیل فوریه در مزدوج آن)و به صورت متقارن به محاسبه ی انرژی موجود در همسایگی نقطه ی فرکانس صفر و به دامنه ی freqIterator (که در کد از آن استفاده کرده ام) می پردازیم و در هر مرحله به مقایسه ی انرژی موجود در همسایگی درنظر گرفته شده با 99 درصد انرژی کل که با جاروب روی کل انرژی حوزه ی فرکانس به دست آمده است می کنیم و در صورت عدم برابری پله پله این freqIterator را افزایش می دهیم تا سرانجام به 99 درصد انرژی کل برسیم و نتیجه نهایی برابر است با :

SignalBW = 2661.1051110193450313090579584241

که مقدار آن به هرتز میباشد پس تا حدودای 2.66KHz محتوای فرکانسی دارد.

با در نظر گرفتن این پهنای باند داریم که نمودار انرژی آن به شکل زیر خواهد بود.



۲.

محاسبه ی فرکانس قطع مناسب(آزمون و خطا)

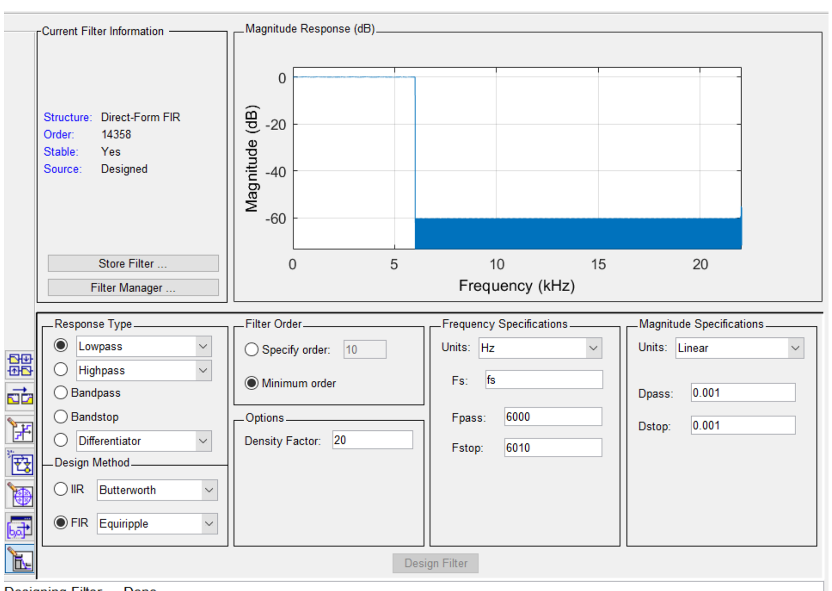
برای این قسمت چون نیاز به آزمون و خطا است ما از دستور LowPass استفاده می کنیم و به کمک قطعه کد زیر روی لبه های مختلف قطع جاروب می زنیم:

```
%% finding appropriate fm
%% playing original voice
sound(signal,fs);
for i = 1:20
    test = lowpass(signal,SignalBW+250*i,fs);
    sound(test,fs);
end
%% my Deaf ear's Best suggestion!!! :
test = lowpass(signal,6000,fs);
sound(test,fs);
%% lowpass with F_cut = SignalBW = 2.6k
test = lowpass(signal,SignalBW,fs);
sound(test,fs);
```

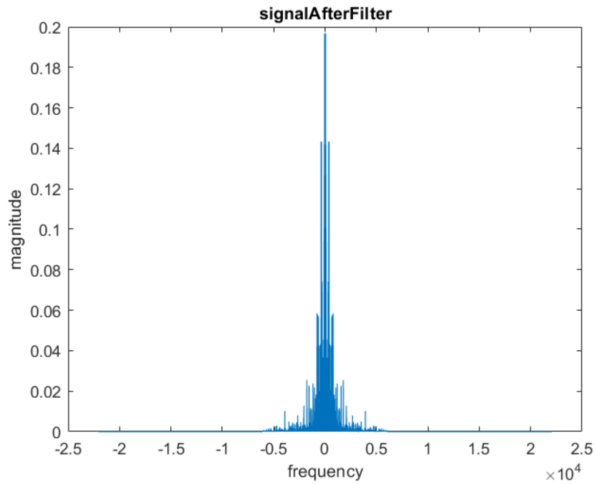
نکته ی مهم: صدا هایی شبیه به jiz jiz به خاطر طبیعی بودن ساز در صوت اولیه موجود است و طبیعتا دارای فرکانس بالایی است و لذا در اثر فیلترینگ پایین گذر این jiz jiz ها تا حد زیادی حذف شده است و ساختار طبیعیه صوت حذف شده است،اما در اثر از یه حد کم کردن فیلترقطع پایین گذر داریم که صدا حالت خفگی به خود میگیرد و انگار از داخل چاه! صدا در می آید که این به خاطر حذف شدن نت های پایین آهنگ است.البته چون آهنگ نت های پایین کمتری دارد یه کم تشخیص آستانه حذف نت های پایین سخت است.

حال سیگنال را از فیلتر با فرکانس طبیعی قطع 6000 عبور می دهیم:

ابتدا فیلترمدنظر را به کمک filterDesigner متلب طراحی می کنیم.



حال داریم که سیگنال را از فیلتر زیر عبور می دهیم و مشاهده می کنیم حالت فیلترشده ی واقعی سیگنال اصلی:



.۳

آپسمپلینگ کردن سیگنال اصلی به کمک دستور مفید معرفی شده ی interp که عبارت است از:

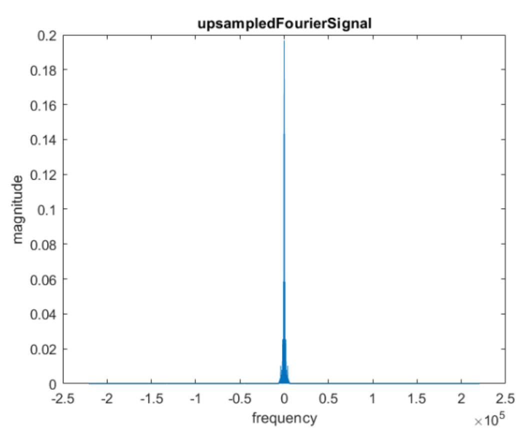
`y = interp(x,r)` increases the sample rate of `x`, the input signal, by a factor of `r`.

`y = interp(x,r,n,alpha)` specifies two additional values:

- `n` is half the number of original sample values used to interpolate the expanded signal.
- `alpha` is the normalized cutoff frequency of the input signal, specified as a fraction of the Nyquist frequency.

خوبی این تابع این است که سیگنال خروجی اش دیگر نیاز به اعمال فیلترینگ جهت استخراج یک نمونه از تبدیل فوریه ی سیگنال(از میان سایر تبدیل های تکراری ایجاد شده در اثر upsampling) نمی باشد و در خروجی اش سیگنال مفید را می دهد.

نکته ی مهم: ما در تک تک مراحل دامنه ی فرکانسی سیگنال را اصلاح می کردیم تا مطابق با مقداری واقعی اش باشد تا مضارب سمپلینگ مطابق به اندیس آرایه ی فرکانس! اما این تابع این نکته را رعایت نمی کند و از آنجاکه ما در نهایت مجدد سیگنال را به حالت اولیه با کمک دستور downSample انجام می دهیم و در صورت سوال نیز استفاده از این تابع بلامانع معرفی شده است با همین روند نمودار ها را رسم و ادامه می دهیم.



۴.

به کمک رابطه ی معرفی در سوال توابع مشتق گیر و انتگرال گیر را طراحی می کنیم،داریم:

$$\int_{nT_s}^{mT_s} x(t)dt = \sum_{k=n}^{m-1} \frac{x((k+1)T_s) + x(kT_s)}{2} T_s$$

$$\frac{dx}{dt}(t = mT_s) \approx \frac{x((n+1)T_s) - x((n-1)T_s)}{2T_s}$$

حال این دو تابع را به کمک قطعه کد زیر پیاده سازی می کنیم:

```
function integral = Integral(signal,fs)
integral = zeros(1,length(signal));
for i = 2 : length(signal)
    integral(1,i) = integral(1,i-1) + ( signal(i) + signal(i-1) )/(2*fs);
end
end
function derivative = derivative(signal,fs)
derivative = zeros(1,length(signal));
for i=2:length(signal)-1
    derivative(i) = (signal(i+1)-signal(i-1))*fs/2;
end
end
```

مطابق روابط گفته شده سیگنال مربوطه را طراحی می کنیم و نتیجه ی حاصل را رسم می کنیم.

نکته ی مهم،ابتدا تعدادی نکته ی دیگر ذکر می کنم و سپس کل نمودارها را می آورم:

حال به پیاده سازی مرحله مرحله ی قسمت های مختلف بلوک دیاگرام می پردازیم تا در نهایت عملیات دمودلاسیون انجام شود:

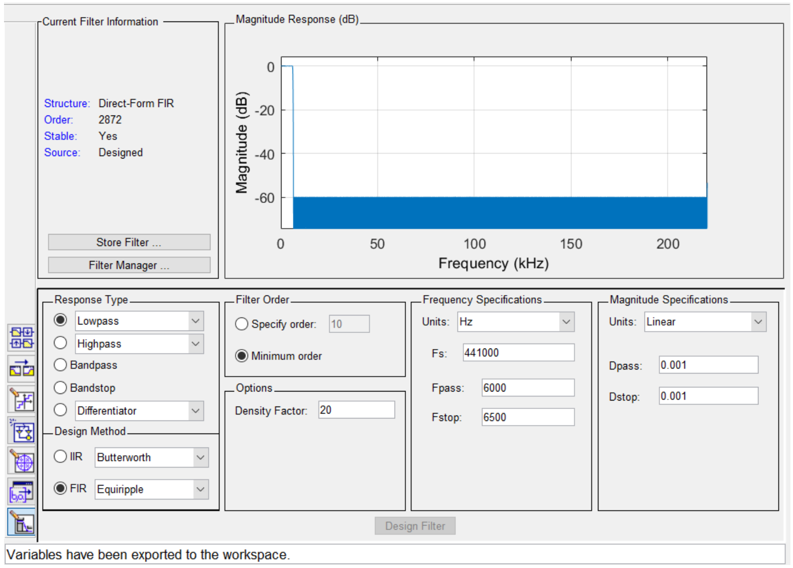
بخش فیلتر پایین گذر انتهایی دمودلاسیون:

همانطورکه گفتیم شرط فیلتر پایین گذر انتهای دمودلاسیون برابر است با:

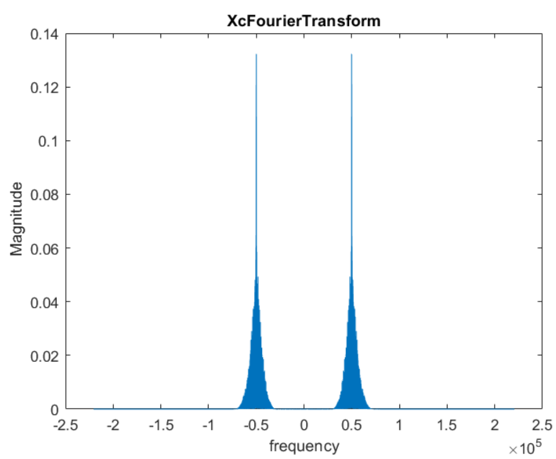
$$\boldsymbol{Signal_{BW} \le LPF_{BW} \le f_c - Signal_{BW}}$$

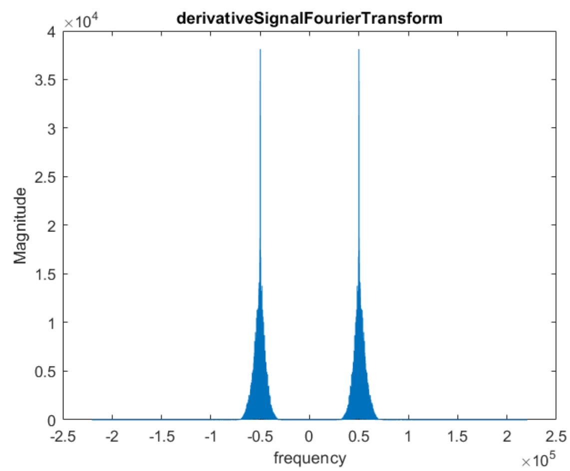
$$\boldsymbol{And\ LPF_a = \frac{1}{\frac{2}{\pi}} = \frac{\pi}{2}}$$

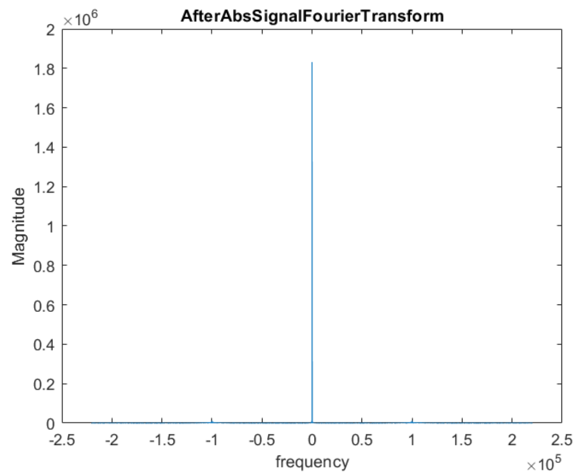
داریم که fc = 50KHz و مقدار SignalBw = 2.66KHz است و حال ما برای صدق کردن در نابرابری بالا فیلتری با مشخصات مشابه همان فیلتر پایین گذری که در ابتدای کار، سیگنال اصلی را ازش عبور دادیم درنظر می گیریم و داریم که فیلتر به شرح زیر است: توجه کنید که نرخ نمونه برداری در این فیلتر 10 برابر نرخ نمونه برداری در فیلتری است که سیگنال اصلی را ازش عبور دادیم.

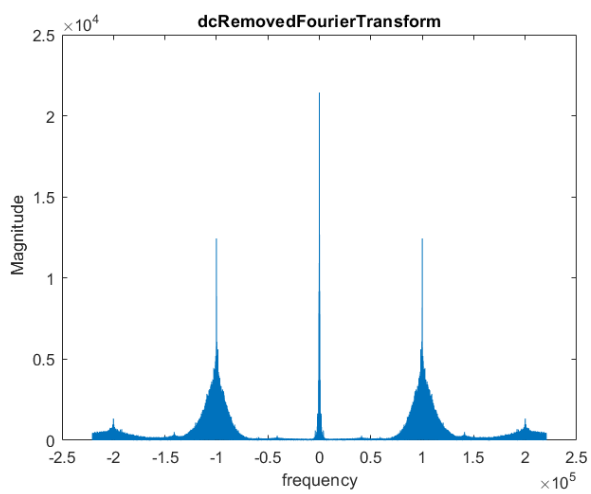


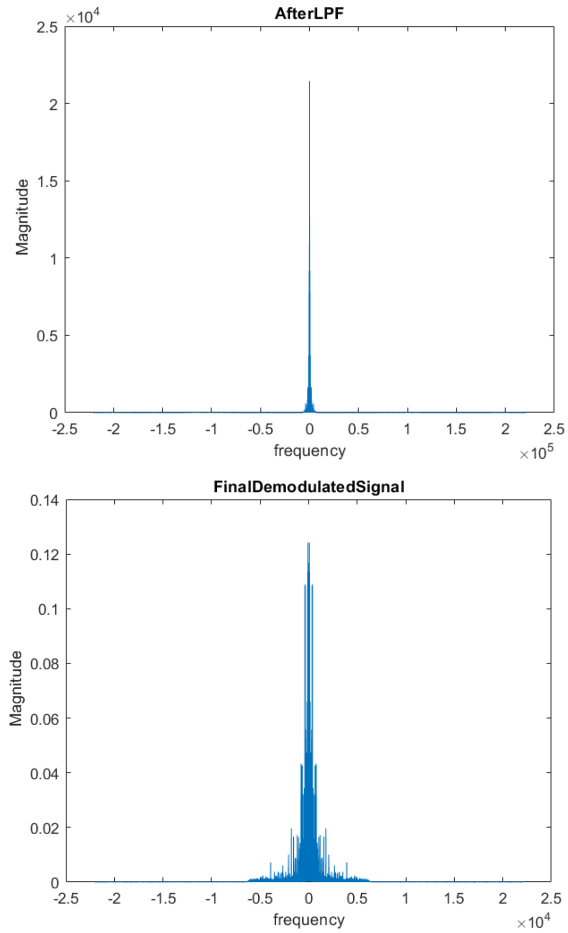
نمودار بخش های مختلف بلوک دیاگرام:




derivativeSignalFourierTransform
Magnitude
frequency
×10^4
×10^5


AfterAbsSignalFourierTransform
Magnitude
frequency
×10^6
×10^5


dcRemovedFourierTransform
Magnitude
frequency
×10^4
×10^5



خطای سیگنال بازیابی شده(دمدوله شده) با سیگنال ورودی:

```
DemoduleVsInputError = 2.6320e-06
```

سوال دوم:

قسمت الف:

برای انتگرال گیری از سیگنال به همان روشی که در سوال یک معرفی شد عمل میکنیم.

#یادآوری:

به کمک رابطه ی معرفی در سوال توابع مشتق گیر و انتگرال گیر را طراحی می کنیم،داریم:

$$\int_{nT_s}^{mT_s} x(t)dt = \sum_{k=n}^{m-۱} \frac{x((k+۱)T_s) + x(kT_s)}{۲} T_s$$

$$\frac{dx}{dt}(t = mT_s) \approx \frac{x((n+۱)T_s) - x((n-۱)T_s)}{۲T_s}$$

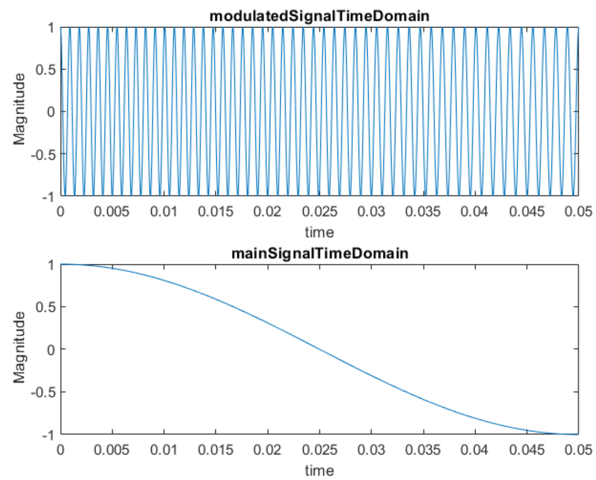
حال این دو تابع را به کمک قطعه کد زیر پیاده سازی می کنیم:

```
function integral = Integral(signal,fs)
integral = zeros(1,length(signal));
for i = 2 : length(signal)
    integral(1,i) = integral(1,i-1) + ( signal(i) + signal(i-1) )/(2*fs);
end
end
function derivative = derivative(signal,fs)
derivative = zeros(1,length(signal));
for i=2:length(signal)-1
    derivative(i) = (signal(i+1)-signal(i-1))*fs/2;
end
end
```

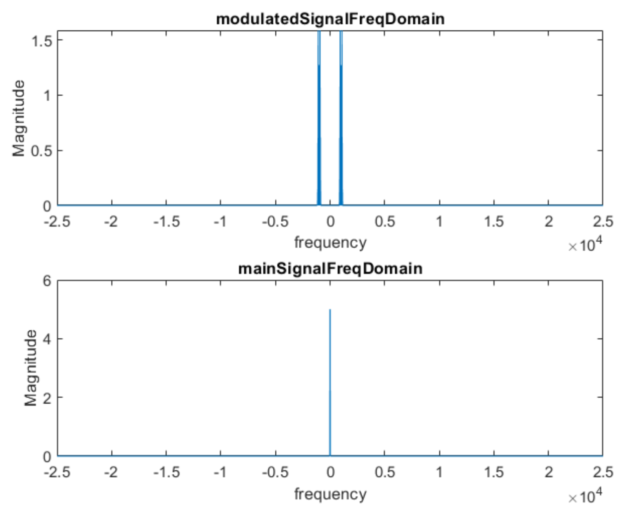
مطابق روابط و مقادیرگفته شده در صورت سوال سیگنال مربوطه را طراحی می کنیم و نتیجه ی حاصل را رسم می کنیم.

نمودار ها:

1ـ حوزه ی زمان



2ـ حوزه ی فرکانس



پهنای باند سیگنال:

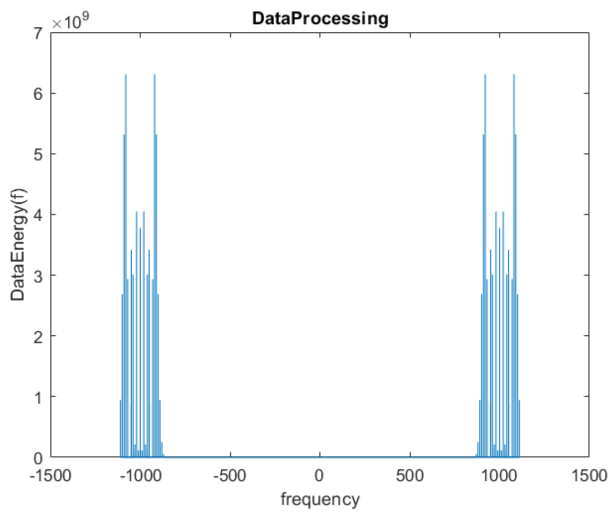
در ابتدا برای منظور پهنای باند سوال،این موضوع که منظور شخص طراح همانند تمرین سری قبل 99 درصد انرژی کل، را در نظر میگیریم و محاسبات را انجام می دهیم:

حال میخواهیم با تعریف ارائه شده پهنای باند سیگنال را پیدا کنیم، ابتدا برای این امر از سیگنال با دستور FFT تبدیل فوریه میگیریم و یک محود برای فرکانس درست میکنیم که از صفر تا fs را به تعداد نقاط فوریه تقسیم میکند. اندازه تبدیل فوریه را تقسیم بر fs کرده و آنرا در مزدوج مختلط خود ضرب میکنیم تا چگالی توان به دست بیاید ولی از انجا که در حوزه گسسته هستیم و جمع این چگالی طیفی ها باید برابر انرژی کل شود و در محاسبه آن ما انتگرال میگیریم دوباره مانند استدلال بخش های قبل باید یکبار دیگر تقسیم بر fs آنرا بکنیم.

ما طیف دوطرفه در اختیار داریم برای جمع هر فرکانس در پهنای باند باید به طور متقارن جلو برویم برای همین از انجا که می دانیم سیگنال ورودی حقیقی و طیف آن زوج میباشد یک طیف یک طرفه درست میکنیم که از ابتدا تا وسط طیف قبلی میباشد حال که به نوعی در مرکز آرایه که برابر است با مصداق فرکانس صفر تبدیل فوریه ی دوطرف است قرار میگیریم(البته ما در تابع انرژی قرار میگیریم که برابر است با ضرب تبدیل فوریه در مزدوج آن)و به صورت متقارن به محاسبه ی انرژی موجود در همسایگی نقطه ی فرکانس صفر و به دامنه ی freqIterator (که در کد از آن استفاده کرده ام) می پردازیم و در هر مرحله به مقایسه ی انرژی موجود در همسایگی درنظر گرفته شده با 99 درصد انرژی کل که با جاروب روی کل انرژی حوزه ی فرکانس به دست آمده است می کنیم و در صورت عدم برابری پله پله این freqIterator را افزایش می دهیم تا سرانجام به 99 درصد انرژی کل برسیم و نتیجه نهایی برابر است با :

```
SignalBW = 1.1100e+03
```

حال طیفی که در آن سیگنال 99درصد انرژی دارد:



قسمت دوم:

ابتدا توجه کنید که می توان ترم های cos(nx) را برحسب ترم های cos(x) نوشت:

Here is a neat way to derive what the answer will be, using Euler's formula $e^{ix} = \cos x + i \sin x$.

We have that $\cos nx$ is the real part of $e^{i(nx)} = \left(e^{ix}\right)^n = (\cos x + i \sin x)^n$

By the binomial formula, $(\cos x + i \sin x)^n = \sum_{k=0}^{n} i^k \binom{n}{k} \sin^k(x) \cos^{n-k}(x)$. Since i^k is real if and only if k is even, we therefore have (replacing k with a new indexing variable 2ℓ)

$$\cos nx = \sum_{\ell=0}^{\lfloor n/2 \rfloor} (-1)^\ell \binom{n}{2\ell} (\sin^2(x))^\ell \cos^{n-2\ell}(x).$$

Finally, we use the Pythagorean identity $\sin^2 = 1 - \cos^2$ to rewrite

$$\cos nx = \sum_{\ell=0}^{\lfloor n/2 \rfloor} (-1)^\ell \binom{n}{2\ell} (1 - \cos^2(x))^\ell \cos^{n-2\ell}(x) = \sum_{\ell=0}^{\lfloor n/2 \rfloor} \binom{n}{2\ell} (\cos^2(x) - 1)^\ell \cos^{n-2\ell}(x).$$

This agrees with Didier's answer.

حال همانطور که ملاحظه می کنید cos(nx) را می توان بر حسب توان هایی از cos(x) نوشت و حال داریم که پس می توان توان های cos(x) را هم بر حسب cos(alphaX) ها نوشت و لذا در نهایت داریم که اگر عنصر غیرخطی بلوک دیاگرام ما که برابر است با $y = \sum a_n v^n$ را در نهایت می توان بر حسب cos(alphaX) هایی که alpha از 1 تا n است نوشت، اما یک مشکل وجود دارد و آن این است که برای بازسازی نیاز است که عبارات فرکانسی با یکدیگر تداخل فرکانس نداشته باشند ولی این امر امکان پذیر نمی باشد به این دلیل که هنگام فیلترینگ قطعا اثراتی از کسینوس های با فرکانس بالاتر در خروجی باقی می ماند ☹
حال تنها کاری که از دستمان برمی آید این است که شرط کارلسون را چک کنیم در هر مرحله (که عبارت است از مقدار زیر)

$$\mathrm{BW} = 2(kf_{\Delta} + f_m)$$

کافی است به ازای دو مقدار k , k+1 پهنای باند مدوله شده ی کسینوسی را محاسبه کنیم و ببینیم آیا k کوچیکتر در k+1 بزرگتر جا می گیرد یا خیر :

$$\frac{BW}{2}\Big|_{k+1} + (k+1)f_c \geq \frac{BW}{2}\Big|_k + kf_c$$

با بررسی این نامساوی به دست می آوریم که بله همیشه این اتفاق می افتد! و کافیست فیلتری بگذاریم که اولین k را استخراج کند تا انشالله به کمک آن سیگنال اصلی را بازیابی کنیم.یعنی کافیست که که یک فیلتر پایین گذر با فرکانس قطع زیر بگذاریم که در نهایت سیگنال اصلی را به کمک شرط کارلسون احیا کنیم:

$$So\ for\ the\ first\ k = 1 \rightarrow \frac{BW}{2}\Big|_{k=1} + (k=1)f_c = \frac{BW}{2} + f_c$$

از فواید این روش در این است که ما به کمک این روش β بزرگتری را تولید کنیم،(می دانیم که تولید بتا های کوچک برای ما ساده تر است) و با این متد می توانیم که بتا را افزایش بدهیم به اصطلاح فنی داریم که با این روند می توانیم:

NBFM(narrowband_FM) را به WBFM (wideband_FM) تبدیل کنیم و از همه مهم تر که با این روند داریم که اثرا عنصر غیر خطی را هندل می کنیم که این خود دستآورد به نسبت زیادی است!!!

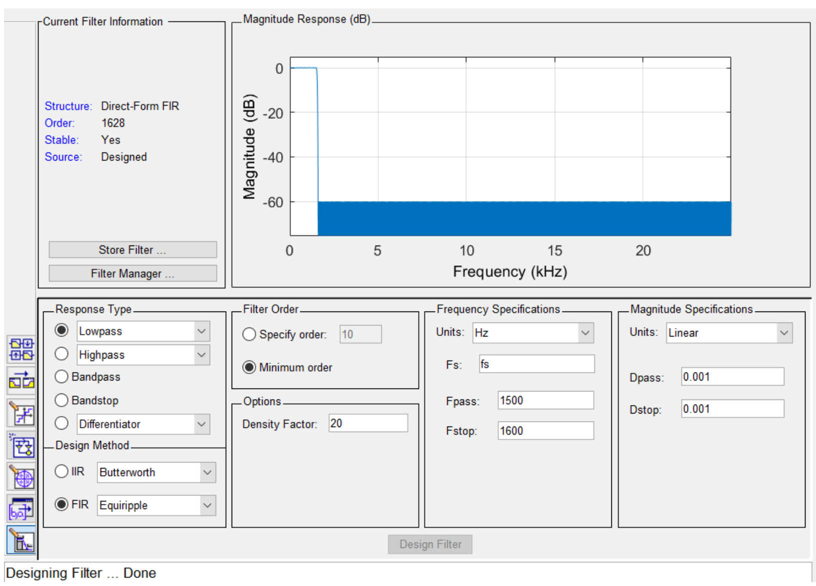
عنصر غیر خطی را v^3 در نظر میگیریم و با آن حل را ادامه می دهیم:

$$\cos(3x) = 4\cos^3(x) - 3\cos(x) \rightarrow \cos^3(x) = \frac{1}{4}\cos(3x) + \frac{3}{4}\cos(x)$$

حال ابتدا فیلتر مدنظر را طراحی می کنیم:

$$So\ for\ the\ first\ k = 1 \rightarrow \frac{BW}{2}|_{k=1} + (k = 1)f_c = \frac{BW}{2} + f_c$$

با توجه به بخش های قبل داریم که BW = 1.11KHz = 1110 Hz و داریم که fc = 1KHz و لذا داریم که حدودا فرکانس 1.5KHz برای فرکانس قطع فیلتر پایین گذرمان مناسب است،شکل فیلترطراحی شده مان به شرح زیر خواهد بود:

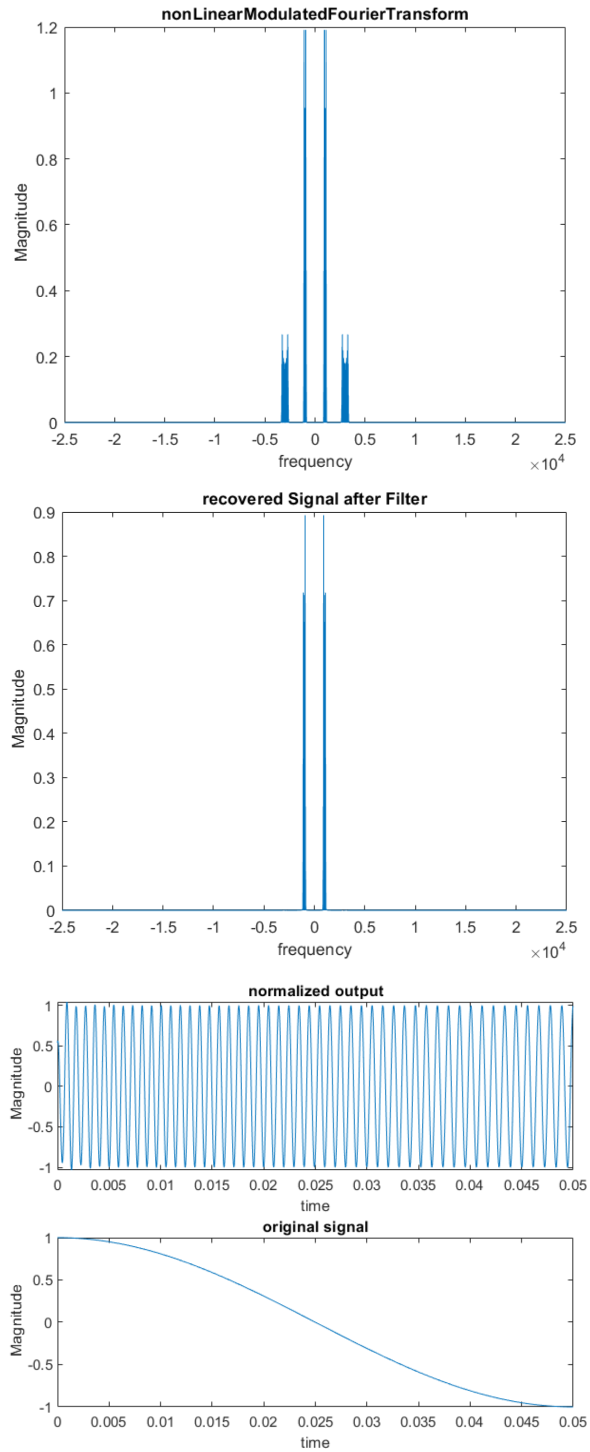


البته توجه کنید:استفاده از این فیلتر باعث می شود که در نهایت خروجی فیلتر دچار دیلی شود و ما به کمک محاسبه ی groupDelay عه تابع تبدیل سیستم این اشکال را بر طرف می کنیم،به این شکل که:

```
delay = mean(grpdelay(Hf)); -> calculates group Delay of Hf(Your Filter)
filt = temp;
filt(1:delay) = []; -> removing occurance of delay!

Filt now is ready to use!
```

حال داریم که نمودار ها به شرح زیر خواهد بود:



خطای سیگنال اصلی با سیگنال بازیابی شده از بلوک غیرخطی:

```
MSEerrorOfMainVSrecoveredNonLinear = 3.0520e-06
```

قسمت پ:

ابتدا تابع hard limiter را مشابه گرفتن عملیات sign از ورودی اش در نظر میگیرم سپس همان فیلتر پایین گذر موجود در قسمت های قبل را قرار می دهیم تا در نهایت محتوای فرکانسی اول را جدا کنیم و از آن در قسمت های قبلی استفاده کنیم،توجه کنید که وقتی سیگنال hardLimiter را از فیلتر پایین گذر می گذرانیم داریم که در نهایت آن بخش تغییرات تیز را که دارای فرکانس بالا هستند فیلتر می شوند و حالات ناپیوسته شکل سیگنال خروجی ما اصلاح می شود.

A hard limiter produces an output vs input slope of zero (or near zero) at a specied output level. A soft limiter produces a gradual reduction of output vs input slope (dynamic gain) starting at a specified output level.

لذا اگر دقت کنید داریم که به نحوی ما در این حالت soft Limiter ساختیم و یعنی:

HardLimiter + LPF = SoftLimiter

$$AfterHardLimiter: \frac{4}{\pi}\sum \frac{(-1)^n}{2n+1}\cos\left[2\pi f_c t(2n+1) + (2n+1)v(t)\right]$$

For n = 0 to ∞

حال برویم سراغ مشتق گیر داریم که:

$$\frac{d}{dt}(Acos(2\pi f\ ct\ +\ 2\pi f_{\Delta}\int x(t)dt))\ \rightarrow\ -2A\pi(fc\ +\ f_{\Delta}x(t))sin(2\pi f\ ct\ +\ 2\pi f_{\Delta}\int x(t)dt)$$

همانطور که می بینید اطلاعات در دامنه سیگنال ما وجود دارد و داریم که با عبور از آشکار سازپوش خواهیم داشت که دامنه بالا که برابر است با DC + coef * x(t) وکافی است که DC سیگنال را حذف کنیم که در نهایتش خواهیم داشت که سیگنال خروجی حذف کننده ی DC عبارت است از : αx(t)

چند نکته:

1- اگر سیگنال ما DC نداشته باشد نیاز به اعمال حذف کننده ی DC نیست.
2- چون از فیلترهای طبیعی استفاده می کنیم داریم که دراثر عبور از فیلتر داریم که سیگنال دچار Delay می گردد لذا نیاز است تا به کمک تابع نوشته شده توسط خودمان که محاسبه می کند groupDelay به محاسبه اش بپردازیم و به تعداد آن داده هایمان را شیفت زمانی به جلو بدهیم تا اصلاحات لازم انجام شود.

اسم تابع حذف کننده ی delay را delaySyncer گذاشتیم.

حال پله پله بلوک دیاگرام را جلو می رویم:

در ابتدا داریم که من روند زیر را برای اعمال بلوک دیاگرام کردم ولی به مشکل برخورد:

```
[up,down] = envelope(afterDerivater);
dcBlocker = dsp.DCBlocker;
dcRemovedSignal = dcBlocker(transpose(up));
x = x(1:length(dcRemovedSignal));
MSEerror = immse(transpose(x),dcRemovedSignal)
```

خط اول جهت گرفتن envelope سیگنال بود،خط بعدی جهت گرفتن بخش DC سیگنال بود و در نهایت به محاسبه ی MSE پرداختیم و خب خطا زیاد بود!

حدس خطاهای احتمالی ممکن:

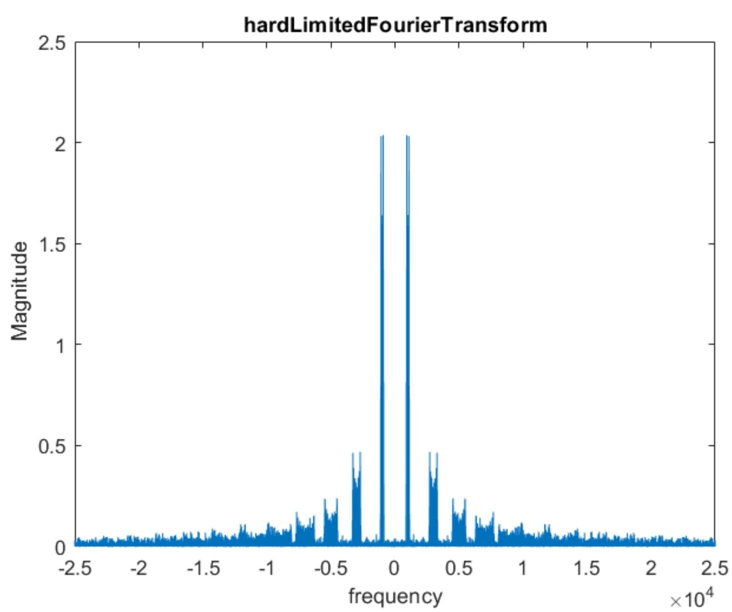
مشکل ساز شدن DC سیگنال،مشکل ساز شدن فیلترهای طبیعی،خاص بودن شرایط کار بلوک های DC و envelope،اما خب چطور این مشکل را برطرف کنیم؟

راه حل:متاسفانه مجبور شدیم تک تک بلوک دیاگرام هارا پیاده سازی کنیم.

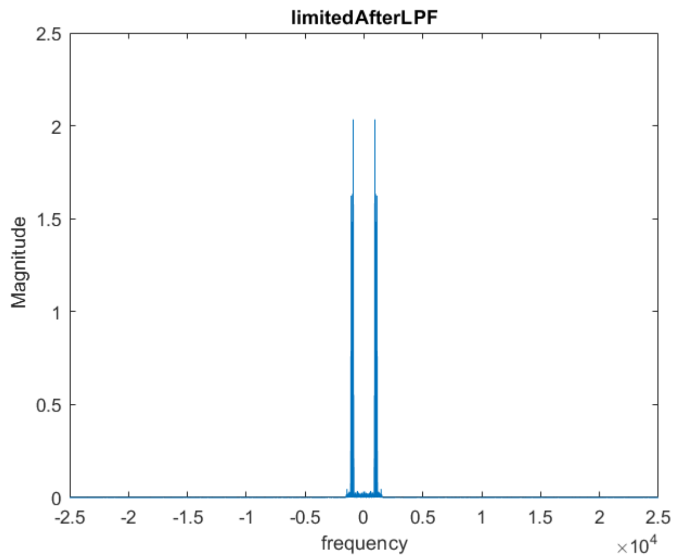
نکته مهم: DC سیگنال را به طریقه ی زیر حذف کردیم:

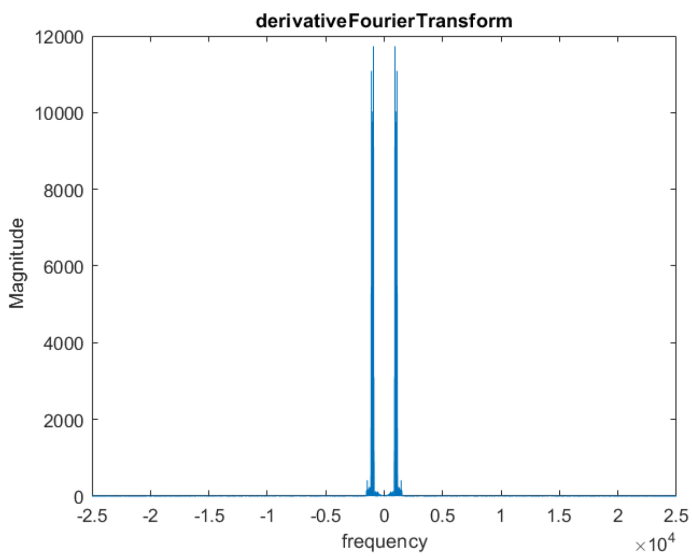
اولا مقدار abs سیگنال را محاسبه می کنیم.سپس میانگین این حاصل را از خودش کم می کنیم که در نهایت منتها می شود به اینکه مولفه ی dc سیگنال حذف می گردد.

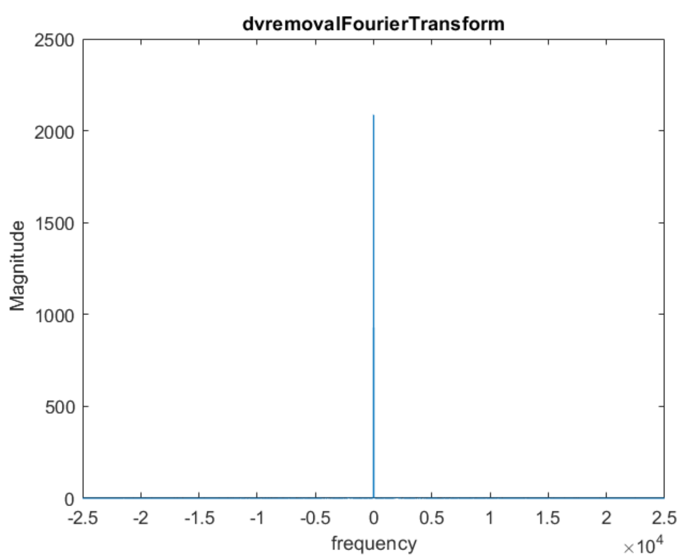
نمودارها:(تیتر نمودار ها مشخص می کند که هر کدام مربوط به کدام بخش از بلوک دیاگرام اند!)

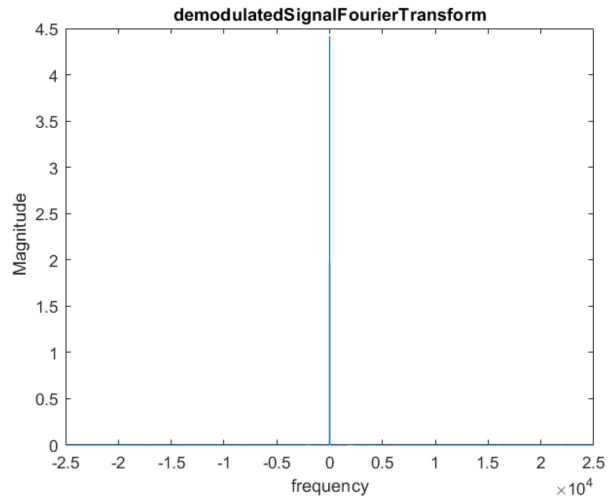


ادامه نمودار ها در ص بعد.

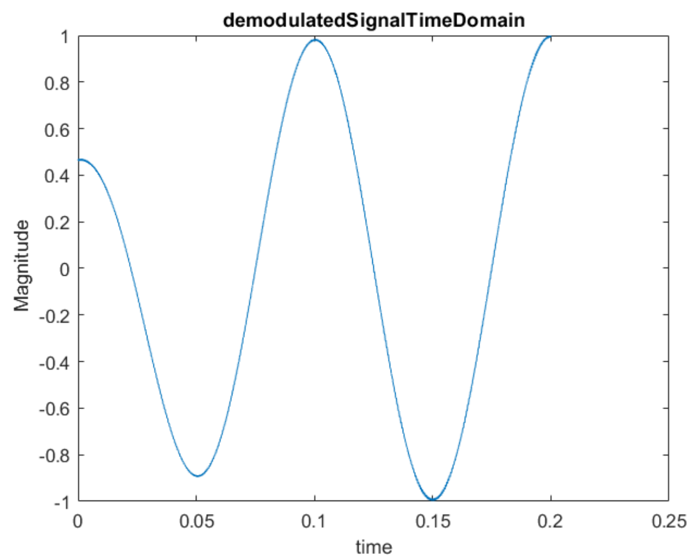

limitedAfterLPF
Magnitude
frequency
×10^4
2.5
2
1.5
1
0.5
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

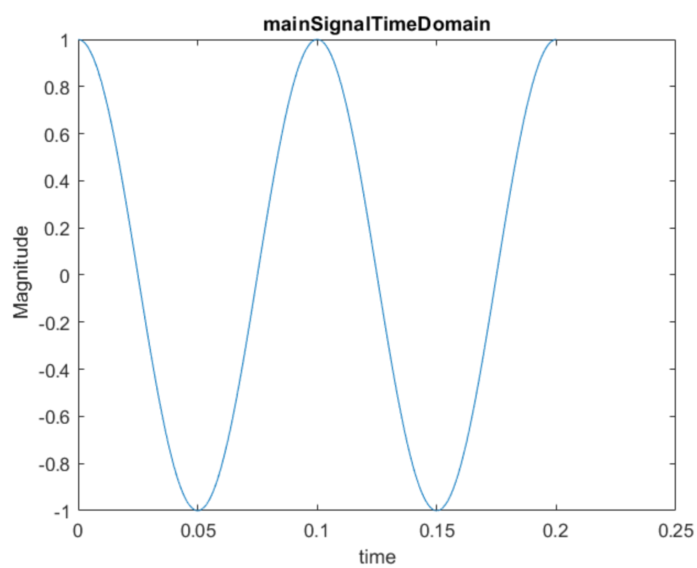

derivativeFourierTransform
Magnitude
frequency
×10^4
12000
10000
8000
6000
4000
2000
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5


dvremovalFourierTransform
Magnitude
frequency
×10^4
2500
2000
1500
1000
500
0
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5



مقایسه ی سیگنال بازیابی شده با سیگنال اصلی در حوزه ی زمان:





علت تفاوت نمودار ها: درواقع اینکه عبارت های اولیه به خوبی که ما انتظار داریم نمیباشد و این مساله را می توان با طبیعی بودن فیلتر ها توجیه کرد که چون ایده آل نیستن حاصل کانولوشن غیر ایده آل آنها مقادیر سیگنال در شروع را نتوانسته است به بهترین حالت ممکن انجام بدهد.

محاسبات MSE:

حالت اول: محاسبه ی MSE در کل دامنه ی سیگنال دمدوله شده

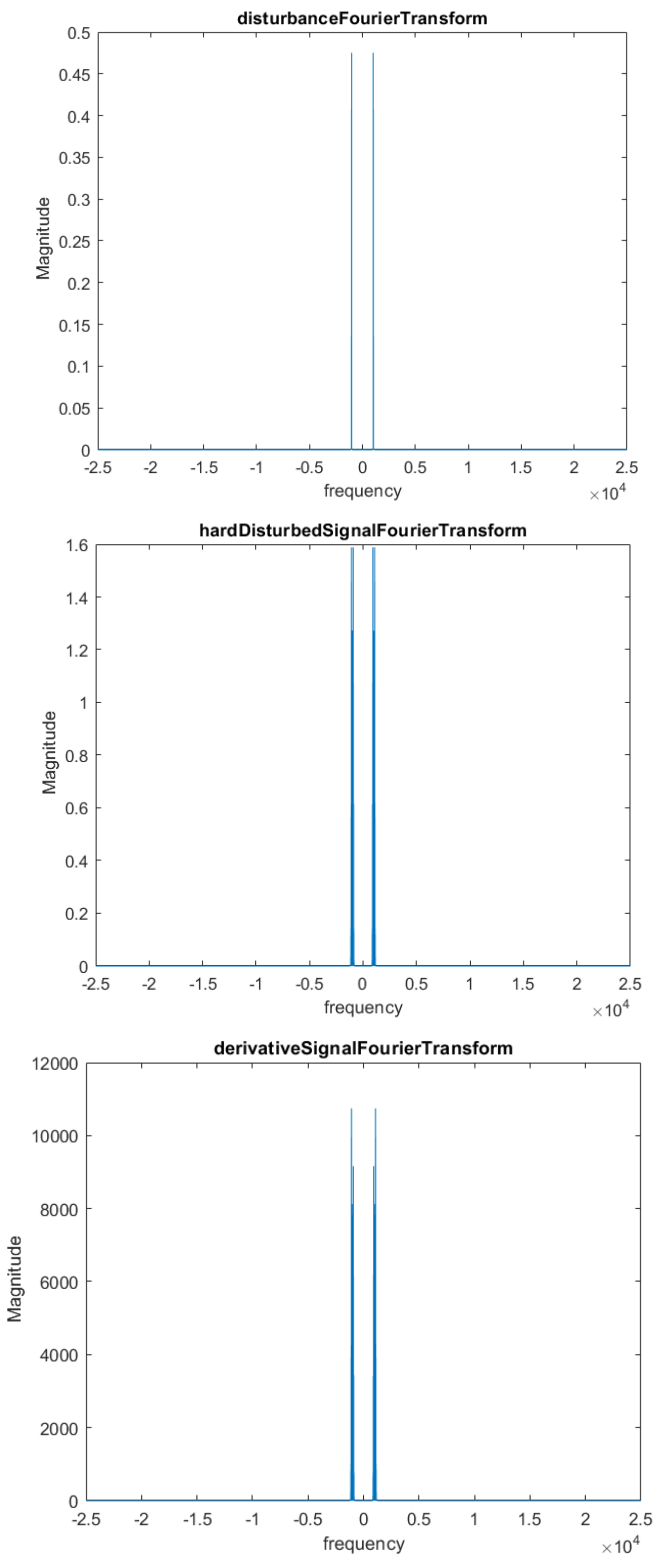
```
MSEerrorALLOfDomain = 4.3610e-04
```

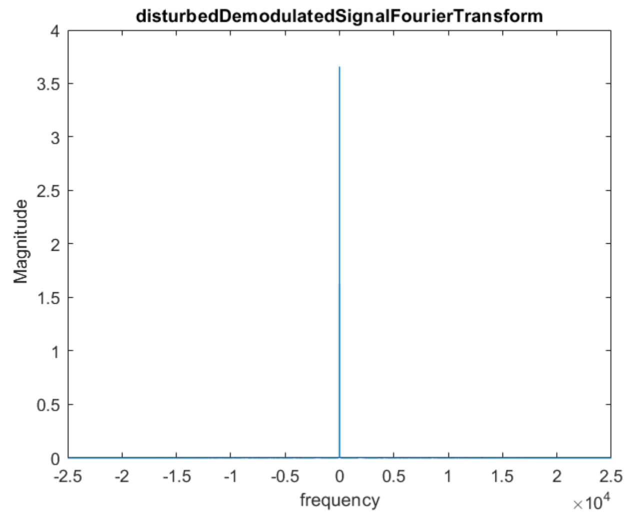
حالت دوم:محاسبه ی MSE در دامنه ی ای اندکی پس از شروع(که اثرات غیرایده آل بودن فیلترخودش راکمترنشان میدهد)

```
MSEerrorCorrectDomain = 4.5886e-05
```

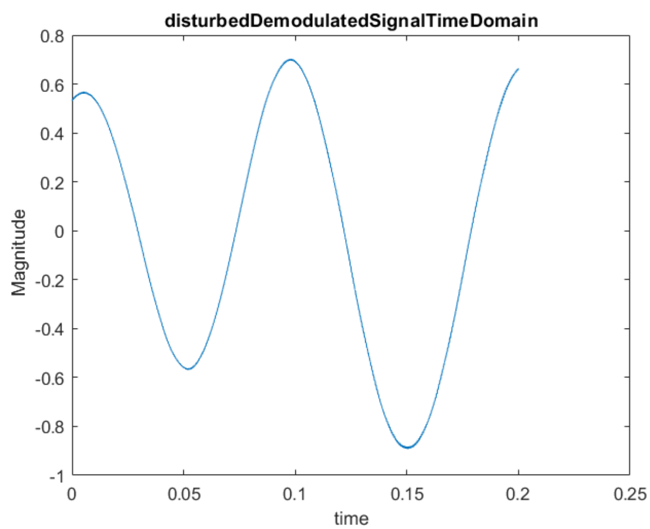
قسمت ت:

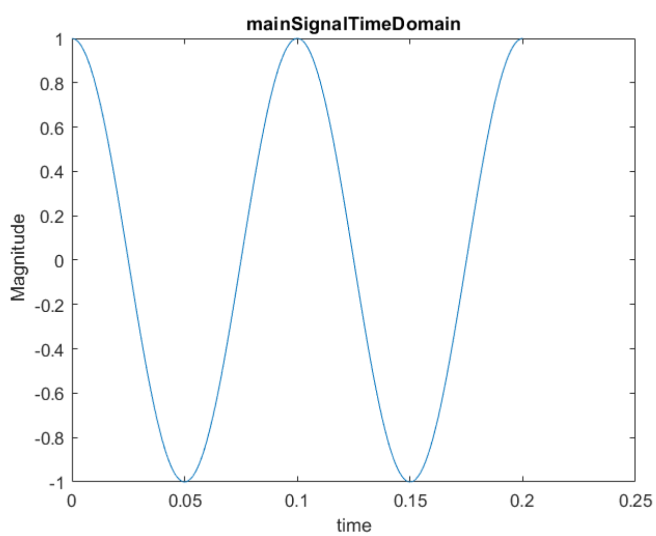
حالت اول: فقط اضافه شدن disturbance:





مقایسه ی سیگنال بازیابی شده با سیگنال اصلی در حوزه ی زمان:





علت تفاوت نمودار ها: درواقع اینکه عبارت های اولیه به خوبی که ما انتظار داریم نمیباشد و این مساله را می توان با طبیعی بودن فیلتر ها توجیه کرد که چون ایده آل نیستن حاصل کانولوشن غیر ایده آل آنها مقادیر سیگنال در شروع را نتوانسته است به بهترین حالت ممکن انجام بدهد.

محاسبات MSE:

حالت اول: محاسبه ی MSE در کل دامنه ی سیگنال دمدوله شده

```
MSEerrorALLOfDomain = 0.0541
```

حالت دوم:محاسبه ی MSE در دامنه ی ای اندکی پس از شروع(که اثرات غیرایده آل بودن فیلترخودش راکمترنشان میدهد)

```
MSEerrorCorrectDomain = 0.0494
```

حالت دوم: فقط اضافه شدن disturbance و هم چنین اعمال فیلتر های پیش تاکید و واتاکید:

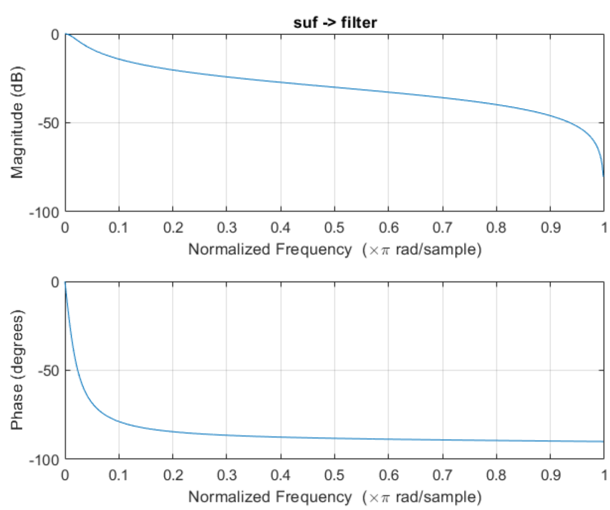
```
%  second part : part C + disturbance + pre/suf filters!!!
[b,a] = butter(1, 1000/(fs));
```

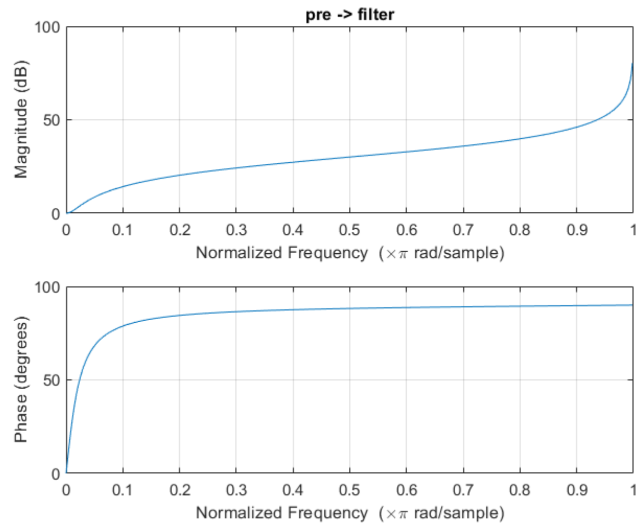
فیلترهای پیش تاکید و واتاکید:

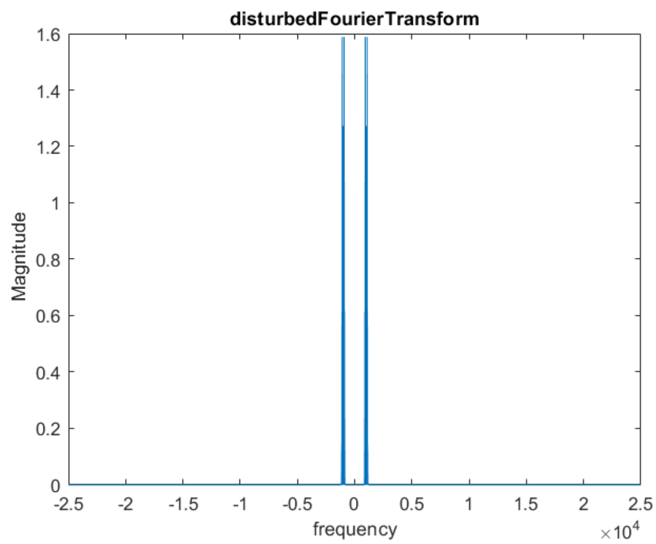
#باپرس و جو و صحبت! تصمیم گرفتم که فیلتر واتاکید را وارون فیلتر پیش تاکید درنظر گرفتم که اگر فیلتر واتاکید تضعیف کننده باشد فیلتر پیش تاکید تقویت کننده باشد.از انجا که مرتبه یک است فاصله بین باند عبور و باند توقف آن نسبتا بزرگ است برای همین آنرا به گونه ای تنظیم میکنیم که حتما مطمعن باشیم که در مرحله واتاکید نویز را تضعیف کنید،لذا فرکانس قطع را کوچیک می گیریم که چون فیلتر از مرتبه ی 1 است طوری رفتار کند که در نهایت فرکانس قطع واقعی آن به اندازه ی باشد تا فیلترینگ مطلوب ما را انجام بدهد.

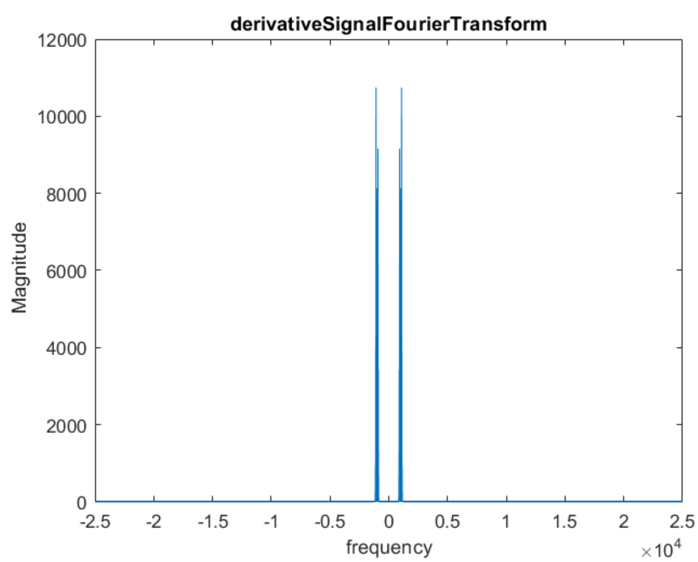
نمودار های فیلتر ها و بخش مدولاسیون/دمدولاسیون به کمک فیتلر های pre/suf:

توجه منظورم از suf fitler فیلتر واتاکید است و از pre filter فیلتر پیش تاکید است.

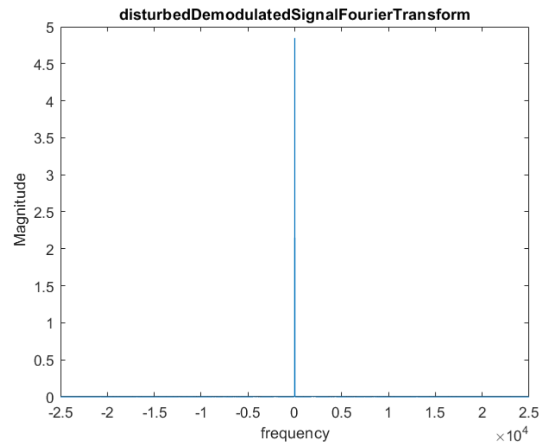


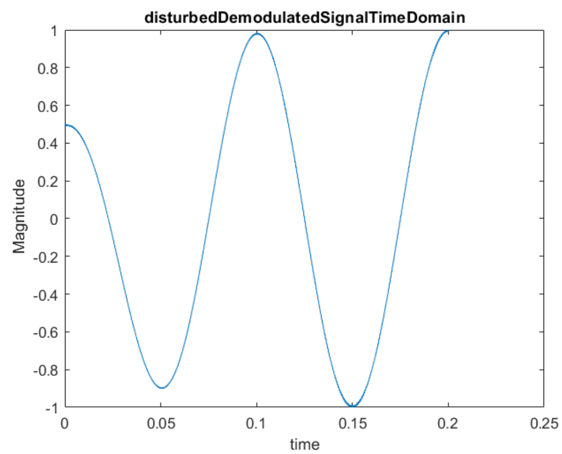

pre -> filter
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)

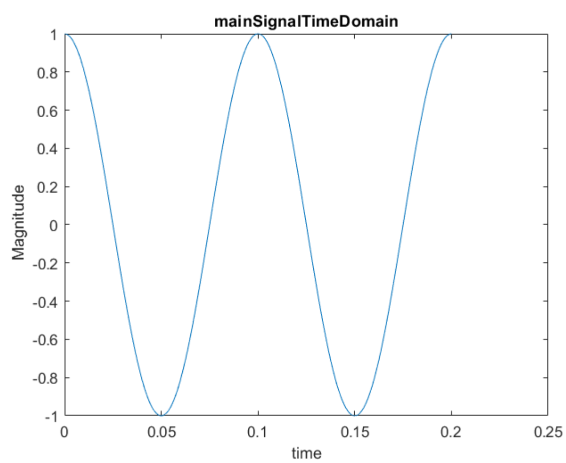

disturbedFourierTransform
Magnitude
frequency
×10^4


derivativeSignalFourierTransform
Magnitude
frequency
×10^4

مقایسه ی سیگنال بازیابی شده با سیگنال اصلی در حوزه ی زمان:







محاسبات MSE:

حالت اول: محاسبه ی MSE در کل دامنه ی سیگنال دمدوله شده

```
MSEerrorALLOfDomain = 3.8725e-04
```

حالت دوم:محاسبه ی MSE در دامنه ی ای اندکی پس از شروع(که اثرات غیرایده آل بودن فیلترخودش راکمترنشان میدهد)

```
MSEerrorCorrectDomain = 4.2797e-05
```