

بسم الله الرحمن الرحيم

درس سیستم های مخابراتی

استاد حمید بهروزی

تمرین سری ۲ متلب

امیرحسین رستمی ۹۶۱۰۱۶۳۵

گزارش تمرین سری ۲

دانشگاه صنعتی شریف

سوال اول:

قسمت الف: (محاسبات دستی)

ابتدا محاسبات دستی هر قسمت را با ذکر ریز جزییات انجام می دهیم:

Caveat: I'm using the normalization $\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-it\omega} dt$.

A cute way to to derive the Fourier transform of $f(t) = e^{-t^2}$ is the following trick: Since

$$f'(t) = -2te^{-t^2} = -2tf(t),$$

taking the Fourier transform of both sides will give us

$$i\omega \hat{f}(\omega) = -2i\hat{f}'(\omega).$$

Solving this differential equation for \hat{f} yields

$$\hat{f}(\omega) = Ce^{-\omega^2/4}$$

and plugging in $\omega = 0$ finally gives

$$C = \hat{f}(0) = \int_{-\infty}^{\infty} e^{-t^2} dt = \sqrt{\pi}.$$

I.e.

$$\hat{f}(\omega) = \sqrt{\pi}e^{-\omega^2/4}.$$

$$F\{x(t)\} = F\{e^{-\pi t^2}\} = e^{-\pi f^2} \rightarrow F\{e^{-t^2}\} = \sqrt{\pi} e^{-\frac{(2\pi f)^2}{4}}$$

$$F\{c(t)\} = \frac{\delta(f - fc) + \delta(f + fc)}{2}$$

$$v(t) = e^{-t^2} \times \cos(2\pi fct) \rightarrow F\{v(t)\} = F\{x(t)\} * F\{c(t)\} = \frac{\sqrt{\pi}}{2} \left(e^{-\frac{(2\pi(f-fc))^2}{4}} + e^{-\frac{(2\pi(f+fc))^2}{4}} \right)$$

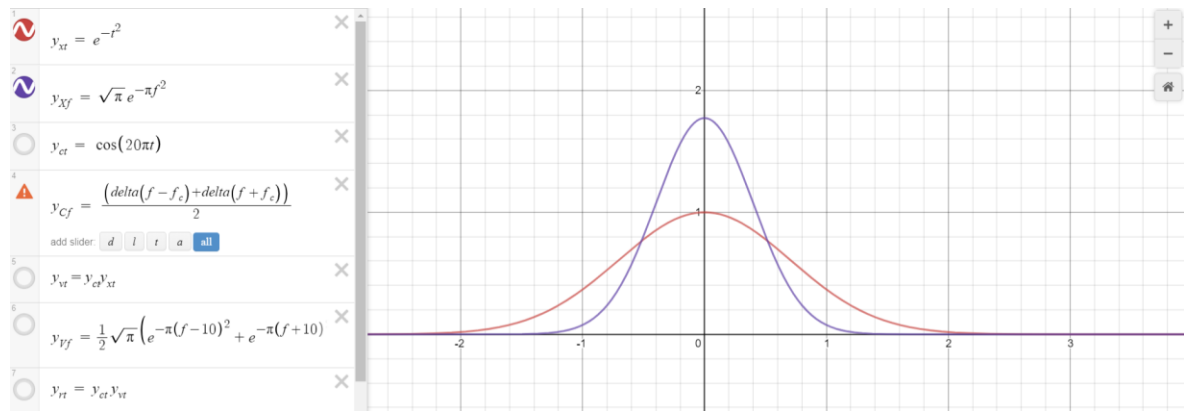
$$r(t) = e^{-t^2} \times \cos^2(2\pi fct) = e^{-t^2} \left(\frac{1 + \cos(4\pi fct)}{2} \right) \rightarrow F\{r(t)\} = \frac{\sqrt{\pi}}{4} \left(e^{-\frac{(2\pi(f-2fc))^2}{4}} + e^{-\frac{(2\pi(f+2fc))^2}{4}} + 2e^{-\frac{(2\pi f)^2}{4}} \right)$$

Lpf ایده آل می باشد لذا فرکانس های بالاتر از fc که 10 هرتز میباشد را حذف میکند سیگنال $e^{-\frac{(2\pi f)^2}{4}}$ نیز در بازه بزرگتر از 10 هرتز هنوز محتوای فرکانسی دارد ولی با توجه به سرعت دمپینگ $-\pi f^2$ میتوانیم حدودا از محتوای فرکانسی بالاتر از 10 هرتز برای آن صفر نظر کنیم (در شکل 8 نیز این ادعا به خوبی مشاهده میشود) برای دو عبارت دیگر نیز به طور مشابه میتوانیم بگوییم که برای $e^{-\frac{(2\pi(f-2fc))^2}{4}}$ برای فرکانس های کمتر از 10 هرتز محتوای ندارد و $e^{-\frac{(2\pi(f+2fc))^2}{4}}$ در فرکانس های بیشتر از - 10 هرتز حدودا محتوایی ندارد پس به صورت تقریبی میتوانیم بگوییم:

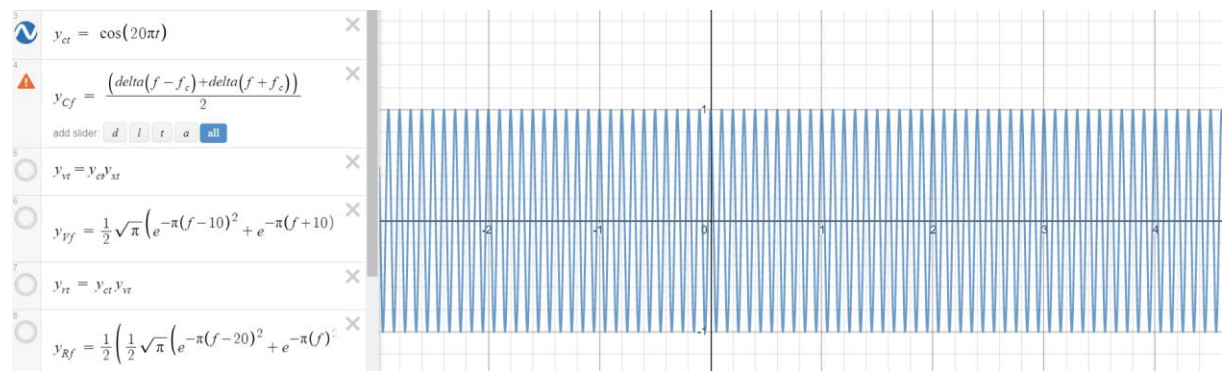
$$F\{d(t)\} \cong \frac{\sqrt{\pi}}{2} e^{-\frac{(2\pi f)^2}{4}} \rightarrow d(t) \cong \frac{e^{-t^2}}{2}$$

نمودار های ترسیمی به کمک نرم افزار Desmos :

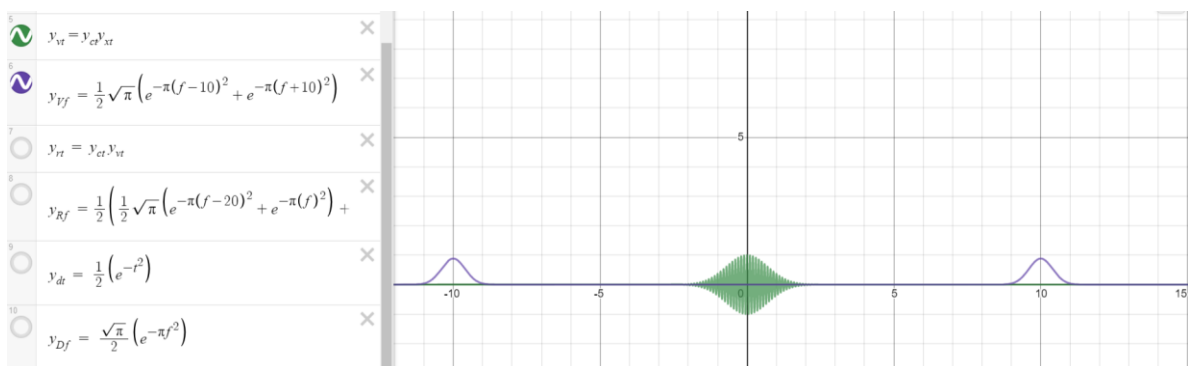
نمودار های $x(t)$ & $X(f)$:



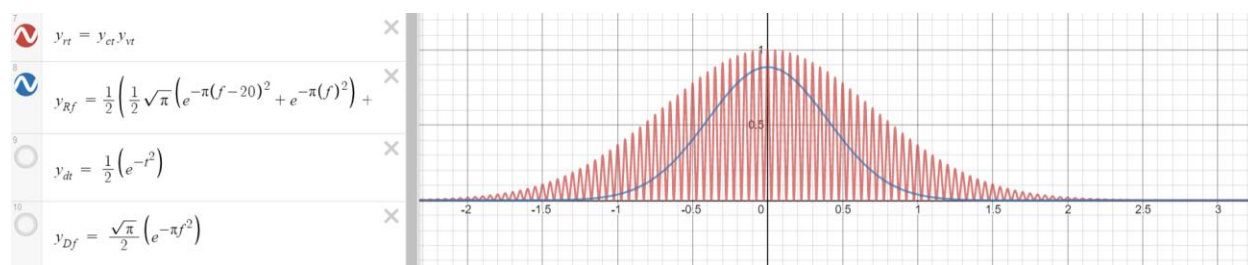
نمودار های $c(t)$ & $C(f)$:



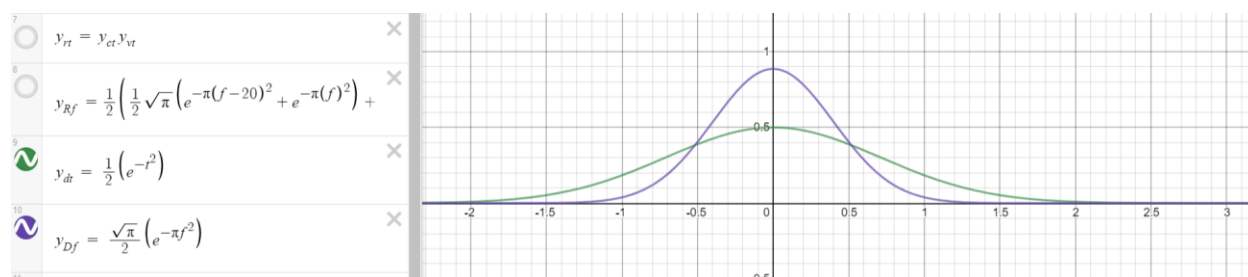
نمودار های $v(t)$ & $V(f)$:



نمودار های $r(t)$, $R(f)$:



نمودار های $d(t)$, $D(f)$:

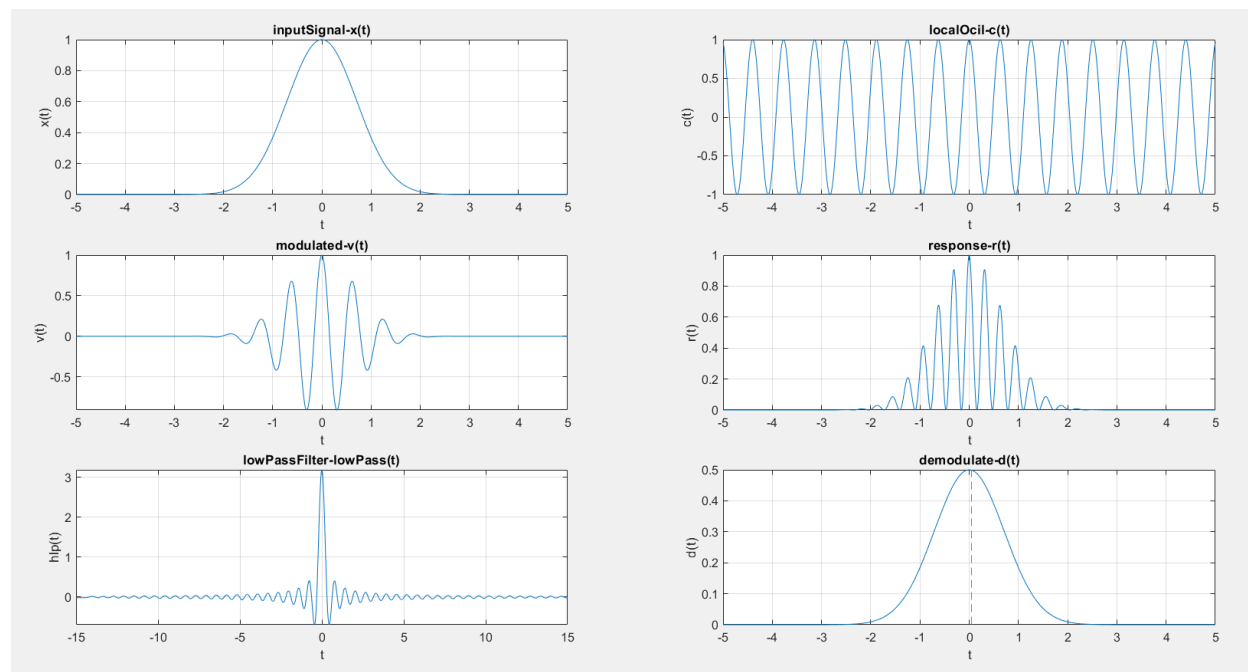


توجه کنید که نرم افزار desmos نمودار های ضربه دار را نمی کشد لذا ترسیم نمودار حوزه ی فوری ی کسینوس ممکن نبود اما رابطه ی اصلی آن هارا در دومین نمودار ذکر کرده ام تحت عنوان زیر:

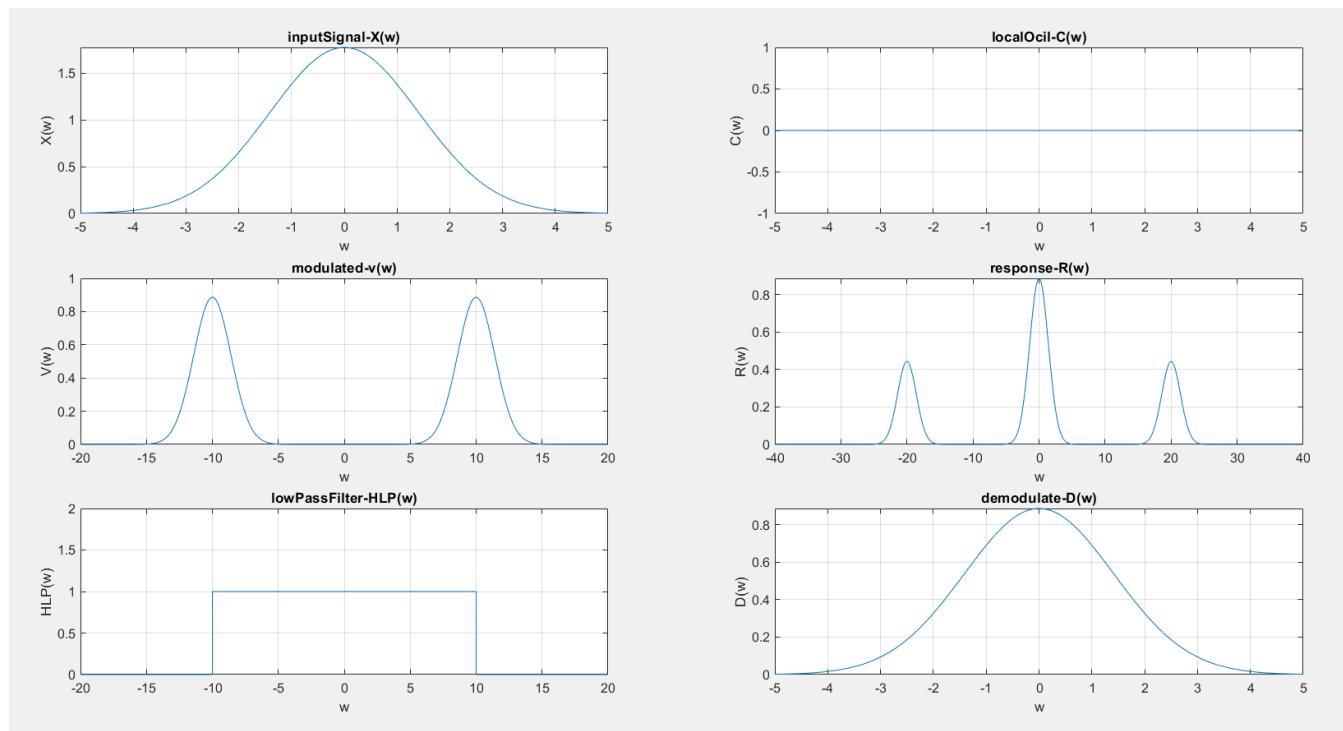
$$y_{Cf} = \frac{(\text{delta}(f - f_c) + \text{delta}(f + f_c))}{2}$$

add slider: ☐ d ☐ l ☐ t ☐ a ☒ all

حال برویم سراغ نمودار ها و شبیه سازی های متلب 😊 (حوزه ی زمان)



نمودار ها و شبیه سازی های متلب 😊 (نمودار های حوزه ی فرکانس)



همانطور که ملاحظه می کنید نتایج در شبیه سازی با symbolic Math toolbox کاملاً مشابه است با شبیه سازی انجام شده توسط نرم افزار Desmos است.

ذکر چند نکته مهم: (که در کد از آن ها استفاده شد).

1 - یکی از ضعف های متلب خود را در این بخش نشان می دهد که ما نمیتوانیم ضربه را به صورت fplot در آن رسم کنیم! ولی این برای ما مشکلی به وجود نمی آورد چون در بخش های بعد نه در رابطه زمانی و نه در فوریه ضربه نخواهیم داشت. فقط در شکل ها به این دقت داشته باشید که متلب فوریه را برحسب w رسم میکند و نه برحسب f بنابراین چون ما در حوزه ی فرکانس ترسیمات اول رو انجام دادیم نیاز شد تا من در کد زنی متلب ذکر کنم که کسینوس را به صورت $\cos(fc*t)$ به جای حالت $\cos(2\pi fct)$ تعریف کنم تا تبدیل فوریه گرفتی در حوزه ی فرکانس به ظاهر انجام شود. (مهم شکل درستی نمودار هاست و اهمیت ندارد که scale محور فرکانس، f باشد یا w).

هرجا تی ای محترم اختلاف scale دید بداند که به خاطر تفاوت محاسبات در فرکانس زاویه ای w و فرکانس عادی f است.

2 - اما فیلتر به صورت Symbolic در متلب وجود ندارد. برای تحقق این فیلتر سیگنال $d(t)$ را در حوزه فرکانس در پاسخ فرکانسی فیلتر پایین‌گذر ضرب می‌کنیم. ابتدا پاسخ ضربه فیلتر را در حوزه زمان به صورت $10 \cdot \text{sinc}(10t)$ تعریف می‌کنیم. سپس از آن تبدیل فوری می‌گیریم تا پاسخ فرکانسی فیلتر پایین‌گذر یعنی $\text{rect}(f/10)$ بدست آید. سپس آنرا در تبدیل فوری سیگنال $r(t)$ ضرب می‌کنیم تا تبدیل فوری سیگنال $d(t)$ یعنی همان $D(f)$ بدست آید. حال برای محاسبه $d(t)$ در حوزه زمان، از تعریف تبدیل فوری و وارون اش استفاده می‌کنیم.

$$d(t) = \int_{-\infty}^{\infty} D(f) e^{j2\pi f t} df$$

همان‌طور که مشاهده می‌شود سیگنال‌ها در حوزه زمان و فرکانس، با نمودارهایی که به صورت تئوری انتظار داشتیم همخوانی دارد و می‌توان گفت که نتایج بدست آمده بر چیزی که در تحلیل تئوری بدست آوردیم منطبق است.

هرچا تی ای محترم اختلاف scale دید بداند که به خاطر تفاوت محاسبات در فرکانس زاویه ای w و فرکانس عادی f است.

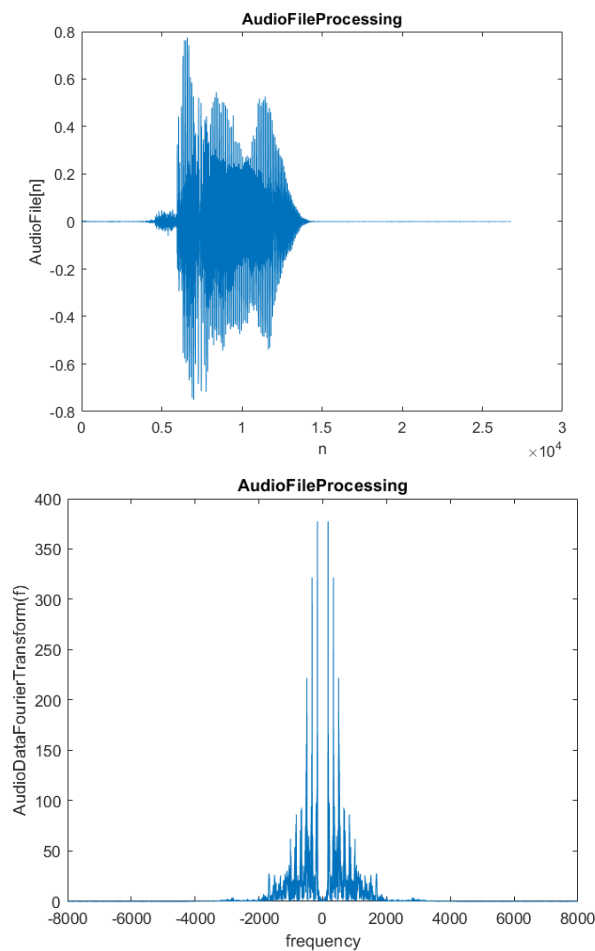
سوال دوم:

سوال 2) ارسال پیام با مدلاسیون دامنه:

الف) ابتدا تابع `standardFourierTransform` را معرفی می کنیم که تبدیل فوریه ی استاندارد از سیگنال ما می گیرد، این تابع، در خروجی اش تبدیل فوریه ی سیگنال را به همراه دامنه فرکانسی استاندارد شده ی آن را برمی گرداند.

```
function [out,standardDomain] = standardFourierTransform(signal,fs)
n = length(signal);
fourier=fft(signal,n);
fourier = fourier/fs;
out = fftshift(fourier);
%% standard Domain -> [-pi,pi]
domain = linspace(-pi,pi,length(out));
standardDomain = domain*fs/(2*pi);
end
```

با دستور `audioread` سیگنال را میخوانیم که f_s آن برابر 16 کیلوهرتز می باشد ابتدا در حوزه ی زمان سیگنال را رسم می کنیم. و سپس به کمک تابع فوق تبدیل فوریه آن را در دامنه فرکانسی استانداردش رسم می کنیم.



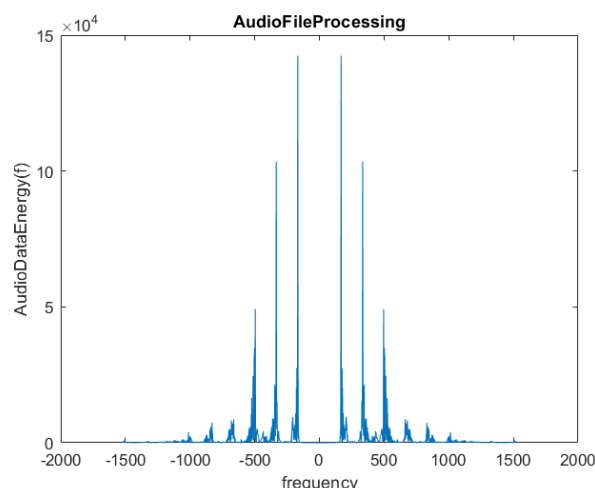
حال می‌خواهیم با تعریف ارائه شده پهنای باند سیگنال را پیدا کنیم، ابتدا برای این امر از سیگنال با دستور FFT تبدیل فوریه می‌گیریم و یک محدوده برای فرکانس درست می‌کنیم که از صفر تا f_s را به تعداد نقاط فوریه تقسیم می‌کند. اندازه تبدیل فوریه را تقسیم بر f_s کرده و آنرا در مزدوج مختلط خود ضرب می‌کنیم تا چگالی توان به دست بیاید ولی از آنجا 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 f_s آنرا بکنیم.

ما طیف دوطرفه در اختیار داریم برای جمع هر فرکانس در پهنای باند باید به طور متقارن جلو برویم برای همین از آنجا که می‌دانیم 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) می‌پردازیم و در هر مرحله به مقایسه ی انرژی موجود در همسایگی در نظر گرفته شده با 99 درصد انرژی کل که با جاروب روی کل انرژی حوزه ی فرکانس به دست آمده است می‌کنیم و در صورت عدم برابری پله پله این freqIterator را افزایش می‌دهیم تا سرانجام به 99 درصد انرژی کل برسیم و نتیجه نهایی برابر است با :

Signal bandwidth = 1525.1465

که مقدار آن به هرتز می‌باشد پس تا حدودای 1.5 کی محتوای فرکانسی دارد.

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



حذف شدن انرژی فرکانس های خارج از مقدار بیان شده : 1.52 کیلوهرتز.

ب) پهنای باند در اختیار ما 4 کیلو هرتز می‌باشد در صورتی که وقتی سیگنال ما به باند میانی منتقل می‌شود پهنای باند آن اگر بخواهیم به طور کامل منتقل کنیم برابر $2*BW$ می‌شود که یعنی در حدود 3.05 کیلو هرتز می‌شود که کمتر از پهنای باند کانال می‌باشد پس سیگنال را به صورت مدلاسیون double side band نیز می‌توانیم منتقل کنیم و خوب به صورت بدیهی بنابراین از مدلاسیون های دامنه single side band نیز می‌توان استفاده کرد. پس از همه AM, DSB, SC, USSB, LSSB, VSB می‌توان استفاده کرد. بدون اینکه مشکل خاصی رخ دهد. (این مساله را حتی بنده در تلگرام هم کامل به تی ای محترم آقای علی تفکر توضیح دادم).

ب) توجه کنید که با توجه به اینکه فرکانس حامل برابر 102 کیلو هرتز است طبق نایکویست داریم که حداقل rate نمونه برداری باید برابر 204 کیلو هرتز باشد (برای کسینوس) و چون داده های صورت فرکانس سمپلینگ 16 کیلو هرتز داشتند و باید برای انجام درست کار های پردازشی حتما فرکانس سمپلینگ کل پروسه با هم سینک باشد نیاز است تا حداقل $204/16 = 12.75$ برابر برای سیگنال صوت upSampling رخ بدهد، و لذا حداقل باید 13 مرتبه صوت را upSample کنیم و بنده برای جلوگیری از رخ داد هرگونه نقص احتمالی به خاطر نزدیک بودن بیش از حد 13 به حدکف upSampling مقدار آپسمپلینگ را برابر 14 را در نظر گرفتیم.

قضیه ی نایکویست:

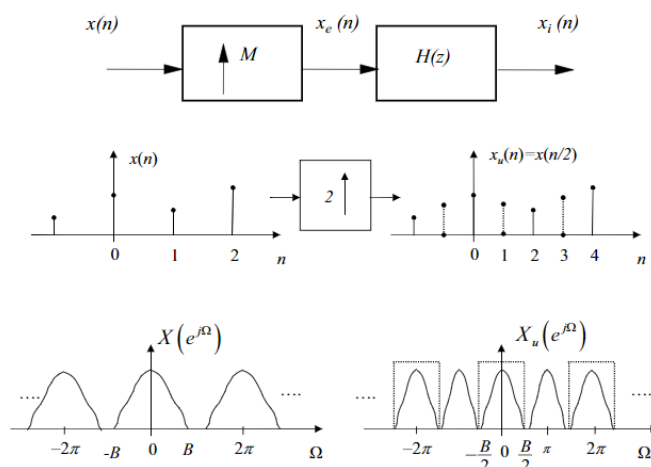
$$2 * BW < fs$$

اما از آنجا که می دانید انجام عملیات upSampling باعث قراردادن شیفت یافته های تبدیل فوری در تبدیل فوری اصلی باشد و لذا بایستی فیلتری طراحی کنیم که به اصطلاح یک نمونه پیرو از تبدیل فوری ی اصلی را استخراج کند. پس در نهایت بلوک upSampling از دو بخش زیر تشکیل شده است.

1- انجام عمل upSampling

2- فیلتر پایین گذر جهت عبور یک عدد از تناوب های فوری ی رخ داده شده.

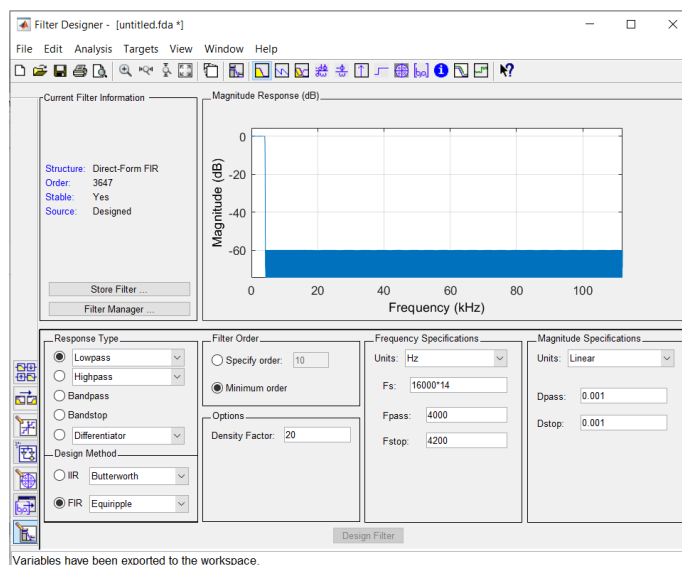
UPSAMPLING AND RECONSTRUCTION



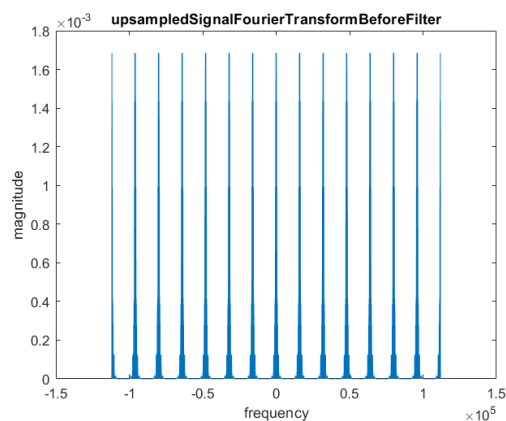
حال کافی است فیلتر مدنظر upSampling را جهت Reconstruction یکی از تبدیل فوری ها طراحی کنیم. داریم:

طبق نمودارهای بالا داریم که فیلتر ماباید حداکثر تا $B/2$ را عبور بدهد لذا داریم که $2*B < fs$ لذا $B < 8000$ و لذا $B/2 < 4000$ و لذا داریم که یک فیلتر پایین گذر تا فرکانس 4000 با ضریب نشستی 0.05 قرار می دهیم که همانطور که در شکل ملاحظه می کنید مشخصات این فیلتر آماده است.

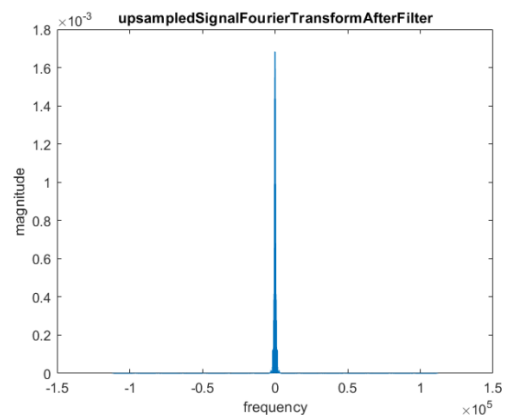
این فیلتر تحت نام Hd در فولدر سوال دوم ضمیمه شده است.



سیگنال پس از upSampling و پیش از عبور از فیلتر:

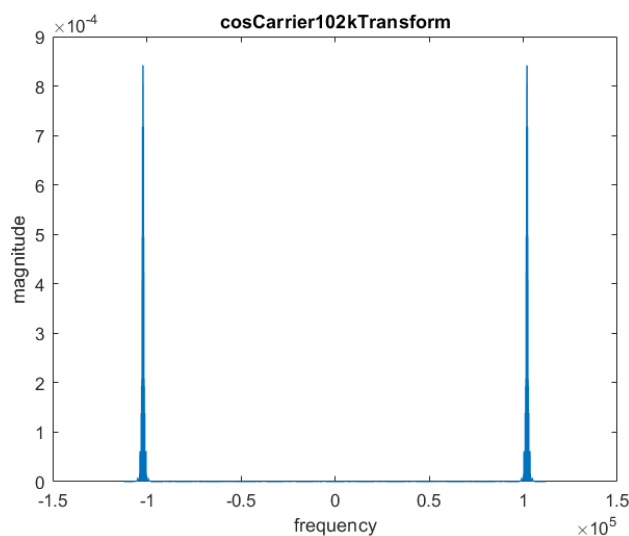


ولی بعد عبور از فیلتر به صورت زیر میشود که مطلوب ما است:

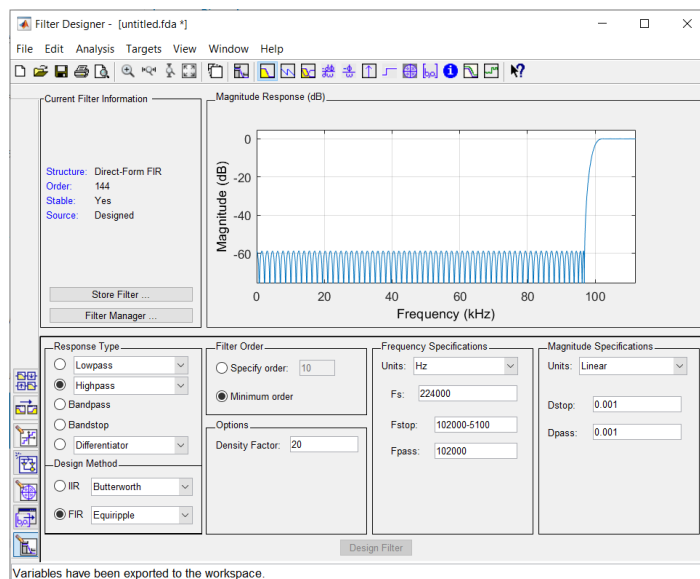


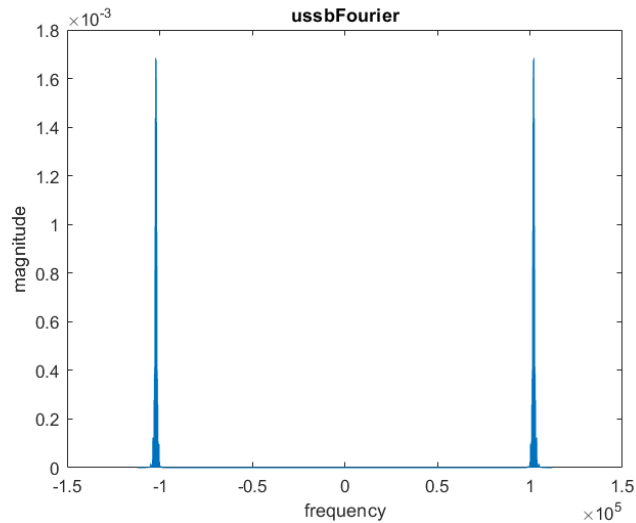
ت) برای این کار تنها کافی است که سیگنال اصلی خود را در کسینوس 102 کیلوهرتز ضرب کنیم و سپس سیگنال نهایی را از یک highPassFilter عبور بدهیم که از فرکانس $f_c = 102\text{kHz}$ به بعد را عبور می دهد، این کار ما عینا USSB را تولید می کند.

این فیلتر دقیقاً کار ussb را برای ما انجام میدهد در صورتی که فیلتر بالاگذر ما ایده ال یا باند گذر آن به صورت خوبی باشد یه کسینوسی با فرکانس نمونه برداری تعیین شده در قسمت قبل (فرکانس نمونه برداری تمامی سیگنال ها باید برابر باشد) تولید کرده و در سیگنال ضرب میکنیم طیف آن به صورت شکل زیر میشود: (طیف carrier)



حال از فیلتر طراحی شده در فرکانس 102 کیلوهرتز عبور می دهیم (فیلتر طراحی شده به شرح زیر است)

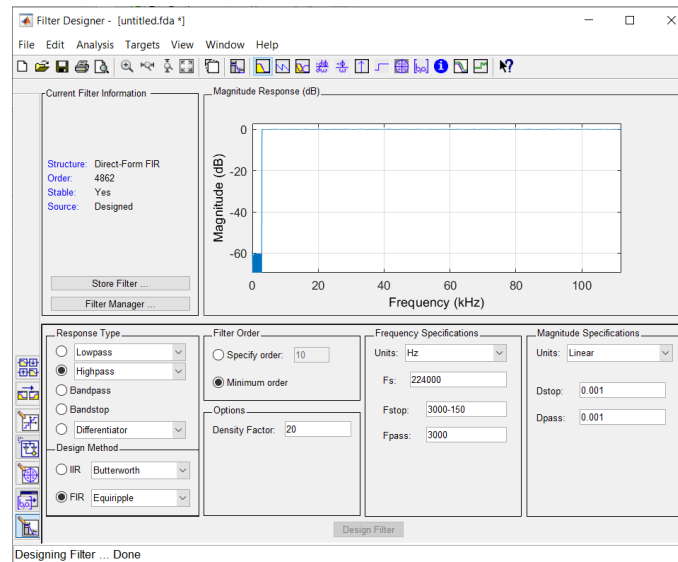




اما همانطور که ملاحظه می کنید انگار نه انگار که USSB است و اما چرا با اینکه مقداری تضعیف شده است فرکانس های پایین تر ولی اصلا نزدیک به چیزی که ما انتظار داشته ایم نمیباشد!! علت این است که در این حالت به قدر پهنای باند گذر آن زیاد است که فقط مقداری SingleSide کناری را تضعیف می کند و آن را حذف نمی کند. توجه کنید که باندگذر ما حدودا 5.1 کیلوهرتز است که تمام پهنای باند یک طرف سیگنال ما را که باندی حدودا برابر 1.52 کیلوهرتز دارد را کامل در بر میگیرد و لذا مدولاسیون ما به درستی انجام نشده است.

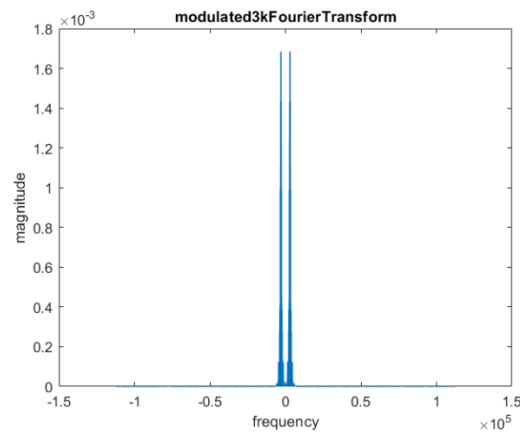
ث) حال با گام های کوچک تر همان کاری که در بخش ت انجام داده ایم را انجام می دهیم در اینصورت باند گذر های ظریف تری میتوان داشت که بهتر کار دلخواه ما را انجام بدهد. (توجه کنید وقتی فرکانس های کسینوس های ما پله ای افزایش پیدا کند در اولین قدم ها که باندگذر فیلترهای ما تنگ تر است) چون طبیعتا فرکانس های مرکزی کسینوس ها کوچک است) عملیات فیلترینگ قضیه به خوبی انجام می شود و لذا پاسخ به حالت ایده آل نزدیک تر خواهد بود. با توجه به اینکه گفته ی سوال ذکر کرده که ما زیر 150 هرتز داده ی مهم نداریم یعنی در ماکزیمم فرکانس اولین کریر باید حداکثر پهنای باند گذر ما 150 هرتز باشد که این مصادف است با داشتن 3 کیلوهرتز فرکانس مرکزی لذا اولین کسینوس ما 3 کیلوهرتز فرکانس کریر دارد. بعد با کریر 27 کیلو کیلوهرتز و در انتها به با کریر 72 کیلو به فرکانس مرکزی نهایی برابر با $72 + 27 + 3 = 102$ کیلوهرتز منتقل میکنیم. (توجه کنید که بر اثر انجام آزمایش های متعدد 3 پله در نظر گرفتیم تا در نهایت MSE ما مینیمم شود توجه شود که ما مکرر این تست را انجام دادیم و نسبت به حالت 2 تایی بنا به همان دلایل گفته شده حالت 3 کریر بهتر عمل می کند و ما کریر های سه تایی 3 کیلو، 27 کیلو و 72 کیلو در نظر گرفتیم).

فیلتر بالاگذر در 3 کیلوهرتز به صورت شکل زیر میشود:

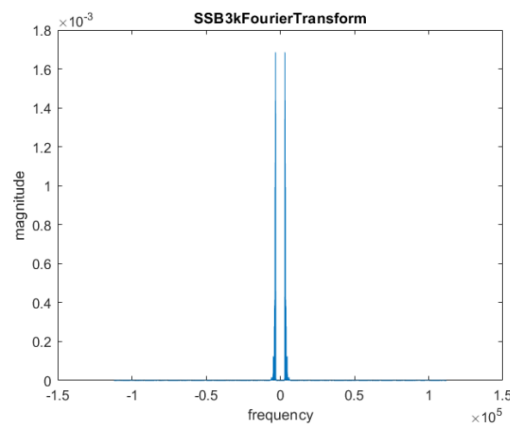


این فیلتر با نام first3kHighPassFilter در فولدر سوال دوم آورده شده است.

سیگنال را ضرب در کریر با فرکانس 3 کیلو هرتز میکنیم و به شکل زیر می شود:



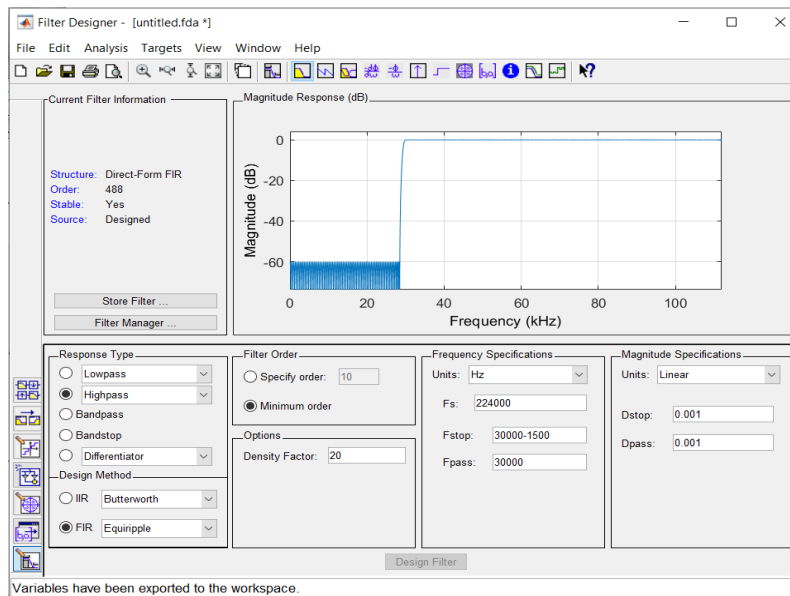
حال از فیلتر آنرا عبور میدهیم که به شکل زیر میشود:



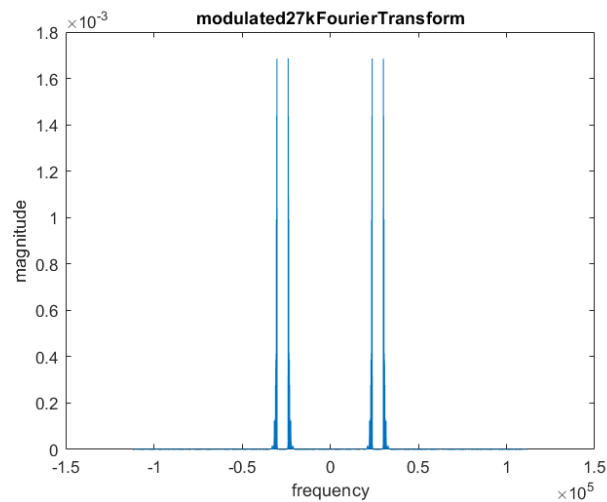
همانطور که مشخص است داریم که به خوبی فیلتر عملیات فیلترینگ را انجام داده است.

حال برویم سراغ کریر بعدی ...

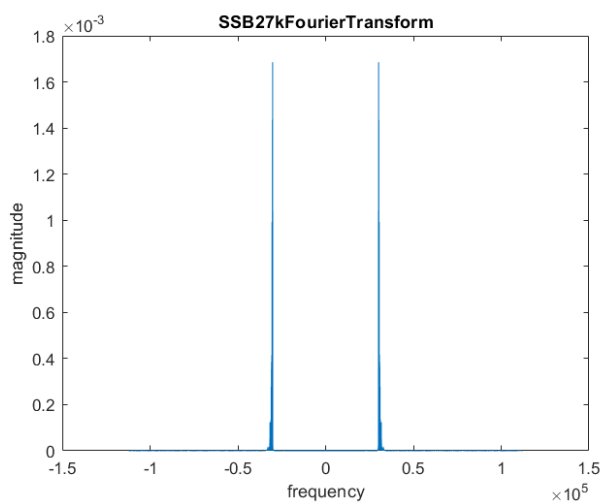
ابتدا مطابق با مرحله ی قبل فیلتر زیر را طراحی می کنیم.



حال سیگنال حاصل از مرحله قبل را در کریر ضرب میکنیم و داریم که:

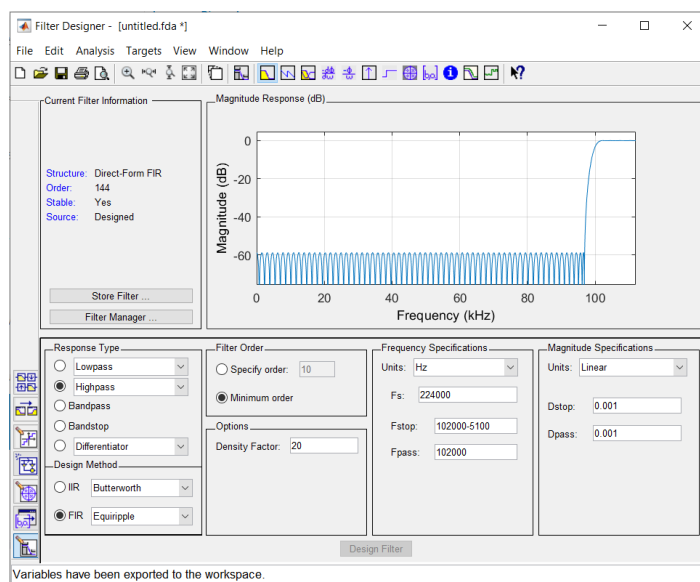


و بعد عبور از فیلتر:

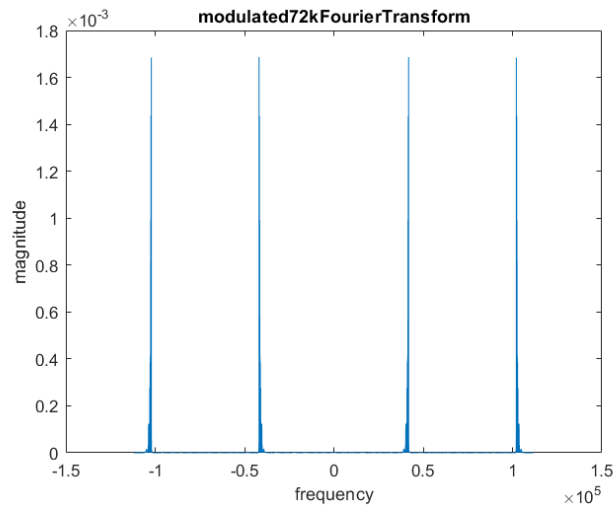


خب همانطور که ملاحظه می کنید کار تا اینجا به خوبی پیش رفته است، حال به آخرین کیر و رسیدن به فرکانس مرکزی 102 کیلوهرتز می پردازیم.

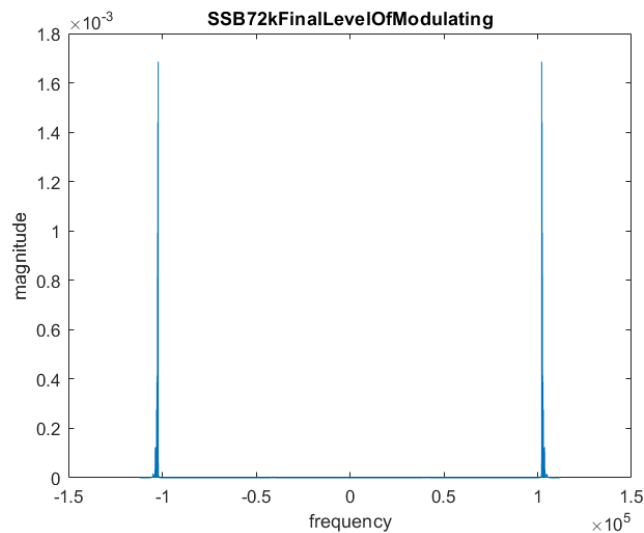
ابتدا به فیلتر طراحی شده ی این قسمت می پردازیم: (توجه کنید این فیلتر مشابه فیلتر بخش اول برای 102 کیلوهرتز است)



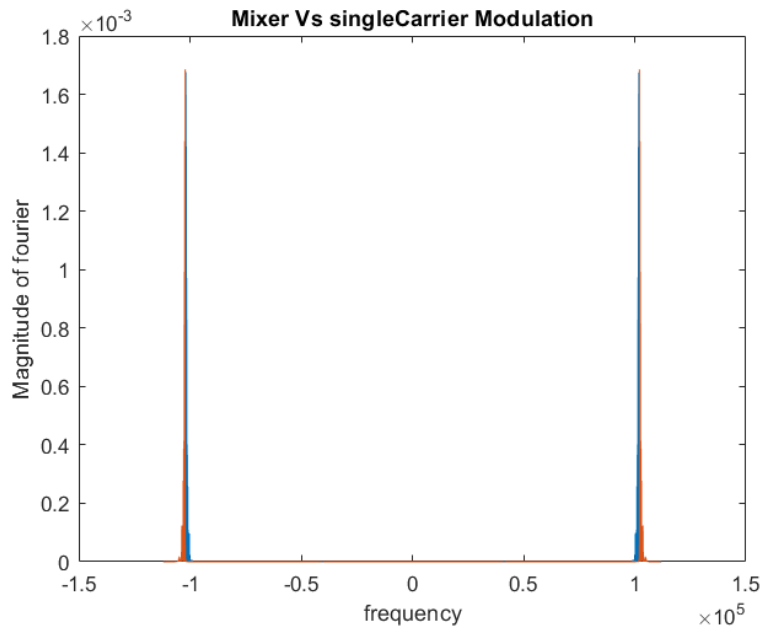
حال پیش از عبور از فیلتر بالا داریم که سیگنال به شرح است:



حال پس از عبور از فیلتر داریم که سیگنال در حوزه ی فرکانس به شرح زیر می گردد:



از آنجا که من فایل را در غالب LiveScript در اختیارتون گذاشتم کافیہ با چرخاندن کرسر موس به راحتی زوم کنید و ببینید که در این حالت نسبت به حالت قبلی با کیفیت فوق العاده ای محتوای فرکانس به صورت Single Side ذخیره شده است. به عنوان مثال پس از زوم کردن در نمودار فوق به شکل زیر می رسم که صحه ای بر درستی مدولاسیون ماست.



نارنجی : مدولاتور میکسری

آبی : مدولاتور تک carrier

همانطور که ملاحظه می کنید داریم که در حالت مدولاسیون چند مرحله ای، Single Side سیگنال درست تر و اصولی تر حذف می شود و مدولاسیون ما درست تر خواهد بود.

آبی رنگ حاصل از روش ت و نارنجی برای روش ت که به خوبی مشاهده می شود در روش ت به صورت خیلی خوبی مدولاسیون انجام شده در حالی که در روش ت تنها در فرکانس های کمتر از 102 کیلوهرتز تضعیف رخ داده و نه اینکه یک مدولاسیون USSB کامل باشد.

ج) اثبات دستی آن به صورت زیر می باشد:

$$f\{x_h(t)\} = -j\operatorname{sgn}(f)H(f) \rightarrow f\{jx_h(t) + x(t)\} = f\{x_+(t)\} = 2X(f)U(f)$$

به طور مشابه:

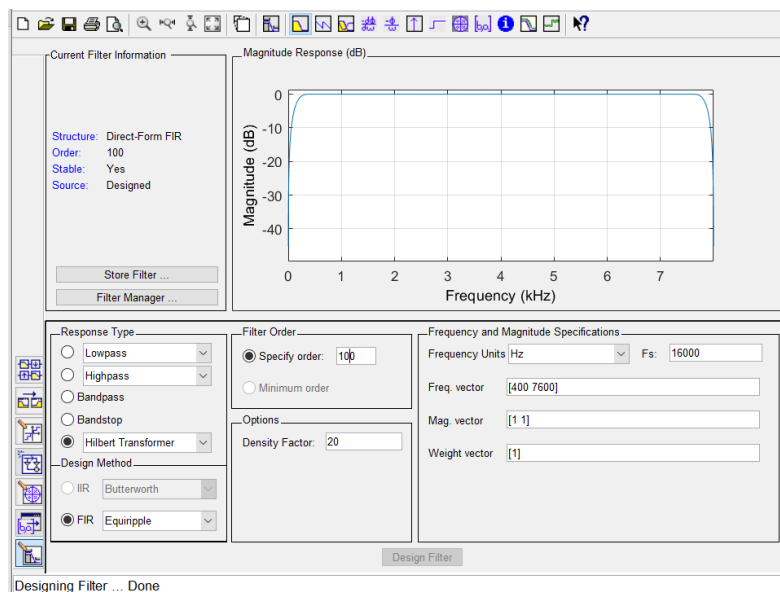
$$f\{-jx_h(t) + x(t)\} = f\{x_-(t)\} = 2X(f)U(-f)$$

$$\begin{aligned} USSB &= X_+(f - fc) + X_-(f + fc) = x_+(t)e^{j2\pi fct} + x_-(t)e^{-j2\pi fct} \\ &\rightarrow \frac{1}{2}(x(t)(e^{j2\pi fct} + e^{-j2\pi fct}) + x_h(t)(je^{j2\pi fct} - je^{-j2\pi fct})) \rightarrow USSB \\ &= x(t)\cos(2\pi fct) - x_h(t)\sin(2\pi fct) \end{aligned}$$

پس اثبات شد (در جمع کننده باید علامت منفی قرار داده میشد)

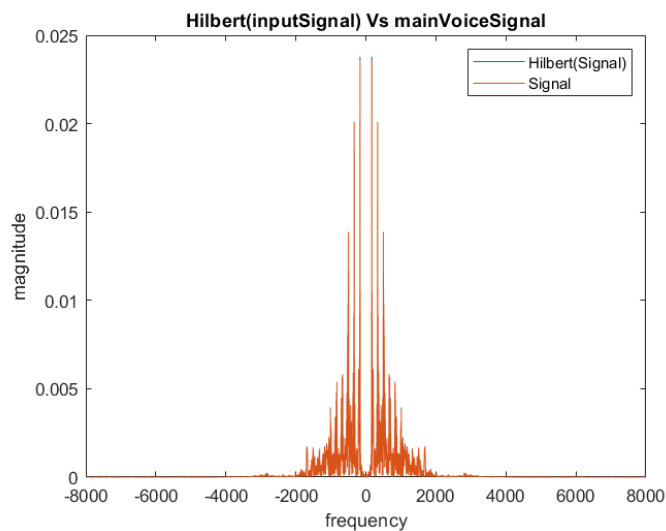
این سوال را به دوروش حل میکنیم در ابتدا می آیم و یک فیلتر هیلبرت توسط FilterDesignerTool طراحی می کنیم در روش دوم از دستور Hilbert استفاده میکنیم که تبدیل هیلبرت ایده آل میگیرد استفاده میکنم. (توجه کنید که تبدیل هیلبرت به دست آمده از روش FilterDesignerTool از دقت کمتری نسبت به تابع هیلبرت ایده آل متلب دارد)

ابتدا با فیلتر هیلبرت واقعی که آنرا به صورت زیر طراحی کرده ایم:



این فیلتر با نام Hilbert در فولدر سوال دوم ضمیمه شده است. (توجه کنید در مرحله اول که ما در سیگنال وویس ضرب می کنیم یکسان بودن سمپلینگمون به معنای برابری با نرخ سمپلینگ وویس که برابر 16000 است می باشد).

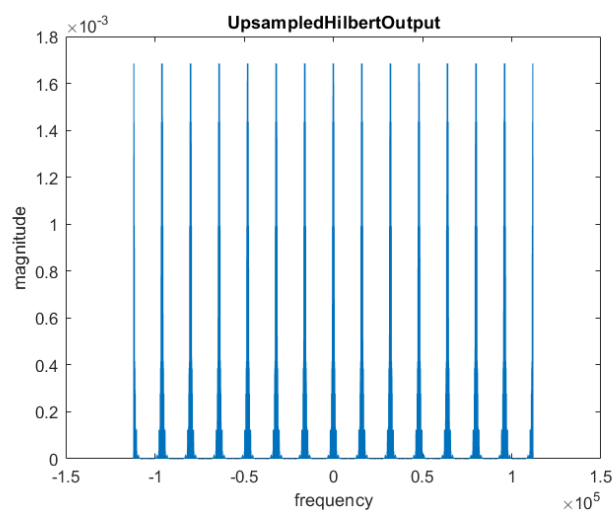
طبق شکل داده شده در تمرین جلو می رویم:



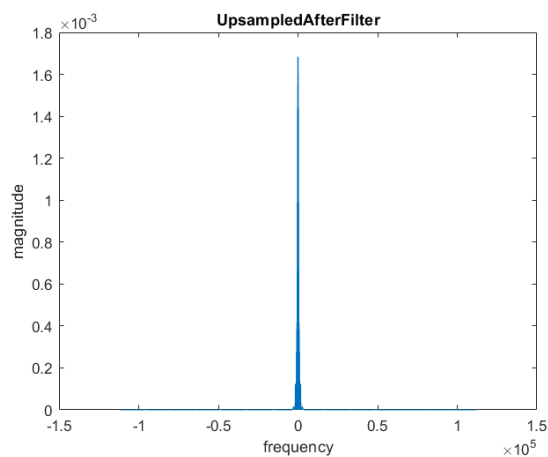
خطای این دو روش برابر است با:

error = 2.0448e-10

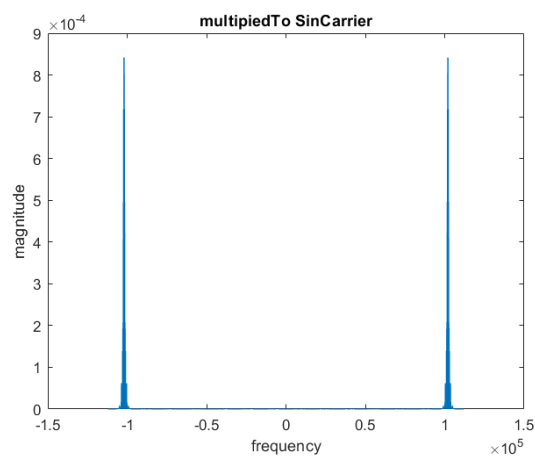
لذا داریم که اندازه ی سیگنال ورودی تا حد خوبی حفظ شده است. از آنجا که پس از عبور از فیلتر هیلبرت نیاز است تا در سینوس ضرب کنیم لذا نیاز است که نرخ نمونه برداری سیگنال ها برابر باشد لذا مجدد به همان upSampling عی که در قسمت های قبلی این سوال داشتیم نیاز خواهیم داشت، حال داریم که سیگنال خروجی هیلبرت پس از upSampling به صورت زیر در خواهد آمد:



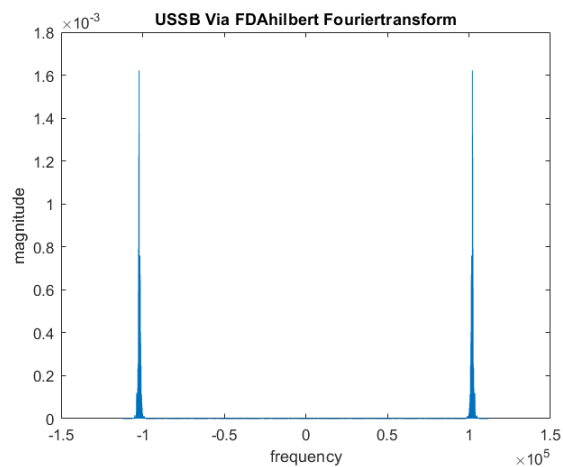
پس از عبور از فیلتر داریم که سیگنال به شکل زیر خواهد بود:



پس از ضرب در کریر سینوسی (سیگنال $x_2(t)$ طبق گفته ی شکل سوال در حوزه ی فرکانس به شرح زیر است:

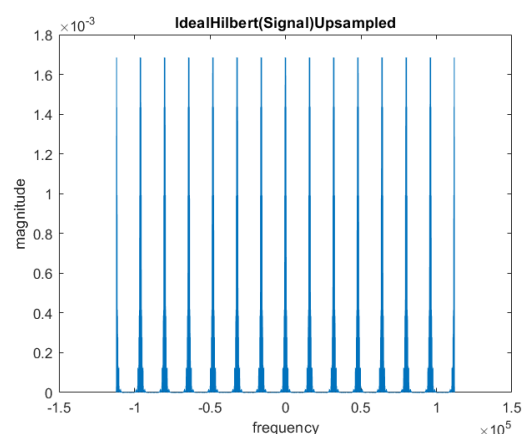


حال چون ما به دنبال SSB ایم سیگنال حاصل از ردیف پایین را از سیگنال حاصل از ردیف بالا کم می کنیم و داریم که در نهایت تبدیل فوریه سیگنال به دست آمده به شرح زیر است:

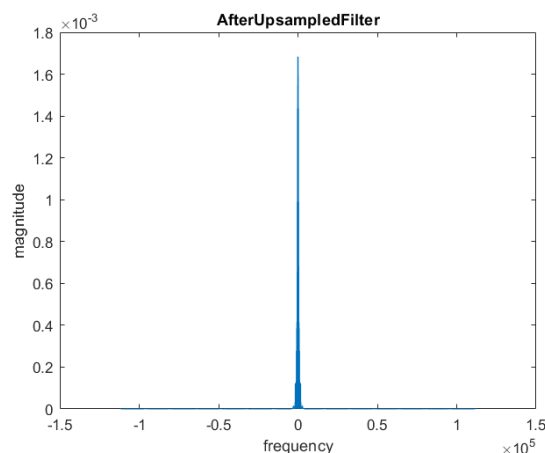


اگر کمی در نمودار ها که در فرمت liveScript موجود اند زوم کنید خواهید یافت که فرکانس های کمتر از 102 هرتز تضعیف شده اند ولی چون فیلتر هیلبرت ایده ال نم یباشد نتوانسته است که به طور کامل حذف کند توجه کنید که تمام شرایط پهنای باندی رعایت شده است (کافی است صحت این رخداد هارا با زوم کردن در نمودار بیابید اما به علت دقت پایین فیلتر هیلبرت طراحی شده با FDA tool داریم که سیگنال خیلی SSB مدوله نشده است و بیشتر به DSB شبیه است).

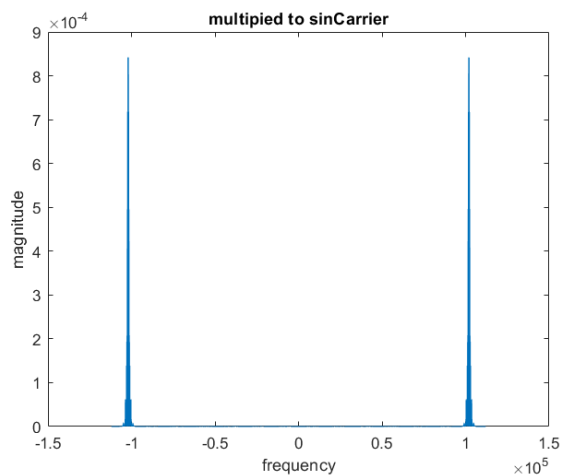
نکته مهم: در متلب فیلتر هیلبرت به صورت ایده آل وجود دارد و حال می خواهیم یکبار این محاسبات را با هیلبرت ایده آل نیز انجام بدهیم. چون توضیحات مشابه قسمت قبل است صرفا به آوردن نمودار ها و اندک توضیح راجع به هر کدام اکتفا می کنیم.



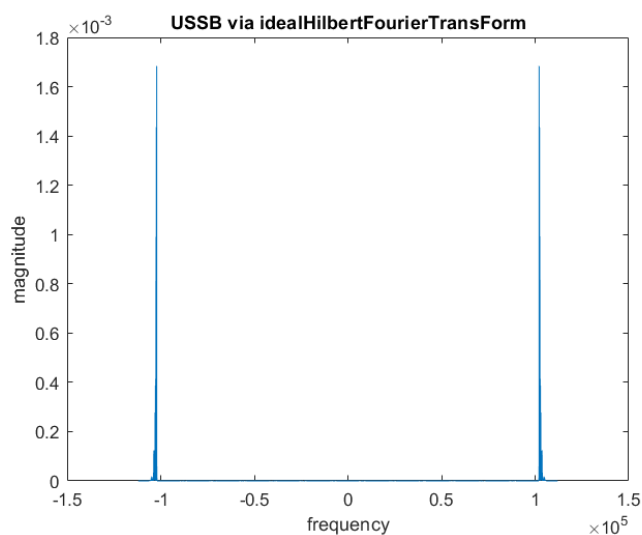
پس از فیلتر داریم:



پس از ضرب در کریر سینوسی داریم که:



و در نهایت امر داریم که سیگنال مدوله شده با هیلبرت ایده آل به شکل زیر درآمده است.



از آنجا که من فایل را در غالب LiveScript در اختیارتون گذاشتم کافیه با چرخاندن کرسر موس به راحتی زوم کنید و ببینید که در این حالت نسبت به حالت قبلی با کیفیت فوق العاده ای محتوای فرکانس به صورت Single Side ذخیره شده است. و این هویداگر اثر هیلبرت ایده آل است که به طرز بی نظیری به سیگنال USSB ما را می رساند.

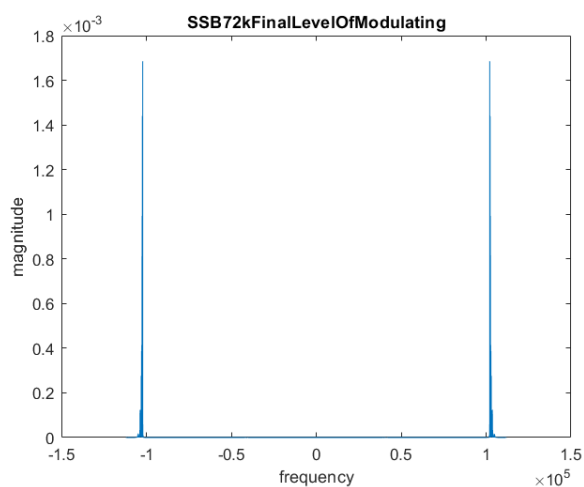
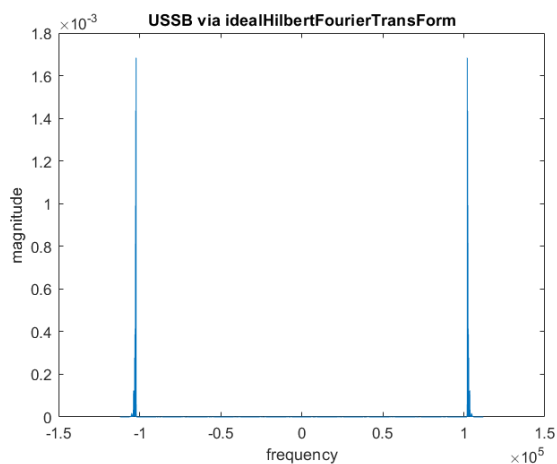
چ) در تک تک بخش ها بررسی نسبی ای به صورت زوم در نمودار ها شده است که همگی نشان دادند که شرایط پهنای باندی و فرکانسی در تک تک مراحل رعایت شده است. (جهت کنجکاوی مجدد کافی است در تک تک نمودار ها زوم کنید).

در برآیند اینجا به 2 جمع بندی مفید رسیده ایم

1- حالت میکسر و حالت هیلبرت ایده آل سیگنال مارا به صورت USSB با دقت بالا مدوله می کنند.

2- حالت تک کریر کسینوسی (سنکرون) و حالت هیلبرت غیر ایده آل سیگنال مارا کمتر به صورت USSB و بیشتر شبیه به DSB مدوله می کنند.

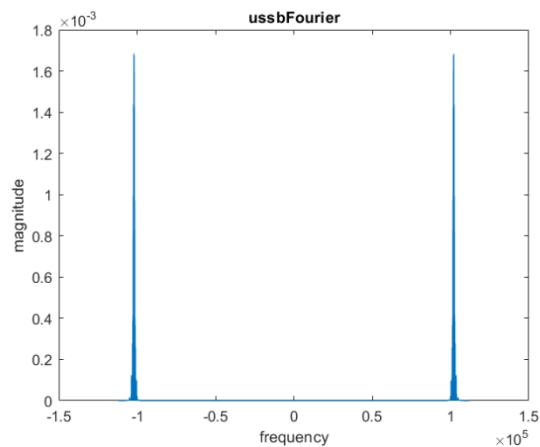
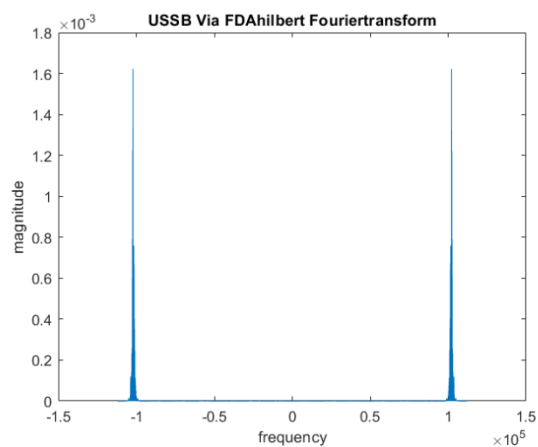
نکته ی مهم: دو نمودار هیلبرت ایده ال و حالت میکسر را مجدد در اینجا می آوریم:



همانطور که ملاحظه می کنید نمودار ها با دقت زیادی شبیه هم اند و با زوم کردن هم خیلی حالت USSB دارند و لذا این دو روش خیلی خوبند.

اما در مقابل روش های تک کریر کسینوسی و هیلبرت غیر آیده آل از دقت کمی برخوردار اند و برای مدولاسیون USSB مفید نمی باشند. مجدد نمودار های آنها را هم در اینجا می آوریم.

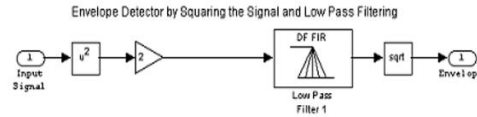
همانطور که ملاحظه می کنید نمودار های با دقت زیادی شبیه هم اند و خیلی از حالت USSB دور اند و برای مدولاسیون USSB خیلی بی کیفیت و بی دقت بوده اند و در عوض برای مدولاسیون DSB می توان از این دو روش استفاده کرد.



سوال سوم: دموولاتورهای مدولاسیون دامنه

(الف)

شکل اول:



$$\text{input signal} = m(t) \cos(2\pi f_c t) \rightarrow u^2: m^2(t) \cos^2(2\pi f_c t) = m^2(t) * \left(\frac{1}{2}\right) * (1 + \cos(4\pi f_c t))$$

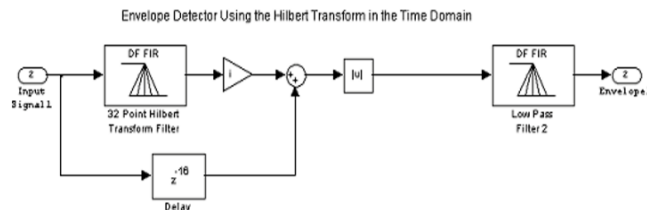
$$\text{after Gain: } m^2(t) \rightarrow \text{now after filtering we have : } m^2(t) * (1 + 0)$$

$$\text{after Sqrt} \rightarrow \text{sqrt}(m^2(t)) \rightarrow |m(t)|$$

حال داریم که اگر $m(t)$ همواره نامنفی باشد داریم که سیگنال به طور کامل بازایی می گردد.

نکته: بدیهی است که ترم $\cos(4\pi f_c t)$ به علت داشتن فرکانس بالای $4f_c$ از فیلتر عبور نمی کند و این ترم پس از فیلتر حذف می گردد.

شکل دوم:



در ابتدا داریم که فرض معقولی است که بگیریم فرکانس کریر کسینوسی از باند فرکانسی سیگنال پیغام ما فاصله کافی داشته باشد به نوعی که فرکانس بالا تلقی گردد و هم چنین داریم که در حالت پیوسته تاخیر نداریم و لذا خود سیگنال $m(t) \cos(2\pi f_c t)$ به جمع کننده می رسد حال داریم که با استفاده از ویژگی های تابع هیلبرت که خروجی مسیر اول به شرح زیر به دست می آید:

$$\text{Hilbert}(m(t) \cos(2\pi f_c t)) \rightarrow \text{according to above consideration} \rightarrow = m(t) \sin(2\pi f_c t) \text{ after Gain}$$

$$\text{after Gain} \rightarrow \text{output} = jm(t) \sin(2\pi f_c t) \rightarrow \text{there is no Dela in continous form} \rightarrow$$

$$\text{Adder output} = m(t) \cos(2\pi f_c t) + jm(t) \sin(2\pi f_c t) \rightarrow |u| \rightarrow \text{sqrt}(m^2(t) \cos^2(2\pi f_c t) + m^2(t) \sin^2(2\pi f_c t))$$

$$\text{simplifing} \rightarrow \text{sqrt}(m^2(t)(\cos^2(2\pi f_c t) + \sin^2(2\pi f_c t))) \rightarrow \text{sqrt}(m^2(t)) \rightarrow |m(t)|$$

حال داریم که اگر $m(t)$ همواره نامنفی باشد داریم که سیگنال به طور کامل بازایی می گردد.

بخش فیلتر پایین گذر موجود در انتها به نظر شخص بنده در جهت **نویرگیری** نهاده شده است.

ممکن است بپرسید که چرا در حالت گسسته delay داریم؟

پاسخ با توجه به داک متلب:

the Hilbert transform of the signal is found using a 32-point Parks-McClellan FIR filter. The Hilbert transform of the signal is then multiplied by i (the imaginary unit) and added to the original signal. The original signal is time-delayed before being added to the Hilbert transform to **match the delay** caused by the **Hilbert transform**, which is **one-half the length of the Hilbert filter**

لذا برای سینک شدن با دیلی تابع هیلبرت به اندازه i نصف سائز هیلبرت گسسته ($32/2 = 16$) نیاز به دیلی دادن ورودی و اضافه کردن آن مسیر بالایی است. (اما توجه کنید که در حالت پیوسته این اتفاق رخ نمی دهد (افتادن دیلی در ورودی)).

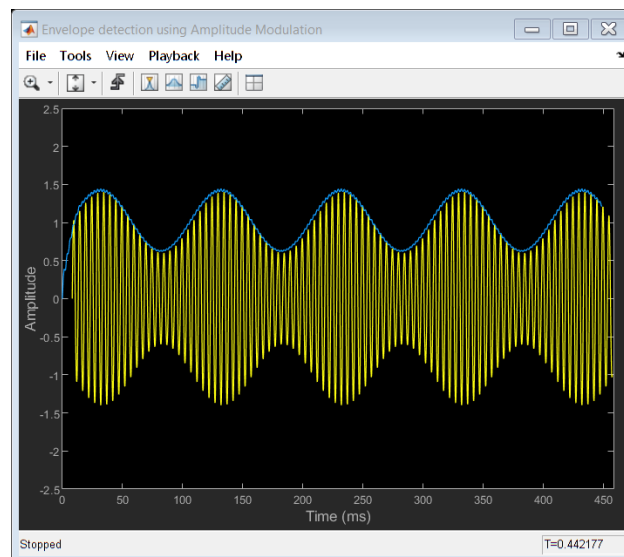
طبق صحبتی که تی ای محترم درس شد، ایشون اجازه دادند که برای این قسمت به اختیار یکی از راه های سیمولینک یا کد را انتخاب کنیم و بنده "کد" را انتخاب کردم، با مطالعه ی داک متلب و استفاده از کد آن به نتایج زیر می رسم:

ابتدا دو سینوسی با فرکانس های به ترتیب 10 و 200 و دامنه های به ترتیب 0.4 و 1 تولید می کنیم:

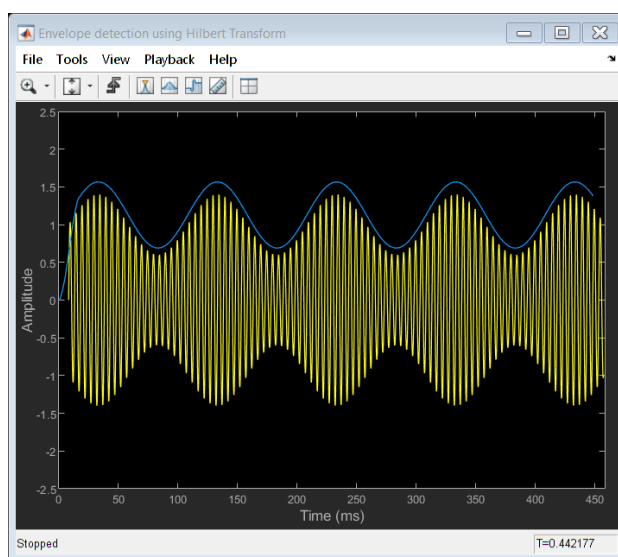
```
sine = dsp.SinWave([0.4 1],[10 200], ...  
    'SamplesPerFrame',frameSize, ...  
    'SampleRate',Fs);  
% first Domain 0.4,second Domain 1  
% first Frequency 10,second Frequency 200
```

به همین ترتیب الباقی بخش هارا در کد متلب طراحی می کنیم و سپس با تعریف کردن 2 اسکوپ به ترسیم محاسبات خود می پردازیم، خروجی ها به شرح زیر گردیدند:

خروجی حالت آشکارساز پوش به کمک مدولاتور AM:



خروجی حالت آشکارساز پوش به کمک Hilbert Transform:



مزایا و معایب:

در حالت AM داریم که دقت بالایی دارد، اما دندان‌دانه دندان‌دانه است و خروجی آن خیلی نرم نیست و از لحاظ فازی هم خیلی سینک و نزدیک به سیگنال اصلی است اما در حالت Hilbert داریم که خروجی بسیار تمیز و نرم است اما اندکی خطا دارد و اندکی نیز شیف‌ت فاز در آن دیده می‌شود.

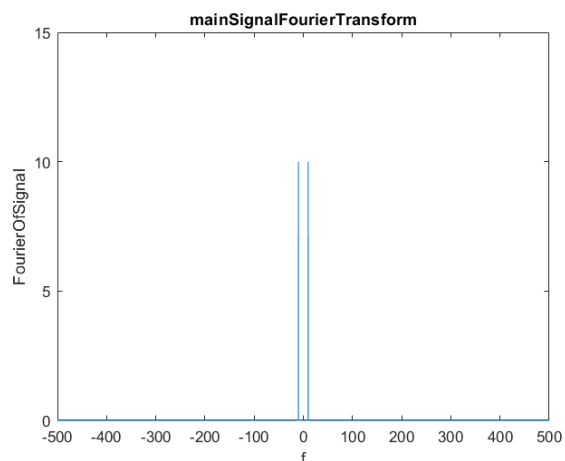
Accuracy & precision: AM better than Hilbert.

Smoothness: Hilbert better than AM.

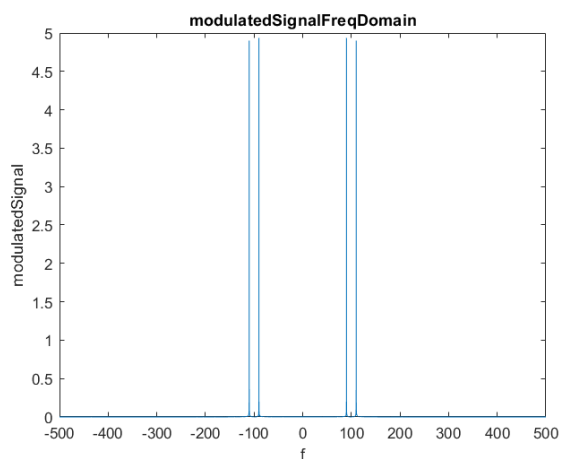
Phase Delay: AM better than Hilbert.

قسمت ۳، ۲: آشکارساز سنکرون (coherent):

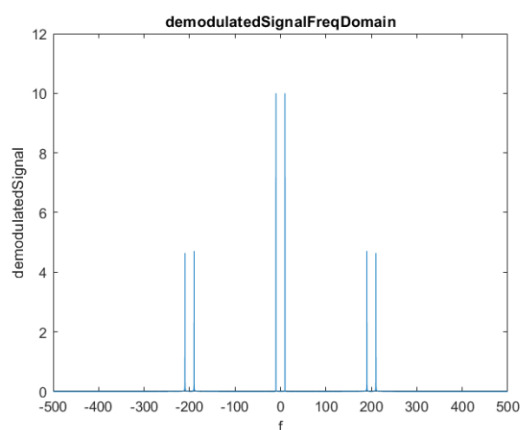
الف: یک سیگنال کسینوسی با فرکانس 10 هرتز تولید می‌کنیم به عنوان سیگنال اصلی و هم چنین کریر کسینوسی آن را نیز با فرکانس 100 هرتز تولید می‌کنیم و داریم که مراحل مدولاسیون به شرح نمودارهای زیر است:



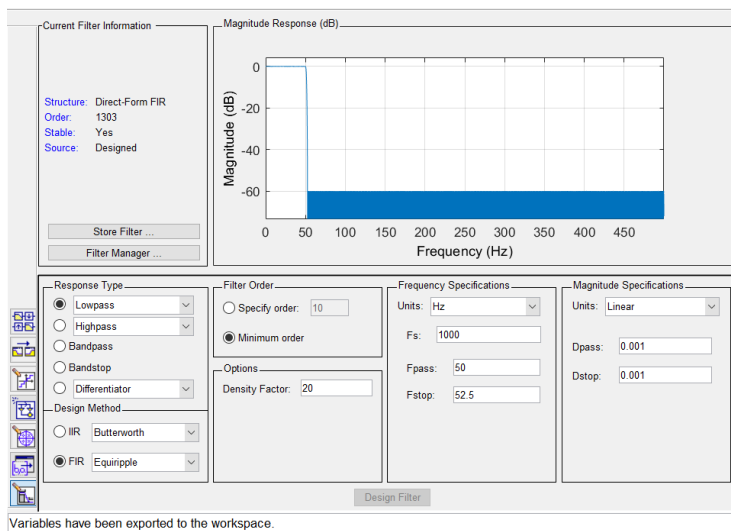
سیگنال اصلی پس از ضرب در کریر کسینوسی با فرکانس 100: (مدولاسیون)



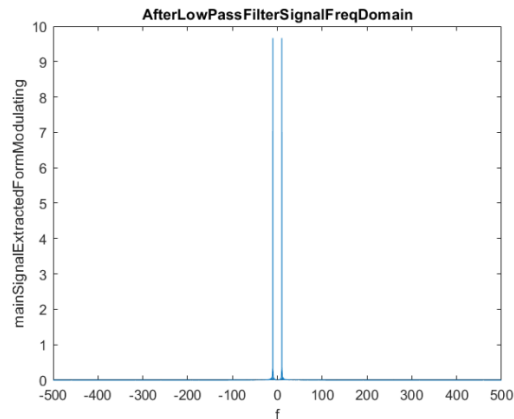
ضرب مجدد در کریر کسینوسی: (دمدولاسیون) (و نیز عبور از گین 2 برای احیای دامنه اصلی سیگنال)



عبور از فیلتر LowPass با مشخصات زیر:



فوريه ي سيگنال پس از عبور از فيلتر:



همانطور که مشاهده می کنید در حالتی که کریر ما در گیرنده فاز صفر داشته باشد با کیفیت خیلی خوبی سیگنال اصلی بازیابی می گردد. (دامنه بازیابی شده با دامنه اصلی برابر است)

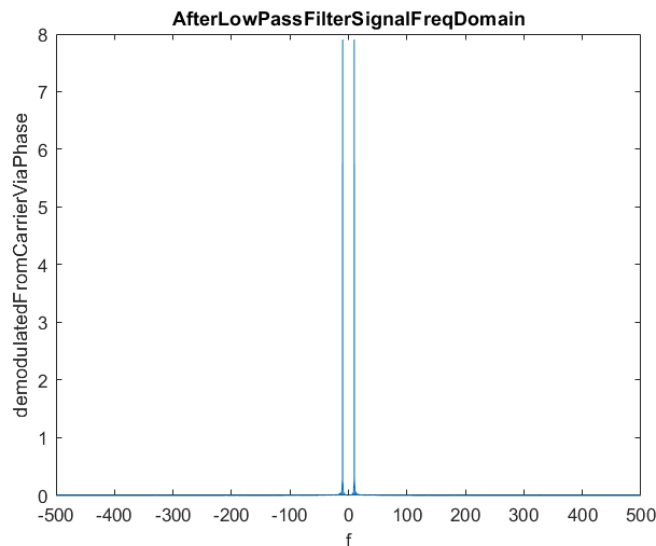
ب) افزودن فاز رندوم در گیرنده (آشکارساز)

```
rndphase = 2*pi*rand
```

```
rndphase = 3.9732;
```

```
cariViaPhase = cos(2*pi*fc*t + rndphase); % same sampling Rates.
```

حال داریم که نمودار ها به شکل زیر می گردند از پس از ضرب در کریر آشکارساز و عبور از فیلتر پایین گذر:



به سطح دامنه ها توجه کنید داریم:

اولا داریم که سیگنال ما در حوزه ی فوريه به صورت 2 ضربه است با دامنه ی زیر:

```
MainSignalFourierTransformAmplitude = 9.9991
```

حال... (ادامه در صفحه ی بعد)

حال اگر داشته باشیم که آشکارساز گیرنده ی ما اختلاف فاز نداشته باشد داریم که پس از دمدولاسیون داریم که اندازه ی دامنه های دلتای ما (تبدیل فوریه ی سیگنال دمدوله شده) به صورت زیر می گردد:

در حالتی که فاز صفر است داریم که:

`DemodulatedSignalFourierTransformAmplitudeWithoutPhase = 9.6635`

و همانطور که ملاحظه می کنید افت دامنه ناچیزی اتفاق می افتد.

اما در حالتی که فاز رندوم به کریر آشکار ساز ما اضافه می شود داریم که:

`When carrier has rndphase = 6.0162 phase`

`FreqAmpViaPhase = 9.3213`

`DampingRate = FreqAmpViaPhase/ MainSignalFourierTransformAmplitude = 0.9322`

`Cos(rndphase) = 0.9446`

با نوشتن روابط ریاضی هم به سادگی به دست می آوریم که وجود فاز در کریر آشکار ساز باعث افت دامنه با ضریب کسینوس (فاز کریر آشکار ساز) است. و همانطور که مشاهده می کنید نیز این روابط تصدیق عملی و "متلب"ی هم دارند لذا برای حل این مشکل کافی است که در خروجی یک گین $1/\cos(\text{phase})$ قرار بدهیم تا افت دامنه جبران شود.

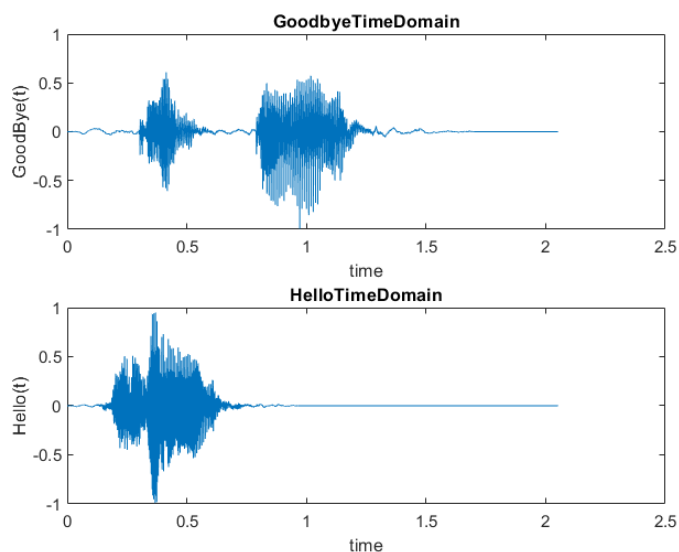
• راه های دیگر:

- 1- استفاده از مدارهای قفل فاز
- 2- استفاده از VCO
- 3- استفاده از PLL
- 4- ...

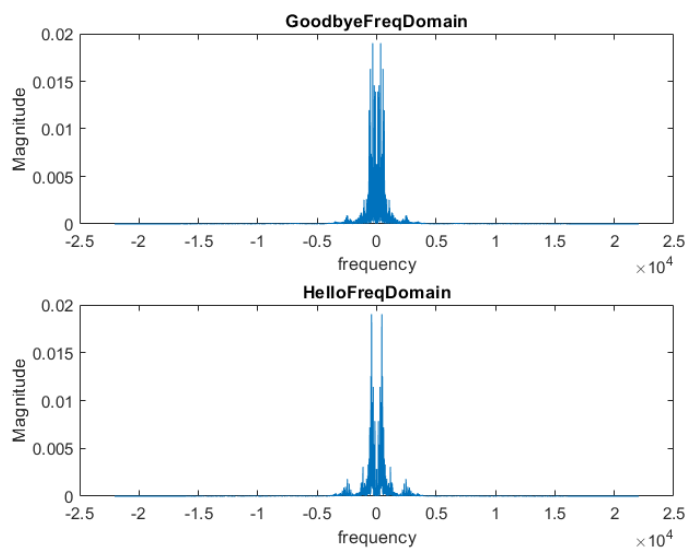
سوال چهارم:

ابتدا نمودار های حوزه ی زمان و فرکانس هر دو پیغام صوتی را رسم می کنیم.

نمودار های حوزه ی زمان دو پیغام “Hello” , “GoodBye”



نمودار های حوزه فرکانس دو پیغام “Hello” , “GoodBye”



همانند سوال دوم قسمت محاسبه ی پهنای باند داریم که:

یادآوری از صفحه ی 8 گزارش:

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 محدوده 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 آنجا 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 آنجا 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 در نظر 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 برای دو سیگنال مختلط Hello , GoodBye برابر است با:

```
BWhello = 3.2631e+03
```

```
BWgoodbye = 1.8308e+03
```

همانند مطالب و قضایای مطرح شده برای بخش upSampling در سوال دوم (از تکرار مجدد مطالب گفته شده

خودداری می کنیم) برای محاسبه ی نرخ سمپلینگ به ترتیب زیر عمل می کنیم:

1-اولا داریم که هر دو سیگنال با نرخ 44.1 کیلو هرتز سمپل شده است.

2-یکی از سیگنال ها قرار است به باند مرکزی 130 کیلو هرتز برود لذا در این حالت کریر کسینوسی حداقل با نرخ 260 کیلو هرتز (قضیه نایکوئیست) نمونه برداری شده است حال داریم که پس حداقل مقدار لازم جهت upSampling سیگنال برابر است با:

$\text{SamplingIntegerRate} = 260\text{KHz}/44.1\text{KHz} = 5.89 \rightarrow \text{SamplingRate} \geq 6$

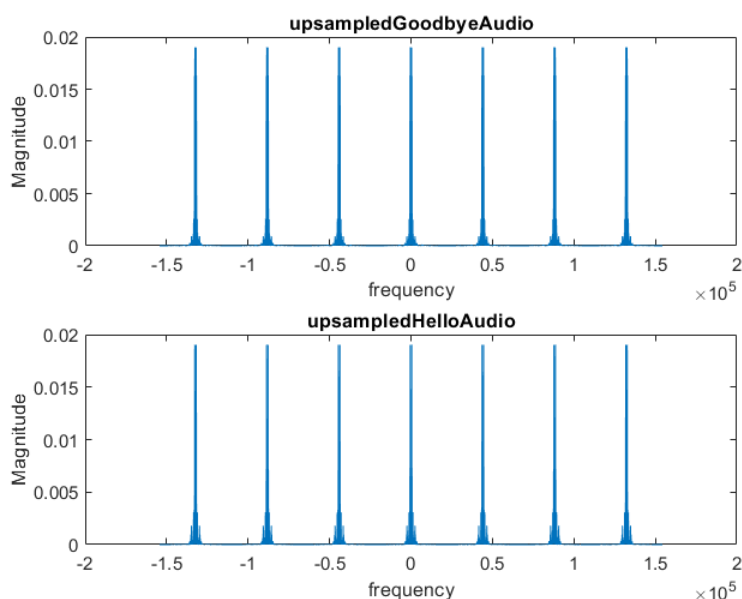
حال جهت جلوگیری از لبه مرز! عمل کردن نرخ آپسمپلینگ را 7 در نظر میگیریم.

سیگنال ها را آپسمپل می کنیم:

```
upsampledHello = upsamplingIntegerFactor.*upsample>HelloSignal,upsamplingIntegerFactor);
```

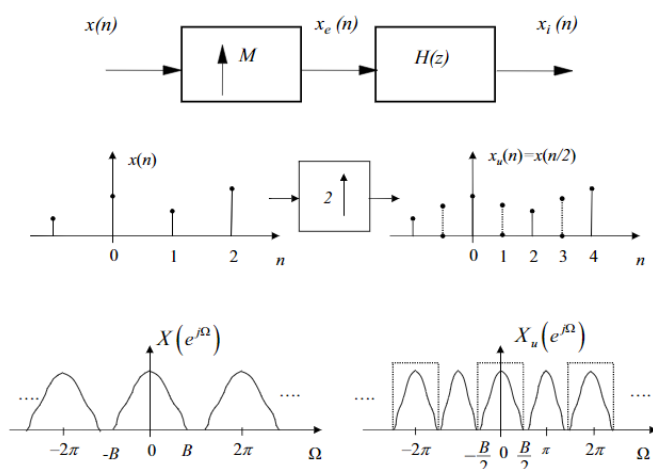
```
upsampledGoodBye = upsamplingIntegerFactor.* upsample(GoodByeSignal,upsamplingIntegerFactor);
```


حال نمودار های Upsample شده ی دو سیگنال را (پیش از عبور از فیلتر جدا کننده تک سیگنال از متناوب های خود در اثر upsampling) را رسم می کنیم:



یادآوری:

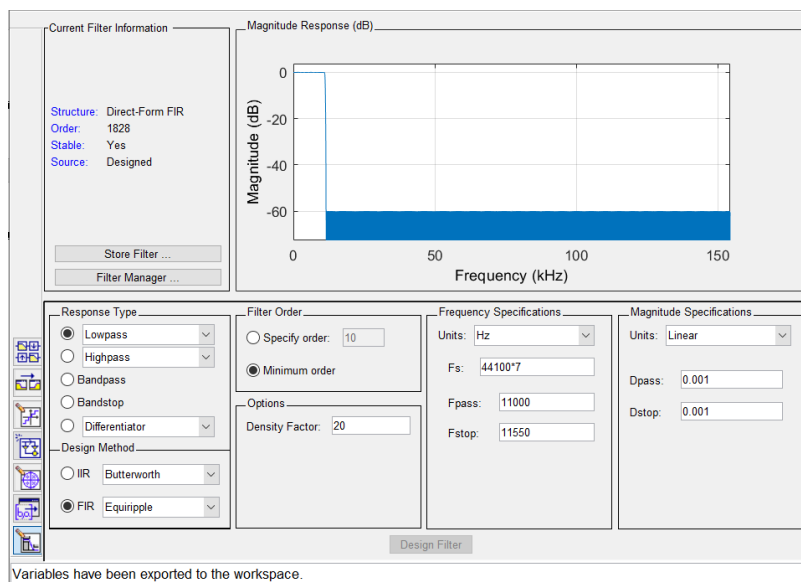
UPSAMPLING AND RECONSTRUCTION



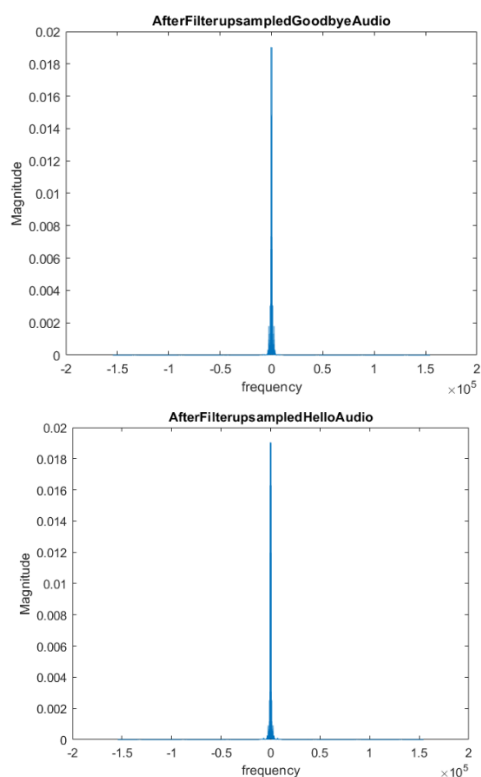
حال کافی است فیلتر مدنظر upSampling را جهت Reconstruction یکی از تبدیل فوریه ها طراحی کنیم. داریم:

طبق نمودارهای بالا داریم که فیلتر ماباید حداکثر تا $B/2$ را عبور بدهد لذا داریم که $2*B < f_s$ لذا $B < 22050$ و لذا $B/2 < 11025$

و لذا داریم که یک فیلتر پایین گذر تا فرکانس 11 کیلوهرتز با ضریب نشتی 0.05 قرار می دهیم که همانطور که در شکل ملاحظه می کنید مشخصات این فیلتر آماده است.

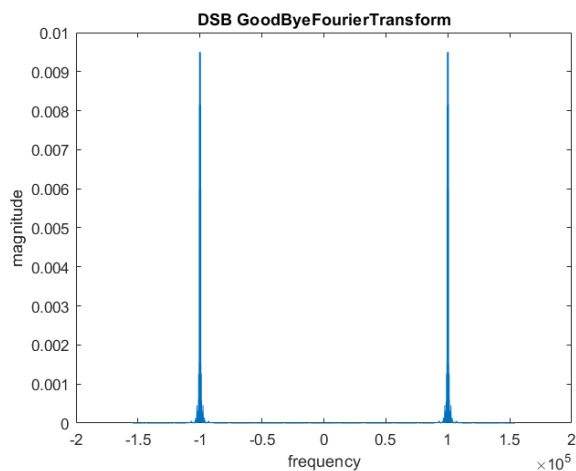
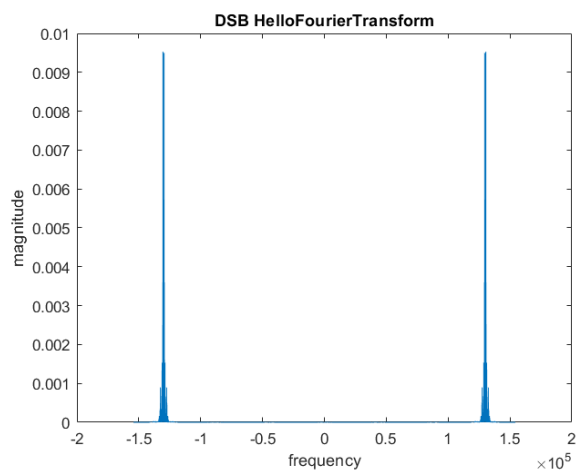


پس از اعمال فیلتر ها روی نمونه های Upsample شده داریم که:

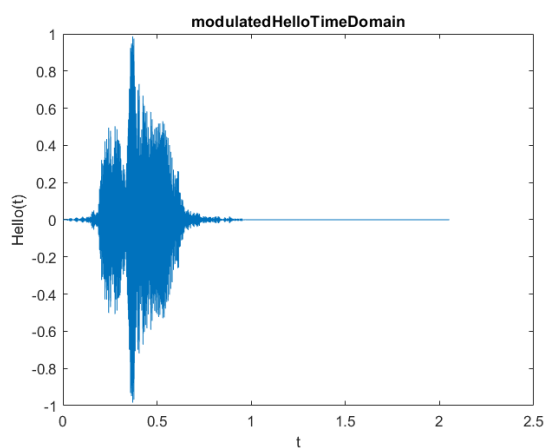


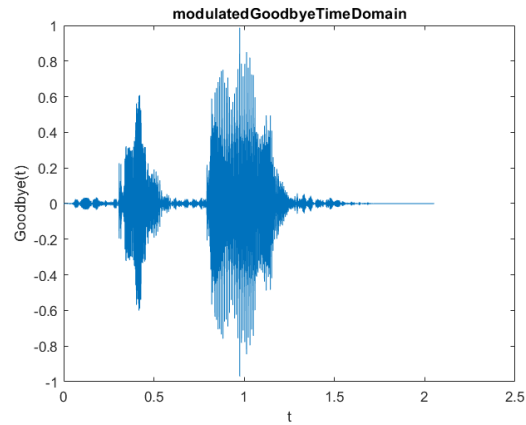
حال سیگنال هایمان را در کریر های کسینوسی مربوطه شان ضرب می کنیم و مدوله شده ی DSB آن ها را رسم می کنیم، (جهت حفظ گین سیگنال در هنگام دمودیلاسیون سیگنال را در 2 ضرب خواهیم کرد) (گین 2 قرار خواهیم داد) تا افت دامنه ناشی از ضرب در کریر کسینوسی جبران شود. داریم که نمودارها به شرح زیر گشت:

سیگنال Hello را در فرکانس 130 کیلو هرتز مدوله می کنیم و سیگنال GoodBye را در فرکانس 100 کیلو هرتز مدوله می کنیم.

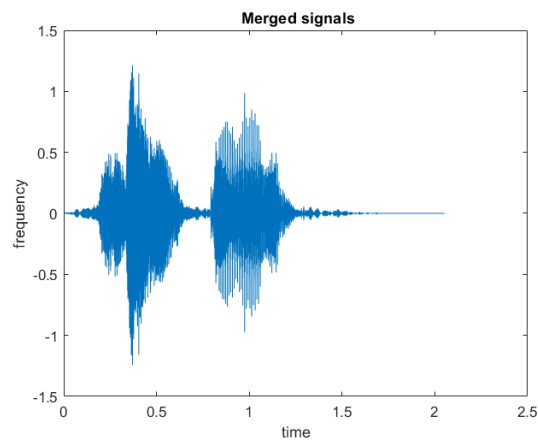


سیگنال های حوزه ی زمان مدوله شده:

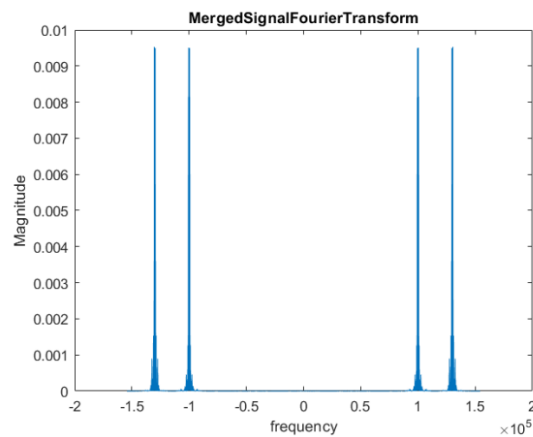




حال سیگنال جمع شده ی مدوله های Hello,GoodBye:



سیگنال جمع شده در حوزه ی فرکانس:

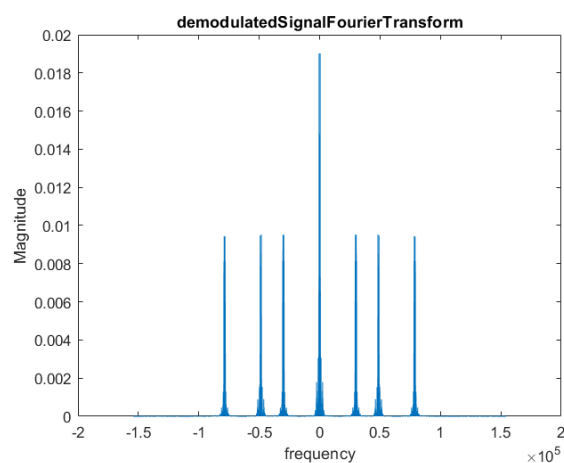
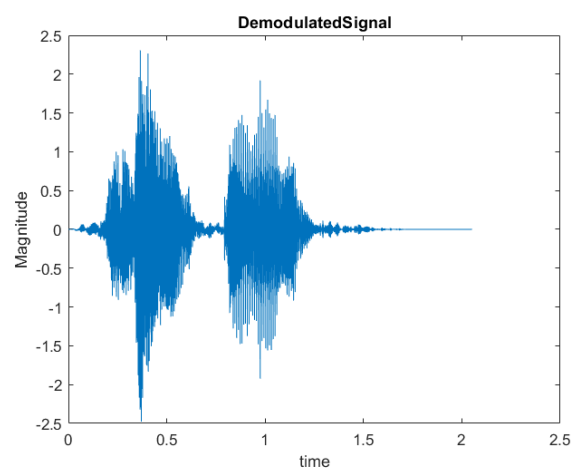


از آنجا که ما سیگنال Hello را در کریر 130 کیلوهرتز ضرب کردیم جهت احیا نسخه ی اصلی سیگنال لازم است تا مجدد در کریر 130 کیلوهرتز ضرب کنیم و از فیلتر پایین گذر عبور دهیم، مراحل ذکر شده را به ترتیب در صفحه ی بعد انجام می دهیم.

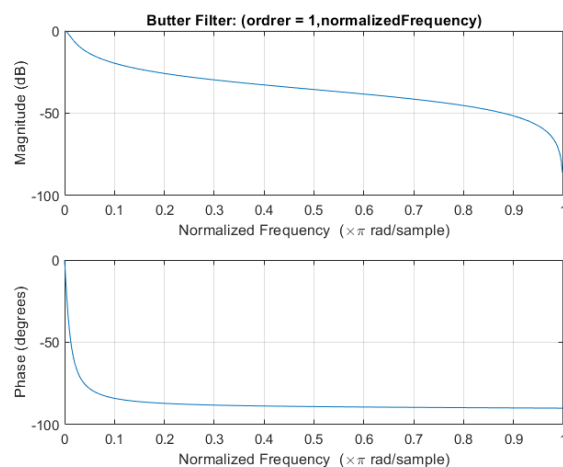
```
% demodulating
demodulated = 2.*firstCarrier.*mergedSendingMessage % Gain 2 inOrder synching
%demodulated Signal's amplitude to it's main Amplitude.
```

ضرب در کریر سیگنال Hello، حال سیگنال دموله شده را در حوزه ی زمان و فرکانس رسم می کنیم، داریم:

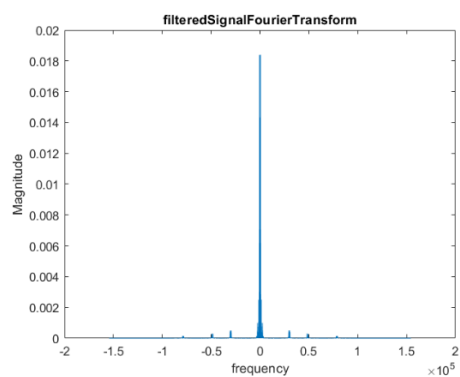
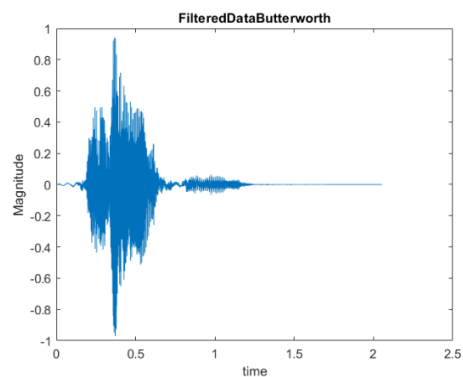
سیگنال به ترتیب در حوزه ی زمان و فرکانس:



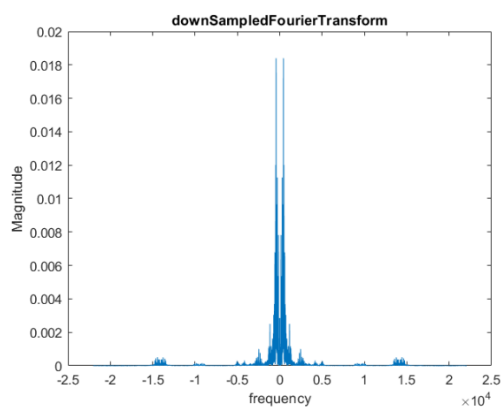
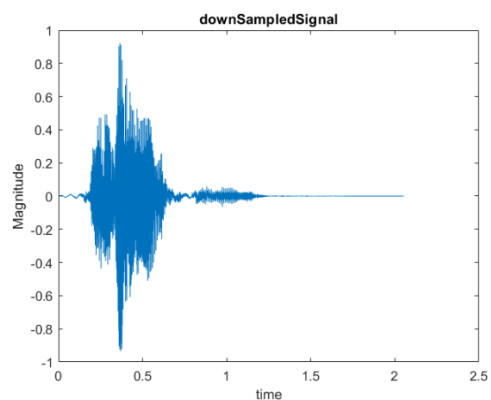
چون داریم که فیلتر butter در ورودی هایش مرتبه و فرکانس قطع **استاندارد** می خواهد لازم است تا پهنای باند سیگنال hello را بر فرکانس سیمپلینگ (طبیعتا پس از upsampling) تقسیم کنیم و به عنوان فرکانس قطع استاندارد به ورودی اش بدهیم. حال داریم ک مشخصات این فیلتر مرتبه ی یک butter به شرح زیر است:

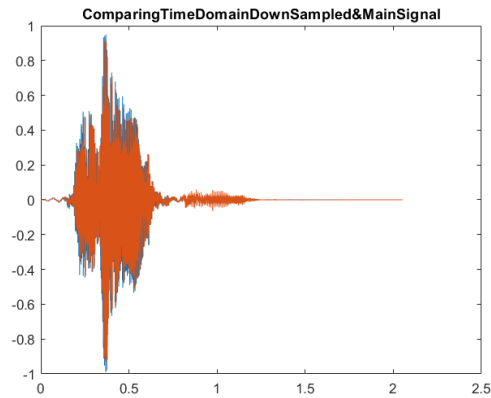


حال داده را به کمک فیلتر butter می‌کنیم: (فیلترینگ در حوزه ی زمان و فرکانس)



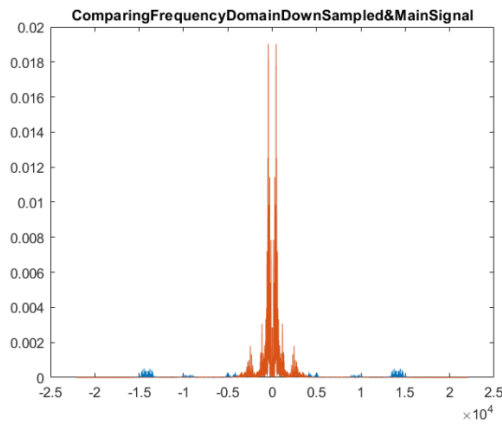
حال سیگنال مربوطه را DownSample می‌کنیم و سیگنال به دست آمده در حوزه ی زمان و فرکانس به ترتیب به شرح زیر است:





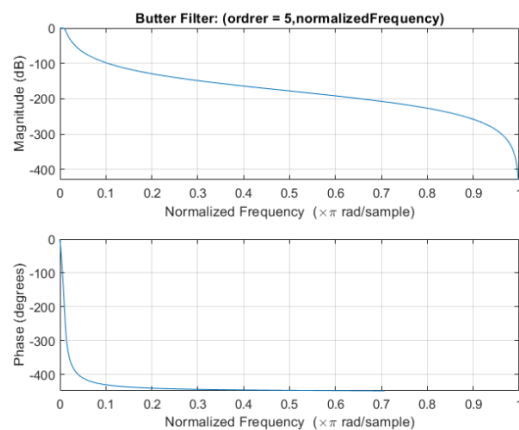
داریم که خطای MeanSquare سیگنال اصلی و سیگنال دمدوله شده در حوزه ی زمان برابر است با:
MSE Error in time Domain = 0.0305

حال برویم به حوزه ی فرکانس:



MSE Error in frequency Domain = 3.4843e-09

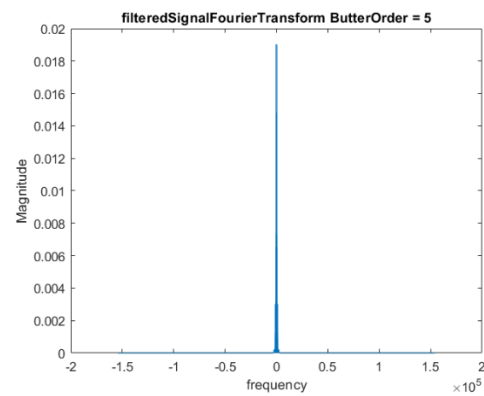
حال محاسبات را با مرتبه ی فیلتری بالاتری مجدد انجام می دهیم...
دراین مرحله مرتبه ی butter را برابر 5 در نظر میگیریم و داریم که نمودار مشخصات فیلتر به شرح زیر می گردند:



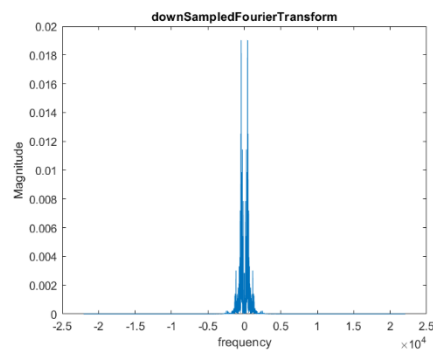
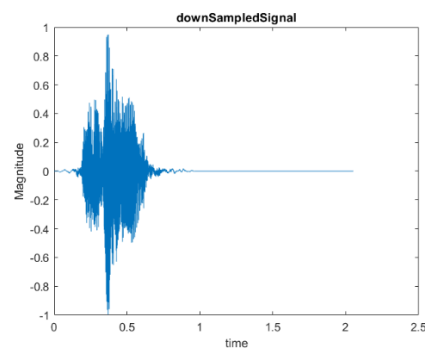
نمودار سیگنال در حوزه ی زمان:



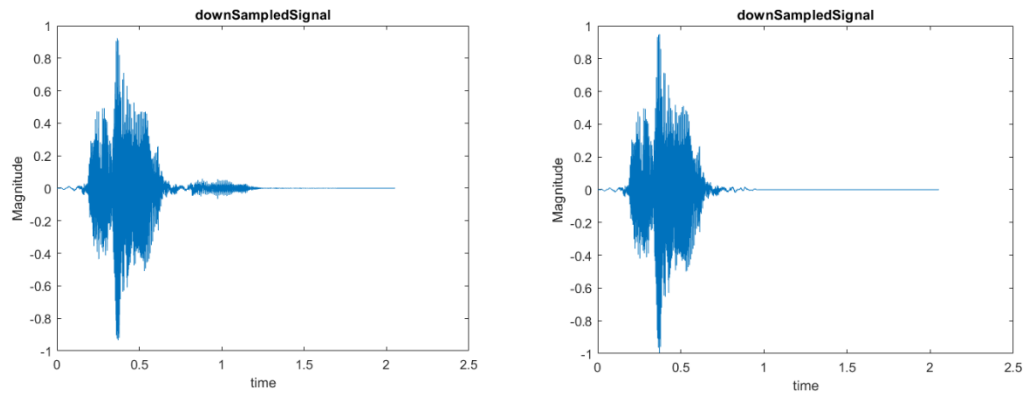
نمودار در حوزه ی فرکانس سیگنال Hello بازیابی شده به صورت زیر است:



حال سیگنال مربوطه را DownSample می کنیم و سیگنال به دست آمده در حوزه ی زمان و فرکانس به ترتیب به شرح می گردد:

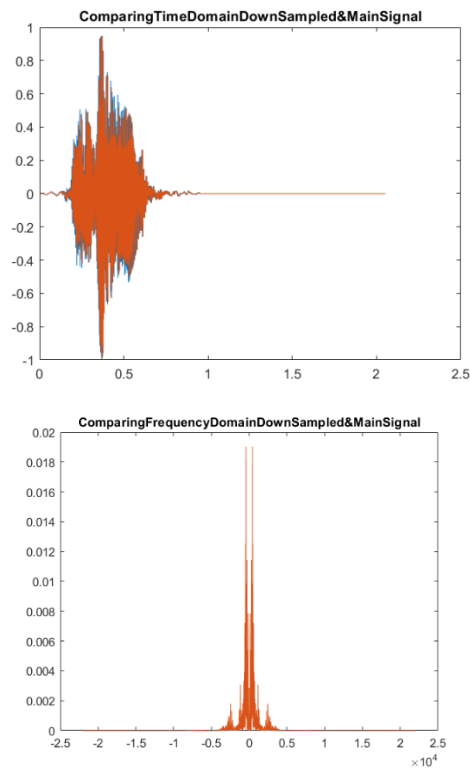


اگر در حوزه ی زمان دو حالت مرتبه ی 1 و مرتبه ی 5 را با یکدیگر مقایسه کنیم داریم که:



با مقایسه ی داده های بالا با شکل اصلی سیگنال Hello که در ابتدای گزارش سوال 4 آورده ام داریم که داده ی بازیابی شده در حوزه زمان به **ظاهر نزدیک** تر است به سیگنال اصلی. (در ادامه خطای MSE را محاسبه می کنیم).

نمودار های مقایسه:



و اما خطا ها:

$\text{errorInTimeDomain} = 0.0345 \rightarrow$ **error increases from 0.0305 to 0.0345**
 $\text{errorInFreqDomain} = 4.7320\text{e-}09 \rightarrow$ **error increases from 3.4843e-09 to 4.7320e-09**

شاید تعجب کرده باشید که چرا خطا افزایش یافته است؟ اما پاسخ... (ادامه در صفحه ی بعد)

علت رخ داد این خطا در این است که بله مرتبه و دقت فیلتر به ظاهر زیاد شده است اما دقت کنید جا دادن ویژگی فیلتر در چند جمله ای های مرتبه بالاتر باعث رخ داد پدیده ای به اسم "overfit" می گردد که این رخ داد باعث می شود که فیلتر به تغییرات کوچک (اصطلاحاً نویز) حساس می شود و از رفتار اصلی شکل فیلترینگ خود دور می شود و بیشتر حساس به جزییات می گردد و این باعث می شود که همانطور که ملاحظه می کنید خطا افزایش پیدا کرده است. (البته ناگفته نماند که افزایش مرتبه فیلتر تا حدی که باعث حساسیت به نویز و **overfitting** نگردد مفید خواهد بود و خطا را کاهش می دهد!)