باسمه تعالی

دانشگاه صنعتی شریف دانشکده مهندسی برق

درس: سیستم های مخابراتی

گزارش تمرین سری اول متلب

امیرحسین رستمی 96101635

استاد: دکتر بهروزی

سوال اول:

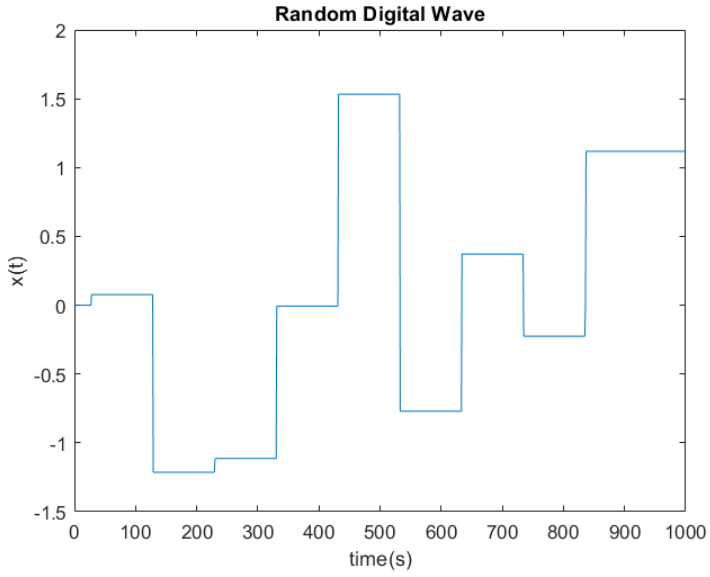
قسمت الف:

برای طراحی موج دیجیتال بنده یه تابع 4 متغیره درنظرگرفتم که متغیر اول طول سیگنال کلی موج دیجیتالی است و متغیر دوم فرکانس نمونه برداری از سیگنال اصلی موج دیجیتالی(پیوسته!) است که یعنی ما قدم های گسسته خود را به صورت `1/frequencySampling` در نظربگیریم و فاصله ی زمانی بین اعضای دامنه ی موج دیجیتالی گسسته بازگشتی برابر وارون فرکانس نمونه برداری داده شده به تابع است.پارامتر سوم این تابع واریانس توزیع ای است که از آن ما a_k هارا استخراج می کنیم.توجه کنید که چون شرط این است که توزیع دارای میانگین صفر باشد و شرط دیگری روی توزیع انتخابی ما ندارد ما توزیع انتخابی خود را گوسی می گذاریم.توجه کنید که هرکدام از a_k از توزیع گوسی انتخاب می شوند اما متغیر D از توزیع یکنواخت انتخاب شده است.پارامتر چهارم هم بیانگر بازی زمانی ای است که هر کدام از a_k دوام پیدا خواهند کرد و چون ما سیگنال گسسته برمی گردانیم با تقسیم Tبر فرکانس نمونه برداری خواهیم داشت که چند نقطه متوالی از سیگنال گسسته ما دارای مقدار ثابت a_k باشد.پیاده سازی این تابع به شرح زیر است:

1- محاسبه ی numberOfIntervals
2- محاسبه ی a_k ها با استخراج از توزیع نرمال.
3- انتصاب اعضای هر interval به مقدار a_k انتخابی(به تعداد T/samplingFrequency عضو باید مقدارa_k به خود بگیرند.

پس از طی مراحل فوق تابع نهایی آماده است.

نمونه خروجی گرفته شده:



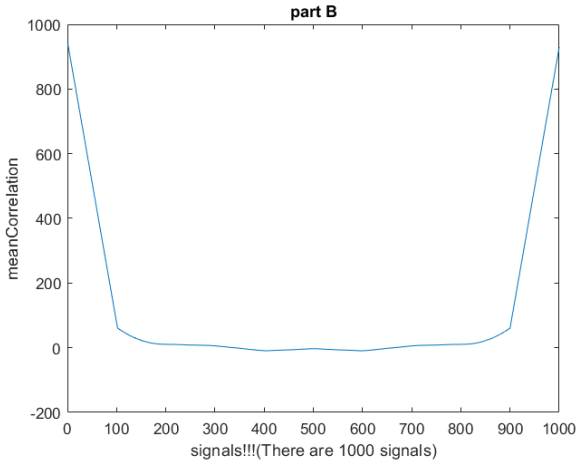
قسمت ب:

برای محاسبه ی تمامی همبستگی های موجود بین تمامی سیگنال های موجود،تابع زیر را پیاده سازی کرده ام که در در ورودی،ورودی های تولید یک سیگنال digitalWave را به همراه تعداد درخواستی برای تولید این تعداد سیگنال تصادفی را میگیرد و در خروجی اش تمامی سیگنال های تولیدی و ماتریس همبستگی سیگنال های تولید شده را بر می گرداند.این ماتریس یک ماتریس دو بعدی به ابعاد تعداد سیگنال های درخواستی در طول هر سیگنال است و درایه ی سطر ij موجود در این ماتریس برابر است با همبستگی سیگنال i ام با شیفت یافته ی خود به اندازه ی j مقدار است. محاسبه ی همبستگی هم به این صورت است که آرایه ی سیگنال اصلی را شیفت می دهم و سپس ضرب نقطه ی ای می کنم سیگنال اصلی را در مزدوج شیفت یافته اش و جمع درایه های آرایه نهایی برابر است با مقداری همبستگی سیگنال اصلی ام با شیفت یافته ی j ام اش(به تک عبارت زیر توجه کنید که همبستگی بین سیگنال اول و سیگنال دوم را محاسبه می کند و در خروجی می دهد.

```
output = sum(firstSignal.*conj(secondSignal));
```

نمونه ی خروجی گرفته شده در این قسمت با فرایض زیر:

```
length = 1;
frequency = 1000;
sigma = 1;
T = 0.1;
numberOfSignals = 1000;
```



توجیه درستی:توجه کنید که همانطورمی دانید من از corr حلقوی استفاده کردم و سیگنال را واحد واحد شیفت دادم و با خودش corr حساب کردم،توجه کنید در اوایل شیفت سیگنال شبیه خودش است و لذا به صورت خطی نسبتا خطی در اول با شیفت دادن هایش کاهش پیدا می کند،علی ذلک در انتهای هم چون من شیفت حلقوی می دهم داریم که در انتهای هم از سر دیگر حلقه دوباره سیگنال شیفت حلقوی یافته،آنقدر شیفت حلقوی خورده تا اینکه خیلی شبیه سیگنال اصلی اش شده است و لذا پس از مدتی این همبستگی دوباره افزایش پیدا می کند تا در نهایت سیگنال آنقدر شیفت حلقوی بخورد تا دور بخورد و دوباره تشابه حداکثر(ماکزیمم همبستگی) رخ بدهد.

قسمت پ:

ابتدا نتایج این بخش را ذکر می کنیم و سپس نمودار هایش را ضمیمه می کنیم،داریم:

اثر T:

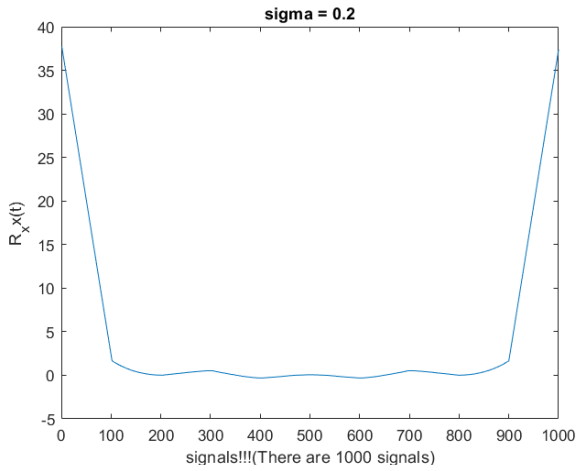
هر چه مقدار T کمتر می شود ماهیت تصادفی سیگنال بیشتر می شود و لذا اشتراک و همبستگی آن با شیفت یافته های خود کاهش می یابد و لذا طول نیم مثلث های موجود در طرفین سیگنال کاهش پیدا می کند.(همانطور که مشاهده می کنید)

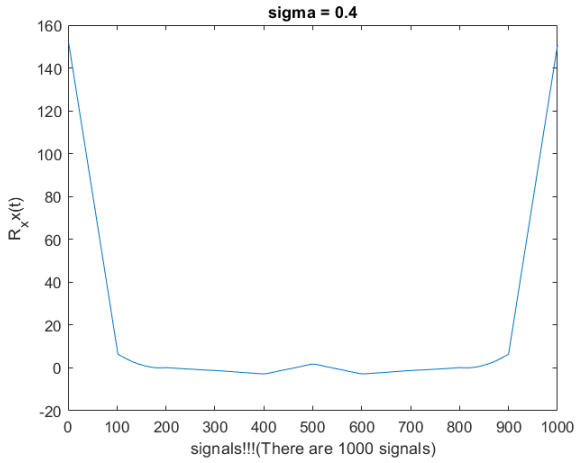
اثر Sigma:

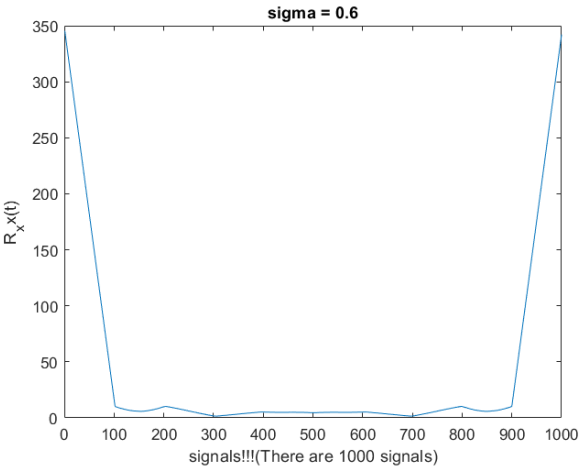
همانطورکه مشاهده می شود تغییر مقدار سیگما اثری روی شکل تابع همبستگی ندارد و فقط اسکیل اندازه ی محور y را تغییر می دهد و همانطورکه مشاهده می شود اندازه ی نمودار با مجذور سیگما رابطه ی مستقیم دارد.

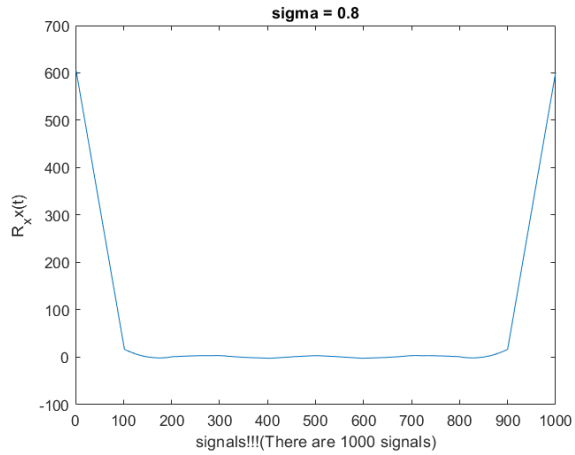
در این قسمت ابتدا مقدار sigma را از 0.2 تا 1 با ثابت بودن T تغییر دادیم و حاصل 5 نمودار زیر گشت و در وهله ی بعد مقدار T را از 0.2 تا 1 با ثابت بودن T تغییر دادیم و حاصل 5 نمودار زیر گشت.

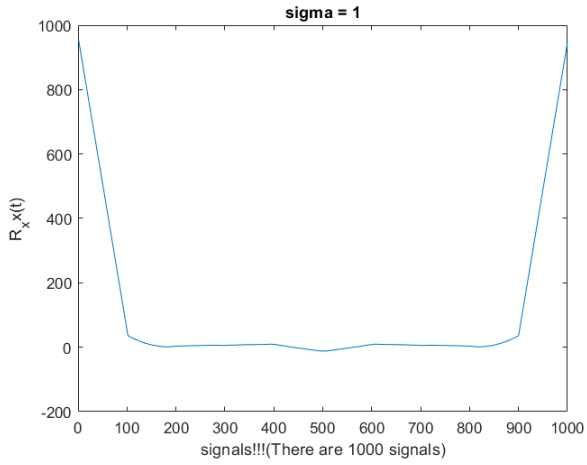
قسمت اول: تغییر sigma با ثابت بودن T



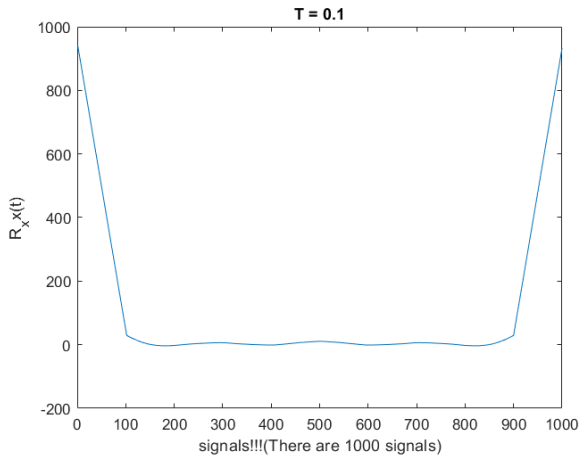


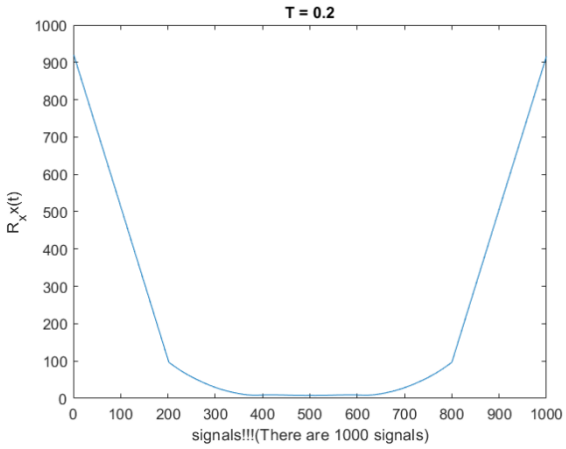


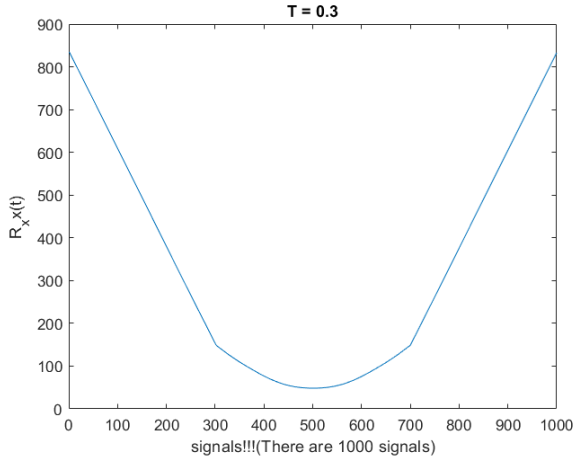


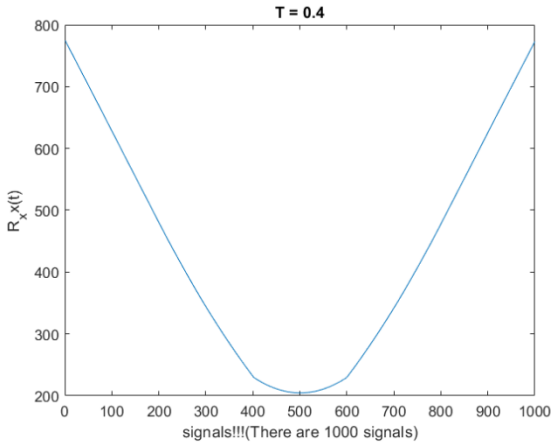


قسمت دوم: تغییر T با ثابت بودن Sigma:









قسمت ت:

می دانیم که رابطه ی تابع چگالی طیف با تابع خودهمبستگی به این صورت است که تبدیل فوریه ی تابع خودهمبستگی برابر است با تابع چگالی طیف سیگنال اصلی است.

$$S_{xx}(f) = |\hat{x}(f)|^2 \quad \text{(Eq.1)}$$

$$S_{xx}(\omega) = \lim_{T\to\infty} \mathbf{E}\left[|\hat{x}(\omega)|^2\right] \quad \text{(Eq.2)}$$

حال داریم که رابطه ی بین تابع چگالی طیف با تابع خودهمبستگی به شرح زیر به دست می آید

$$\mathbf{E}\left[|\hat{x}(\omega)|^2\right] = \mathbf{E}\left[\frac{1}{T}\int_0^T x^*(t)e^{i\omega t}\,dt \int_0^T x(t')e^{-i\omega t'}\,dt'\right] = \frac{1}{T}\int_0^T\int_0^T \mathbf{E}\left[x^*(t)x(t')\right] e^{i\omega(t-t')}\,dt\,dt'.$$

$$R_{xx}(\tau) = \langle X(t)X(t+\tau)\rangle = \mathbf{E}[X(t)X(t+\tau)]$$

$$R_{xx}(\tau) = \langle X(t)X(t-\tau)^*\rangle = \langle X(t)^*X(t+\tau)\rangle$$

$$S_{xx}(\omega) = \int_{-\infty}^{\infty} R_{xx}(\tau)e^{-i\omega\tau}\,d\tau = \hat{R}_{xx}(\omega) \quad \text{(Eq.3)}$$

حال برای رعایت اصل reusability در کد خود،تابع زیر را پیاده سازی می کنیم که یک سیگنال خودهمبستگی می گیرد و چگالی طیف انرژی را برمی گرداند.

ابتدا ذکر نکته ی زیر حایز اهمیت است که برای گرفتن fft از یک سیگنال M نقطه ای ورودی که بخواهد در حوزه ی فرکانسی fft را در بازه ی منفی pi تا pi بدهد لازم است علاوه بر fft گرفتن از سیگنال اصلی،از fft شیفت نیز استفاده بکنیم لذا لازم است تا:

1ـ ابتدا fft M نقطه ای سیگنال ورودی را محاسبه کنیم.

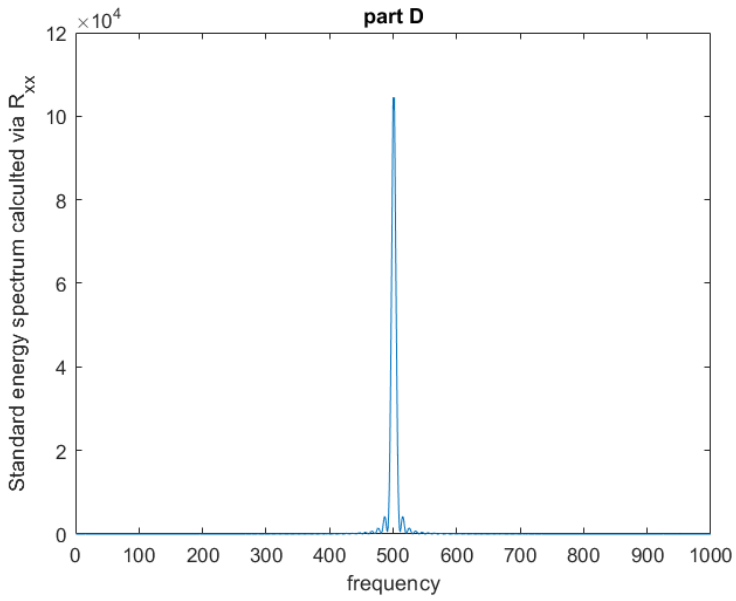
2ـ سپس شیفت فرکانسی سیگنال محاسبه شده در قسمت 1 جهت قرار دادن محدوده ی فرکانسی در بازه ی $[-\pi,\pi]$

- مرحله ی اول را به کمک تابع fft(x,M) انجام می دهیم
- مرحله ی دوم را به کمک تابع fftshift انجام می دهیم.

برآیند کار های فوق در تکه کد زیر آورده شده است.

```
function out = sdf(signal)
out = shiftfft(fft(signal,size(signal,2)))
end
```

حال برویم سراغ نمودار چگالی طیف انرژی(محاسبه شده به کمک اعمال تابع بالا روی سیگنال خودهمبستگی)



قسمت ث:

یادآوری: قضیه ی wiener kinchin و اثبات آن

طبق تعریف می دانیم که تابع خودهمبستگی سیگنال E به صورت زیر تعریف می شود.

$$C(t) \equiv \int_{-\infty}^{\infty} \overline{E}(\tau)\, E(t+\tau)\, d\tau.$$

هم چنان می دانیم که تبدیل فوریه ی سیگنال E نیز به صورت زیر تعریف می شود

$$E(\tau) = \int_{-\infty}^{\infty} E_\nu\, e^{-2\pi i \nu \tau}\, d\nu,$$

با مزدوج گرفتن از رابطه ی بالا داریم که :

$$\overline{E}(\tau) = \int_{-\infty}^{\infty} \overline{E}_\nu\, e^{2\pi i \nu \tau}\, d\nu.$$

ادامه در صفحه ی بعد.

به کمک دو تعریف بالا و قرار دادن معادل های آنها در رابطه ی همبستگی داریم که:

$$
\begin{aligned}
C(t) &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\overline{E}_{\nu}\, e^{2\pi i \nu \tau}\, d\nu\right]\left[\int_{-\infty}^{\infty}E_{\nu'}\, e^{-2\pi i \nu' (t+\tau)}\, d\nu'\right] d\tau \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\overline{E}_{\nu}\, E_{\nu'}\, e^{-2\pi i \tau (\nu'-\nu)}\, e^{-2\pi i \nu' t}\, d\tau\, d\nu\, d\nu' \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\overline{E}_{\nu}\, E_{\nu'}\, \delta(\nu'-\nu)\, e^{-2\pi i \nu' t}\, d\nu\, d\nu' \\
&= \int_{-\infty}^{\infty}\overline{E}_{\nu}\, E_{\nu}\, e^{-2\pi i \nu t}\, d\nu \\
&= \int_{-\infty}^{\infty}|E_{\nu}|^2\, e^{-2\pi i \nu t}\, d\nu \\
&= \mathcal{F}_{\nu}\left[|E_{\nu}|^2\right](t),
\end{aligned}
$$

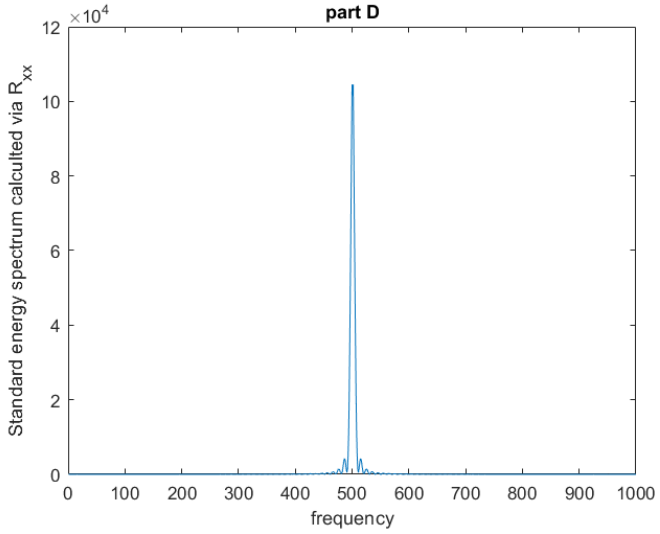
به طرز عجیبی داریم که خودهمبستگی به سادگی توسط تبدیل فوریه مربع اندازه یE به دست می آید.

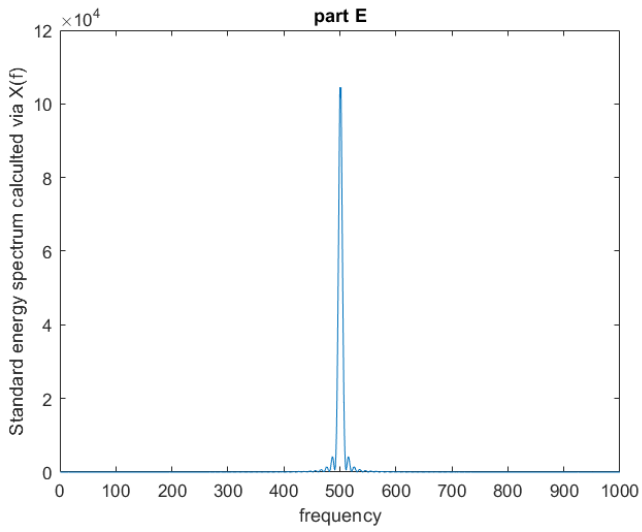
ما در این قسمت چگالی طیف انرژی را هم به کمک X(f) و هم به کمک R_xx(t) محاسبه کردیم و در نهایت خطای MSE آنها را نیز محاسبه و ترسیم کردیم.

$$S_x(f) = E\{|X(f)|^2\}$$

$$S_x(f) = F\{R_{xx}(t)\}$$

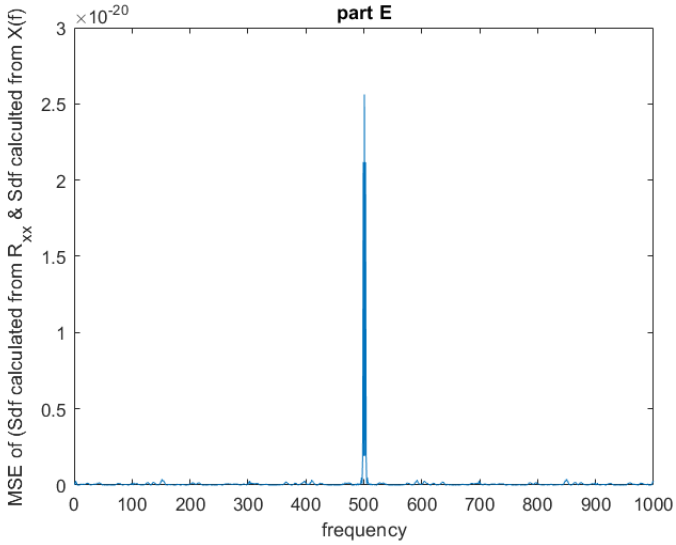
حال به کمک روابط رو به رو به محاسبه و ترسیم نمودار های مطلوب می پردازیم:





حال به محاسبه MSE می پردازیم که رابطه ی MSE به صورت زیر است:

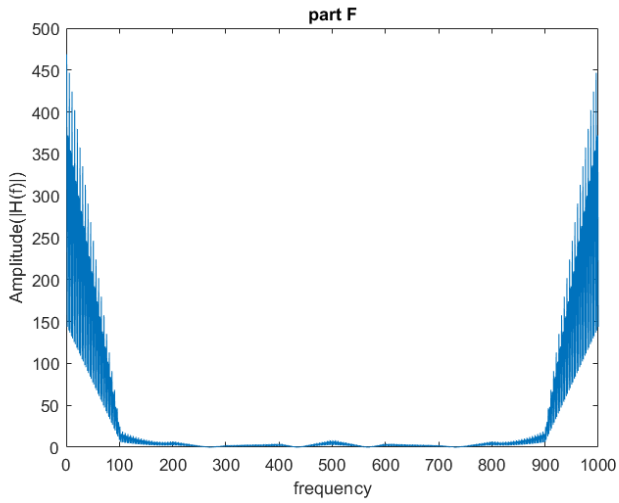
```
MSE = abs(energy_viaRxx - Energy_fourierTrans).^2;
```

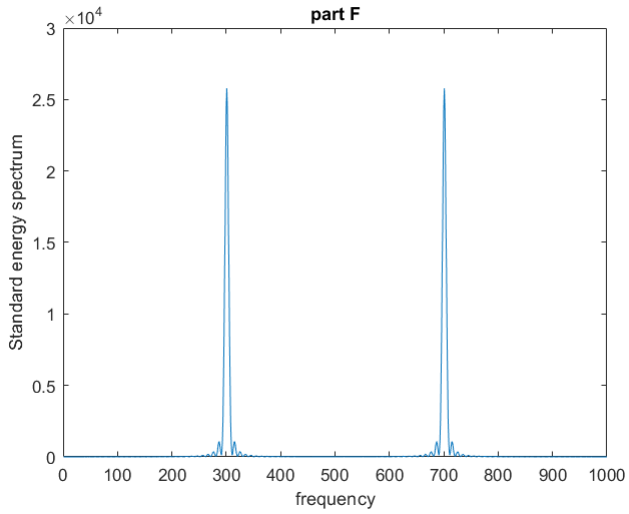


به order محور y توجه کنید همانطور که ملاحظه می کنید تفاوت این دو روش در به دست آوردن تابع چگالی طیف انرژی بسیار ناچیز است! و اگر خطای محاسبه ی متلب و محدودیت گرد/قطع کردن محاسبات در کامپیوتر وجو نمی داشت این تفاوت به صفر مطلق می گرایید لذا با چنین نتیجه ای صحت قضیه ی wiener-kinchin به عینه مشخص است.

قسمت ج:

در این قسمت ابتدا سیگنال های x1 تا xn را تولید می کنیم و سپس سیگنال سینوسی خالص تولید می کنیم و در نهایت همه این داده هارا به تابع dotDigitalWave می دهیم تا سیگنال سینوسی خالص مارا در تک تک xi ها ضرب کند و محاسبات خودهمبستگی را انجام دهد و در نهایت طبق ب تابع خودهمبستگی و چگالی طیف توان سیگنال V را محاسبه و ترسیم می کنیم.





قسمت چ:

$$\sin(wt) = (e^{jwt} - e^{-jwt})/2j$$

$$X(w) = \frac{delta(w - w_0) - delta(w + w_0)}{2j}$$

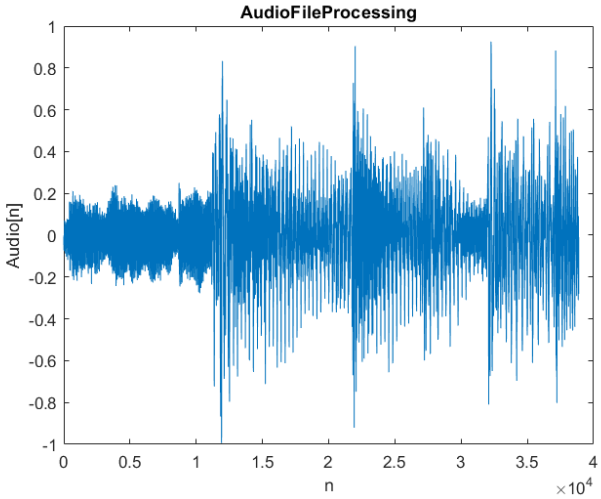
توجه کنید که ضرب کردن سینوس باعث کانولوشن در حوزه ی فرکانس با 2 تا ضربه می شود و حاصل این کار به صورت مدوله شده ی سیگنال ورودی می شود و سیگنال ورودی حول دو فرکانس w0+- قرار می گیرد و این پهنای باند فرکانسی سیگنال مبدا را به پهنای دیگری منتقل می کند که این کار بسته به شرایط و امکانات موجود باعث بهره برداری بیشتر از ابزار های موجود می گردد هم چنین با این کار می‌توانیم **اطلاعات خود را به یک فرکانس دلخواه انتقال** دهیم.این کار باعث می‌شود تا پردازش داده و مخابره کردن آن در کانال ، **در فرکانسی مطلوبی که متناسب با ویژگی‌های سیستم ماست و سیستم ما در آن بهترین عملکرد را دارد** ، اتفاق بیفتد.**هم چنین گاهی وقتا پهنای باند فرکانسی اطلاعات موجود ما با پهنای باند فرکانسی سیستم عبور متفاوت است و با این کار می توانیم این دو پهنا را با یکدیگر سینک کرده و انتقال سیگنال خود را در حالت اپتیمم ادامه دهیم**.

هم چنین توجه کنید با این کار حتی می توانیم **چند سیگنال را با هم انتقال بدهیم** بدیم بدین شرح که ابتدا هر کدام را **با ضرب در سینوسی دلخواه به باند فرکانسی خاص خود انتقال می دهیم**(توجه کنید که فرکانس هایی که سیگنال های مختلف را به آنها انتقال می دهیم باید **آنقدر از یکدیگر فاصله داشته باشند که تداخل رخ ندهد!**) سرانجام سیگنال **برآیند** حاصل را **انتقال** می دهیم و در سمت گیرنده با **قرار دادن فیلتر های متناسب با پهناهای باندی** موجود،سیگنال های اصلی را **جدا جدا بازیابی** می کنیم.

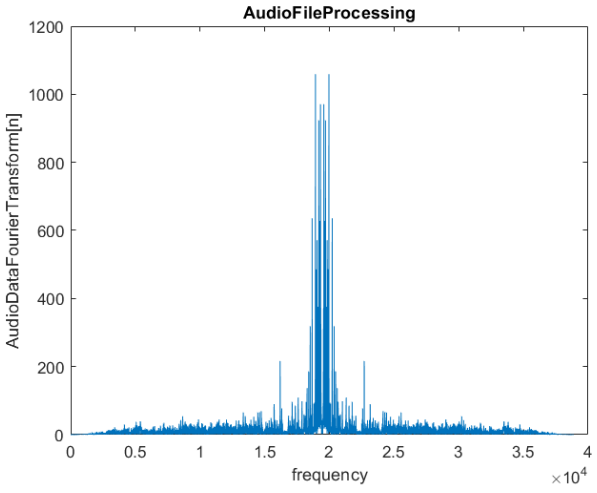
نکته ی دیگر به خاطر وجود ½ در تبدیل فوریه ی سینوس رخ می دهد اگر دقت کنید داریم که **چگالی طیف جدید دامنه اش $\frac{1}{4}$ شده است** و **دو پالس** از آن ایجاد شده است(توجه کنید در اثر **مدولاسیون دامنه ½** برابر شده و براثر **محاسبه ی چگالی از تبدیل فوریه که توان 2** آن است **دامنه چگالی ¼ شده است** پس **انرژی کل سیگنال ما نصف شده** است. پس این کار باعث می شود که **توان سیگنال انتقالی ما نصف** گردد و با **مصرف هزینه ی کمتری سیگنال اصلی** را **انتقال** بدهیم.

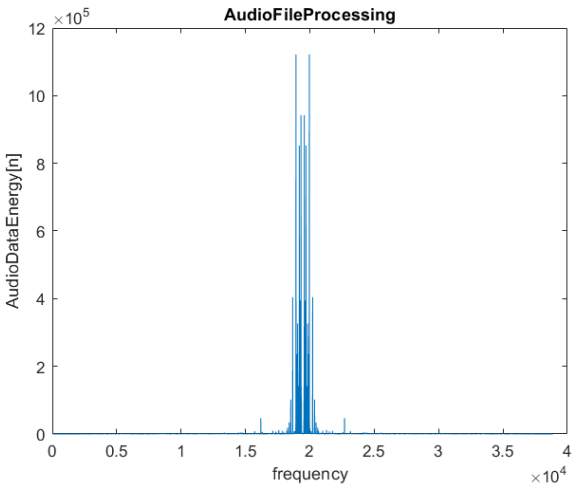
سوال دوم:

قسمت الف:ابتدا سیگنال موجود را لود می کنیم و دیتای حاصل را رسم می کنیم



و سپس تبدیل فوریه ی گسسته اش را در بازه ی منفی pi تا pi رسم می کنیم و هم چنین به کمک رابطه ی بین چگالی طیف و تبدیل فوریه آن،نمودار چگالی طیف آن را نیز رسم می کنیم.





به کمک تابع تعریفی زیر بین تبدیل فوریه ی سیگنال و چگالی طیف آن رابطه برقرار می کنیم.

```
function [out,Tsignal] = sdf(signal)
Tsignal = fourierDiscreteStandardTransform(signal)
out = Tsignal.*conj(Tsignal);
end
```

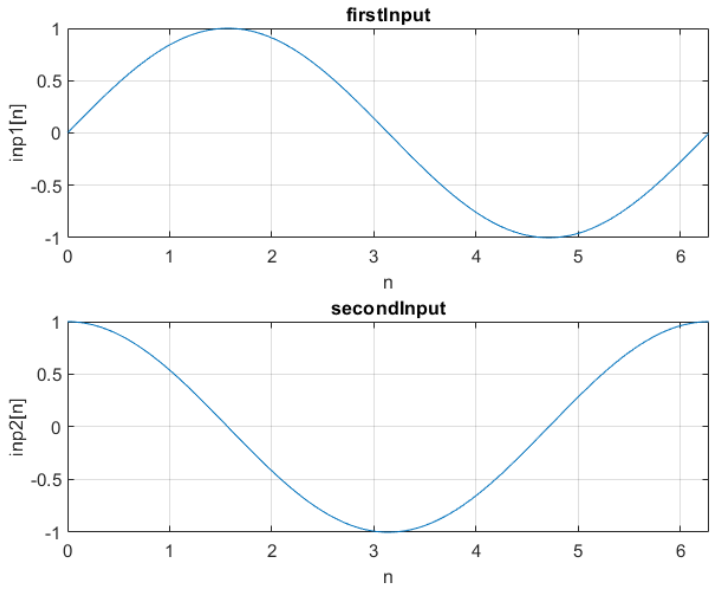
قسمت ب:

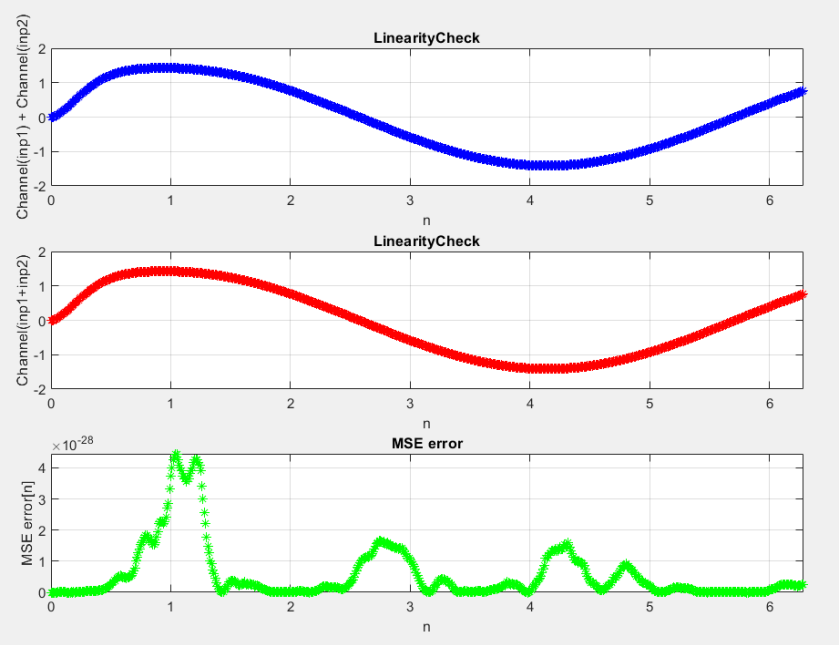
در قسمت به چک کردن 2 چیز می پردازیم

1ـ چک کردن خطی بودن:

ابتدا دو ورودی مختلف می سازیم و هر کدام را جدا به کانال می دهیم و یکبار هم جمع دو ورودی اول را یکباره به کانال می دهیم و خطای خروجی مجموع دو ورودی را از مجموع خروجی دو ورودی (MSE) را محاسبه و ترسیم می کنیم و داریم که نتایج به شرح زیر است:

$$firstInput = \sin(t) \quad secondInput = \cos(t) \quad domain = [0\ 2\pi]$$

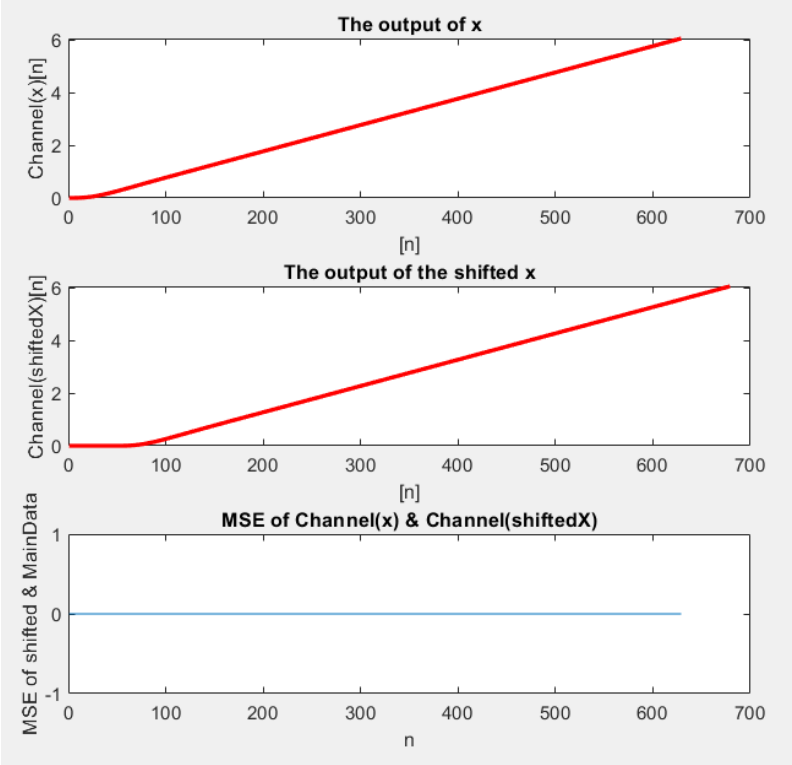




همانطور که ملاحظه می کنید MSE دو مقدار Channel(out1+out2) و Channel(out1)+Channel(out2) در حدود 10^{-28} است و این مقدار بسیار ناچیز است و اگر خطای محاسبه ی متلب و محدودیت گرد/قطع کردن محاسبات در کامپیوتر وجود نمی داشت این تفاوت به صفر مطلق می گرایید و لذا با تقریب خیلی خوبی می توان قبول کرد که کانال خطی است.

2ـ چک کردن مستقل از زمان بودن

برای اینکار ابتدا یک ورودی دلخواه x در نظر میگیریم و به کانال می دهیم و در وهله ی بعد ورودی دیگر را با شیفت دادن ورودی تولیدی اولیه به اندازه ی دلخواه(که من در کد ضمیمه شده 50 واحد زمانی درنظر گرفتم) شیفت می دهیم و سپس به کانال می دهیم و در نهایت خطای MSE شیفت یافته ی خروجی ورودی اول به اندازه ی مقدار شیفت ورودی(که من در کد ضمیمه شده 50 واحد زمانی درنظر گرفتم) از خروجی ورودی شیفت یافته را محاسبه و رسم می کنیم و همانطور که ملاحظه می کنید در نمودار سوم داریم که این خطا صفر است و لذا کانال کاملا مستقل از زمان بوده است.



با توجه به استدلال دو بخش اول و دوم داریم که کانال مخابراتی ما هم Linear و هم Time Invariant می باشد و لذا در نهایت داریم که کانال ما LTI است.

قسمت پ:

ابتدا مروری کنیم بر تعاریف phaseDelay و groupDelay.

$$x(t) = e^{i\omega t}$$

the output is

$$\begin{aligned} y(t) &= H(i\omega)\, e^{i\omega t} \\ &= \left(|H(i\omega)|e^{i\phi(\omega)}\right)\, e^{i\omega t} \\ &= |H(i\omega)|\, e^{i(\omega t + \phi(\omega))} \end{aligned}$$

where the phase shift ϕ is

$$\phi(\omega) \stackrel{\text{def}}{=} \arg\{H(i\omega)\}\,.$$

Additionally, it can be shown that the group delay, τ_g, and phase delay, τ_ϕ, are frequency-dependent, and they can be computed from the phase shift ϕ by

$$\tau_g(\omega) = -\frac{d\phi(\omega)}{d\omega}$$

$$\tau_\phi(\omega) = -\frac{\phi(\omega)}{\omega}\,.$$

حال رابطه ی دیگری برای groupDelay محاسبه می کنیم. برای محاسبه تأخیر گروه ، ابتدا ثابت می‌کنیم که

$$(H(w) = A(w)e^{j\varphi(w)}) : gd(w) = Re\left\{\frac{j\frac{d}{dw}H(w)}{H(w)}\right\} :$$

$$Re\left\{\frac{j\frac{d}{dw}H(w)}{H(w)}\right\} = Re\left\{\frac{j\frac{d}{dw}\left(A(w)e^{j\emptyset(w)}\right)}{A(w)e^{j\emptyset(w)}}\right\} =$$

$$Re\left\{\frac{j\left(A'(w)e^{j\emptyset(w)} + A(w)j\emptyset'(w)e^{j\emptyset(w)}\right)}{A(w)e^{j\emptyset(w)}}\right\} = Re\left\{\frac{-A(w)\emptyset'(w)e^{j\emptyset(w)} + jA'(w)}{A(w)e^{j\emptyset(w)}}\right\}$$

$$= \frac{-A(w)\emptyset'(w)e^{j\emptyset w}}{A(w)e^{j\emptyset w}} = -\emptyset'(w) \rightarrow \quad gd(w) = Re\left\{j\frac{d}{dw}\frac{H(w)}{H(w)}\right\}$$

لذا داریم که در نهایت فرمول group delay به صورت زیر در می آید:

$$\mathrm{gd}(\omega) = \mathbf{Re}\left\{\frac{j\frac{d}{d\omega}H(\omega)}{H(\omega)}\right\}$$

می دانیم که مشتق گیری در حوزه فرکانس معادل است با ضرب در زمان در حوزه ی زمان حال(خاصیت زیر).

Differentiation in Frequency $nx[n]$	$j\frac{dX(e^{j\omega})}{d\omega}$

حال با رهیافت زیر و نکات بالا به محاسبه ی group Delay می پردازیم.

1ـ ابتدا ورودی ضربه ای به اندازه ی دامنه ای برابر با سیگنال کلیپ تولید می کنیم(که طبیعتا در صفر یک و در سایر نقاط صفر است)

2ـ سپس این ورودی را به کانال می دهیم و می دانیم که خروجی کانال عبارت است از پاسخ ضربه سیستم h[n]

3ـ حال طبق این رابطه که مشتق در حوزه ی فرکانس معادل ضرب در زمان در حوزه ی زمان است groupDelay را به صورت بخش Real تبدیل فوریه ی nh[n] تقسیم بر تبدیل فوریه ی h[n] به دست می آوریم،به عبارت زیر توجه کنید:

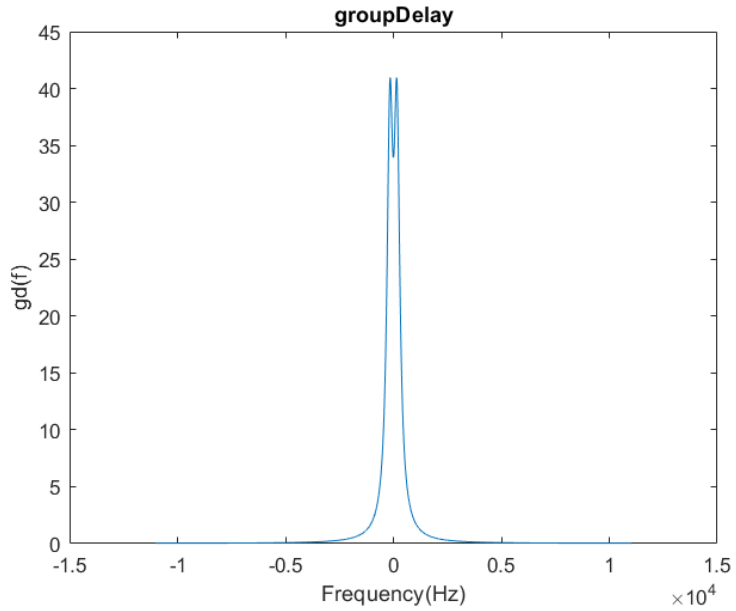
$$gd(f) = \frac{Re\left\{\frac{F\{nh[n]\}}{F\{h[n]\}}\right\}}{2\pi}$$

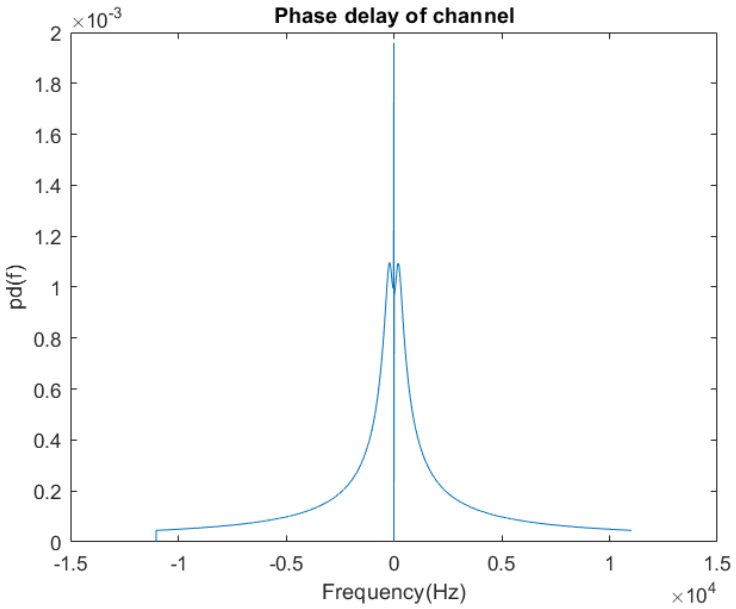
1ـ داریم که gd(f) محاسبه شده است،حال در قدم بعدی باید phaseDelay را محاسبه کنیم،برای انجام این کار کافیست که تبدیل فوریه ی h[n] به دست آمده در قسمت قبل را بر f تقسیم کنیم و نتیجه ی حاصل phaseDelay ما خواهد بود.

- نکته ی مهم: به کمک رابطه ی زیر ما دامنه را بین [-frequency/2,frequency/2] می اندازیم

```
f = (0:size(groupDelay,2)-1)*(frequency)/(size(groupDelay,2)-1) - frequency/2;
```

نمودارها:





اثرات روی سیگنال خروجی:

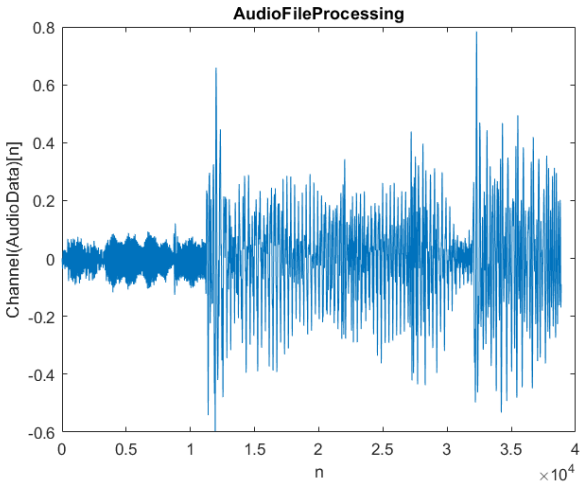
همان طور که مشاهده می‌شود ، تأخیر فاز یک خط ثابت نمی‌باشد و به همین دلیل اعوجاج فاز داریم. این اعوجاج فاز به این صورت است که فرکانس‌های کم را دچار شیفت حدود 1.2 واحدی در حوزه‌ی زمان می‌کند و هنگامی که فرکانس افزایش می‌یابد ، این شیفت زمانی به سمت 0 می‌رود و در فرکانس‌های بالا ، دیگر اختلاف فازی بین ورودی و خروجی وجود نخواهد داشت.از آنجا که شیفت اندکی در فاز وجود دارد می توان گفت که اعوجاج فاز کانال اندک است و قابل صرف نظر کردن است.

اگر به اندازه پاسخ فرکانسی کانال دقت کنیم ، متوجه می‌شویم که اندازه آن در فرکانس 0 ، تقریباً 1 است و با افزایش فرکانس این نمودار اندازه به صورت مثلثی از طرفین نقطه ی صفر شروع به افت می کند و در نهایت صفر می‌شود.
بنابراین اندازه پاسخ فرکانسی کانال در همه‌ی فرکانس‌ها یکسان نیست و باتوجه به اینکه با دور شدن از صفر دامنه ی فرکانسی افت می کند انگار کانال ما یک فیلتر پایین‌گذر است که در باند عبور تقویت همواری ندارد و **لذا اعوجاج دامنه ی کانال زیاد است.**

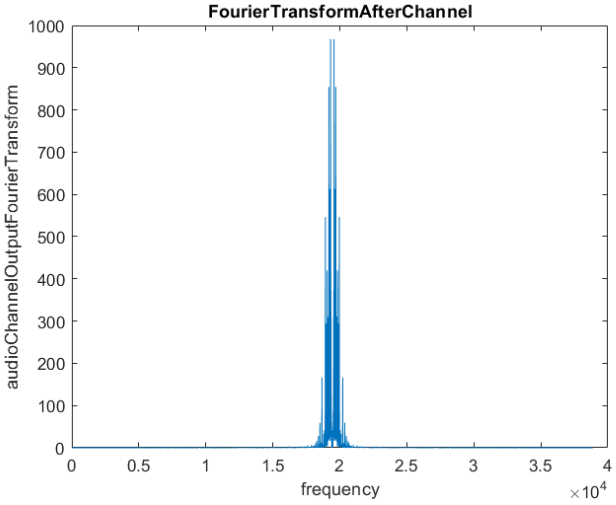
قسمت ت:

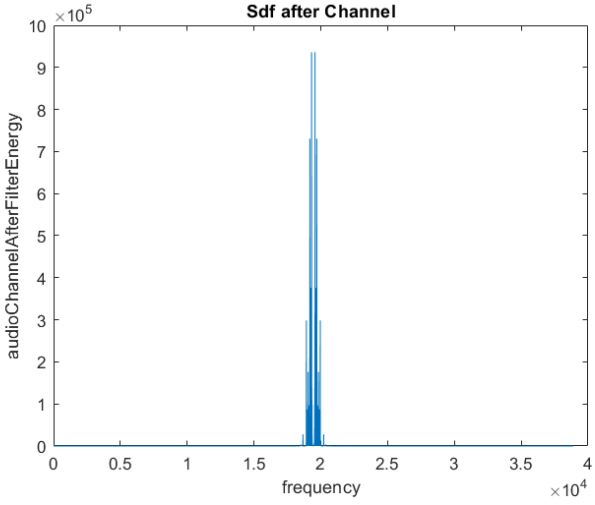
سیگنال کلیپ داده شده را لود می کنیم و به ورودی کانال می دهیم و نمودار حوزه ی زمان و تبدیل فوریه و چگالی طیف انرژی آن را رسم می کنیم.

نمودار زیر،نمودار در حوزه ی زمان کلیپ داده شده است.



نمودار های زیر تبدیل فوریه و چگالی طیف کلیپ خروجی کانال اند.





قسمت ث:

برای حل این قسمت از equalizer استفاده می کنیم،همانطور که می دانیم که سیستمی که دارای اعوجاج نباشد در نهایت خروجی اش دارای scale و شیفت زمانی است نسبت به ورودی اش و این یعنی اینکه نسبت تبدیل فوریه ی خروجی به ورودی به صورت عددی ضرب در نمایی ای است حال این بدان معناست که اگر ما واحدی به نام equalizer پس از کانال خود بگذاریم حاصل ضرب تبدیل فوریه ی کانال ما و equalizer به صورت عددی ضرب در نمایی است.

$input \rightarrow Channel(f) \rightarrow Equalizer(f) \rightarrow output$ (1)

$output(f) = Equalizer(f)Channel(f)$input(f) (2)

$System\ with\ no\ Distortion : output(t) = Ainput(t - t_0)$ (3)

$So\ output(f) = Ae^{2\pi ft0}input(f)$(4)

$Conclusion : (2),(4) \rightarrow Equalizer(f)Channel(f) = Ae^{2\pi ft0}$

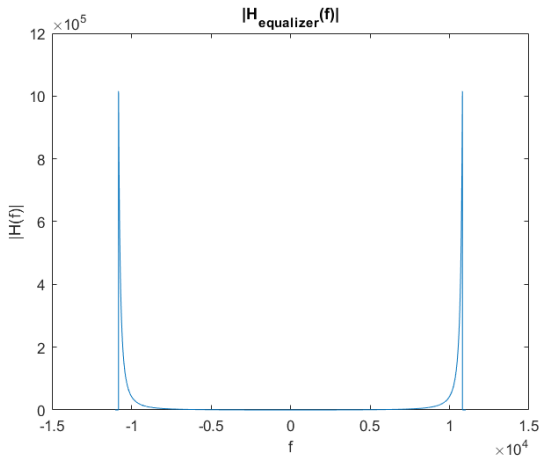
لذا داریم که در نهایت:

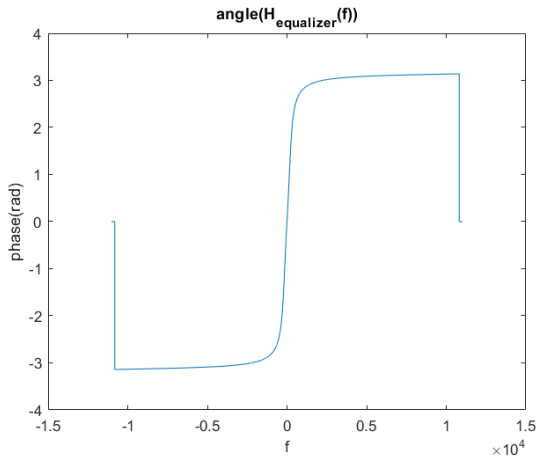
$$Equalizer(f) = \frac{Ae^{2\pi ft0}}{Channel(f)}$$

از آنجا که در صورت سوال اشاره ای به مقدار scale و shift نشده است جهت سهولت انجام کار فرض می کنیم که A برابر یک و مقدار t0 برابر صفر است و معادله ی Equalizer به صورت زیر در می آید.

$$Equalizer(f) = \frac{1}{Channel(f)}$$

حال به رسم نمودار پاسخ فرکانسی واحدEqualizer می پردازیم:

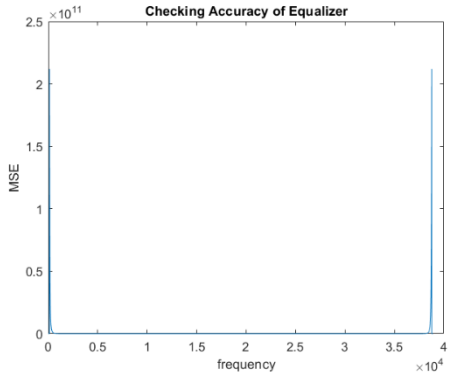




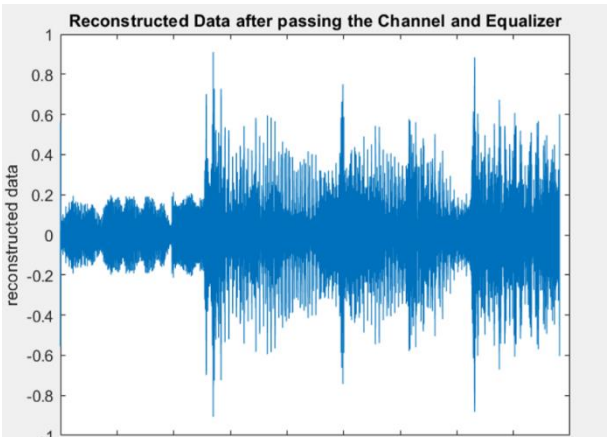
اما توجه کنید که من یک ترشهولدی در نظر گرفتم برای equalizer که این equalizer که خود فیلتری میانگذر است تا چه فرکانسی از پاسخ فرکانسی به دست آمده از کانال را فیلتر کند و از چه حدی به بعد کوچکتر را ignore کند.

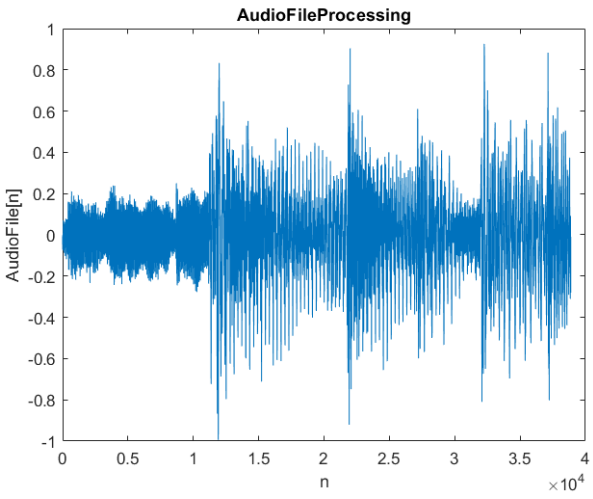
```
threshHoldValue = 0.0000001;
gonnaRemoveIndexes = find(abs(H(floor(length(H)/2):end)) < threshHoldValue*max(abs(H)));
```

لذا در بازه های خارج از gonnaRemoveIndex را فیلتر دیگر رد نمی کند.حال نمودار MSE را در زیر مشاهده می کنید: همانطورکه ملاحظه می کنید به جز باز های انتهایی(هرچقدر `threshHoldValue` را کمتر و نزدیک تر به صفر بگیرید داریم که MSE به صفر کامل بیشتر می گراید)خطا در سایر نقاط صفر است اما به علت خطای نسبتا موجود در انتهای باند فرکانسی این فیلتر می تواند در مواقعی خیلی هم خوب نباشد و خود این واحد **طبیعتا تاثیراتی در تغییر دادن سیگنال** نسبت به **اولیه** آن در کنار **بازیابی برخی تغییرات اعمال** شده ی **کانال** دارد.



حال نمودار حوزه ی زمان سیگنال ورودی و سیگنال خروجی اکولایزر را(به کمک گرفتن ifft) به دست می آوریم ونتیجه ی حاصل را مشاهده می کنیم:





قسمت ج:

برای محاسبه ی صفر ها و قطب های سیستم از آنجا که ما معادله ی سیستم را به صورت صریح نداریم(به صورتsyms نداریم) و لذا نمی توانیم فعلا از توابع معروف محاسبه ی صفر،قطب اعم از residue و ... استفاده کنیم،اما متلب toolbox عی به اسم systemIdentification دارد که یک ورودی و یک خروجی سیستم را می گیرد و سیستم را برایمان مشخص می کند.

از آنجا که ما وارون اثر تبدیل کانال را به عنوان اکولایزر در نظر گرفتیم در پنجره ی systemIndentification مقادیر را به شرح زیر دادیم :

ابتدا داده هارا به شرح زیر در پنجره ی systemIdentification اد می کنیم:

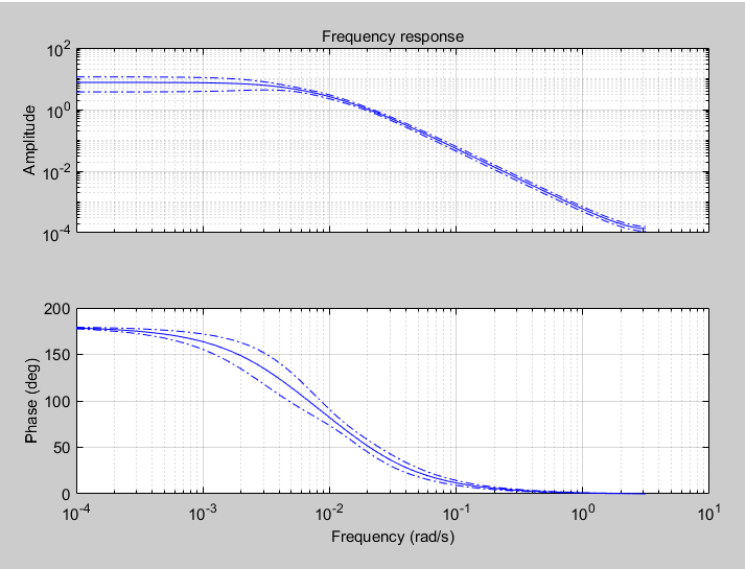
Import Data -> **input = audio Channel Output**

Output = audio Data

سپس انتخاب می کنیم از پنجره ی Transfer Function models < - estimate و سپس از قسمت پایین سمت راست انتخاب می کنیم نمودار های پاسخ فرکانسی و نمودار صفر ـ قطب تبدیل فوریه به دست آمده را.

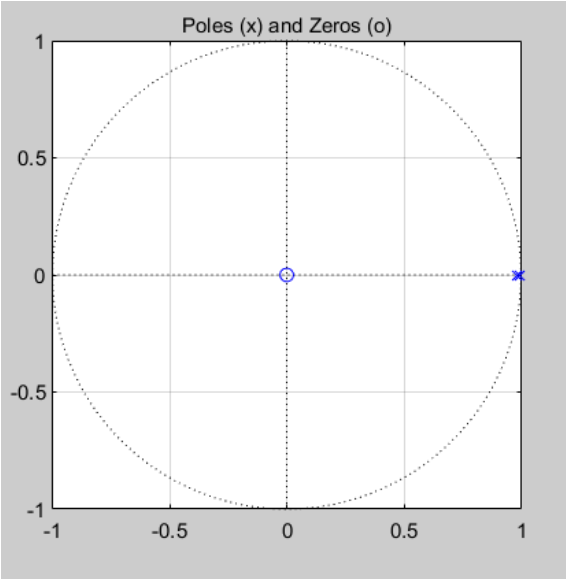
نکته: از آنجا که ابزار های پاسخ فرکانسی و نمودار صفر ـ قطب در پنجره ی systemIdentification موجود است لذا به جای زدن کد اضافه و لود کردن دیتا از امکانات این پنجره استفاده می کنیم.

نمودار پاسخ فرکانسی:



نمودار صفر ـ قطب تبدیل سیستم:

توجه کنید که سیستم 2 قطب و یک صفر دارد و در نمودار شکل زیر قطب ها به علت نزدیکی در کنار هم به صورتی دیده شده اند که انگار یک عدد قطب داریم حال انکه ما برای بهتر شدن شبیه سازی خود در هنگام انتخاب ویژگی های تابع تبدیل برگزیدیم که 2 عدد قطب و یک عدد صفر داشته باشیم.



با توجه به نمودار های به دست آمده داریم که این فیلتر علی نیست و لذا داریم که نمی توان آن را به صورت مستقیم ساخت و باید با ترفند های دیگری این فیلتر را به تکه های تقریبا علی تقسیم کرده و استفاده کرد.

نکته:بررسی و مقایسه ی سیگنال خروجی فیلتر(اکولایزر) با سیگنال اصلی در قسمت قبل انجام شد و خطای MSE آن ها هم ترسیم شد.

سوال سوم:

قسمت الف:

یافتن پاسخ ضربه فیلتر

پاسخ ضربه فیلتر هیلبرت را به کمک خواص دوگنای تبدیل فوریه به دست می آوریم و نکته ی زیر به دست می آوریم:

$$H_Q(f) = -j\,\mathrm{sgn}\,f = \begin{cases} -j & f > 0 \\ +j & f < 0 \end{cases} \qquad \textbf{(1a)}$$

$$h_Q(t) = \frac{1}{\pi t} \qquad \textbf{(1b)}$$

We obtain this result by applying duality to $\mathcal{F}[\mathrm{sgn}\,t] = 1/j\pi f$ which yields $\mathcal{F}[1/j\pi t] = \mathrm{sgn}\,(-f) = -\mathrm{sgn}\,f$, so $\mathcal{F}^{-1}[-j\,\mathrm{sgn}\,f] = j/j\pi t = 1/\pi t$.

همانطور که مشاهده می کنید پاسخ ضربه در مقادیر منفی مقدار غیر صفر دارد لذا این تابع تبدیل **علی** نیست.

جهت بررسی پایدار بودن فیلتر بررسی می کنیم که آیا پاسخ ضربه سیستم مطلقا جمع پذیر است یا خیر،داریم:

$$\int_{-\infty}^{+\infty} |h(t)|dt = \int_{-\infty}^{\infty} \left|\frac{1}{\pi t}\right| dt = \infty$$

لذا چون این فیلتر مطلقا جمع پذیر نیست لذا این فیلتر **پایدار نیست**.

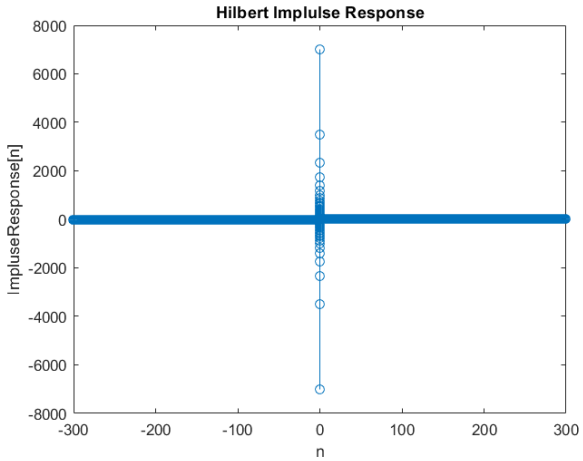
قسمت ب:

نمودار پاسخ ضربه ی فیلتر هیلبرت($\frac{1}{\pi t}$) را رسم می کنیم:برای این کار تابعی در کد خود نوشته ام که با گرفتن ۲ ورودی ای که عبارت اند از ۱ ــ فرکانس نمونه برداری و ۲ ــ طول سیگنال هیلبرت در خروجی اش سیگنال هیلبرت گسسته را برمی گرداند.

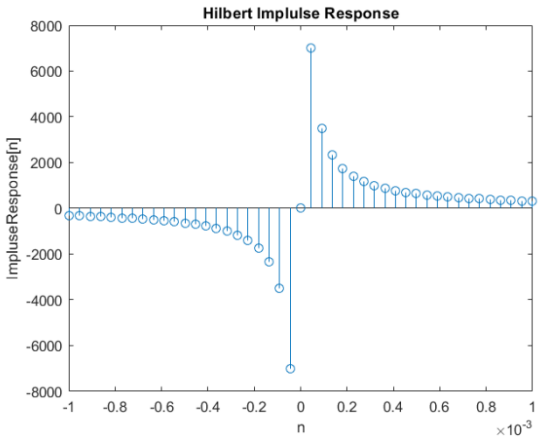
```
function [out,zeroIndex,domain] = hilbert(length,frequency)
domain = (-1)*length/2 + 1/(2*frequency) :1/frequency:length/2 + 1/(2*frequency);
signal = zeros(1,size(domain,2));
zeroInit = ceil(size(domain,2)/2);
signal(1:end) = 1./(pi*domain(1:end));
zeroIndex = zeroInit;
out = signal;
end
```

حال به کمک تابع فوق به ترسیم نمودار های لازم می پردازیم.

نکته: حول صفر اندازه ی مقادیر تابع به شدت نسبت به سایرنقاط افزایش پیدا می کند.



اگر محدوده ی n را کوچکتر کنیم نمودار حالت هموگرافیک خود را بهتر نشان می دهد.



قسمت پ:

برای انجام این قسمت مراحل زیر را در کد پیاده سازی کردیم

1- تولید سیگنال هیلبرت

a. می دانیم که حول صفر داده های به شدت **بزرگ** اند،حال سوال اینجاست که **صفر** را چه مقداری بگذاریم؟ من اومدم گفتم چون مقدار **صفر** به صورت **حدی خیلی بزرگ** می شود لذا طوری **سمپلنیگ رو انجام بدهم که 0 در نقاط سمپلنیگ قرار نگیرد** این باعث شد که **دامنه من خیلی بهتر** شد **اما فازم اندکی در این حالت جالب نبود**،ولی درحالتی که صفر را ایگنور می کردم و مقدارش را صفر قرار می دادم باعث می شد که **فاز خیلی قشنگ بشه ولی اندازه به شدت damp می کرد(چون مقادیر بزرگی در حول صفر و من جمله خود صفر ایگنور می شدند)** لذا اومدم اثر اون شیفت اولیه رو با ضرب کردن تابع تبدیل در نمایی با وزن w(که در کد گفته ام چه است) اون **شیفت رو اثرش رو در فاز خنثی کردم و در نهایت نمودار فاز ام هم خیلی بهتر شد.**

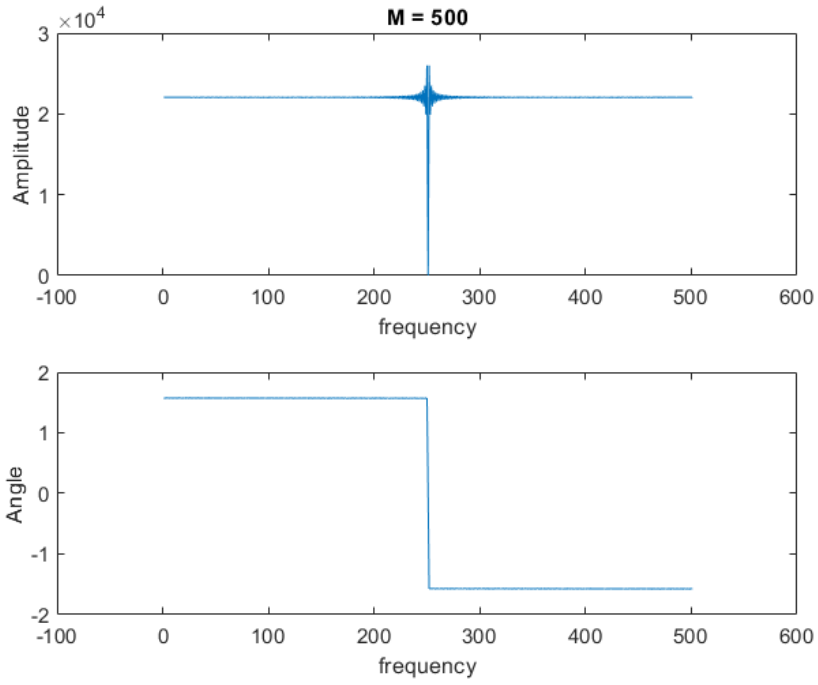
2- شیفت دادن سیگنال هیلبرت جهت علی کردن آن به منظور اعمال فیلتر بر روی تمام محدوده ی منفی تا مثبت

3- ضرب کردن فیلتر مربعی در هیلبرت شیفت یافته

4- تبدیل فوریه استاندارد گرفتن از سیگنال قسمت 3

5- رسم نمودار های اندازه و فاز تابع تبدیل به دست آمده در قسمت 4

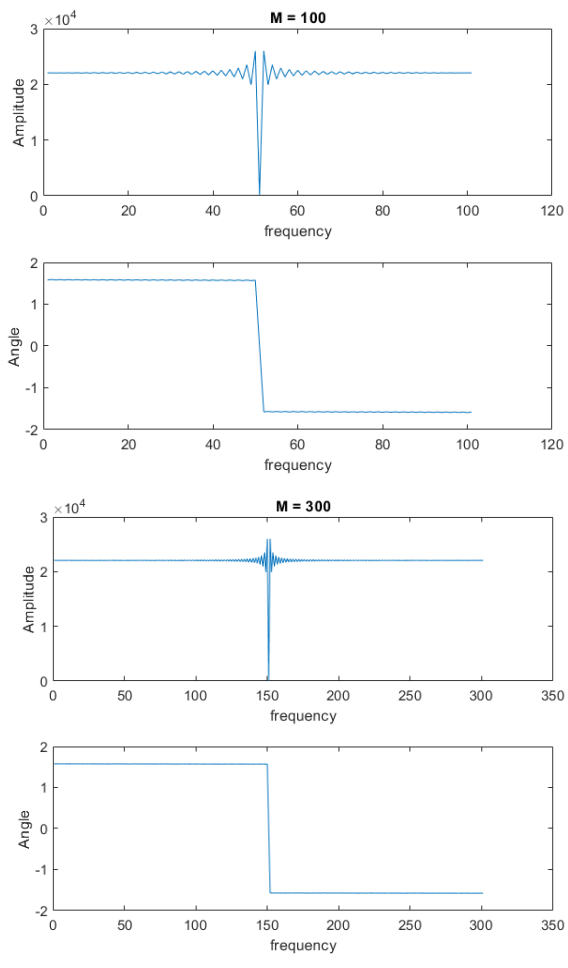
نمودار های اندازه و فاز:

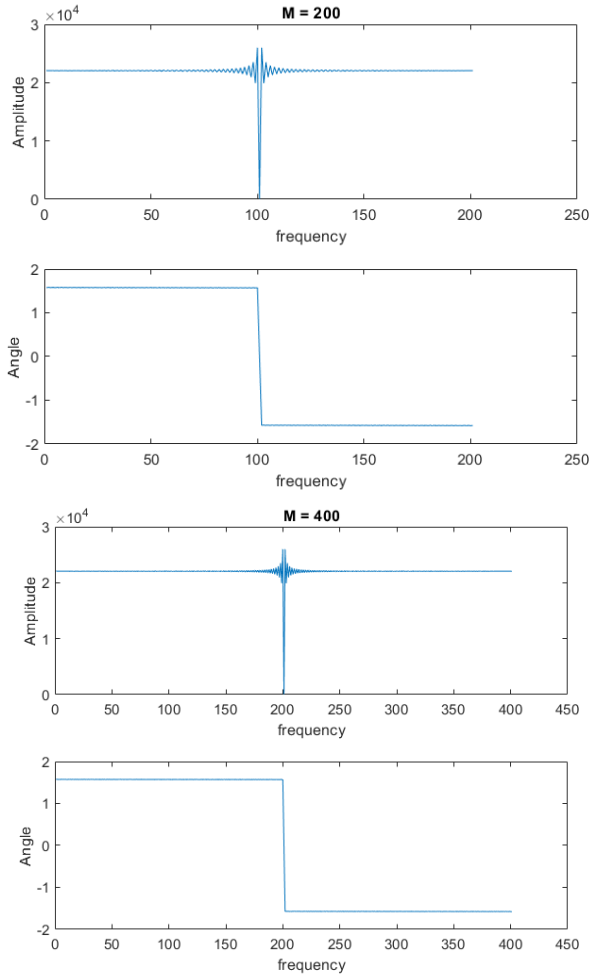


قسمت ت:

از آنجا که هر سیگنالی که بخواهیم در متلب بر روی آن پردازش بزنیم به نوعی finite است و پس از اندیسی دیگر صفر می شود طبیعتا هر چه قدر طول پنجره ی بزرگتر شود به علت در برگرفتن نقاط بیشتر خواهیم داشت که تایج به دست آمده بیشتر به حالت ایده آل فیلتر هیلبرت نزدیک خواهید بود(چون همانطور که میدانید پاسخ زمانی فیلتر هیلبرت به صورت نامتناهی از منفی بی نهایت تا مثبت بی نهایت ادامه دارد و با افزایش پنجره اعضای بیشتری را در برمیگیرد لذا نتیجه به هیلبرت نامتناهی نزدیک تر می گردد).

نمودار ها: مقدار M را از 100 تا 400 افزایش می دهیم.





قسمت ث:

برای انجام این قسمت مراحل زیر را در کد پیاده سازی کردیم

1- تولید سیگنال هیلبرت

a. می دانیم که حول صفر داده های به شدت **بزرگ** اند،حال سوال اینجاست که **صفر** را چه مقداری بگذاریم؟ من اومدم گفتم چون مقدار **صفر** به صورت **حدی خیلی بزرگ** می شود لذا طوری **سمپلنیگ رو انجام بدهم که 0 در نقاط سمپلنیگ قرار نگیرد** این باعث شد که **دامنه من خیلی بهتر** شد **اما فازم اندکی در این حالت جالب نبود**،ولی درحالتی که صفر را ایگنور می کردم و مقدارش را صفر قرار می دادم باعث می شد که **فاز خیلی قشنگ بشه ولی اندازه به شدت damp می کرد(چون مقادیر بزرگی در حول صفر و من جمله خود صفر ایگنور می شدند)** لذا اومدم اثر اون شیفت اولیه رو با ضرب کردن تابع تبدیل در نمایی با وزن w(که در کد گفته ام چه است) اون **شیفت رو اثرش رو در فاز خنثی کردم و در نهایت نمودار فاز ام هم خیلی بهتر شد.**

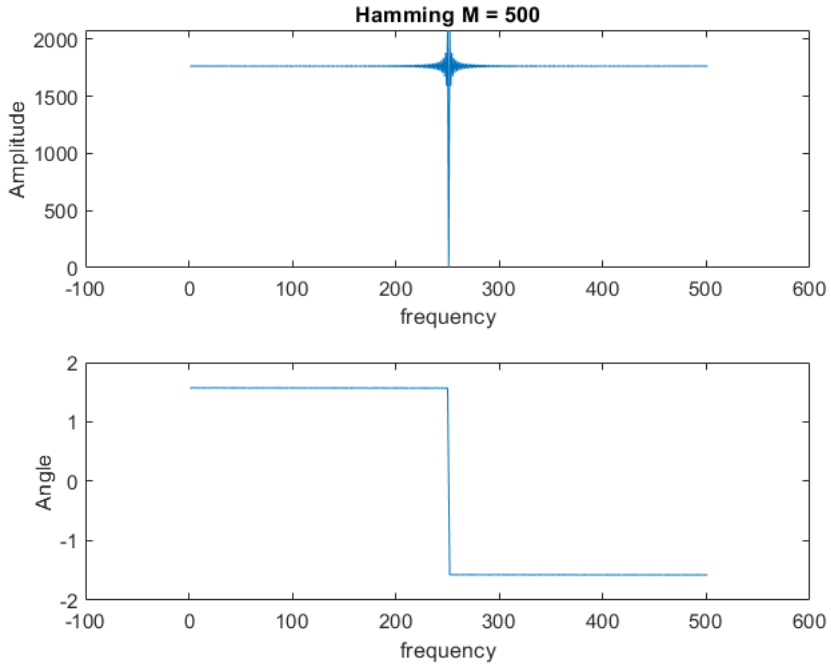
2- شیفت دادن سیگنال هیلبرت جهت علی کردن آن به منظور اعمال فیلتر بر روی تمام محدوده ی منفی تا مثبت

3- تولید فیلتر Hamming و شیفت دادن آن(به همان اندازه ای که سیگنال اصلی را شیفت دادیم) جهت قابلیت اعمال بر تمامی محدوده ی منفی تا مثبت سیگنال هیلبرت قسمت 2

4- ضرب کردن فیلتر Hamming یا همان پنجره ی Hamming در هیلبرت شیفت یافته

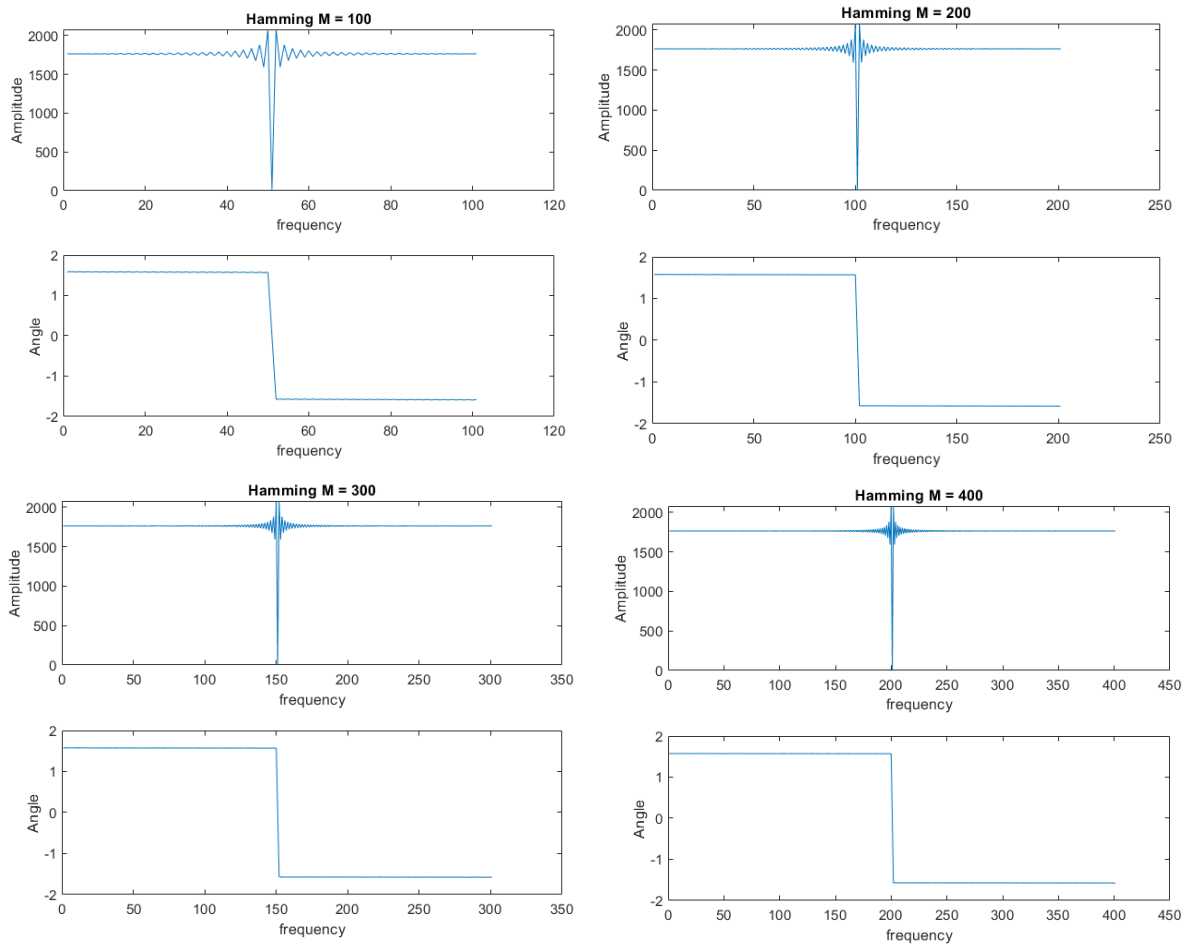
5- تبدیل فوریه استاندارد گرفتن از سیگنال قسمت 4

6- رسم نمودار های اندازه و فاز تابع تبدیل به دست آمده در قسمت 5



بررسی تاثیر سایز پنجره در پاسخ فرکانسی Hamming filter

از آنجا که هر سیگنالی که بخواهیم در متلب بر روی آن پردازش بزنیم به نوعی finite است و پس از اندیسی دیگر صفر می شود طبیعتا هر چه قدر طول پنجره ی بزرگتر شود به علت در برگرفتن نقاط بیشتر خواهیم داشت که تایج به دست آمده بیشتر به حالت ایده آل فیلتر هیلبرت نزدیک خواهید بود(چون همانطور که میدانید پاسخ زمانی فیلتر هیلبرت به صورت نامتناهی از منفی بی نهایت تا مثبت بی نهایت ادامه دارد و با افزایش پنجره اعضای بیشتری را در برمیگیرد لذا نتیجه به هیلبرت نامتناهی نزدیک تر می گردد).



قسمت ج:

همانطور که در نمودار های صفحات قبل مشاهده می کنید،نمودار فاز تبدیل هیلبرت در هر دو حالت پنجره ی hamming و حالت rectangle با کیفیت خیلی نزدیک به واقعی به دست آمد،اما توجه کنید که در حالت Hamming نوسانات حول فرکانس صفر در نمودار اندازه بسیار تعدیل پیدا کرده است و مصداق نزدیک به واقع تری از فیلتر هیلبرت است،لذا به کمک پنجره ی Hamming فیلتر نزدیک به واقع تری از هیلبرت را شبیه سازی کردیم.

و در قسمت بعد هم با استفاده از پنجره ی Kaiser،این شباهت به حداکثر خود می رسد و نوسانات حول صفر فیلتر هیلبرت کاملا برطرف شده و مصداق بسیار نزدیک به واقع تری از فیلتر هیلبرت با اوزان Kaiser به دست می آید و این هم به لطف توزیع زنگوله ای حول صفر فیلتر Kaiser است.

قسمت امتیازی:

ابتدا گریزی بزنیم به تابع تولید اوزان فیلتر Kaiser در داک متلب، داریم:

Algorithms

The coefficients of a Kaiser window are computed from the following equation:

$$w(n) = \frac{I_0\left(\beta\sqrt{1-\left(\frac{n-N/2}{N/2}\right)^2}\right)}{I_0(\beta)}, \quad 0 \le n \le N,$$

where I_0 is the zeroth-order modified Bessel function of the first kind. The length $L = N + 1$. Thus `kaiser(L,beta)` is equivalent to `besseli(0,beta*sqrt(1-(((0:L-1)-(L-1)/2)/((L-1)/2)).^2))/besseli(0,beta)`.

To obtain a Kaiser window that designs an FIR filter with sidelobe attenuation of α dB, use the following β.

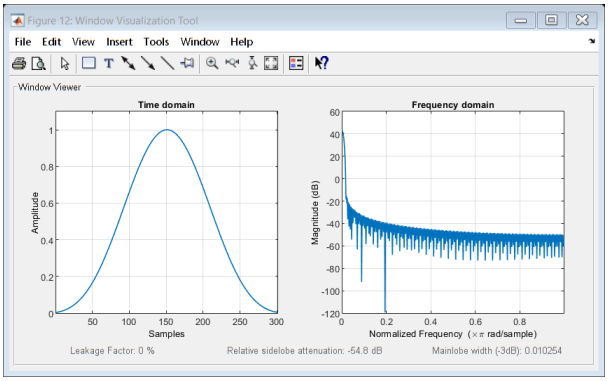
$$\beta = \begin{cases} 0.1102(\alpha - 8.7), & \alpha > 50 \\ 0.5842(\alpha-21)^{0.4} + 0.07886(\alpha-21), & 50 \ge \alpha \ge 21 \\ 0, & \alpha < 21 \end{cases}$$

Increasing β widens the mainlobe and decreases the amplitude of the sidelobes (i.e., increases the attenuation).

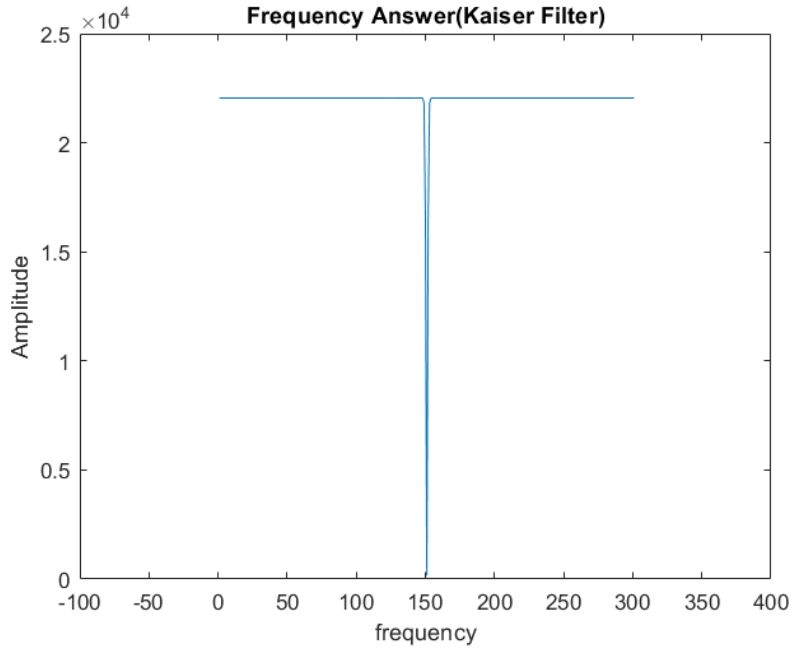
حال برای انجام مطالبات سوال مراحل زیر را به ترتیب انجام می دهیم:

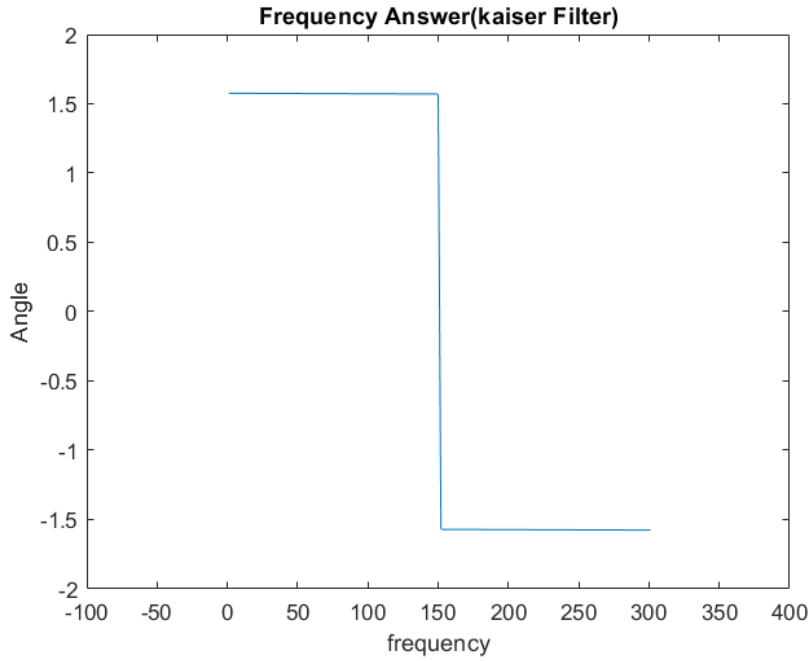
1- تولید سیگنال هیلبرت
2- شیفت دادن سیگنال هیلبرت جهت علی کردن آن به منظور اعمال فیلتر بر روی تمام محدوده ی منفی تا مثبت
3- تولید فیلتر kaiser و شیفت دادن آن(به همان اندازه ای که سیگنال اصلی را شیفت دادیم) جهت قابلیت اعمال بر تمامی محدوده ی منفی تا مثبت سیگنال هیلبرت قسمت 2
   - تولید فیلتر Kaiser به کمک دستور Kaiser(length,Beta)
   - رسم مشخصات فیلتر به کمک دستور wvtool
4- ضرب کردن فیلتر kaiser یا همان پنجره ی kaiser در هیلبرت شیفت یافته
5- تبدیل فوریه استاندارد گرفتن از سیگنال قسمت 4
6- رسم نمودار های اندازه و فاز تابع تبدیل به دست آمده در قسمت 5

ابتدا مشخصات فیلتر Kaiser را به کمک دستور wvtool رسم می کنیم.



نمودار های اندازه فاز:





اندرکرامات فیلتر Kaiser:

همانطور که مشاهده می کنید فیلتر Kaiser نسبت به فیلتر Rectangle از دقت بیشتری برخوردار است اما چرا؟ توجه کنید که تابع هیلبرت در نقطه ی صفر از اهمیت ویژه ای برخوردار است چرا که در اپسیلون مقدار سمت راست نقطه ی صفر سیگنال عددی بسیار بزرگی دارد و در اپسیلون مقدار سمت چپ نقطه ی صفر نیز سیگنال مقداری بسیار منفی دارد و در نقطه ی صفر نیز مقدار سیگنال صفر است،لذا نقاط همسایگی صفر از اهمیتی بیشتری برخوردارند تا نقاط با فاصله ی بیشتر از صفر،چرا که هر چه از صفر دورتر می شویم اندازه ی مقادیر سیگنال به شدت افت می کند(حتی نمودار کشیده شده برای تابع هیلبرت در قسمت ب همین سوال نیز این نکته را یادآوری می کند)،خب حال ما که سیگنال هیلبرت را ا فیلتری رد می کنیم خوب است این فیلتر این حساسیت تابع به نقطه ی صفر و همسایگی اطرافش را حفظ کند،اما همانطور که می دانید فیلتر مربعی وزن برابری به همه ی نقاط عبور خود(وزن 1) می دهد و لذا کمکی در حفظ حساسیت سیگنال هیلبرت نمی کند،اما فیلتر kaiser با توجه به نحوه ی تعریف این فیلتر به نقاط در همسایگی صفر وزن بیشتری را نسبت می دهد تا نقاط با فاصله ی بیشتر از صفر(اگر به نمودار کشیده شده در قسمت مشخصات فیلتر Kaiser در بالا مراجعه کنید خواهید دید که این فیلتر شکل گوسی دارد و وزن بیشتری به نقاط حول صفر ما نسبت می دهد و به نقاط با فاصله ی بیشتر وزن کمتر می دهد و این خاصیت این فیلتر باعث می شود که دقت و حساسیت سیگنال هیلبرت حول نقطه ی صفر بیشتر حفظ شود و این باعث می شود که نمودار های اندازه فاز سیگنال هیلبرت ضرب شده در فیلتر Kaiser،نرمی و دقت بیشتری دارند نسبت به حالتی که فیلتر rectangle یا hamming است.