



دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

Principles of Communication Systems

CA1

Parsa Darban

810100141

بخش اول (عیب یابی)

قسمت الف)

ابتدا با استفاده از دستور Load فایل داده شده در صورت پروژه را فرا میخوانیم.

با توجه به اینکه نمودار ما در ۲ ثانیه دارای طول ۲۰۱ است، فرکانس نمونه برداری برابر ۱۰۰.۵ هرتز است. ($f=length/t$) سپس آن ها را دو حوزه زمان پلات می کنیم.

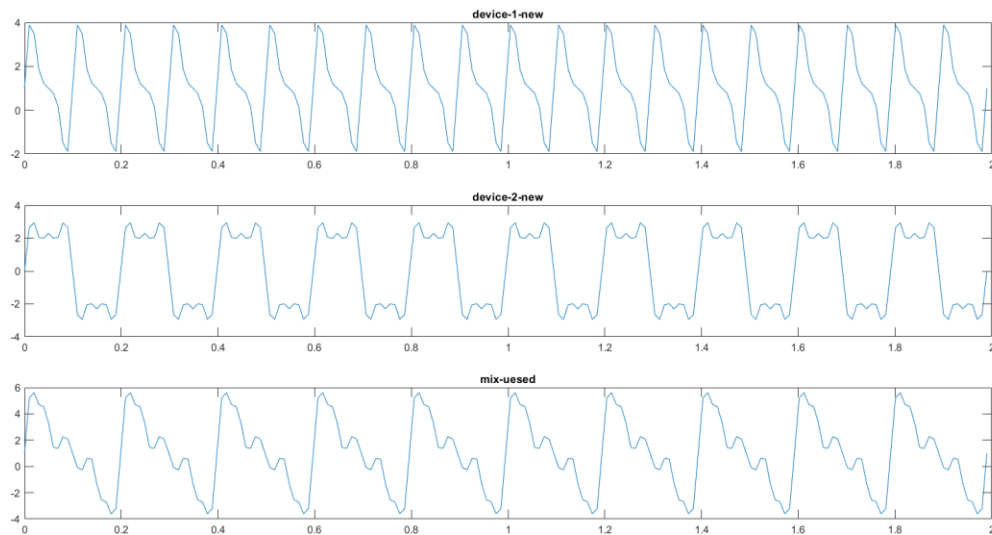


Figure 1.1(plot signals in time)

fftshift

این تابع در متلب مولفه فرکانس های صفر تبدیل فوریه را (fft) به مرکز طیف فرکانس انتقال میدهد. این کار ضربه ها را به مرکز (در کد $f=0$) انتقال میدهد.

حال با استفاده از fft و fftshift اندازه و فاز آن را در بازه 50 : 50- رسم می کنیم.

نتایج به شرح زیر است:

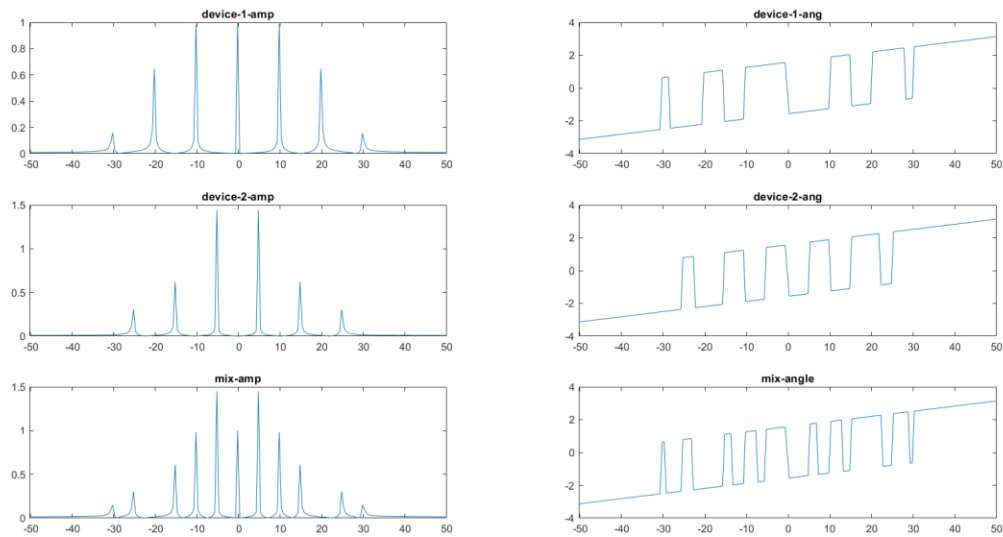


Figure 1.2(plot in frequency)

قسمت ب)

در این بخش می‌خواهیم در حوزه زمان دو دستگاه را باهم مقایسه کنیم.

با توجه به صورت پروژه ابتدا سیگنال‌های دو دستگاه را از هم کم می‌کنیم سپس با سیگنال‌های مربوط مقایسه می‌کنیم.

روش‌های متفاوتی برای مقایسه کردن سیگنال‌ها وجود دارد. یکی از روش‌ها استفاده از cross correlation است. زیرا یکی از ویژگی‌های این تابع در ریاضی نشان دهنده شباهت بین دو سیگنال است.

نتایج کد به شکل زیر است

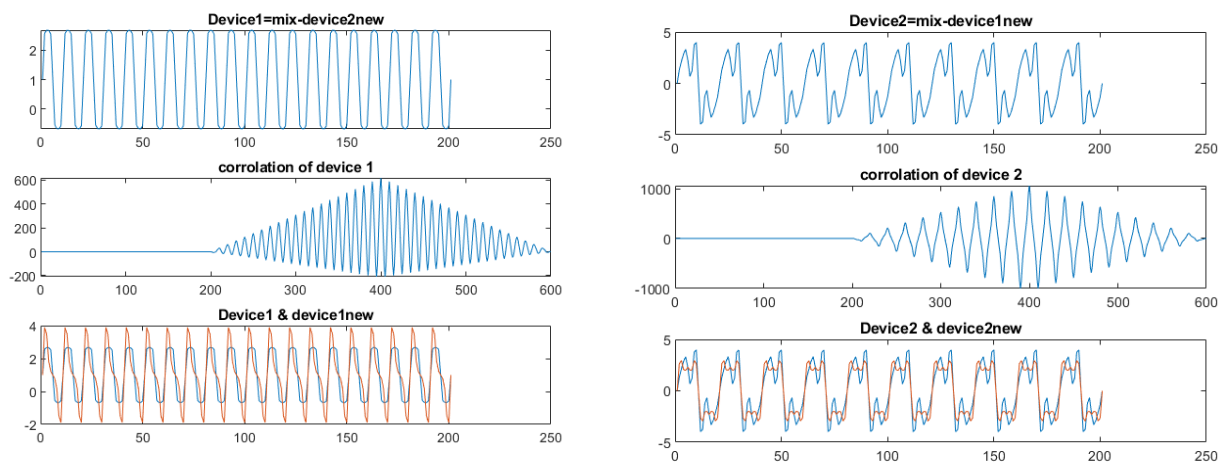


Figure 1.3(comparing in time)

حال با توجه به پیک نمودار همبستگی سیگنال ها ، در مولفه میانی ، مشاهده می شود پیک در سیگنال دستگاه اول کمتر از پیک در دستگاه دوم است و این نشان میدهد سیگنال تفاضل و سیگنال سالم دستگاه یک با یکدیگر شباهت زیادی ندارند.

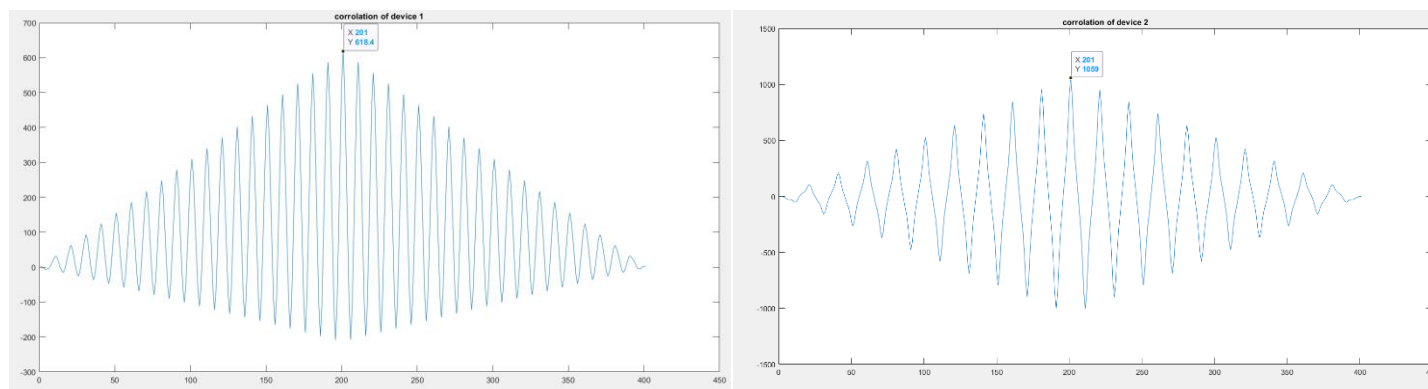


Figure 1.4(peak in correlation)

پس با توجه به نتایج دستگاه دو در حوزه زمان سالم است.

این روش به علت اینکه دقت کافی را ندارد اصلا مناسب نیست. زیرا برد دو سیگنال با یکدیگر متفاوت است و این موضوع به صورت غیر مستقیم در پیک cross correlation اثر میگذارد و همانطور که مشخص است در پلاتی که دو نمودار بر روی یکدیگر رسم شده است ، اختلاف برد آنها میتواند در نتیجه اثر بگذارد.

اما در ادامه در حوزه فرکانس که به مراتب روش بهتری برای مقایسه است ، استفاده می کنیم.

قسمت ج)

در این قسمت همانند قسمت دوم عمل میکنیم و این دفعه در حوزه فرکانس سیگنال ها را از هم کم میکنیم و با سیگنال مربوط در حوزه فرکانس مقایسه می کنیم.

نتایج به صورت زیر است:

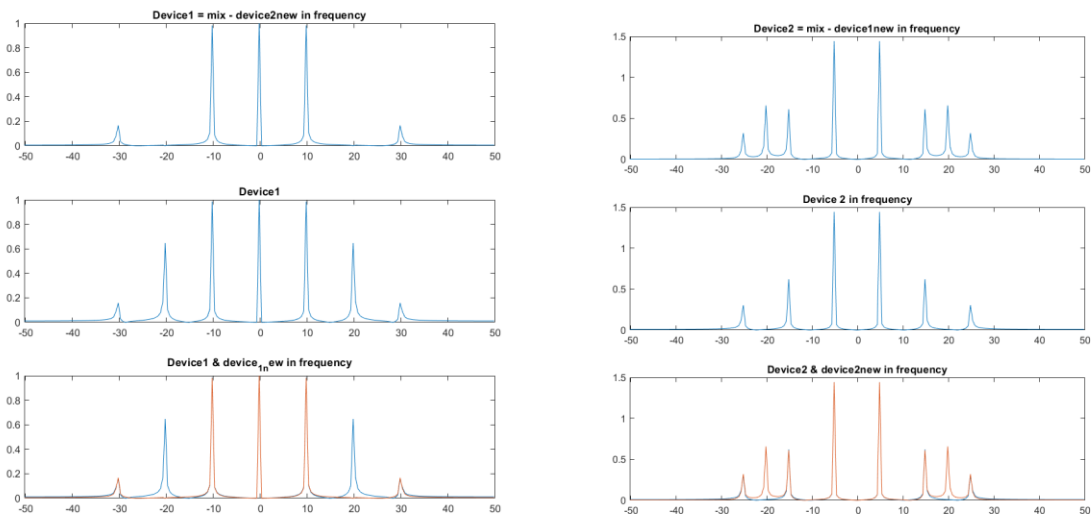


Figure 1.5(compare in frequency)

همانطور که ملاحظه می شود دستگاه یک در فرکانس های مثبت و منفی ۲۰ دارای پیکی است که این پیک در هنگام محاسبه تفاضل از بین رفته . این موضوع نشان میدهد دستگاه یک که در فرکانس های مشخصی باید عمل خاصی را انجام دهد (برای مثال در ۲۰ هرتز) اما در تفاضل (که صدای آن در میکس است) در این فرکانس تقریباً به صفر میل میکند.

اما در دستگاه دو کل پیک های فرکانس در صدای سالم آن در تفاضل با دامنه های متفاوت وجود دارد که این موضوع نشان دهنده وجود یک gain در آن است.

همچنین برای بهبود بهتر میتوان اندازه سیگنال های یک و دو را از اندازه سیگنال mix کم کرد.

که در شکل زیر به وضوح آشکار است که در فرکانس ۲۰ هرتز در قسمت منفی پیکی وجود دارد که حاصل از تفاضل mix و سیگنال یک است و این نشان میدهد که مشکل از سیگنال یک است که در سیگنال mix در همان ۲۰ هرتز هیچ اثری نداشته.

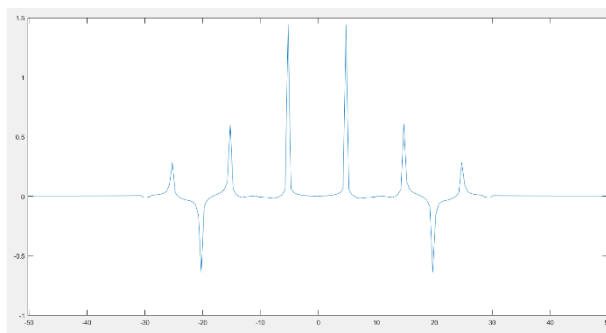


Figure 1.6

با استدلال های ذکر شده دستگاه دو در حوزه فرکانس سالم است.

در حوزه فرکانس بهتر است زیرا دو دستگاه در فرکانس های مشخص باید پیک ها و صداهایی را تولید میکردند و اگر هر دو دستگاه سالم بودند ، در فرکانس های برابر باید در جمع آن ها پیک برابر یا با gain متفاوت در mix نشان داده میشد.

بخش دوم (اعوجاج و همسان ساز)

پاسخ فرکانسی آن به روش دستی :

$$H_c(f) = 0.7 e^{-2\pi j f (0.17)} + 0.9 e^{-2\pi j f (0.55)}$$

با توجه به آنکه $t_d=0$, $K=1$ حاصل ضرب پاسخ فرکانسی سیستم اعوجاج دار و همسان ساز باید برابر با یک باشد.

پس پاسخ فرکانسی برابر است با :

$$H_{eq}(f) = \frac{1}{H_c(f)} = \frac{1}{0.7 e^{-2\pi j f (0.17)} + 0.9 e^{-2\pi j f (0.55)}}$$

اعوجاج از نوع فاز داریم زیرا حاصل تاخیر گروهی عدد ثابت نیست

$$\tau_{gc} = -\frac{1}{2\pi} \left(\frac{d}{df} \text{ang}(H_c(f)) \right) = ((0.119)j * e^{-2\pi j f (0.17)} + (0.495)j * e^{-2\pi j f (0.55)})$$

اما میتوان گفت در سیستم کلی (H_c , H_{eq}) اعوجاج نداریم زیرا مشتق آن صفر (عدد ثابت) است.

تابع دلتا دیراک را ابتدا رسم کرده و تبدیل فوریه آن را رسم میکنیم. (برای رسم بهتر فرکانس نمونه برداری ۱۰۰ و بازه زمانی را از ۰ تا ۲ ثانیه در نظر گرفتیم)

نتیجه به شرح زیر است:

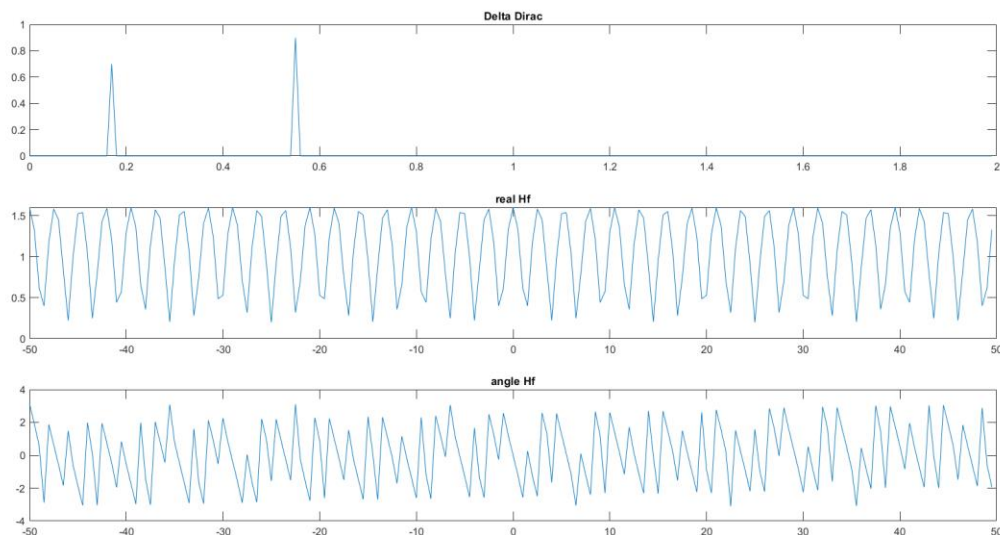


Figure 2.1

حال کانولوشن سیگنال ورودی و کانال را محاسبه میکنیم و نام آن را z_t میگذاریم.

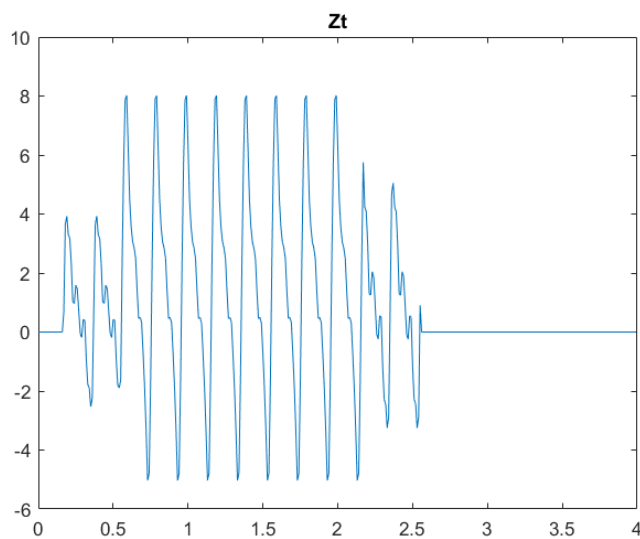


Figure 2.2

برای طراحی همسان ساز از دو روش استفاده شده.

راه اول:

با توجه به آنکه $K=1$, $t_d=0$ است، یعنی شیفت زمانی و بهره در خروجی نداریم و خروجی کلی سیستم دقیقاً همان ورودی کل است.

حال اگر z_t ورودی همسان ساز باشد و خروجی آن $x_t(\text{mixed_used})$ میتوانیم با تقسیم x_f بر z_f (در حوزه فرکانس) همسان را ایجاد کنیم.

برای اینکار فوریه z_t و x_t را محاسبه میکنیم و تقسیم را انجام میدهیم.

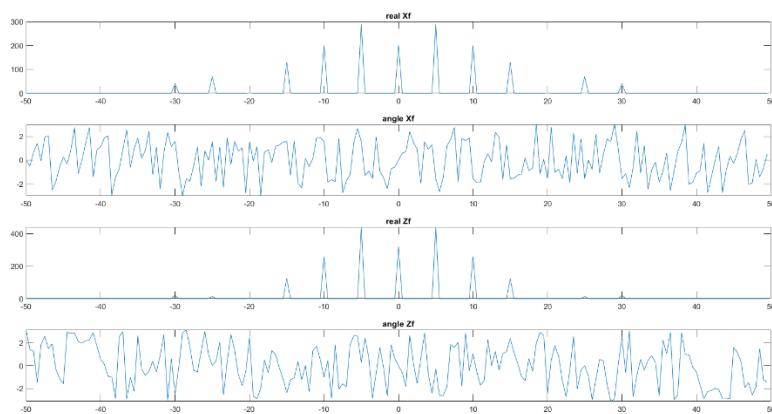


Figure 2.3(fourier of x_t and z_t)

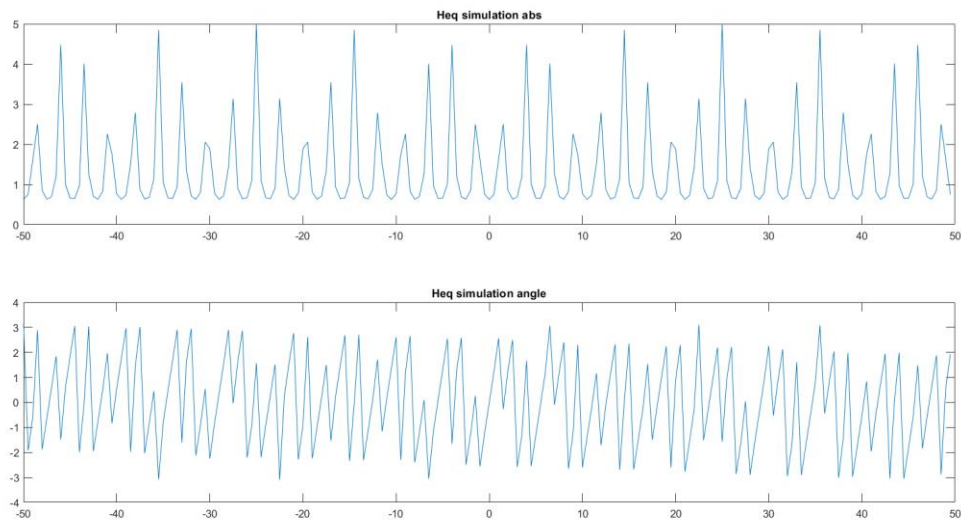


figure 2.4(Heq with first route)

راه دوم:

با توجه به آنکه حاصل ضرب دو سیستم درهم (با فرض داده شده در صورت پروژه) برابر با یک می باشد. پس پاسخ فرکانسی همسان ساز برابر با $1/H_f$ است.

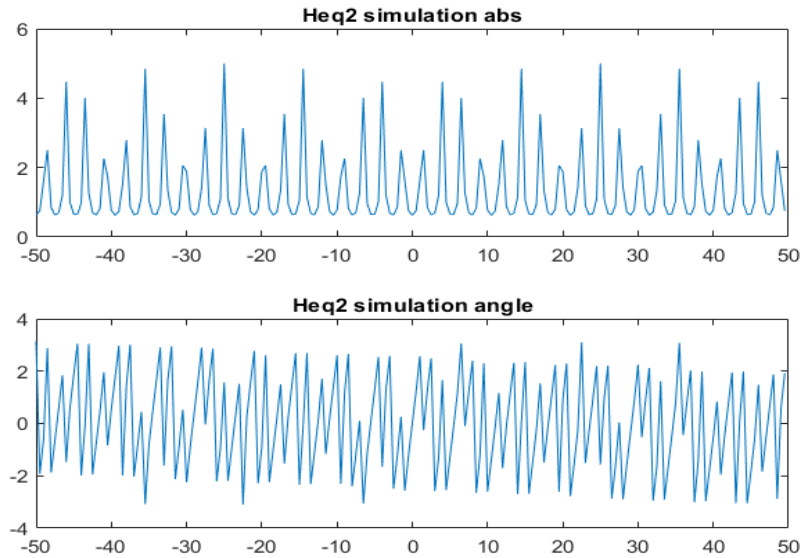


Figure 2.5(Heq with second route)

همچنین برای محاسبه دستی که بالاتر نوشته شد داریم:

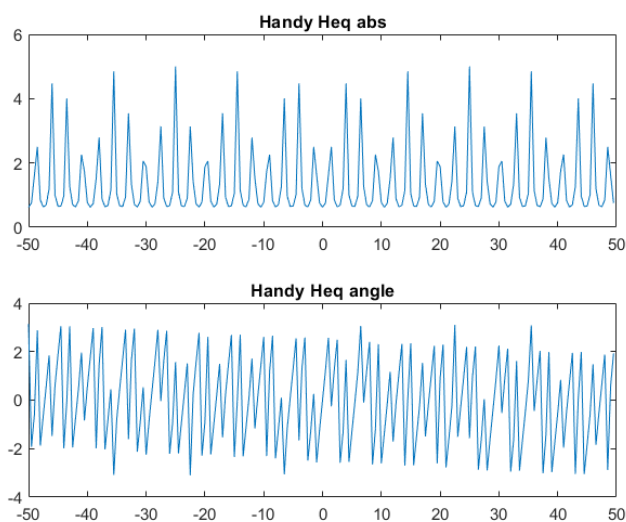


Figure 2.6(Handy Heq)

حال هر سه را باهم پلات میکنیم و مشاهده میشود که با هم برابر هستند.

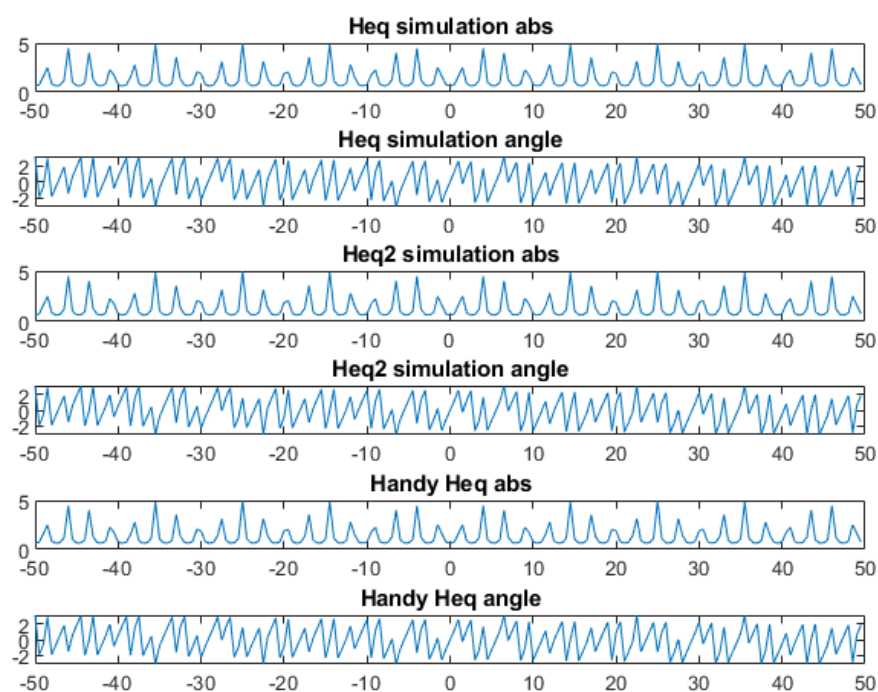


Figure 2.7(comparing all equalizers in frequency)

حال آن ها را در حوزه زمان رسم میکنیم و باز مشاهده می شود باهم برابر هستند.

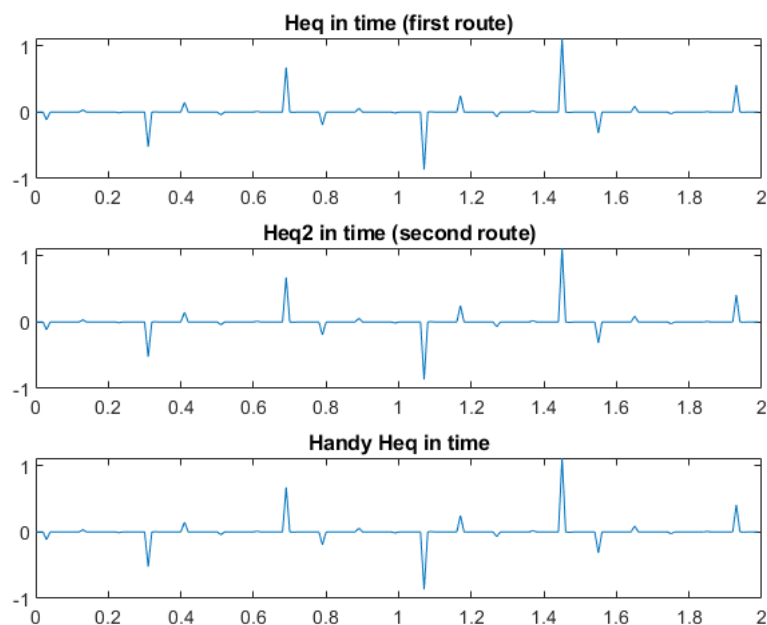


Figure 2.8 (comparing all equalizers in time)

حال میخواهیم چک کنیم که آیا همسان ساز ما درست کار میکند یا خیر.

برای این کار Z_f را Heq ضرب میکنیم و پاسخ آن باید با $X_f(\text{mixed used fourier})$ برابر باشد.

این کار در کد یکبار Heq و بار دیگر با Heq_{exp} انجام شده و نتایج به شکل زیر است.

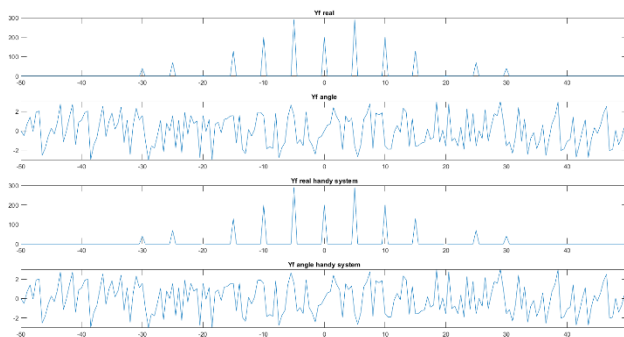


Figure 2.9 (comparing $Y_f(\text{output})$ with two equalizer)

در این قسمت میخواهیم در حوزه زمان پاسخ را مقایسه کنیم پس فوریه معکوس دو پلات بالا را گرفته و باید آن را با پلات $mixed_Used$ مقایسه کنیم اگر باهم یکسان بودند همسان سازهای تولید شده درست هستند.

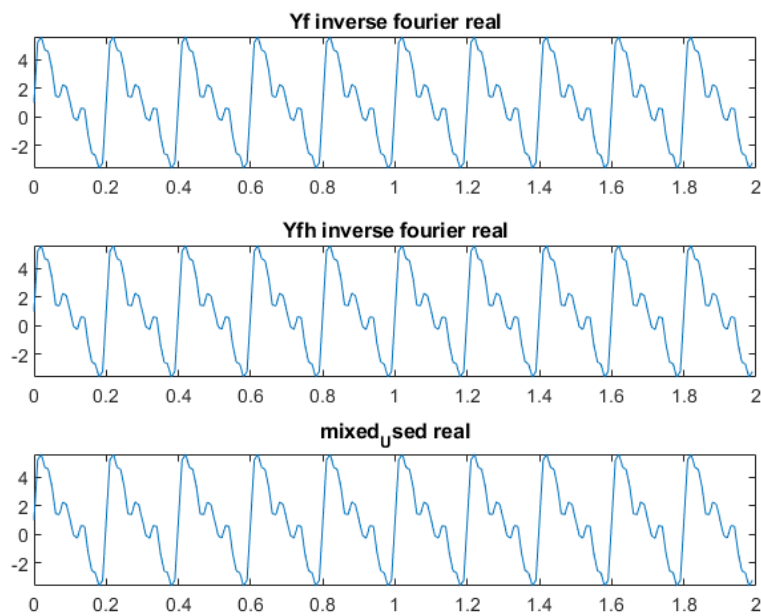


Figure 2.10(comparing Y(output(2 first plot)) with mixed_Used)

همانطور که ملاحظه شد باهم برابر هستند و همسان ساز ها درست هستند.

در قسمت بعدی باید محاسبات ریاضی را با توابع نوشته شده نوشت و immse آن را محاسبه کرد.

در قسمت اول که مربوط به همبستگی میشود.

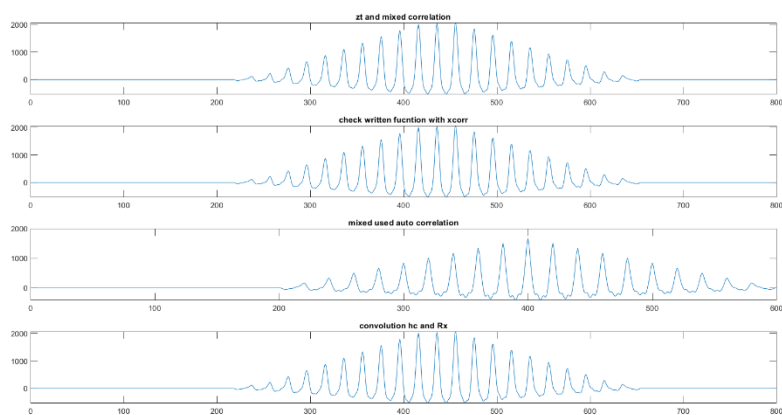



Figure2.11

`ERR=immse(Rxy,Rxy1);`

 ERR

5.6174e-26

figure 2.11 خانه دوم :

این خانه برای مقایسه کد نوشته شده برای همبستگی با `xcorr` متلب است که نتایج برابر است.

آخر گزارش درباره کد نوشته برای `cross correlation` توضیح داده می شود.

و برای قسمت آخر این بخش چگالی طیف را محاسبه میکنیم. برای این کار از ابتدا از فوریه `auto correlation` استفاده کردم که جواب را به طور حدودی درست میداد اما `immse` را به علت برابر نبودن طول آرایه نمیداد. (آخر کد Q2 کامنت شده).

پس از فرمول مقابل برای اینکه درگیر افزایش طول هنگام `correlation` نباشم استفاده کردم.

$$G_v(f) = (|v(f)|)^2$$

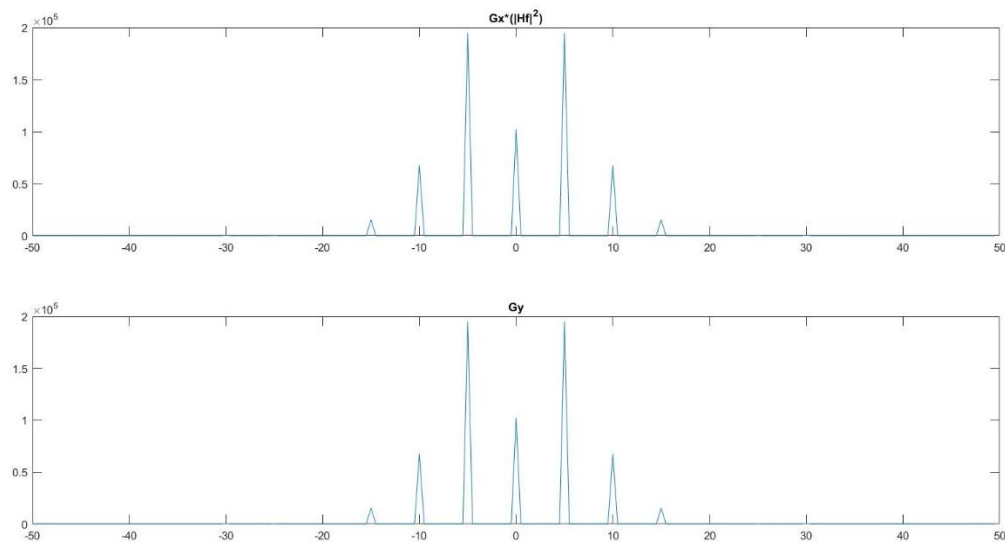



Figure 2.12

`ERRR=immse(Gxxx,Gyy);`

 `ERRR`

4.2683e-24

بخش سوم (نمودار زمان فرکانس)

STFT سیگنال را به بخش های کوچکتری تقسیم میکند و سپس برای هر بخش، تبدیل فوریه را محاسبه می کند تا بتوانیم تغییرات زمان و فرکانس را هم زمان در یک پلات مشاهده کنیم.

این کار زمان های شروع و پایان و تغییرات فرکانس و طیف آن را به ما میدهد.

نمودار تابع S_1 در زمان و فرکانس به شکل زیر است.

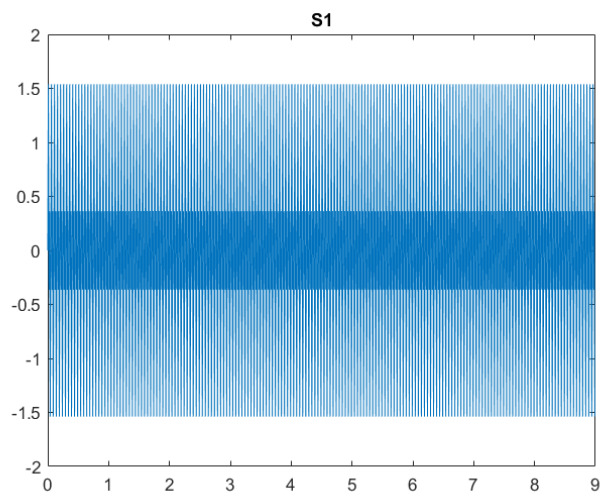


Figure 3.1 (S_1 in time)

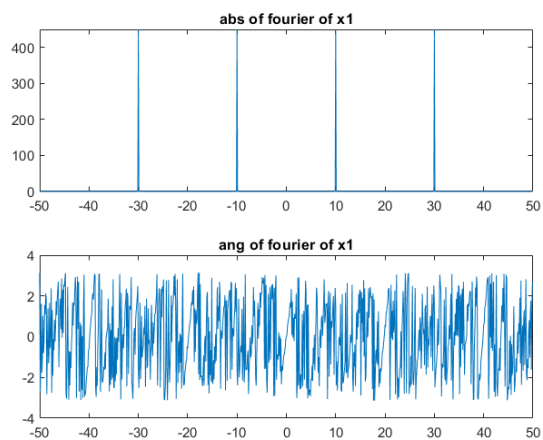


Figure 3.2 (S_1 in frequency)

همچنین S_2 همانند نمودار های زیر میشود.

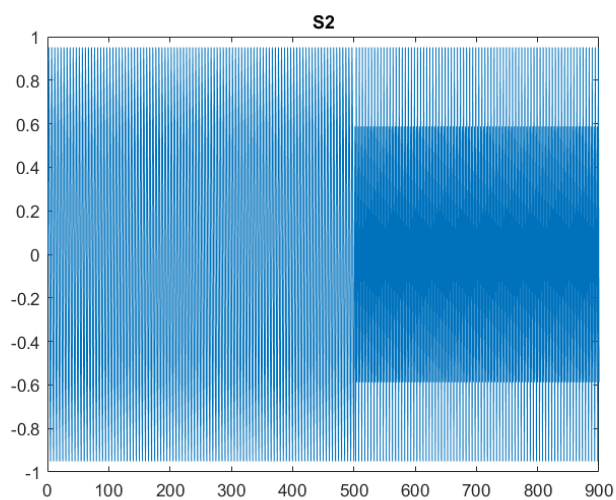


Figure 3.3(S_2 in time)

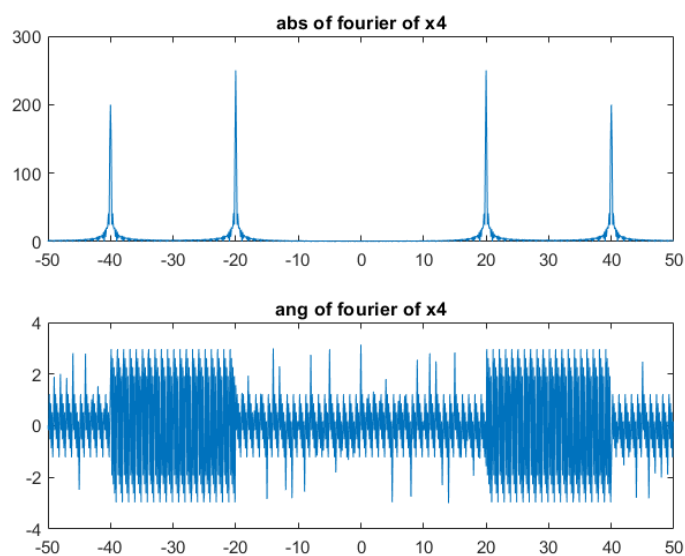


Figure 3.4(S_2 in frequency)

حال باید در این بخش از تابع spectrogram استفاده کنیم.

این تابع کار STFT را در متلب برای ما انجام میدهد.

در آرگومان های آن دو مولفه مهم وجود دارد.

یکی از آن ها window هستش که با تقسیم سیگنال به بخش های کوتاه رزولوشن فرکانسی را مشخص میکند. با افزایش آن رزولوشن فرکانس بالا میرود اما باعث افزایش مدت زمان نیز میشود.

یکی دیگر از آرگومان ها overlap هست که به معنای همپوشانی میان Windows هستش. با کاهش آن نمونه برداری برای فوریه کاهش میابد و رزولوشن زمان را افزایش میدهد.

با این ویژگی های تعریف شده با توازن بین زمان و فرکانس را با تغییر اعداد ایجاد کرد.

بهترین حالت تولید شده در کد:

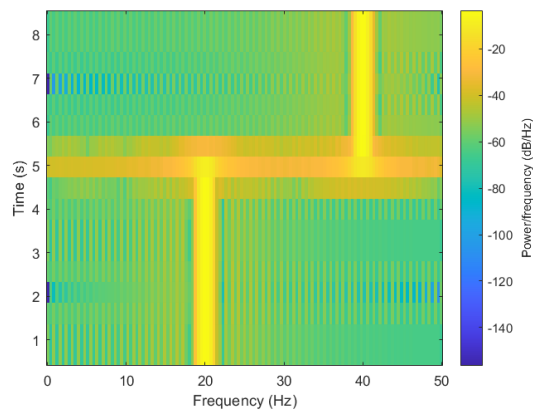


Figure 3.5(S₂ spectrogram)

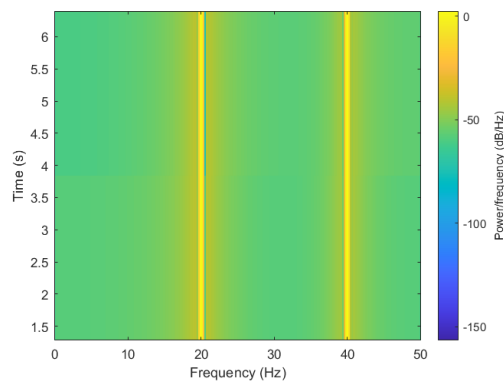


Figure 3.6(S₁ spectrogram)

علت آنکه این روش خیلی مناسب نیست این است که **window** ثابت باشد و رزولوشن زمانی نیز ثابت باشد و این میتواند برخی از جزئیات زمان را از دست بدهد همچنین در برخی اوقات ارتباط ناخواسته **window** به علت همپوشانی یکسان فرکانس های جدیدی ایجاد کند.

Cross Correlation

برای نوشتن کد برای **cross correlation** ابتدا طول دو سیگنال ورودی را گرفتیم و خروجی را نیز به مقدار یکی کمتر از مجموع آنها میسازم.

سپس برای آنکه یک سیگنال در طول یکی دیگر از سیگنال ها شیفت بخورد طولش را به اندازه طول خروجی منهای طول سیگنال ثابت منهای یک صفر میدهیم و آن را **flip** میکنیم زیرا در **cross correliation** این اتفاق طبق رابطه ریاضی آن میوفتد.

حال برای به دست آوردن نتیجه از تابع **convolution** که باز کد آن نوشته شده استفاده میکنیم . یعنی بعد از آن مانند کانالو گرفتن شروع به حرکت دادن سیگنال **flip** شده و ضرب نقطه ای آرایه های آن در سیگنال ثابت انجام می شود.

منابع:

Mathwork.com

iranbmemag.com

سوال از تی ها