به نام خدا

سیگنال ها و سیستم ها

استاد:دکتر کربلائی

گزارش پروژه

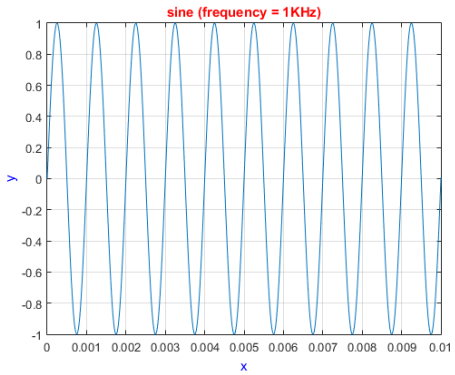
تمرین شماره ۱ متلب

سیّد محمّدامین منصوری طهرانی

۹۴۱۰۵۱۷۴

قسمت ۲- نمایش و بررسی نویز

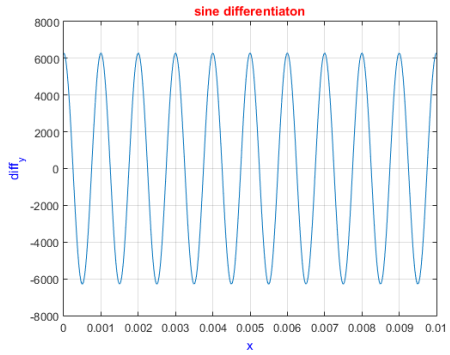
۱. با استفاده از دستور $x = 0:1e-6:10$ بین صفر تا ۱۰ مقادیر محور x را می‌سازیم. سپس از روی این بردار خروجی y سینوسی را می‌سازیم. برای نمایش آن از دستور $plot(x,y)$ استفاده می‌کنیم. مقدار فاصله پس از چندین بار رسم و مشاهده نمودار انتخاب شد. اگر فاصله کم باشد(در محدوده دوره تناوب سیگنال) شکل نمایش داده شده هیچ شباهتی به سینوسی نخواهد داشت. اگر تعداد بیش از حد زیاد باشد نیز تعداد محاسبات برنامه بالا رفته و نمایش نتیجه زمان بر خواهد بود. بنابراین عدد داده شده مناسب به نظر می‌رسد. ضمنا اگر کل ۱۰ ثانیه نمایش داده شود تمام صفحه به علت بالا بودن فرکانس پر می‌شود. پس با دستور $xlim([x1,x2])$ محدوده نمایش را به مقدار گفته شده که معادل ۱۰ سیکل سیگنال است($10ms$) یعنی صفر تا ۰/۰۱ تغییر می‌دهیم. نتیجه در شکل شماره ۱ مشاهده می‌شود. بقیه دستورات مربوط به عنوان ها و رنگ می‌باشد.



شکل ۱

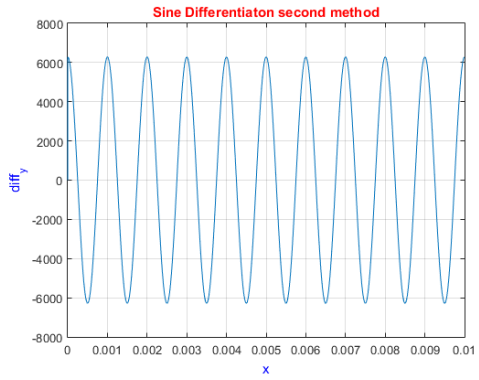
۲. تقریب مشتق: از رابطه تقریبی زیر به ازای همان فاصله های بخش قبل استفاده می‌کنیم.محدوده را نیز با همان دستور $xlim([x1,x2])$ برای نمایش تنظیم می‌کنیم. نتیجه در شکل ۲ قابل مشاهده است. که با انتظار سازگار است.(دامنه خروجی حدود ۶۲۸۰ است و کسینوسی است.)

$$\frac{dy}{dx} \approx \frac{f(x+\Delta x)-f(x)}{\Delta x} \quad , \frac{d}{dx}(\sin 2\pi\times 1000\times x) = 2000\times\pi\cos 2\pi\times 1000\times x$$



شکل ۲

البته پس از جستجو دستور مشتق گیری یافت شد. نمودار مشتق با روش دوم هم در کد نشان داده می‌شود. تصویر آن هم در شکل ۳ مشاهده می‌شود.



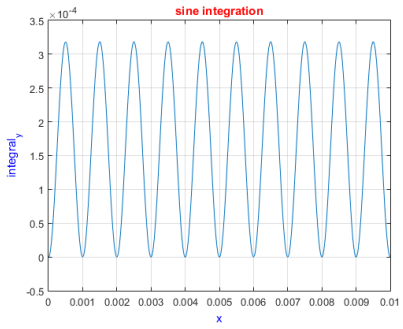
شکل ۳

تقریب انتگرال: از رابطه تقریبی زیر به ازای همان فاصله های بخش قبل استفاده می‌کنیم. برای تقریب می‌توان از تقریب مساحت با نقطه های میانی، چپ یا راست و یا تقریب ذوزنقه استفاده کرد و همه نتیجه مطلوب را بدست می‌دهند. اما ما برای دقت زیاد از روش بهتر یعنی تقریب سیمسن استفاده می‌کنیم. محدوده را نیز با همان دستور $xlim([x1, x2])$ برای نمایش تنظیم می‌کنیم. نتیجه در شکل ۴ قابل مشاهده است. که با انتظار سازگار است.(ماکزیمم خروجی حدود 3.18×10^{-4} است و سینوسی توان دوم است.)

$$simpson\ approximation$$

$$\int_a^b f(x)dx = \sum \frac{\Delta x}{6}\left(f(x_{i+1}) + 4f\left(\frac{x_{i+1}+x_i}{2}\right) + f(x_i)\right)$$

$$\int_0^x \sin(2000\pi x')dx' = \frac{1-\cos 2000\pi x}{2000\pi} = \frac{(\sin 1000\pi x)^2}{1000\pi}$$



شکل ۴

۳. هیستوگرام نموداری است که در آن توزیع یک متغیر نشان داده می‌شود. برای مثال در این مثال هیستوگرام سینوس نموداری است که مقادیر خروجی که بین ۱ و ۱- هستند را به تعدادی بازه مناسب با طول برابر تقسیم نموده و نشان می‌دهد چه تعداد از داده ها در این بازه هستند. می‌توان این تابع را به تعداد کل داده ها تقسیم کرد تا نرمالیزه شود و به صورت کسری از داده ها که در بازه ای مشخص هستند به کل داده ها در بیاید. به طور ریاضی یعنی به ازای مقادیر $y, y+dy$ چه بازه از x ها در این بازه هستند. فرم نمودار هیستوگرام را با توجه همین مفهوم می‌توان بدست آورد. یعنی چه کسری از کل x ها به $y, y+dy$ مربوطند:

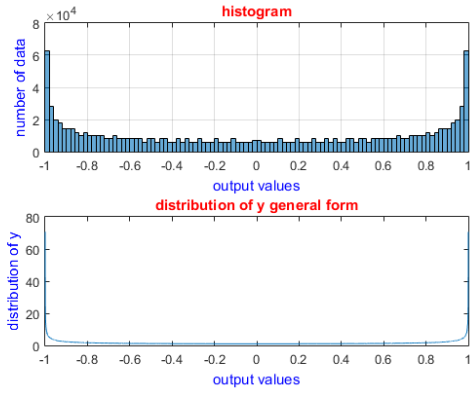
$$\frac{\sin^{-1}(y+dy) - \sin^{-1}(y)}{T} \times 2 = p(y)dy$$

$$p(y) = \frac{2}{T}\frac{d}{dy}(\sin^{-1} y) = \frac{2}{T}\frac{1}{\sqrt{1-y^2}}$$

توضیح: احتمال اینکه خروجی در بازه $y, y+dy$ باشد به این ربط دارد که چه کسری از x در دوره تناوب را می‌پوشاند. صورت کسر این مقدار است و باید در ۲ ضرب شود چون بازه $y, y+dy$ برای شکل سینوسی در هر دوره تناوب به ۲ برابر $\sin^{-1}(y+dy) - \sin^{-1}(y)$ مربوط است. نهایتا بر دوره تناوب نیز تقسیم می‌شود تا احتمال را بدهد. سمت راست معادله هم که یعنی احتمال اینکه بین $y, y+dy$ باشد برابر مساحت زیر نمودار احتمال است.($p(y)dy$) برای رسیدن به هیستوگرام باید در تعداد داده های y ضرب شود.

نمودار این شکل در متلب نمایش داده خواهد شد تا با هیستوگرامی که توسط خود متلب با دستور $histogram(y)$ نمایش داده می‌شود مقایسه شود.

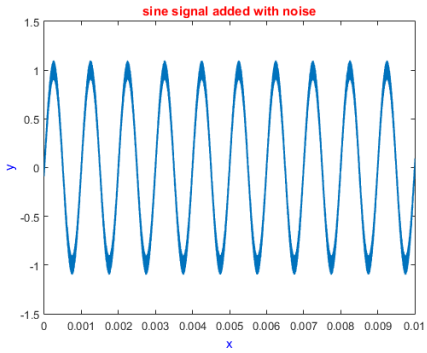
توجیه شهودی: هر چه به مقادیر اکسترمم نزدیک شویم، شیب تابع کم شده و به ازای بازه های برابر، در این مقادیر تعداد بیشتری x قرار می‌گیرد و هیستوگرام در این مقادیر بیشتر است. در صفر به دلیل بیشتر بودن شیب تعداد کمتری داده در هر بازه قرار می‌گیرد و مینیمم هیستوگرام را در صفر مشاهده خواهیم کرد. نتایج در شکل ۵ ملاحظه می‌شوند.



شکل ۵

۴. ابتدا به خاطر دقت بیشتر، بین صفر و ۱۰ می‌سازیم. سپس خروجی سینوسی را می‌سازیم. بعد از آن با دستور زیر نویز مورد نظر را تولید می‌کنیم و با سینوسی جمع می‌زنیم. دقت می‌کنیم که دستور $rand$ داده‌های تصادفی را با توزیع یکنواخت بین صفر و ۱ تولید می‌کند. پس برای اینکه در بازه مورد نظر سوال نویز تولید کنیم باید در طول بازه ضرب شود و سپس مرکز آن جابجا شود. نتیجه در شکل ۶ مشاهده می‌شود. دقت می‌کنیم که چون خروجی تا ۱۰ تولید شده تعداد داده های تصادفی باید تنظیم شود که هنگام جمع به نابرابری تعداد المان‌های آرایه برنخوریم.

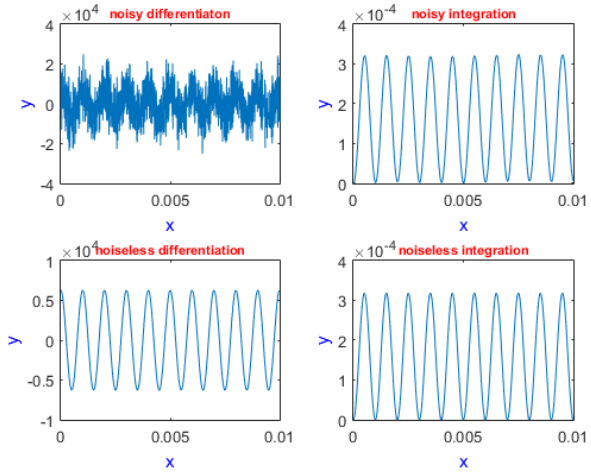
```
xn = (rand( 1 , 1e8 + 1 ))*0.2 - 0.1
```



شکل ۶

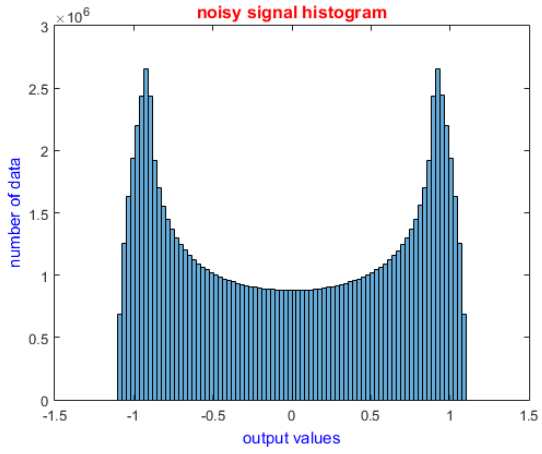
۵. تقریب‌ها برای عملیات ها همان روابط قبل اند. تاثیر نویز در عملیات: تاثیر آن در مشتق بیشتر است زیرا مشتق به معنی تغییرات تابع است و وقتی از سیگنال حاوی نویز مشتق می‌گیریم تغییرات ناگهانی جدیدی در نقاطی که نویز به آنها اضافه شده است، رخ می‌دهد که این به معنی تغییر فاحش مشتق تابع نسبت به وقتی است که نویز به آن اضافه نشده بود.

در انتگرال گیری اثر کم است. در واقع اگر تعداد داده ها زیاد باشد و نویز به صورت تصادفی یکنواخت باشد(مانند همین مثال) نویزهای تصادفی همدیگر را خنثی می‌کنند و در نتیجه تاثیری نمی‌گذارند. چون این تغییرات در تابع در انتگرال گیری مساحت هایی را اضافه و کم می‌کنند که در صورت تصادفی و یکنواخت بودن توزیع نویز این مساحت‌ها تقریبا یکدیگر را خنثی می‌کنند. نتیجه را می‌توان در شکل ۷ مشاهده کرد.



شکل ۷

۶. هیستوگرام با همان دستور $histogram(a,b)$ نمایش داده می‌شود($a = x\_n$)($b = number\ of\ bins$).(در کد من سیگنال جمع شده با نویز به جای x_n ،xn است!)تعداد بازه‌ها اگر خیلی زیاد باشد یا خیلی کم باشد نتیجه مطلوب نخواهد بود. در کد متلب این تعداد ۸۰ انتخاب شده است. شکل ۸ نتیجه را نمایش می‌دهد. نویز باعث شده در مقادیر خارج برد سینوس هم فراوانی داشته باشیم.



شکل ۸

قسمت ۳- حذف نویز

۱. درست است که این سیستم علّی نیست اما یک روش ممکن این است: چون ما احتمالا نیازی نداریم که دقیقا در لحظه سیگنال را تصحیح کنیم، پس زمان کافی خواهیم داشت تا تعداد داده‌ی لازم در زمان های بعدی بگیریم و بعد با این روش سیگنال حاوی نویز را

تصحیح کنیم. در واقع اگر نیاز سریعی به تصحیح سیگنال نداشته باشیم، ابتدا مقدار مناسبی از سیگنال را ذخیره می‌کنیم و بعد به تصحیح آن می‌پردازیم. در این کد هم ابتدا سیگنال نویزی تولید می‌شود و سپس تصحیح می‌شود. بنابراین علّی نبودن تعریف روش، مشکل عملی ایجاد نمی‌کند.

۲. ** تابع در فایل جداگانه نوشته شده و در فایل زیپ با نام MA ذخیره شده است و در کد اصلی از آن استفاده می‌شود. لطفا برای کارکرد صحیح کد، مسیر سرچ را به فایل اصلی اضافه کنید.**

تابع نوشته شده از یک حلقه for تشکیل شده است. با توجه به عملیات مورد نظر، برای تعدادی از داده‌های اول و آخر امکان محاسبه تابع وجود نخواهد داشت.در واقع به دلیل اینکه نباید اندیس آرایه ای منفی شود، از تعدادی از داده های اولیه صرفنظر می‌شود و آنها تصحیح نمی‌شوند. همچنین به دلیل اینکه نباید اندیس المانی از طول آرایه بیشتر شود، تعدادی از داده های آخر هم تصحیح نمی‌شوند ولی دسته دوم در محدوده صفر تا ۱۰ میلی ثانیه قرار ندارد و مشکلی ایجاد نمی‌کند. برای هر دو دسته همان مقدار اصلی تابع قرار داده می‌شود. برای داده های بین این دو حد، با استفاده از تابع میانگیری متلب($mean$) آرایه داده های حول نقطه مورد نظر به تابع داده می‌شود و محاسبه صورت می‌گیرد.

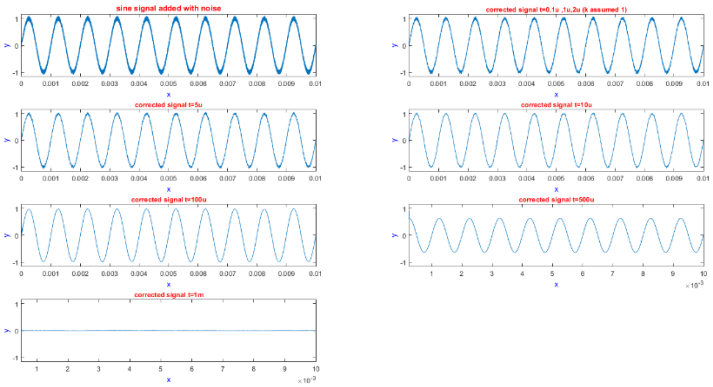
۳. به سادگی مقدار k بدست می‌آید. اگر فاصله زمانی بین نمونه‌ها δ باشد و طول پنجره Δ، رابطه زیر مقدار k را بدست می‌دهد.

$$k = \frac{\Delta}{2\delta}$$

چون بعضی از طول های پنجره داده شده بسیار کم هستند، باید فاصله نمونه برداری کم شود تا k طبیعی شود. اما این کار تعداد محاسبات کد را به شدت افزایش می‌دهد و کد را زمان بر می‌کند. لذا از این کار صرفه نظر می‌کنیم و به ازای طول‌های $0.1\mu s, 1\mu s, 2\mu s$، k را ۱ در نظر می‌گیریم.(برای سه طول پنجره اول، **یک** شکل رسم شده است.) بقیه را از رابطه بالا بدست می‌آوریم.(اگر جایی لازم بوده جز صحیح گرفته شده است.)

۴. نتایج در شکل ۹ مشاهده می‌شوند. برای نمایش بهتر سیگنالها محدوده y را کم کردیم. برای مشاهده دقیقتر، بهتر است پس از اجرای کد و باز شدن پنجره نمودار ها، آنرا $maximize$ کنید تا تغییرات بهتر دیده شوند.
**توجیه سیگنال به ازای $1000\mu s$

$t = 500\mu s$: این مقدار برابر خود دوره تناوب سیگنال اصلی است. یعنی هر داده با تمام داده های مربوط به نیم تناوب چپ و راست خود جمع می‌شود و میانگین گرفته می‌شود. که این به وضوح میانگین سیگنال در **یک** دوره تناوب است و سیگنال سینوسی در این بازه میانگینی برابر صفر دارد. نویز ها هم در اثر این عمل تقریبا خنثی شده و نتیجه صفر خواهد بود. البته محدوده نمایش محور x نیز در این مورد عوض شده و علت آن همان است که در قسمت ۲ گفته شد؛ تعدادی از داده های اولیه را به دلیل اهمیت کم تصحیح نمی‌کنیم. وقتی k زیاد شود این تعداد قابل ملاحظه می‌شود. چون عملیات تصحیح از مقدار k به بعد انجام می‌شود، آن داده ها را در نمایش خود نمی‌آوریم.
*استثنائا برای این قسمت چون زمان اجرای برنامه به شدت زیاد شد(پس از حدود ۱۰ دقیقه با دستور $break$ برنامه را متوقف کردم. به علت رسم چندین نمودار و استفاده های متعدد از تابع تعریف شده این قسمت بسیار زمان بر می‌شود.)، محدوده تعریف سیگنال را به ۱ ثانیه کاهش دادم.



شکل ۹

۵. با توجه به تصویر بالا نتیجه می‌گیریم که جواب به سیگنال وابسته است. از طرفی اگر k مقدار کمی باشد نویزهای تصادفی کمتری با هم جمع شده و میانگین گیری می‌شوند که این تضعیف نویز را کاهش می‌دهد. پس بهتر است k زیاد شود. از طرفی اگر برای حذف بیشتر نویزها k بزرگی انتخاب کنیم که طول پنجره از مرتبه دوره تناوب سیگنال باشد، این عملیات باعث جایگزینی مقدار میانگین سیگنال با سیگنال اصلی می‌شود که اصلا مطلوب نیست. پس طول پنجره باید مقداری بزرگ اما در مقایسه با دوره تناوب سیگنال کوچک باشد که مقدار مناسب آن با توجه به سیگنال خاص تحت بررسی بدست می‌آید.

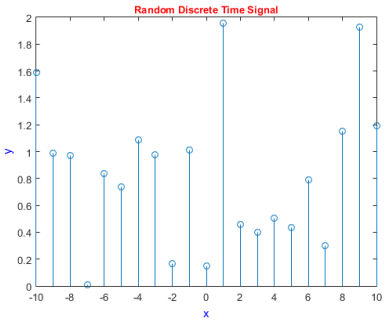
قسمت ۴- کانولوشن و سیستم

۱. این دستور دو بردار می‌گیرد و حاصل کانولوشن آنها را می‌دهد. برای مثال اگر فرض کنیم درایه‌های بردارها ضرایب چندجمله ای باشند، حاصل $conv\ (\ a\ ,b\ )$ برابر برداری خواهد بود که حاوی درایه های حاصل ضرب چندجمله ای های فوق است. بنابراین اگر طول بردار اول $length\_a$ و طول بردار دوم $length\_b$ باشد، طول بردار خروجی برابر $length_a\ +\ length_b\ -1$ خواهد بود چون اگر چندجمله ای اول از درجه m باشد طول آن $m+1$ و چندجمله ای دوم از درجه n باشد طول آن $n+1$ خواهد بود. درجه خروجی $n+m$ است و لذا طول آن $length_a\ +\ length_b\ -1$ خواهد بود.

هم چنین این تعریف با تعریف کانولوشن کاملا سازگار است زیرا این ضرب چندجمله ای تک تک جمله ها را در هم ضرب می‌کند و به سادگی می‌توان دید که این با تعریف کانولوشن سیستم های گسسته سازگار است.

۲. این دستور اصلا به اینکه مبدا سیگنالهای ما کجا هستند کاری ندارد و هر دو را بردارهایی با مبدا ۱ می‌بیند و برداری به عنوان خروجی از خروجی کانولوشن می‌دهد و مبدا آن ۱ است. ما باید با توجه به سیگنالها و طول آنها محدوده دامنه کانولوشن را بیابیم و مقادیر n را با توجه به آن بدست آوریم و با نگاشت مناسب به مقادیر منفی آنرا رسم کنیم.

۳. با دستور $rand$ مقادیر رندم را می‌سازیم. سپس یک نگاشت به مقادیر منفی می‌نویسیم. با دستور $stem$ با دو ورودی که اولی مقادیر روی محور x (مقادیر نگاشت که شامل مقادیر منفی n هستند) و دومی همان خروجی تابع رندم است، نمودار سیگنال گسسته را رسم می‌کنیم. نتیجه در شکل ۱۰ قابل مشاهده است.



شکل ۱۰

۴. پاسخ ضربه هر کدام از سیستم ها داده شده است. دقت می‌کنیم که بنا به توضیح صورت سوال مبدا پاسخ ها ۱ است. سیگنال تصادفی ما از ۱۰- تا ۱۰+ است.

نتایج در شکل‌های ۱۱ تا ۱۴ ملاحظه می‌شوند.

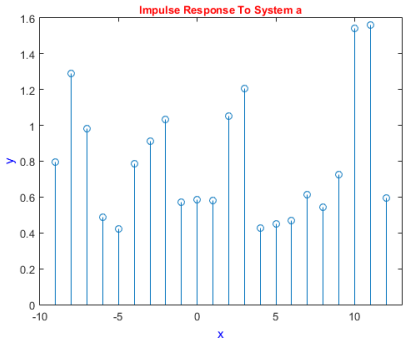
سیگنال a : این سیگنال ۲ درایه دارد و حاصل کانولوشن آن با سیگنال تصادفی با توجه به رابطه قسمت ۱ مجموعا ۲۲ درایه خواهد داشت. بدست آوردن محدوده کانولوشن و سپس نگاشت به مقادیر مثبت و منفی مناسب:

این پاسخ ضربه ۲ مقدار غیر صفر دارد، پس از انعکاس نسبت به محور y ها با توجه به رابطه $h[m-n]$ که $-\infty < m < \infty$ جابجا می‌شود، باید m هایی یافت شود که به ازای آنها خروجی صفر نیست. مقادیر منفی آن تا ۹- می‌رود چون پاسخ ضربه پس از قرینه شدن به ازای m های بزرگتر از این مقدار هیچ اشتراکی با ورودی نخواهد داشت و حاصل ضرب آنها صفر خواهد بود. از طرف مثبت هم m تا مقدار ۱۲ می‌رود.(با توجه به توجیهی مشابه قبل). پس این نگاشت بدست آمده به ورودی اول $stem$ داده می‌شود تا همراه با حاصل کانولوشن رسم شود.

سیگنالهای b , c , d : با همان روش گفته شده در قبل محدوده ها بدست می‌آیند و سپس با نگاشت مربوط پاسخ سیستم به این ورودی ها رسم می‌شود.

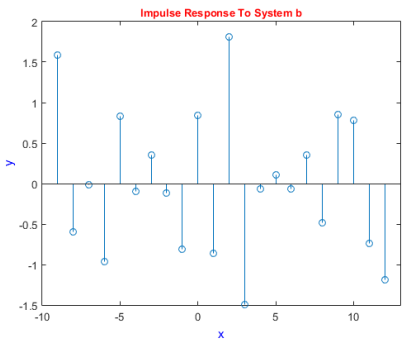
*برای تقارن و زیبایی از دستور $xlim$ در جواب نهایی استفاده شده است و نیازی به آن نبوده است.

پاسخ به سیستم a :



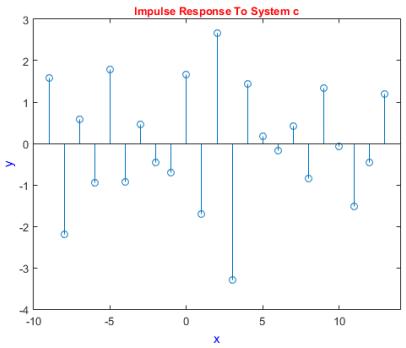
شکل ۱۱

پاسخ به سیستم b :



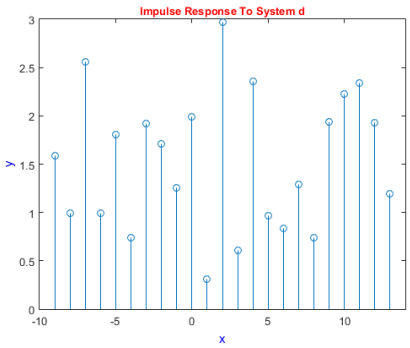
شکل ۱۲

پاسخ به سیستم c :



شکل ۱۳

پاسخ به سیستم d :

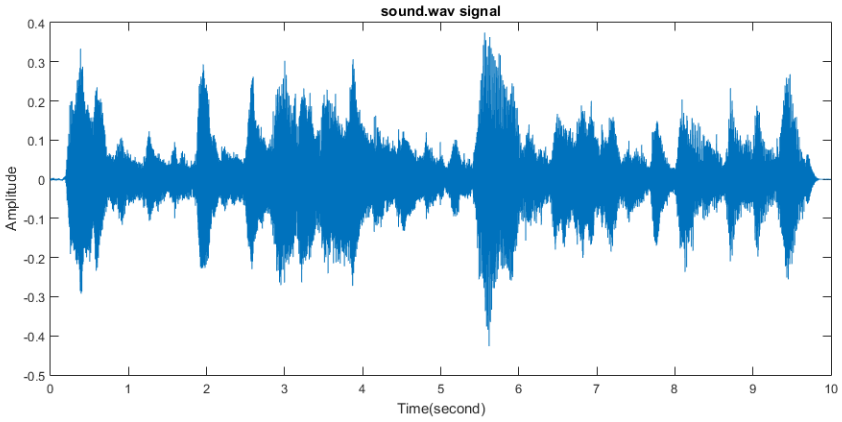


شکل ۱۴

قسمت ۵- کار با فایل صوتی

۱. دستور $audioread$: ورودی آن نام یک فایل(و آدرس دقیق آن اگر در $path$ مربوط به $m\ file$ نباشد) و خروجی های آن داده های نمونه برداری شده فایل و همچنین فرکانس نمونه برداری می‌باشند. از فایل پیوست تمرین استفاده می‌کنیم. فرکانس نمونه برداری برابر ۴۴۱۰۰ هرتز است.

۲. با دستور $length(y)$ که y همان مقادیر سیگنال هستند، طول سیگنال یعنی تعداد داده ها بدست می‌آید. اگر طول بازه ها که همان F_s^{-1} است را در طول سیگنال ضرب کنیم، زمان سیگنال را بدست می‌آوریم. به این ترتیب می‌توانیم مقادیر سیگنال را برحسب زمان رسم کنیم. نتیجه در متلب نمایش داده خواهد شد و در زیر هم(شکل ۱۵) مشاهده می‌شود.



شکل ۱۵

۳. این دستور سیگنال (y) و فرکانس نمونه برداری آنرا می‌گیرد و یک شئ می‌سازد. این شئ کنترل صدا را به ما برمی‌گرداند. از دستورات آن می‌توان به موارد زیر اشاره کرد:

$play$: برای شروع صدا استفاده می‌شود.(در فایل متلب هم از همین دستور برای پخش استفاده شده است.)

$pause$: برای متوقف کردن صدا استفاده می‌شود.

$resume$: پس از توقف برای ادامه دادن پخش صدا استفاده می‌شود.

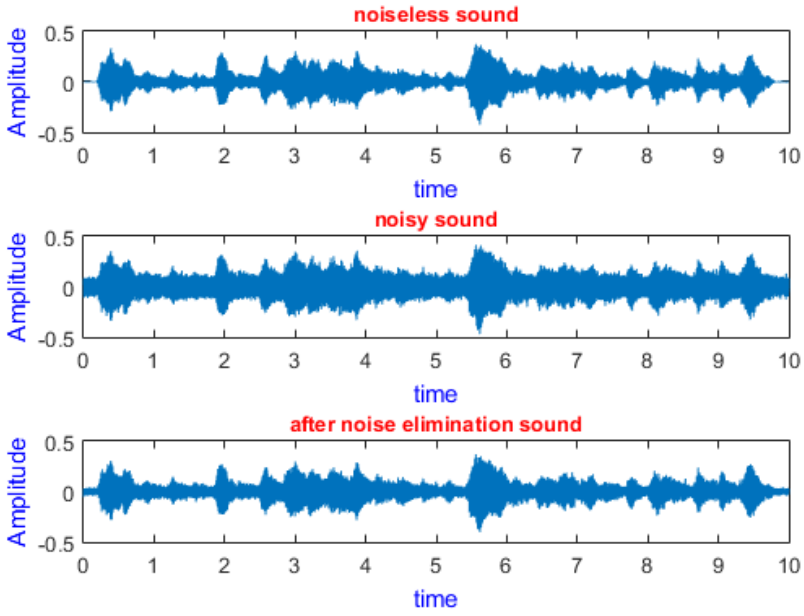
$stop$: برای قطع صدا استفاده می‌شود.

۴. برای تولید متغیر تصادفی با توزیع گاوسی با میانگین m و انحراف معیار σ می‌توان از دستور زیر استفاده نمود:

$$x = \sigma * randn(length(y), 1) + m$$

با این کار مقادیر تصادفی به تعداد داده های سیگنال تولید می‌شوند. اگر x بدست آمده با y جمع شود، سیگنال نویز دار جدید بدست می‌آید. شئ $audioplayer$ جدیدی برای این سیگنال می‌سازیم و با آن این سیگنال جدید را پخش می‌کنیم. صدای شنیده شده دارای صدای ثابتی مربوط به نویز است. همچنین برای انحراف معیار از دستور std استفاده می‌کنیم. این دستور همانطور که ما می‌خواهیم و در کلاس صحبت شد، مقدار $\sum\sqrt{(x_i - \bar{x})^2}$ را به $n-1$ تقسیم می‌کند. سپس صدای جدید پخش می‌شود.

۵. نتیجه نموداری در شکل ۱۶ قابل مشاهده است. البته به خاطر بخش بندی کد و اینکه به طور خاص این نمودار خواسته نشده بود، با اجرای کد این نمودار نشان داده نمی‌شود و من در کد خودم این نمودار را رسم کردم.



شکل ۱۶

توضیح: باید دقت کرد که این سیگنال سینوسی نیست و مانند مثال‌های قبل با k نسبتا بزرگ نمی‌توان به نتیجه مطلوبی دست یافت. اگر k زیاد باشد مانند شکل های آخر سوال ۴ قسمت ۳، کل سیگنال هم تضعیف شده و عملا چیزی شنیده نمی‌شود. پس از آزمون و خطا برای شنیدن صدای با نویز کم(در این سیگنال با روش قسمت ۳ نویز از بین نمی‌رود.) ضمن کم نشدن صدای سیگنال اصلی، مقدار k را ۳۰ انتخاب کردم. هر چه این عدد بیشتر باشد صدای سیگنال اصلی و نویز هر دو کاهش می‌یابند. برای شنیده شدن راحت‌تر صدا در زمان پخش، دامنه آنرا در ۴ ضرب کردم. (البته می‌توان این ضرب را انجام نداده و صدای بلندگو را افزایش داد!)

۶. دقت می‌کنیم باید با همان فرکانس نمونه برداری صوت اصلی سیگنال جدید را بسازیم تا در جمع کردن آن با صوت اصلی به مشکل نابرابری تعداد المان‌های آرایه ها برنخوریم. همچنین طول این سیگنال باید با طول سیگنال اصلی برابر باشد. ضمنا ترانهاده سینوسی با سیگنال اصلی جمع شده است چون یکی از آرایه ها سطری و دیگری ستونی است و برای عمل صحیح جمع باید یکی ترانهاده شود. صوت بدست آمده حاوی یک بوق ثابت خواهد بود. با اجرای کد پخش خواهد شد.

۷. با دستور گفته شده سیگنال مورد نظر را می‌سازیم. دستور $chirp$ ، ۴ ورودی می‌گیرد. ورودی اول برداری است که حاوی زمان هایی است که در هرکدام یک سینوسی با فرکانس مشخص که از ورودی های بعدی بدست می‌آید باید در این زمانها پخش شود. ورودی دوم فرکانس شروع است. یعنی در زمان اولین المان بردار ورودی اول، سینوسی با این فرکانس تولید می‌شود و به همین ترتیب به صورت خطی

ادامه می‌یابد. ورودی سوم زمانی است که سینوسی با فرکانس ورودی چهارم تولید می‌شود. در زمان های بین، سینوسی با فرکانس بین ۲ مقدار اول و آخر تولید می‌شود.(به صورت خطی بین این مقادیر زیاد می‌شود.) نتیجه با اجرای کد پخش می‌شود. این صوت مانند آژیری است که زیرتر و زیرتر شده و به فرکانس های زیاد می‌رسد.

قسمت ۶- بازسازی صوت

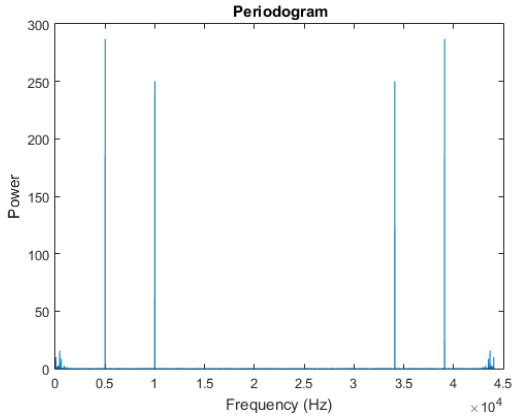
۱. تغییرات یافت شده به شرح زیر بود:

- با توجه به قسمت قبل بوق شنیده می‌شود که مربوط است به جمع شدن با سینوسی(یا سینوسی ها)
- نویز هم اضافه شده است.
- سیگنال اکو شده است. مطابق توضیحات کلاس، شیفت یافته صوت ضربدر ضریبی با خود صوت جمع می‌شود.
- چند ثانیه هم فقط نویز پخش می‌شود که می‌توان آنها را از فایل حذف نمود.

۲. ترتیب حذف نویز به این شرح خواهد بود؛ ابتدا بوق را حذف می‌کنیم. سپس اکو را حذف می‌کنیم و پس از آن نویز تصادفی. چند ثانیه ای که در اثر اکو اضافه شده را نیز به صورت دستی صفر می‌کنیم. در زیر به توضیح قدم به قدم عملیات ها می‌پردازیم.

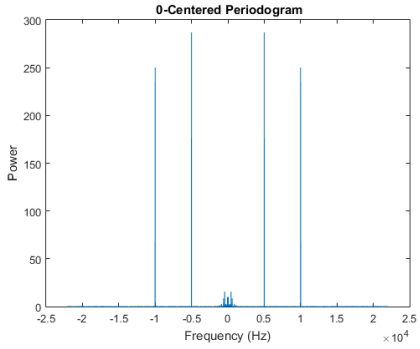
گام اول: رسم تبدیل فوریه سیگنال نویز دار

سیگنال را می‌خوانیم. بازه زمانی و طول سیگنال را بدست می‌آوریم. سپس تبدیل فوریه سیگنال را می‌گیریم. چون این مقدار مختلط است، در مزدوج مختلط آن ضرب می‌کنیم و نمایش می‌دهیم. نتیجه در شکل ۱۷ مشاهده می‌شود.

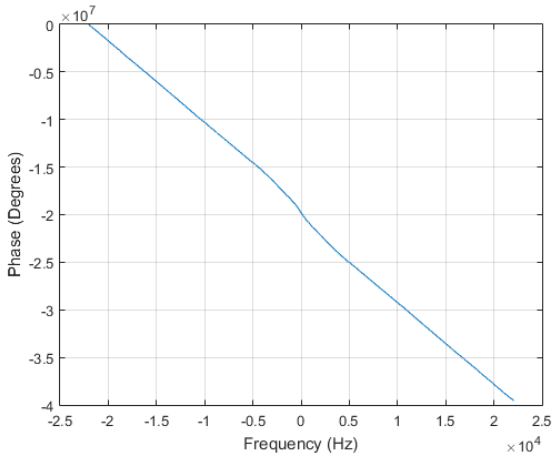


شکل ۱۷

سپس با دستور $fftshift$ تبدیل فوریه را متقارن می‌کنیم و همچنین نمودار فاز تبدیل فوریه را رسم می‌کنیم. در همه از توابع ساخته شده متلب استفاده شده است. نتایج در تصاویر ۱۸ و ۱۹ مشاهده می‌شود.



شکل ۱۸



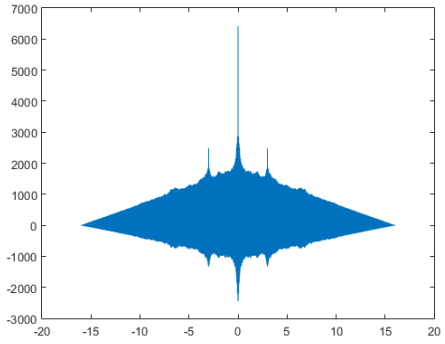
شکل ۱۹

گام دوم: حذف نویز سینوسی

پهنای باندی را تعریف می‌کنیم. از شکل ۱۸ مشخص است که داده های اطلاعات ما در فرکانس های پایین هستند. ۲ سینوسی هم به سیگنال اضافه شده اند که یکی در فرکانس ۵ و دیگری در ۱۰ کیلوهرتز هستند. بنابراین می‌توانیم از فرکانس مشخصی به بعد را حذف کنیم. پهنای باند را ۲۰۰۰۰۰ گرفتیم. برای بدست آوردن سیگنال اصلی باید از این تبدیل فوریه، فوریه وارون بگیریم. پس از جستجو معلوم شد باید آرگومان $symmetric$ را اضافه کرد تا کارکرد صحیح از دست نرود. در آخر هم صدای تصحیح شده پخش می‌شود.

گام سوم: حذف اکو

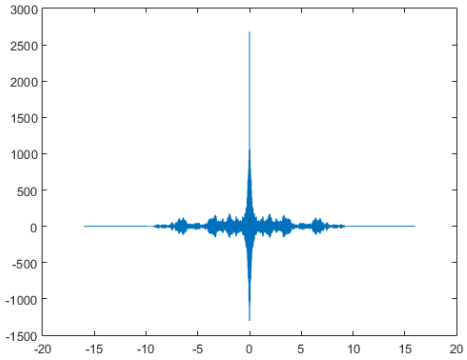
ابتدا با دستور $xcor$، $correlation$ سیگنال نویزدار را با خودش حساب می‌کنیم. توجه می‌کنیم که این دستور تقارن دارد و باید نگاشت مناسب بدهیم تا مقادیر مثبت و منفی به طور متقارن حول صفر مشخص شوند. نمودار آن در شکل زیر(شکل ۲۰) آمده است. پیک اول یعنی سیگنال بدون شیفت با خودش شباهت دارد و پیک دوم که در ۳ ثانیه است، یعنی در این زمان شیفت داشته ایم. نسبت قله ها بهره اکو را بدست می‌دهد.



شکل ۲۰

سپس به صورت سیستم فیدبک دار عمل می‌کنیم. به این ترتیب که تا قبل ثانیه ۳ را در متغیر جدیدی میریزیم. از این به بعد سیگنال نویزدار را منهای سیگنال جدید می‌کنیم. در واقع در هر زمان سیگنال نویزدار از سیگنال ۳ ثانیه قبل آن کم می‌شود و به این ترتیب اکو حذف می‌شود.

پس از آن برای اطمینان از باقی نماندن اکو، دوباره کوریلیشن سیگنال جدید را با خودش رسم می‌کنیم. ملاحظه می‌شود که فقط در مبدا قله داریم و بنابراین شیفت دیگری نمانده است.(شکل ۲۱)



شکل ۲۱

گام چهارم: تلاش برای حذف نویز تصادفی

در این مرحله با استفاده از تابع قسمت سوم و روش پنجره سعی می‌کنیم نویز سیگنال را حذف کنیم. مقدار k را ۲۰ انتخاب می‌کنیم. البته دقت می‌کنیم که با این روش نویز کاملا رفع نمی‌شود. البته برای شنیده شدن بهتر صدا در زمان پخش آن دامنه را در ۲ ضرب کردم. البته می‌توان آنرا حذف نموده و صدای کامپیوتر را زیاد کرد. در آخر نیز با استفاده از متغیر کنترلی تعریف شده می‌توانید در $command\ window$ صدا را پخش کنید.