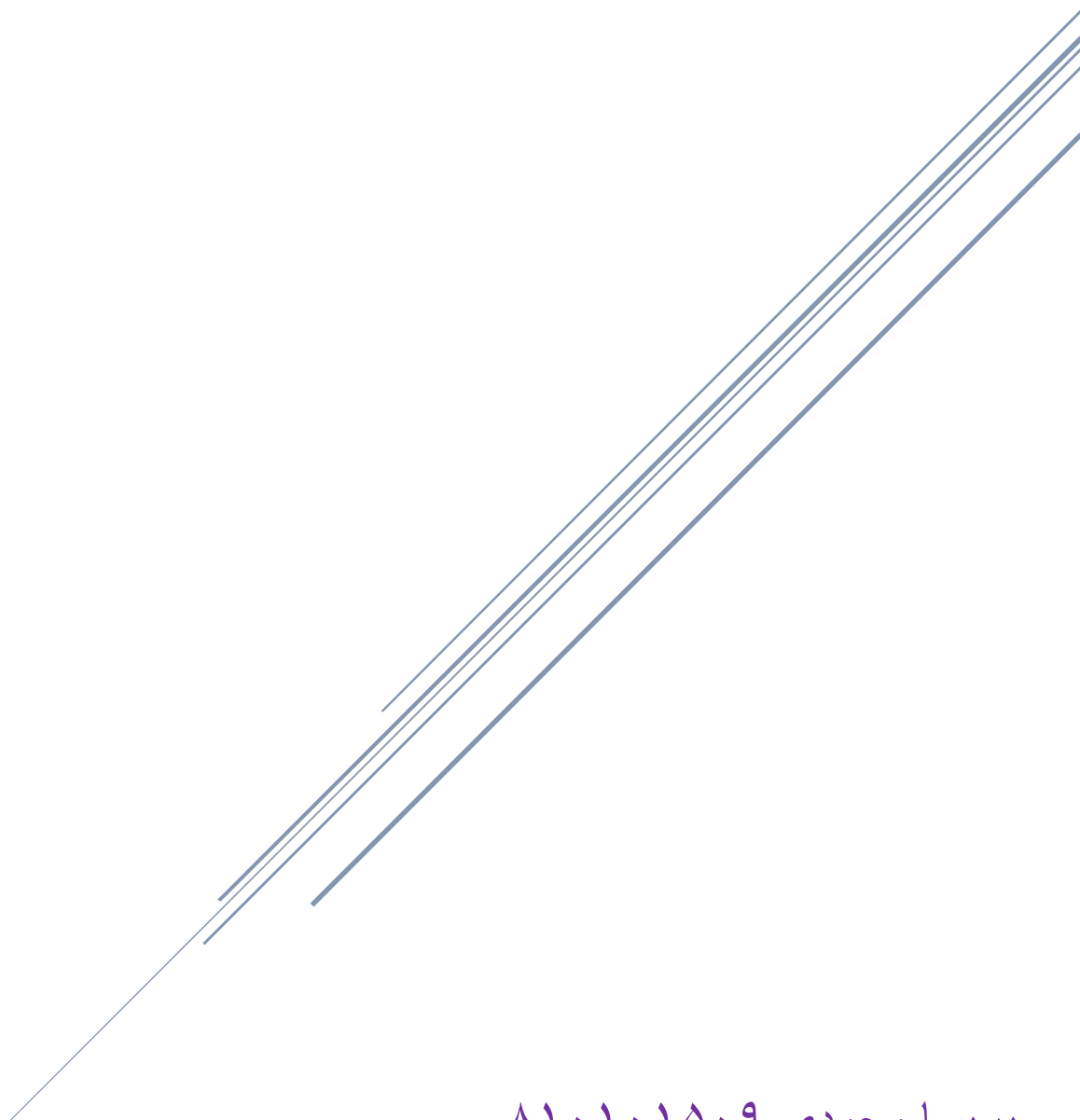


سیگنال ها و سیستم ها

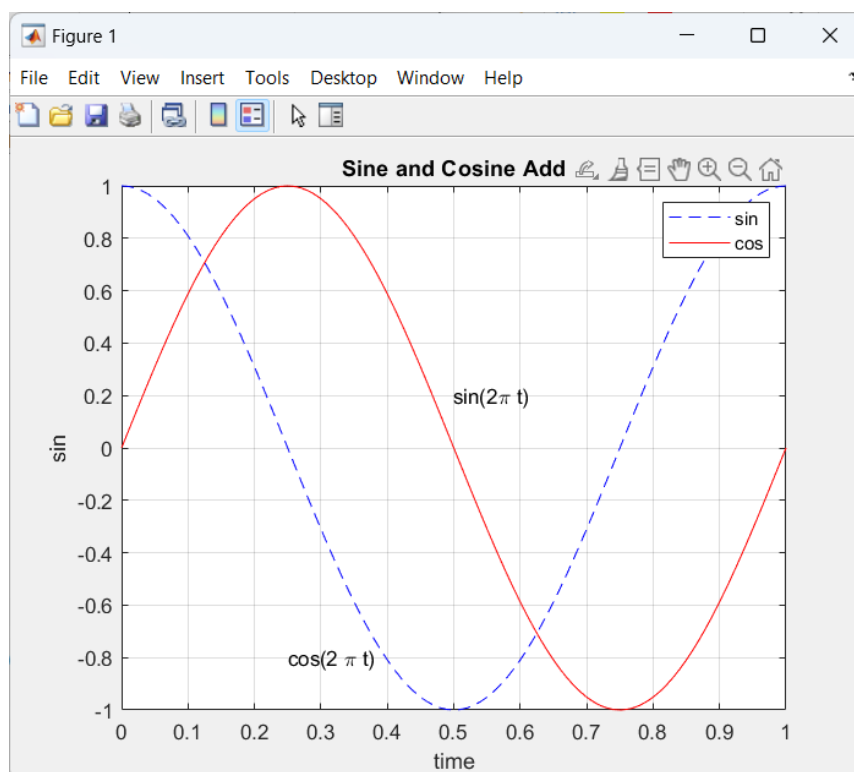
تمرین کامپوتری ۱



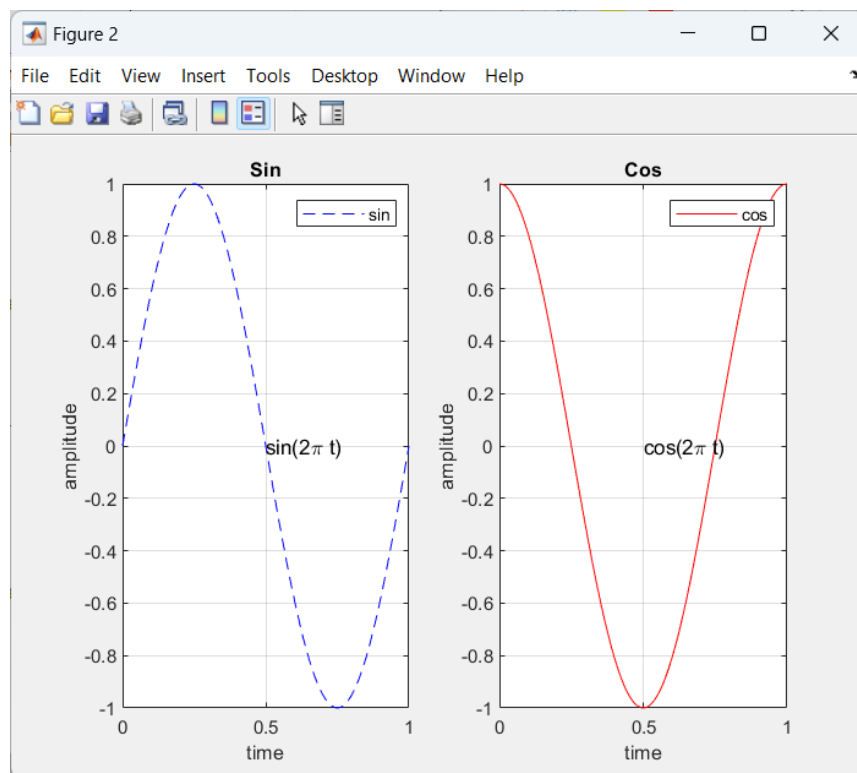
پریسا محمدی ۸۱۰۱۰۱۵۰۹

سوال ۱

بخش ۱) با اجرای کد به صورت خط به خط با دستور `run + step` می بینیم که هر خط کد چه عملکردی دارد. در ابتدا بردار زمان را تعریف کرده ایم که از ۰ تا ۱ واحد زمانی را با گام های ۰.۰۱ می پیماید و در نتیجه طول بردار حاصل برابر با ۱۰۱ خواهد. سپس سیگنال های سینوس و کسینوس را به صورت برداری از زمان تعریف می کنیم. برای ایجاد صفحه ای برای رسم `plot` از دستور `figure` استفاده کرده و سپس در دستور `plot` بردار های محور افقی و عمودی را تعیین کرده و همچنین می توانیم جزییاتی را مربوط به شکل ظاهری `plot` به این دستور اضافه کنیم مانند `'--b'` که باعث می شود تابع با خط به فرم خط چین و به رنگ ابی رسم شود. پس از رسم نمودار اول می توانیم از دستور `hold on` استفاده کنیم تا نمودار بعدی نیز در همین پنجره و روی نمودار قبلی رسم شود. حال می خواهیم متنی را در کنار هر نمودار قرار دهیم تا که مشخص شود که هر نمودار معرف چیست. برای این کار از دستور `text` استفاده کرده که دارای دو بخش است یکی متن و عباراتی که می خواهیم اضافه کنیم و دیگری محل آنها است که برای متن آن را در یک `string` دو تایی ذخیره کرده و برای مکان هم مختصات افقی عبارات را به ترتیب در بردار `x0` و مختصات عمودی آنها را نیز به ترتیب در بردار `y0` ذخیره کرده و آنها را به دستور `text` می دهیم. در خطوط بعدی کد به ترتیب به `figure` امان تیترا می دهیم. بعد در یه باکس کوچک به نام `legend` راهنمایی برای هر تابع می گذاریم و سپس لیبلی برای محور های عمودی و افقی تعیین می کنیم. در انتها نیز `grid` را فعال کرده تا دقیق تر بتوانیم مقادیر تابع را در نقاط مختلف بررسی کنیم.



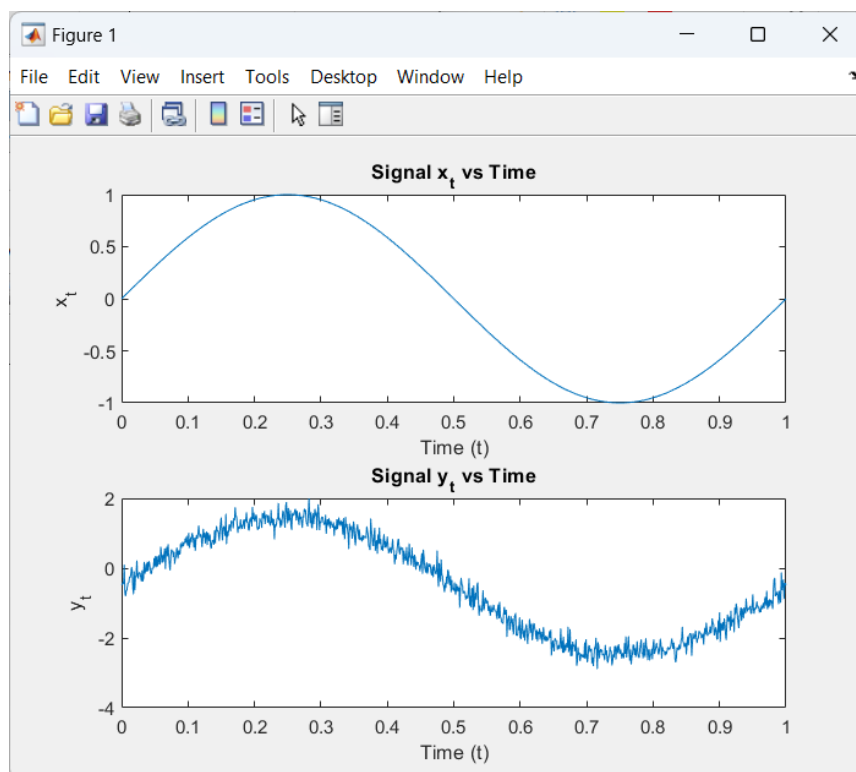
بخش ۲) در بخش دوم این سوال برای رسم دو نمودار به جای دستور `hold on` از دستور `subplot` استفاده می کنیم. در `hold on` هر دو تابع بر روی یک صفحه مختصات رسم می شوند ولی با دستور `subplot` می توانیم چند تابع را در صفحه های مختصات جداگانه ولی در کنار هم رسم کنیم. وقتی که می خواهیم موقعیت یک سیگنال را در این مجموعه صفحات مختصات مشخص کنیم به `subplot` به این صورت ورودی می دهیم که در ابتدا ماتریسی را شامل همه سیگنال هاست می شود را در نظر میگیریم حال برای تعیین موقعیت این تابع خاص شماره صفحه مختصات ای که به آن می خواهیم اختصاص دهیم را، ورودی می دهیم. باید توجه داشته باشیم که نحوه شماره گذاری این صفحات در ماتریس `subplot` به این صورت است که اول هر سطر شماره گذاری شده و بعد ادامه شماره گذاری از ابتدای سطر بعدی و در ادامه به همین ترتیب است.



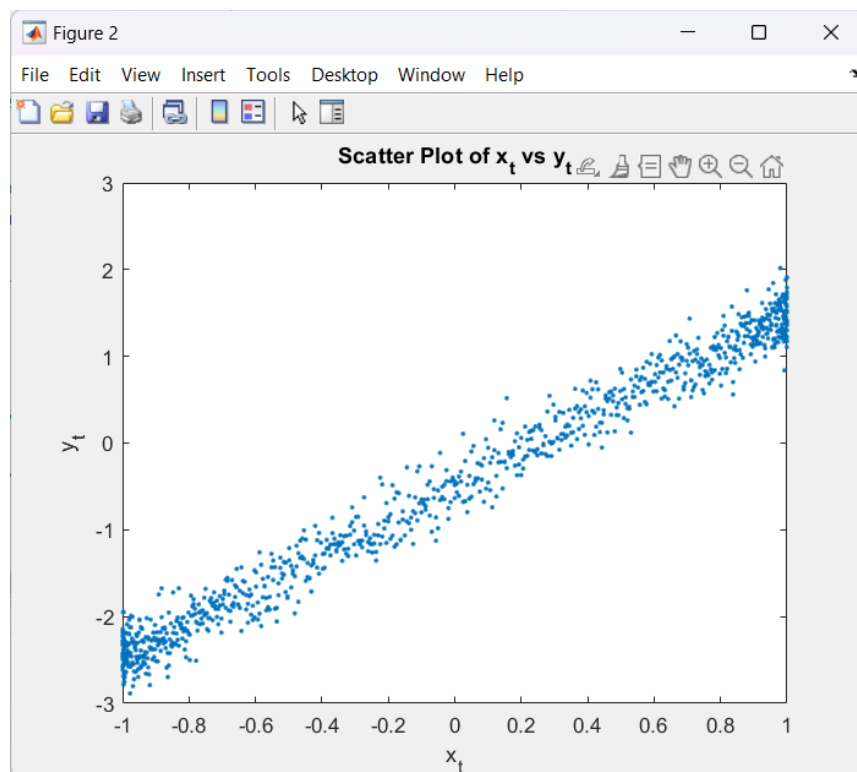
سوال ۲

بخش (ا۲) در این سوال در حال بررسی سیستمی خطی هستیم که ورودی آن سیگنال $x(t)$ بوده و سیگنال $y(t)$ را نیز خروجی می دهد و برای اینکه سیستمی که تعریف کرده ایم بیش تر به سیستم هایی که در دنیای واقعی هستند شبیه باشد برای خروجی آن نویز هم در نظر گرفته شده است که جلوتر با رسم نمودار تاثیر آن را به وضوح می بینیم.

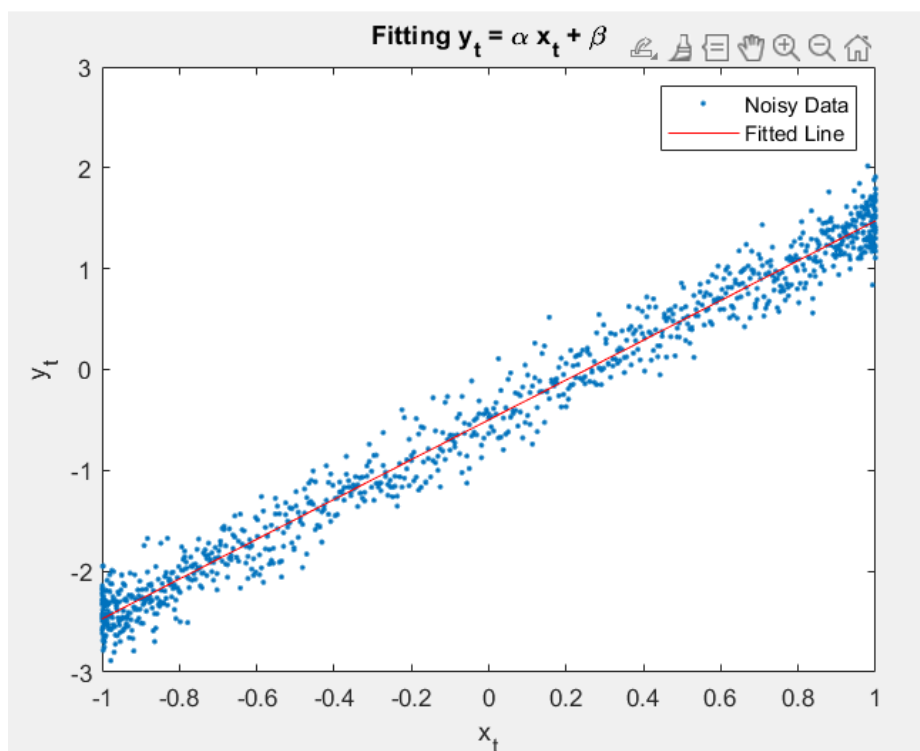
سیگنال های ورودی و خروجی و بردار زمان در فایل `p2.mat` به ما داده شدند که آنها را با دستور `load` خوانده و در متغیر `data` ذخیره کردیم. حال برای دسترسی به بردار ها به طور جداگانه نیاز است بدانیم که به چه اسمی در فایل `.mat` قرار داشتند، بدین منظور `filename(data)` را چاپ کرده تا این اسمی را بیابیم. حال می توانیم آنها را از `data` استخراج کرده و در متغیر های مربوطه (x_t, y_t, t) ذخیره کنیم. برای نمایش سیگنال x_t, y_t از `subplot` با ابعاد دو در یک استفاده کرده تا آنها را بر حسب t رسم کنیم و همچنین برای کامل بودن توضیحات نمودار، لیبل محور افقی و عمودی و تیتل نمودار را هم تعیین میکنیم. در خروجی می بینیم که نمودار x_t به فرم یک سیگنال سینوسی تمیز با دامنه یک و دوره تناوب یک واحد زمانی است و نمودار y_t به فرم یک سیگنال سینوسی امیخته با نویز با دامنه ۲ و دوره تناوب یک واحد زمانی است.



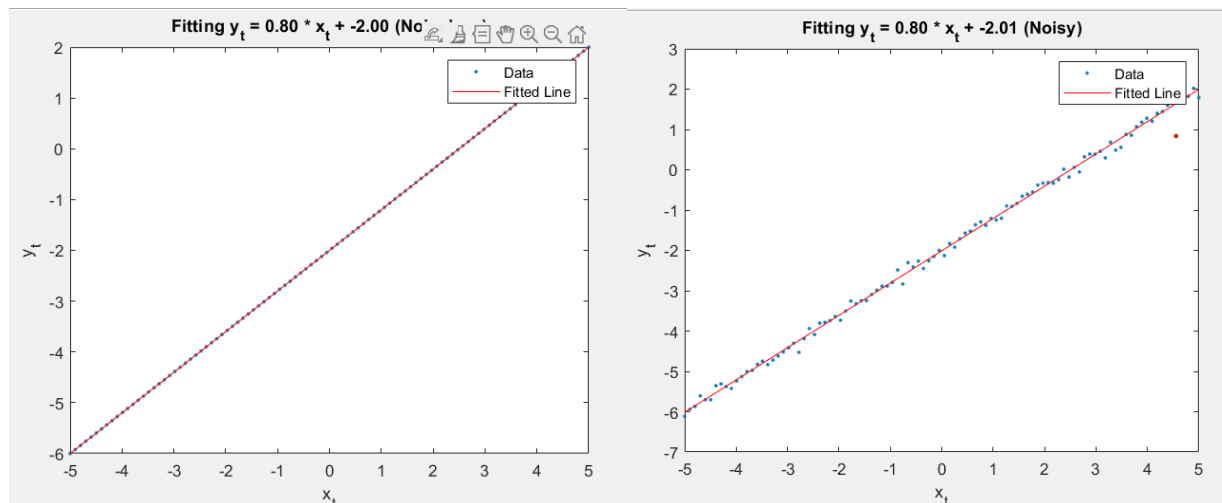
بخش ۳) نموداری هم برای رسم نمودار سیگنال خروجی بر حسب سیگنال ورودی رسم می کنیم. در رسم این نمودار از رسم نقطه ای استفاده می کنیم چرا که اینگونه نمایش واضح تری خواهیم داشت و نقاط به هم وصل نمی شوند و رابطه بین x_t , y_t بهتر مشخص می شود. خروجی این نمودار به صورت مجموعه ای از خطوط هستند که حول یک خط متمرکز شده اند. به صورت چشمی می بینیم که بین سیگنال ورودی و خروجی رابطه خطی وجود دارد که شیب آن در حدود ۲ و عرض از مبدا نیز حدودا مقدار -۰.۵ را دارد.



بخش ۴ حال برای محاسبه این پارامتر های مجهول شیب(alpha) و عرض از مبدا (beta) یک راه طبق راهنمایی سوال مینیمم کردن تابع خطای بین مقدار واقعی و مقدار تخمین زده شده ما در رابطه خطی است. در واقع ما می خواهیم α, β بهینه ای را به گونه ای بیابیم که تابع RMS را کمینه کنند. می دانیم که برای یافتن مینیمم از ان تابع مشتق (گرادیان) گرفته و انرا برابر با صفر قرار می دهیم. یک بار نسبت به α و بار دیگر نسبت به β مشتق گرفته و البته که در هر بار متغیر دیگری را عدد ثابت در نظر می گیریم.

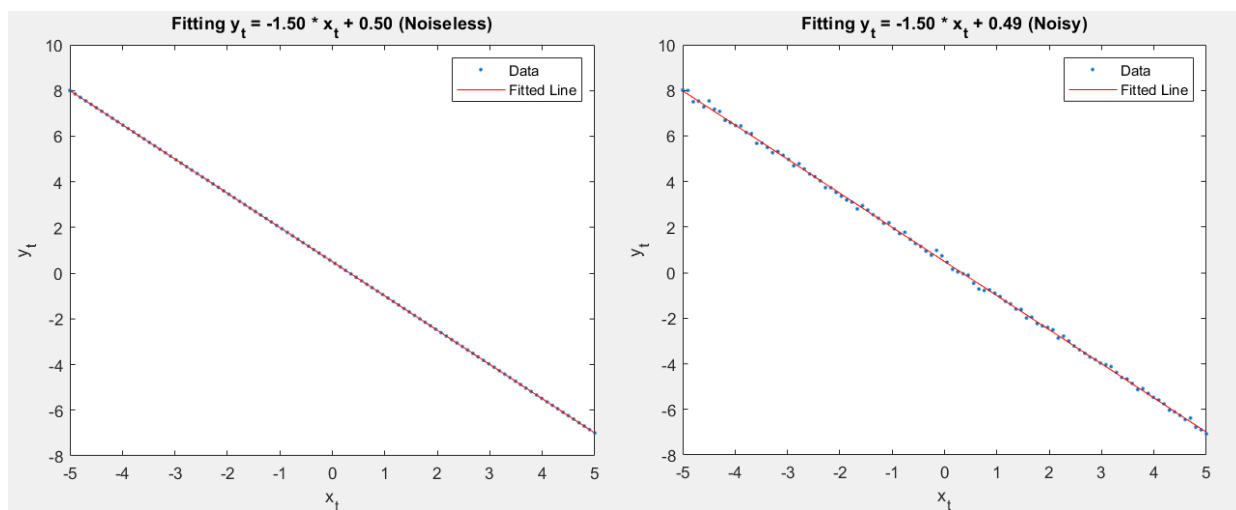


ما در این سوال در ابتدا مقدار واقعی شیب و عرض از مبدا را نداشته و در نتیجه نتوانستیم که خروجی تخمین خود را با مقدار واقعی مقایسه کنیم. حال برای بررسی صحت عملکرد تابع مان بایستی ۳ سری داده که با یکدیگر رابطه خطی مشخصی دارند را ایجاد کرده و برای اینکه ببینیم که تابع مان به نویز هم robust هست یا خیر یک بار خروجی را با نویز و بار دیگر بدون نویز بررسی کرده و بعد چک می کنیم که با در نظر گرفتن threshold ای برای خطا، آیا شیب و عرض از مبدا ما با پارامتر های واقعی تطابق دارند یا خیر.



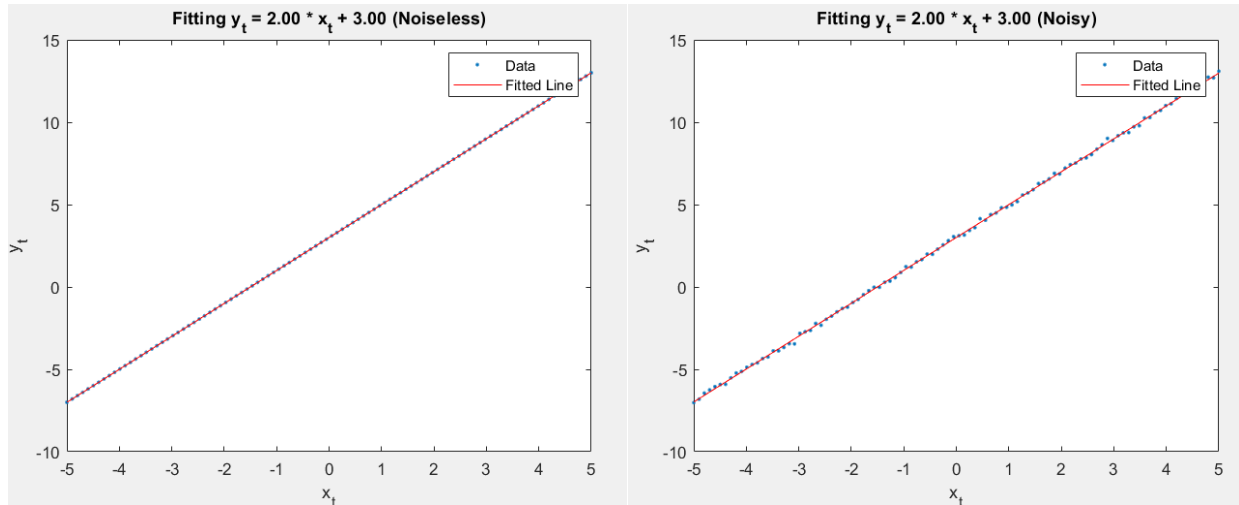
Results for Noiseless data:
 True alpha: 0.8000, Optimized alpha: 0.8000
 True beta: -2.0000, Optimized beta: -2.0000
 The optimized parameters match the true values!

Results for Noisy data:
 True alpha: 0.8000, Optimized alpha: 0.8009
 True beta: -2.0000, Optimized beta: -2.0084
 The optimized parameters match the true values!



Results for Noiseless data:
 True alpha: -1.5000, Optimized alpha: -1.5000
 True beta: 0.5000, Optimized beta: 0.4999
 The optimized parameters match the true values!

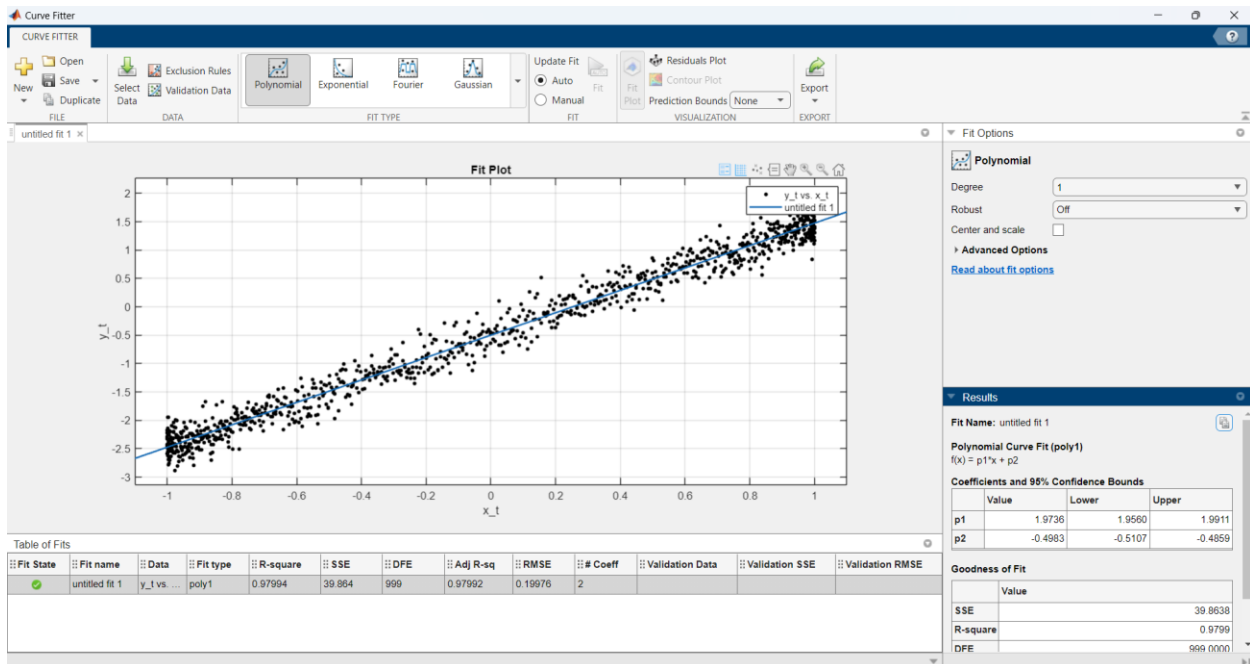
Results for Noisy data:
 True alpha: -1.5000, Optimized alpha: -1.4978
 True beta: 0.5000, Optimized beta: 0.4917
 The optimized parameters match the true values!



Results for Noiseless data:
 True alpha: 2.0000, Optimized alpha: 2.0000
 True beta: 3.0000, Optimized beta: 3.0000
 The optimized parameters match the true values!

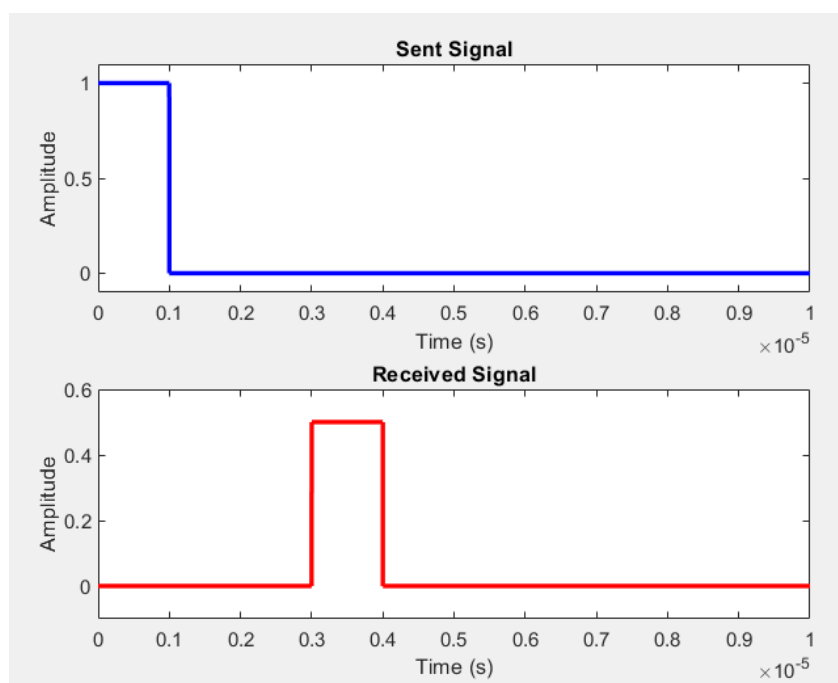
Results for Noisy data:
 True alpha: 2.0000, Optimized alpha: 1.9970
 True beta: 3.0000, Optimized beta: 3.0033
 The optimized parameters match the true values!

بخش ۵)



سوال ۳

بخش ۱۹۲) دستگاهی به نام رادار وجود دارد که یکی از کاربردهای آن این هستش که سیگنال الکترومغناطیسی رو فرستاده و این سیگنال پس از برخورد به یک جسم منعکس شده و ما می‌توانیم سیگنال الکترومغناطیسی دیگری را دریافت کنیم. در این سوال ما می‌خواهیم با بررسی مدت زمانی تاخیر بین ارسال سیگنال اولیه و دریافت سیگنال دوم فاصله میان دستگاه رادار و جسم منعکس کننده را پیدا کنیم. در این سوال فرض شده در ابتدا که ما سیگنال ارسال شده و سیگنال دریافتی رو داریم و فقط می‌خواهیم به کمک matlab این سیگنال‌ها را رسم کنیم. برای رسم این دو نمودار ابتدا بردار زمانی t را معرفی می‌کنیم که طبق خواسته صورت سوال طول آن T ثانیه هستش و این بردار رو هم به استپ‌های t_s ثانیه تقسیم می‌کنیم. سیگنال x_t به صورت یک پالس مستطیلی هست که عرض این پالس τ ثانیه هستش و دامنه اون یک ثانیه هست. این τ ثانیه رو به t_s تقسیم می‌کنیم تا تعداد نقاطی که از بردارمان باید مقدار دامنه یک رو به داشته باشن رو حساب کنیم (N) و به بردار صفرای که طول آن به اندازه بردار زمانمون هست از مبدا تا ایندکس N مقدار دامنه یک رو نسبت میدیم. حال برای رسم سیگنال دریافتی که ان هم به صورت یک پالت مستطیلی هست اما با این تفاوت که دامنه اش برابر α هست و از مبدا زمانی هم t_d ثانیه فاصله زمانی دارد. حال برای محاسبه ایندکس t_d رو تقسیم بر t_s می‌کنیم و سپس برای رسم بردار سیگنال دریافتی از ایندکس t_d تا ایندکس $t_d + N$ به اضافه N مقدار دامنه α را به بردار سیگنال دریافتی نسبت می‌دهیم. و اینکه برای محاسبه تی دی از رابطه‌ای که توی صورت سوال داده شده بود که برابر هستش با دو R به روی c استفاده کرده و در این بخش از سوال هم فرض شده که R برابر با ۴۵۰ هست.



بخش ۳) در بخش بعدی می‌خواهیم که R رو به کمک به دست آوردن t_d محاسبه کنیم خب یک راه برای محاسبه t_d استفاده از روش template matching یا correlation هست. این روش از یک سیگنال به نام سیگنال سمپل استفاده می‌کنیم که این سیگنال را هم باید شبیه سیگنال دریافتی مان تعیین کنیم چرا که در روش correlation ما سیگنال سمپل رو در طول بردار زمان استپ به استپ جابجا می‌کنیم تا بتوانیم به استپ یا ایندکس یا فاصله زمانی برسیم که این سیگنال سمپل با سیگنال دریافتی مان به بیشترین تطابق برسند یا بیشترین حاصل ضرب داخلی رو با همدیگه داشته باشند. پس ما توی روش correlation برای ذخیره حاصل حاصل ضرب داخلی بین سیگنال دریافتی و سیگنال سمپل مان برداری به نام ro رو معرفی کردیم که طولی به اندازه طول بردار زمانی منهای عرض پالس مان (N) دارد. برای به دست آوردن ایندکس لحظه‌ای که بیشترین حاصل ضرب داخلی بین دو سیگنال دریافتی و سیگنال سمپل وجود دارد باید ایندکس ماکسیمم بردار ro رو به دست بیاریم. و حال برای به دست آوردن مدت زمانی t_d ایندکسی که برای t_d به دست آوردیم رو منهای یک کرده که تحلیل ساده‌ای هم داره و بعد اون رو ضربدر t_s می‌کنیم از فضای ایندکس به فضای زمانی برگردیم. در نهایت برای محاسبه فاصله میان دستگاه رادار و جسم منعکس کننده سرعت سیگنال رو که با c تو صورت سوال به ما داده شده و t_d رو که به دست آوردیم و با جایگذاری توی رابطه می‌تونیم R رو به دست بیاریم. طبق انتظارمان R که ما تخمین زدیم با R که صورت سوال داده بود با هم برابر هستند.

True R: 450.00 meters

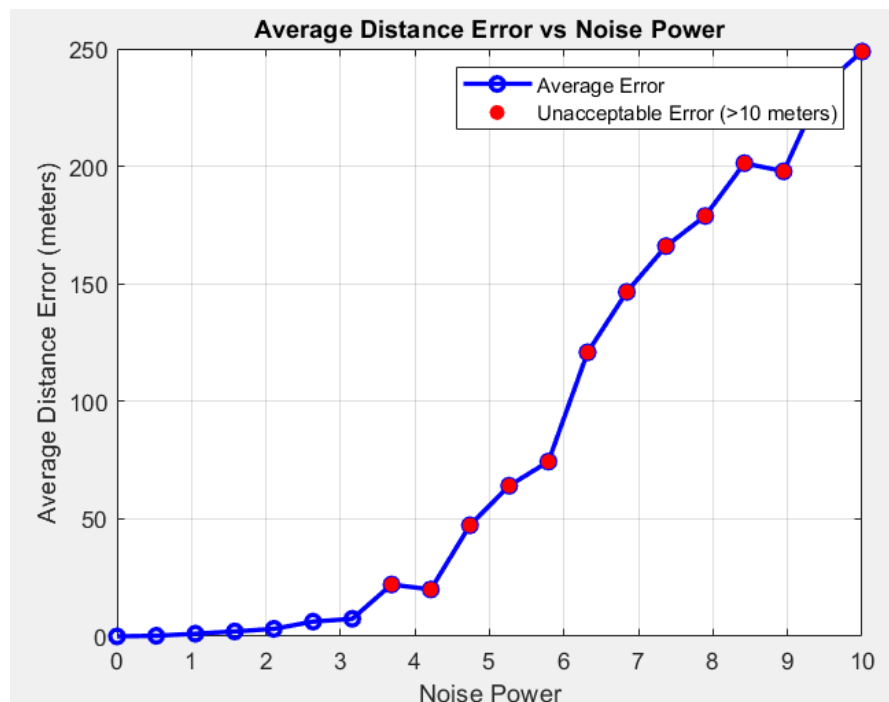
Estimated R: 450.00 meters

بخش ۴) برای بخش بعدی از ما می‌خواهد که تاثیر نویز رو بر روی سیگنال دریافتیمون بررسی کنیم و ببینیم با تابعی که ما تعریف کردیم تا چه مقدار نویز همچنان می‌تونیم جواب خوبی با خطای ۱۰ متر پیش بینی کرد. دوباره مثل بخش قبلی سیگنال سمپل معرفی می‌کنیم و می‌خواهیم که correlation میان این رو محاسبه کنیم ولی این بار درجه نویز سیگنال دریافتیمان را به صورت یک بردار تعریف می‌کنیم که توی یک حلقه می‌خواهیم روی این درجه‌های نویز مختلف correlation را محاسبه کنیم و پس از به دست آوردن ماکسیمم مثل قبل t_d را به دست آورده و پس از محاسبه t_d می‌تونیم R به دست بیاریم و در پایان هم پس از محاسبه R که تخمین زدیم اون را میایم با R واقعی مقایسه می‌کنیم و اگر که اختلافشان کمتر از خطای ۱۰ متر بود نتیجه را قابل قبول معرفی می‌کنیم اما اگه خطا بالاتر بود اولین لول نویزی که باعث خطای بیش از ۱۰ متر بشود را ما به عنوان مرز کارکرد صحیح این تابع معرفی می‌کنیم. در انتها باید اشاره هم بکنم که در این سوال به خاطر اینکه توی بخش‌های مختلفش مقدار زیادی پارامتر ثابت داشتیم و برای جلوگیری از نوشتن دوباره این چند خط پارامتر ثابت توی هر فایل اسکریپت در یک فایل به صورت جداگانه تمامی پارامترهای ثابت رو نوشته و اون رو در `parameters.m` کرده و در ابتدای هر فایل اسکریپت که به این پارامترها نیاز بود تنها

اسم فایل اسکریپت رو در یک خط کد آورده و این کار باعث می‌شود که همه پارامترها باز در اون فایل اسکریپت شامل شده باشند و ما بتونیم از اونها تو ادامه کدمان استفاده کنیم .

```
% Add Gaussian noise to the clean received signal
noise = noise_power * randn(size(received_signal_clean));
received_signal_noisy = received_signal_clean + noise;
```

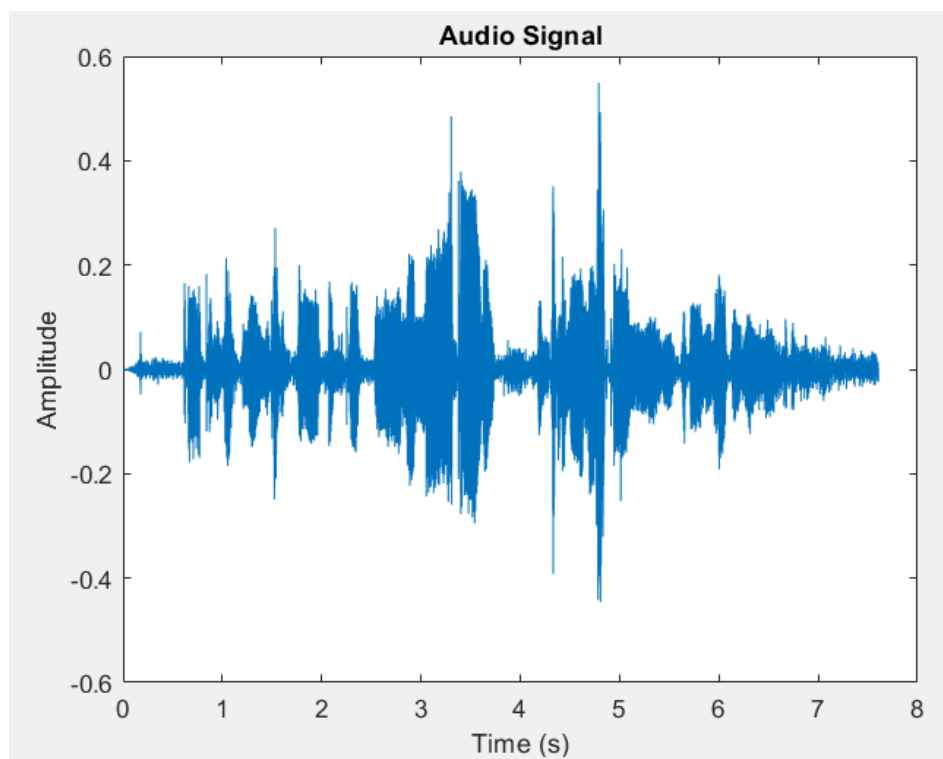
```
Noise Power = 0.0000: Distance correctly estimated with average error = 0.0000 meters
Noise Power = 0.5263: Distance correctly estimated with average error = 0.2490 meters
Noise Power = 1.0526: Distance correctly estimated with average error = 1.1295 meters
Noise Power = 1.5789: Distance correctly estimated with average error = 2.0880 meters
Noise Power = 2.1053: Distance correctly estimated with average error = 3.1905 meters
Noise Power = 2.6316: Distance correctly estimated with average error = 6.2700 meters
Noise Power = 3.1579: Distance correctly estimated with average error = 7.4760 meters
Noise Power = 3.6842: Distance estimation failed, average error = 22.0560 meters
Noise Power = 4.2105: Distance estimation failed, average error = 19.9470 meters
Noise Power = 4.7368: Distance estimation failed, average error = 47.2665 meters
Noise Power = 5.2632: Distance estimation failed, average error = 64.0950 meters
Noise Power = 5.7895: Distance estimation failed, average error = 74.3685 meters
Noise Power = 6.3158: Distance estimation failed, average error = 120.8295 meters
Noise Power = 6.8421: Distance estimation failed, average error = 146.5695 meters
Noise Power = 7.3684: Distance estimation failed, average error = 166.0335 meters
Noise Power = 7.8947: Distance estimation failed, average error = 178.8795 meters
Noise Power = 8.4211: Distance estimation failed, average error = 201.3255 meters
Noise Power = 8.9474: Distance estimation failed, average error = 197.8680 meters
Noise Power = 9.4737: Distance estimation failed, average error = 234.1260 meters
Noise Power = 10.0000: Distance estimation failed, average error = 248.8260 meters
```



سوال ۴

بخش ۱) در این سوال ما به بررسی فرکانس و سرعت سیگنال‌های صوتی می‌خواهیم بپردازیم. در ابتدا می‌خواهیم بررسی کنیم که چگونه می‌توانیم توی matlab یک سیگنال صوتی را که به صورت فایل wav ذخیره شده، بخونیم برای این کار matlab تابعی به نام audioread دارد که خروجی اش شامل سیگنال صوتی مان و فرکانس نمونه برداری مان هست (f_s) توی هر ثانیه اومده از سیگنال صوتی مون چند تا سمپل برداشته).

توی بخش دوم این سوال می‌خواهیم این بردار صوتی رو که به دست آوردیم را در نموداری رسم بکنیم که محور افقی این نمودار را هم زمان قرار میدیم. متغیر زمان را برداری قرار میدیم که طولش برابر با طول بردار صوتی مان برابر هست ولی برای اینکه از فضای ایندکس به فضای زمان منتقل بشیم اون رو ضربدر t_s یا تقسیم بر f_s می‌کنیم و همچنین در این سوال از ما خواسته شده که با دستور matlab برای پخش کردن سیگنال صوتی هم کار کنیم که دستور sound هست که سیگنال صوتی و فرکانس نمونه برداری رو به عنوان ورودی دریافت می‌کند. همچنین برای ذخیره سیگنال صوتی توی matlab می‌توانیم از دستور audiowrite استفاده کنیم که ورودی‌هاش بردار سیگنال صوتی مان، فرکانس نمونه برداری و اسم فایل صوتی که می‌خوایم اون رو به اون نام سیو کنیم هستند.



بخش ۲) این بخش سوال از ما می‌خواهد که تابعی بنویسیم که سرعت بخش سیگنال را عوض کند یعنی ما بتوانیم توی ورودی ضربی برای افزایش سرعت سیگنال یا کاهش سرعت سیگنال داشته باشیم. البته در این بخش اگر که این ضریب تغییر سرعت سیگنال چیزی جز نیم یا دو باشد ما باید ارور مقدار نادرست رو برای سرعت چاپ بکنیم. در بدنه تابع برای پیاده سازی بخش افزایش دو برابری سرعت بردار x_fast رو این گونه تعریف می‌کنیم که می‌خواهیم از بردار صوتی اولیه مان به صورت یک درمیان نمونه برداری کرده، به همین خاطر بردار صوتی اولیه مان را از ایندکس اول با استپ‌های دوتایی طی می‌کند. فقط برای پخش این سیگنال هم باید توجه داشته باشیم که حالا این سیگنال صوتی سریعتر ما فرکانس نمونه برداری اش هم دو برابر شده است. برای کاهش سرعت سیگنال به نصف، سیگنال x_slow را تعریف می‌کنیم که طولش برابر هستش با دو برابر طول سیگنال اولیه مان، چرا که طبق راهنمایی صورت سوال ما باید بین مقادیر سیگنال صوتی اولیه مان به صورت یک درمیان این مقادیرهای جدید رو اضافه کنیم که عملاً فرکانس نمونه برداری مان رو هم نصف کردیم. حالا این مقدار جدید میانگین دو تا مقدار مجاورش هست که از سیگنال اولیه هستند است. برای جایگذاری این مقادیرهای سیگنال x_slow می‌ایم یک بار ان مقادیری رو که شبیه مقادیر سیگنال اولیه مان هستند را مشخص می‌کنیم یعنی مقادیر ایندکس های فرد سیگنال x_slow برابر مقادیر سیگنال اولیه مون هستند برای مقادیر ایندکس های زوج سیگنال x_slow می‌ایم از روش میانگین گیری از مقدار سیگنال اولیه مون استفاده می‌کنیم و در نهایت هم برای پخش این سیگنال کند شده باید دقت داشته باشیم که به تابع `sound` باید فرکانسی، نصف فرکانس سیگنال اولیه رو به عنوان ورودی بدهیم.

توی بخش چهارم این سوال، باید تابعی رو بنویسیم که سرعت پخش سیگنال صوتی مان را عوض بکند ولی فقط محدود به نصف یا دو برابر نباشد و هر سرعتی رو بین نصف تا ۲ بتونه تنظیم بکند البته که باید این سرعت‌ها مضربی از یک دهم باشند. اگر هم که ضریب سرعت کمتر از نیم یا بیشتر از دو بود، خطای سرعت غیر قابل قبول رو بدهد. سه تا حالت را در نظر می‌گیریم یک حالت که سرعت دو برابر باشد و یک حالت که سرعت نصف باشه و یک حالت هم اینکه سرعت بین نصف تا دو برابر باشد. برای دو تا حالت اول از همان روشی که توی سوال قبل نوشتیم استفاده می‌کنیم. حالت سوم هم از روش درون یابی می‌رویم. یک فایل اسکریپت هم برای پیاده سازی این توابعی نوشتیم که در کنار فایل‌های این چهار بخش دیگه سوال داریم و توی این فایل اسکریپت از دستور `pause` استفاده کردیم که با کمک این دستور، پس از اجرای یک تابع به اندازه چند ثانیه (ارگومان `pause`) توقف می‌کند و بعد تابع بعدی رو صدا می‌زند.

پایان