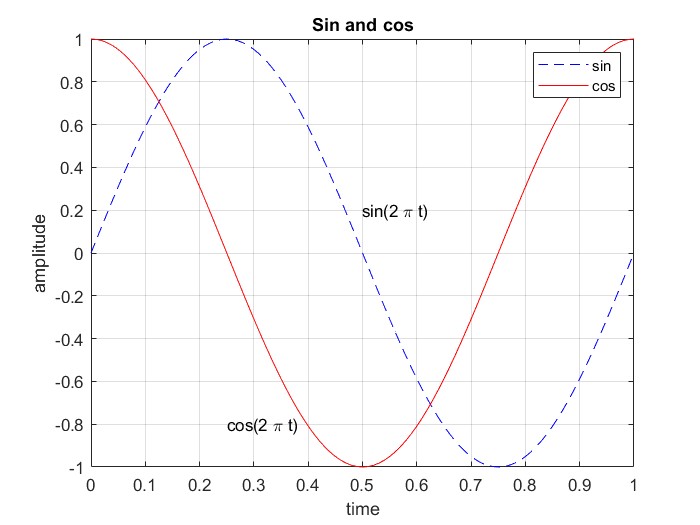
گزارش کار تمرین کامپیوتری یک سیگنال و سیستم

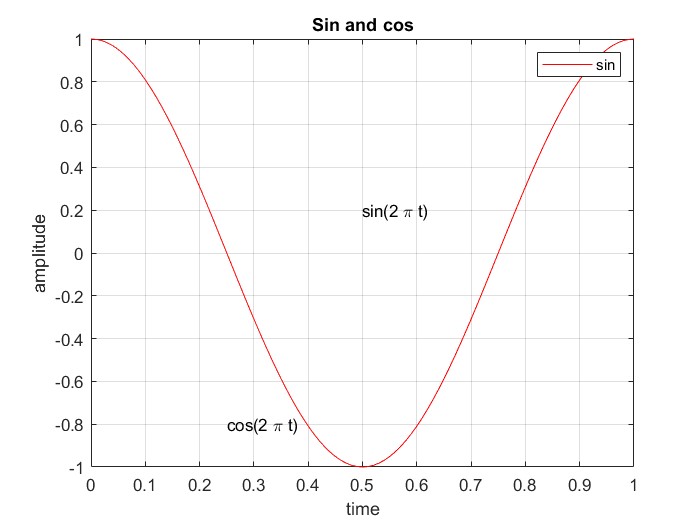
مرضیه موسوی-810101526

1-1

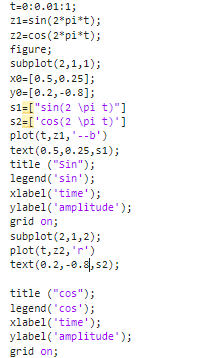
شکل نهایی به صورت زیر است:



در صورت حذف خط 7 نمودار اول که مربوط به سینوس است روی فیگور نمی ماند:

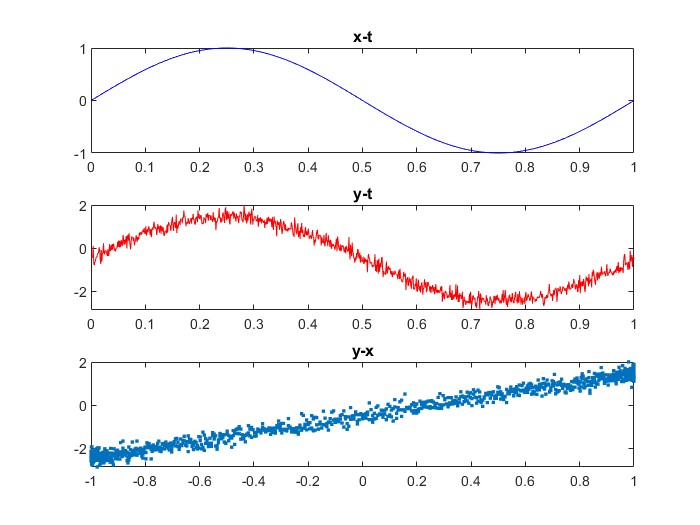


1-2



بخش دوم:

شکل نمودار های بخش یک تا سه به شرح زیر است:

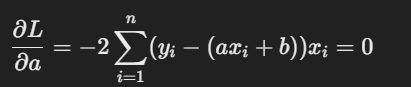


به نظر می رسد شیب نمودار y-x که همان ضریب آلفا در رگرسیون خطی ماست تقریبا برابر 2 و عرض از مبدا که همان ضریب بتا در رگرسیون خطی است تقریبا کمی کوچکتر از 0 است.

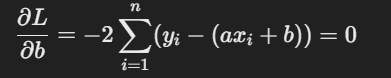
تمرین 2-4

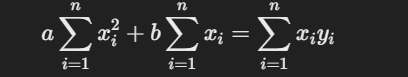
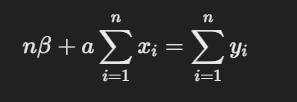
باید از loss function داده شده یکبار نسبت به آلفا و بار دیگر نسبت به بتا مشتق بگیریم و با حل کردن معادلات داده شده آلفا و بتا مورد نظر را بیابیم.

مشتق تابع loss داده شده نسبت به آلفا:



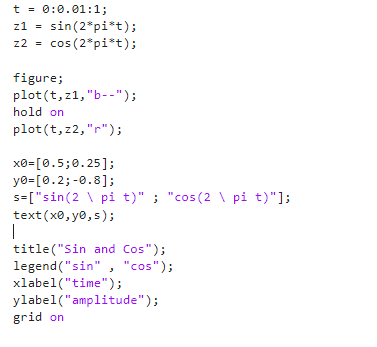
مشتق تابع loss داده شده نسبت به بتا:



بنابراین باید از دو معادله ی زیر آلفا و بتا را بدست آوریم:  
  


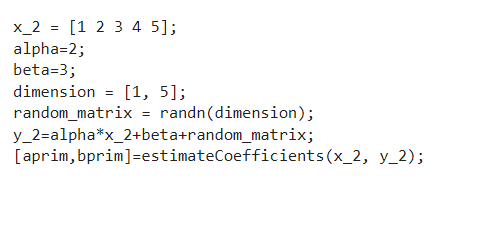
اسکریپت متلب به صورت زیر است:

که مطابق با معادلات به دست آمده است.

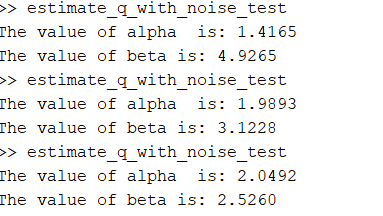


بخش تست:

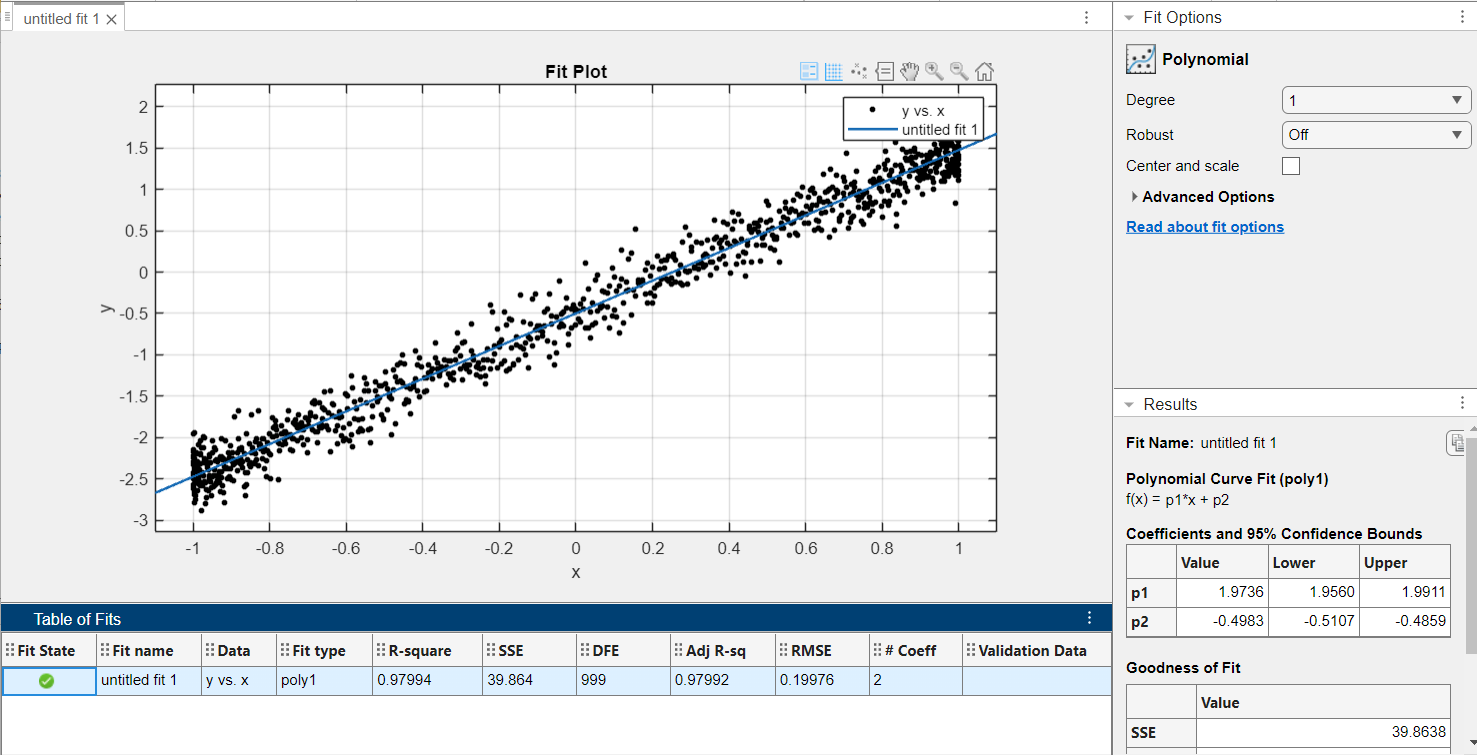
اسکریپت زیر برای تست نوشته شده:



همانطور که در کامند مشاهده می کنید آلفا و بتای بدست آمده نزدیک به آلفا و بتای محاسبه شده می باشد:



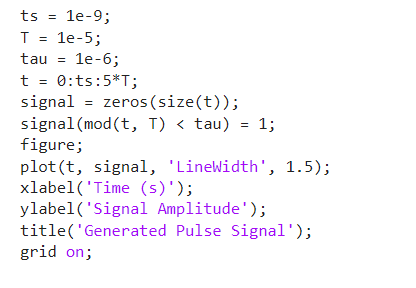
2-5)



همانطور که در تصویر مشخص است نتایج همخوانی دارد

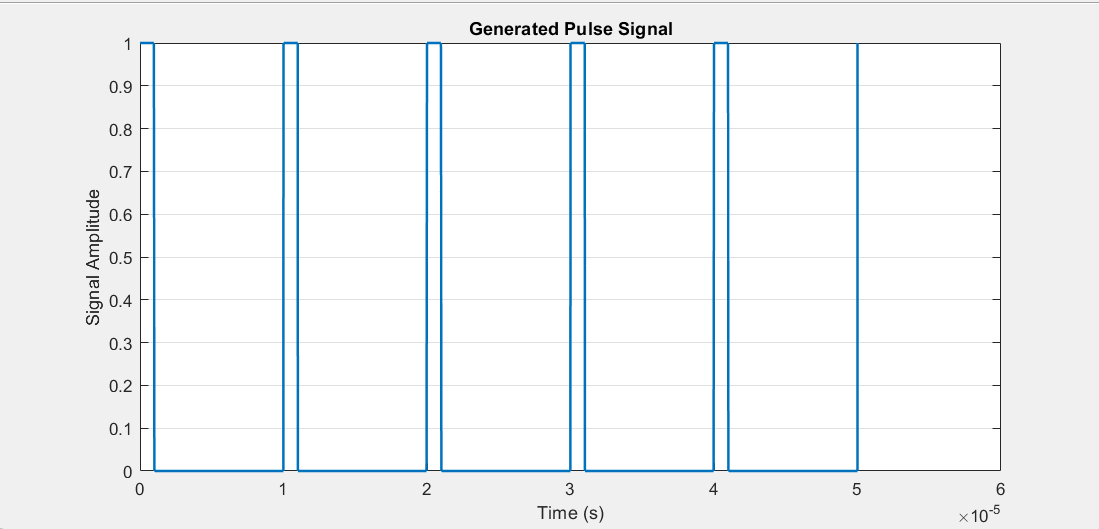
بخش سه:

1-3)

اسکریپت متلب به صورت زیر است:  


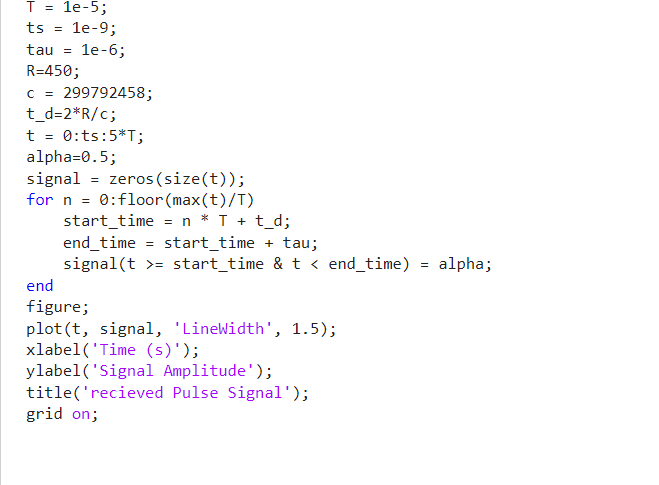
همانطور که مشخص است ابتدا وکتوری از 0 به اندازه بازه زمانیمان گرفتیم.سپس در هر بازه زمانی اگر بین 0 وtau را یک قرار دادیم.(t هایی که در آن بازه هستند)

شکل سیگنال به صورت زیر است:



3-2)

اسکریپت متلب به صورت زیر است:

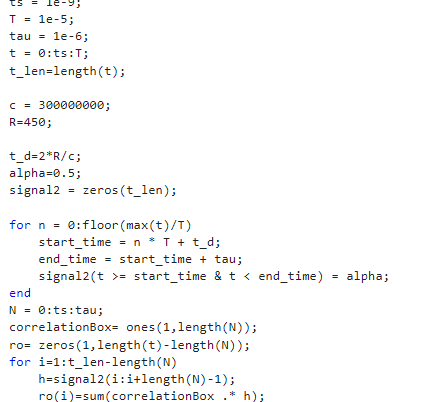


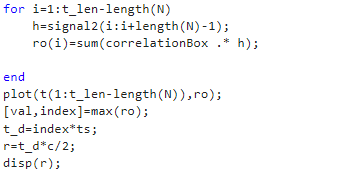
همانطور که مشخص است یک وکتور سیگنال به اندازه ی بازه زمانی و با مقدار 0 گرفته ایم و سپس در هر دوره ی تناوب ابتدا زمان شروع سیگنال و زمان پایان آن را حساب کرده ایم و آن بخش را به اضافه ی آلفا کردیم.

3-3)

می دانیم کورلیشن دو سیگنال مربعی یک سیگنال مثلثی می باشد.در این سوال اگر سیگنالی به عنوان باکس کانولوشن و سیگنال دریافتی را باهم کانولوشن بگیریم چون ماکسیمم سیگنال مثلثی ایجاد شده در td می باشد بنابراین td بدست می آید و با بدست آورد td می توانیم R را حساب کنیم.

اسکریپت متلب به صورت زیر است:

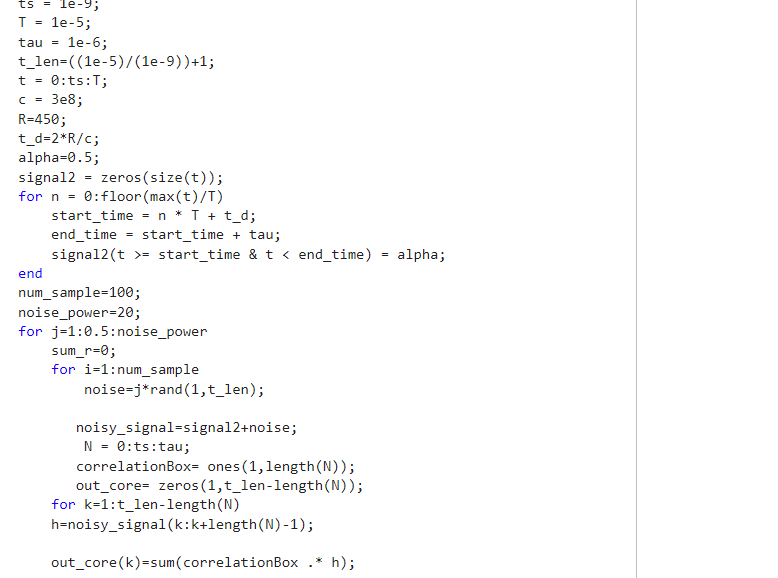


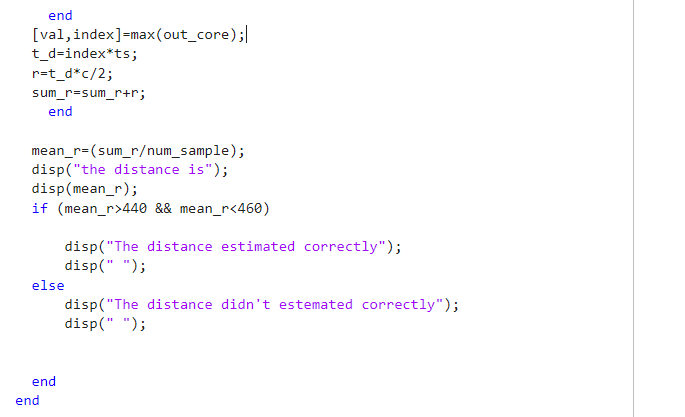


سیگنال دریافتی را با روشی که قبلا رسم کرده ایم دوباره وکتور آن را بدست می آوریم.کورلیشن باکس را نیز بدست می آوریم و سپس ro که سیگنال خروجی است را بدست می آوریم.مکسیمم مقدار آن در نقطه ی tdاست و با استفاده از td بدست آمده R را محاسبه می کنیم.

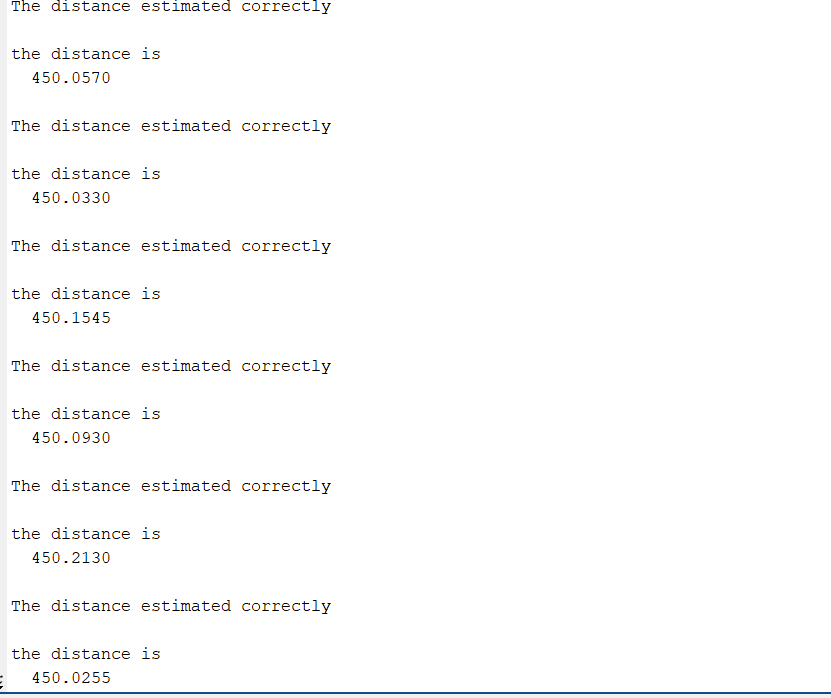
3-4)

به اسکریپت قبلی یک حلقه اضافه می کنیم.حلقه ی بیرونی قدرت نویز را تعیین می کند.عدد رندومی به دست آورده و آن را در قدرت نویز ضرب می کنیم و به سیگنال دریافتی اصلی اضافه می کنیم.از روش تمرین قبلی استفاده می کنیم و r را در هر 100 نمونه بدست آورده و میانگین می گیریم.اگر میانگین بین 440 و 460 بود یعنی خطای ده درصدی داشت تقریب ما درست است.در غیر این صورت تقریب ما مشکل دارد.





خروجی به صورت زیر است:

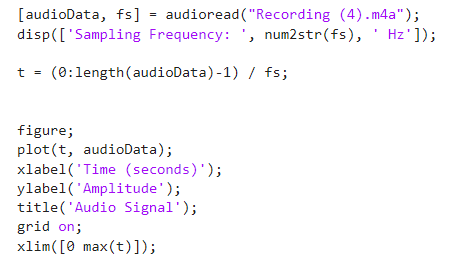


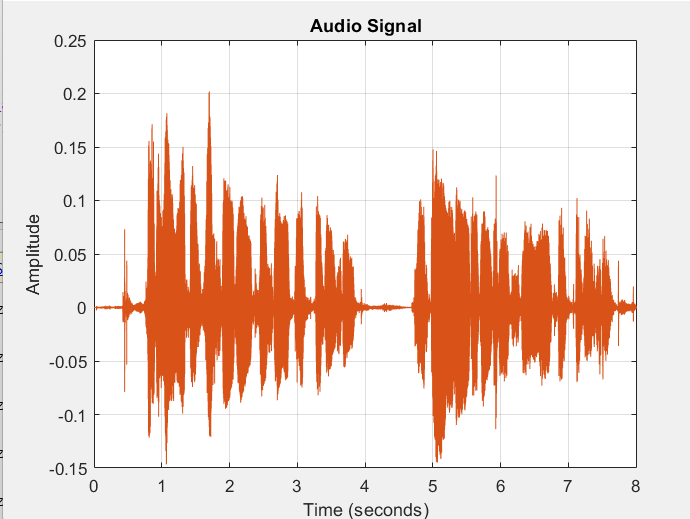


بخش چهار)

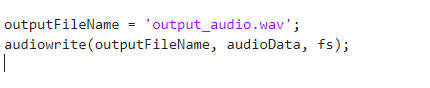
4-1)

ابتدا با استفاده از دستور audio read فرکانس نمونه برداری و دیتای صوت را خواندیم.برای نمایش اندازه ی دیتای خوانده شده را تفسیم بر فرکانس کردیم.بازه زمانی را تا آخرین t قرار دادیم و نمودار را کشیدیم

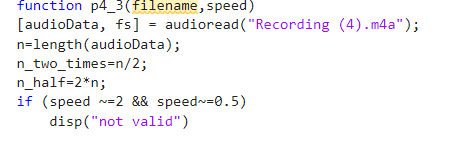




4-2)

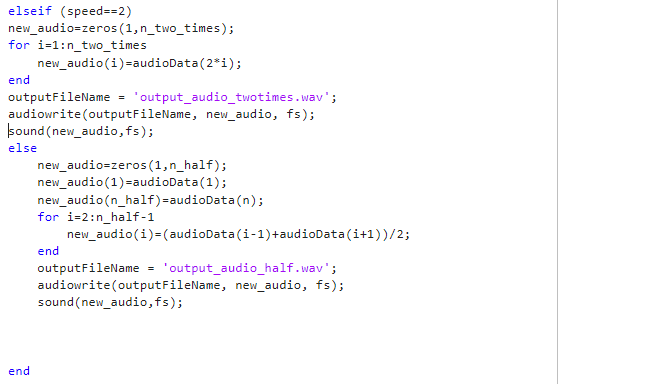
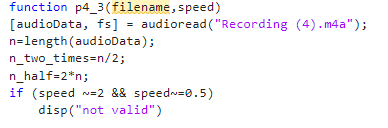
نحوه ی سیو کردن صوت به صورت زیر است:  


4-3)

ابتدا برای مقاوم سازی شرط گذاشتیم و ولیو های نامعتبر را نپذیرفتیم.همچنین سایز وکتور صوت جدید را با استفاده از اندازه ی صوت اولیه بدست آوردیم:  


برای سرعت دو برابر یکی درمیان دیتاها را حذف کردیم:

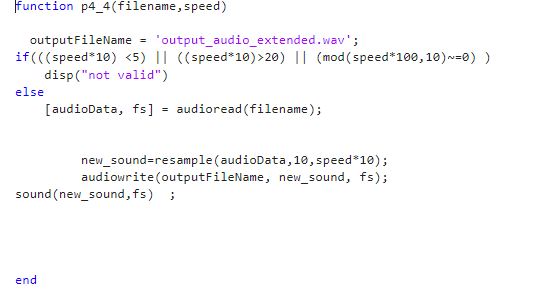
برای سرعت نصف اولین و آخرین دیتا متناظر صوت اولیه می باشد و برای بقیه از میانگین دو دیتای مجاور استفاده کردیم



4-3)

برای مقاوم سازی کد مقادیر بزرگتر از 2 و کوچکتر از نیم و هرچه مضرب یک دهم نبود را نپذیرفتیم

از تابع resample برای ساخت فایل جدید استفاده می کنیم.



هر دو تابع را در part 4 تست کردیم:

