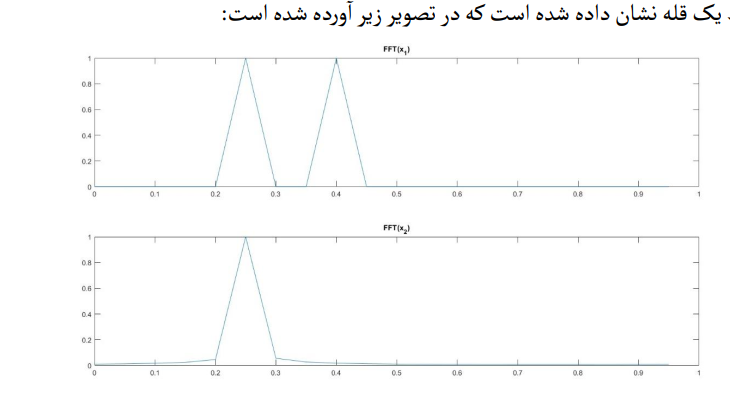
تمرین کامپیوتری 5 سیگنال ها و سیستم ها

مرضیه موسوی 810101526

مهراد لیویان 810101501

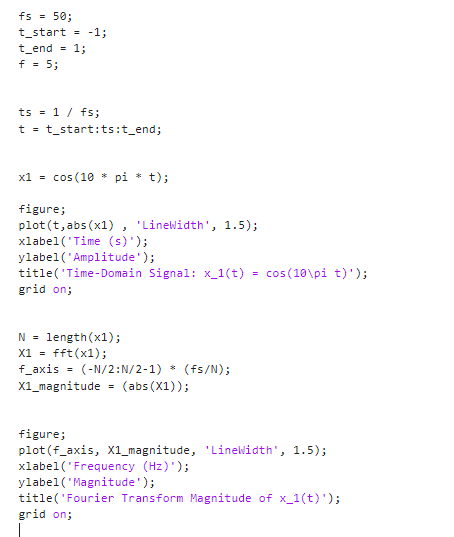
0-1:

در حالتی که فرکانسهای تابع را برابر با 5و 8 در نظر بگیریم، دو قله بر روی 5و 8به وضوح قابل مشاهده هستند. اما اگر این مقادیر را 5و 5.1درنظر بگیریم، چون اختلاف فرکانس تک تن ان کمتر از 1هرتز است این تفاوت قابل مشاهده نیستو فقط یک قله نشان داده شده است که در تصویر زیر اورده شده است:

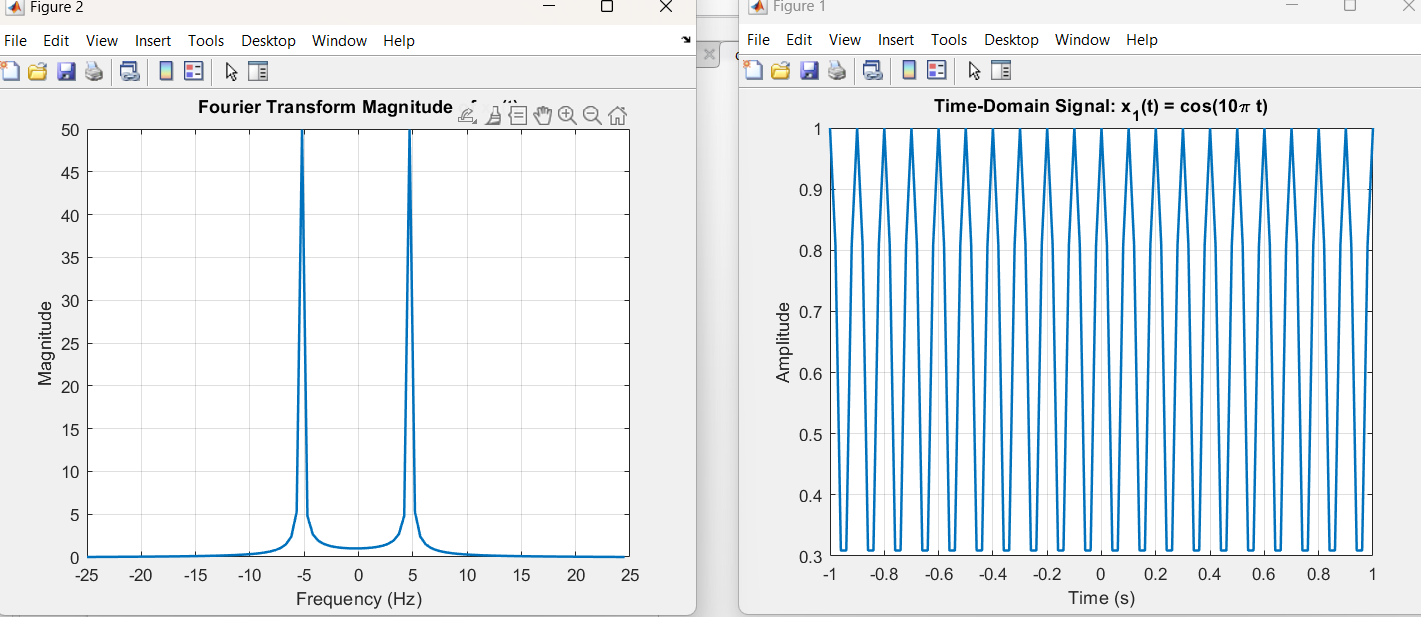


1-1:

اسکریپت زیر برای این تابع نوشته شده است:

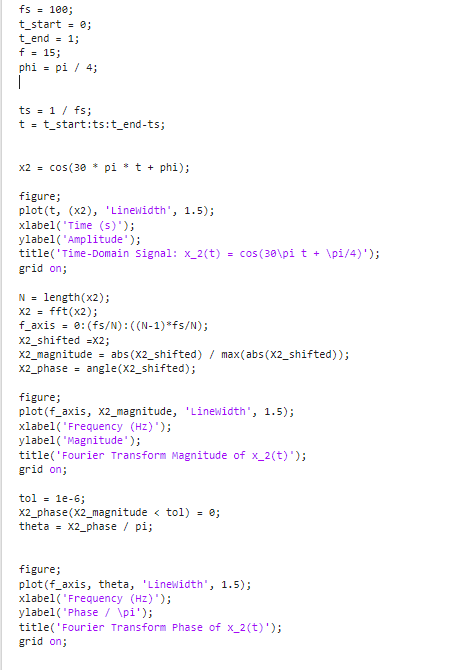


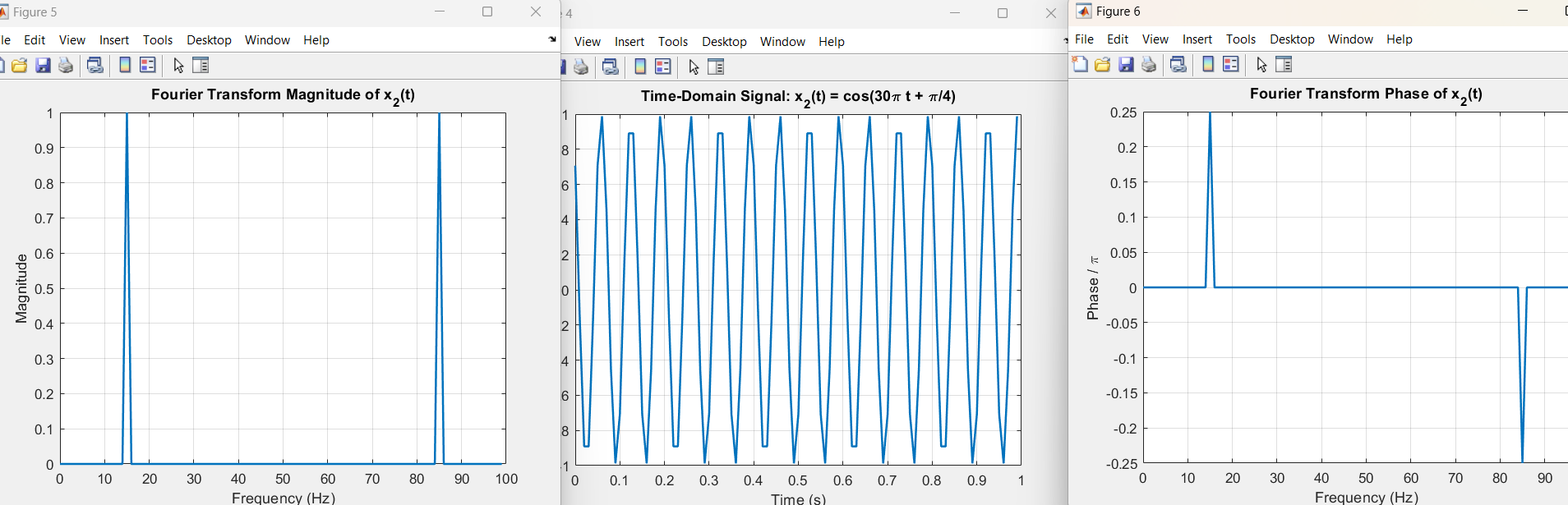
شکل های به دست آمده به صورت زیر است:



همانطور که در شکل مشخص است تبدیل فوریه به صورت مجموع دو ضربه به دست می آید که مطابق دانسته های ماست.

1-2:



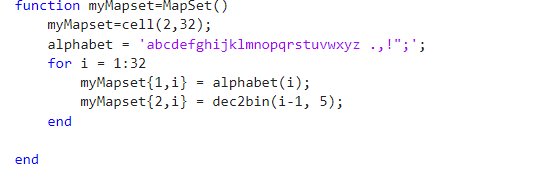


فرکانس آن 85 و 15 است که در صورت تبدیل فوریه گرفتن از تابع x به صورت مجموع دو ضربه در 85 و 15 به دست می آید که با دانسته ها مطابقت دارد.

بخش دوم:

تمرین 2-1 و 2-2:

دو اسکریپت زیر برای درست کردن mapset و کد کردن ورودی نوشته شده است:



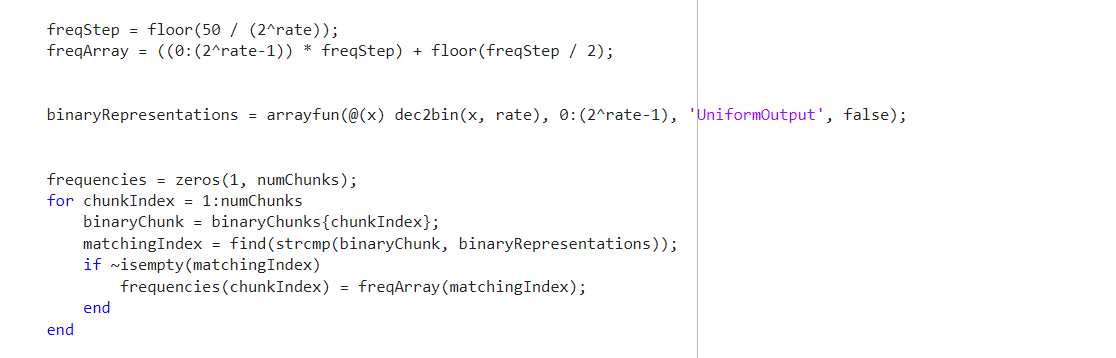
تمرین 2-3:



ابتدا mapset راinitializeمی کنیم.تعداد تکه هایی که می خواهیم کد کنیم را به دست می آوریم.

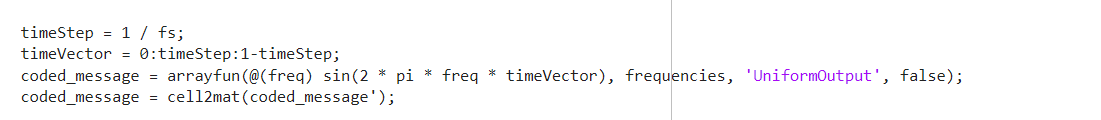
یک حلقه روی پیام می زنیم.اگر کاراکترش با کاراکتری که در مپ ست است مچ بود کد ان را به binary massage اضافه می کنیم.

بخش باینری را به تکه هایی تقسیم می کنیم.در صورتی که تعداد بیت های کل به ریت بخش پذیر نبود چند کاراکتر در انتها اضافه می کنیم.



آرایه ای متشکل از فرکانس هایمان درست می کنیم.

برای هر فرکانس یک نمایش باینری درست می کنیم و سپس به جای هر فرکانسی در binary chunk مان آن را می گذاریم.

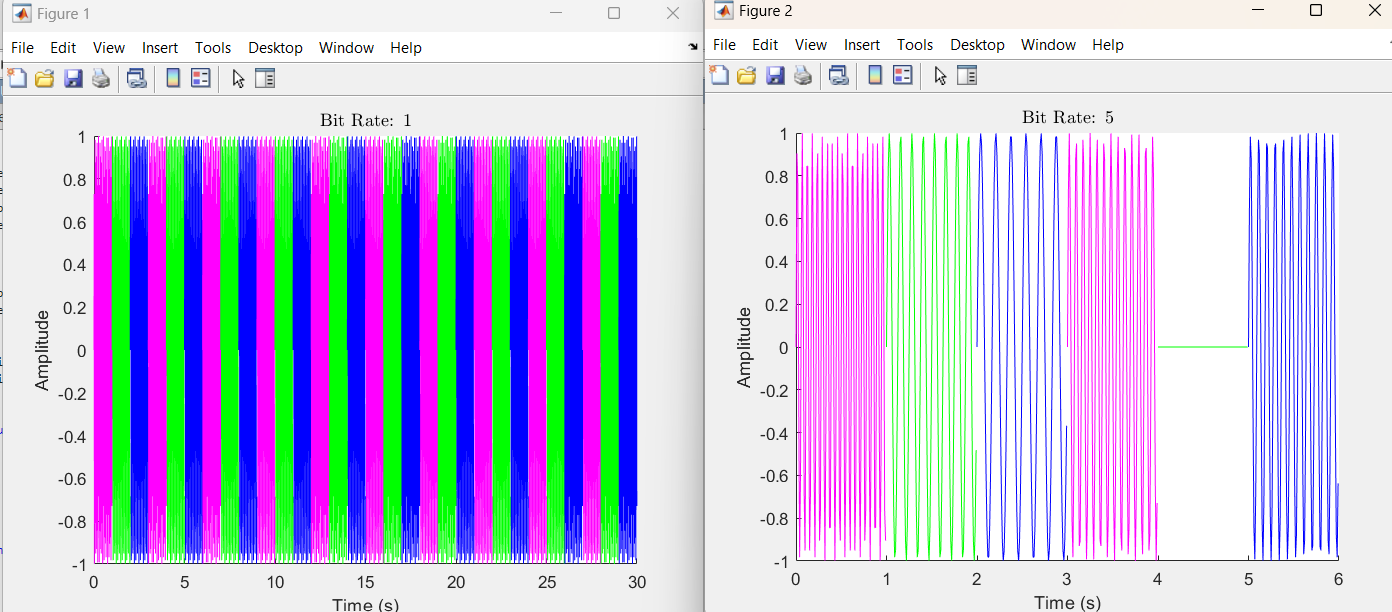


به ازای هر فرکانس یک موج سینوسی تولید می کنیم.



هر بخش از موج سینوسی با استفاده از آفستی مپ شده و کشیده می شود.

نتیجه ی تست:

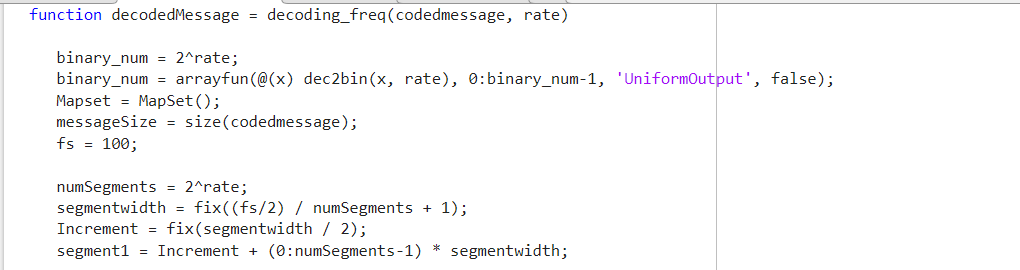


اسکریپت زیر برای تست نوشته شده است:



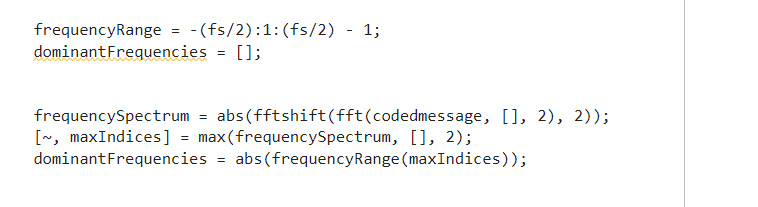
تمرین 2-4:

اسکریپت زیر مربوط به decode فانکشن است:

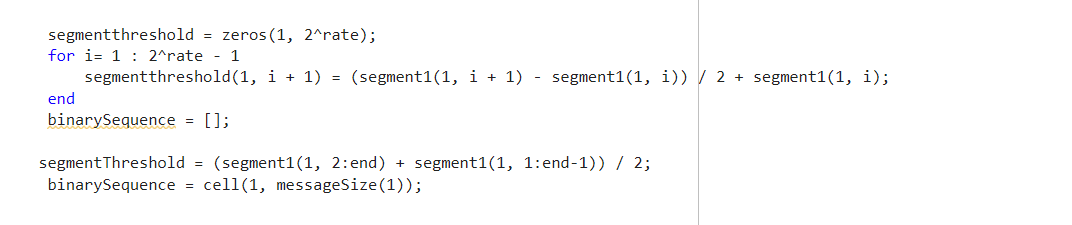


ابتدا تمام مقادیری را که براساس بیت ریت می توانیم داشته باشیم را بدست می آوریم.

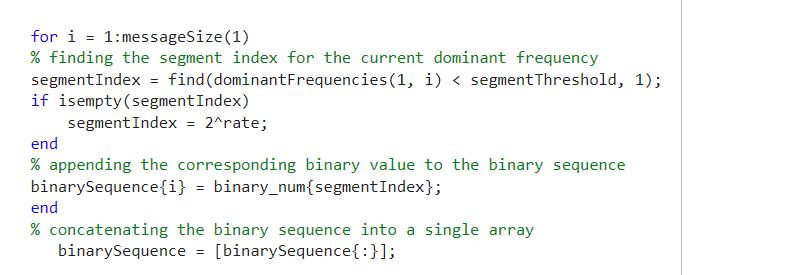
سپس width هر بخش فرکانس را بدست می آوریم.فرکانس مرکز هر بخش را بدست می آوریم تا برای بدست آوردن مقادیر باینری استفاده کنیم.



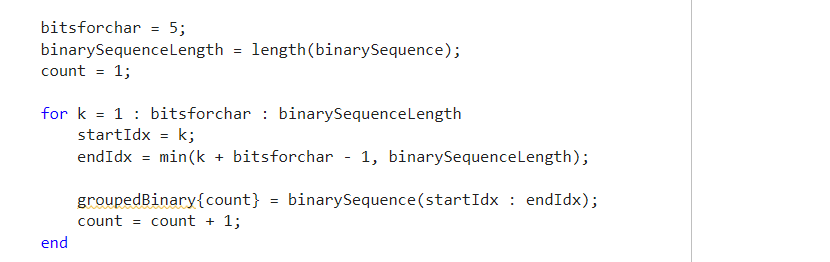
حال تبدیل فوریه مسیج را بدست می آوریم.ایندکس ماکس تبدیل فوریه را بدست آورده و مقدار آن را در dominant frequencies ذخیره می کنیم.



آستانه های بین های فرکانس را بدست می آوریم که بفهمیم هر فرکانس dominant در چه محدوده ای می باشد.

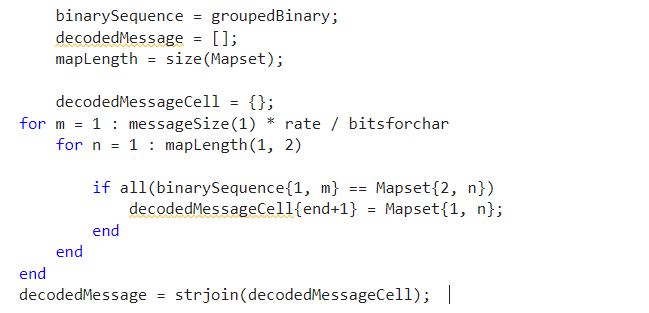


با استفاده از آستانه های بدست آمده بخش مربوط به هر فرکانس را پیدا می کنیم.سپس به مقادیر باینری مپ می کنیم و مقادیر باینری را concat می کنیم.

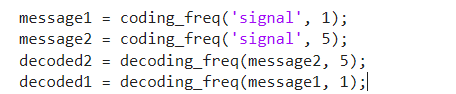


حال برای decoding سیکوینس بدست آمده را 5 تا 5 تا جدا می کنیم.

کاراکتر مربوط به هر بخش باینری را در مپست می یابیم و مسیج را دیکود می کنیم.



اسکریپت زیر برای تست نوشته شده است:

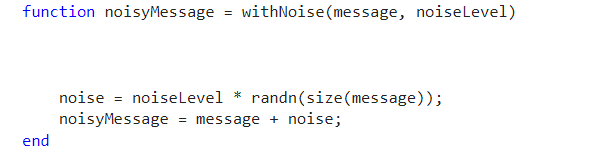


نتیجه ی تست به صورت زیر است:

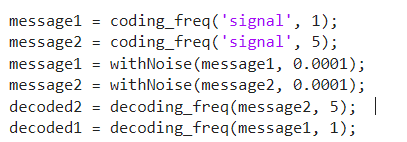


2-5:

تابع اضافه کردن noise به صورت زیر است:



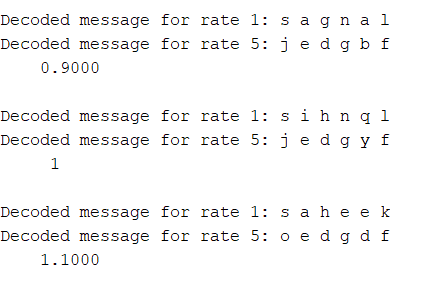
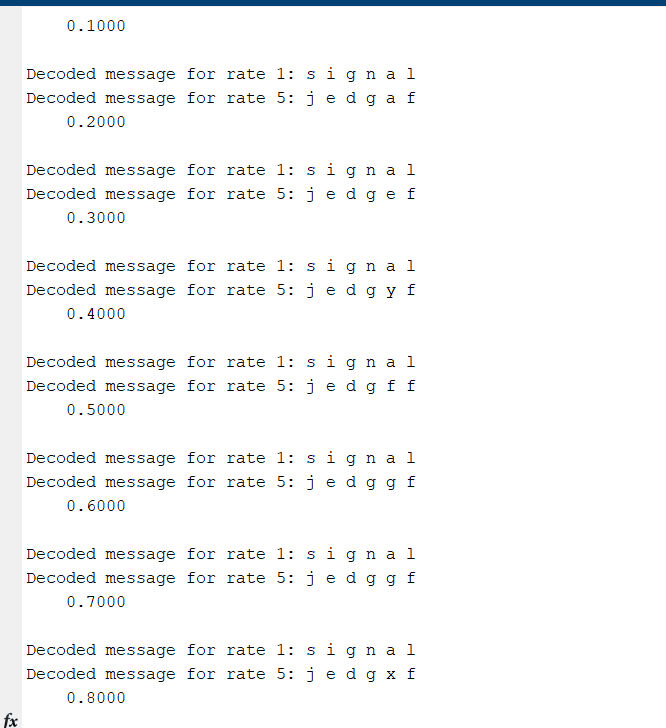
برای جنریت کردن نویز ابتدا سایز مسیج کد شده را بدست می آوریم.با استفاده از تابع randn اعداد رندوم را جنریت می کنیم و سیگما را در آن ضرب می کنیم.

اسکریپت بالا برای تست نوشته شده است.



همانطور که در خروجی مشخص است بیت ریت یک تفاوتی ندارد ولی بیت ریت 5 نتایج مرحله ی قبل را نمی دهد.

2-6



همانطور که از نتایج مشخص است بیت ریت 1 به نویز مقاوم تر بود.

2-7:

بیت ریت 5 با کوچکترین نویزی نتایج اشتباه می داد درحالی که بیت ریت یک تا حدود نویز 0.9 مقاوم است.

2-8:

اگر پهنای باند را بیشتر کنیم فاصله ی فرکانس هایی که وجود دارد را می توانیم بیشتر در نظر بگیریم و نسبت به نویز مقاوم تر است.

2-9:

وقتی نرخ نمونه‌برداری افزایش پیدا می‌کند، تعداد نمونه‌های سیگنال در هر بازه زمانی بیشتر می‌شود. این امر باعث می‌شود سیگنال بازسازی شده دقیق‌تر باشد و بتوان اطلاعات اصلی را با دقت بیشتری تشخیص داد. حتی اگر نویز حضور داشته باشد، شانس استخراج اطلاعات درست بیشتر می‌شود.