**به نام خدا**

**سید محمد حسین مظهری 810101520 پروژه 5 درس سیگنال و سیستم**

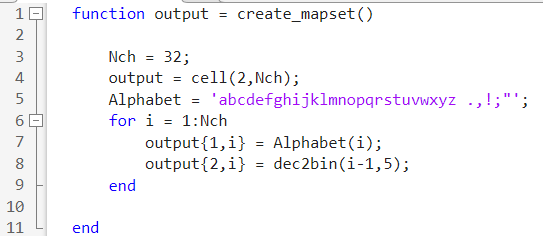
**پارسا دقیق 810101419**

**بخش دوم :**

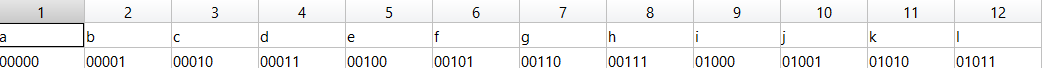
**تمرین 2-1)**

برای ایجاد mapset یک تابع ایجاد میکنیم که در سطر اول کاراکتر ها را قرار دهد و در سطر دوم عدد متناظر باینری آنها را بگذارد .

تابع ایجاد کردن mapset :

:

بخشی از خروجی تابع :



**تمرین2-2)**

در این تابع ابتدا پیامی که قرار است کدگذاری شود را با استفاده از mapset به یک عدد باینری تبدیل میکنیم و آن عدد را در یک آرایه قرار میدهیم که در هر دور حلقه یک عدد به انتهای این لوپ اضافه می شود تا در نهایت کل پیام به شکل باینری تبدیل شود .

حال پس از باینری شدن پیام میتوانیم سیگنال مورد نظر را ایجاد کنیم .

برای ایجاد سیگنال باید ابتدا یکی یکی بیت های سیگنال را به اندازه ی bit rate جدا کنیم. اگر تعداد بیت های پیام بر bit rate بخش پذیر باشد کار به سادگی انجام میشود . در غیر این صورت باید بیت های انتهای پیام را extend کنیم (به انتهای پیام صفر اضافه میکنیم. ) تا ایجاد سیگنال به درستی انجام شود . مثلا اگر در همین تمرین bit rate = 4 باشد و در انتهای پیام دو رقم 11 دااشته باشیم ، بدون extend کردن این 2 رقم معادل 3 در نظر گرفته میشوند در صورتی که با extend کردن به 1100 (12 ) میرسیم .

اگر در لوپ آخر تعداد ارقام باقی مانده بر bit rate بخش پذیر باشند که همان روند قبلی طی میشود . در غیر این صورت ارقام باقی مانده extend می شوند .

در هر دور حلقه بیت های پیام به تعداد bit rate از پیام جدا میشوند . سپس این بیت ها به یک عدد دسیمال تبدیل میشوند تا فرکانس متناظر با آنها محاسبه شود .

نحوه ی محاسبه فرکانس در این تمرین :

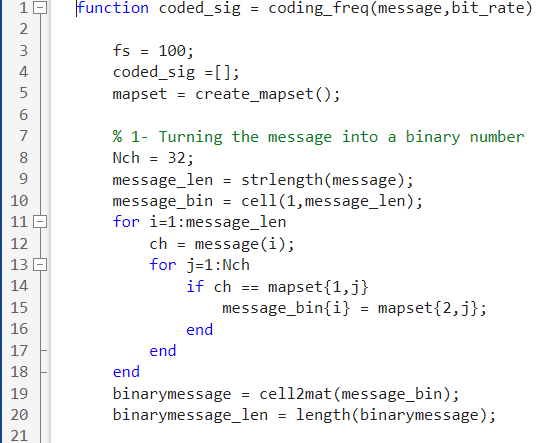
کمترین فرکانس در 5 هرتز و بیشترین فرکانس در 44 هرتز قرار میگیرد . بقیه فرکانس ها میان 5 و44 با فواصل مساوی قرار میگیرند . مقادیر threshold نیز دقیقا وسط هر دو فرکانس در نظر گرفته میشود .

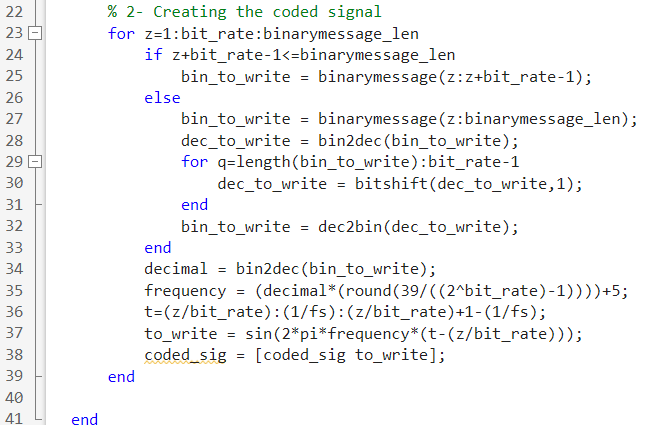
با توجه به روش بیان شده داریم :

Frequency = 5 + decimal number \*

در هر لوپ فرکانس متناظر را محاسبه میکنیم و سیگنالی با آن فرکانس به مدت یک ثانیه تولید میکنیم . در هر لوپ یک ثانیه دیگر از این سیگنال ایجاد شده و به سیگنال قبلی اضافه میشود تا در انتهای لوپ تمام سیگنال مورد نظر درست شده باشد .

اسکریپت این تابع :

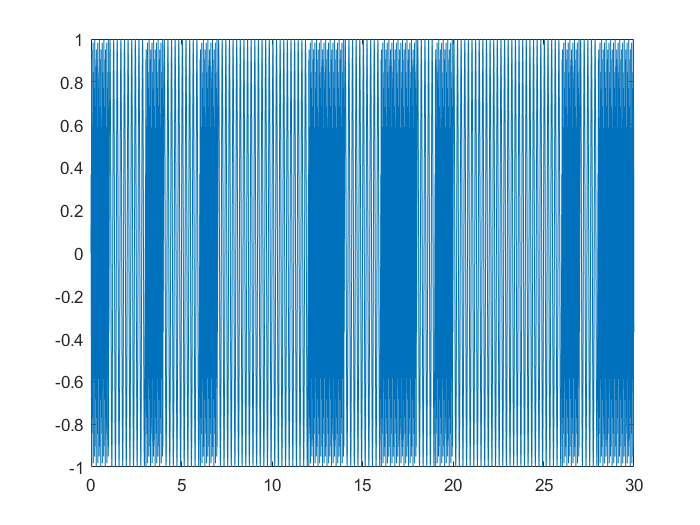




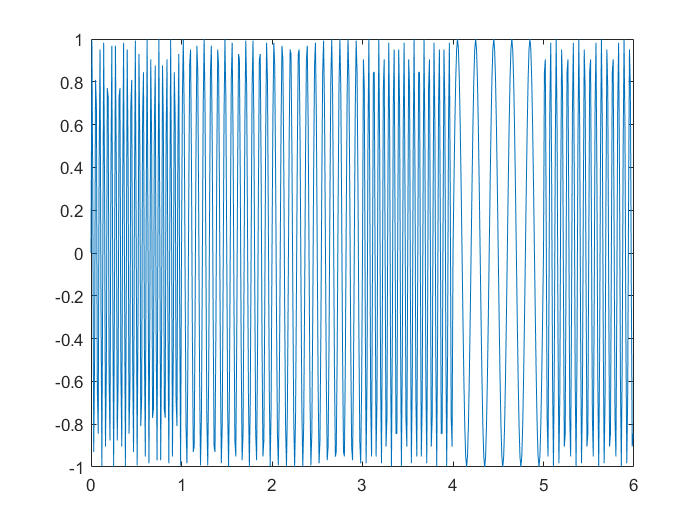
**تمرین 2-3)**

خروجی تابع coding\_freq برای پیام signal :

**Bit rate = 1**



**Bit rate = 5**

****

**تمرین 2-4 )**

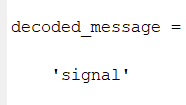
ابتدا باید threshold ها را با روش توضیح داده شده در تمرین قبل به دست آوریم . مقادیر را در یک آرایه نگهداری میکنیم. به سراغ سیگنال ورودی تابع رفته و یک حلقه ایجاد میکنیم که در هر تکرار ، یک ثانیه از سیگنال رو مورد بررسی قرار میدهد . از این یک ثانیه تبدیل فوریه گرفته و و محل قله این تبدیل فوریه را با استفاده از دستور max می یابیم . حال فرکانس متناظر با این محل قله را پیدا میکنیم . این فرکانس همان فرکانسی است که پیام در آن نهفته است . سپس این فرکانس را در یک حلقه با مقادیر threshold مقایسه میکنیم . اولین threshold ای که فرکانس از آن کمتر باشد ، threshold مد نظر است . با توجه به روش انتخاب فرکانس ها و threshold ها میدانیم :

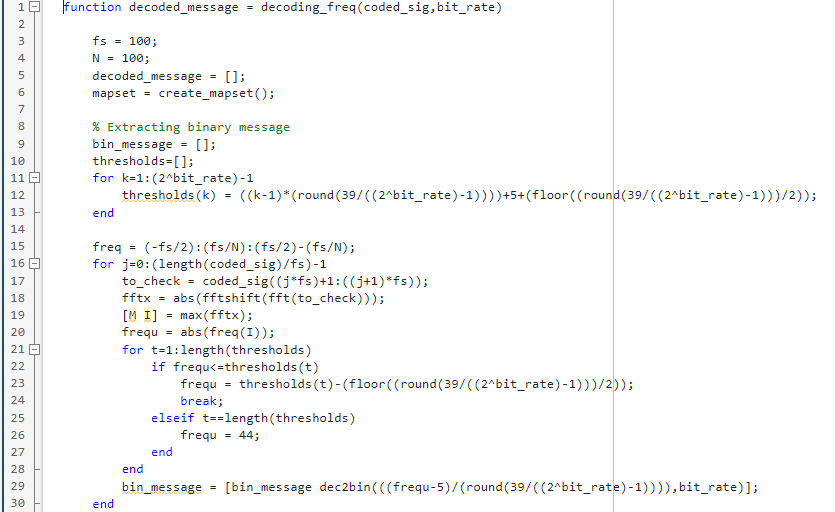
Frequency = threshold –

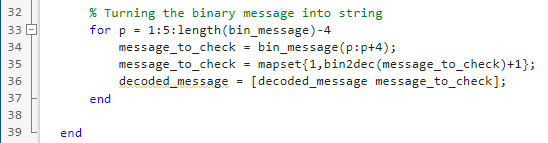
با رابطه بالا فرکانس خود را اصلاح میکنیم . هدف از این اصلاح و استفاده از threshold ، جلوگیری از تغییر پیام توسط نویز های احتمالی روی سیگنال است . حال با استفاده از فرکانس اصلاح شده میتوانیم از رابطه ی ذکر شده در تمرین قبل استفاده کنیم و عدد دسیمال متناظر با این فرکانس را به دست آوریم .

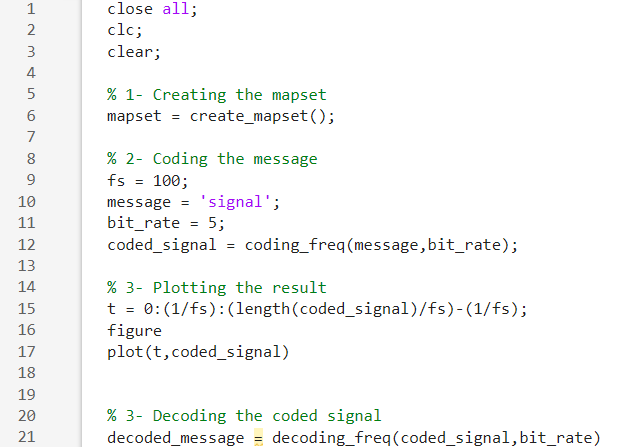
پس از به دست آوردن عدد دسیمال ، این عدد را به یک عدد باینری تبدیل میکنیم و به یک آرایه اضافه میکنیم . حلقه تا رسیدن به پایان سیگنال ادامه پیدا میکند و در پایان معادل باینری پیام نهفته شده به دست آمده است . حال با استفاده از mapset باید پیام باینری را به کاراکتر تبدیل کنیم .

حاصل تست تابع برای bit rate های 1 تا 5 :



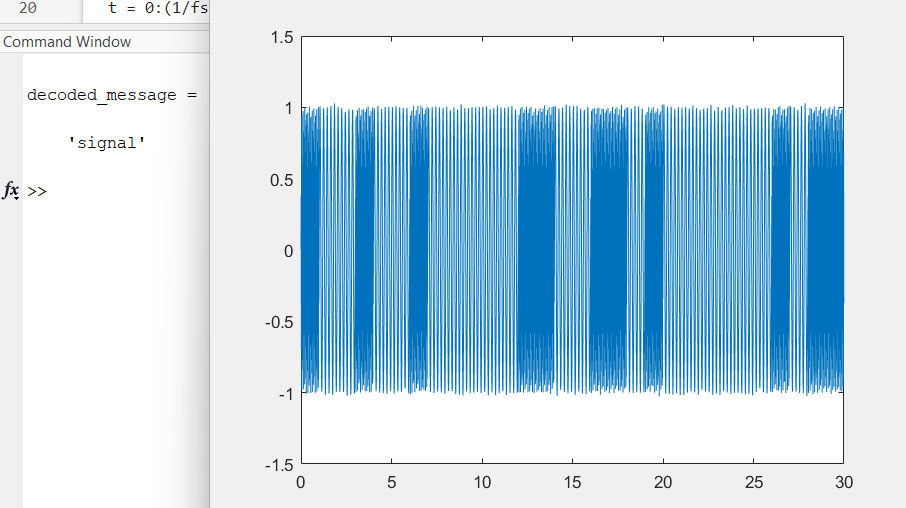
اسکریپت این تابع :

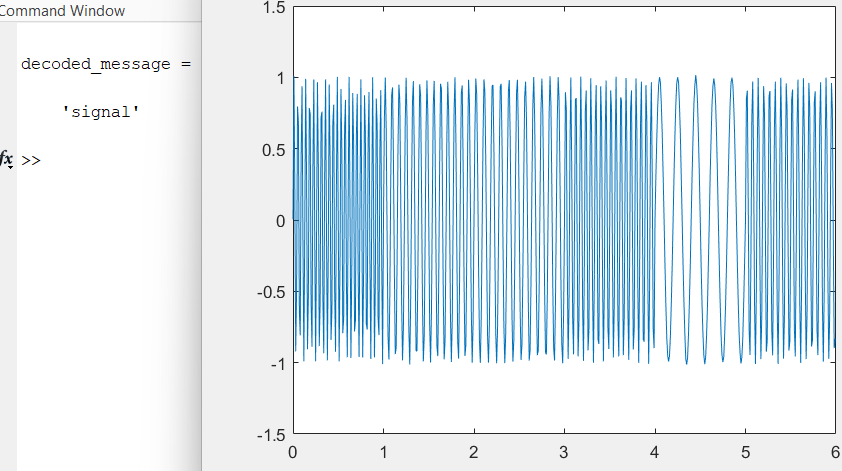


اسکریپت استفاده شده برای تمارین یک تا چهار : 

**تمرین 5-2)**

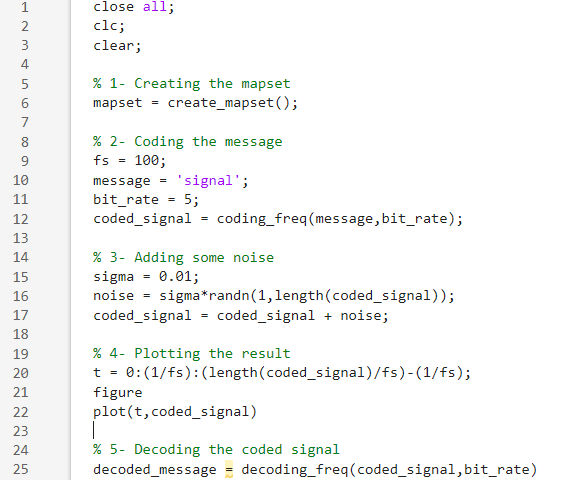
**Bit rate = 1**

****

**Bit rate = 5**

پس از اعمال نویز همچنان پیام signal به خوبی استخراج می شود .

اسکریپت این تمرین :

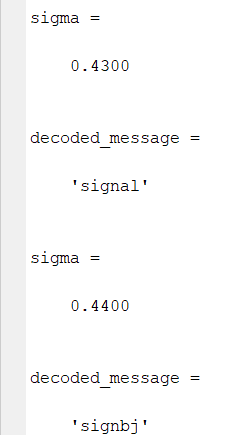


**تمرین 2-6)**

در این تمرین به تدریج قدرت نویز را از 0.01 تا 1 افزایش داده و خروجی ها را مقایسه میکنیم .( δ را تغییر میدهیم)

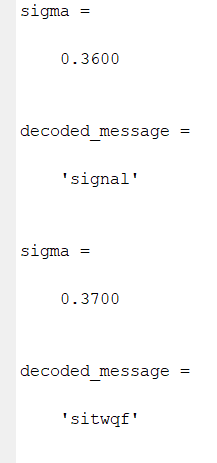
تا δ = 0.43 پیام به درستی استخراج میشود اما با بیشتر شدن قدرت نویز از این مقدار ، دیگر قادر به رمزگشایی پیام نیستیم .

Bit rate = 1

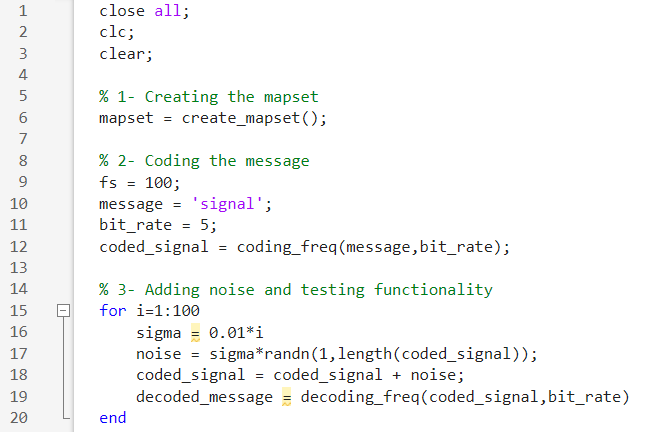


تا δ = 0.36 پیام به درستی استخراج میشود اما با بیشتر شدن قدرت نویز از این مقدار ، دیگر قادر به رمزگشایی پیام نیستیم .

bit rate = 5



همان طور که مشاهده میشود سیگنال با bit rate = 1 نسبت به سیگنال با bit rate = 5 به نویز مقاوم تر است . که مطابق با مطلب بیان شده در مقدمه است .

اسکریپت این تمرین :

**تمرین 2-7 )**

بیشترین واریانس نویز که bit rate = 5 به آن مقاوم بود تقریبا 0.1296 بود . برای bit rate = 1 این عدد تقریبا 0.1849 بود .

**تمرین 2-8)**

انجام شد .

**تمرین 2-9)**

اگر نرخ نمونه برداری را افزایش دهیم ، خود به خود پهنای باند نیز افزایش می یابد . میدانیم : 

با افزايش نرخ نمونه برداري ، رزولوشن فركانسي هيچ تغييري نمي كند ، اما با توجه به رابطه بالا ، پهناي باند

افزايش مي يابد. يعني ما دسترسي به پهناي باند بيشتري داريم اما با توجه به صورت تمرين ، مجاز به استفاده از تمام اين پهناي باند نيستيم . بنابرين اين حالت هيچ فرقي با حالت قبل ندارد و مقاومت نسبت به نويز با حالت قبل يكسان است . شايان ذكر است كه اگر صورت سوال پهناي باند را محدود نمي كرد ، با افزايش نرخ نمونه برداري ، پهناي باند هم خود به خود افزايش مي يافت و مي توانستيم مقاومت نسبت به نويز را افزايش بدهيم. اما اين اتفاق نمي افتد زيرا پهناي باند توسط صورت تمرين محدود شده است و پهناي باند مصرفي از پهناي باند قابل استفاده كمتر است.