

به نام خدا

سیگنال ها و سیستم ها

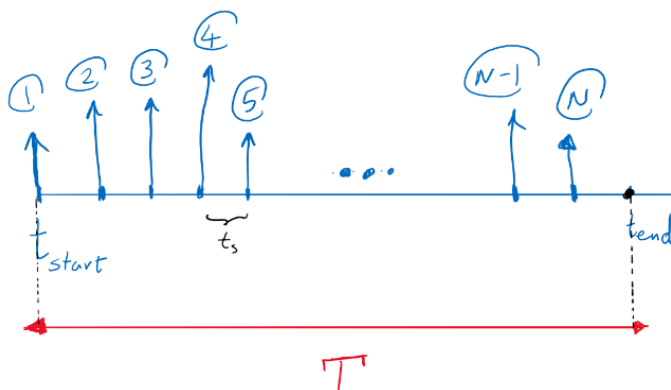
تمرین کامپیوتری سوم

مهلت تحویل: پنجشنبه ۱۵ دی ساعت ۱۷:۰۰

مقدمه ی اول:

از آنجایی که شبیه سازی ها در محیط متلب و با کامپیوتر انجام می شود، همه ی سیگنال هایی که در شبیه سازی ها با آن ها سر و کار داریم سیگنال های گسسته است. لذا همه ی نتایج باید با عناوین مطرح شده در کلاس در حوزه ی گسسته تطابق داشته باشد. اما همان گونه که چندین بار سر کلاس مطرح شد دو حوزه ی گسسته و پیوسته، روابط بسیار نزدیکی دارند و مفاهیم آنها با یکدیگر در تطابق است.

فرض کنید در MATLAB، در حوزه ی زمان، یک سیگنال (بردار) x به طول N سمپل و معادل T ثانیه داریم ($t_s = \frac{1}{f_s} = \frac{T}{N}$) و می خواهیم آن را به حوزه ی فوریه ببریم. (به شکل زیر نگاه کنید). در واقع f_s فرکانس نمونه برداری و t_s فاصله ی زمانی بین دو سمپل را نشان می دهد. این دو پارامتر به ما می گویند که از سیگنال پیوسته اصلی به چه صورت نمونه برداری شده است و سیگنال گسسته ی فعلی تولید شده است. برای رسم دقیق سیگنال در حوزه ی زمان اگر بازه ی زمانی مربوط به سیگنال معلوم باشد $t = t_{start}:t_s:t_{end} - t_s$ به طوری که $t_{start} - t_{end} = T$ ، می توان به راحتی آن را با دستور $\text{plot}(t, x)$ رسم کرد.



برای بردن سیگنال به حوزه ی فوریه از دستور $y = \text{fftshift}(\text{fft}(x))$ استفاده می کنیم. دستور اصلی است و fftshift فقط بازه ی متقارن حول فرکانس صفر را ایجاد می کند. خروجی این دستور یعنی y یک بردار با N سمپل است که هر درایه ی آن یک عدد مختلط است لذا هر درایه یک اندازه و یک فاز دارد. نکته ی مهم در این جا این است که هر یک از این N عدد به دست آمده متعلق به چه فرکانسی می باشد؟

فرکانس ها به صورت $f = \frac{-f_s}{2} : \frac{f_s}{N} : \frac{f_s}{2} - \frac{f_s}{N}$ خواهند بود. بنابراین برای رسم اندازه می توان از دستور $\text{plot}(f, \text{abs}(y))$ و برای رسم فاز می توان از دستور $\text{plot}(f, \text{angle}(y))$ استفاده کرد.

راجع به بازه ی فرکانس های در نظر گرفته شده (هایلایت سبز)، سه نکته ی زیر حائز اهمیت هستند:

نکته ی اول:

در کلاس برای تطابق با کتاب اپنهایم، کلمه ی فرکانس به ω اطلاق شد ولی در این تمرین کامپیوتری به $f = \frac{\omega}{2\pi}$ عنوان فرکانس می دهیم. دلیل این امر این است که از دست عدد π راحت شویم.

نکته ی دوم:

اگر خاطرتان باشد در کلاس گفتیم بیشترین فرکانس (از منظر تغییرات سریع زمانی) در حوزه ی گسسته فرکانس $\frac{1}{2}$ است که در این جا می بینید جای آن عدد $\frac{f_s}{2}$ نشسته است. در واقع در سرتاسر درس سیگنال نرخ نمونه برداری برابر $f_s = 1$ هرتز در نظر گرفته می شود (یک سمپل در هر ثانیه). از این نکته می توان نتیجه گرفت هر چه از سیگنال پیوسته در حوزه ی زمان با نرخ بالاتری نمونه برداری کنیم ($\uparrow f_s$)، در سیگنال گسسته به دست آمده، می توان فرکانس های بالاتر را نیز (در صورت وجود) مشاهده کرد چون بازه ی فرکانسی قابل مشاهده افزایش پیدا می کند. در حالت حدی، اگر نرخ نمونه برداری به سمت بی نهایت برود (یا به عبارت دیگر $t_s \rightarrow 0$) برود، عملاً سیگنال گسسته به دست آمده با سیگنال پیوسته اصلی یکی خواهد بود و هر مولفه ی فرکانسی که در سیگنال اصلی بوده است، در سیگنال گسسته نیز مشاهده می شود. در این قسمت کاملاً باید درک کرده باشید که در sampling که یک سیگنال پیوسته را به یک سیگنال گسسته تبدیل می کند چه چیزی از بین می رود.

نکته ی سوم:

نکته ی آخر راجع به رزولوشن فرکانسی است که ایجاد شده است یعنی $\delta_f = \frac{f_s}{N}$ (هایلات سبز را نگاه کنید). اگر از رابطه ای که با هایلایت زرد رنگ مشخص شده است استفاده کنید می توان دید که رزولوشن فرکانسی برابر است با $\delta_f = \frac{1}{T}$ ، یعنی رزولوشن فرکانسی برابر با عکس طول زمانی سیگنال است و هیچ ربطی هم به نرخ نمونه برداری f_s ندارد. هرچه می خواهید نرخ نمونه برداری را افزایش دهید اما رزولوشن فرکانسی مادامی که طول زمانی سیگنال (T ثانیه) تغییری نکند هیچ تغییری نمی کند. حال ببینیم مفهوم رزولوشن چیست؟ رزولوشن فرکانسی گام های فرکانسی است که می توان در نظر گرفت تا سیگنال گسسته را در فضای فوریه توصیف کرد. این مفهوم را با یک مثال توضیح می دهیم. فرض کنید طول زمانی یک سیگنال $T = 1$ ثانیه است و نرخ نمونه برداری $f_s = 20 \text{ Hz}$ است. بنابراین رزولوشن فرکانسی برابر $\delta_f = 1 \text{ Hz}$ می شود و بازه ی فرکانسی که در حوزه ی فوریه ی

سیگنال نمونه برداری شده می توان مشاهده کرد به صورت $f = -10 : 1 : 9$ هرتز خواهد بود (هایلات سبز). اگر سیگنال اصلی حاوی دو سیگنال تک تن به صورت

$$x_1(t) = \exp(1j * 2\pi * 5 * t) + \exp(1j * 2\pi * 8 * t)$$

باشد، طبیعتاً قله هایی در اندازه ی سیگنال در حوزه ی فوریه، در فرکانس های 5 و 8 هرتز مشاهده خواهید کرد. حال فرض کنید سیگنال اصلی حاوی دو سیگنال تک تن به صورت

$$x_2(t) = \exp(1j * 2\pi * 5 * t) + \exp(1j * 2\pi * 5.1 * t)$$

باشد. در این حالت فقط یک قله در اندازه ی سیگنال در حوزه ی فوریه، در فرکانس 5 هرتز مشاهده خواهید کرد و دیگر توانایی تفکیک این دو سیگنال را در حوزه ی فوریه نخواهید داشت زیرا اختلاف فرکانس دو سیگنال تک تن کمتر از $\delta f = 1 \text{ Hz}$ می باشد. بنابراین رزولوشن فرکانسی قدرت تفکیک پذیری فرکانسی را در حوزه ی فوریه نشان می دهد. حتما این مثال را به عنوان "تمرین شماره ی ۱-۰" در نظر بگیرید و آن را در متلب شبیه سازی کنید و نتایج را گزارش کنید تا بهتر آن را درک کنید. بازه ی زمانی سیگنال را از $t_{start} = 0$ تا $t_{end} = 1$ ثانیه در نظر بگیرید.

حال به سراغ بخش اول تمرین می رویم. قبل از آن توجه داشته باشید در هر سوالی که از شما خواسته شده اندازه تبدیل فوریه را رسم کنید، ماکزیمم خروجی را برابر یک در نظر بگیرید. برای این کار کافی است خروجی دستور $\text{fftshift}(\text{fft}(x))$ یعنی y را به $\max(\text{abs}(y))$ تقسیم کنید. دلیل این است که دستور fft یک ضریب ثابتی به تبدیل فوریه اضافه می کند که برای ما اهمیتی ندارد. همچنین از دستورهایی xlim و ylim به درستی استفاده کنید تا سیگنال ها به خوبی نمایش داده شوند. برای رسم ها اگر تمایل داشتید می توانید از دستور db استفاده کنید تا تغییرات کوچک بهتر نمایش داده شوند.

تمرین ۱-۱)

سیگنال $x_1(t) = \cos(10\pi t)$ را در نظر بگیرید.

الف) این سیگنال را در حوزه ی زمان با در نظر گرفتن بازه ی زمانی $t_{start} = -1$ تا $t_{end} = 1$ ثانیه و فرکانس نمونه برداری $f_s = 50 \text{ Hz}$ رسم کنید.

ب) اندازه ی تبدیل فوریه ی این سیگنال را با همان مفروضات قسمت الف رسم کنید.

ج) به صورت تئوری تبدیل فوریه ی پیوسته این سیگنال را حساب کنید. همان طور که مشاهده می کنید شکل کلی نتیجه به دست آمده با نتیجه ی قسمت ب تطابق دارد.

تمرین ۱-۲)

سیگنال $x_2(t) = \Pi(t)$ را در نظر بگیرید.

الف) این سیگنال را در حوزه ی زمان با در نظر گرفتن بازه ی زمانی $t_{start} = -1$ تا $t_{end} = 1$ ثانیه و فرکانس نمونه برداری $f_s = 50 \text{ Hz}$ رسم کنید.

ب) اندازه ی تبدیل فوریه ی این سیگنال را با همان مفروضات قسمت الف رسم کنید.

ج) به صورت تئوری تبدیل فوریه ی پیوسته این سیگنال را حساب کنید. همان طور که مشاهده می کنید شکل کلی نتیجه به دست آمده با نتیجه ی قسمت ب تطابق دارد.

تمرین ۱-۳)

سیگنال $x_3(t) = x_1(t)x_2(t) = \cos(10\pi t)\Pi(t)$ را در نظر بگیرید.

الف) این سیگنال را در حوزه ی زمان با در نظر گرفتن بازه ی زمانی $t_{start} = -1$ تا $t_{end} = 1$ ثانیه و فرکانس نمونه برداری $f_s = 50 \text{ Hz}$ رسم کنید.

ب) اندازه ی تبدیل فوریه ی این سیگنال را با همان مفروضات قسمت الف رسم کنید.

ج) به صورت تئوری تبدیل فوریه ی پیوسته این سیگنال را حساب کنید. همان طور که مشاهده می کنید شکل کلی نتیجه به دست آمده با نتیجه ی قسمت ب تطابق دارد.

* توجه داشته باشید هر چه طول پالس Π افزایش پیدا کند، سیگنال $x_3(t)$ به سیگنال $x_1(t)$ شبیه تر خواهد شد و بنابراین دو ضربه در فرکانس های 5 و -5 هرتز بهتر نشان داده خواهند شد.

تمرین ۴-۱

سیگنال $x_4(t) = \cos(30\pi t + \frac{\pi}{4})$ را در نظر بگیرید.

الف) اندازه ی تبدیل فوری ی این سیگنال را با در نظر گرفتن بازه ی زمانی $t_{start} = 0$ تا $t_{end} = 1$ ثانیه و فرکانس نمونه برداری $f_s = 100 \text{ Hz}$ رسم کنید.

ب) فاز تبدیل فوری ی این سیگنال را با همان مفروضات قسمت الف رسم کنید. برای این کار ابتدا فاز فرکانس هایی که اندازه ی تبدیل فوری در آنها ناچیز است را برابر صفر کنید و سپس فاز را به صورت مضربی از π نمایش دهید. برای این کار از دستور زیر استفاده کنید.

```
tol = 1e-6;  
y(abs(y) < tol) = 0;
```

```
theta = angle(y);
```

```
plot(f,theta/pi)  
xlabel 'Frequency (Hz)'  
ylabel 'Phase / \pi'
```

ج) به صورت تئوری تبدیل فوری ی پیوسته این سیگنال را حساب کنید. آیا نتایج به دست آمده در قسمت الف و ب با مقدار تئوری مطابقت دارد؟

تمرین ۵-۱

سیگنال $x_5(t) = \sum_{k=-9}^9 \Pi(t - 2k)$ را در نظر بگیرید.

الف) این سیگنال را در حوزه ی زمان با در نظر گرفتن بازه ی زمانی $t_{start} = -19$ تا $t_{end} = 19$ ثانیه و فرکانس نمونه برداری $f_s = 50 \text{ Hz}$ رسم کنید.

ب) اندازه ی تبدیل فوری ی این سیگنال را با همان مفروضات قسمت الف رسم کنید.

ج) سیگنال $x_5(t)$ به نوعی متناوب شده ی سیگنال $x_2(t) = \Pi(t)$ است. چرا یک تعدادی ضربه در خروجی مشاهده می کنید؟ فواصل ضربه هایی که مشاهده می کنید چه قدر است؟ همان طور که مشاهده می کنید نتایج با آنچه در درس تحت موضوع "تبدیل فوری سیگنال های متناوب" آموختیم تطابق دارد.

مقدمه ی دوم:

فرکانس یعنی تغییرات در حوزه زمان و تبدیل فوریه قرار است ما را از حوزه زمان به حوزه فرکانس ببرد. پس هرچه تغییرات در حوزه زمان شدید تر باشد یعنی تابع ما دارای فرکانس های بالا تری است و وقتی از آن تبدیل فوریه بگیریم فرکانس های بالاتری در تبدیل فوریه آن ظاهر میشود.

شدید ترین تغییرات در حوزه زمان چیست؟ مسلماً ناپیوستگی شدید ترین تغییرات در حوزه زمان است پس تمام توابعی که ناپیوستگی دارند تبدیل فوریه آن ها باید بزرگترین فرکانس ها را نیز شامل شود. این یعنی باید فرکانس های بی نهایت و منفی بی نهایت را نیز داشته باشد. هم چنین برای بیان کردن این تغییرات تعداد محدودی فرکانس پاسخ گو نخواهد بود پس تبدیل فوریه آن فقط روی بخش محدودی از محور فرکانس مقدار ندارد و باید از $-\infty$ تا $+\infty$ گسترده باشد.

کندترین تغییرات در زمان چیست؟ طبیعتاً تابع ثابت هیچ تغییری در حوزه ی زمان ندارد که تبدیل فوریه ی آن فقط یک ضربه در فرکانس صفر می شود. بنابراین اگر یک تابع در حوزه زمان تغییرات زیادی نداشته باشد و ناپیوستگی هم نداشته باشد دارای فرکانس های بالایی نیست و با فرکانس های پایین تغییرات آن قابل بیان است. حال با توجه به مقدمه ی دوم به تمارین بعدی پاسخ دهید.

تمرین ۱-۲

سیگنال $x_6(t) = \delta(t)$ را در نظر بگیرید.

الف) این سیگنال را در حوزه ی زمان با در نظر گرفتن بازه ی زمانی $t_{start} = -1$ تا $t_{end} = 1$ ثانیه و فرکانس نمونه برداری $f_s = 50 \text{ Hz}$ رسم کنید.

ب) اندازه ی تبدیل فوریه ی این سیگنال را با همان مفروضات قسمت الف رسم کنید.

ج) با توجه به مقدمه ی دوم، دلیل آنچه مشاهده می کنید را شرح دهید. همچنین به صورت تئوری تبدیل فوریه ی پیوسته این سیگنال را حساب کنید.

تمرین ۲-۲

سیگنال $x_7(t) = 1$ را در نظر بگیرید.

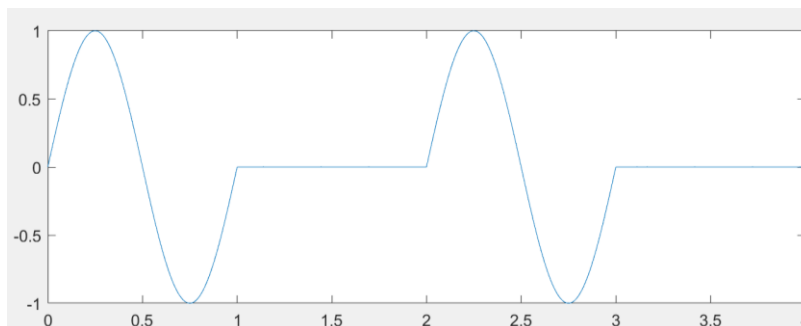
الف) این سیگنال را در حوزه ی زمان با در نظر گرفتن بازه ی زمانی $t_{start} = -1$ تا $t_{end} = 1$ ثانیه و فرکانس نمونه برداری $f_s = 50 \text{ Hz}$ رسم کنید.

ب) اندازه ی تبدیل فوریه ی این سیگنال را با همان مفروضات قسمت الف رسم کنید.

ج) با توجه به مقدمه ی دوم، دلیل آنچه مشاهده می کنید را شرح دهید. همچنین به صورت تئوری تبدیل فوریه ی پیوسته این سیگنال را حساب کنید.

مقدمه ی سوم:

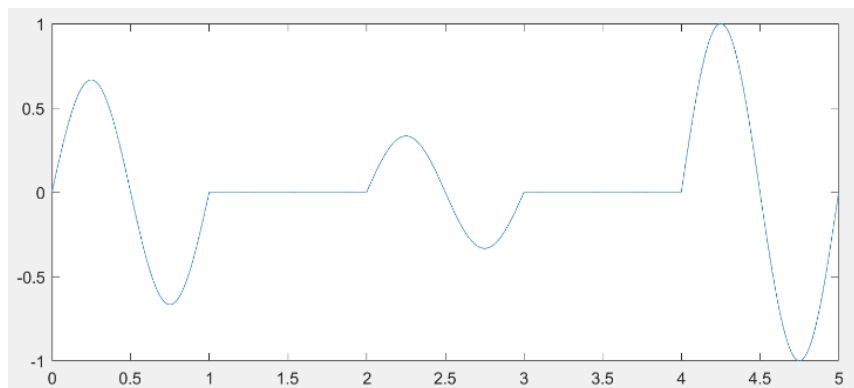
فرض کنید می خواهیم پیامی را به صورت بی سیم از طریق امواج برای شخصی ارسال کنیم. ابتدا پیام را در کامپیوتر به صورت یک رشته بیت باینری در آورده و سپس بیت ها را کدگذاری می کنیم. مثلاً به جای هر بیت یک سیگنال خاص را ارسال می کنیم. فرض کنید به جای بیت صفر، سیگنال $x_0(t) = 0$ را به مدت یک ثانیه و به جای بیت یک، سیگنال $x_1(t) = \sin(2\pi t)$ را به مدت یک ثانیه ارسال کنیم. مثلاً رشته بیت 1010 به صورت زیر ارسال می شود:



فرض کنید قدرت فرستنده به گونه ای است که حداکثر دامنه ای که به سیگنال می تواند بدهد، دامنه ی یک است. همچنین فرکانس نمونه برداری سیگنال را $f_s = 100 \text{ Hz}$ در نظر بگیرید که باعث می شود هر سیگنال یک ثانیه ای شامل ۱۰۰ سمپل باشد. این فرض را تا آخر تمرین کامپیوتری در نظر داشته باشید.

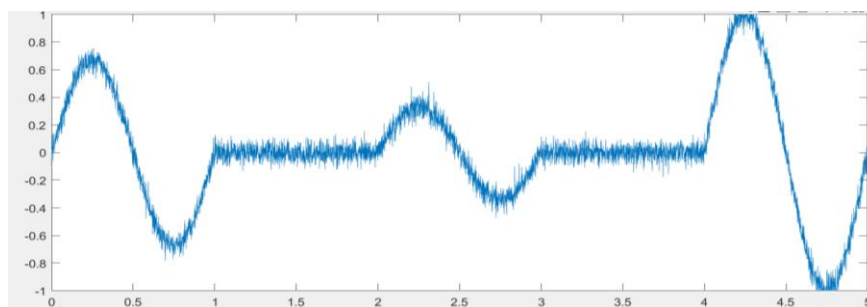
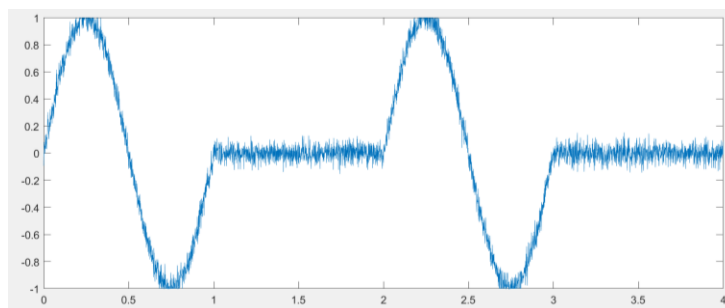
در گیرنده این سیگنال دریافت می شود. با فرض این که فرستنده و گیرنده از نظر زمانی کاملاً سنکرون باشند، برای رمزگشایی از پیام در گیرنده کافی است در هر یک ثانیه، correlation سیگنال دریافتی را با $2 \sin(2\pi t)$ حساب کنیم. هر جایی correlation مقدار داشته باشد (مقدار $\int 2 \sin^2(2\pi t) dt = 1$) می فهمیم بیت ارسالی برابر یک بوده و هر جا مقدار correlation صفر شود می فهمیم بیت ارسالی برابر صفر بوده است. توجه داشته باشید عدد 2 در دامنه ی $2 \sin(2\pi t)$ هیچ اهمیتی ندارد و فقط برای رُند شدن مقدار correlation است.

در سناریوی بالا سرعت ارسال پیام (bit rate) $1 \frac{\text{bit}}{\text{sec}}$ است. حال فرض کنید بخواهیم bit rate را افزایش دهیم. برای این کار باید کدگذاری را روی بیت های بیشتری انجام دهیم. مثلاً به جای هر دو بیت، یک سیگنال خاص را ارسال کنیم. برای ارسال 00 سیگنال $x_0(t) = 0$ را به مدت یک ثانیه، برای ارسال 01 سیگنال $x_1(t) = \frac{1}{3} \sin(2\pi t)$ را به مدت یک ثانیه، برای ارسال 10 سیگنال $x_2(t) = \frac{2}{3} \sin(2\pi t)$ را به مدت یک ثانیه و برای ارسال 11 سیگنال $x_3(t) = \sin(2\pi t)$ را به مدت یک ثانیه ارسال می کنیم. مثلاً رشته بیت 100010011 به صورت زیر ارسال می شود.



در گیرنده این سیگنال دریافت می شود. مجدداً برای رمزگشایی از پیام در گیرنده کافی است در هر ثانیه، correlation سیگنال دریافتی را با $2\sin(2\pi t)$ حساب کنیم و بسته به خروجی correlation که می تواند مقادیر صفر، $\frac{1}{3}$ ، $\frac{2}{3}$ و یک بگیرد راجع به بیت های ارسالی تصمیم گیری کنیم. در این سناریو سرعت ارسال پیام (bit rate) $2 \frac{\text{bit}}{\text{sec}}$ است.

این روند را می توان به همین ترتیب ادامه داد و سرعت ارسال پیام را افزایش داد. یعنی هر سه بیت را کدگذاری کرد تا سرعت ارسال سه برابر شود و ... اما این اضافه شدن سرعت در عمل یک مشکل اساسی دارد که به ما اجازه نمی دهد سرعت را از حدی بیشتر کنیم. مشکل نویز است! وقتی فرستنده سیگنالی را به گیرنده ارسال می کند، در گیرنده علاوه بر سیگنال اصلی، نویز نیز به سیگنال اصلی اضافه می شود. مثلاً در سناریوی اول و دوم سیگنال های زیر به گیرنده می رسد:



حال اگر correlation سیگنال دریافتی را با $2\sin(2\pi t)$ حساب کنیم، خروجی ها دقیقاً اعدادی که انتظار داشتیم نمی شوند و بنابراین ممکن است به اشتباه رمز گشایی کنیم. هر چه سرعت ارسال بیت (bit rate) بیشتر باشد این خطا بیشتر خواهد بود. چرا؟ قدری فکر کنید و بعد پاسخ این سوال را که در ادامه آمده است بخوانید.

به عنوان مثال سناریوی اول که سرعت ارسال $1 \frac{bit}{sec}$ بود را در نظر بگیرید. انتظار داشتیم مقادیر خروجی correlation عدد صفر یا عدد یک باشد ولی در حضور نویز این اعداد کمی تغییر می کنند. پس به نظر می رسد باید عدد 0.5 را به عنوان threshold (آستانه) تعریف کنیم. اگر correlation بیش از 0.5 شد اعلام کنیم بیت مورد نظر یک بوده و اگر کمتر از 0.5 شد اعلام کنیم بیت مورد نظر صفر بوده است.

در سناریوی دوم که سرعت ارسال $2 \frac{bit}{sec}$ بود، انتظار داشتیم مقادیر خروجی correlation عدد صفر، $\frac{1}{3}$ ، $\frac{2}{3}$ و یک شود، اما به خاطر وجود نویز این اعداد ممکن است هر مقداری بگیرند. بنابراین باید سه آستانه ی $\frac{1}{6}$ ، $\frac{3}{6}$ و $\frac{5}{6}$ را تعریف کنیم و با توجه به آنها راجع به بیت ها تصمیم گیری کنیم. چون فاصله ی این اعداد از هم کمتر شده است بنابراین احتمال خطا در تصمیم گیری بیشتر می شود.

پس در محیط عملیاتی، در روش پیشنهاد شده، یک trade off بین افزایش سرعت انتقال پیام و مقاوم بودن نسبت به نویز وجود دارد. حال می خواهیم مفاهیم بیان شده در این بخش را شبیه سازی کنیم.

هدف این تمرین ارسال پیام به زبان انگلیسی از فرستنده به گیرنده است. هر پیام فقط شامل حروف کوچک انگلیسی، فاصله، نقطه، ویرگول، علامت تعجب، سمی کالن (;) و کوتیشن (") است. بنابراین در مجموع ۳۲ کاراکتر داریم. به هر کاراکتر ۵ بیت مرتبط می کنیم.

تمرین ۱-۳) یک سلول به اسم Mapset با ابعاد 2×32 درست کنید. در سطر اول خود کارکترها را قرار دهید و در سطر دوم ۵ بیتی که به آنها مرتبط کردید را قرار دهید. به عنوان مثال شکل زیر قسمت ابتدایی Mapset را نشان می دهد. (دستور dec2bin برای باینری کردن اعداد کمک کننده خواهد بود).

	1	2	3
1	'a'	'b'	'c'
2	'00000'	'00001'	'00010'

تمرین ۲-۳) تابعی (*function*) به نام *coding_amp* بنویسید که ورودی های آن (۱) پیام مورد نظر برای ارسال و (۲) سرعت ارسال اطلاعات باشد و خروجی آن پیام کدگذاری شده باشد.

راهنمایی: مثلاً فرض کنید می خواهیم پیام *bc* را با سرعت $1 \frac{bit}{sec}$ ارسال کنیم. ابتدا حروف پیام را به صورت باینری در آورید که (با توجه به شکل بالا) می شود: *0000100010*، سپس رشته باینری به دست آمده را مطابق آنچه در مقدمه توضیح داده شد کدگذاری کنید. توجه داشته باشید پیام ممکن است هر کلمه یا جمله ای با طول دلخواه شامل ۳۲ کاراکتر ذکر شده باشد. در بخش های بعدی بیشترین سرعت ارسال اطلاعات را ۳ بیت بر ثانیه در نظر خواهیم گرفت گرچه کد شما می تواند هر عدد طبیعی دلخواهی را پشتیبانی کند! (دستورات *extract* و *strlength* برای جداسازی کاراکترهای پیام کمک کننده خواهند بود).

تمرین ۳-۳) خروجی تابع *coding_amp* را برای پیام (کلمه‌ی) *signal* با سرعت ارسال اطلاعات یک، دو و سه بیت بر ثانیه به صورت جداگانه رسم کنید.

تمرین ۴-۳) تابعی به نام *decoding_amp* بنویسید که ورودی های آن (۱) پیام کدگذاری شده (سیگنال زمانی تولید شده در قسمت قبل) و (۲) سرعت ارسال اطلاعات باشد و خروجی آن پیام رمز گشایی شده باشد. تابعی که نوشتید را روی همان پیام *signal* که در قسمت ۳-۳ با سرعت ارسال های مختلف کد کردید تست کنید تا مطمئن شوید کدتان درست کار می کند.

توجه: اگر می خواهید نتایج *correlation* کاملاً شبیه آنچه که در مقدمه بیان شد باشد، عدد 0.01 را پشت *correlation* ضرب کنید. چون به صورت گسسته *correlation* را حساب می کنید و تعداد نمونه ها در هر بازه‌ی *correlation* گیری ۱۰۰ نمونه است این اتفاق رخ می دهد.

تمرین ۵-۳) در این قسمت می خواهیم به سیگنال دریافتی در گیرنده نویز اضافه کنیم تا شبیه سازی مشابه شرایط واقعی شود. برای تولید نویز از دستور *randn* استفاده کنید. این دستور نویز گوسی با میانگین صفر و واریانس یک تولید می کند. برای زیاد یا کم کردن قدرت نویز کافی است در پشت این دستور عدد دلخواه σ (بدون توان ۲) را ضرب کنید. این عدد کاری می کند تا واریانس نویز σ^2 شود. به پیام *signal* بعد از این که کدگذاری شد نویز گوسی با واریانس 0.0001 و میانگین صفر اضافه کنید و بعد آن را *decode* کنید. آیا بازهم پیام *signal* استخراج شد؟ نتایج *decoding* را برای هر سه *bit rate* یک، دو و سه به صورت جداگانه گزارش کنید.

تمرین ۳-۶) قدرت نویز را کم و طی چندین مرحله افزایش دهید و هر بار قسمت ۳-۵ را تکرار کنید. مشاهدات خود را گزارش کنید. کدام یک از سه $bit\ rate$ به نویز مقاوم تر بودند؟ آیا نتایج با آنچه در مقدمه بیان شد همخوانی داشتند؟

تمرین ۳-۷) طی شبیه سازی هایی که در قسمت ۳-۶ انجام دادید، بیشترین واریانس نویز که $bit\ rate$ سه به آن مقاوم بود به صورت تقریبی چند بود؟ برای $bit\ rate$ دو و یک چه طور؟

تمرین ۳-۸) به نظر شما با چه راهکاری می توان $bit\ rate$ را نسبت به نویز مقاوم تر کرد؟ پاسخ خود را در هایلایت قرمز جستجو کنید. حال در ادامه پاسخ را مطالعه کنید.

اگر قدرت فرستنده ی ما بیشتر می بود می توانستیم دامنه ی سیگنال بیشتری داشته باشیم و بنابراین فاصله ی $threshold$ هایی که برای تصمیم گیری در نظر گرفته بودیم بیشتر می شد و در نتیجه به نویز حساسیت کمتری ایجاد می شد. مثلاً برای ارسال اطلاعات با سرعت $2 \frac{bit}{sec}$ ، اگر قدرت فرستنده به گونه ای می بود که بیشینه ی دامنه ی سیگنال سه می شد، شرایط بهتر می شد. به عبارت دیگر، برای ارسال 00 سیگنال $x_0(t) = 0$ را به مدت یک ثانیه، برای ارسال 01 سیگنال $x_1(t) = \sin(2\pi t)$ را به مدت یک ثانیه، برای ارسال 10 سیگنال $x_2(t) = 2 \sin(2\pi t)$ را به مدت یک ثانیه و برای ارسال 11 سیگنال $x_3(t) = 3 \sin(2\pi t)$ را به مدت یک ثانیه ارسال می کردیم.

پس در کل می توان نتیجه گرفت در روش کدگذاری دامنه اگر می خواهیم سرعت ارسال اطلاعات را افزایش دهیم باید $power$ بیشتری نیز مصرف کنیم تا در برابر نویز مقاوم باشیم.

مقدمه ی چهارم:

در بخش قبل کدگذاری روی دامنه ی سیگنال اعمال شد، حال می خواهیم کدگذاری روی فرکانس داشته باشیم.

به عنوان مثال برای سرعت ارسال اطلاعات $1 \frac{\text{bit}}{\text{sec}}$ ، به جای بیت صفر سیگنال $x_0(t) = \sin(2\pi f_0 t)$ را به مدت یک ثانیه و به جای بیت یک، سیگنال $x_1(t) = \sin(2\pi f_1 t)$ را به مدت یک ثانیه ارسال می کنیم. در گیرنده، برای رمز گشایی از سیگنال دریافتی، کافی است در هر یک ثانیه از سیگنال دریافتی تبدیل فوریه بگیریم (fft) و فرکانس اصلی سیگنال را پیدا کنیم و متناظر با آن بیت ارسالی را استخراج کنیم.

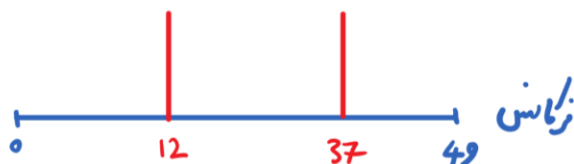
به عنوان یک مثال دیگر اگر بخواهیم سرعت ارسال اطلاعات $2 \frac{\text{bit}}{\text{sec}}$ باشد، برای ارسال 00 سیگنال $x_0(t) = \sin(2\pi f_0 t)$ را به مدت یک ثانیه، برای ارسال 01 سیگنال $x_1(t) = \sin(2\pi f_1 t)$ را به مدت یک ثانیه، برای ارسال 10 سیگنال $x_2(t) = \sin(2\pi f_2 t)$ را به مدت یک ثانیه و برای ارسال 11 سیگنال $x_3(t) = \sin(2\pi f_3 t)$ را به مدت یک ثانیه ارسال می کنیم. برای رمز گشایی در گیرنده، مشابه حالت قبلی، کافی است در هر یک ثانیه، از سیگنال دریافتی تبدیل فوریه بگیریم (fft) و فرکانس اصلی سیگنال را پیدا کنیم و متناظر با آن دو بیت ارسالی را استخراج کنیم (توجه داشته باشید، fft گرفتن نقش همان $correlation$ گرفتن با سیگنال های سینوسی با فرکانس های مختلف را ایفا می کند. همچنین استفاده از $fftshift$ را فراموش نکنید!)

حال سوال اینجاست که مقادیر فرکانس ها را چگونه انتخاب کنیم؟

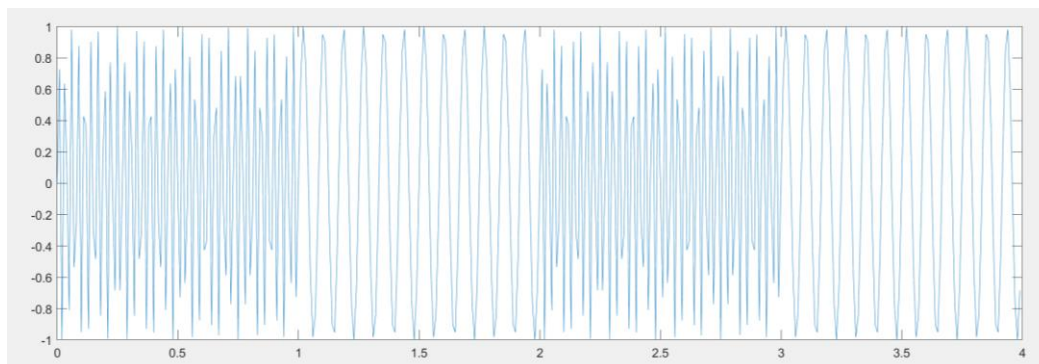
طبق آنچه در مقدمه ی یک گفته شد، بازه ی فرکانسی یک سیگنال به طول $N=100$ سمپل و یک ثانیه به صورت 50:1:49- هرتز خواهد بود. می دانیم سیگنال های حقیقی هم فرکانس منفی و هم فرکانس مثبت دارند، پس فقط به بخش فرکانس های مثبت توجه می کنیم.



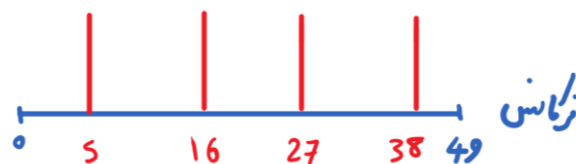
برای سرعت ارسال اطلاعات $1 \frac{\text{bit}}{\text{sec}}$ دو فرکانس را به صورت زیر انتخاب می کنیم (فرکانس های ۱۲ و ۳۷ هرتز):



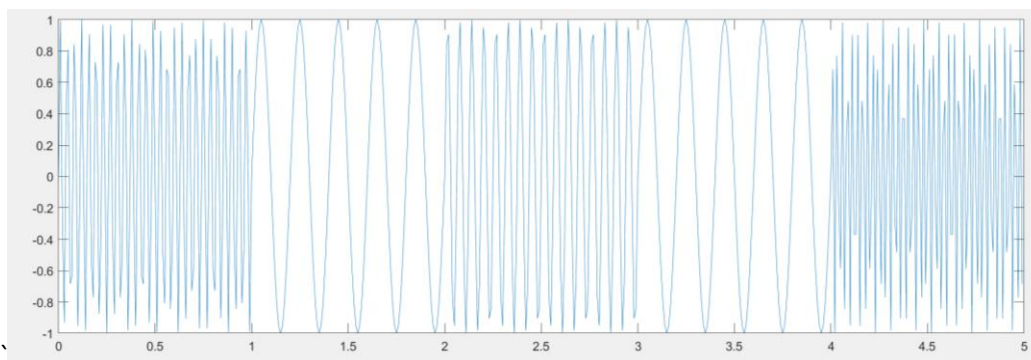
مثلاً رشته بیت 1010 به صورت زیر ارسال می شود:



برای سرعت ارسال اطلاعات $2 \frac{bit}{sec}$ ، چهار فرکانس را به صورت زیر انتخاب می کنیم (فرکانس های ۵، ۱۶، ۲۷ و ۳۸ هرتز):



مثلاً رشته بیت 1000010011 به صورت زیر ارسال می شود.



توجه داشته باشید لزومی ندارد انتخاب فرکانس ها به گونه ای که در بالا ذکر شده است باشد. در این انتخاب ها سعی شده است فرکانس ها بیشترین فاصله ی ممکن را از هم داشته باشند تا وقتی داده نویزی می شود در تصمیم گیری دچار اشتباه کمتری شویم. توجه داشته باشید وقتی داده نویزی شود و در یک بازه ی یک ثانیه ای از داده تبدیل فوریه بگیریم، دیگر لزوماً فقط یک قله ی مرتفع نخواهیم دید و ممکن است قله جابجا شود و فرکانسی که قله در آن رخ داده است را اشتباه تخمین بزنیم. به همین علت هرچه فاصله ی فرکانس ها از هم بیشتر شود تصمیم گیری دچار خطای کمتری می شود. شایان ذکر است در داده های نویزریال مشابه بخش قبل، آستانه هایی تعریف می کنیم و بر اساس آن آستانه ها تصمیم گیری می کنیم. این آستانه ها در کدگذاری فرکانس، در

حوزه ی فرکانس تعریف می شوند. به عنوان مثال با توجه به فرکانس های در نظر گرفته شده برای سرعت ارسال اطلاعات $1 \frac{bit}{sec}$ ، آستانه ای که باید تعریف کنیم فرکانس ۲۴.۵ هرتز است. در واقع از سیگنال کدگذاری شده به طول یک ثانیه که می تواند نویزی هم باشد تبدیل فوریه می گیریم و سپس فرکانسی که قله ی تبدیل فوریه در آن شکل گرفته است را استخراج می کنیم. اگر این فرکانس از ۲۴.۵ هرتز کمتر بود می فهمیم بیت صفر ارسال شده است و اگر این فرکانس از ۲۴.۵ هرتز بیشتر بود می فهمیم بیت یک ارسال شده است. حال به سراغ شبیه سازی این موارد می رویم.

تمرین ۴) تمرینات های بخش سوم را این بار با فرض کدگذاری در فرکانس تکرار می کنیم.

هدف این تمرین ارسال پیام به زبان انگلیسی از فرستنده به گیرنده است. هر پیام فقط شامل حروف کوچک انگلیسی، فاصله، نقطه، ویرگول، علامت تعجب، سمی کالن (;) و کوتیشن (") است. بنابراین در مجموع ۳۲ کاراکتر داریم. به هر کاراکتر ۵ بیت مرتبط می کنیم.

تمرین ۴-۱) یک سلول به اسم Mapset با ابعاد 2×32 درست کنید. در سطر اول خود کارکترها را قرار دهید و در سطر دوم ۵ بیتی که به آنها مرتبط کردید را قرار دهید (پاسخ این قسمت کاملاً مشابه بخش قبل است).

تمرین ۴-۲) تابعی به نام `coding_freq` بنویسید که ورودی های آن (۱) پیام مورد نظر برای ارسال و (۲) سرعت ارسال اطلاعات باشد و خروجی آن پیام کدگذاری شده باشد. بیشترین سرعت ارسال اطلاعات را ۳ بیت برثانیه در نظر خواهیم گرفت گرچه کد شما می تواند اعداد طبیعی بزرگتری را نیز پشتیبانی کند!

تمرین ۴-۳) خروجی تابع `coding_freq` را برای پیام (کلمه ی) `signal` با سرعت ارسال اطلاعات یک، دو و سه بیت بر ثانیه به صورت جداگانه رسم کنید.

تمرین ۴-۴) تابعی به نام `decoding_freq` بنویسید که ورودی های آن (۱) پیام کدگذاری شده (سیگنال زمانی تولید شده در قسمت قبل) و (۲) سرعت ارسال اطلاعات باشد و خروجی آن پیام رمز گشایی شده باشد. تابعی که نوشتید را روی همان پیام `signal` که در قسمت ۳-۴ با سرعت ارسال های مختلف کُد کردید تست کنید تا مطمئن شوید کدتان درست کار می کند.

تمرین ۴-۵) در این قسمت می خواهیم به سیگنال دریافتی در گیرنده نویز اضافه کنیم تا شبیه سازی مشابه شرایط واقعی شود. به پیام `signal` بعد از این که کدگذاری شد نویز گوسی با واریانس 0.0001 و میانگین صفر اضافه کنید و بعد آن را `decode` کنید. آیا بازهم پیام `signal` استخراج شد؟ نتایج `decoding` را برای هر سه `rate` یک، دو و سه به صورت جداگانه گزارش کنید.

تمرین ۴-۶) قدرت نویز را کم و طی چندین مرحله افزایش دهید و هر بار قسمت ۴-۵ را تکرار کنید. مشاهدات خود را گزارش کنید. کدام یک از سه $bit\ rate$ به نویز مقاوم تر بودند؟ آیا نتایج با آنچه در مقدمه بیان شد همخوانی داشتند؟

تمرین ۴-۷) طی شبیه سازی هایی که در قسمت ۴-۶ انجام دادید، بیشترین واریانس نویز که $bit\ rate$ سه به آن مقاوم بود به صورت تقریبی چند بود؟ برای $bit\ rate$ دو و یک چه طور؟

تمرین ۴-۸) به نظر شما با چه راهکاری می توان $bit\ rate$ را نسبت به نویز مقاوم تر کرد؟ قدری درنگ کنید و سپس پاسخ را مطالعه کنید. پاسخ این است که هر چه فاصله ی فرکانس های انتخابی بیشتر باشند کدگذاری نسبت به نویز مقاوم تر می شود. بنابراین هر چه پهنای باند بیشتری مصرف کنیم می توانیم با سرعت بیشتری اطلاعات را ارسال کنیم و در عین حال نسبت به نویز مقاوم باشیم.

آنچه که بارها راجع به افزایش پهنای باند شنیدید که منجر به سرعت بیشتری در اینترنت می شود همین نکته است.

نکات کلی:

- در صورت وجود هرگونه پرسش و ابهام به استاد ایمیل بزنید.
- فایل نهایی شما باید به صورت یک فایل زیپ شامل گزارشکار به فرمت PDF و کد های متلب باشد.