به نام خدا

سیگنال ها و سیستم ها

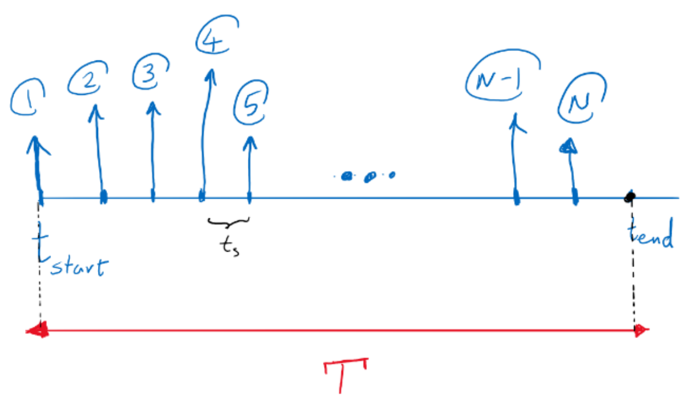
تمرین کامپیوتری پنجم

مهلت تحویل: یکشنبه ۲۵ آذر ساعت ۱۷:۰۰

بخش 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 و در ادامه درس نیز خواهیم دید، 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 $t = t_{start}\colon t_s\colon t_{end} - t_s$ به طوری که $t_{start} - t_{end} = T$ ، می توان به راحتی آن را با دستور $\mathrm{plot}(t,x)$) یا $stem(t,x)$) رسم کرد.



برای بردن سیگنال به حوزه ی فوریه از دستور $y = \mathrm{fftshift}(\mathrm{fft}(x))$ استفاده می کنیم. fft دستور اصلی است و fftshift فقط بازه ی متقارن حول فرکانس صفر را ایجاد می کند (مشابه آنچه که در درس در بخش سری فوریه گسسته مطرح شد). 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 $\mathrm{plot}(f, \mathrm{abs}(y))$ و برای رسم فاز می توان از دستور $\mathrm{plot}(f, \mathrm{angle}(y))$ 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 ($f_s \uparrow$)، 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 $t_s \to 0$) برود، عملا 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 کاملا باید درک کرده باشید که در sampling (نمونه برداری) 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 $f_s = 20\ Hz$ است. بنابراین رزولوشن فرکانسی برابر $\delta_f = 1\ Hz$ 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 تُن (تک فرکانس) به صورت

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

باشد، طبیعتا 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 تُن کمتر از $\delta_f = 1\ Hz$ 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 <u>۱-۰</u>" 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 و $f_s = 20\ Hz$ را در نظر بگیرید.

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

<u>تمرین ۱-۱)</u>

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 $f_s = 50\ Hz$ رسم کنید.

ب) اندازه ی این سیگنال در حوزه ی فوریه را با همان مفروضات قسمت الف رسم کنید. آیا نتیجه به دست آمده با دانسته های شما تطابق دارد؟

تمرین ۱-۲)

سیگنال $x_2(t) = \cos\left(30\,\pi t + \frac{\pi}{4}\right)$ را در نظر بگیرید.

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

ب) اندازه ی این سیگنال در حوزه ی فوریه را با همان مفروضات قسمت الف رسم کنید. آیا نتیجه به دست آمده با دانسته های شما تطابق دارد؟

ج) فاز این سیگنال در حوزه ی فوریه را با همان مفروضات قسمت الف رسم کنید. برای این کار ابتدا فاز را در فرکانس هایی که اندازه در آنها ناچیز است صفر کنید و سپس فاز را به صورت مضربی از π نمایش دهید. برای این کار از دستور زیر استفاده کنید. . آیا نتیجه به دست آمده با دانسته های شما تطابق دارد؟

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

theta = angle(y);

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

بخش دوم:

در تمرین کامپیوتری قبل، برای ارسال پیام، کدگذاری روی دامنه ی سیگنال اعمال شد، حال می خواهیم کدگذاری روی فرکانس داشته باشیم.

به عنوان مثال برای سرعت ارسال اطلاعات $1\ \frac{bit}{sec}$ ، 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، سیگنال دریافتی را به حوزه ی فوریه ببریم (دستور *fft*) و در بین فرکانس های انتخابی اولیه، فرکانس اصلی سیگنال را پیدا کنیم و متناظر با آن بیت ارسالی را استخراج کنیم.

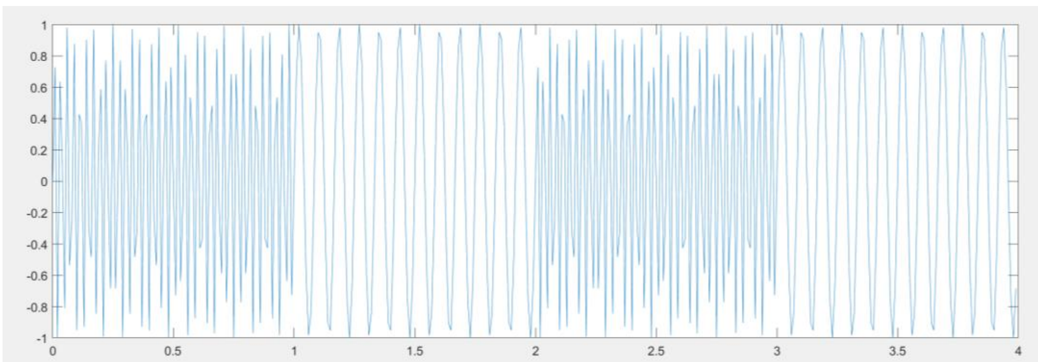
به عنوان یک مثال دیگر اگر بخواهیم سرعت ارسال اطلاعات $2\ \frac{bit}{sec}$ باشد، برای ارسال 00 سیگنال $x_0(t) = \sin\,(2\pi\, f_0\, t)$ را به مدت یک ثانیه، برای ارسال 01 سیگنال $x_1(t) = \sin\,(2\pi\, f_1\, t)$ را به مدت یک ثانیه، برای ارسال 10 سیگنال $x_2(t) = \sin\,(2\pi\, f_2\, t)$ را به مدت یک ثانیه 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، سیگنال دریافتی را به حوزه ی فوریه ببریم (*fft*) و فرکانس اصلی سیگنال را از بین فرکانس های انتخابی اولیه پیدا کنیم و متناظر با آن دو بیت ارسالی را استخراج کنیم (توجه داشته باشید، گرفتن *fft* 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؟

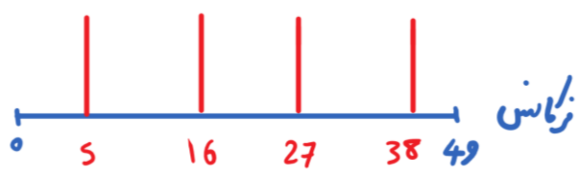
طبق آنچه در مقدمه ی بخش یک گفته شد، بازه ی فرکانسی یک سیگنال به طول N=100 سمپل و یک ثانیه ($f_s = 100\ Hz$) به صورت 50:1:49- هرتز خواهد بود. می دانیم سیگنال های حقیقی هم فرکانس منفی و هم فرکانس مثبت دارند، پس فقط به بخش فرکانس های مثبت توجه می کنیم.

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

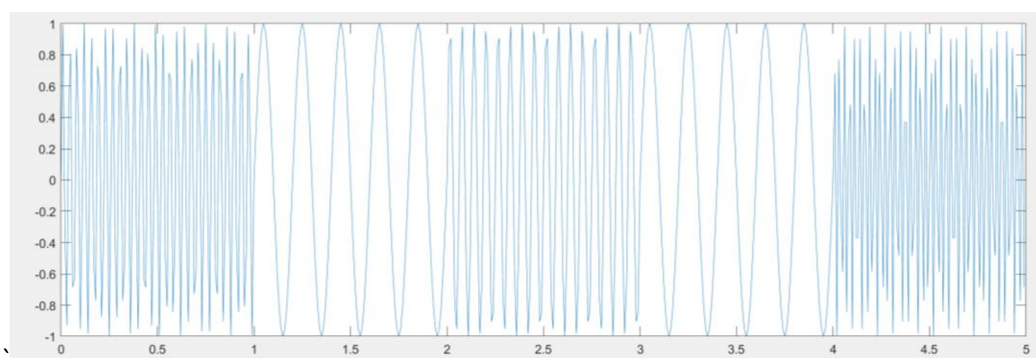
مثلاً رشته بیت 1010 به صورت زیر ارسال می شود:



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):



مثلا 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 یک بازه ی یک ثانیه ای از داده را

به حوزه ی فوریه می بریم، دیگر لزوماً فقط یک قله ی مرتفع نخواهیم دید و ممکن است قله جابجا شود. 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 نویزی، از میان فرکانس های انتخابی اولیه، مولفه ای که بیشترین مقدار را در اندازه ی تبدیل فوریه دارد به عنوان فرکانس اصلی در نظر می گیریم و سپس متناظر با آن سیگنال اصلی استخراج می شود. حال به سراغ شبیه سازی این موارد می رویم.

تمرین۲) تمرین کامپیوتری قبل 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 ۲-۱) یک سلول به اسم Mapset با ابعاد 32×2 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 تمرین کامپیوتری قبل است).

تمرین ۲-۲) 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 ۵ بیت برثانیه در نظر خواهیم گرفت.

تمرین ۲-۳) خروجی تابع *coding_freq* را برای پیام (کلمه‌ی) *signal* با سرعت ارسال اطلاعات یک و پنج بیت بر ثانیه به صورت جداگانه رسم کنید.

تمرین ۲-۴) 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 ۲-۳ با سرعت ارسال های مختلف کُد کردید تست کنید تا مطمئن شوید کدتان درست کار می کند.

تمرین ۲-۵) 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 *bit rate* های یک و پنج به صورت جداگانه گزارش کنید.

تمرین ۲-۶) قدرت نویز را کم کم و طی چندین مرحله افزایش دهید و هر بار قسمت ۲-۵ را تکرار کنید. مشاهدات خود را گزارش کنید. کدام یک از دو *bit rate* به نویز مقاوم تر بودند؟ آیا نتایج با آنچه در مقدمه بیان شد همخوانی داشتند؟

تمرین ۲-۷) طی شبیه سازی هایی که در قسمت ۲-۶ انجام دادید، بیشترین واریانس نویز که *bit rate* پنج به آن مقاوم بود به صورت تقریبی چند بود؟ برای *bit rate* یک چه طور؟

تمرین ۲-۸) 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.

تمرین ۲-۹) راجع به پاسخ مطرح شده در قسمت قبل، اگر نرخ نمونه برداری f_s را افزایش دهیم ولی پهنای باند مصرفی را افزایش ندهیم، بازهم سرعت ارسال اطلاعات به نویز مقاوم می شود؟

نکات کلی:

- درصورت وجود هرگونه پرسش و ابهام به امیرحسین نیکوسخن و استاد ایمیل بزنید.
- فایل نهایی شما باید به صورت یک فایل زیپ شامل گزارشکار به فرمت PDF و کد های متلب و سایر فایل های خواسته شده باشد.