



بسمه تعالی

دانشکده ی مهندسی برق و کامپیوتر

درس سیگنال ها و سیستم ها

تمرین کامپیوتری ۳



استاد: دکتر ربیعی

مهلت تحویل: جمعه ۲۸ خرداد ۱۴۰۰

طراح: امیرحسین ناظری

هدف از این تمرین آشنایی با کاربرد های تبدیل فوریه و اسپکتروگرام سیگنال ها در محیط نرم افزار متلب می باشد.

۱. فیلتر و حذف نویز

بیوسیگنال ها به هر سیگنال قابل اندازه گیری در بدن موجودات زنده گفته می شود. از انواع این سیگنال ها می توان به ECG یا نوار قلب اشاره کرد. این سیگنال داده های مهمی را برای تشخیص درست وضعیت بیمار در بر دارد. اما پیش از آنالیز سیگنال، باید سیگنال خام را به جهت حذف نویز های موجود پردازش کرد. نویز های بیو سیگنال ها شامل سه دسته ی زیر می باشند:

- انحراف از خط اصلی (نویز فرکانس پایین): این نویز را که ناشی از حرکت بدن، تنفس و تغییر در امپدانس الکتروود هنگام اندازه گیری است، که می توان با استفاده از یک فیلتر بالاگذر (highpass) آن را حذف کرد.
- نویز ناشی از سیگنال های عصبی عضلات (نویز فرکانس بالا): این نویز های نیز از حرکت هنگام اندازه گیری ناشی شده اما معمولاً در فرکانس های بالاتر رخ می دهد و حذف آن ها به دلیل تداخل طیفشان با طیف ضربان قلب به آسانی قرار دادن یک فیلتر نیست.
- نویز برق شهری: این نوع نویز بسته به فرکانس برق شهری، در ۵۰ هرتز یا ۶۰ هرتز مشاهده می شود، که برای حذف این نویز از فیلتر میانگذر استفاده می شود.

با توجه به توضیحات فوق، مراحل زیر را انجام داده و نتایج هر فسمت را در گزارش خود بیاورید.

۱.۱ رسم طیف

۱. نوار قلب قبلاً توسط دستگاه با فرکانس نمونه برداری $f_s = 500 \text{ Hz}$ ثبت شده است. این داده ها در فایل به نام ecg.dat در اختیار شما قرار گرفته است، می توانید آن ها را با استفاده از دستور زیر وارد Matlab کنید.

```
ecg_data = load('ecg.dat')
```

سپس بازه ی ۵ ثانیه از قسمت دلخواهی از سیگنال (به اندازه $5 \times f_s$ سمپل) را جدا کنید، آن را `ecg_data_short` نامگذاری کرده و رسم کنید.

دقت کنید که محور افقی نمودار بر حسب زمان مدرج شود. تقریباً چند دوره تناوب سیگنال در این بازه قرار دارد؟

۲. اندازه ی طیف سیگنال *ecg_data_short* را رسم کرده و فرکانس نویز برق شهری را مشخص کنید.

*دقت کنید محور فرکانس را به درستی مدرج کرده باشید.

۲.۱ حذف نویز فرکانس پایین

ابتدا می خواهیم نویز فرکانس پایین سیستم را از بین ببریم. توضیح دهید طبق طیف بخش قبل، این نویز تقریباً در چه بازه ی فرکانسی قرار دارد؟

برای حذف این نویز از فیلتر بالاگذر استفاده می کنیم. یک فیلتر بالاگذر ساده، مشتق گیر مرتبه یک می باشد.

$$y[n] = \frac{1}{T_s} (x[n] - x[n-1])$$

به صورت شهودی توضیح دهید چگونه این فیلتر فرکانس های پایین را عبور نمی دهد.

پاسخ فرکانسی پیوسته-زمان این فیلتر به صورت $H(f)=2\pi/f$ است. از روی اندازه ی طیف نیز بالا گذر بودن فیلتر را استدلال کنید.

این فیلتر را با فیلتر *Moving Average* زیر ترکیب کرده:

$$\begin{aligned} y_1[n] &= \frac{1}{2} (y[n] + y[n-1]) = \frac{1}{2T_s} (x[n] - x[n-1] + x[n-1] - x[n-2]) \\ &= \frac{1}{2T_s} (x[n] - x[n-2]) \end{aligned}$$

از معادله ی تفاضلی فوق بهره ی فیلتر $\frac{1}{2T_s}$ بدست آمده و ضرایب آن به صورت ماتریس $[1,0,-1]$ خواهد بود. برای پیاده سازی این فیلتر و اعمال آن بر سیگنال *ECG* از دستور زیر استفاده کنید.

```
ecg_highpassed = filter([1,0,-1], [1], ecg_data_short)
```

حال دوباره این سیگنال را بر حسب زمان رسم کنید و تفاوت را توصیف کنید.

*دقت کنید که در قطعه کد فوق بهره ی فیلتر، قرار داده نشده و شما باید بهره را در آن ضرب کنید.

۳.۱ حذف نویز فرکانس بالا

در این قسمت قصد داریم، فیلتری در حوزه ی فرکانس طراحی کنیم که نویز های فرکانس بالا را با کمترین از دست رفتن فرکانس های باند میانی، حذف کند.

بدین منظور از فیلتر باترورث (*Butterworth*) استفاده می کنیم. که اندازه ی پاسخ فرکانسی آن به صورت زیر است.

$$|H(f)| = \sqrt{\frac{1}{1 + \left(\frac{2\pi f}{j f_c}\right)^{2N}}}$$

در این رابطه f_c فرکانس قطع فیلتر و N مرتبه ی فیلتر است و این دو درواقع پارامتر های این نوع فیلتر هستند، که توسط طراح تعیین می شوند. درباره ی این فیلتر، پاسخ فرکانسی آن و تاثیر مرتبه آن تحقیق کنید و در گزارش کار خود بیاورید.

اندازه ی پاسخ فرکانسی فیلتر را برای فرکانس قطع ۱۰۰ هرتز و مرتب های ۲، ۴، ۸ نیز رسم کنید و در گزارش کار خود بیاورید. فیلتری که در این بخش طراحی می کنیم، یک فیلتر باترورث پایین گذر مرتبه ۸ با فرکانس قطع ۷۰ هرتز (فرکانسی بالاتر از نویز برق شهر و فرکانس های مربوط به داده اصلی) می باشد، برای این کار به قطعه کد زیر توجه کنید:

```
fc = 70;
wn = fc/(fs/2) % what does fs/2 represent?
n = 8;
[b, a] = butter(n, wn, 'low')
ecg_lp = filter(b, a, ecg_hp)
```

۴.۱ حذف نویز فرکانس میانی

در این قسمت فیلتری طراحی می کنیم که فرکانس برق شهر را حذف کند. این فیلتر یک فیلتر میان‌گذر یا *notch* می باشد و باید بسیار تیز باشد تا بتواند یک بازه ی کوچک از فرکانس را حذف کند.

با استفاده از قطعه کد زیر این فیلتر را روی سیگنال اعمال کنید.

```
f_line = ? % ? is powerline frequency
b = [1, -2*cos(2*pi*f_line/fs), 1];
a=[1];
ecg_filtered = filter(b, a, ecg_lp);
```

در نهایت سیگنال *ecg_filtered* آماده ی پردازش داده های مفید است.

۱.۵ بدست آوردن ضربان قلب

بدین منظور تابه خود همبستگی سیگنال بدست آمده از بخش قبل را بدست آورده و رسم می کنیم.

راجع به عمل ریاضی پیاده سازی شده در این تابع، ارتباط آن بابدست آوردن نرخ ضربان قلب و کاربرد های آن تحقیق کنید و در گزارشکار خود بیاورید.

با استفاده از قطعه کد زیر نمودار خود همبستگی سیگنال مذکور را رسم کنید و توضیح دهید از روی این نمودار، چگونه ضربان قلب بدست می آید و همچنین مقدار تقریبی برای نرخ ضربان قلب را گزارش کنید.

```
[cor, lag] = xcorr(ecg_filtered);  
figure  
plot(lag, cor)  
title('Auto-correlation function of filtered ECG')  
xlabel('samples')
```

اسپکتروگرام (spectrogram)، یک نوع نمایش بصری از طیف فرکانسی صوت یا سیگنال هایی است که در طول زمان تغییرات زیادی دارند. از اسپکتروگرام می توان برای شناسایی کلمات گفته شده به صورت آوایی و تحلیل آوا های مختلف حیوانات استفاده کرد. از آنجا که در این عمل، صوت به صورت تصویر مدلسازی می شود، یکی از کاربرد های بسیار مهم اسپکتروگرام تحلیل صوت با استفاده از روش های پردازش تصویر می باشد که موضوع بحث این درس نمی باشد.

اسپکتروگرام در واقع نشان دهنده مقدار طیف سیگنال در فرکانس های مختلف و بازه های زمانی مختلف می باشد.

فرض کنید $x[n]$ سیگنالی با طول N است. قطعه هایی متوالی از سیگنال x به طول m که $m \ll N$ را در نظر بگیرید. همچنین فرض کنید $X \in \mathbb{R}^{m \times (N-m+1)}$ ماتریسی از این قطعه های متوالی در ستون های متوالی است. به عبارت دیگر $[x[0].x[1].\dots.x[m-1]]^T$ در ستون اول، $[x[1].x[2].\dots.x[m]]^T$ در ستون دوم و به همین ترتیب دیگر ستون های X ساخته می شوند. ستون ها و ردیف های این ماتریس دارای اندیس زمانی هستند و با گرفتن FFT از هر ستون ماتریس \hat{X} حاصل می شود.

اسپکتروگرام سیگنال $x[n]$ به طول پنجره m ماتریس \hat{X} است، که ستون های آن تبدیل فوریه ی ستون های X است. دقت کنید که ردیف های \hat{X} دارای اندیس فرکانس و ستون های آن دارای اندیس زمان هستند. هر نقطه در ماتریس \hat{X} مربوط به یک فرکانس و یک زمان مشخص است. پس \hat{X} نمایشی ترکیبی از فرکانس-زمان سیگنال $x[n]$ می باشد.

۱.۱ رسم اسپکتروگرام

همانطور که ذکر شد، اسپکتروگرام یک ماتریس است. برای تجسم آن می توانید ماتریس را به عنوان یک تصویر مشاهده کنید که مقدار ماتریس در $[i, -1]$ امین محل، رنگ تصویر در آن را مشخص می کند. در متلب می توانید از تابع `imagesc()` استفاده کنید.

با استفاده از تابع `audioread()` فایل `tel.wav` را بخوانید، سپس برای طول پنجره ی ۵۰ ماتریس X را بسازید و با گرفتن `fft` از آن ماتریس \hat{X} را بسازید و اندازه ی آن را با دستور `imagesc` رسم کنید، تصویر خروجی را با برچسب های مناسب در گزارشکار خود بیاورید.

حال اسپکتروگرام سیگنال صوتی را که از فایل خواندید، با دستور `spectrogram()` با طول پنجره ی ۵۰ رسم کنید و تصویر را در گزارشکار خود بیاورید.

۱. دو تصویر رسم شده چه تفاوت هایی دارند و علت این تفاوت چیست؟

۲. آیا می توان با استفاده از اسپکتروگرام شماره ی تلفن را از روی صدای کلیدها بدست آورد؟ اگر پاسخ شما مثبت است، توضیح دهید چگونه؟

۲. صدای کلید تلفن

در این بخش با چگونگی استفاده از فرکانس های مختلف برای تشخیص کلید فشار داده شده در تلفن، آشنا خواهید شد. صدایی که هنگام فشار دادن یک کلید می شنوید، جمع دو سیگنال سینوسی است. سیگنال با فرکانس بالا ستون کلید و سیگنال با فرکانس پایین ردیف کلید در صفحه کلید تلفن را مشخص می کند. در دو جدول پایین فرکانس ها استفاده شده برای کلید های مختلف را مشاهده می کنید. همچنین در این بخش فرض کنید فرکانس نمونه برداری $f_s = 8192 \text{ Hz}$ است و از fft با $n = 2048$ نقطه استفاده می شود.

کلید	ω_{row}	ω_{col}
0	0.7217	1.0247
1	0.5346	0.9273
2	0.5346	1.0247
3	0.5346	1.1328
4	0.5906	0.9273
5	0.5906	1.0247
6	0.5906	1.1328
7	0.6535	0.9273
8	0.6535	1.0247
9	0.6535	1.1328

ω_{col}	0.9273	1.0247	1.1328
ω_{row}			
0.5346	1	2	3
0.5906	4	5	6
0.6535	7	8	9
0.7217		0	

برای مثال سیگنال صوتی کلید شماره ۶ به صورت زیر می باشد:

$$key_6[n] = \sin(0.5906n) + \sin(1.1328n)$$

۱.۲ تولید صدای کلید

۱. ابتدا ماتریس های key_0 تا key_9 را برای $0 \leq n \leq 999$ را تشکیل دهید و سپس این سیگنال ها را با فرکانس نمونه برداری ذکر شده در فایل های **wav**. در پوشه ی **key_sounds** ذخیره کنید و به صدای آن ها گوش دهید. (باید صدایی مانند فشردن کلید های تلفن داشته باشند)

۲. با استفاده از تابع **my_fft** که در تمرین کامپیوتری قبل پیاده سازی کردید، تبدیل فوریه سیگنال های key_0 تا key_9 را در حوزه ی فرکانس نمایش دهید و فرکانس های مشاهده شده را با جدول فوق مطابقت دهید.

۳. یک ماتریس سطری به طول ۱۰۰ با نام **space** با درایه های صفر بسازید و سپس سیگنال (ماتریس سطری) **phone_num** را با کلید هایی که ارقام شماره دانشجویی خود را بیان می کند، به صورتی که بین هر دو کلید یک **space** وجود دارد، تشکیل دهید.

و سپس آن را در یک فایل **wav**. ذخیره نمایید.

۲.۲ تشخیص صدای کلید

در این بخش باید از روی سیگنال داده شده، شماره ی گرفته شده را تشخیص دهید. ابتدا فایل `phone.csv` را بخوانید.

در سیگنال `phone` طول شماره ی گرفته شده ۷ می باشد و از فرمت `space` مانند حالت قبل استفاده شده است (یعنی ۱۰۰۰ نمونه از سیگنال هر کلید و ۱۰۰ نمونه برای `space` به عنوان سکوت).

سپس سیگنال مربوط به هر رقم را جدا کنید و با استفاده از `my_fft` آن ها را در حوزه ی فرکانس نمایش دهید. از روی نمودار های رسم شده شماره ی گرفته شده را بیابید.

سپس تابعی بنویسید که برای هر سیگنال که از فرمت فوق تبعیت میکند، شماره گرفته شده را نمایش دهد.

***جهت اطمینان از عملکرد تابع می توانید از سیگنال شماره دانشجویی خود که بخش قبل ایجاد کردید، استفاده کنید.

نکات تحویل:

- فایل های خود را به صورت زیپ شده با فرمت CA#3_full name_student number در صفحه ی درس آپلود کنید.
- هدف این تمرین یادگیری شماسست. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
- سیاست تحویل با تاخیر در فایل قوانین تمرین های کامپیوتری آورده شده است.
- سوالات خود در خصوص این تمرین را در گروه تلگرامی درس و نیز از طریق ایمیل زیر می توانید مطرح نمایید:

ah.nazeri1@gmail.com

موفق باشید...