



باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

سیگنال‌ها و سیستم‌ها - گروه دکتر بهروزی - بهار ۱۴۰۱-۰۲

تمرین متلب سری دوم

موعد تحویل: ۲۰ اردیبهشت - ساعت ۲۳:۵۵

نحوه تحویل:

- گزارش تمرین خود را در قالب یک فایل pdf. تحویل دهید. در گزارش لازم است تمامی خروجی‌ها و نتایج نهایی، پرسش‌های متن تمرین و توضیح مختصری از فرآیند حل مسأله خود در هر قسمت ذکر کنید.
- کد کامل تمرین را در قالب یک فایل m. تحویل دهید. لازم است بخشهای مختلف تمرین در section های مختلف تفکیک شوند و کد تحویلی منظم و دارای کامنت گذاری مناسب باشد. بدیهی است آپلود کردن کدی که به درستی اجرا نشود، به منزله ی فاقد اعتبار بودن نتایج گزارش شده نیز می باشد.
- توابعی را که (در صورت لزوم) نوشته اید، حتما در انتهای کد در یک section جدا تحت عنوان functions ضمیمه کنید و از ایجاد فایل جدای m. برای آن‌ها خودداری کنید.
- مجموعه ی تمامی فایل‌ها (گزارش، کد اصلی و خروجی‌های دیگر در صورت لزوم) را در قالب یک فایل zip/.rar. ذخیره کرده و از طریق سامانه CW تحویل دهید.
- نام گذاری فایل‌های تحویلی را به صورت zip.rar.m CHW02_StudentNumber انجام دهید.
- ابهام یا اشکالات خود را می‌توانید با طراحان تمارین از طریق آی‌دی‌های @MMohseniZ و @ilia_ad7 مطرح نمایید.

معیار نمره‌دهی:

- ساختار مرتب و حرفه ای گزارش
- استفاده از توابع و الگوریتم‌های مناسب
- پاسخ به سؤالات تئوری و توضیح روش‌های مطلوب سوال
- کد و گزارش خروجی کد برای خواسته‌های مسأله

نکته مهم:

- شرافت انسانی ارزشی به مراتب والاتر از تعلقات دنیوی دارد. در صورت کشف شباهت غیر قابل توجیه بین کدها و گزارش‌های آپلود شده، نمره نهایی تمرین مذکور برای همه افرادی که مشارکت داشته‌اند قرینه نمره‌ای که کسب کرده‌اند ثبت خواهد شد.

۱ سری فوریه

در این سوال قصد داریم ضرایب سری فوریه چند تابع را پیدا کرده و آن‌ها را با استفاده از سری فوریه تقریب بزنیم. همانطور که در درس دیدیم میتوان ضرایب سری فوریه یک سیگنال پیوسته زمان و متناوب $x_c(t)$ با دوره تناوب T را از رابطه زیر بدست آورد:

$$a_k = \frac{1}{T} \int_T x_c(t) e^{-j\frac{2\pi}{T}kt} dt$$

حال برای محاسبه این ضرایب با استفاده از متلب، میتوان انتگرال بالا را با سری زیر تقریب زد:

$$a_k \approx \frac{T_s}{T} \sum_{n=0}^{N-1} x_c(nT_s) e^{-j\frac{2\pi}{T}k(nT_s)}$$

که در آن از سیگنال پیوسته-زمان $x_c(t)$ در بازه $0 \leq t < T$ با فرکانس F_s نمونه برداری شده است. (طبعاً فاصله زمانی بین دو نمونه از رابطه $T_s = \frac{1}{F_s}$ بدست می‌آید و تعداد نقاط نمونه‌برداری هم $N = F_s T$ میشود). حال با تعریف سیگنال گسسته زمان $x_d[n] = x_c(nT_s)$ سری بصورت زیر بازنویسی میشود:

$$a_k \approx \frac{1}{N} \sum_{n=0}^{N-1} x_d[n] e^{-j\frac{2\pi}{N}kn} = \frac{1}{N} \text{DFT}(x_d[n], N)$$

در اینجا کافیت عبارت $\text{DFT}(x_d[n], N)$ را با دستور `fft` محاسبه کنید. در اینصورت، ضرایب سری فوریه گسسته تابع $x_d[n]$ را برای یک دوره تناوب N تایی آن بدست آورده‌اید. در واقع مولفه‌های بردار حاصل، به ترتیب متناظر با اندیس‌های $k = 0, 1, \dots, N-1$ هستند. بدین صورت با انتخاب مناسب فرکانس نمونه‌برداری، میتوانید ضرایب سری فوریه تابع پیوسته زمان را برای هر اندیس دلخواه بدست آورید.

۱. در این قسمت میخواهیم تابعی طراحی کنیم که ضرایب سری فوریه سیگنال پیوسته زمان $x_c(t)$ را طبق توضیحات داده شده محاسبه کند. تابع را بصورت زیر تعریف کنید:

`FSC_calculator(xc, Fs, T, k_min, k_max)`

که در آن:

- `xc` یک `function handle` است که از آن نمونه‌برداری خواهید کرد.
- محدوده مطلوب برای اندیس‌های سری فوریه از `k_min` و `k_max` بدست می‌آید.
- `Fs` باید به قدری بزرگ انتخاب شود که `k_max - k_min > Fs * T`
- بازه نمونه برداری هم بصورت $0 \leq t < T$ خواهد بود.

۲. سیگنال‌های پیوسته زمان متناوب زیر را در نظر بگیرید. با استفاده از تابع قسمت قبل، ضرایب سری فوریه را برای گسترش با دوره تناوب T آنها بدست آورید. سپس مقادیر حقیقی و موهومی آنها را با مقادیر تحلیلی‌شان مقایسه کرده و تطابق آنها را بررسی کنید:

- پالس مثلثی با دامنه $A = 1$ و دوره تناوب $T = 2$ و `duty cycle = 0.5`.

- موج دندان اره‌ای با دامنه $A = 2$ و دوره تناوب $T = 1$. دقت کنید بهتر است این تابع را بصورت $x_{c2}(t) = |t|$ تعریف کنید.

• تابع $x_{c3}(t) = e^{2t} + 2x^3$ با دوره تناوب $T = 3$.

آیا پدیده Gibb's را در نمودارهای بالا مشاهده میکنید؟ در صورت وجود آنرا توجیه کنید.

۳. تابعی طراحی کنید که سیگنال پیوسته زمان متناوب را با داشتن ضرایب سری فوریه ak با اندیس‌های مشخص شده، از طریق سری فوریه بازایی کند. تابع را بصورت زیر تعریف کنید:

FS_calculator(ak, k_min, k_max, T, t)

که در آن t بردار زمان دلخواه است. حال سیگنال‌های تعریف شده در قسمت قبل را با استفاده از دو تابعی که نوشتید، با تقریب حاصل از سری فوریه آنها در یک plot مقایسه کنید. در این قسمت از $|k_{\min}| = k_{\max} = 10$ استفاده کنید.

۴. سیگنال متناوب $x_c(t)$ با دوره تناوب $T = 8$ را در نظر بگیرید که یک دوره تناوب آن بصورت زیر است:

$$x_c(t) = \Pi\left(\frac{t+1}{4}\right)$$

حال ضرایب سری فوریه a_k این سیگنال را به ازای هر $K \in [1, 30]$ برای $-K \leq k \leq K$ بدست آورید. نمودار سیگنال $x_c(t)$ و سیگنال بازایی شده از طریق سری فوریه را در یک دوره تناوب رسم کرده و مقایسه کنید. این فرآیند باید بصورت real time باشد. برای این منظور از دستورات pause و clf استفاده کنید. در نهایت حاصل را بصورت یک فایل gif. با نام FS_Simulation.gif ذخیره کنید و به فایل تحویلی ضمیمه کنید.

۲ صفحه کلید

در این بخش از تمرین قصد داریم با DTMF (Dual-tone Multi-frequency signaling) که یک نوع روش سیگنال‌دهی مخابراتی است آشنا شویم. در این روش برای انتقال داده در کانال از ترکیب سیگنال از سیگنال‌های سینوسی با فرکانس‌های از پیش تعیین شده استفاده میشود. یکی از کاربردهای رایجی که برای این نوع Signaling میتوان مثال زد شماره‌گیری در تلفن ثابت است که با استفاده از آن مرکز مخابرات شماره گرفته شده را تشخیص داده و مارا به مخاطب مربوطه وصل میکند. هر سطر از صفحه کلید تلفن و هر ستون آن متناظر با یک سیگنال سینوسی با فرکانس منحصر به فرد است. زمانی که یک کلید را فشار میدهید یک سیگنال که حاصل جمع سیگنال‌های سینوسی متناظر با سطر و ستون کلید فشار داده شده است تولید و فرستاده میشود (همان صدایی که خودتان هم موقع شماره‌گیری میشنوید، سیگنال با فرکانس بالا مربوط به ستون و سیگنال با فرکانس پایین مربوط به سطر شماره است). با توجه به اینکه هر سطر و ستون دارای فرکانس بخصوصی است در مرکز مخابرات میتوان فهمید که چه کلیدی توسط کاربر فشار داده شده است. در این سوال نرخ نمونه برداری از سیگنال‌ها (F_s) را 8192 Hz در نظر گرفته و از تابع fft با ۲۰۴۸ نقطه استفاده کنید.

۱. در جدول زیر فرکانس‌های استفاده شده برای هر شماره را مشاهده میکنید:

Freq.	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

برای مثال هنگامی که کلید شماره ۲ فشرده میشود سیگنال زیر تولید خواهد شد:

$$k_2(t) = \cos(2\pi \cdot 1336 \cdot t) + \cos(2\pi \cdot 697 \cdot t)$$

با فرض اینکه طول فشرده شدن هر کلید ۱۰۰۰ نمونه باشد و بین هر فشرده شدن نیز ۱۰۰ نمونه (space) داشته باشیم، صوت حاصل از شماره‌گیری شماره دانشجوییتان را در فایل با نام StudentID.wav ذخیره کنید. با فرض اینکه d_i بردار نمونه برداری شده از $k_i(t)$ با طول ۱۰۰۰ باشد، سیگنال حاصل برای شماره فرضی ۱۲۳۴ بصورت زیر میشود:

$$S = [d_1 \text{ space } d_2 \text{ space } d_3 \text{ space } d_4]$$

۲. با استفاده از تابع fft سیگنال‌های d_0, d_1, \dots, d_9 را در حوزه فرکانس بدست آورده و رسم کنید، دقت کنید محور فرکانس باید برحسب Hz بوده و حول ۰ متمرکز باشد. آیا مشاهدات شما با چیزی که انتظار داشتید تطابق دارد؟

۳. تابعی بنویسید که با خواندن فایل‌های صوتی dialing1.wav, dialing2.wav و dialing3.wav شماره گرفته شده را تشخیص داده و در خروجی برگرداند. فرمت شماره‌گیری این فایل‌های صوتی دقیقاً مانند فرمت فایل StudentID.wav میباشد، یعنی هر شماره دارای ۱۰۰۰ نمونه و بین شماره‌ها ۱۰۰ نمونه (space) قرار دارد.

۴. با استفاده از تصاویری که در فولدر Numbers قرار داده شده است، تابع قسمت قبل را به گونه‌ای تغییر دهید که شماره را بصورت تصویر خروجی دهد. برای مثال اگر شماره گرفته شده ۹۶۷۸۴۵ باشد خروجی باید به شکل زیر باشد:



شکل ۱: نمونه خروجی برای شماره ۹۶۷۸۴۵

۵. (امتیازی) این بخش مانند بخش قبلی است، با این تفاوت که باید شماره‌گیری‌های واقعی را تشخیص دهید. در این شماره‌گیری‌ها دیگر زمان فشرده‌شدن هر کلید و فاصله بین فشردن کلیدها ثابت نیستند و می‌توانند طول متفاوتی داشته باشند. تابعی بنویسید که با خواندن فایل‌های صوتی `realDialing1.wav`, `realDialing2.wav`, و `realDialing3.wav` شماره گرفته شده در آن‌ها را بصورت تصویر در خروجی برگرداند. (فاصله بین شماره‌ها کمتر از ۱۰۰ نیست)

۳ مقدمه‌ای بر پردازش صوت (۲)

در این سوال می‌خواهیم از یکی از کاربردهای تبدیل فوریه در پردازش صوت استفاده کنیم. می‌دانیم که یک فایل صوتی در واقع یک سیگنال گسسته-زمان است. مشابه آنچه که در مورد توابع پیوسته-زمان در درس دیدیم، برای سیگنال‌های گسسته-زمان نیز می‌توان سری فوریه و تبدیل فوریه تعریف کرد و از آن‌ها استفاده کرد. از آنجایی که در متلب هنگام پردازش یک فایل صوتی، با یک ماتریس (آرایه) اعداد کار می‌کنیم، می‌توان فایل‌های صوتی را سیگنال‌های گسسته-زمان در نظر گرفت. البته چون تبدیل فوریه یک سیگنال گسسته-زمان، یک تابع پیوسته-فرکانس است، در عمل از مفهوم دیگری به نام DFT استفاده می‌کنیم. مطابق تعریف داریم:

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn}$$

که $x[n]$ همان سیگنال صوت دیجیتال است. اگر دقت کنید متوجه می‌شوید ای تعریف مانند آن است که N نقطه ابتدایی سیگنال را در نظر بگیریم سپس آن را بصورت پی در پی کنار هم قرار دهیم یا replicate کنیم و ضرایب سری فوریه این سیگنال گسسته متناوب را حساب کرده و در آرایه X ذخیره کنیم. با توجه به اینکه ما می‌خواهیم تمام اطلاعات سیگنال ما حفظ شود تمام نقاط آن را در نظر می‌گیریم و N برابر با طول سیگنال است.

آنچه که دستور fft متلب محاسبه می‌کند در حقیقت همین آرایه $X(k)$ است، به این صورت که بردار $x[n]$ و N را تحویل می‌گیرد و $X(k)$ را در خروجی برمیگرداند. ثابت می‌شود تعریف با تعاریف قبلی سازگار است و در عمل از این تعریف برای تبدیل فوریه توابع در متلب استفاده می‌شود.

اگر بخواهیم دقیق‌تر بیان کنیم، اگر تبدیل فوریه یک سیگنال گسسته را یک سیگنال پیوسته در نظر بگیریم، تبدیل فوریه گسسته (DFT) سیگنالی گسسته است که از سیگنال پیوسته تبدیل فوریه نمونه برداری شده است.

۱. فایل صوتی sound.wav را در محیط متلب لود کرده و با استفاده از تابع fft آن را در حوزه فرکانس نمایش دهید. دقت کنید که محور فرکانس را درست حول ۰ تنظیم کنید و برحسب Hz باشد. این سیگنال در چه محدوده فرکانسی محتوا دارد؟ (برای راحتی می‌توانید به جای کار با سیگنال stereo با میانگین کانال‌های چپ و راست کار کنید)

۲. به سیگنال صوت نویز زیر را اضافه کرده و در یک فایل صوتی با نام noisySound.wav ذخیره کنید.

$$n_0 \sim \mathcal{N}(0, 0.0025)$$

۳. اکنون می‌خواهیم نویزی که در قسمت قبل داشتیم را فیلتر کنیم. فایل فیلتری که در اختیاران قرار گرفته است را لود کرده و با استفاده از آن و دستورات متلب سیگنال نویزی قسمت قبل را فیلتر کرده و با نام filteredNoisySound1.wav ذخیره کنید.

۴. تبدیل فوریه سیگنال اصلی، سیگنال نویزی و سیگنال فیلتر شده را در کنار یکدیگر رسم کرده و مقایسه کنید. به سیگنال‌های نویزی و فیلتر شده گوش کنید، آیا نویز به طور مناسبی حذف شده است؟ از مقایسه فایل اصلی با فایل فیلترشده متوجه چه تفاوت‌هایی می‌شوید؟

۵. در این قسمت باید با توجه به تبدیل فوریه سیگنال اصلی و سیگنال نویزی فیلتری با فرکانس‌های قطع مناسب طراحی کنید، به گونه‌ای که با اعمال فیلتر نویز به اندازه خوبی حذف شود بدون اینکه اطلاعات صوت بیش از حد از بین برود. برای اینکار می‌توانید فیلر را ایده‌آل در نظر بگیرید که در فرکانس‌های قطع مقدار ۰ و در فرکانس‌های عبور مقدار ۱ دارد.

۶. فیلتر طراحی شده را به دو روش زیر سیگنال نویزی اعمال کنید:

- در حوزه فرکانس، تبدیل فوریه فیلتر را در تبدیل فوریه سیگنال ضرب کنید و با استفاده از `ifft` سیگنال فیلتر شده را به حوزه زمان برگردانید. سیگنال حاصل را با نام `filteredNoisySound2.wav` ذخیره کنید.
- در حوزه زمان فیلتر را با سیگنال نویزی کانوالو کنید. برای اینکار میتوانید از دستور `conv` استفاده کنید و از ویژگی `'same'` برای یکسان شدن طول سیگنال خروجی با سیگنال اصلی بهره ببرید. سیگنال حاصل را با نام `filteredNoisySound3.wav` ذخیره کنید.

۷. خروجی هر یک از دو روش ذکر شده در قسمت قبل را گوش دهید. آیا نویز به طور مناسبی حذف شده است؟ آیا خروجی دو روش بالا با یکدیگر تفاوت محسوسی دارند؟ این فایل‌های صوتی چقدر به صوت اصلی شباهت دارند؟