



## ۱. (۱۲٪) [ویژگی‌های سیستم‌ها]

خواص علی بودن، خطی بودن، تغییرپذیری با زمان، و حافظه‌دار بودن در مورد سیستم‌های زیر بررسی کنید.

ب-  $y[n] = nx[n]$

آ-  $y(t) = 2x\left(\frac{t}{2}\right) - 1$

ت-  $y(t) = \begin{cases} x(t) + x(t-2) & t \geq 0 \\ 0 & o.w. \end{cases}$

پ-  $y(t) = \int_{-\infty}^t x(\tau) d\tau$

ج-  $y(t) = \frac{1}{2M+1} \sum_{k=-M}^{+M} x[n-k]$

ث-  $y(t) = \sin(x(t))$

## ۲. (۱۸٪) [سیستم‌های LTI و کانولوشن]

آ- سیستم LTI، با پاسخ ضربه  $h(t) = \begin{cases} 4|1-t| & |t| \leq 1 \\ 0 & o.w. \end{cases}$  و ورودی  $x(t) = \begin{cases} 2 & -2 \leq t < 0 \\ 0 & o.w. \\ -2 & 0 \leq t \leq 1 \end{cases}$  را

در نظر بگیرید. بدون محاسبه انتگرال کانولوشن به پرسش‌های زیر پاسخ دهید:

- خروجی در کدام بازه‌ها دقیقاً برابر با صفر است؟

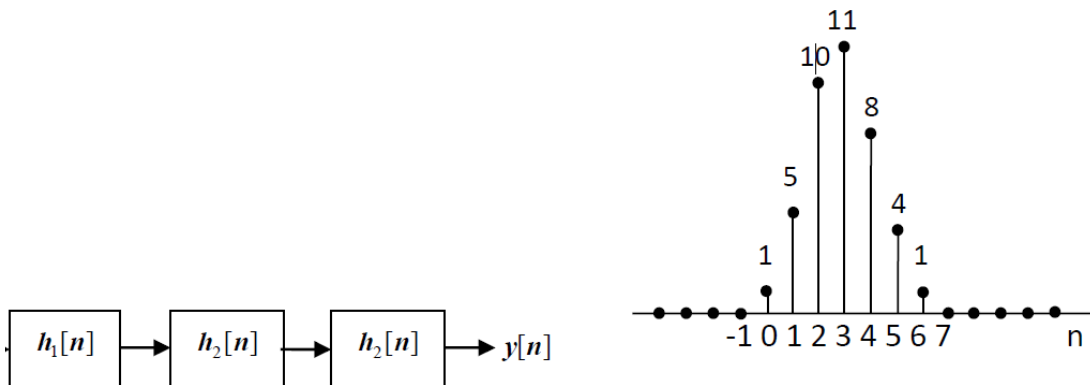
- خروجی در کدام  $t$  بیشینه است؟

- خروجی در  $t=1$  چه مقداری دارد؟

ب- همانگونه که در شکل زیر (سمت چپ) نشان داده شده است، رابطه ورودی و خروجی یک سیستم علی از اتصال سری سه سیستم دیگر درست شده است. پاسخ ضربه کل سیستم، در سمت راست نشان داده شده است. اگر  $h_2[n] = u[n] - u[n-2]$  باشد:

- پاسخ ضربه سیستم یکم ( $h_1[n]$ ) چیست؟

- پاسخ کل سیستم به ورودی  $x[n] = u[n] - 2u[n-1] + u[n-2]$  حساب کنید.



### ۳. (۲۰٪) [تبدیل فوریه]

بخش ۱: تبدیل فوریه گسسته را برای هر یک از سیگنال‌های زیر محاسبه کنید.

- $x[n] = 3(5)^{-|n-2|}$ , for all  $n$  (Break up the function and solve using superposition.)
- $x[n] = \alpha^n \cos(\omega_0 n + \phi)u[n]$ ,  $|\alpha| < 1$
- $x[n] = 7$  for all  $n$
- $x[n] = A \cos(\omega_0 n + \phi)$  for all  $n$
- $x[n] = A \sin(\omega_0 n + \phi)(u[n] - u[n - 9])$

**بخش ۲:** در ادامه قصد داریم که به کمک زبان برنامه نویسی صحت نتایج بدست آورده در بخش قبل را بررسی و ارزیابی کنید. به این منظور قطعه کد موجود در فایل `compute_dtfft.py` را که همراه تمرین ارسال شده است، برای ۵ سیگنال داده شده در بخش قبل تکمیل و محاسبات و نمودارهای خواسته شده را برای هر کدام را در گزارش خود قید کنید. پیاده‌سازی‌های خواسته شده را در فایل `main_3.py` به همراه گزارش خود ارائه کنید.

(الف) تابعی برای محاسبه DTFT توابع زمان گسسته پیاده‌سازی کنید.  
 (ب) در ادامه قطعه کد توسعه داده شده را برای انجام محاسبات و نمایش نمودارها بر حسب مقدار و فاز DTFTها کامل کنید.

دقت داشته باشید که نتایج ممکن است به دلیل بی‌نهایت بودن توابع زمان در طول زمان دقیق نباشد و



به دلیل پاسخ‌های فرکانسی در بعضی موارد ضربه‌ای هستند. با این وجود مقادیر بردارهای  $w$  و  $n$  مختلف را برای بدست آوردن تخمینی معقولی کاوش کنید. (پ) حال برای اطمینان از پیاده‌سازی‌های خود، به کمک `freqz` (از کلاس `scipy.signal.freqz`) نتایج خود را با نتایج حاصل از کتابخانه نام برده مقایسه کنید. در نظر داشته باشید که `freqz` فرکانس‌های نرمال شده را دریافت کرده و مقادیر را در دسیبل خروجی می‌دهد.

#### ۴. (۲۵٪) پاسخ فرکانس با فیلتر پایین‌گذر/بالاگذر

بخش ۱: تبدیل زیر را در نظر بگیرید:

$$h_{LP}[n] = \frac{\sin(\omega_c n)}{\pi n} \Leftrightarrow H_{LP}(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases}$$

همان‌طور که می‌دانید، تبدیل از دامنه زمان به دامنه فرکانس دارای پیچیدگی است. با این حال، تبدیل از دامنه فرکانس به دامنه زمان با استفاده از تبدیل فوریه معکوس زمان گسسته (IDTFT) آسان‌تر است. بنابراین این رابطه به راحتی با محاسبه تبدیل فوریه معکوس پایین‌گذر از  $H_{LP}(e^{j\omega})$  ممکن می‌شود.

الف) با استفاده از خواص DTFT،  $h_{HP}[n]$  (که پاسخ ضربه واحد از فیلتر بالاگذر ایده‌آل با پاسخ فرکانسی است) را بدست آورید.

$$H_{HP}(e^{j\omega}) = \begin{cases} 0, & |\omega| \leq \omega_c \\ 1, & \omega_c < |\omega| \leq \pi \end{cases}$$

ب) حال برای  $h_{BP}[n]$  که توسط DTFT به صورت زیر توصیف می‌شود:



$$H_{BP}(e^{j\omega}) = \begin{cases} 0, & 0 \leq |\omega| \leq \omega_1 \\ 1, & \omega_1 < |\omega| < \omega_2 \\ 0, & \omega_2 \leq |\omega| \leq \pi \end{cases}$$

۱. با در نظر گرفتن خواص DTFT،  $h_{BP}[n]$  را به صورت اختلاف پاسخ ضربه واحد دو فیلتر پایین‌گذر با فرکانس‌های برش مختلف به دست آورید (حتماً فرکانس‌های برش فیلترها را مشخص کنید).

۲.  $h_{BP}[n]$  را به صورت ضرب پاسخ ضربه واحد یک فیلتر پایین‌گذر در کسینوس به دست آورید (حتماً فرکانس برش فیلتر پایین‌گذر و دامنه و فرکانس کسینوس را مشخص کنید).

۳. حال نشان دهید که عبارتهای ریاضی حاصل شده با دو روش فوق دقیقاً معادل هستند.

**بخش ۲:** همانطور که پاسخ ضربه واحد را برای فیلتر بالاگذر و میان‌گذر در بخش قبل بدست آوردید، مشکل بنیادی که با این پاسخ‌ها داشتیم این بود که این پاسخ‌های ضربه واحد غیر صفر برای تمامی  $n$ ها و در بازه منفی تا مثبت بی‌نهایت بودند. در این مسئله، یک راه عملی برای تبدیل این پاسخ‌های ایده‌آل به فیلترهای دیجیتال قابل تحقق را بررسی خواهیم کرد. این رویکرد شامل قطع کردن هر دو انتهای پاسخ ایده‌آل، محدود کردن آن و سپس تنظیم آن به صورت علی است. به عنوان مثال، برای یک فیلتر پایین‌گذر، ما این کار را انجام می‌دهیم:

تعریف  $h_N[n]$  و  $h[n]$  به صورت زیر است:

$$h_N[n] = \begin{cases} h_{LP}[n], & \text{for } -N \leq n \leq N \\ 0, & \text{otherwise} \end{cases}$$

$$h[n] = h_N[n - N] = \begin{cases} \frac{\sin(\omega_c \cdot (n - N))}{\pi \cdot (n - N)}, & \text{for } 0 \leq n \leq 2N \\ 0, & \text{otherwise} \end{cases}$$



فایل `main_4.py` در اختیار شما قرار گرفته شده است، با توجه به موارد خواسته شده آن را تکمیل کنید.

الف) فیلتر پایین گذر  $h[n]$  را در فایل `compute_lowpass_impulse_response.py` پیاده سازی کنید، ورودی های این فیلتر دو آرگومان  $W_c$  و  $N$  است.

۱. پاسخ فرکانسی فیلتر خود را به کمک `freqz(h,1)` نمایش دهید (منظور از `freqz` کلاس `scipy.signal.freqz` است).

۲. با تغییر مقادیر  $N$  و  $W_n$  مشاهدات حاصل از این تغییرات را گزارش دهید. با توجه به بررسی های خود چه مقادیری را برای این پارامترها مناسب می دانید؟

ب) حال می خواهیم تاثیر فیلتر پایین گذر را بر فایل صوتی بررسی کنیم. به این منظور فایل صوتی داده شده را به صورت تک کاناله بخوانید. مقدار  $W_c$  را برابر  $0.35\pi$  قرار دهید و از میانگین دو رقم آخر شماره دانشجویی خود به عنوان  $N$  استفاده کنید (از آنجایی که نرخ نمونه برداری ۱۶۰۰۰ هرتز است، این مقدار  $W_c$  با فرکانس قطع واقعی ۲۸۰۰ هرتز مطابقت دارد). با در نظر گرفتن دستور `y = conv(x,h)`، به کمک پاسخ فیلتر  $h$  کانولوشن سیگنال صوتی  $X$  را محاسبه کنید.

۱. به دو ثانیه اول فایل صوتی، قبل و بعد از اعمال فیلتر گوش دهید. فیلتر تا چه حد انتظاری که داشتید را برآورده کرده؟

۲. طیف نگار را برای دو ثانیه اول فایل صوتی، قبل و بعد از اعمال فیلتر رسم کنید.

## ۵. (۲۵٪ + ۱۰٪ امتیازی) [پیاده سازی کاربردی: چت بات صوتی فارسی]

الف) در این سوال می خواهیم یک چت بات شبیه زیگپ (نسخه اندروید و PWA و تحت وب آن در سایت [Zigap.ir](http://Zigap.ir) است) بسازیم. بدین صورت که ابتدا در خود پایتون برنامه ی شما صدا (با فرض اینکه شما فارسی حرف می زنید) را ضبط می کند (برای این کار پیشنهاد می شود از کتابخانه ی `pyaudio` کمک بگیرید) سپس صدای ضبط شده باید به یک مدل تشخیص گفتار که فارسی را پشتیبانی می کند مثل مدل هزار داده شود تا تبدیل به متن شود.



<https://github.com/hezarai/hezar>

حال باید از api یکی از هوش مصنوعی ها مانند chatgpt یا gemini یا copilot یا ... استفاده کنید.

پیشنهاد می شود از api زیر که نیاز به api key ندارد استفاده کنید

<https://github.com/xtekky/gpt4free>

حال متن تبدیل شده خود را چاپ کرده و به مدل دهید و جواب بات را نیز دریافت کنید و چاپ کنید. برنامه شما باید هر بار از کاربرد بپرسد که می خواهد ادامه دهد و صدای جدید ضبط کند یا از برنامه خارج شود.

در انتها باید تمام گفتگوها ما بین کاربر و چت بات در یک فایل متنی (.txt) ذخیره شود.

در گزارش خود بیاورید که با تغییر پارامتر های مختلف در هنگام ضبط صدا مانند `sample_rate(frame_rate)` آیا در هنگام تبدیل صدا به متن تفاوتی پیدا می شود یا خیر. می توانید برای سنجش این کار چند بار یک جمله ی ثابت را با پارامتر های مختلف تست کنید) همه پارامترها را نام ببرید و برای هر کدام با ثابت نگه داشتن بقیه حداقل دو مقدار را امتحان کنید.

**ب) (امتیازی)** می توان برای هر بار استفاده از برنامه ابتدا از کاربر اسم کاربر و پسورد خاص خود را بگیریم و در دیتابیس چک کنیم اگر قبلا این کاربر و پسورد دارای گفتگو است، ابتدا گفتگو قبلی را بخوانیم و به بات نشان دهیم، سپس در اخر برنامه نیز گفتگوهای جدید به این گفتگو اضافه شوند و در فایل txt مربوط به آن کاربر نوشته شوند.

**نکته:** استفاده از هر کتابخانه ای در این سوال کاملا مجاز است.