



بنام خدا

دانشکده‌ی مهندسی برق و کامپیوتر

درس سیگنال و سیستم‌ها

تمرین کامپیوتری 2

استاد : دکتر اخایی

مهلت تحویل : 4 دی

سر فصل مطالب

- سوال 1 : Intro to Fourier Transform 2
- سوال 2 : Fast Fourier Transform(fft) /Discrete Fourier Transform 3
- سوال 3 : Inverse Fast Fourier Transform(iff)/ Discrete Inverse Fourier Transform 4
- سوال 4 : Applicaions of Fourier Transform 6
- نکات تحویل..... 10

سوال 1 :

بخش اول (Sine/Complex Sine Waves)

(الف)

در ابتدا برای $\text{Sample Rate} = 500$ ، موج سینوسی با فرکانس 3 هرتز و دامنه 2 و با فاز $\pi/3$ در بازه 0 تا 2 رسم کنید .

(ب)

مانند قسمت قبل این کار را برای موج سینوسی مختلط ($e^{i2\pi ft}$) تکرار کرده و قسمت حقیقی و موهومی را با Legend در یک نمودار رسم کنید.

(ج)

ابتدا یک Figure ایجاد کنید و سپس به کمک دستور fig.gca و قرار دادن Projection بر روی '3d' ، موج سینوسی مختلط را در فضای سه بعدی بر حسب سه مختصات زمان (time) ، مقدار حقیقی (Real Part) و مقدار موهومی (Imaginary Part) رسم کنید.

بخش دوم (Dot/Complex Dot Product)

(الف)

مفهوم ضرب داخلی (Dot Product) را با روش آماری تحلیل کنید (می توانید از Correlation Coefficient استفاده کنید) .

(ب)

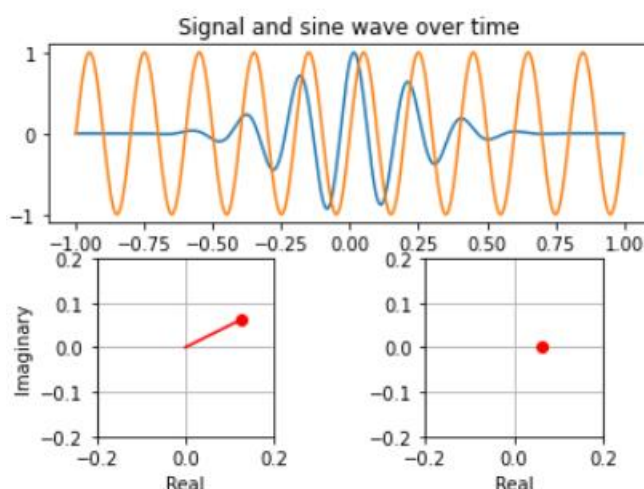
1. از [بخش اول](#) ، دو موج سینوسی مشابه با $\text{Sample Rate} = 500$ ، فرکانس 5 هرتز ، دامنه 2 و فاز $\pi/2$ را به کمک دستور np.dot در یکدیگر ضرب داخلی کنید.
2. با ثابت نگه داشتن Sample Rate ، فرکانس یک از دو موج سینوسی را ± 0.5 تغییر داده و دوباره ضرب داخلی را انجام دهید . این بار فرکانس ذکر شده را ± 0.35 تغییر دهید و ضرب داخلی را تکرار کنید. نتیجه را توجیه کنید.
3. این بار مانند قسمت قبل، یکی از دو فاز سیگنال را $\pm \pi/2$ و بار دیگر $\pm \pi/3$ تغییر داده و ضرب داخلی را انجام دهید . نتیجه را توجیه کنید.

(ج)

1. سیگنال با ضابطه $f(t) = \sin(2 * \pi * 5 * t + \theta) * e^{-t^2}$ را ایجاد کنید و آنرا برای $\text{Sample Rate} = 1000$ و $\theta = 0$ در بازه $[-1,1]$ رسم کنید.
2. حال برای فرکانس های در بازه $(2,10,0.5)$ ، `np.arange(2,10,0.5)` ، حاصل ضرب داخلی سیگنال های سینوسی با فرکانس های داده شده در $f(t)$ را در یک نمودار با دستور stem رسم کنید. برای کدام مقدار فرکانس ، نمودار بدست آمده بیشینه است ؟
3. قسمت قبل را برای $\theta = \pi/4, \pi/2$ انجام دهید . با تغییر θ چه تغییری در نمودار ها ایجاد می شود ؟ بنظر شما ضرب داخلی فعلی چه مشکلی را ایجاد می کند ؟

(د)

1. قسمت سوم بخش ج را این بار برای سیگنال های سینوسی مختلط در همان بازه فرکانسی و برای اندازه خروجی تست کنید. تفاوت نتایج را توجیه کنید.
2. * امتیازی : برای درک بهتر قسمت قبل ، برای مقادیر مختلف فاز در بازه $np.linspace(0, 7 * \frac{\pi}{2}, 100)$ در داخل یک Loop ، حاصل ضرب داخلی $f(t)$ را یکبار در سیگنال سینوسی و بار دیگر در سیگنال سینوسی مختلط حساب کرده و نمودار های مقدار حقیقی در حالت اول و مقدار حقیقی بر حسب مقدار موهومی در حالت دوم را در طی تغییرات فاز رسم کنید .
* راهنمایی : یک shot از خروجی بدست آمده به صورت زیر می باشد :



شکل 1-2-4-1: خروجی بدست آمده در یک Shot

سوال 2 :

فایل Signal.txt موجود در فایل پروژه را در ابتدا بخوانید . فرض کنید سیگنال بدست آمده در بازه $[0, 2]$ باشد.

بخش اول (نکات)

(الف)

1. Sample Rate سیگنال بدست آمده را محاسبه کنید.
 2. در مورد فرکانس Nyquist و ربط آن به فرکانس مثبت و منفی تحقیق کنید.
 3. برای آنکه اسکیل اندازه ضریب فوریه درست باشد ، نیاز به دو ضریب تصحیح دارد :
 - ضریب تصحیح اول به فرکانس مثبت و منفی مرتبط می باشد .
 - ضریب تصحیح دوم برای میانگین گیری از ضرایب فوریه بدست آمده می باشد .
- دو ضریب ذکر شده را اعلام کنید .

بخش دوم (پیاده سازی)

(الف)

1. در این قسمت باید ابتدا ضرایب فوریه سیگنال مجهول را به کمک ضرب داخلی در سیگنال سینوسی مختلط در یک آرایه ذخیره کنید. (راهنمایی برای اینکه نتایج حاصل دقیق باشد ، پارامتر t (Fourier time) در $e^{-i2\pi ft}$ را نرمالایز کنید (بین 0 و 1)).
2. سپس اندازه این ضرایب (Amplitudes) را ، بر حسب فرکانس (از صفر تا فرکانس Nyquist) رسم کنید و سیگنال مجهول را گزارش دهید .

(ب)

1. حال نمودار قسمت قبل را اینبار به کمک تابع `fft` در کتابخانه `scipy` ، رسم کنید (اسکیل های لازم را در قسمت 3 انجام دهید)
2. برای سیگنال با ضابطه ، $g(t) = \sin\left(\frac{\pi}{2}\right) + 2 * \sin(2 * \pi * 5 * t)$ ، نمودار قسمت قبل را رسم کنید . آیا اندازه ضرایب فوریه درست بدست آمده اند ؟ به نظر شما مشکل در کجاست ؟ با اعمال ضریب تصحیح جدید ، مشکل را حل کنید .

(ج)

*امتیازی : در مورد الگوریتم مورد استفاده در `fft` تحقیق کنید. با کمک کتابخانه `time` ، تفاوت زمان اجرایی `fft` در مقابل روش `for loop` مورد استفاده در `dft` را به صورت `Bar plot` برای یکی از سیگنال های دلخواه بالا نمایش دهید.

سوال 3 :

(الف)

در این قسمت می خواهیم با ضرایب فرویه سیگنال موجود در [سوال 2](#)، سیگنال اصلی را به کمک ضرب داخلی ، دوباره بازیابی کنیم. برای این منظور ، آرایه ای به طول سیگنال اصلی ایجاد کنید و مانند بخش اول سوال 2 ، اینبار به جای سیگنال سینوسی مختلط سوال 2 ، از قرینه شده آن نسبت به محور x ها استفاده کنید (به جای t ، $-t$ می گذاریم).
*راهنمایی : دقت کنید که در اینجا همچنان نیاز به استفاده از ضریب تصحیح دوم می باشد)
در نهایت سیگنال اصلی و بازیابی شده را در یک نمودار همراه با `legend` رسم کنید . آیا دو سیگنال بر هم منطبق هستند ؟ اگر خیر ، علت را توجیه کنید.

(ب)

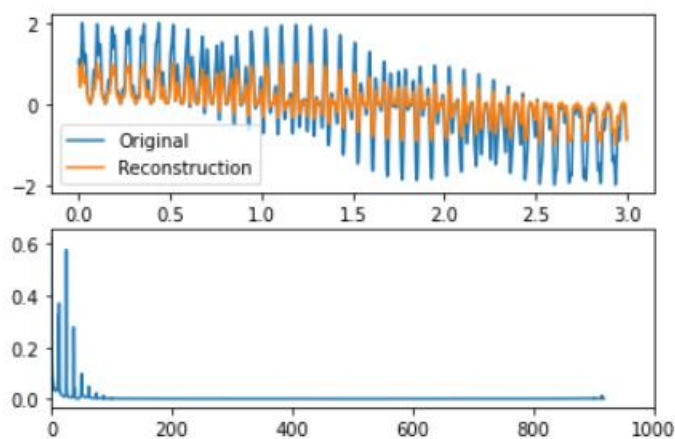
مراحل قسمت قبل را اینبار با توابع `fft` و `ifft` در کتابخانه `scipy` ، تکرار کنید .

(ج)

1. در ابتدا برای سیگنال با ضابطه $k(t) = (1 + \sin(2 * \pi * 12 * t)) * \cos(\sin(2 * \pi * 25 * t) + t)$ ، ضرایب فوریه را به کمک ضرب داخلی پیدا کنید . ($t \in [0,3]$, Sample Rate=1000)

2. اینبار مانند قسمت الف عمل کنید با این تفاوت که در هر loop، دو خروجی سیگنال اصلی ($k(t)$) بر حسب زمان (سیگنال ثابت) و سیگنال بازیابی شده بر حسب زمان (با گذشت زمان تکمیل می‌شود) در یک نمودار و نمودار ضرایب فوریه (با گذشت زمان تکمیل می‌شود) را در نمودار دیگر رسم کنید و روند تغییرات سیگنال بازیابی شده را در مرور زمان بررسی کنید.

*راهنمایی: یک Shot از خروجی بدست آمده به صورت زیر می‌باشد:



شکل 3-3-1: خروجی بدست آمده در یک Shot

(د)

در [سوال دوم](#)، [بخش دوم](#) ذکر شد، که برای نتایج دقیق، Fourier time را نرمالایز می‌کنیم و اولین عنصر آن از صفر شروع می‌شود (تمامی عناصر بین صفر و 1 هستند). حال می‌خواهیم این بار قبل نرمالیز کردن، ابتدا به هر عنصر آرایه فوق (Fourier time) مقدار α اضافه کرده که $\alpha \in np.linspace(0,10,3)$ بوده و سپس آنرا نرمالایز کنیم.

تمام مراحل [قسمت الف سوال 3](#) را برای سیگنال $h(t) = \sin(2 * \pi * 4.5 * t) + \cos(2 * \pi * 3 * t)$ و مقادیر α انجام داده و در نهایت در یک نمودار، سیگنال‌های بازیابی شده را با سیگنال اصلی مقایسه کنید. دلیل را به کمک نمودار فاز بدست آمده برای ضرایب فوریه توجیه کنید (*برای رسم فاز، از دستور `np.angle` استفاده کنید)

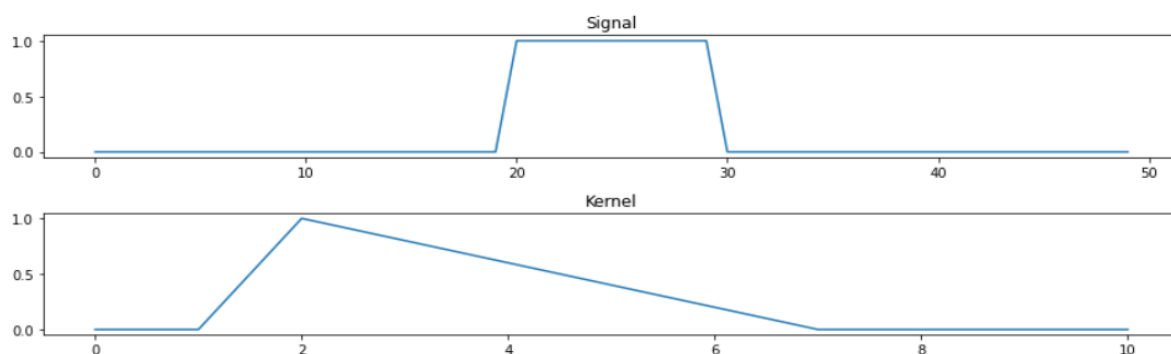
(ه)

سیگنال با ضابطه $M(t) = \sin(2 * \pi * 4 * t) + \sin(2 * \pi * 10 * t)$ را در نظر بگیرید. می‌خواهیم با انتقال سیگنال به حوزه فرکانسی، فرکانس 4 هرتز را حذف کرده و به کمک `ifft`، سیگنال سینوسی 10 هرتز را در فضای اصلی بازیابی کنیم. مراحل لازم را انجام دهید. آیا سیگنال بازیابی شده تنها شامل فرکانس 10 هرتز می‌باشد؟ در صورت لزوم، فرآیند حذف فرکانس ذکر شده را بازبینی کنید تا نتایج مطلوب بدست آید.

سوال 4 :

محاسبه کانولوشن در حوزه فرکانسی

در بسیاری از موارد ، محاسبه کانولوشن در حوه فرکانس ، بسیار ساده تر از حوزه زمان می باشد . سیگنال و کرنل زیر را در نظر بگیرید :



شکل 4-1-1: سیگنال و کرنل مورد نظر

(الف)

سیگنال و کرنل بالا را یک بار در حوزه زمان با دستور `np.convolve` ، کانوالو کرده و بار دیگر به کمک 2 تابع `fft` و `ifft` ، سیگنال بازیابی شده را در یک نمودار رسم کنید .

فیلتر کردن سیگنال در یک بعد (Narrow band Temporal Filtering):

در ابتدا دیتای مربوط به سیگنال های الکتریکی مغز را که در صورت پروژه قرار داده شده است را بصورت زیر لود کنید :

```
import scipy.io as sio
matdata = sio.loadmat('braindata.mat')
```

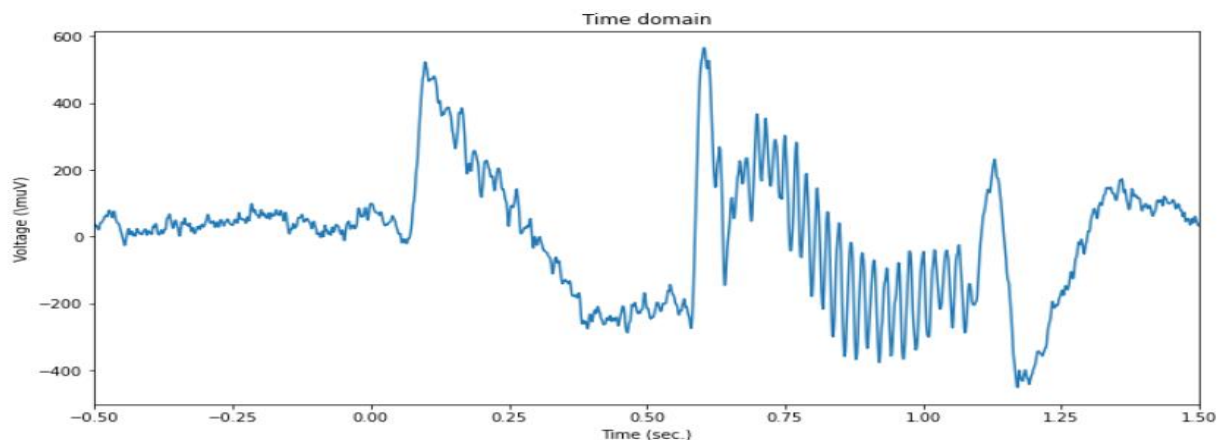
شکل 4-1-1: خواندن دیتاست 'braindata.mat'

همچنین مقادیر `Timevec` و `braindata` را به صورت دستور زیر بخوانید :

```
# extract data from mat file
timevec = matdata['timevec']
braindata = matdata['braindata']
```

شکل 4-2-2: خواندن لیبل ها

سیگنال فوق به باید به صورت باشد :



شکل 2-2-4 : نمودار ولتاژ بر حسب زمان

(الف)

در ادامه مراحل زیر را انجام دهید :

1. بردن سیگنال به حوزه فرکانس
2. رسم نمودار Spectrum (حداکثر خروجی را بر روی 500 قرار دهید)
3. انتخاب دو فرکانس دلخواه برای فیلتر
4. شیفت تمامی فرکانس ها برای هر کدام از فرکانس های انتخاب شده
5. استفاده از فیلتر گوسی به صورت $e^{-\left(\frac{x}{4}\right)^2}$ که در آن x ، فرکانس های شیت یافته می باشد.
6. ضرب فیلتر گوسی در ضرایب فوریه braindata
7. بازنمایی و plot سیگنال بدست آمده برای دو فرکانس فیلتر شده

(ب)

طبق نمودار Spectrum ، تحلیلی از نمودار بدست آمده در قسمت 7 ارائه دهید .

فیلتر کردن عکس در دو بعد (Image Filtering 2D)

در ادامه از عکس معروف "Lenna" در پردازش استفاده می‌کنیم :



شکل 4-2-1 : عکس مورد استفاده در پردازش تصویر به صورت RGB

عکس فوق را توسط دستور زیر در یک آرایه ذخیره کنید :

```
# load image and convert to float  
lenna = np.asarray( Image.open("Lenna.png") )
```

شکل 4-3-1 : خواندن عکس 'Lenna.png'

Low Pass Filtering

(الف)

به دلیل اینکه فیلتر کردن را برای عکس سیاه و سفید انجام می‌دهیم ، نیاز است که ماتریس اولیه را به 2 بعد کاهش دهیم . به این منظور در ماتریس اولیه ، به جای هر سطر میانگین آن سطر را قرار دهید . ماتریس جدید بدست آمده چه ابعادی دارد ؟ عکس سیاه و سفید بدست آمده را به کمک دستور `plt.imshow` بدست آورید.

(ب)

در این بخش مشابه فیلتر کردن در یک بعد عمل کرده ولی در عوض به جای `fft` از تابع `fft2` استفاده کنید و مراحل زیر را انجام دهید :

1. در مورد نحوه کارکرد `fft2` و `fftshift` در 2 بعد تحقیق کنید .

2. بدست آوردن ضرایب فوریه تصویر سیاه و سفید (برای اینکه فرکانس های پایین تر به وسط عکس و فرکانس های بالاتر به لبه عکس منتقل شوند ، در نهایت از `fftshift` استفاده کنید)

3. بدست آوردن Spectrum (در اینجا برای اینکه تمییز بهتری داشته باشیم ، به جای توان 2 ، از لگاریتم (`np.log`) استفاده کنید) و رسم آن به کمک دستور `plt.imshow`.

4. رسم فیلتر گوسی در دو بعد . (*راهنمایی : فیلتر گوسی در دو بعد را به صورت زیر ایجاد کنید) :

```
width = .1 # width of gaussian (normalized Z units)
lims = np.shape(b_and_w)
xr = stats.zscore(np.arange(lims[0]))
[x,y] = np.meshgrid(xr,xr)
gaus2d = np.exp( -( x**2 + y**2 ) / (2*width**2) )
```

شکل 4-3-2 : نحوه ایجاد فیلتر گوسی در دو بعد

5. رسم Spectrum ضرایب فوریه فیلتر شده (حاصل از ضرب در تابع گوسی) (در اینجا هم به جای توان 2 از لگاریتم استفاده کنید)

6. در نهایت تصویر سیاه و سفید فیلتر شده را رسم کنید (*راهنمایی : عکس بدست آمده باید مانند زیر باشد) :



شکل 4-3-3 : تصویر فیلتر شده پایین گذر

High Pass Filtering:

(الف)

تمامی مراحل ذکر شده در قسمت قبل را اینبار برای فیلتر بالا گذر انجام دهید (*راهنمایی : تنها نیاز به تغییر تابع گوسی در کد مرحله قبل دارید)



شکل 4-3-4 : تصویر فیلتر شده بالا گذر

(ب)

در عکس بدست آمده ، کدام ویژگی های صورت بیشتر به چشم می آیند ؟ دلیل را توجیه کنید

نکات تحویل :

1. زدن پروژه در IDE های مختلف (Pycharm , jupyter , colab , ...) به شرط استفاده از زبان پایتون مجاز است . پیشنهاد می شود که در Jupyter و یا Colab کد های خود را بنویسید که قابلیت Markdown کردن هم داشته باشید .
 2. در صورت استفاده از هر تابع آماده به غیر موارد ذکر شده در پروژه ، آن را در گزارش کار خود ذکر کنید.
 3. هر کدام از سوال ها در فایل های جداگانه ipynb یا py زده شود و اسم آن فایل را مطابق صورت سوال قرار دهید و در نهایت بصورت فایل zip به فرمت CA_num-Last_name-std_num در صفحه درس آپلود کنید .
 4. تمامی شکل های خروجی خواسته شده در هر سوال را با زیرنویس مربوط به آن سوال (به شکل های در صورت پروژه دقت کنید) مشخص کرده و در گزارش خود قرار دهید.
 5. هدف از تمرین های کامپیوتری کمک به یادگیری شماسست. بنابراین در صورت مشابهت بیش از حد در بخش های پروژه ، از شما نمره کسر خواهد شد .
 6. در صورتی که نسبت به پروژه سوال یا ابهامی داشتید ، از طریق ایمیل sh.vassef@ut.ac.ir یا در گروه تلگرامی با من در ارتباط باشید.
- موفق باشید .