



Notes:

- You have to write your code for the practical problems in python.
- You are free to use any libraries unless said otherwise, i.e., said to implement.
- Delivering a Jupyter notebook is enough, if you want to deliver your codes in python code, then you also need to write a document as report for the plots and explanations requested.
- **For each day of delay, a penalty of 33% is imposed.**
- Please include **"Signal PHW"** as the **SUBJECT** while you submit the solution to the homework.

If you have any questions, please feel free to contact us by email: Signal.Ce.Course@gmail.com

سوال ۱:

در این سوال قصد استفاده از خاصیت‌های جذاب حوزه فرکانس در کاربردهای صوتی را داریم. هدف ما ترکیب دو فایل صوتی در یک فایل است به گونه‌ای که باشیفت فرکانسی فایل نهایی هر بار یکی از صداها شنیده شود.

الف) فایل‌های music1.wav و music2.wav را باز کنید. ابتدا برای اطمینان، هر کدام از این دو فایل را پخش نموده و صدای آن‌ها را گوش دهید. اکنون اطلاعات زمانی هر کدام از فایل‌ها را به حوزه فرکانسی ببرید. سپس اطلاعات فرکانسی فایل music1 را شیفت فرکانسی دهید و حاصل را دوباره به حوزه زمان برگردانید. اکنون سیگنال بدست آمده را پخش کنید. مشاهده خواهید کرد که صدای واضحی شنیده نمی‌شود. علت چیست؟

ب) حال اطلاعات فرکانسی شیفت‌یافته برای فایل music1 را با اطلاعات فرکانسی music2 جمع نمایید. سیگنال نهایی را به حوزه زمان برگردانید و آن را پخش کنید. کدام یک از صداها پخش می‌شود؟

ج) اکنون سیگنال نهایی مرحله قبل را شیفت فرکانسی دهید و به حوزه زمان برگردانید و آن را پخش کنید. این بار کدام یک از صداها پخش می‌شود؟ در گزارش خود علت این پدیده را توضیح دهید. فایل صوتی خروجی هر مرحله را ضمیمه گزارش خود نمایید. توجه داشته باشید که برای جمع دو فایل صوتی ابتدا باید فرکانس نمونه برداری هر دو را یکسان نمایید.

Problem 2:

In this problem, we will take a practical approach toward the subject of Fourier transform. What we have discussed so far is the Fourier transform of continuous-time signals, which is only available in analog devices, and when it comes to computers, there is no way to truly define continuous-time signals as everything in computers is digitized (considering you are not using symbolic programming, which is another thing and its computational power is limited). In light of this, python and other libraries in other languages implement something called the Discrete Fourier Transform (DFT). Consider a vector $x[n]$ with length N , then its DFT is another vector, $X[k]$ with the same length N and is defined as:

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

Now, answer/implement the following questions:

1. With respect to the given equation for DFT, find the equation for IDFT (Inverse DFT).



2. Implement a function called **fft** that takes a vector as input and produces its DFT vector as output.
3. Implement the function for IDFT called **ifft**.
4. Consider the signal $x(t) = \cos(\pi t)$. Construct the vector $x[n] = x(nT)$ with length N . Plot this vector for $T = \frac{1}{2}$ and $N = 1000$ for n ranging from -500 to 500.
 - Obviously, when trying to reference the vector's elements, $x[n]$, one should use a valid index ranging from 0 to $N - 1$ (for python).
5. Compute Fourier transform of $x(t)$.
6. Use **fft** to compute DFT of $x[n]$, $X[k]$. Plot $X[k]$. Compare the result with 5.
7. Use **ifft** on $X[k]$. Is the resulting vector equal to $x[n]$?
8. Show the effects of changing N and T on the DFT of $x[n]$, in different plots.

Problem 3: In this question, we want to cancel unwanted effect of a system on a sound. This action is called *channel equalization*. For such purpose, we have *clean1.wav* and *distorted1.wav* as input and output signals of this system. Clean1 is a signal without any distortion and distorted1 is the output of the system which has been distorted by the system. *distorted2.wav* is another signal that has been the output of this system. The problem is that we do not have access to the clean2 signal. Using *clean1.wav* and *distorted1.wav* signals, first calculate the frequency response of the system. Then try to recover *clean2* signal using system's inverse response. Discuss your methodology and attach the recovered signal named *recovered2.wav*

Problem 4:

In this question we want to recover a noisy signal by applying a suitable filter (*signal de-noising*). Consider the *noisy1.wav* signal which is made from a clear sound added with some noise. You should design an appropriate filter and apply it to the noisy signal. In order to find the frequency of the noise and attenuate it in the filter, you should first examine the signal's spectrum and detect the speech gaps. In these gaps you can find the noise frequency. Then, design a filter which attenuates these frequencies. Explain your methodology and attach the final de-noised signal named *denoise1.wav* to your report.

سوال ۵:

برای ترسیم نمودار *Bode* در پایتون می‌توانید از بسته *control* استفاده کنید. نحوه نصب و استفاده از این پکیج را می‌توانید [اینجا](#) بیابید. برای پاسخ فرکانسی زیر نمودار *Bode* را یکبار با استفاده و یکبار بدون استفاده از پکیج *control* ترسیم کنید. قطبها و صفرها را با استفاده از بسته *control* محاسبه کنید و در خصوص نمودارهای ترسیم شده بحث کنید.

$$H(j\omega) = \frac{(j\omega)^2 + 2j\omega + 10}{(j\omega)^3 + 25j\omega + 1.9 * 10^6 j\omega + 10}$$