Term Project

Signals and Systems for Computer Engineering

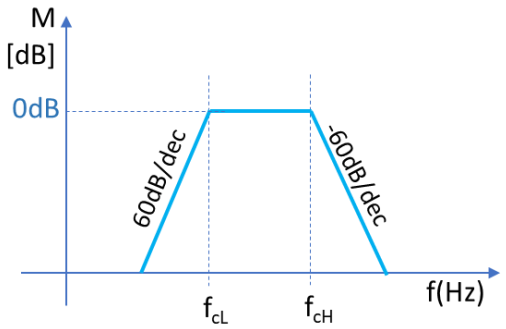
Assigned: May 17, 2021

Due: May 31, 2021, **23:59** (submit through Ninova)

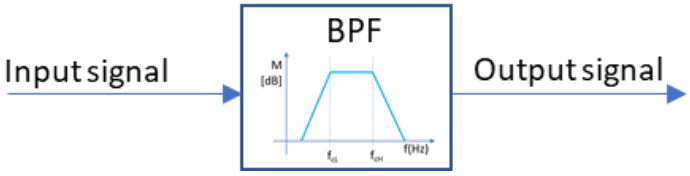
- Your submission must include your code and report.
- In the report, briefly explain your solution and include your plots (if applicable).
- Use only the Python programming language.
- Please **write your full name** (first name and last name) and **Student ID** at the top of your solution.
- Please show ALL work. Answers with no supporting explanations or work will not receive any partial credit. Your homework is not just a final report of your results; we want to see your steps. Upload all your steps to get to the solution.
- Project is individual.
- Do not copy & paste anything from anywhere. Use your own words.
- **No late submissions** will be accepted. Do not send your solutions by e-mail. We will only accept files that have been uploaded to the Ninova e-learning system before the deadline. Do not risk leaving your submission to the last few minutes.
- If you have any questions regarding the assignment, please e-mail **Enes Albay (albay@itu.edu.tr)**.

Two audio record samples (music) are provided in the attachment as ".wav" files. Sampling rate of these 16bit mono files is 44100Hz.

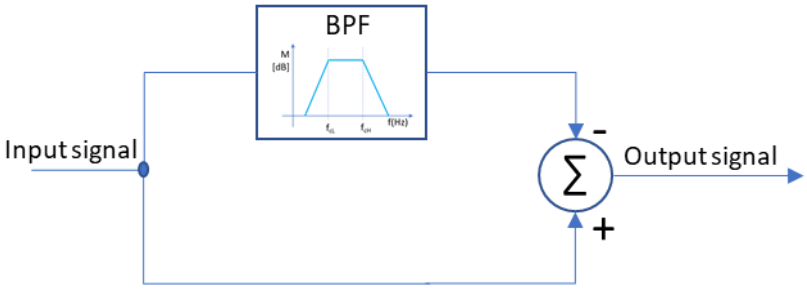
a) Write a pseudo code that performs the below given (asymptotical) Band Pass Filter (BPF) function.



b) Perform this BPF for f_{cL}=400Hz and f_{cH}=3000Hz by using Python and write the output data into a file by adding BPF to its name. (for example AfricaBPF.wav). Please provide the Python code that include comments and the filtered wave files (Do not use external library for the BPF part).



c) Subtract the band pass filtered audio data in "b" from the original audio data as shown in the below figure and get a Band stop filtered (BSP or Notch Filter) audio stream; write the output into a file by adding BSP to its name (e.g. AfricaBSP.wav).



d) Design a graphical user interface that enables choosing a band to be passed or stopped by using Python code (Use PyQt library for GUI). Central frequency $f_c=(f_{cL}+f_{cH})/2$ will be entered between 200Hz to 7000Hz and the Bandwidth (BW with respect to -3dB attenuation) will be chosen as integer percentage between 5%..50% of the f_c . (For example if f_c=500Hz and BW=20% then 400Hz to 600Hz will be the pass or stop frequency range with respect to -3dB criteria). There will be a logical checkbox to select code as "BPF" (default) or "BSP". Audio file to be played and/or filtered will be chosen from the selected directory from the files having .wav extension. Filtered audio stream will be played when a play button is pressed. When "save" button will be pressed, filtered file will be recorded by adding B to its original name (such as AfricaB.wav). Provide the code that includes comments (Do not use external library for the filtering part).