Department of Electrical and Computer Engineering
The University of Alabama in Huntsville
Spring 2020

CPE 381: Fundamentals of Signals and Systems for Computer Engineers

Programming Assignment: Real-time Signal Processing

Phase I due: Wednesday, March 4, 2020 at 9:35 am – (8% of the grade)
Phase II due: Wednesday, April 15, 2020 at 9:35 am – (12% of the grade)
Bonus Assignment (optional) due: Wednesday, April 22, 2020 at 5 pm
(bonus 4% of the grade)

Please bring hardcopy of the report to the class and
upload a single ZIP file with all required files to Canvas

Write a program in C/C++ to perform real-time audio signal processing. The program will read audio record from WAV file, perform the processing sample-by-sample, and store the result back to the new WAV file. All necessary functions must be implemented, external libraries are not allowed, except for the Bonus Assignment.

Phase I (40 points, due: Wednesday, March 4, 2020 at 9:35 am)

Please follow the following steps:

1. Record your own voice in wave file (.WAV). The record must start with your introduction of yourself, followed with additional content (e.g. music). The record length must be 16 s< T_{rec} <18 s.
2. Read WAV file in Matlab and replace signal from time *t*=10sec to the end of the file with the sine wave *f*=2,500 Hz and amplitude equal to 10% of the maximum possible amplitude of the signal, and initial phase of $\pi/6$. Save modified WAV file. When you play the file you will be able to hear your original recording for 10 seconds followed by annoying sine wave @2,500 Hz for the rest of the file.
3. Write C/C++ program to read the file from previous step and write modified WAV file. Program should process file sample by sample, and write the output at time *t* as

   $$out(t) = in(t) + 0.1*n(t) + 0.05*rn(t)$$

   where:

   - *in(t)* is input from WAV file at time *t,*
   - *n(t)* is added "noise" sine wave with amplitude equal to the maximum amplitude of the signal and frequency of 2,000 Hz (check size of your sample to determine maximum amplitude of the signal), and
   - *rn(t)* is random noise.

   Please note that the processing may create overflow that you must compensate for. Measure the performance of the program (*end-time – start_time*) with minimum resolution of 1 ms.
4. Write another summary text file that outputs: filename, the sampling frequency, number of channels, number of bits per sample, record length in seconds calculated from the number of samples, maximum absolute value of the sample in input file, channel that has the maximum absolute value of the sample, and execution time of the program.

Prepare Report and submit both hard and soft copy of the report. The report must contain the following numbered sections:

1. Short description of the problem and proposed solution.
2. Short description of WAV file format. What control fields do you have in the header? What is their location?
3. Summary text file. Explain if your program can work in real-time.

Deliverables

- Complete source code of your project (MS Visual Studio or Linux gcc/g++ with all files necessary to compile the project & make file in the single ZIP file).
    - Projects that can not be compiled and run in either environment will lose 40% of the grade. Only original and independent work will be graded. Plagiarized work will not receive any credit.
- PDF or DOC/DOCX file of your Report with the following format of the filename: Lastname_first-initial.DOC or PDF
- Summary file in the following format: LastName_FirstInitial_sum.txt
- Sound files in the following format (use your last name and first initial):
    - LastName_FirstInitial_orig.WAV - original record
    - LastName_FirstInitial_mod.WAV - processed record

Phase II (60 points, due: Wednesday, April 15, 2020 at 9:35 am)

Please follow the following steps:

1. Use wave file from Phase I.
2. Read WAV file in Matlab and find the dominant spectral component in the signal (frequency and amplitude) for t<10s and t>10s.
3. Design a filter to eliminate added sine wave @2,000 Hz from Phase I with minimum attenuation of 60dB. Pay attention to trade-off between real-time performance and the quality of the filter. Describe implementation of your filter. Discuss your decisions in the report.
4. Write C/C++ program to read the file, check the sampling frequency from the file header and select appropriate filter for that frequency. It is acceptable to assume that you will have only two standard sampling frequencies (Fs), but you must process the signal according to the sampling frequency of the record. Measure the performance of the program (*end-time – start_time*).

Prepare Report and submit hard and soft copy of the report. The report must contain the following numbered sections:

1. Short description of the problem and proposed solution.
2. Documented design of the filter in Matlab that includes
    a. Plot of filter characteristics (magnitude and phase) for both filters/sampling frequencies
    b. Type of filter (FIR/IIR) and reasons for choosing that particular filter.
    c. Organization of processing; how do you perform your processing in C/C++.
3. Spectrum of the input WAV file and spectrum of the processed file for t<10s and t>10s.
4. Performance of the program (execution time). Explain if your program can work in real-time.
5. Short description of your experience and “lessons learned”

Deliverables

- Complete source code of your project (MS Visual Studio or Linux gcc/g++ with all files necessary to compile the project & make file in the single ZIP file).
- Projects that can not be compiled and run in either environment will lose 40% of the grade. Only original and independent work will be graded. Plagiarized work will not receive any credit.
- PDF or DOC file of your Report with the following format of the filename: LastName_FirstInitial.DOC or PDF
- Matlab script used to prepare the assignment.
- Sound files in the following format (use your last name and first initial):
    - o LastName_FirstInitial_orig.WAV - original record
    - o LastName_FirstInitial_mod.WAV - modified record
    - o LastName_FirstInitial_filt.WAV - filtered output

Bonus Assignment (extra 20% of the assignment grade, or 4% of your overall course grade)
due: Wednesday, April 22, 2020 at 5 pm)

Write a C/C++ program to determine the dominant spectral component in WAV file using FFT analysis and total power of the signal (RMS value) for each one second window with 50% window overlap. Save CSV text file with “LastName_FirstInitial_bonus.csv”. The file must have first line representing names of variables: “time, fmax,power” followed with results in each window.

Full credit (20% bonus) can be received if you provide the same documentation of the design and performance of the additional solution.