Department of Electrical and Computer Engineering
The University of Alabama in Huntsville
Spring 2023

CPE 381: Fundamentals of Signals and Systems for Computer Engineers

Programming Assignment: Real-time Signal Processing

Phase I due: Wednesday, March 8, 2023 at 9:35 am – (8% of the grade)
Phase II due: Monday, April 17, 2023 at 9:35 am – (12% of the grade)
Bonus Assignment (optional) due: Monday, April 24, 2023 at 5 pm
(bonus 4% of the grade)

Please 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 8, 2023 at 9:35 am at 9:35 am)

Please follow the following steps:

1. Record your own voice in wave file (.WAV) as stereo file with sampling frequency Fs=11,025Hz. You can use Audacity or similar program. The record must start with your introduction of yourself, followed with additional content (e.g. music).

   The record length must be $T_{rec} = 14$ s.
2. Read WAV file in Matlab and replace signal from time *t*=7sec as follows:
   - Left channel will have sine wave at *f*=1,810 Hz, amplitude equal to 0.3, and initial phase of $\pi/4$.
   - Right channel will have sine wave at *f*=1,990 Hz, amplitude equal to 0.25, and initial phase of $3\pi/4$.

   Save modified WAV file. When you play the file you will be able to hear your original recording for 7 seconds followed by annoying sound around 1,900 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

   $$out(t) = in(t) + 300*rn(t) - 150$$

   where:
   - *in(t)* is input from WAV file at time *t,*
   - *rn(t)* is random noise with amplitude [0..1]).

   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. The program must ask user for the WAV filename or read it from the command line. Hardcoded file names are not allowed.

5. Write separate summary text file that outputs in this order:
    - your name,
    - filename,
    - number of channels,
    - the sampling frequency,
    - number of bits per sample,
    - record length in seconds calculated from the number of samples,
    - maximum absolute value of the sample in each channel, and
    - the execution time of the program.

Prepare Report and submit both hard and soft copy of the report. The report must be generated as PDF file and 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. The content of the summary text file.
4. Real-time operation. Discuss 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 cannot 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.
    - Program that does not follow all requirements will not receive any credit.
- PDF file of your Report with the following format of the filename: *Last_first-initial.PDF*
- Summary file in the following format: *Last_first-initial_sum.txt*
- Sound files in the following format (use your last name and first initial):
    - *Last_first-initial_orig.WAV* - original record
    - *Last_first-initial_mod.WAV* - processed record

Phase II (60 points, due: Monday, April 17, 2023 at 9:35 am at 9:35 am)

Please follow the following steps:

1. Use wave file *Last_first-initial_mod.WAV* from Phase I.
2. Read WAV file in Matlab and find the dominant spectral component in the signal (frequency and amplitude) for t<8s and t>8s.
3. Design a filter to eliminate added sine wave around 1,900 Hz from Phase I with minimum attenuation of 80dB for signal sampled at Fs=11,025 Hz. 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. Design filter with similar characteristics for signal sampled at Fs=22,050 Hz.
5. 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) described above, 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<7s and t>7s.
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.
    - Program that does not follow all requirements will not receive any credit.
    - PDF file of your Report with the following format of the filename: *Last_first-initial.PDF*
- Matlab script used to prepare the assignment.
- Sound files in the following format (use your last name and first initial):
    - *Last_first-initial_orig.WAV* - original record
    - *Last_first-initial_mod.WAV* - modified record
    - *Last_first-initial_filt.WAV* - filtered output

Bonus Assignment (extra 20% of the assignment grade, or 4% of your overall course grade)
due: Monday, April 24, 2023)

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 NFFT=2,048 and NFFT=4,096 with 25% window overlap using Hanning window. Test the program using modified wave file from your project. Save CSV text file with "*Last_first-initial_bonus.csv*". The file must have first line representing names of variables: "time, fmax, power" followed with results calculated for each time window.

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