Due Date: 31/5/2019 **Spring 2019**

EE 497 Real-time Applications of Digital Signal Processing

PROJECT

Implementation of Overlap-and-Save Algorithm in myRIO

This project is composed of two parts. In the first part, you will implement the Overlap-and-Save (OLS) method in MATLAB. In the second part, you will implement the same code in C and run it in myRIO.

1st Part (Due: May 3, 2019): (%25 of the total grade)

In the MATLAB implementation, you will use an input signal x[n] with length P=1024, a filter h[n] with length M=16, and block size L=128. You will use L=128 point fft to implement the OLS method. Note that P,M and L are variables and can be changed arbitrarily. You can generate random input and filter sequences. The result of the OLS implementation will the convolution of x[n] and h[n]. Note that the time sequence is obtained through fft and ifft operations. Then you will compare your result with the "conv" command result in MATLAB. You will send your MATLAB code to your assistant by e-mail before due date.

2nd Part (Due: May 31, 2019): (%75 of the total grade)

In this part of the project, you will run a C program in myRIO. The function of the program is to implement an FIR filter and obtain the filter output to display it on the user interface. The user interface will be used to enter the length of the FIR filter and the coefficient values. Input signal is selected by the user from the built-in waveforms, such as sine, triangular, chirp waveforms. The processing and filtering should be continuous so that there is no missing samples or blocking artifacts. In order to achieve this, you will implement "**Overlap-and-Save**" method by employing the FFT algorithm. This is described in detail in [1]. Filtering code should be written as a custom C code and no library function can be used for this purpose except the fft() function. The main function of the LabVIEW program in this project is to call a shared library which is coded in C and generate the user interface. Since the program is to be run on myRIO, it should be compiled by Eclipse. Hence you would need to compile a shared object file, or .so file since myRIO has NI Linux Real-Time operating system. Once you have obtained the .so file in your computer, you will call it in LabVIEW by appropriately defining the parameters, inputs, etc. Since LabVIEW has a special calling convention and supports limited libraries, it is more reasonable that you develop your C code in *Labwindows C* environment and migrate the code (copy as it is) to Eclipse and compile to get your .so shared library file. The details of running a C code under LabVIEW is given in the following references. Once you have the .so file you should place it in myRIO in order to call it. You will submit your written project report together with your MATLAB, C and LabVIEW codes.

1) Integrating C Code with LabVIEW on NI Linux Real-Time Targets

http://www.ni.com/tutorial/14690/en/

2) Getting Started with C/C++ Development Tools for NI Linux Real-Time, Eclipse Edition.

http://www.ni.com/tutorial/14625/en/

3)https://www.youtube.com/watch?v=CdG--UMpNhg Video.

4) An Overview of Accessing DLLs or Shared Libraries from LabVIEW-DevZone Article

5) Example 1: Call a Shared Library That You Built

http://zone.ni.com/reference/en-XX/help/371361L-01/lvexcodeconcepts/ex_1_call_a_shared_library/

6) LabWindow/CVI Programmer Reference Manual

7) Using LabWindows/CVI DLLs in LabVIEW Real-Time ... - LabVIEW360

[1] Alan V. Oppenheim, Ronald W. Schafer, "Discrete-time Signal Processing", Pearson Education

Note: The project demo will be scheduled for each group. You will bring your printed project report at the project demo session and send your complete project through e-mail, etuncer@metu.edu.tr.