J. Reilly, Last revision Oct. 2014.

CoE 4TL4 Lab 2
FIR filter implementation on the TMS 320 DSP processor

This lab is structured similarly to Lab 1. The lab will nominally begin **Monday Oct 20, 2014**. Your report is due 3 weeks hence on **Mon Nov 10**. Work in pairs and submit one report per pair. Submit in the CoE4TL4 slot in the ITB copy room. Late penalty = 10% per day, up to a maximum of 50%.

1 Overview

Two matlab .mat speech files (SPM1.mat (male speaker), and SPF1.mat (female speaker)) are available for downloading from the course website (www.ece.mcmaster.ca/faculty/reilly/coe4tl4). Download the file onto your working directory, then issue the command load fname within matlab. The variable “speech” will then contain the speech samples.

Additional matlab commands that may be useful are ”hamming”, ”blackman”, ”hanning”, etc., that provide the respective window function. The command ”fdatool” gives a comprehensive filter design and analysis tool– you will find this useful, but in your reports, please show that you know how to design and analyze filters from scratch, without using this package. Two other useful matlab process for filter visualization are “fvtool” and “freqz”.

Note that the default sampling frequency of the DSP devices is 24KHz. Note also that the speech generated by Matlab that is output from the computers is bandlimited to 4 KHz. So any filtering done on the DSP processors should only involve frequencies in the range $0 \leq f \leq 4$KHz.

The DSP processors work on an interupt basis. That is, upon the arrival of each new sample, the computer executes the filter routine. The maximum number of filter coefficients the processor can handle in one interupt cycle is about 110. So, in your experiments, keep the number of coefficients below this number.

2 Procedure

Here are a few suggestions. You are welcome to try out any additional experiments you can think of.

Do all your design work (i.e., generating the coefficients) using Matlab. You will be interested to know that the Matlab package “FDAtool” is very useful for filter design. You are welcome to use this package in this lab, but I would like at least one example of an FIR design using the windowing procedure (i.e., without using FDAtool).

Please show your finished DSP implementation to a TA to verify you have successfully implemented the exercises below.

1. There are several steps required to compile, link and load the code for the DSP processors. These steps are carefully outlined in the document *Configuring CCS5* provided by Tyler Ackland, available on the Y-drive in the lab.

   Please also refer to the videos by our TA Mostafa Medra which are posted on the website. They are very helpful in getting the DSP processors up and running.

2. Connect the function generator into the DSP board input, and the DSP board output into the scope.

   As a first step, it is recommended you initialize and run the program ”pass through”, which is indicated in the Config document. This program simply reads in values from the function generator into the DSP, and then immediately writes them back out again, to display on the scope. No filter coefficients are required with this program. The fact you are able to get this program running means that compilation and linking process has been completed correctly and everything is set up properly.

3. Implement an FIR filter of your choice using Matlab. Save the coefficients in ascii format– the matlab command “dlmwrite” is very useful for this purpose. It saves the coefficients in a .txt file as a row vector, separated by commas. The coefficients are included in the C program by appending the following statements at the end of your C program: (here ’num’ is the number of coefficients)

   define BL num
   double B[num] = { list the num coefficient values in fractional or exponential format; e.g., 0.0004568938 or 1.21334908e-17, separated by commas. This may be done by cutting and pasting from the txt file above.};

   Evaluate the magnitude and phase responses of the filter using Matlab. Compare the Matlab responses with what you actually observe on the output of the processors. Run one of the speech files from lab 1 through the filter (by connecting the audio output of the computer into the DSP board) and comment on the results.

4. Add a sinusoidal tone at a frequency of your choice to one of the speech files you used in Lab 1, at a level 20 dB above the level of the speech. Design a bandstop FIR filter whose stopband includes the frequency of your sinusoid, and load it into the processor. Run the program and comment on the degree of suppression of the tone. In your report, explain how you calculated the level (amplitude) of the tone.