ECE4703 B Term 2006 Project 3

Signoff due by 1:50pm on 15-Nov-2006
Report due by 4:00pm on 16-Nov-2006

The goals of this laboratory assignment are:

- to familiarize you with infinite impulse response (IIR) filtering on the TMS320C6713 and
- to demonstrate the effects of coefficient quantization and realization structure on the performance and stability of IIR filters.

1 Problem Statement

In this assignment, you will write a C program for the TMS320C6713 DSK that realizes the same filters as in Project 2 except that the filter type will be an *elliptic IIR* filter. The filter design parameters are as follows:

1. Your filters must be elliptic IIR.
2. Your sampling rate must be 8kHz.
3. You must use interrupts for all I/O.
4. The lowpass and highpass filters must meet the same frequency specifications as in Project 2.
5. You should design the filters to have minumum order.

The filters' structure and quantization parameters will be specified in the sections below.

2 Part 1: Floating Point Design and Implementation

Using the Matlab filter design tools, design ellitpic IIR lowpass and highpass filters that satisfy the requirements. Make sure you set the sampling frequency for your design to 8kHz. First realize the filters as a "Single Section Direct Form II" filter and generate a C header with single-precision `float` coefficients. Write your code to realize the filters in this structure. Keep all intermediate results in `float` format until output by the codec. Test your filters and confirm that their magnitude responses agree with Matlab's predictions. Profile the number of execution cycles needed to implement your filters and compare the results to the results you obtained for the filters implemented in Project 2.

Next, convert both filters to "Direct Form II - Second Order Sections" and generate a C header with single-precision `float` coefficients. Write new code to realize the filters in this structure.

The only difference with respect to the prior filters is that the realization structure has changed. Matlab will automatically compute new filter coefficients for this realization structure. Note that each second order section, in addition to the filter coefficients, has a scaling factor that will need to be implemented in order to achieve the correct frequency response. Use the “Show Filter Structure” option in Matlab to see how this scale factor should be implemented. As before, keep all intermediate results in `float` format until output by the codec. Test your filters and confirm that their magnitude response agrees with Matlab’s predictions. Profile the number of execution cycles needed to implement these filters (in this structure) and compare the results to the single-section versions of the IIR filters.

Test your filters on DTMF tones to ensure that they are working as expected. Make sure you have these filters working correctly before proceeding to the next part of the project.

3 Part 2: 16-bit Fixed-Point Design and Implementation

In this part, you will repeat your filter designs from Part 1 except that the filters will be realized entirely with fixed-point math. All intermediate results can be stored in 32-bit signed integers but the final result will need to be converted to a 16-bit signed integer before output by the AIC23. No floating point data or math operations are permitted here.

Begin by setting the filter structure to “Single Section Direct Form II” and designing your filters with double-precision floating point coefficients (in Matlab’s `fdatool`). Then press the “set quantization parameters” button (or, alternatively, write your own quantization script) and convert all of the filter coefficients to 16-bit signed format. Note that the “coefficient word length” setting in `fdatool` should be set to 15 because one bit is required for the sign. You can let Matlab pick the best precision fractional lengths for the numerator and denominator. Examine the performance of the fixed-point realization of this filter and note any discrepancies with respect to the double-precision case. Export the 16-bit filter coefficients for use in CCS and write the C code needed to implement your filters in fixed-point under this realization structure. Test your filters and confirm that their magnitude response agrees with Matlab’s predictions. Profile the number of execution cycles needed to implement your filters and compare the results to the other results you have obtained so far.

Next, set everything back to double-precision floating point format and redesign your filters in “Direct Form II - Second Order Sections”. This is an important step because Matlab will not do the conversion correctly if you leave your filter coefficients in fixed-point and do the conversion directly. Make sure you go back to floating point and change the filter structure from there. Once the proper realization structure has been selected and the filter coefficients have been calculated, press the “set quantization parameters” and convert all of the filter coefficients to 16-bit signed format (15-bit coefficient word length). Have Matlab generate the appropriate coefficient header files and write the C code needed to implement your filters in fixed-point under this realization structure. Test and profile the filters as before.

As with fixed-point FIR filtering, it is important to be careful about scaling with fixed-point implementations of IIR filters. This is even more critical when dealing with IIR filters realized as cascaded second order sections. Your filters in this part of the assignment should have magnitude responses that are virtually indistinguishable from the floating point implementation.

4 Part 3: 8-bit Fixed-Point Design and Implementation

Repeat the last part of the assignment except quantize all coefficients to 8-bit fixed point precision. Note that the "coefficient word length" setting in `fdatool` should be set to 7 because one bit is required for the sign. You can let Matlab pick the best precision fractional lengths for the numerator and denominator. Measure the frequency response and note any changes in the filters' behavior. Does Matlab predict any difference between "single section" and "second order sections"? Are you able to see these differences in your real-time implementation?

Using FDA tool, see how coarsely you can quantize your filter coefficients (in both "single stage" and "second order sections" realizations) before the filters become unstable. You don't need to implement these filters but you should determine the number of bits at which coefficient quantization causes the filter to become unstable.

5 Additional Remarks

Make sure you are clear on the signal flow of Direct Form II before writing any code. Recall that you will not be buffering the input signal $x[n]$ in this realization form. Rather, you will be calculating and buffering the intermediate signal $u[n]$ in this realization form. If you are unclear on this, please ask the TA or the Instructor for help.

You may want to structure your code with functions to simplify the development and troubleshooting. There are some good examples of IIR filtering in your textbook (Lab 5, Part 2) and in Chassaing's example code (in the myproject directory). When dealing with second order sections, it is possible to write one small function that computes the output of a second order section. You can call this function as many times as necessary to compute the overall output of the filter.

Please structure your work to facilitate a smooth signoff, preferably without rebuilding several projects.

6 In Lab

You will work with the same lab partner(s) as in the prior laboratory assignments. Please contact the instructor if your lab partner has dropped the course or if you have concerns about your lab partner's performance on the prior assignment.

7 Laboratory Report and Grading

See Project 2.

7.1 Specific Items to Discuss in Your Report

The emphasis of this assignment is similar to Project 2 with the addition of considering two different realization forms as well as the sensitivity of IIR filter performance (and stability) to the choice of realization structure and coefficient quantization. Discuss any insight gained from the experimental results and any discrepancies observed with respect to theoretical predictions. Also discuss the advantages and disadvantages of the IIR filters considered in this assignment with respect to the FIR filters in Project 2. How do your IIR profiling results compare to your FIR profiling results? Under what circumstances would you use an IIR filter? When would an FIR filter be better?