

# LAB 3: PRELAB (PART 2)\*

Thomas Shen  
Douglas L. Jones

Based on *IIR Filtering: Filter-Coefficient Quantization Exercise in MATLAB*<sup>†</sup> by

Douglas L. Jones  
Swaroop Appadwedula  
Matthew Berry  
Mark Haun  
Jake Janovetz  
Michael Kramer  
Dima Moussa  
Daniel Sachs  
Brian Wade

This work is produced by OpenStax-CNX and licensed under the  
Creative Commons Attribution License 2.0<sup>‡</sup>

## Abstract

You will design a fourth-order notch filter and investigate the effects of filter-coefficient quantization. You will compare the response of the filter having unquantized coefficients with that of a filter having coefficients quantized as a single, fourth-order stage and with that of a filter having coefficients quantized as a cascade of two, second-order stages.

## 1 Filter-Coefficient Quantization

One important issue that must be considered when IIR filters are implemented on a fixed-point processor is that the filter coefficients that are actually used are quantized from the "exact" (high-precision floating point) values computed by MATLAB. Although quantization was not a concern when we worked with FIR filters, it can cause significant deviations from the expected response of an IIR filter.

By default, MATLAB uses 64-bit floating point numbers in all of its computation. These floating point numbers can typically represent 15-16 digits of precision, far more than the DSP can represent internally. For

---

\*Version 1.1: Aug 20, 2006 10:20 pm -0500

<sup>†</sup><http://cnx.org/content/m10813/2.5/>

<sup>‡</sup><http://creativecommons.org/licenses/by/2.0/>

this reason, when creating filters in MATLAB, we can generally regard the precision as "infinite," because it is high enough for any reasonable task.

NOTE: Not all IIR filters are necessarily "reasonable"!

The DSP, on the other hand, operates using 16-bit fixed-point numbers in the range of -1.0 to  $1.0 - 2^{-15}$ . This gives the DSP only 4-5 digits of precision and only if the input is properly scaled to occupy the full range from -1 to 1.

For this section exercise, you will examine how this difference in precision affects a **notch filter** generated using the `butter` command: `[B,A] = butter(2,[0.07 0.10], 'stop')`.

### 1.1 Quantizing coefficients in MATLAB

It is not difficult to use MATLAB to **quantize** the filter coefficients to the 16-bit precision used on the DSP. To do this, first take each vector of filter coefficients (that is, the *A* and *B* vectors) and divide by the smallest power of two such that the resulting absolute value of the largest filter coefficient is less than or equal to one. This is an easy but fairly reasonable approximation of how numbers outside the range of -1 to 1 are actually handled on the DSP.

Next, quantize the resulting vectors to 16 bits of precision by first multiplying them by  $2^{15} = 32768$ , rounding to the nearest integer (use `round`), and then dividing the resulting vectors by 32768. Then multiply the resulting numbers, which will be in the range of -1 to 1, back by the power of two that you divided out.

### 1.2 Effects of quantization

Explore the effects of quantization by quantizing the filter coefficients for the notch filter. Use the `freqz` command to compare the response of the unquantized filter with two quantized versions: first, quantize the entire fourth-order filter at once, and second, quantize the second-order ("bi-quad") sections separately and recombine the resulting quantized sections using the `conv` function. Compare the response of the unquantized filter and the two quantized versions. Which one is "better?" Why do we always implement IIR filters using second-order sections instead of implementing fourth (or higher) order filters directly?

Be sure to create graphs showing the difference between the filter responses of the unquantized notch filter, the notch filter quantized as a single fourth-order section, and the notch filter quantized as two second-order sections. Save the MATLAB code you use to generate these graphs, and be prepared to reproduce and explain the graphs as part of your quiz. Make sure that in your comparisons, you rescale the resulting filters to ensure that the response is unity (one) at frequencies far outside the notch.