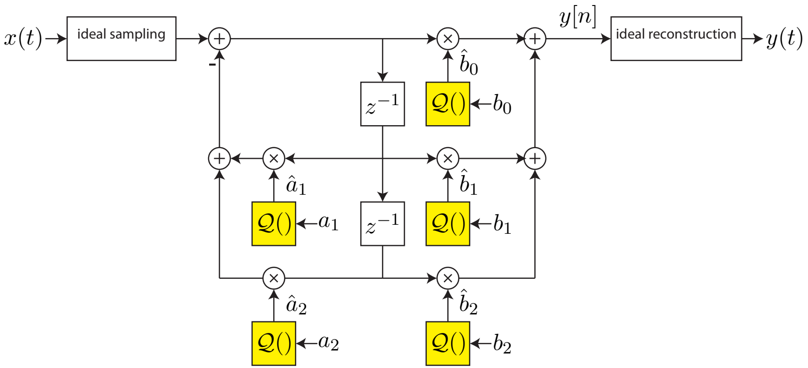
Digital Signal Processing
Effect of Coefficient Quantization on IIR Filters

D. Richard Brown III

IIR Filter Coefficient Quantization

For a direct-form II IIR filter, we have



with $\hat{b}_n = b_n + \Delta b_n$ and $\hat{a}_n = a_n + \Delta a_n$. The feedback in the system prevents us from expressing the quantized transfer function as $\hat{H}(z) = H(z) + E(z)$ we did with FIR filters. Analytical techniques, e.g., pole-displacement sensitivity analysis, can be used but we can get some intuition from examples...

Generate Unquantized IIR Filter Coefficients

```
% ---------------------------------------------------------------
% generate 8th order IIR LPF
% ---------------------------------------------------------------

Fs = 48000;  % Sampling Frequency

Fpass = 9600;     % Passband Frequency
Fstop = 12000;    % Stopband Frequency
Apass = 1;        % Passband Ripple (dB)
Astop = 80;       % Stopband Attenuation (dB)
match = 'both';   % Band to match exactly

% Construct an FDESIGN object and call its ELLIP method.
h  = fdesign.lowpass(Fpass, Fstop, Apass, Astop, Fs);
Hd = design(h, 'ellip', 'MatchExactly', match);
% Get the transfer function values.
[b, a] = tf(Hd);
[H,w] = freqz(b,a,1024);
```

Quantize the Filter Coefficients

```
% first determine the number of non-frac bits needed to quantize the
% numerator and denominator
nonfraca = ceil(log2(max(abs(a(2:end)))));
nonfracb = ceil(log2(max(abs(b))));

% quantize coefficients (we know there won't be overflow)
B = 7;                    % we have B+1 total bits, including sign
qa = B-nonfraca;          % fractional bits for denominator
qb = B-nonfracb;          % fractional bits for numerator
ahat = round(a*2^qa)/2^qa;  % quantized denominator
bhat = round(b*2^qb)/2^qb;  % quantized numerator

% compute quantized coefficients frequency response
[Hhat,what] = freqz(bhat,ahat,1024);
```

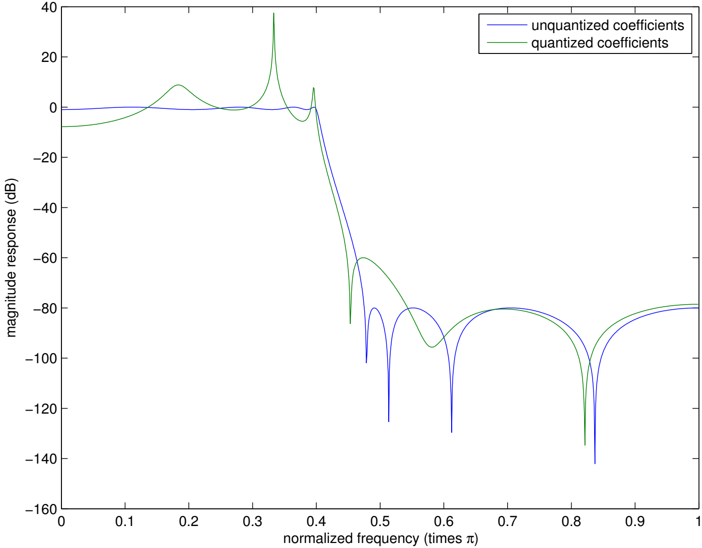
Quantized IIR Filter Coefficients

In this example, we have $H(z) = \frac{b_0+b_1z^{-1}+\cdots+b_8z^{-8}}{1+a_1z^{-1}+\cdots+a_8z^{-8}}$ with unquantized and 8-bit quantized coefficients:

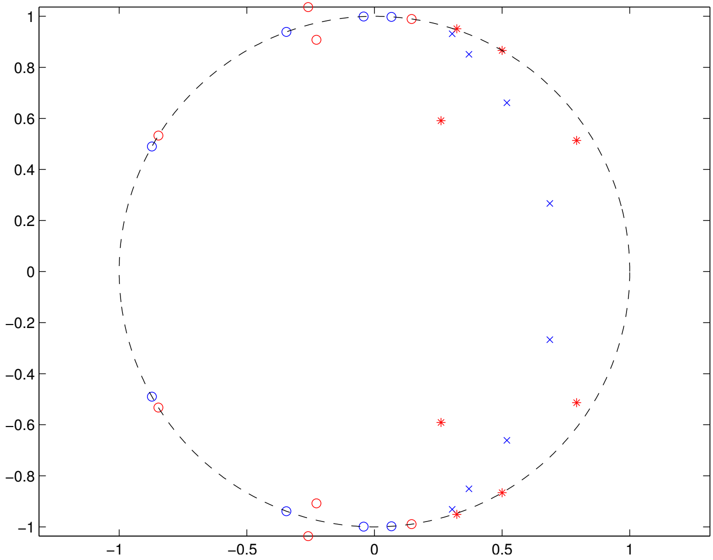
i	a_i	$\hat{a}_i$	b_i	$\hat{b}_i$
0	1.0000	1.0000	0.0039	0.0039
1	-3.7597	-3.7500	0.0093	0.0093
2	8.1976	8.2500	0.0198	0.0200
3	-11.8524	-11.8750	0.0276	0.0273
4	12.3314	12.3750	0.0317	0.0317
5	-9.2974	-9.2500	0.0276	0.0273
6	4.9767	5.0000	0.0198	0.0200
7	-1.7419	-1.7500	0.0093	0.0093
8	0.3172	0.3750	0.0039	0.0039

The numerator coefficients have $q_b = 11$ fractional bits and the denominator coefficients have $q_a = 3$ fractional bits. Dynamic range of denominator coefficients: $\frac{12.3314}{0.3172} \approx 38.9$.

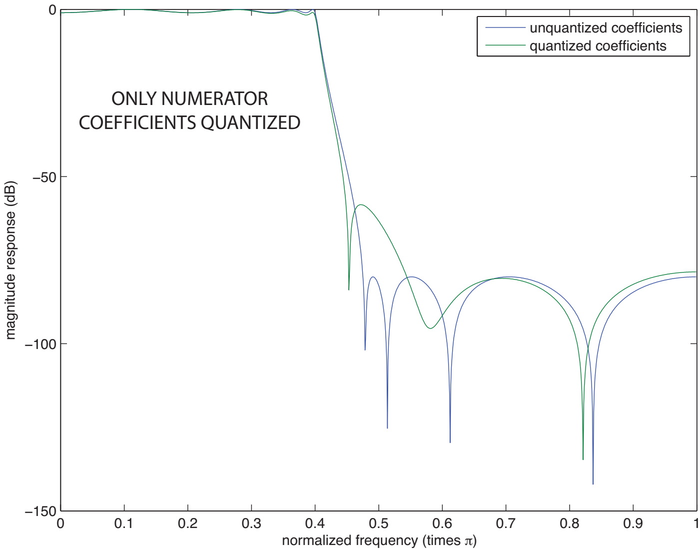
IIR Coefficient Quantization Example (8th order DF-II)



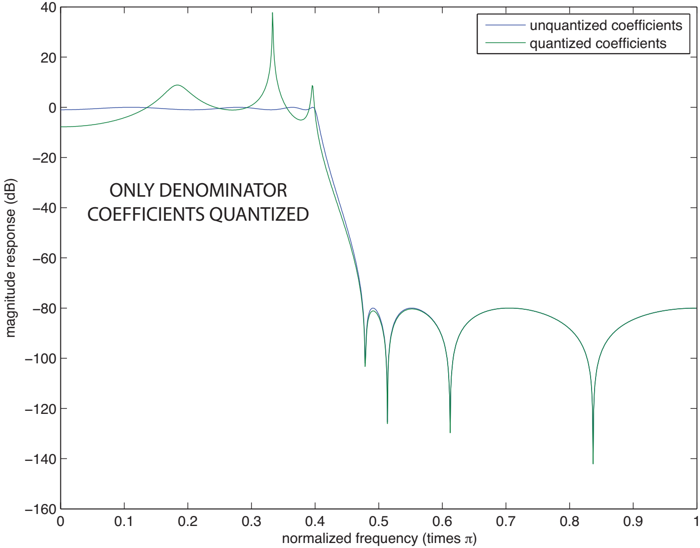
IIR Coefficient Quantization Example (8th order DF-II)



IIR Coefficient Quantization Example (8th order DF-II)



IIR Coefficient Quantization Example (8th order DF-II)



IIR Filter Coefficient Quantization Remarks/Observations

1. Previous examples all used 8-bit coefficient quantization and a single 8th order DF-II section realization structure.
2. Each quantized numerator coefficient changes **all** of the zeros.
3. Each quantized denominator coefficient changes **all** of the poles.
4. IIR filter response is often quite sensitive to denominator coefficient quantization. In fact, denominator coefficient quantization can cause an IIR filter to become unstable.
5. A cascaded second order sections (SOS) realization is usually preferred with finite precision coefficients because:
   - we can quantize the coefficients in each section separately, thus affecting only a pair of poles and zeros
   - the dynamic range of the coefficients in each second order section is reduced, thus allowing for more fractional bits and better quantization accuracy.