ECE503: Finite Precision Signal Processing: Part II
Lecture 12

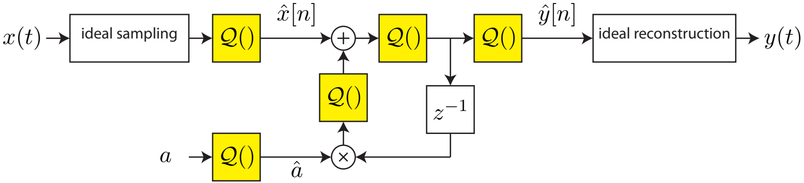
D. Richard Brown III

WPI

23-Apr-2012

Quantization Effects on Digital Filters

Last week, we looked at the various sources and effects of quantization error in digital filters.



Specifically, we looked at (Chap 12.1-12.6)

- Input quantization through the ADC.
- Coefficient quantization.
- Product roundoff quantization.

Note that input quantization noise (even when propagated to the output) is not affected by the filter structure. The other forms of quantization noise **are** affected by the filter structure, however.

Coefficient Quantization: Pole Sensitivity Analysis

Suppose we have a causal stable system with transfer function

$$H(z) = \frac{P(z)}{B(z)} = \frac{P(z)}{z^N + b_{N-1}z^{N-1} + \cdots + b_0} = \frac{P(z)}{(z-\lambda_1)\cdots(z-\lambda_N)}$$

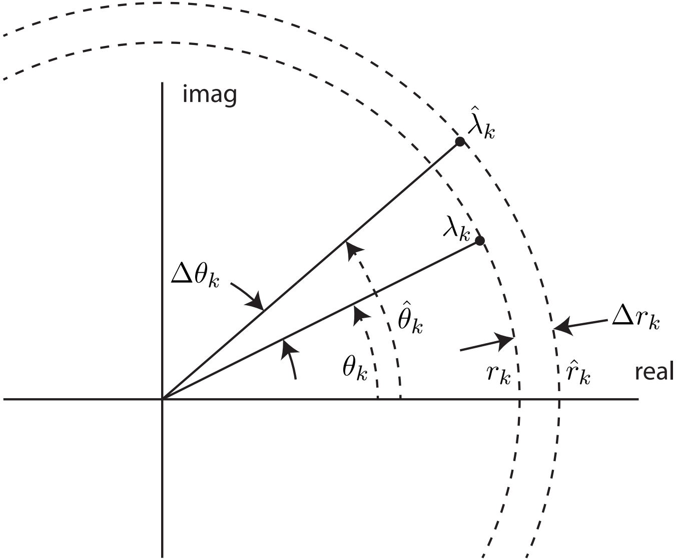
We assume that the leading coefficient of $B(z)$ is one and that all the coefficients are real, i.e. $b_k \in \mathbb{R}$ for $k = 0, \dots, N-1$. Each (potentially complex-valued) pole in the system is denoted as $\lambda_k = r_k e^{j\theta_k}$ for $k = 1, \dots, N$.

In general, quantizing/changing the real-valued transfer function coefficients $\{b_0, \dots, b_{N-1}\} \to \{\hat{b}_0, \dots, \hat{b}_{N-1}\}$ causes the poles to change:

$$\{\lambda_1, \dots, \lambda_N\} \to \{\hat{\lambda}_1, \dots, \hat{\lambda}_N\} \quad \Leftrightarrow \quad \begin{array}{l} \{r_1, \dots, r_N\} \to \{\hat{r}_1, \dots, \hat{r}_N\} \\ \{\theta_1, \dots, \theta_N\} \to \{\hat{\theta}_1, \dots, \hat{\theta}_N\} \end{array}$$

Pole sensitivity analysis: How do small changes to the denominator coefficient(s) affect the magnitude/angle of the pole(s) of the system?

Quantized Coefficients → Pole Displacement



1st Order System Pole Sensitivity Analysis Example

Suppose we have a causal stable system with transfer function

$$H(z) = \frac{P(z)}{B(z)} = \frac{P(z)}{z + b_0} = \frac{P(z)}{z - \lambda_1}$$

In the case of a first order system, the relationship between the coefficient b_0 and the root λ_1 is trivial: $\lambda_1 = -b_0$.

Since λ_1 is real, we have $\lambda_1 = r_1 e^{j\theta_1}$ with $r_1 = |b_0|$ and

$$\theta_1 = \begin{cases} 0 & b_0 < 0 \\ \pi & b_0 \geq 0. \end{cases}$$

Denoting $\hat{b}_0 = b_0 + \Delta b_0$, what can we say about the relationship between Δb_0, Δr_1, and $\Delta\theta_1$? Analysis on board...

2nd Order System Pole Sensitivity Analysis Example

Suppose we have a causal stable system with transfer function

$$H(z) = \frac{P(z)}{B(z)} = \frac{P(z)}{z^2 + b_1 z + b_0} = \frac{P(z)}{(z-\lambda_1)(z-\lambda_2)}$$

We can use the quadratic formula to write

$$\lambda_1 = \frac{-b_1 + \sqrt{b_1^2 - 4b_0}}{2} \text{ and } \lambda_2 = \frac{-b_1 - \sqrt{b_1^2 - 4b_0}}{2}.$$

Note that the poles may be complex, even if the transfer function coefficients are real.

In this case, we could do an **exact** analysis by replacing $\{b_0, b_1\}$ with $\{b_0 + \Delta b_0, b_1 + \Delta b_1\}$ and the exactly computing Δr_0, Δr_1, $\Delta\theta_0$, and $\Delta\theta_1$. The expressions would be pretty messy, however, and we probably wouldn't get much intuition.

Result from Lecture 11: Quantized Root Displacements

Given the partial fraction expansion of the inverse polynomial

$$\frac{1}{B(z)} = \sum_{i=1}^{N} \frac{\rho_i}{z - \lambda_i} = \sum_{i=1}^{N} \frac{\alpha_i + j\beta_i}{z - \lambda_i}$$

we showed that, for a direct form realization, we can estimate

$$\begin{aligned} \Delta r_k &= \left(-\alpha_k \boldsymbol{P}_k + \beta_k \boldsymbol{Q}_k\right) \Delta \boldsymbol{B} \\ \Delta \theta_k &= -\frac{1}{r_k} \left(\beta_k \boldsymbol{P}_k + \alpha_k \boldsymbol{Q}_k\right) \Delta \boldsymbol{B} \end{aligned}$$

for $k = 1, \dots, N$ where N is the order of $B(z)$ and

$$\begin{aligned} \boldsymbol{P}_k &= \begin{bmatrix} \cos\theta_k & r_k & r_k^2 \cos\theta_k & \dots & r_k^{N-1}\cos((N-2)\theta_k) \end{bmatrix} \in \mathbb{R}^{1\times N} \\ \boldsymbol{Q}_k &= \begin{bmatrix} -\sin\theta_k & 0 & r_k^2 \sin\theta_k & \dots & r_k^{N-1}\sin((N-2)\theta_k) \end{bmatrix} \in \mathbb{R}^{1\times N} \\ \Delta \boldsymbol{B} &= \begin{bmatrix} \Delta b_0 \\ \vdots \\ \Delta b_{N-1} \end{bmatrix} \in \mathbb{R}^{N\times 1} \end{aligned}$$

2nd Order System Pole Sensitivity Analysis Example

Following the **approximate** analysis from Lecture 11, we can estimate the magnitude/angle sensitivity of the first pole at $\lambda_1 = r_1 e^{j\theta_1}$ as

$$\Delta r_1 = -\frac{1}{2r_1 \sin\theta_1} \begin{bmatrix} -\sin\theta_1 & 0 \end{bmatrix} \begin{bmatrix} \Delta b_0 \\ \Delta b_1 \end{bmatrix} = \frac{\Delta b_0}{2r_1}$$

$$\Delta\theta_1 = = -\frac{1}{r_1}\left(-\frac{1}{2r_1 \sin\theta_1} \begin{bmatrix} \cos\theta_1 & r_1 \end{bmatrix} \begin{bmatrix} \Delta b_0 \\ \Delta b_1 \end{bmatrix} \right) = \frac{\Delta b_0}{2r_1^2 \tan\theta_1} + \frac{\Delta b_1}{2r_1 \sin\theta_1}$$

Similarly, the magnitude/angle sensitivity of the second pole at $\lambda_2 = r_2 e^{j\theta_2} = \lambda_1^* = r_1 e^{-j\theta_1}$ can be estimated as

$$\Delta r_2 = -\frac{1}{2r_1 \sin\theta_2} \begin{bmatrix} -\sin\theta_2 & 0 \end{bmatrix} \begin{bmatrix} \Delta b_0 \\ \Delta b_1 \end{bmatrix} = \frac{\Delta b_0}{2r_1}$$

$$\Delta\theta_2 = = -\frac{1}{r_1}\left(-\frac{1}{2r_1 \sin\theta_2} \begin{bmatrix} \cos\theta_2 & r_1 \end{bmatrix} \begin{bmatrix} \Delta b_0 \\ \Delta b_1 \end{bmatrix} \right) = \frac{-\Delta b_0}{2r_1^2 \tan\theta_1} - \frac{\Delta b_1}{2r_1 \sin\theta_1}$$

It is not difficult to numerically verify the accuracy of these estimates. Interpretation?

4th Order System Pole Sensitivity Analysis

Now suppose we have a causal stable system with transfer function

$$\begin{aligned} H(z) &= \frac{P(z)}{B(z)} \\ &= \frac{P(z)}{(z-\lambda_1)(z-\lambda_2)(z-\lambda_3)(z-\lambda_4)} \\ &= \frac{P(z)}{z^4 + b_3 z^3 + b_2 z^2 + b_1 z + b_0} \end{aligned}$$

with $\lambda_k = r_k e^{j\theta_k}$ for $k = 1, 2, 3, 4$.

We would like to understand how small changes to the coefficients $\{b_0, b_1, b_2, b_3\} \to \{\hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3\}$ affect the magnitude/angle of the poles.

$$\begin{aligned} \hat{B}(z) &= z^4 + \hat{b}_3 z^3 + \hat{b}_2 z^2 + \hat{b}_1 z + \hat{b}_0 \\ &= (z-\hat{\lambda}_1)(z-\hat{\lambda}_2)(z-\hat{\lambda}_3)(z-\hat{\lambda}_4) \end{aligned}$$

with $\hat{\lambda}_k = \hat{r}_k e^{j\hat{\theta}_k}$ for $k = 1, 2, 3, 4$.

4th Order System Pole Sensitivity Analysis Example

Let's pick some numbers and work through an example. Let's pick

$$\begin{aligned}\lambda_1 &= 0.9e^{j\pi/4}\\ \lambda_2 &= 0.9e^{-j\pi/4}\\ \lambda_3 &= 0.9e^{j\pi/2}\\ \lambda_4 &= 0.9e^{-j\pi/2}\end{aligned}$$

Then

$$\begin{aligned}B(z) &= z^4 - 1.2728z^3 + 1.62z^2 - 1.0310z + 0.6561\\ &= z^4 + b_3z^3 + b_2z^2 + b_1z + b_0\end{aligned}$$

This is our unquantized polynomial. When we implement this in a fixed-point DSP system, we will need to quantize these coefficients.

4th Order System Pole Sensitivity Analysis Example

We have the unquantized polynomial

$$\begin{aligned} B(z) &= z^4 - 1.2728z^3 + 1.62z^2 - 1.0310z + 0.6561 \\ &= z^4 + b_3z^3 + b_2z^2 + b_1z + b_0 \end{aligned}$$

As an example of coefficient quantization, let's change coefficient b_2 from 1.62 to 1.5.

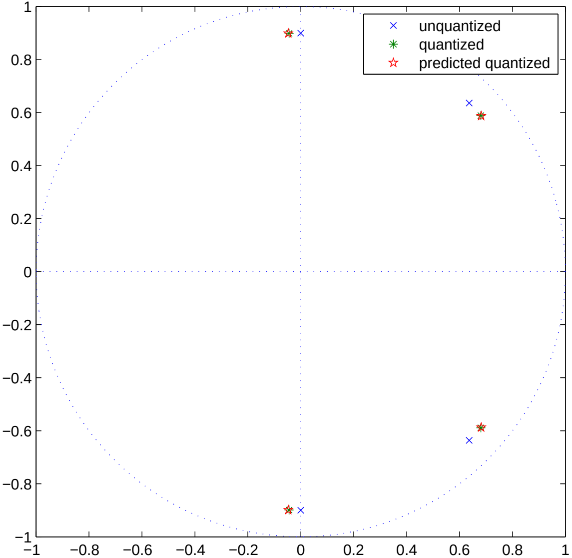
$$\Delta \boldsymbol{B} = \begin{bmatrix} \Delta b_0 \\ \Delta b_1 \\ \Delta b_2 \\ \Delta b_3 \end{bmatrix} = \begin{bmatrix} \hat{b}_0 - b_0 \\ \hat{b}_1 - b_1 \\ \hat{b}_2 - b_2 \\ \hat{b}_3 - b_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -0.12 \\ 0 \end{bmatrix}$$

We can easily compute the new pole locations in Matlab

$$\begin{aligned} \hat{\lambda}_1 &= 0.6805 + j0.5890 = 0.9000e^{j0.7135} \\ \hat{\lambda}_2 &= 0.6805 - j0.5890 = 0.9000e^{-j0.7135} \\ \hat{\lambda}_3 &= -0.0441 + j0.8989 = 0.9000e^{j1.6198} \\ \hat{\lambda}_4 &= -0.0441 - j0.8989i = 0.9000e^{-j1.6198} \end{aligned}$$

Note that changing b_2 seems to have no effect on the pole magnitudes (the pole angles changed, however). Was this just a lucky coincidence?

4th Order System Pole Sensitivity Analysis Example



4th Order System Pole Sensitivity Analysis Example

We have the unquantized polynomial

$$\begin{aligned} B(z) &= z^4 - 1.2728z^3 + 1.62z^2 - 1.0310z + 0.6561 \\ &= z^4 + b_3 z^3 + b_2 z^2 + b_1 z + b_0 \end{aligned}$$

with roots

$$\begin{aligned} \lambda_1 &= 0.9e^{j\pi/4} \\ \lambda_2 &= 0.9e^{-j\pi/4} \\ \lambda_3 &= 0.9e^{j\pi/2} \\ \lambda_4 &= 0.9e^{-j\pi/2} \end{aligned}$$

Assuming a **direct form realization**, we would like to understand how small changes to the coefficients $\{b_0, b_1, b_2, b_3\} \to \{\hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3\}$ affect the magnitude and angle of the poles. We've seen a small change in b_2 doesn't seem to change the pole magnitudes. Can we confirm this analytically with the approximate analysis technique? On board...

Interpreting the Pole Sensitivity Matrices

In this example, we saw that the pole magnitude sensitivity matrix was

$$\boldsymbol{S}_b^r = \begin{bmatrix} 0.6859 & 0.4365 & 0 & -0.3536 \\ 0.6859 & 0.4365 & 0 & -0.3536 \\ 0 & -0.4365 & 0 & 0.3536 \\ 0 & -0.4365 & 0 & 0.3536 \end{bmatrix}$$

This confirms that small changes in b_2 have no effect on the magnitude of any of the poles. Also note that small changes in b_0 have no effect on the magnitude of the poles λ_3 and λ_4.

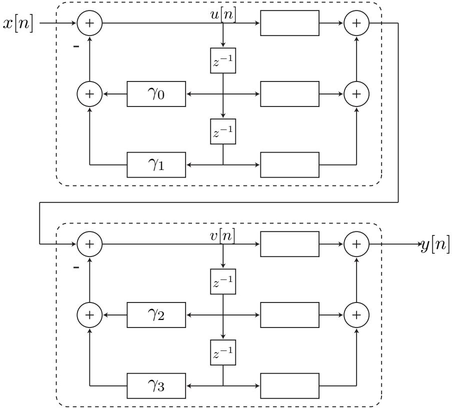
We also saw that the pole angle sensitivity matrix was

$$\boldsymbol{S}_b^\theta = \begin{bmatrix} 0 & 0.4850 & 0.6173 & 0.3928 \\ 0 & -0.4850 & -0.6173 & -0.3928 \\ 0.5389 & 0 & -0.4365 & 0 \\ -0.5389 & 0 & 0.4365 & 0 \end{bmatrix}$$

Small changes in b_0 have no effect on the angle of the poles λ_1 and λ_2, etc. Now we see that small changes in b_2 significantly affect on the angle of the poles.

4th Order Example: Cascade Realization

Suppose instead of direct form, we implemented our transfer function as the cascade of two second-order DF-II sections:



Note the new parameters γ_0, γ_1, γ_2, and γ_3. First, given the original $H(z)$, how can we determine γ_0, γ_1, γ_2, and γ_3 (Chapter 8 review)?

4th Order Example: Cascade Realization

Recall

$$H(z) = \frac{P(z)}{(z-\lambda_1)(z-\lambda_2)(z-\lambda_3)(z-\lambda_4)}$$

with

$$\lambda_1 = 0.9e^{j\pi/4} \qquad \lambda_2 = 0.9e^{-j\pi/4}$$
$$\lambda_3 = 0.9e^{j\pi/2} \qquad \lambda_4 = 0.9e^{-j\pi/2}$$

We can rewrite the denominator as a product of second order polynomials

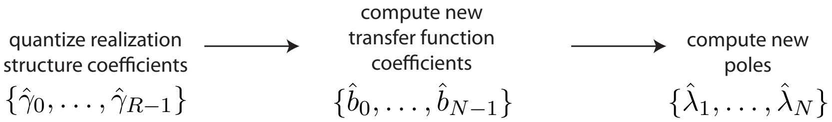
$$\begin{aligned} H(z) &= \frac{P(z)}{(z^2 - 1.8\cos(\pi/4)z + 0.81)(z^2 - 1.8\cos(\pi/2)z + 0.81)} \\ &= \frac{P_1(z)}{z^2 + \gamma_0 z + \gamma_1} \cdot \frac{P_2(z)}{z^2 + \gamma_2 z + \gamma_3} \end{aligned}$$

Hence $\gamma_0 = -1.2728$, $\gamma_1 = 0.81$, $\gamma_2 = 0$, and $\gamma_3 = 0.81$.

4th Order Example: Cascade Realization

We have the unquantized coefficients $\gamma_0 = -1.2728$, $\gamma_1 = 0.81$, $\gamma_2 = 0$, and $\gamma_3 = 0.81$. If we implement this filter with a fixed-point DSP, we need to quantize these coefficients.

Note that we are **not** directly quantizing the b_0, b_1, b_2, b_3 coefficients here. We are quantizing the $\gamma_0, \gamma_1, \gamma_2, \gamma_3$ coefficients in our structure, which changes the b_0, b_1, b_2, b_3 coefficients, which then changes the magnitude/angle of the poles.



4th Order Example: Cascade Realization

As an example, let's quantize coefficient γ_0 from -1.2728 to -1.5.

$$\Delta\boldsymbol{\gamma} = \begin{bmatrix} \Delta\gamma_0 \\ \Delta\gamma_1 \\ \Delta\gamma_2 \\ \Delta\gamma_3 \end{bmatrix} = \begin{bmatrix} \hat{\gamma}_0 - \gamma_0 \\ \hat{\gamma}_1 - \gamma_1 \\ \hat{\gamma}_2 - \gamma_2 \\ \hat{\gamma}_3 - \gamma_3 \end{bmatrix} = \begin{bmatrix} -0.2272 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

How does this change affect b_0, b_1, b_2, b_3?

$$\begin{aligned} B(z) &= (z^2 + \gamma_0 + \gamma_1)(z^2 + \gamma_2 + \gamma_3) \\ &= z^4 + (\gamma_0 + \gamma_2)z^3 + (\gamma_1 + \gamma_3 + \gamma_0\gamma_2)z^2 + (\gamma_0\gamma_3 + \gamma_1\gamma_2)z + \gamma_1\gamma_3 \\ &= z^4 + b_3 z^3 + b_2 z^2 + b_1 z + b_0 \end{aligned}$$

We see that changing γ_0 affects all of the polynomial coefficients except b_0. Changing γ_0 from -1.2728 to -1.5 results in $\hat{b}_0 = 0.6561$, $\hat{b}_1 = -1.215$, $\hat{b}_2 = 1.62$, and $\hat{b}_3 = -1.5$ (the original coefficients were $b_0 = 0.6561$, $b_1 = -1.031$, $b_2 = 1.62$, and $b_3 = -1.2728$).

4th Order Example: Cascade Realization

Changing γ_0 from -1.2728 to -1.5 results in $\hat{b}_0 = 0.6561$, $\hat{b}_1 = -1.215$, $\hat{b}_2 = 1.62$, and $\hat{b}_3 = -1.5$. Hence

$$\hat{B}(z) = z^4 - 1.5z^3 + 1.62z^2 - 1.215z + 0.6561$$

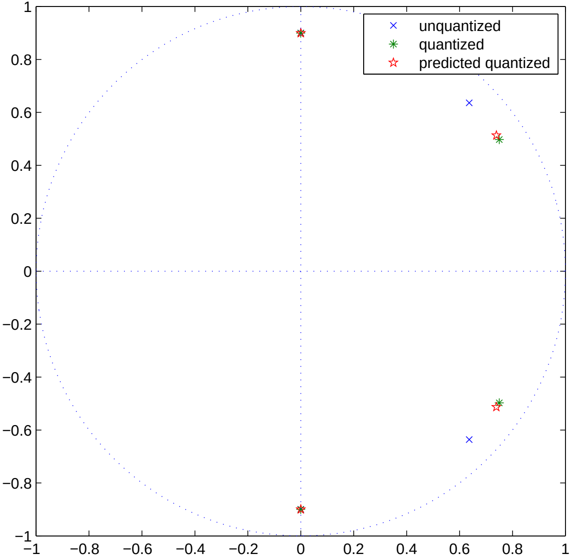
We can easily compute the new pole locations in Matlab

$$\begin{aligned}
\hat{\lambda}_1 &= 0.75 + j0.4975 = 0.9000e^{j0.5857} \\
\hat{\lambda}_2 &= 0.75 - j0.4975 = 0.9000e^{-j0.5857} \\
\hat{\lambda}_3 &= 0 + j0.9 = 0.9000e^{j1.5708} \\
\hat{\lambda}_4 &= 0 - j0.9 = 0.9000e^{-j1.5708}
\end{aligned}$$

Remarks:

- Changing γ_0 seems to have no effect on the pole magnitudes.
- In fact, this change to γ_0 seems to have no effect at all on λ_3 and λ_4.
- The only thing that changed seems to be the angles of λ_1 and λ_2.
- Can we confirm this analytically?

4th Order System Pole Sensitivity Analysis Example



4th Order Example: Cascade Realization

With our cascaded SOS realization, if we change γ_2 such that

$$\Delta\boldsymbol{\gamma} = \begin{bmatrix} -0.2272 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{then} \quad \Delta\boldsymbol{B} = \begin{bmatrix} 0 \\ -0.1840 \\ 0 \\ -0.2272 \end{bmatrix}$$

and we can apply our previous result for the direct form case to determine $\Delta\boldsymbol{r}$ and $\Delta\boldsymbol{\theta}$.

$$\Delta\boldsymbol{r} = \boldsymbol{S}_b^r \Delta\boldsymbol{B} = \begin{bmatrix} 0.6859 & 0.4365 & 0 & -0.3536 \\ 0.6859 & 0.4365 & 0 & -0.3536 \\ 0 & -0.4365 & 0 & 0.3536 \\ 0 & -0.4365 & 0 & 0.3536 \end{bmatrix} \begin{bmatrix} 0 \\ -0.1840 \\ 0 \\ -0.2272 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Delta\boldsymbol{\theta} = \boldsymbol{S}_b^\theta \Delta\boldsymbol{B} = \begin{bmatrix} 0 & 0.4850 & 0.6173 & 0.3928 \\ 0 & -0.4850 & -0.6173 & -0.3928 \\ 0.5389 & 0 & -0.4365 & 0 \\ -0.5389 & 0 & 0.4365 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -0.1840 \\ 0 \\ -0.2272 \end{bmatrix} = \begin{bmatrix} -0.1785 \\ 0.1785 \\ 0 \\ 0 \end{bmatrix}$$

This agrees nicely with what we saw when we computed the roots in Matlab.

4th Order Example: Cascade Realization

Our procedure:

1. Start with a given $\Delta\boldsymbol{\gamma}$ (this represents the change in the parameters of your realization structure due to quantization).
2. Determine $\Delta\boldsymbol{B}$ (this is the change in the polynomial coefficients). You need to know how the $b_0, b_1, \dots$ coefficients are related to the $\gamma_0, \gamma_1, \dots$ parameters of your realization structure to do this.
3. Use the direct form analysis sensitivity matrices $\boldsymbol{S}_b^r$ and $\boldsymbol{S}_b^\theta$ to predict the pole magnitude and angle changes, i.e.

$$\Delta\boldsymbol{r} = \boldsymbol{S}_b^r \Delta\boldsymbol{B}$$
$$\Delta\boldsymbol{\theta} = \boldsymbol{S}_b^\theta \Delta\boldsymbol{B}$$

There is one more thing we can do to obtain an even more direct analytical intuition for the relationship between $\Delta\boldsymbol{\gamma}$, $\Delta\boldsymbol{r}$ and $\Delta\boldsymbol{\theta}$, ...

Taylor Series Approximation Review

Recall the first-order Taylor series approximation of $f : \mathbb{R} \mapsto \mathbb{R}$ around the point $x = a$:

$$f(x) \approx f(a) + (x - a)\left[\frac{d}{dx} f(x)\right]_{x=a}$$

The multivariable version of this for $f : \mathbb{R}^N \mapsto \mathbb{R}$ around the point $\boldsymbol{x} = \boldsymbol{a}$ is

$$f(\boldsymbol{x}) \approx f(\boldsymbol{a}) + (\boldsymbol{x} - \boldsymbol{a})^\top \left[\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\right]_{\boldsymbol{x}=\boldsymbol{a}}$$

where $\nabla_{\boldsymbol{x}} = \left[\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_N}\right]^\top$ is the gradient operator you may recall from multivariable calculus. We can rewrite this last expression as

$$f(\boldsymbol{x}) - f(\boldsymbol{a}) \approx \sum_{n=1}^{N} (x_n - a_n)\left[\frac{\partial}{\partial x_n} f(\boldsymbol{x})\right]_{\boldsymbol{x}=\boldsymbol{a}}$$

Relating Structure Parameters & Polynomial Coefficients

In general, given a vector of realization structure parameters $\boldsymbol{\gamma} = [\gamma_0, \dots, \gamma_{R-1}]^\top$, each denominator polynomial coefficient will be a continuous function of these parameters, i.e.,

$$b_k = f_k(\boldsymbol{\gamma}) \text{ for } k = 0, \dots, N-1.$$

and, for the quantized parameters,

$$\hat{b}_k = f_k(\hat{\boldsymbol{\gamma}}) \text{ for } k = 0, \dots, N-1.$$

We can use our first-order Taylor series approximation to write

$$\begin{aligned} \Delta b_k &= \hat{b}_k - b_k \\ &= f_k(\hat{\boldsymbol{\gamma}}) - f_k(\boldsymbol{\gamma}) \\ &\approx \sum_{n=0}^{R-1} (\hat{\gamma}_n - \gamma_n) \left[\frac{\partial}{\partial \hat{\gamma}_n} f_k(\hat{\boldsymbol{\gamma}}) \right]_{\hat{\boldsymbol{\gamma}} = \boldsymbol{\gamma}} \\ &= \sum_{n=0}^{R-1} \Delta \gamma_n \frac{\partial b_k}{\partial \gamma_i} \end{aligned}$$

for $k = 0, \dots, N-1$. This approximation is reasonable for small $\Delta \boldsymbol{\gamma}$.

Relating Structure Parameters & Polynomial Coefficients

So we have the result

$$\Delta b_k \approx \sum_{n=0}^{R-1} \Delta\gamma_n \frac{\partial b_k}{\partial \gamma_i} \text{ for } k = 0, \dots, N-1.$$

We can stack these up into a matrix/vector form to to write

$$\begin{bmatrix} \Delta b_0 \\ \Delta b_1 \\ \vdots \\ \Delta b_{N-1} \end{bmatrix} \approx \underbrace{\begin{bmatrix} \frac{\partial b_0}{\partial \gamma_0} & \frac{\partial b_0}{\partial \gamma_1} & \cdots & \frac{\partial b_0}{\partial \gamma_{R_1}} \\ \frac{\partial b_1}{\partial \gamma_0} & \frac{\partial b_1}{\partial \gamma_1} & \cdots & \frac{\partial b_1}{\partial \gamma_{R-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial b_{N-1}}{\partial \gamma_0} & \frac{\partial b_{N-1}}{\partial \gamma_1} & \cdots & \frac{\partial b_{N-1}}{\partial \gamma_{R-1}} \end{bmatrix}}_{\boldsymbol{C} \in \mathbb{R}^{N \times R}} \begin{bmatrix} \Delta\gamma_0 \\ \Delta\gamma_1 \\ \vdots \\ \Delta\gamma_{R-1} \end{bmatrix}$$

Hence $\Delta \boldsymbol{B} \approx \boldsymbol{C} \Delta \boldsymbol{\gamma}$. Furthermore,

$$\Delta \boldsymbol{r} = \boldsymbol{S}_b^r \boldsymbol{C} \Delta \boldsymbol{\gamma}$$
$$\Delta \boldsymbol{\theta} = \boldsymbol{S}_b^\theta \boldsymbol{C} \Delta \boldsymbol{\gamma}.$$

Relating Structure Parameters & Polynomial Coefficients

Note that the C matrix relates **realization structure parameter changes** to **transfer function coefficient changes**. This allows our approximate pole sensitivity analysis to be extended to any realization structure.

Examples:

1. What is the C matrix for a direct form realization, i.e. $b_k = \gamma_k$ for $k = 0, 1, \dots, N-1$?
2. What is the C matrix for our cascaded SOS realization?

Since

$$\Delta \boldsymbol{r} = \boldsymbol{S}_b^r \boldsymbol{C} \Delta \boldsymbol{\gamma} = \boldsymbol{S}_\gamma^r \Delta \boldsymbol{\gamma}$$
$$\Delta \boldsymbol{\theta} = \boldsymbol{S}_b^\theta \boldsymbol{C} \Delta \boldsymbol{\gamma} = \boldsymbol{S}_\gamma^\theta \Delta \boldsymbol{\gamma}.$$

we can think of $\boldsymbol{S}_\gamma^r = \boldsymbol{S}_b^r \boldsymbol{C}$ and $\boldsymbol{S}_\gamma^\theta = \boldsymbol{S}_b^\theta \boldsymbol{C}$ as being the pole sensitivity matrices for the realization form described by $\boldsymbol{C}$.

Comparison of Direct form and Cascaded SOS Sensitivity

Pole magnitude sensitivity matrices for direct form and SOS cascade:

$$\boldsymbol{S}_b^r = \begin{bmatrix} 0.6859 & 0.4365 & 0 & -0.3536 \\ 0.6859 & 0.4365 & 0 & -0.3536 \\ 0 & -0.4365 & 0 & 0.3536 \\ 0 & -0.4365 & 0 & 0.3536 \end{bmatrix} \qquad \boldsymbol{S}_\gamma^r = \begin{bmatrix} 0 & 0.5556 & 0 & 0 \\ 0 & 0.5556 & 0 & 0 \\ 0 & 0 & 0 & 0.5556 \\ 0 & 0 & 0 & 0.5556 \end{bmatrix}$$

Pole magnitude sensitivity matrices for direct form and SOS cascade:

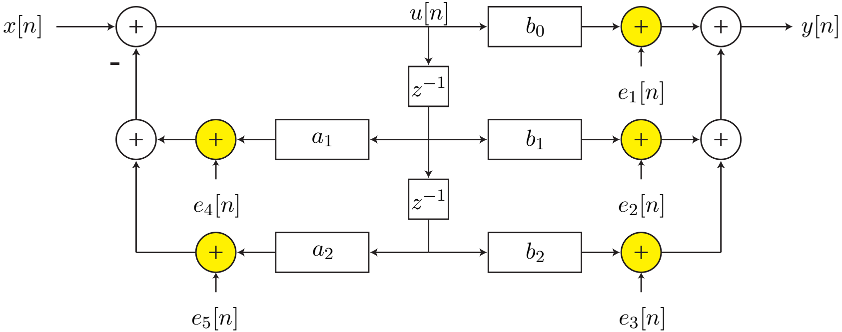
$$\boldsymbol{S}_b^\theta = \begin{bmatrix} 0 & 0.4850 & 0.6173 & 0.3928 \\ 0 & -0.4850 & -0.6173 & -0.3928 \\ 0.5389 & 0 & -0.4365 & 0 \\ -0.5389 & 0 & 0.4365 & 0 \end{bmatrix}$$

$$\boldsymbol{S}_\gamma^\theta = \begin{bmatrix} 0.7857 & 0.6173 & 0 & 0 \\ -0.7857 & -0.6173 & 0 & 0 \\ 0 & 0 & 0.5556 & 0 \\ 0 & 0 & -0.5556 & 0 \end{bmatrix}$$

What do these results tell us about the advantages/disadvantages of the SOS cascade realization (at least for this particular fourth order system)?

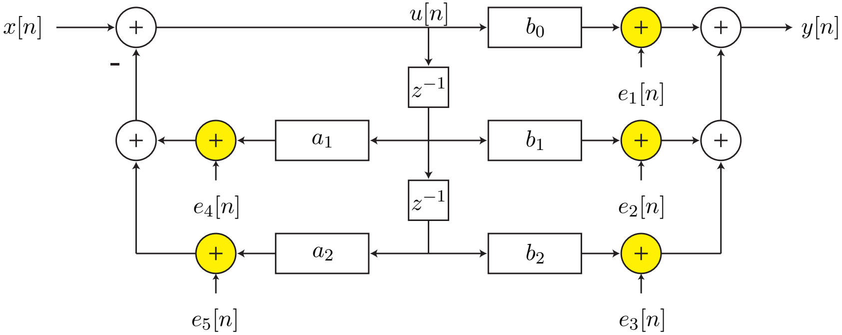
Product Roundoff Noise: Effect of Realization Structure

Switching gears, consider the following second order DF-II realization example with explicit product roundoff errors $e_k[n]$:



In this case, we are not concerned with coefficient quantization (the coefficients are all assumed to be unquantized). Instead, we wish to understand how the product roundoff noise appears in the output.

Product Roundoff Noise: Effect of Realization Structure



Using the principle of superposition and ignoring the input, we see that

$$Y(z) = E_1(z) + E_2(z) + E_3(z) + H(z)(E_4(z) + E_5(z))$$

where $H(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}$.

Product Roundoff Noise: Effect of Realization Structure

Assumptions to facilitate statistical analysis:

- Each $e_k[n]$ is identically distributed and independent of $e_\ell[n]$ for $k \neq \ell$.
- Each $e_k[n]$ wide-sense stationary with zero mean and variance σ_e^2 for all n.

$$Y(z) = E_1(z) + E_2(z) + E_3(z) + H(z)(E_4(z) + E_5(z))$$

which implies that the variance of the product roundoff noise at the output is

$$\sigma_y^2 = 3\sigma_e^2 + 2\sigma_e^2 \cdot \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\omega)|^2 \, d\omega$$

where we used our result from Lecture 11 regarding the propagation of noise through an LTI system.

Product Roundoff Noise: Effect of Realization Structure

Let's pick some numbers to continue the example. Suppose $\sigma_e^2 = 1$ and

$$H(z) = \frac{0.6 + 0.54z^{-1} + 0.108z^{-2}}{1 - 1.3z^{-1} + 0.4z^{-2}}$$

with ROC $|z| > 0.8$ (causal and stable).

With these numbers, we can compute the integral (using, for example, the algebraic technique in 12.5.5) to be

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 \, d\omega \approx 12.7719$$

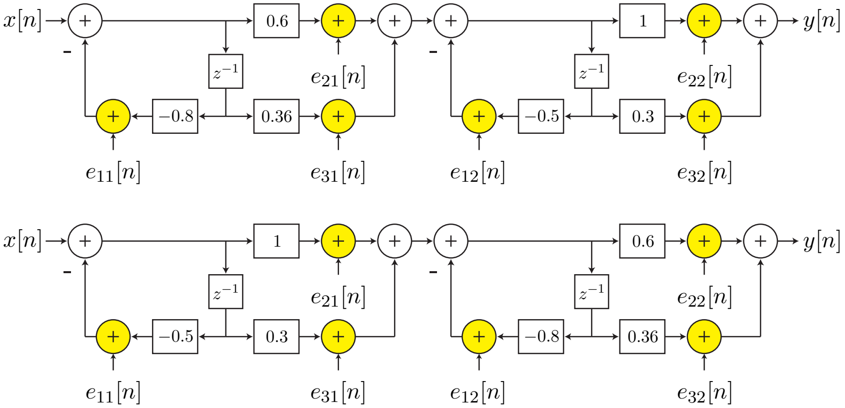
which means that

$$\sigma_y^2 = 3 + 2 \cdot 12.7719 = 28.5438$$

We see that the product noise in the feedback coefficients is the dominating source of noise in the output.

Product Roundoff Noise: Effect of Realization Structure

Now, what if we split this realization structure up into a cascade of four first order sections? There are (at least) four possibilities. Here are two:



Note both realizations have the same $H(z) = H_1(z)H_2(z) = H_2(z)H_1(z)$. Is there any difference?

Product Roundoff Noise: Effect of Realization Structure

There is a difference in how the roundoff noise propagates.

In the first realization:

- $e_{11}[n]$ propagates through $H(z)$ to get to the output.
- $e_{21}[n]$, $e_{31}[n]$, and $e_{12}[n]$ propagate through $H_2(z)$ to get to the output.
- $e_{22}[n] = 0$.
- $e_{23}[n]$ is directly connected to the output.

In the second realization:

- $e_{11}[n]$ propagates through $H(z)$ to get to the output.
- $e_{21}[n] = 0$.
- $e_{31}[n]$, and $e_{12}[n]$ propagate through $H_1(z)$ to get to the output.
- $e_{22}[n]$ and $e_{23}[n]$ are directly connected to the output.

Which structure is better? How do these compare to the non-cascade form?

Product Roundoff Noise: Effect of Realization Structure

Given

$$H_1(z) = \frac{0.6 + 0.36z^{-1}}{1 - 0.8z^{-1}}$$
$$H_2(z) = \frac{1 + 0.3z^{-1}}{1 - 0.5z^{-1}}$$

we can use the geometric methods in the textbook (or the usual series convergence results) to compute

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |H_1(\omega)|^2 \, d\omega \approx 2.32$$
$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |H_2(\omega)|^2 \, d\omega \approx 1.8533$$

Also recall our earlier result

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 \, d\omega \approx 12.7719$$

Product Roundoff Noise: Effect of Realization Structure

For the first realization:

- $e_{11}[n]$ propagates through $H(z)$ to get to the output.
- $e_{21}[n]$, $e_{31}[n]$, and $e_{12}[n]$ propagate through $H_2(z)$ to get to the output.
- $e_{22}[n] = 0$.
- $e_{23}[n]$ is directly connected to the output.

Hence

$$\sigma_y^2 = 12.7719 + 3 \cdot 1.8533 + 0 + 1 \approx 19.33$$

In the second realization:

- $e_{11}[n]$ propagates through $H(z)$ to get to the output.
- $e_{21}[n] = 0$.
- $e_{31}[n]$, and $e_{12}[n]$ propagate through $H_1(z)$ to get to the output.
- $e_{22}[n]$ and $e_{23}[n]$ are directly connected to the output.

Hence

$$\sigma_y^2 = 12.7719 + 0 + 2 \cdot 2.32 + 0 + 2 \approx 19.41$$

Not much difference, but both better than the full DF-II realization.

Final Exam

1. 6pm 30-Apr-2012. 180 minutes.
2. Open book.
3. Two cheat sheets, double sided, letter sized, **in your own handwriting**.
4. Calculator permitted.
5. Comprehensive: Chapters 1-9, parts of Chapter 11 (FFT and number representation), Chapter 12.1-12.6.
6. There will definitely be some material from the second half of the class, e.g.
    - Realization structures
    - IIR filter design
    - FFT
    - Fixed-point number representation and quantization basics
    - Effects of finite precision on filtering, e.g. pole sensitivity, roundoff error effects, ...
7. Two-hour special help session on Saturday 28-Apr-2012 (time to be announced via email later this week).