EE482: Digital Signal Processing Applications

Spring 2014
TTh 14:30-15:45 CBC C222

Lecture 06
IIR Design 2
14/03/06

http://www.ee.unlv.edu/~b1morris/ee482/
Outline

- Review IIR Design
- Implementation Considerations
- Stability
- Coefficient Quantization
- Roundoff Effects
- Cascade Pairing and Ordering
IIR Design

- Reuse well studied analog filter design techniques (books and tables for design)
- Need to map between analog design and a digital design
  - Mapping between s-plane and z-plane
IIR Filter Design

- IIR transfer function

\[
H(z) = \frac{\sum_{l=0}^{L-1} b_l z^{-l}}{1 + \sum_{l=0}^{M} a_l z^{-l}}
\]

- Need to find coefficients \(a_l, b_l\)
  - Impulse invariance – sample impulse response
    - Have to deal with aliasing
  - Bilinear transform
    - Match magnitude response
    - “Warp” frequencies to prevent aliasing
Bilinear Transform Design

- Convert digital filter into an “equivalent” analog filter
  - Use bilinear “warping”
- Design analog filter using IIR design techniques
- Map analog filter into digital
  - Use bilinear transform

Figure 4.5 Digital IIR filter design using the bilinear transform
Bilinear Design Steps

1. Convert digital filter into an “equivalent” analog filter
   ▫ Pre-warp using
     \[ \Omega = \frac{2}{T} \tan \left( \frac{\omega}{2} \right) \]

2. Design analog filter using IIR design techniques
   ▫ Butterworth, Chebyshev, Elliptical

3. Map analog filter into digital
   ▫ \[ H(z) = H(s) \bigg|_{s=\frac{2}{T} \left( \frac{1-z^{-1}}{1+z^{-1}} \right)} \]
Direct Form I

- **Straight-forward implementation of diff. eq.**
  - \( b_l \) - feed forward coefficients
    - From \( x(n) \) terms
  - \( a_l \) - feedback coefficients
    - From \( y(n) \) terms

- Requires \((L + M)\) coefficients and delays

\[ x[n] \rightarrow z^{-1} \rightarrow b_0 \rightarrow v[n] \rightarrow b_1 \rightarrow b_2 \rightarrow \ldots \rightarrow b_N \rightarrow z^{-1} \rightarrow y[n] \]

\[ x[n-1] \rightarrow z^{-1} \rightarrow b_1 \rightarrow \ldots \rightarrow b_N \rightarrow z^{-1} \rightarrow y[n-1] \]

\[ x[n-2] \rightarrow z^{-1} \rightarrow b_2 \rightarrow \ldots \rightarrow b_N \rightarrow z^{-1} \rightarrow y[n-2] \]

\[ x[n-N+1] \rightarrow z^{-1} \rightarrow b_{N-1} \rightarrow \ldots \rightarrow b_N \rightarrow z^{-1} \rightarrow y[n-N+1] \]

\[ x[n-N] \rightarrow b_N \rightarrow \ldots \rightarrow b_N \rightarrow z^{-1} \rightarrow y[n-N] \]
Direct Form II

- Notice that we can decompose the transfer function
  - \( H(z) = H_1(z)H_2(z) \)
    - Section to implement zeros
    - Section to implement poles

- Can switch order of operations
  - \( H(z) = H_2(z)H_1(p) \)
  - This allows sharing of delays and saving in memory

---

Figure 4.7 Direct-form I realization of second-order IIR filter
Cascade (Factored) Form

- Factor transfer function and decompose into smaller sub-systems
  - \( H(z) = H_1(z)H_2(z) \ldots H_K(z) \)

- Make each subsystem second order
  - Complex conjugate roots have real coefficients
  - Limit the order of subsystem (numerical effects)
    - Effects limited to single subsystem stage
    - Change in a single coefficient affects all poles in DF

- Preferred over DF because of numerical stability

---

**Figure 4.10** Cascade realization of digital filter
Parallel (Partial Fraction) Form

- Decompose transfer function using a partial fraction expansion
  - $H(z) = H_1(z) + H_2(z) + \ldots + H_K(z)$
  - $H_k(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}$
- Be sure to remember that PFE requires numerator order less than denominator
  - Use polynomial long division
Matlab Filter Design

- **Realization tools:**
  - Finding polynomial roots
    - `roots.m`
    - `tf2zp.m`
  - Cascade form
    - \( H(z) = G \prod_{k=1}^{K} \frac{b_0k + b_1kz^{-1} + b_2kz^{-2}}{1 + a_1kz^{-1} + a_2kz^{-2}} \)
    - `zp2sos.m`
  - Parallel form
    - `Residuez.m`

- **Filter design tools:**
  - Order estimation tool
    - `butterord.m`
  - Coefficient tool
    - `butter.m`
  - Frequency transforms
    - `lp2hp.m`, `lp2bp.m`, `lp2bs.m`
  - Useful exploration tool
    - `fvtool.m`
  - Useful design tool
    - `fdatool.m`
  - Useful processing tool
    - `sptool.m`
Stability

- (Causal) IIR filters are stable if all poles are within the unit circle
  - \(|p_m| < 1\)
  - We will not consider marginally stable (single pole on unit circle)
- Consider poles of 2\textsuperscript{nd} order filter (used in cascade and parallel forms)
  - \(A(z) = 1 + a_1 z^{-1} + a_2 z^{-2}\)
- Factor
  - \(A(z) = (1 - p_1 z^{-1})(1 - p_2 z^{-1})\)
  - \(A(z) = 1 - (p_1 + p_2) z^{-1} + p_1 p_2 z^{-2}\)
- Because poles must be inside the unit circle
  - \(|a_2| = |p_1 p_2| < 1\)
  - \(|a_1| < 1 + a_2\)
Coefficient Quantization

- Using fixed word lengths results in a quantized approximation of a filter
  \[ H'(z) = \frac{\sum_{k=0}^{L-1} b'_k z^{-k}}{1+\sum_{k=1}^{M} a'_k z^{-k}} \]
- This can cause a mismatch from desired system \( H(z) \)
- Poles that are close to the unit circle may move outside and cause instability
  - This is exacerbated with higher order systems
Rounding Effects

• Using $B$ bit architecture, products require $2B$ bits
  ▫ Must be rounded into smaller $B$ bit container
• This results in noise error terms
  ▫ Can be simply modeled as additive term
• The order of cascade sections influences power of noise at output
  ▫ How should sections be paired and ordered?
• Need to optimize SQNR
  ▫ Trade-off with probability of arithmetic overflow
  ▫ Need to use scaling factors to prevent overflow
  ▫ Optimality when signal level is maximized without overflow
Cascade Ordering and Pairing

- Good results are obtained using simple rules
- Cascade ordering and pairing algorithm

1. Pair pole closest to unit circle with zero that is closest in z-plane
   - Minimize the chance of overflow
2. Apply 1 repeatedly until all poles and zeros are paired
3. Resulting 2\textsuperscript{nd} -order sections can be ordered in two alternative ways
   - Increasing closeness to unit circle
   - Decreasing closeness to unit circle

Figure 6.67 Output noise power spectrum for 123 ordering (solid line) and 321 ordering (dashed line) of 2\textsuperscript{nd}-order sections.
Recursive Resonator

- Filter with frequency response dominated at a single peak
  - Use complex-conjugate pole pair inside unit circle

- \( H(z) = \frac{A}{(1-r_p e^{j\omega_0 z^{-1}})(1-r_p e^{-j\omega_0 z^{-1}})} \)

- \( H(z) = \frac{A}{1-2r_p \cos(\omega_0)z^{-1}+r_p^2 z^{-2}} \)
  - \( A \) – normalization constant for unity gain at \( \omega_0 \)
  - \( 0 < r_p < 1 \)
- Close to unit circle
  - Bandwidth \( \approx 2(1 - r_p) \)
  - Closer to \( r_p = 1 \), more peaked
Parametric Equalizer

- Add nearby zeros to the resonator
  - At same angle as poles $\omega_0$
  - Similar radius
- Pole and zero counter balance one another
- $r_z < r_p$
  - Pole dominates because it is closer to unit circle
  - Generates peak at $\omega = \omega_0$
    - Provides boost to freq
- $r_z > r_p$
  - Zero dominates pole
  - Generates dip at $\omega = \omega_0$
    - Cuts freq
- Bandwidth still determined by $r_p$

Ex 4.18
- Create equalizer by changing gain at given frequency