
ELEN E4810: Digital Signal Processing

Topic 7:

Filter types and structures

1. More filter types
2. Minimum and maximum phase
3. Filter implementation structures



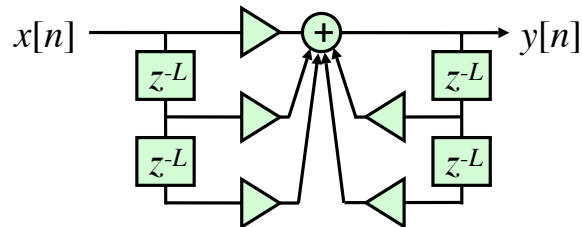
1. More Filter Types

- We have seen the basics of filters and a range of simple examples
- Now look at a couple of other classes:
 - **Comb filters** - multiple pass/stop bands
 - **Allpass filters** - only modify signal **phase**



Comb Filters

- Replace all system delays z^{-1} with **longer** delays z^{-L}



→ System that behaves ‘the same’ at a **longer** timescale



Comb Filters

- ‘Parent’ filter impulse response $h[n]$ becomes **comb filter** output as:

$$g[n] = \{h[0] \underbrace{0 \ 0 \ 0 \ 0}_{L-1 \text{ zeros}} h[1] \ 0 \ 0 \ 0 \ 0 \ h[2] \dots\}$$

- Thus,
$$G(z) = \sum_n g[n] z^{-n} = \sum_n h[n] z^{-nL} = H(z^L)$$

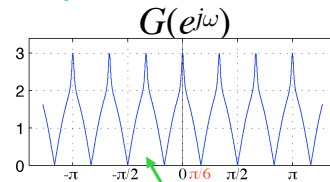
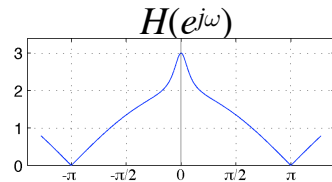


Comb Filters

- Hence frequency response:

$$G(e^{j\omega}) = H(e^{j\omega L})$$

parent frequency response compressed & repeated L times



- Low-pass response \rightarrow

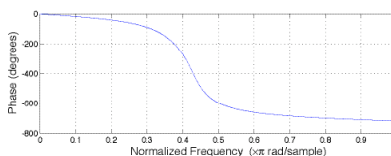
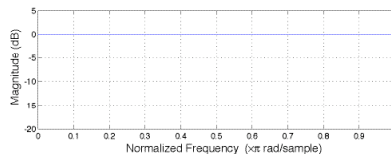
- pass $\omega = 0, 2\pi/L, 4\pi/L, \dots$
- cut $\omega = \pi/L, 3\pi/L, 5\pi/L, \dots$

useful to enhance a harmonic series



Allpass Filters

- Allpass filter has $|A(e^{j\omega})|^2 = K$ for all ω
i.e. spectral energy is not changed
- Phase response is **not** zero (else trivial)
 - phase correction
 - special effects
- e.g.



Allpass Filters

- Allpass has special form of system fn:

$$A_M(z) = \pm \frac{d_M + d_{M-1}z^{-1} + \dots + d_1z^{-(M-1)} + z^{-M}}{1 + d_1z^{-1} + \dots + d_{M-1}z^{-(M-1)} + d_Mz^{-M}}$$

$$= \pm z^{-M} \frac{D_M(z^{-1})}{D_M(z)}$$

mirror-image polynomials

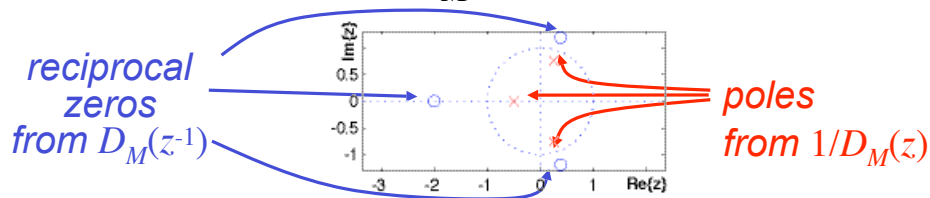
- $A_M(z)$ has **poles** λ where $D_M(\lambda) = 0$
 $\rightarrow A_M(z)$ has **zeros** $\zeta = 1/\lambda = \lambda^{-1}$



Allpass Filters

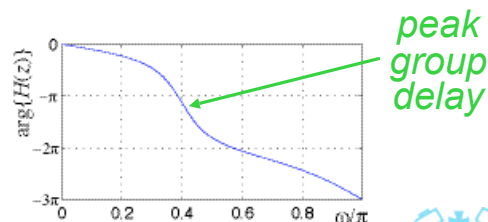
$$A_M(z) = \pm z^{-M} \frac{D_M(z^{-1})}{D_M(z)}$$

- Any (stable) D_M can be used:



- Phase is always decreasing:

$\rightarrow -M\pi$ at $\omega = \pi$



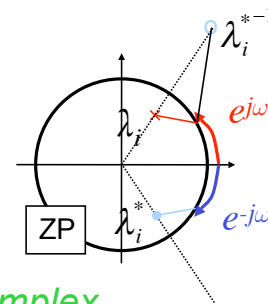
Allpass Filters

Why do mirror-imag poly's give const gain?

- **Conj-sym** system fn can be factored as:

$$A_M(z) = \frac{K \prod_i (z - \lambda_i^{*-1})}{\prod_i (z - \lambda_i)}$$

$$= \frac{K \prod_i \lambda_i^{*-1} z (\lambda_i^* - z^{-1})}{\prod_i (z - \lambda_i)}$$



+ complex conjugate p/z

- $z = e^{j\omega} \rightarrow z^{-1} = e^{-j\omega}$ also on u.circle...



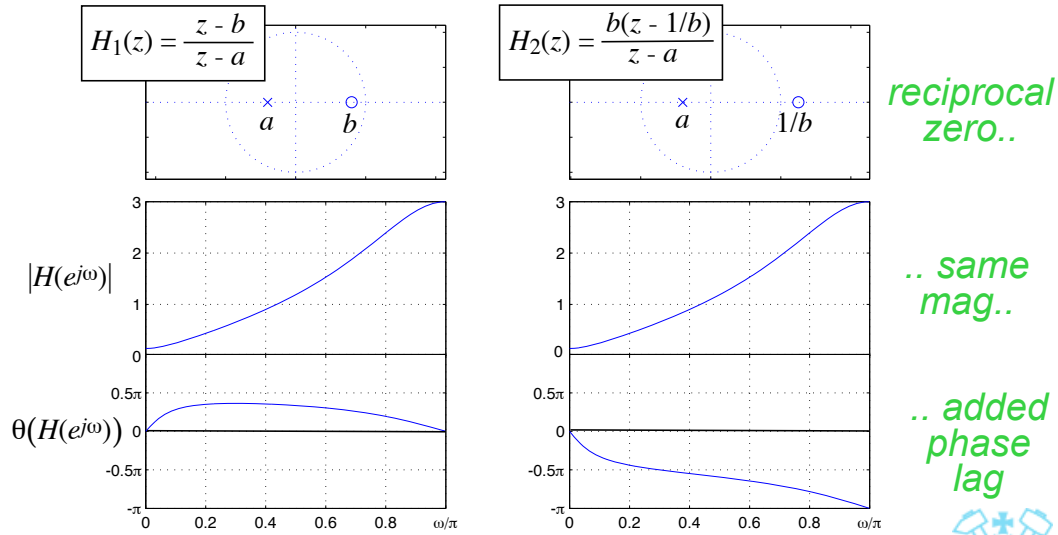
2. Minimum/Maximum Phase

- In AP filters, **reciprocal roots** have..
 - **same** effect on **magnitude** (modulo const.)
 - **different** effect on **phase**
 - In normal filters, can try **substituting reciprocal roots**
 - reciprocal of stable **pole** will be unstable ✗
 - reciprocals of **zeros**?
- **Variants** of filters with **same** magnitude response, **different** phase



Minimum/Maximum Phase

■ Hence:



Minimum/Maximum Phase

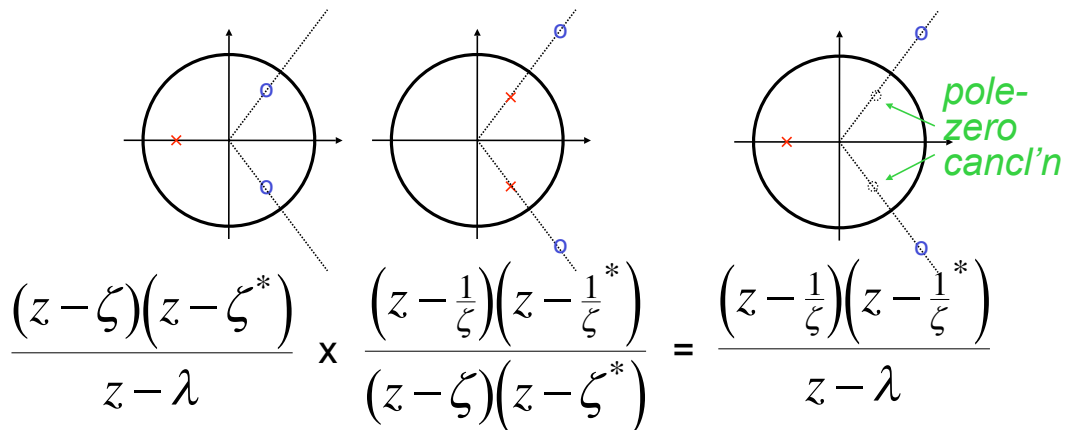
- For a given magnitude response
 - All zeros *inside* u.circle → **minimum phase**
 - All zeros *outside* u.c. → **maximum phase**
(greatest phase dispersion for that order)
 - Otherwise, **mixed phase**
- i.e. for a given magnitude response several filters & phase fns are possible; **minimum phase** is canonical, 'best'



Minimum/Maximum Phase

- Note:

Min. phase + Allpass = Max. phase

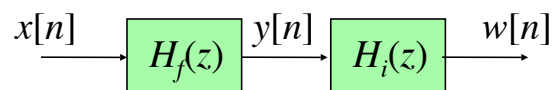


Inverse Systems

- $h_i[n]$ is called the inverse of $h_f[n]$ iff

$$h_i[n] \otimes h_f[n] = \delta[n]$$

- Z-transform: $H_f(e^{j\omega}) \cdot H_i(e^{j\omega}) = 1$



$$W(z) = H_i(z)Y(z) = H_i(z)H_f(z)X(z) = X(z)$$

$$\Rightarrow w[n] = x[n]$$

- i.e. $H_i(z)$ recovers $x[n]$ from o/p of $H_f(z)$



Inverse Systems

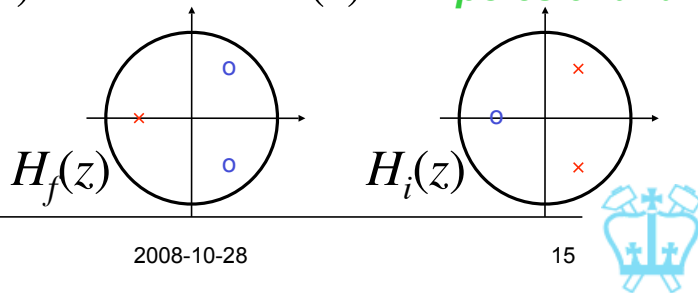
- What is $H_i(z)$? $H_i(z)H_f(z) = 1$
 $\Rightarrow H_i(z) = 1/H_f(z)$

- $H_i(z)$ is reciprocal polynomial of $H_f(z)$

$$H_f(z) = \frac{P(z)}{D(z)} \Rightarrow H_i(z) = \frac{D(z)}{P(z)}$$

poles of fwd \rightarrow zeros of bwd
 zeros of fwd \rightarrow poles of bwd

- Just swap poles and zeros:



Inverse Systems

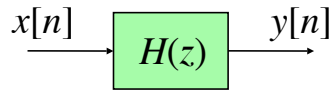
When does $H_i(z)$ exist?

- Causal+stable \rightarrow all $H_i(z)$ **poles** inside u.c.
 \rightarrow all **zeros** of $H_f(z)$ must be inside u.c.
 $\rightarrow H_f(z)$ must be **minimum phase**
- $H_f(z)$ zeros **outside** u.c. \rightarrow unstable $H_i(z)$
- $H_f(z)$ zeros **on** u.c. \rightarrow unstable $H_i(z)$

$$H_i(e^{j\omega}) = 1/H_f(e^{j\omega}) = 1/0|_{\omega=\zeta}$$

\rightarrow **only invert if min.phase, $\Rightarrow H_f(e^{j\omega}) \neq 0$**

System Identification



- **Inverse filtering** = given y and H , find x
- **System ID** = given y (and $\sim x$), find H
- Just run convolution backwards?

$$y[n] = \sum_{k=0}^{\infty} h[k]x[n-k]$$

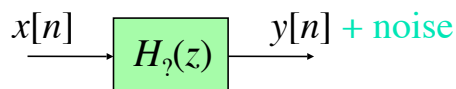
$$\Rightarrow y[0] = h[0]x[0] \rightarrow h[0]$$

$$y[1] = h[0]x[1] + h[1]x[0] \rightarrow h[1]...$$

*deconvolution
but: errors
accumulate*



System Identification



- Better approach uses **correlations**;
Cross-correlate input and output:

$$r_{xy}[l] = y[l] \otimes x[-l] = h_2[l] \otimes x[l] \otimes x[-l]$$

$$= h_2[l] \otimes r_{xx}[l]$$

- If r_{xx} is 'simple', can recover $h_2[n]...$
- e.g. (pseudo-) white noise:

$$r_{xx}[l] \approx \delta[l] \Rightarrow h_2[n] \approx r_{xy}[l]$$



System Identification

- Can also work in frequency domain:

$$S_{xy}(z) = H(z) \cdot S_{xx}(z) \leftarrow \text{make a const.}$$

- $x[n]$ is not observable $\rightarrow S_{xy}$ unavailable, but $S_{xx}(e^{j\omega})$ may still be known, so:

$$\begin{aligned} S_{yy}(e^{j\omega}) &= Y(e^{j\omega})Y^*(e^{j\omega}) \\ &= H(e^{j\omega})X(e^{j\omega})H^*(e^{j\omega})X^*(e^{j\omega}) \\ &= |H(e^{j\omega})|^2 \cdot S_{xx}(e^{j\omega}) \end{aligned}$$

- Use e.g. min.phase to rebuild $H(e^{j\omega})$...



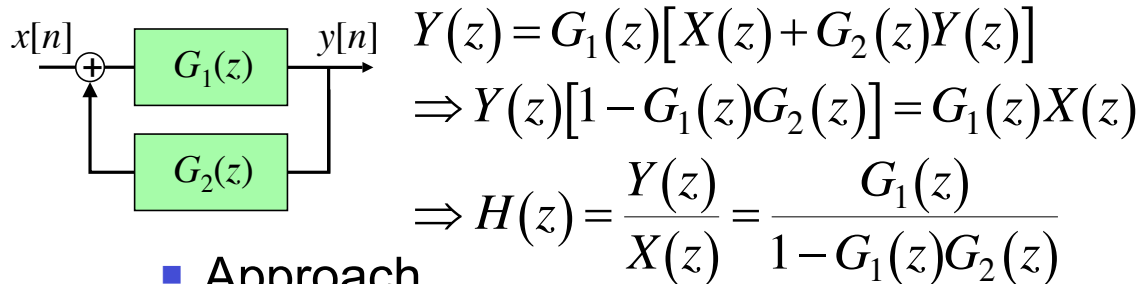
3. Filter Structures

- Many different implementations, representations of same filter
- Different costs, speeds, layouts, noise performance, ...



Block Diagrams

- Useful way to illustrate implementations
- Z-transform helps analysis:

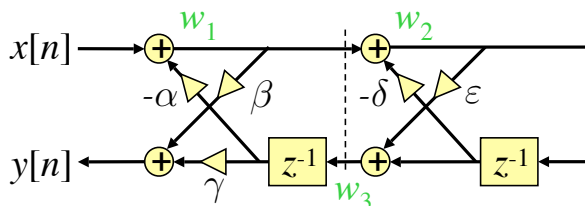


- Approach
 - Output of summers as dummy variables
 - Everything else is just multiplicative



Block Diagrams

- More complex example:



$$Y = \gamma z^{-1} W_3 + \beta W_1$$

$$\Rightarrow \frac{Y}{X} = \frac{\beta + z^{-1}(\beta\delta + \gamma\epsilon) + z^{-2}(\gamma)}{1 + z^{-1}(\delta + \alpha\epsilon) + z^{-2}(\alpha)}$$

*stackable
2nd order section*

$$W_1 = X - \alpha z^{-1} W_3$$

$$W_2 = W_1 - \delta z^{-1} W_2$$

$$W_3 = z^{-1} W_2 + \epsilon W_2$$

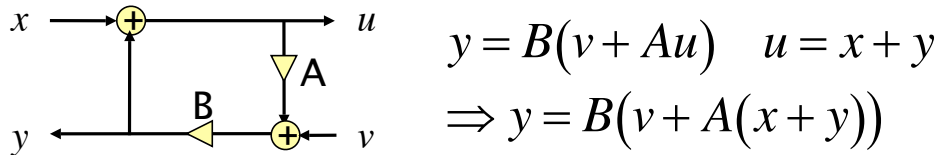
$$W_2 = \frac{W_1}{1 + \delta z^{-1}}$$

$$W_3 = \frac{(z^{-1} + \epsilon) W_1}{1 + \delta z^{-1}}$$



Delay-Free Loops

- Can't have them!

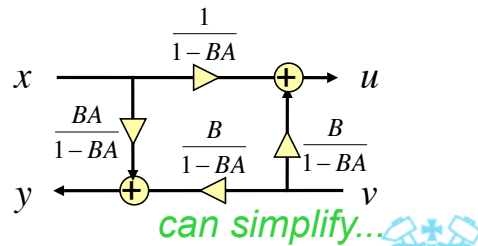


- At time $n = 0$, setup inputs x and v ; need u for y , also y for u → **can't calculate**

- Algebra:

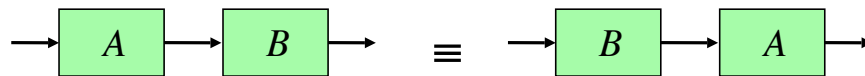
$$y(1 - BA) = Bv + BAx$$

$$\Rightarrow y = \frac{Bv + BAx}{1 - BA}$$

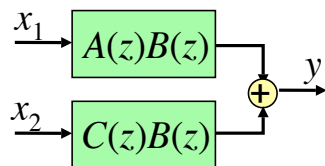


Equivalent Structures

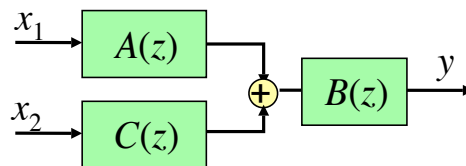
- Modifications to block diagrams that do not change the filter
- e.g. **Commutation** $H = AB = BA$



- Factoring** $AB + CB = (A + C) \cdot B$



fewer blocks



less computation



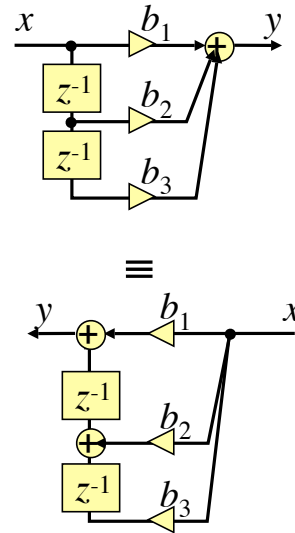
Equivalent Structures

- Transpose

- reverse paths
- adders ↔ nodes
- input ↔ output

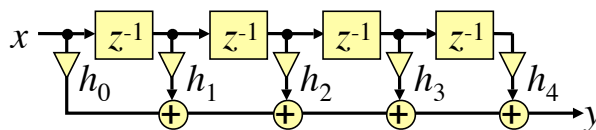
$$Y = b_1X + b_2z^{-1}X + b_3z^{-2}X$$

$$= b_1X + z^{-1}(b_2X + z^{-1}b_3X)$$



FIR Filter Structures

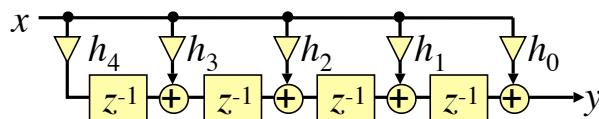
- Direct form “Tapped Delay Line”



$$y[n] = h_0x[n] + h_1x[n-1] + \dots$$

$$= \sum_{k=0}^4 h_k x[n-k]$$

- Transpose



- Re-use delay line if several inputs x_i for single output y ?

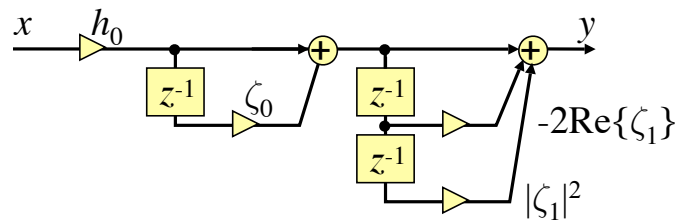


FIR Filter Structures

- Cascade

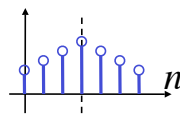
- factored into e.g. 2nd order sections

$$\begin{aligned}
 H(z) &= h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} \\
 &= h_0 (1 - \zeta_0 z^{-1})(1 - \zeta_1 z^{-1})(1 - \zeta_1^* z^{-1}) \\
 &= h_0 (1 - \zeta_0 z^{-1})(1 - 2 \operatorname{Re}\{\zeta_1\} z^{-1} + |\zeta_1|^2 z^{-2})
 \end{aligned}$$



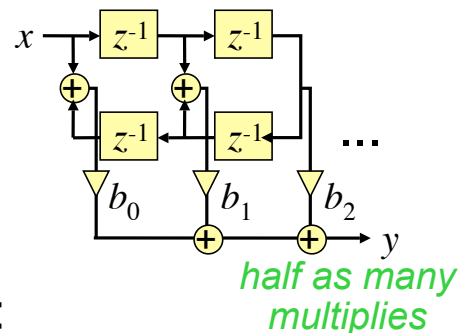
FIR Filter Structures

- Linear Phase:



Symmetric filters with $h[n] = (-)h[N - n]$

$$\begin{aligned}
 y[n] &= b_0(x[n] + x[n - 4]) \\
 &+ b_1(x[n - 1] + x[n - 3]) \\
 &+ b_2 x[n - 2]
 \end{aligned}$$



- Also Transpose form:

gains first, feeding folded delay/sum line

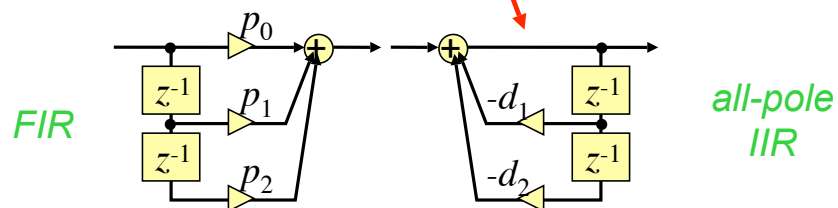


IIR Filter Structures

- IIR: numerator + denominator

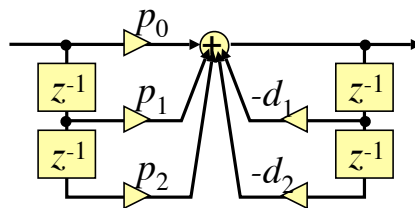
$$H(z) = \frac{p_0 + p_1z^{-1} + p_2z^{-2} + \dots}{1 + d_1z^{-1} + d_2z^{-2} + \dots}$$

$$= P(z) \cdot \frac{1}{D(z)}$$

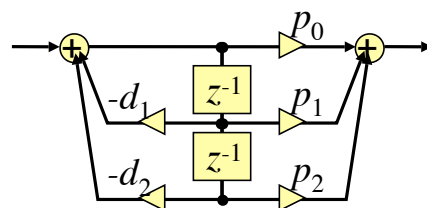


IIR Filter Structures

- Hence, Direct form I



- Commutation → Direct form II (DF2)



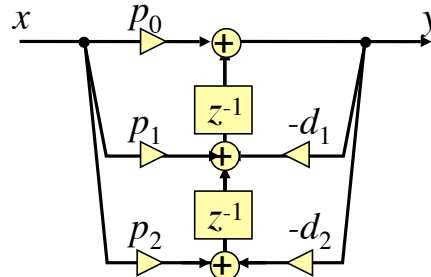
• same signal
∴ delay lines merge

• “canonical”
= min. memory usage



IIR Filter Structures

- Use Transpose on FIR/IIR/DF2



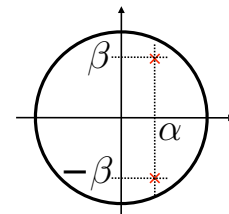
- “Direct Form II Transpose”



Factored IIR Structures

- Real-output filters have conjugate-symm roots:

$$H(z) = \frac{1}{(1 - (\alpha + j\beta)z^{-1})(1 - (\alpha - j\beta)z^{-1})}$$



- Can always group into 2nd order terms with real coefficients:

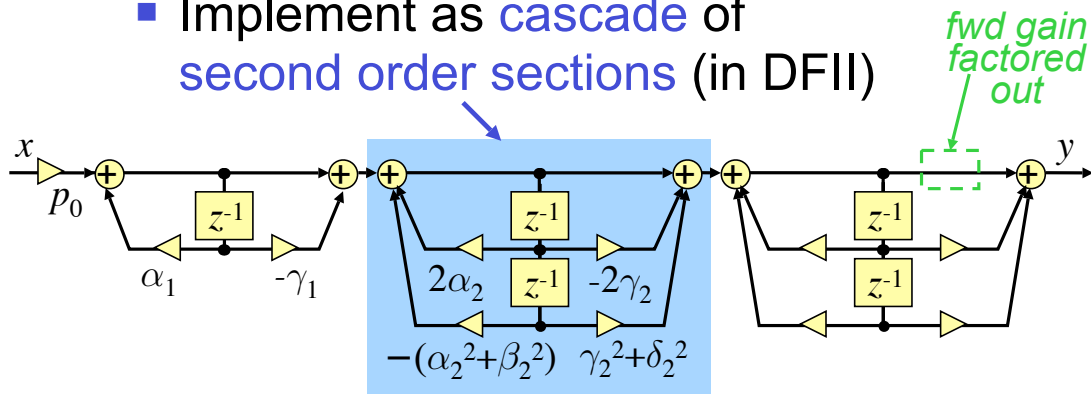
$$H(z) = \frac{p_0(1 - \gamma_1 z^{-1})(1 - 2\gamma_2 z^{-1} + (\gamma_2^2 + \delta_2^2)z^{-2}) \dots}{(1 - \alpha_1 z^{-1})(1 - 2\alpha_2 z^{-1} + (\alpha_2^2 + \beta_2^2)z^{-2}) \dots}$$

real root →



Cascade IIR Structure

- Implement as cascade of second order sections (in DFII)



- Second order sections (SOS):
 - modular - any order from optimized block
 - well-behaved, real coefficients (sensitive?)



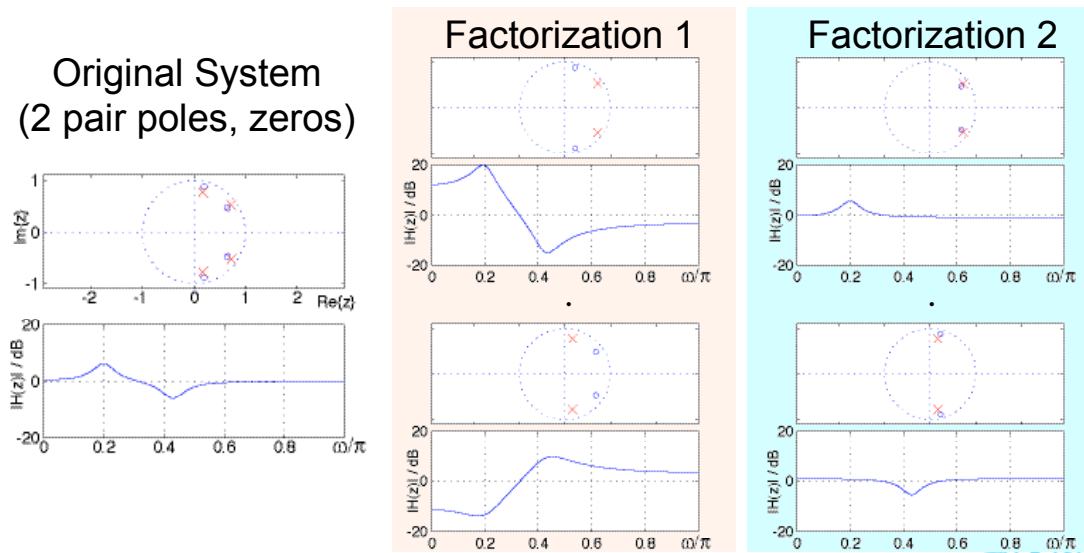
Second-Order Sections

- 'Free' choices:
 - grouping of pole pairs with zero pairs
 - order of sections
- Optimize numerical properties:
 - avoid **very large** values (overflow)
 - avoid **very small** values (quantization)
- e.g. Matlab's `zp2sos`
 - attempt to put 'close' roots in same section
 - intersperse gain & attenuation?



Second Order Sections

- Factorization affects intermediate values



Dan Ellis

2008-10-28

35



Parallel IIR Structures

- Can express $H(z)$ as sum of terms (**IZT**)

$$H(z) = \text{consts} + \sum_{\ell=1}^N \frac{\rho_{\ell}}{1 - \lambda_{\ell} z^{-1}} \quad \rho_{\ell} = (1 - \lambda_{\ell} z^{-1}) F(z) \Big|_{z=\lambda_{\ell}}$$

- Or, second-order terms:

$$H(z) = \gamma_0 + \sum_k \frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}}$$

- Suggests **parallel** realization...

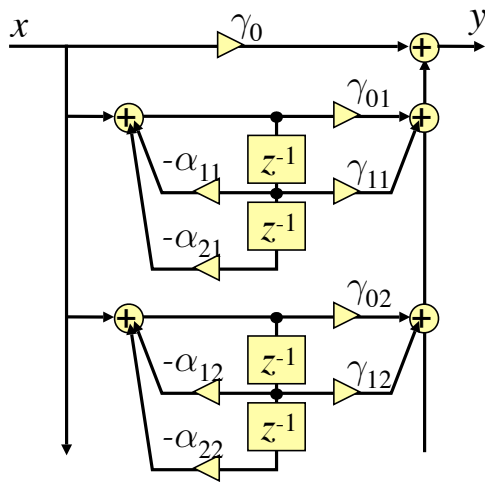
Dan Ellis

2008-10-28

36



Parallel IIR Structures



- Sum terms become parallel paths
- **Poles** of each SOS are from full TF
- System **zeros** arise from output sum
- Why do this?
 - stability/sensitivity
 - reuse common terms

