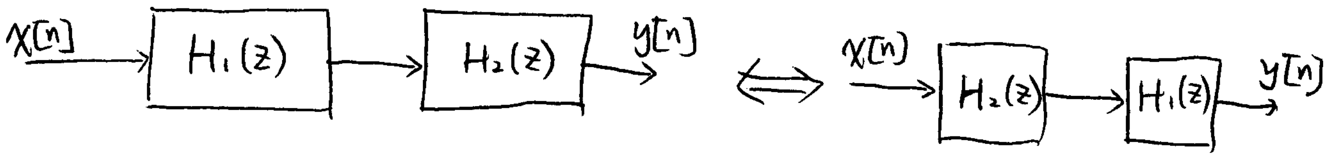
ECE 272/472. Lecture 8: LTI Filter Structures

Series

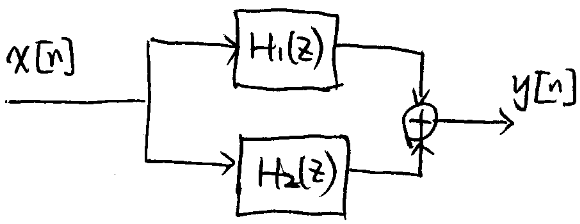


$$Y(z) = X(z) \cdot H_1(z) \cdot H_2(z)$$

$$Y(z) = X(z) \cdot H_2(z) \cdot H_1(z)$$

$$H(z) = H_1(z) H_2(z) = H_2(z) H_1(z)$$

Parallel:



$$Y(z) = X(z) H_1(z) + X(z) H_2(z)$$

$$H(z) = H_1(z) + H_2(z)$$

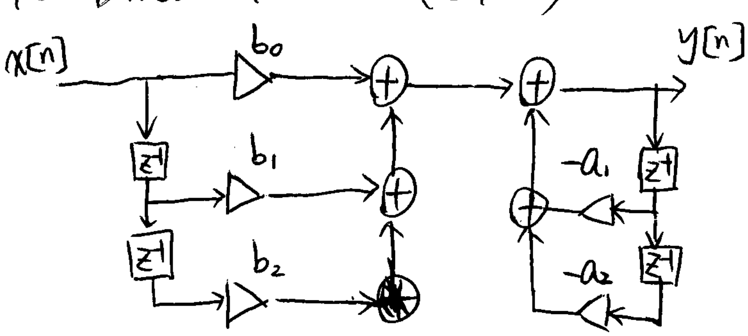
Remember a general causal LTI filter

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{l=1}^{N} a_l y[n-l]$$

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{l=1}^{N} a_l z^{-l}}$$

How to Implement the filter structure?

1. Direct-Form I (DF-I)

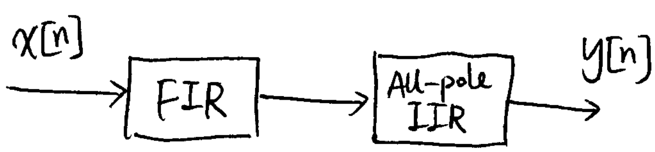


Here $M = N = 2$.

Properties:

① No internal overflow issues
Reason: ~~there is only~~ All summations are performed at the same place.

② More delay modules than necessary.
— 4 delays are used here, but the difference equation is only of order 2.

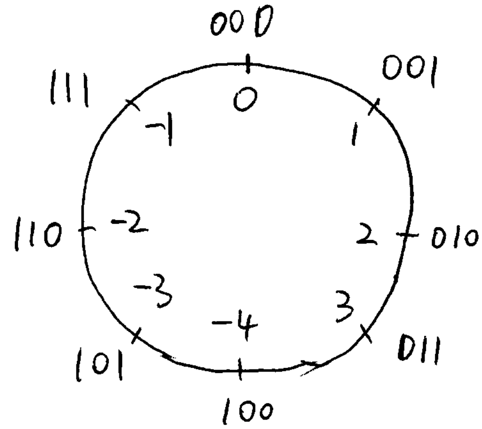


Two's compliment wrap-around

3-bit signed fixed-point arithmetic

available numbers

Decimal	Binary
-4	100
-3	101
-2	110
-1	111
0	000
1	001
2	010
3	011



Suppose we are performing summation $\underbrace{3+3}_{\text{internal overflow}} - 4 = 2$

as $3+3=6$ outside of data range

However, $3 + 3 = 6$

$011 + 011 = 110 \Rightarrow -2$

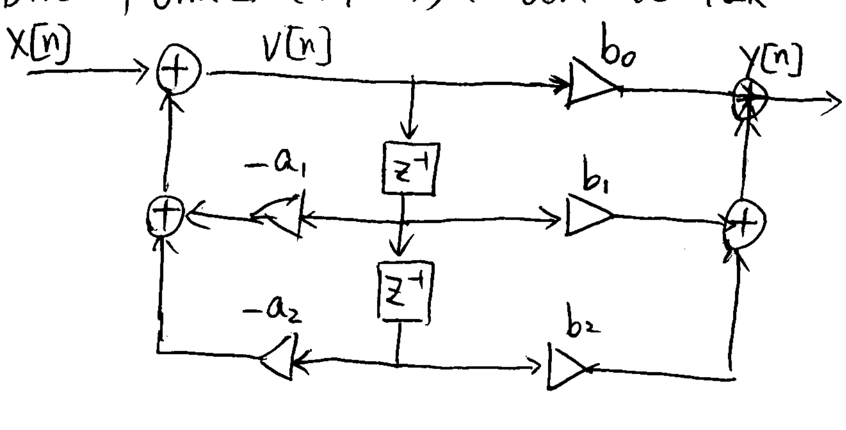
$6 - 4 = 2$

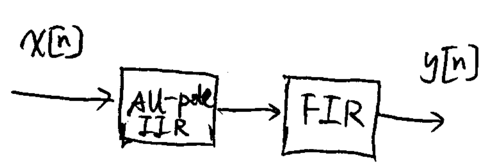
$110 + 100 = 010 \Rightarrow 2$

Nice!

Although internal number can be outside of data range, as long as final result is within the range, internal overflow doesn't matter!

2. Direct Form II (DF-II): Commute FIR and All-pole IIR





Properties: 1 Internal overflow happens (Actually may happen often!)

Reason: Summations are not in the same place.

v[n] may overflow.

In fact, since IIR $\boxed{\text{all-pole}}$ filter tends to amplify the signal $\left(\begin{array}{l}\text{poles amplify signal} \\ \text{zeros attenuate signal}\end{array}\right)$, so v[n] is easy to overflow!

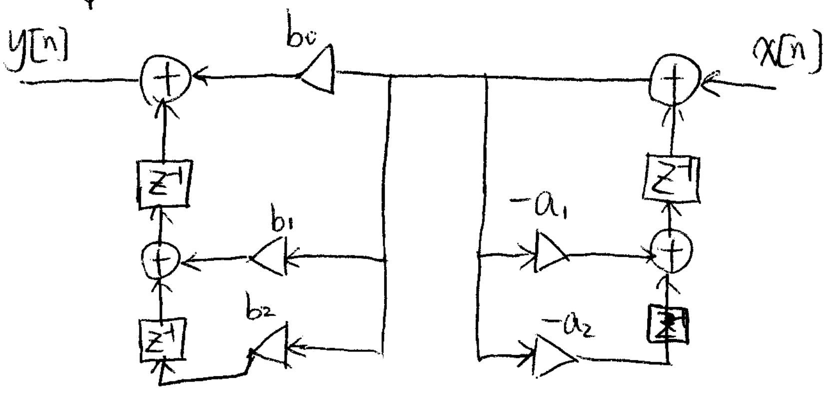
2 There are minimum number of delay modules (i.e., canonical w.r.p. delay), as IIR and FIR sections share delay modules.

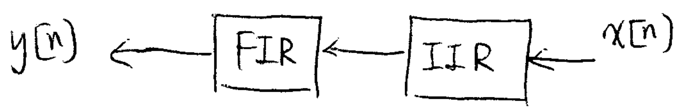
1 ⇒ requires more memory to prevent internal delay.

2 ⇒ ---- Less --- for delay modules.

⇒ Overall, the advantage of DF-2 over DF-I in terms of memory is not significant for fixed-point.

~~3~~ 3. Transposed Direct-Form I (TDF-I)





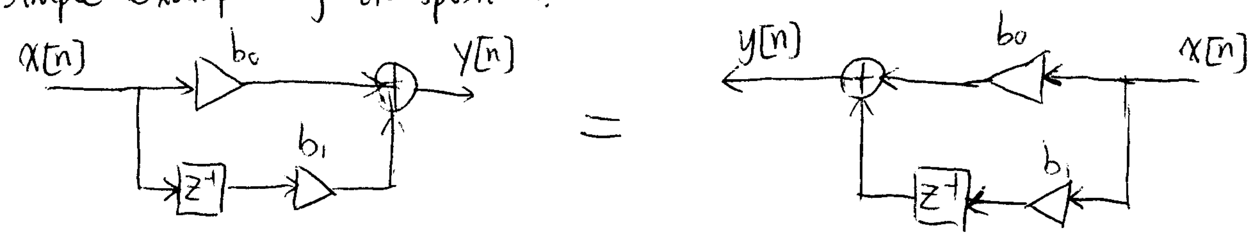
Manson's gain theorem:

Just reverse signal flow, and branching points ⟷ summers, gives equivalent filter.

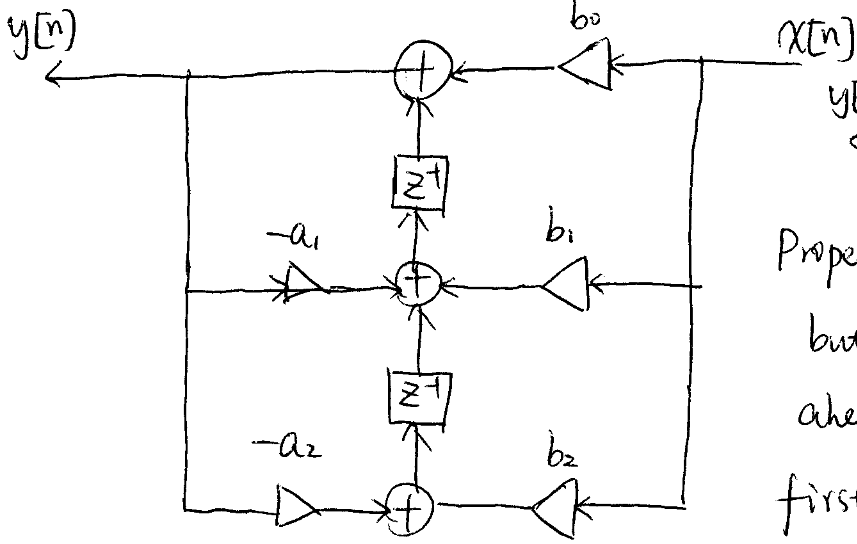
Properties: 1 Internal overflow. Severer as "first IIR then FIR"

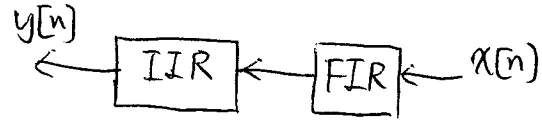
2 More delay modules than necessary

Simple example of transposition:



4. Transposed Direct-Form II (TDF − II) : Nice structure.





Properties: ① Overflow still may happen but less likely, as zeros are ahead of poles, so signals are first attenuated.

② Canonical delay.

Problems of direct-forms of high-order filters:

— Poles and zeros are very sensitive to filter coefficients, so it's hard to control poles and zeros by changing filter coefficients.

— In time varying cases, we do want to chang poles and zeros often.

— Overflow issues.

Ideas to address the problems: Use series or parallel second-order sections to implement high-order filters.

Remember general causal LTI filter transfer function:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{l=1}^{N} a_l z^{-l}} = b_0 \frac{\prod_{k=1}^{M} (1 - c_k z^{-1})}{\prod_{k=1}^{N} (1 - d_l z^{-1})}$$

(Fundamental Theorem of Algebra).

For real filters (where coefficients a_k and b_k are all real), poles and zeros can still be complex, but form conjugate pairs.

If we pair those pairs, we would get some multiplications of 2-order filter transfer functions (and possibly some 1-order functions).

This gives us the series ~~st~~ 2nd-order sections implementation!

For example: $$H(z)=\frac{1+z^{-5}}{1+0.9z^{-5}}$$ 5th-order filter

$$=\left(\frac{1+0.61803z^{-1}+z^{-2}}{1+0.60515z^{-1}+0.95873z^{-2}}\right)\left(\frac{1-1.61803z^{-1}+z^{-2}}{1-1.58430z^{-1}+0.95873z^{-2}}\right)\left(\frac{1+z^{-1}}{1+0.97915z^{-1}}\right)$$

2nd-order (bi-quad)

1st-order.

In matlab, use function ~~tf~~ tf2sos to do this, or zp2sos

(transfer function) (second order series.) (zero-pole)

On the other hand, we can write $H(z)$ as.

$$H(z)=F(z)+\underbrace{z^{-(K+1)}}_{\text{delay}}\sum_{i=1}^{N_P}\sum_{k=1}^{m_K}\frac{r_{i,k}}{(1-d_i z^{-1})^k}$$ ← residue

FIR filter with order K

$K=M-N\geq 0.$

N_p: # unique poles.

m_K: multiplicity of pole d_i

$$=\frac{\sum_{k=0}^{M} b_0 z^{-k}}{1+\sum_{l=1}^{N} a_l z^{-1}}=\frac{\sum_{k=0}^{M} b_0 z^{-k}}{\prod_{l=1}^{N}(1-d_l z^{-1})}$$

For example: $$H(z)=\frac{1}{(1-z^{-1})(1-0.5z^{-1})}$$ all-pole filter

$$=\frac{2}{1-z^{-1}}-\frac{1}{1-0.5z^{-1}}$$ parallel of two 1st order filters

$$H(z)=\frac{1+0.5^3z^{-3}}{1+0.9^5z^{-5}}=\frac{0.16571}{1+0.9z^{-1}}+$$

delay of 2 samples

$$H(z)=\frac{2+6z^{-1}+6z^{-2}+2z^{-3}}{1-2z^{-1}+z^{-2}}=(2+10z^{-1})+z^{-2}\left[\frac{8}{1-z^{-1}}+\frac{16}{(1-z^{-1})^2}\right]$$

FIR with order 1.

parallel of IIR filters

Impulse response of the IIR ~~f~~ sections start after the FIR section has died out, due to the delay.