Fast convolution using FFT

Say we want to compute $\{x[n]\}_{n=0}^{L-1} * \{h[n]\}_{n=0}^{M-1}$ (Result should have length $L+M-1$)

Zero-pad to length N

$$\{x[n]\}_{n=0}^{N-1} \xrightarrow[\text{(FFT)}]{\text{DFT}} \{X[k]\}_{n=0}^{N-1}$$

$$\{h[n]\}_{n=0}^{N-1} \xrightarrow[\text{(FFT)}]{\text{DFT}} \{H[k]\}_{n=0}^{N-1}$$

multiply

$$? \xleftarrow[\text{(IFFT)}]{\text{IDFT}} \{Y[k] = H[k]X[k]\}_{n=0}^{N-1}$$

DFT property: $x[n] \circledast_N h[n] \overset{\text{DFT}}{\longleftrightarrow} X[k]H[k]$

$$\sum_{m=0}^{N-1} x[m]\, h[\langle n-m\rangle_N]$$

$\rightarrow$ this is not what we wanted!

How can we get (linear) convolution from circular convolution?

<u>Fact</u> : If we zero-pad $x[n]$ and $h[n]$ to $N = L+M-1$,

cyclic convolution = linear convolution

Why? $\{x_{zp}[n]\} = \{x[0], \dots, x[L-1], \underbrace{0, \dots, 0}_{M-1}\}$, $\{h_{zp}[n]\} = \{h[0], \dots, h[M-1], \underbrace{0, \dots, 0}_{L-1}\}$

cyclic convolution : $(x_{zp} \circledast h_{zp})[n] = \sum_{m=0}^{L+M-2} x_{zp}[m]\, h_{zp}[\langle n-m \rangle_N]$

$$= \sum_{m=0}^{L-1} x[m]\, h_{zp}[\langle n-m \rangle_N] = \sum_{m=0}^{L-1} x[m]\, h[n-m] = (x*h)[n]$$

$$h_{zp}[\langle n-m \rangle_N] = \begin{cases} h[n-m] & \text{if } 0 \le m \le n \\ \underset{\substack{\| \\ 0}}{h_{zp}[N+n-m]} & n < m < L-1 \end{cases}$$

(because of zero padding

cyclic convolution becomes linear convolution

Great! Now we can compute fast convolutions!

Fast FIR filtering

↳ finite impulse response $\Rightarrow \{h[n]\}_{n=0}^{M-1}$ ..

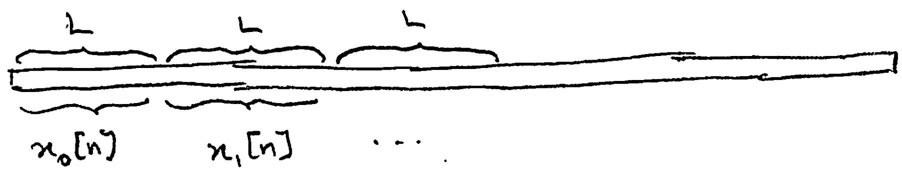
$$\{x[n]\}_{n=0}^{N'-1} \longrightarrow \boxed{h} \longrightarrow \{y[n]\} = ?$$

often $N' \gg M$ and $x[n]$ is streamed (e.g. digital comm)

$\Rightarrow$ long delay in computing convolution.

Solution: Break $x[n]$ into blocks of length L.

$x[n]$



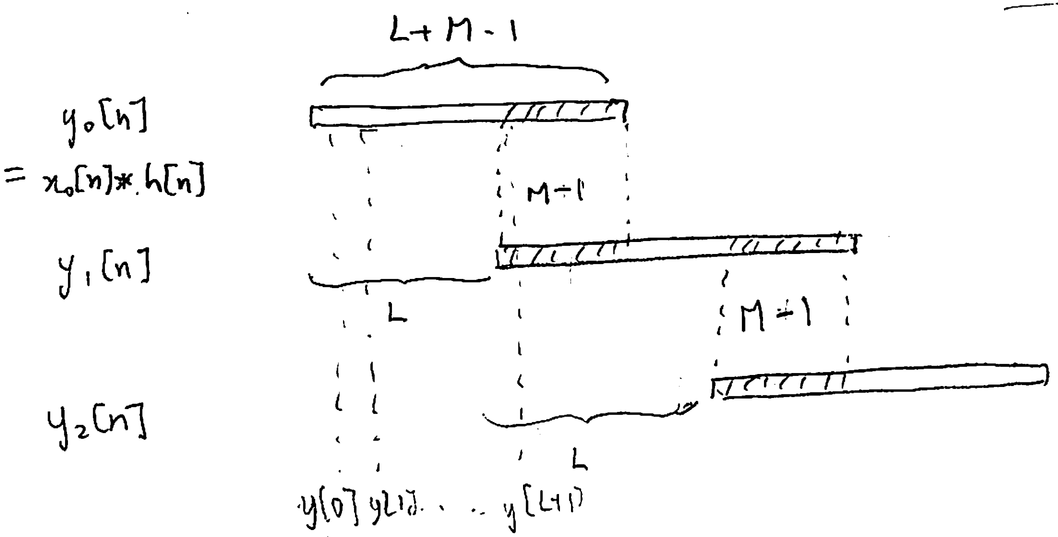
$$x_i[n] = \begin{cases} x[iL+n], & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

$$x[n] = \sum_{i=0}^{\frac{N'}{L}-1} x_i[n-iL] \rightarrow y[n] = x[n] * h[n] = \sum_{i=0}^{\frac{N'}{L}-1} x_i[n-iL] * h[n] = \sum_{i=0}^{\frac{N'}{L}-1} y_i[n-iL]$$

$y_i[n] \triangleq x_i[n] * h[n]$ time invariance

We can compute each $y_i[n]$ fast (with our previous approach)

Finally, we compute $y[n]$ by "overlap - and - add":



- just need to add overlapping parts
- we don't need to wait for all y_i's to computed to start computing $y[n]$

An alternative approach is called "overlap and save"

- choose overlapping blocks from $x[n]$
- cut a part of the resulting convolutions and stitch them together

Lecture 19 - Digital Filter Structures

$$x \longrightarrow \boxed{H(z)} \longrightarrow y$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}} = \frac{B(z)}{A(z)}$$

LCCDE: $y[n] = b_0 x[n] + b_1 x[n-1] + \ldots + b_M x[n-M]$
$\quad - a_1 y[n-1] - \ldots - a_N y[n-N]$

FIR (finite impulse response): all $a_1, \ldots, a_N = 0$

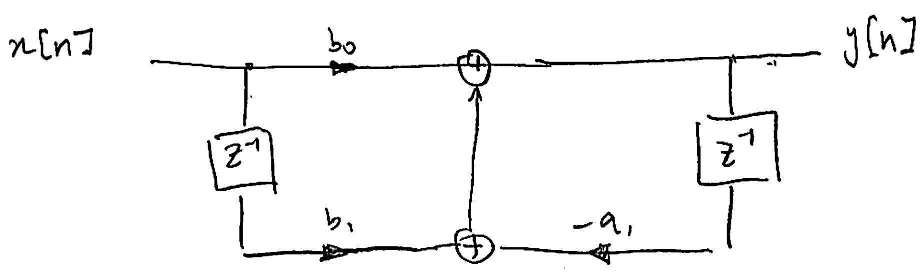
$H(z) = B(z) \Rightarrow h[n] = \{b_0, \ldots, b_M\}$ (finite)

IIR (infinite impulse response): some $a_i \neq 0$ (after possible cancelations)

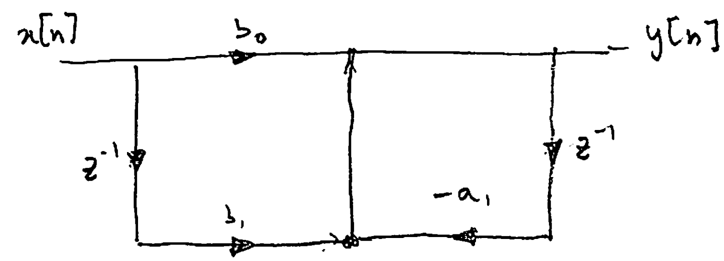
e.g. $H(z) = \dfrac{1 - z^{-2}}{1 - z^{-1}} = \dfrac{(1 + z^{-1})\cancel{(1 - z^{-1})}}{\cancel{(1 - z^{-1})}} \Rightarrow$ FIR

<u>structure</u> : block diagram (or flow diagram) representing the system's implementation

e.g: $y[n] = b_0 x[n] + b_1 x[n-1] - a_1 y[n-1]$



(block diagram)



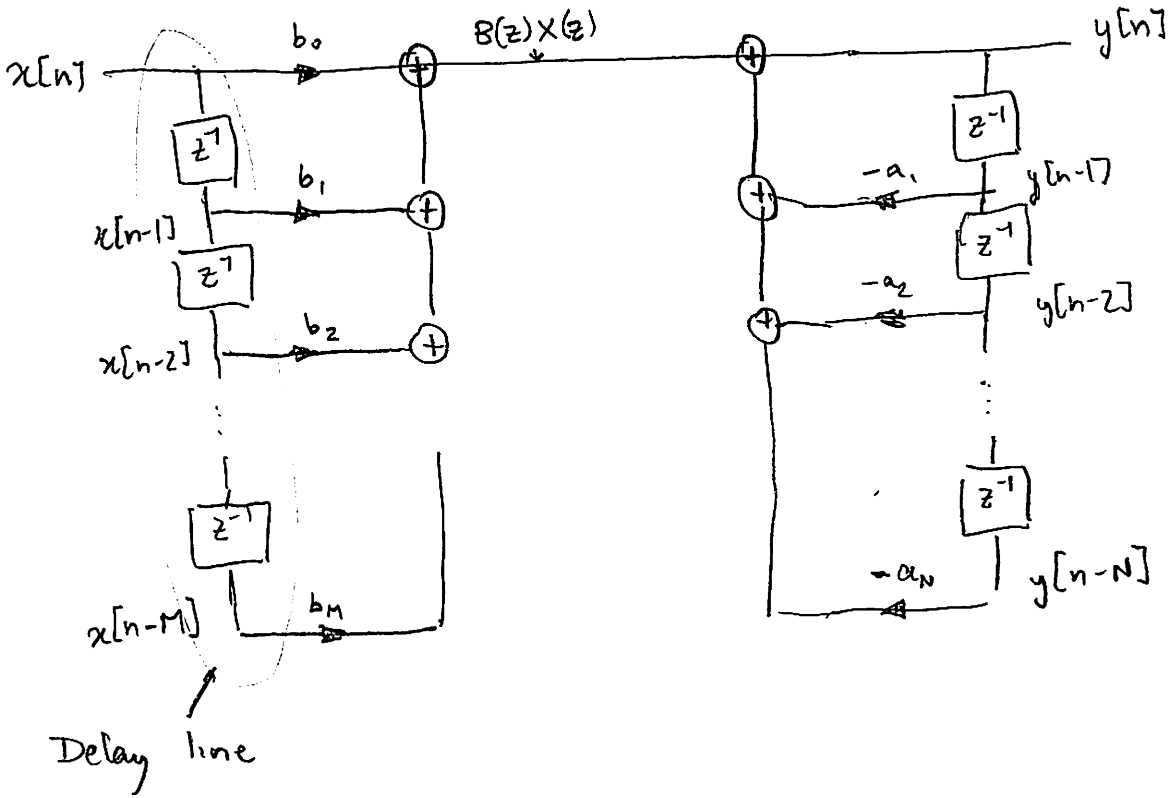
(flow diagram)

Same LCCDE system will have different structures

- different # of multiplication, memory units, numerical stability, ...

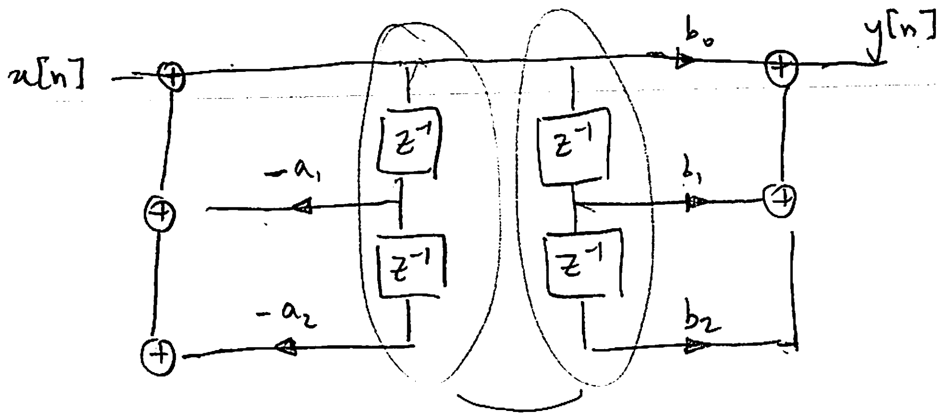
Direct Form I :

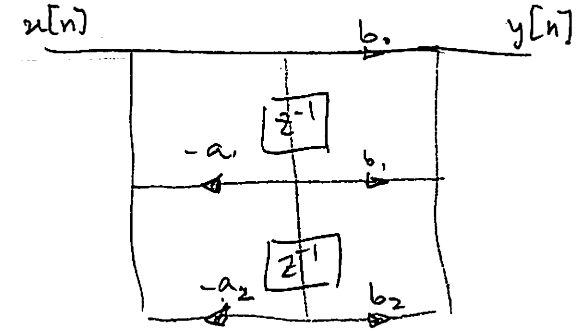
$$H(z) = \frac{b_0 + \dots + b_n z^{-M}}{1 + \dots + a_N z^{-N}} = B(z) \cdot \frac{1}{A(z)}$$



Direct Form II : $H(z) = \frac{1}{A(z)} \cdot B(z)$

For $N = M = 2$





combine!

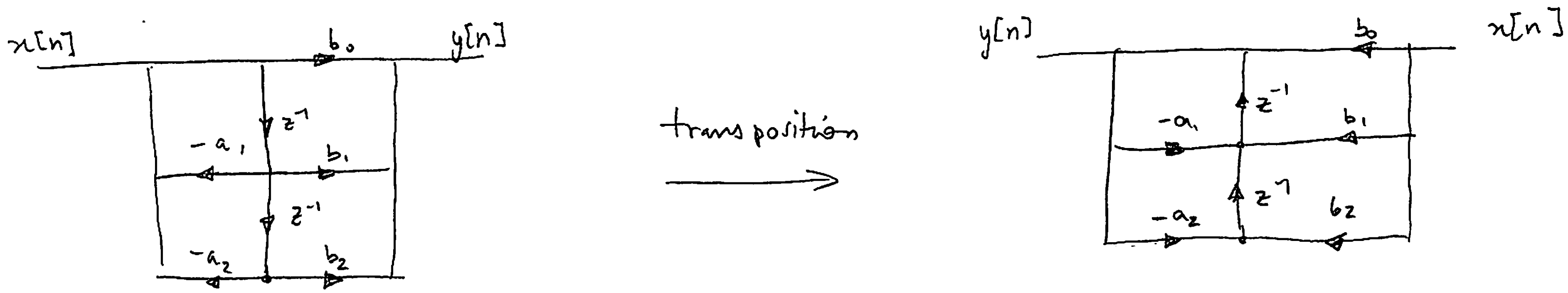
just need one delay line!

Transposition Theorem

- An equivalent structure can be obtained by:
  - reversing all flows
  - swapping adders and splitters
  - swapping $x[n]$ and $y[n]$

Ex: $H(z) = \dfrac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$

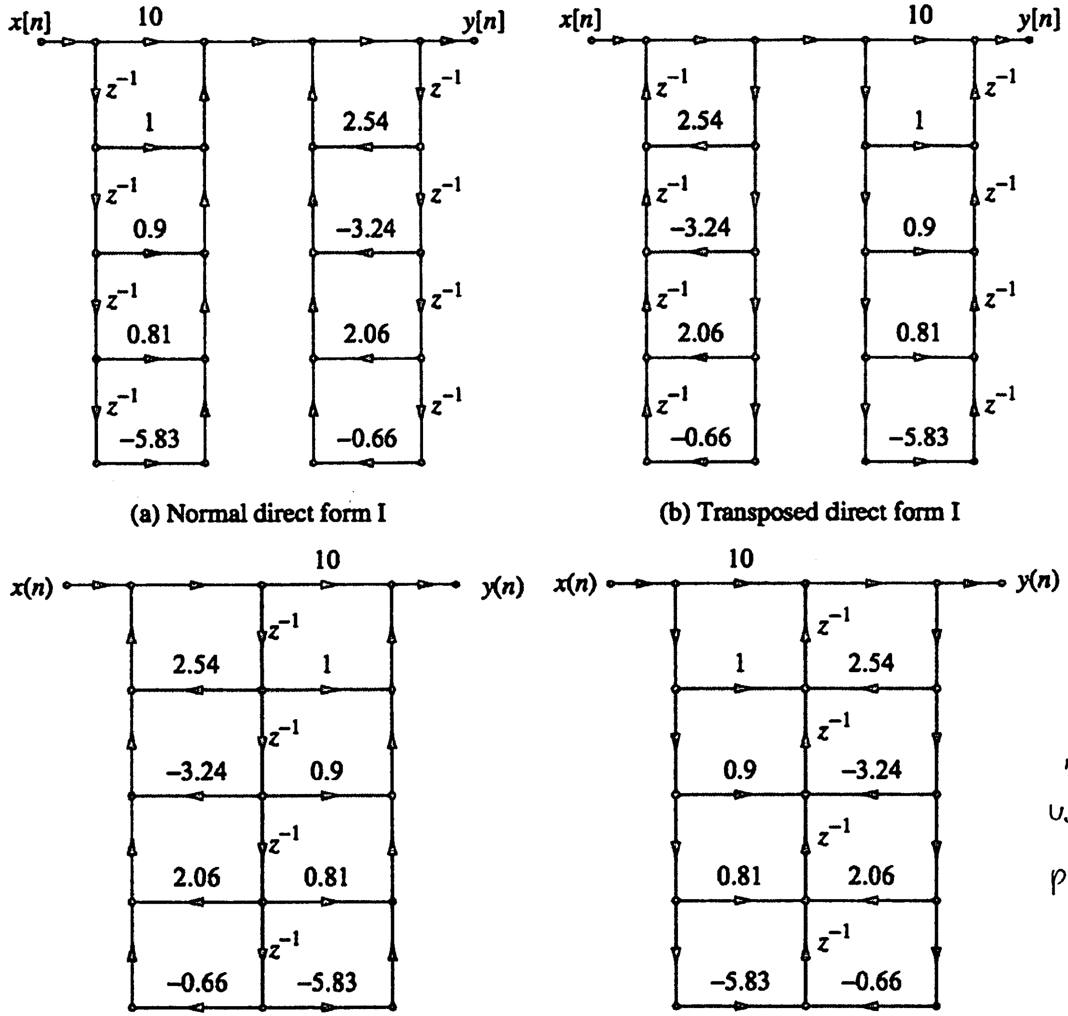
Direct Form II



(typically we flip diagram so that $x[n]$ is on the left)

⇒ we end up with 4 different structures

$$H(z) = \frac{10 + z^{-1} + 0.9z^{-2} + 0.8z^{-3} - 5.8z^{-4}}{1 - 2.54z^{-1} + 3.24z^{-2} - 2.06z^{-3} + 0.66z^{-4}}$$



(a) Normal direct form I

(b) Transposed direct form I

(c) Normal direct form II

(d) Transposed direct form II

Figure 9.9 Direct form structures for the system in Example 9.2

mostly used in practice