Chapter - 6 -
Structures For Discrete-Time Systems

* Difference equation, Impulse Response ($h(n)$) and system Function ($H(z)$) are equivalent characterization of Input-output relation of LTI system.

* For Implementing LTI system characterized by diff. equation or system function by discrete-time analogue or digital Hardware ⇒ Diff. eq. or system function must be converted to an algorithm or structure that can be ~~red~~ realized.

* In this chapter → systems can be represented by structures consisting of an interconnection of the basic operations of ① addition ② Multiplication by a constant and ③ delay.

e.g.

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a z^{-1}}, \quad |z| > |a|$$

$$h(n) = b_0 a^n u(n) + b_1 a^{n-1} u(n-1)$$

$$y(n) - a y(n-1) = b_0 x(n) + b_1 x(n-1) \qquad \text{———} \quad (*)$$

⇒ Since this system has an infinite-duration impulse response, it is not possible to implement the system by discrete convolution.

However, re-writing (*) in form

$$y(n) = a y(n-1) + b_0 x(n) + b_1 x(n-1)$$

provides the basis for an algorithm for recursive computation of the output at any time (n) in terms of previous output $y(n-1)$, current Input sample $x(n)$, and previous Input sample $x(n-1)$.

(If we assume Initial-reset conditions (i.e. $x(n) = 0$ for $n<0$), then $y(n) = 0$ for $n<0$).

* Similar procedure can be applied to more general case of an N^{th} order difference equation.

* However, this algorithm is not the only computational algorithm and it is not the most preferable one.

Implementation of LTI systems

Block diagram representation of LCCDE for Implementation:

- we need storage of past sequence values (delay).
- Multiplication of delayed seq. values by the coefficients.
- Addition for adding the resulting product.

So, the basic elements for implementing LTI system are:

* Adders $\quad x_1(n) \longrightarrow \oplus \longrightarrow x_1(n) + x_2(n)$, with $x_2(n)$ into $\oplus$ (may have any no. of inputs But we use only two in the course).

* Multiplier (by a constant) $\quad x(n) \xrightarrow{a} a\,x(n)$

* Memory for storing delayed seq. values.

$$x(n) \longrightarrow \boxed{z^{-1}} \longrightarrow x(n-1)$$

* In digital implementation, delay operation can be implemented by storage Register for each unit delay. (or shift-Register).

* In analogue discrete-time implementation, delay unit is a charge storage device.

* Delay of more than one sample (M samples) can be done by cascading M unit delays. In IC-implementation, these unit delays are a shift-Reg. clocked at sampling rate of input signal.

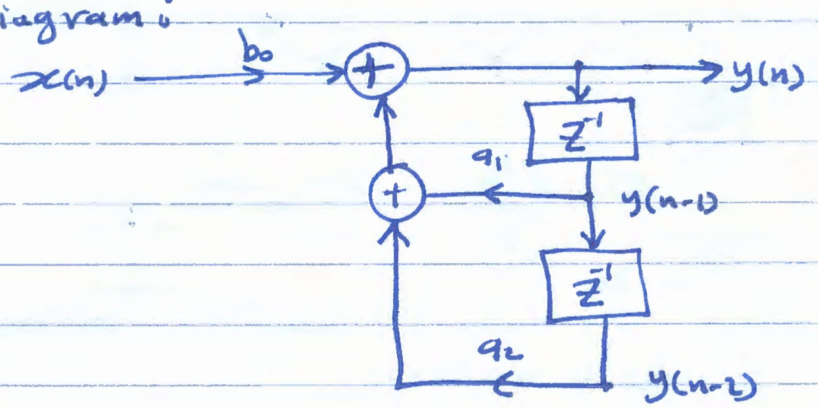
* In software implementation ⇒ M cascaded unit delays can be implemented as M consecutive memory Registers.

Example: 2nd order diff. eq.

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + b_0 x(n).$$

$$H(z) = \frac{b_0}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

Block diagram:



* Implementing system on general purpose computer or DSP chip. this network structure is the basis for the program that implement this system.

$\Rightarrow$ What we need for this implementation

* storage for delayed samples $(y(n-1), y(n-2))$
* Multiply $a_1 y(n-1)$ and $a_2 y(n-2)$ and adding them
* ADD the result to $b_0 x(n)$.

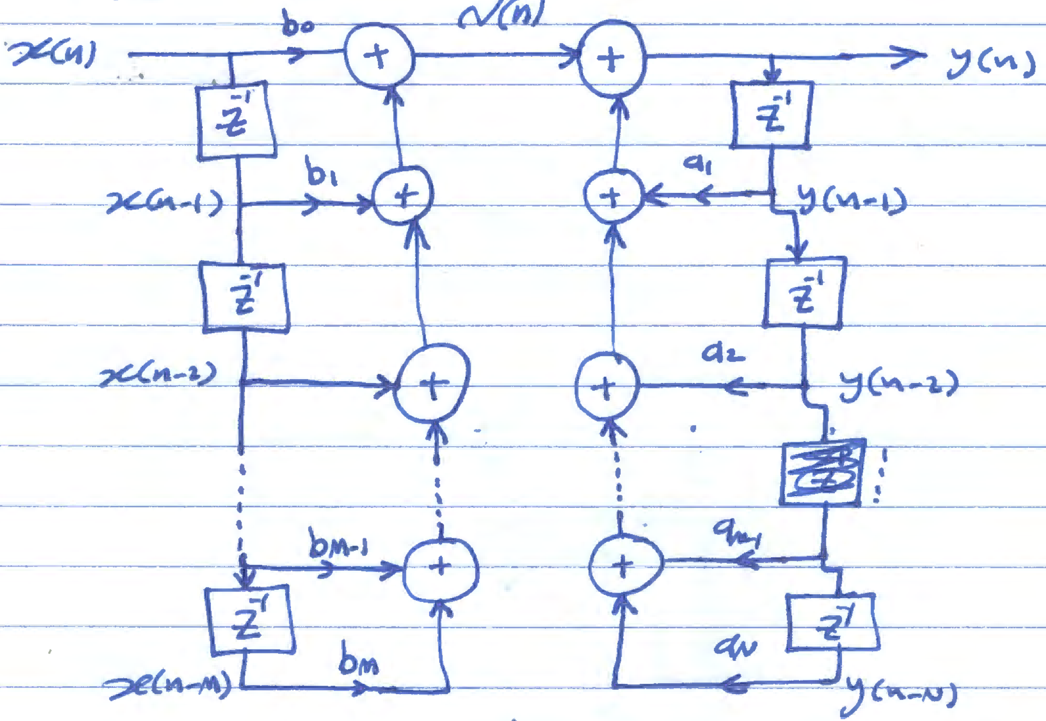
* This Example can be generalized into higher order diff. eq.

$$y(n) - \sum_{k=1}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k)$$

$$\Rightarrow H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Fig 1



Assume $N=M$

$$v(n) = \sum_{k=0}^{M} b_k\, x(n-k)$$

$$\Rightarrow y(n) = \sum_{k=1}^{N} a_k\, y(n-k) + v(n)$$

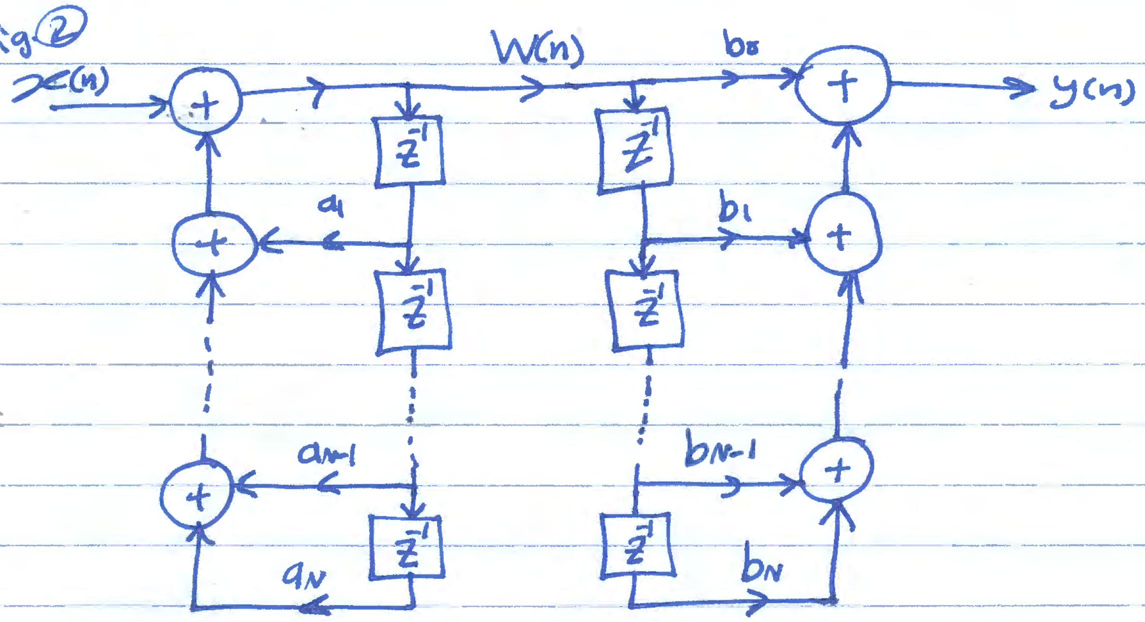
* This block diagram can be re-arranged or modified in a varity of ways without changing the overall sys. function.

→ Each arrangment represents a computational algorithm for the same system.

* This block diagram can be viewed as a cascade of two systems. ⇒ First computation of $v(n)$ from $x(n)$
second " " $y(n)$ " $v(n)$.

Each system is LTI, so the order of cascade can be reversed without afecting the overall system function.

Fig 2



N=M

* If $M \neq N$, some of a_k or b_k would be zero.

$$H(z) = H_2(z)\,H_1(z) = \left(\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}}\right)\left(\sum_{k=0}^{M} b_k z^{-k}\right)$$

$$V(z) = H_1(z)\,X(z) = \left(\sum_{k=0}^{M} b_k z^{-k}\right) X(z)$$

$$Y(z) = H_2(z)\,V(z) = \left(\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}}\right) V(z)$$

on therhand,

$$H(z) = H_1(z)\,H_2(z) = \left(\sum_{k=0}^{M} b_k z^{-k}\right)\left(\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}}\right)$$

$$W(z) = H_2(z)\,X(z) = \left(\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}}\right) X(z)$$

$$Y(z) = H_1(z)\,W(z) = \left(\sum_{k=0}^{M} b_k z^{-k}\right) W(z)$$

In Time-domain:

$$w(n) = \sum_{k=1}^{N} a_k\, w(n-k) + x(n)$$

$$y(n) = \sum_{k=0}^{M} b_k\, w(n-k)$$

* Differences between Fig 1 and Fig 2:

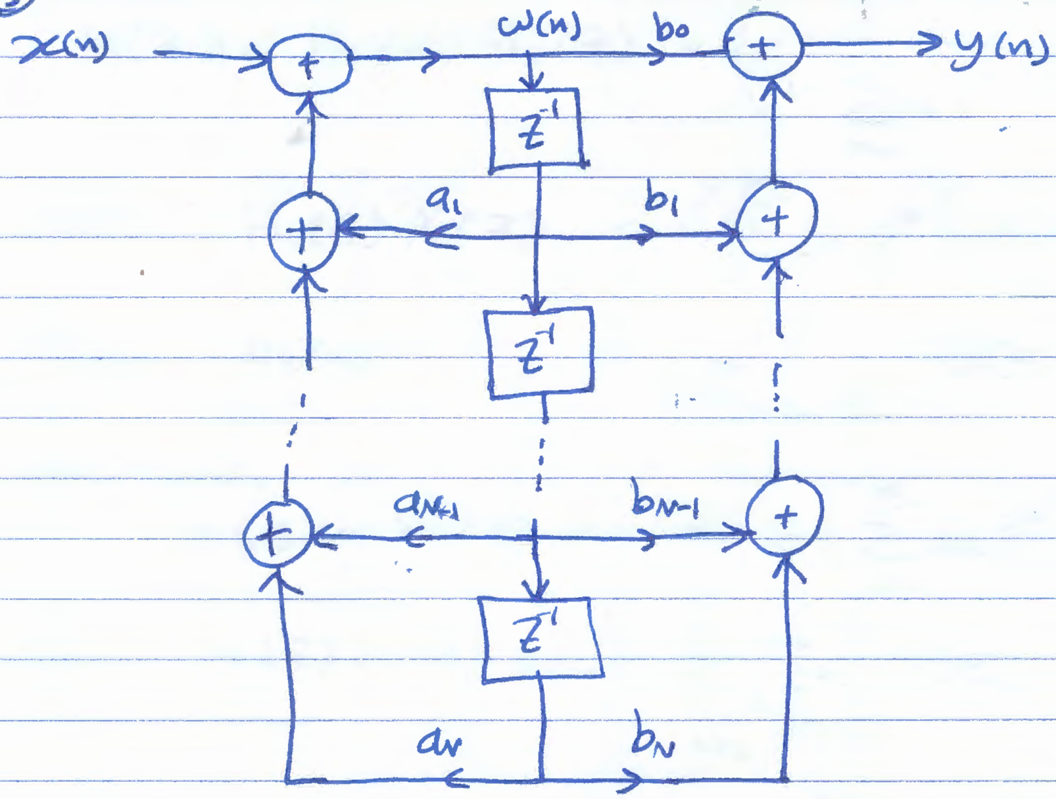
1- In the first block diagram (Fig 1), zeros of $H(Z)$ [redacted] ($H_1(Z)$) are implemented first followed by the poles ($H_2(Z)$).

2. In Fig. 2 poles are implemented first, then zeros.

3. No. of delays units in the two structures is the same ($N+M$).

<u>BUT</u> Fig 2 two chains of $\boxed{Z^{-1}}$ can be collapsed into one chain.

Fig 3



* No. of delay elements is less than in Fig 1 and 2

⇒ Minimum Number of delay Units is $\max(N, M)$.

⇒ This Implementation (with min. no. of delays) is Called

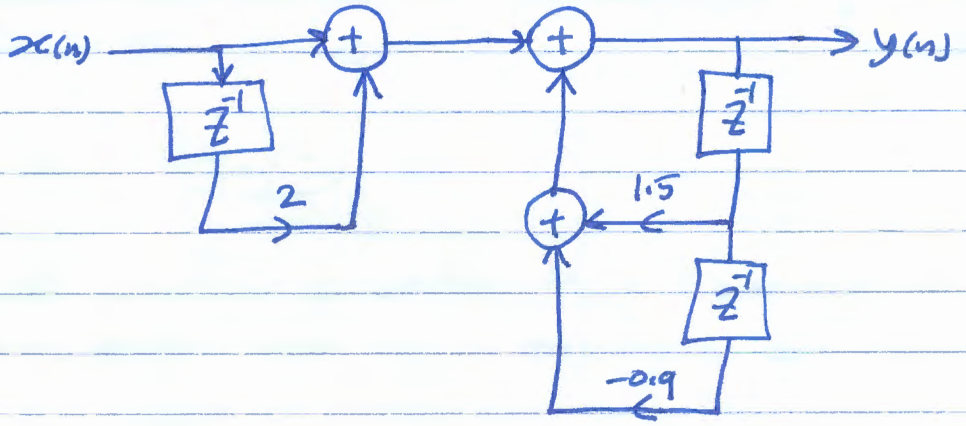
⇒ Canonic form. <u>or</u> Direct Form II

* Block diagram in Fig 1 is called Direct form I.

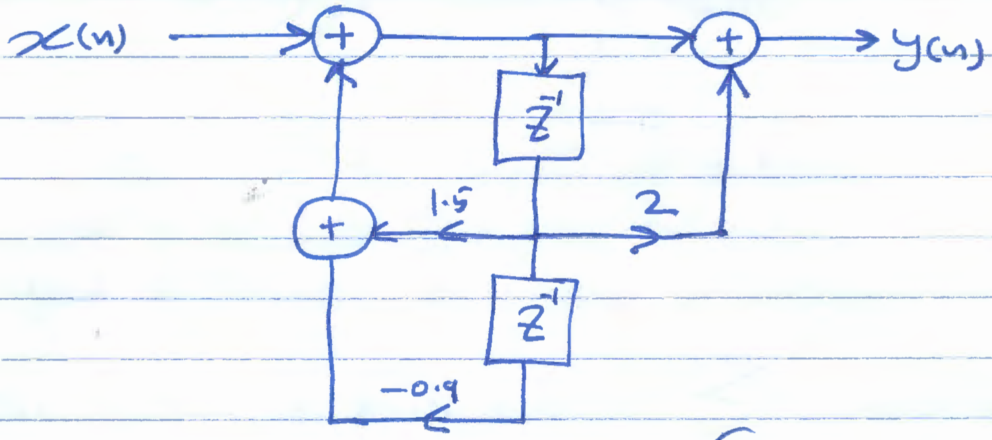
Example:

$$H(z) = \frac{1 + 2z^{-1}}{1 - 1.5z^{-1} + 0.9z^{-2}}$$ Draw Direct form I ? 〃 〃 〃 II ?

$b_0 = 1$
$b_1 = 2$
$a_1 = 1.5$
$a_2 = -0.9$

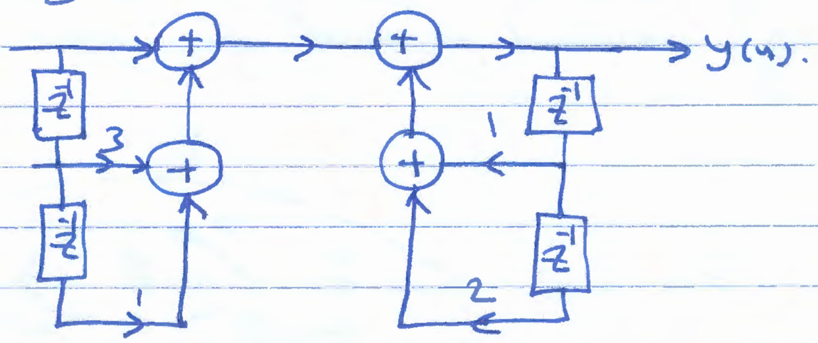


Direct form I



Direct form II (Canonic Direct form).

Example 2: LTI system

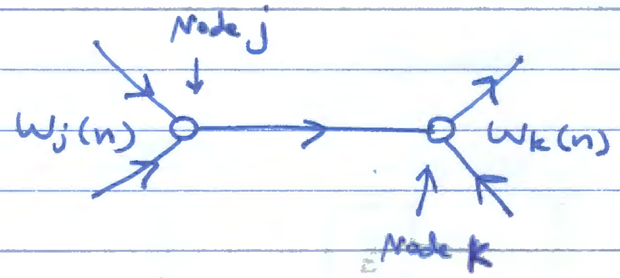


(a) Write difference equation of this LTI system?
(b) Find system Function H(z)?
(c) How many real multiplications and real additions are required to compute each sample of the output?
(Assume x(n) is real and multiplication by one doesn't count to the total).

(d) This realization requires four Storage Register (delay). Is it possible to reduce no. of Storage Registers? If so, draw the Structure.

Signal Flow Graph

is a network of directed branches that connect at nodes. Associated with each node is a variable or node value.

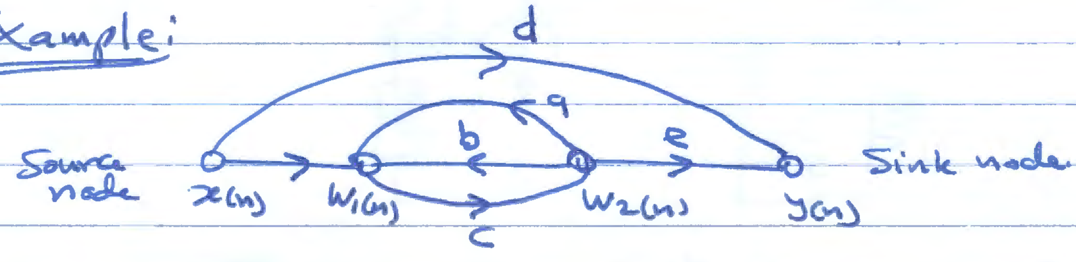


* each Branch has input and output.
* input to branch (j, k) is $W_j(n)$
* output of Branch = input * constant or (1).

* The value at each node = $\sum$ output of all Branches entering the node.

Source Node: No entering Branches (used for source signal)
Sink nodes: have only entering branches (extract output from graph)

Example:



$W_1(n) = x(n) + a W_2(n) + b W_2(n)$ ——— ①

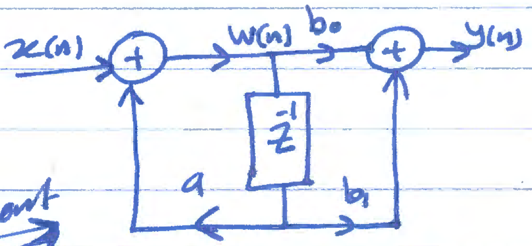
$W_2(n) = c W_1(n)$ ——— ②

$y(n) = d x(n) + e W_2(n)$ ——— ③

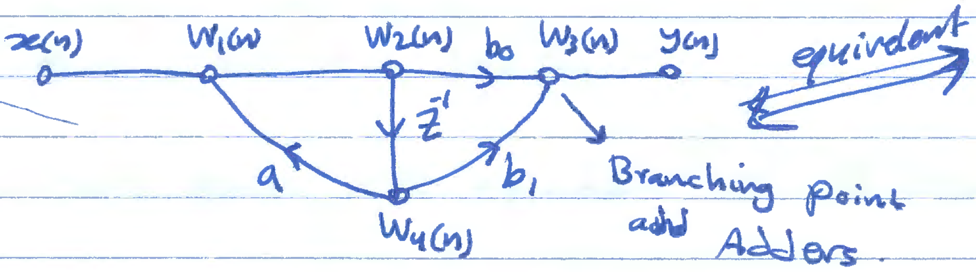
Addition, Multiplication by a constant, delay ⇒ required for linear systems.

Example: First-order Digital Filter.

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a z^{-1}}$$



Block Diagram
Direct form II



Signal flow graph

Multi-Step Algorithm:

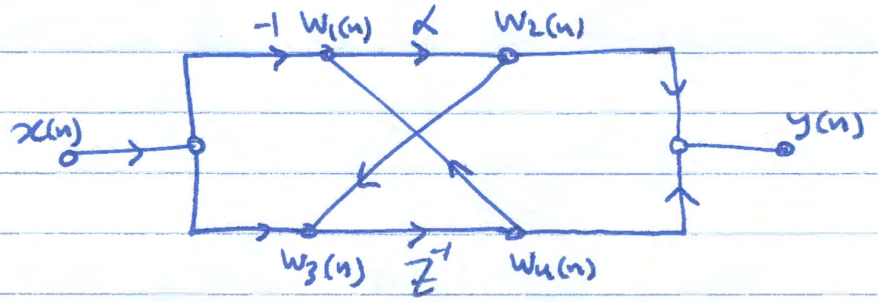
$W_1(n) = a W_4(n) + x(n)$
$W_2(n) = W_1(n)$
$W_3(n) = b_0 W_2(n) + b_1 W_4(n)$
$W_4(n) = W_2(n-1)$
$y(n) = W_3(n)$

Cannot be computed in an arbitrary order.

Intial Reset Condition
$W_2(-1) = 0$ or
$W_4(0) = 0$.

$W_2(n) = a W_4(n-1) + x(n)$
$y(n) = b_0 W_2(n) + b_1 W_2(n-1)$.

Example:



Find System Function $H(z)$?

*This graph is not in direct form ⟹ $H(z)$ cannot be written by inspection, but a set of difference equations can be written:

$w_1(n) = w_4(n) - x(n)$

$w_2(n) = \alpha w_1(n)$

$w_3(n) = w_2(n) + x(n)$

$w_4(n) = w_3(n-1)$

$y(n) = w_2(n) + w_4(n).$

* convert these diff. equations into Z-transform :

$$W_1(Z) = W_4(Z) - X(Z) \text{ ——— (a)}$$
$$W_2(Z) = \alpha W_1(Z) \text{ - - - - (b)}$$
$$W_3(Z) = W_2(Z) + X(Z) \text{ ---- (c)}$$
$$W_4(Z) = Z^{-1} W_3(Z) \text{ - - - - (d)}$$
$$Y(Z) = W_2(Z) + W_4(Z) \text{ --- (e)}$$

* we can eleminate $W_1(Z)$ and $W_3(Z)$ by substituting (a) into (b) and (c) into (d).

$$W_2(Z) = \alpha \left(W_4(Z) - X(Z) \right)$$
$$W_4(Z) = Z^{-1} \left(W_2(Z) + X(Z) \right)$$
$$Y(Z) = W_2(Z) + W_4(Z)$$

→ can be solved for $W_2(Z)$ and $W_4(Z)$ ⟹

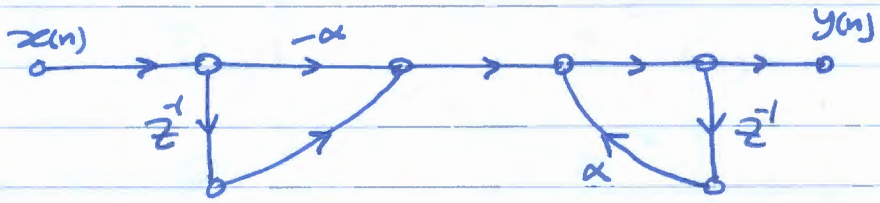
$$W_2(Z) = \frac{\alpha(Z^{-1}-1)}{1-\alpha Z^{-1}} X(Z)$$

$$W_4(Z) = \frac{Z^{-1}(1-\alpha)}{1-\alpha Z^{-1}} X(Z)$$

$$Y(Z) = \left(\frac{\alpha(Z^{-1}-1) + Z^{-1}(1-\alpha)}{1-\alpha Z^{-1}} \right) X(Z) = \left(\frac{Z^{-1}-\alpha}{1-\alpha Z^{-1}} \right) X(Z)$$

$$\Rightarrow H(Z) = \frac{Z^{-1}-\alpha}{1-\alpha Z^{-1}} \Rightarrow \underline{h(n) = \alpha^{n-1} u(n-1) - \alpha^{n+1} u(n)}$$

Direct form I flow graph is:



* This example shows how Z-transform converts the time-domain expressions, which involves feedback and thus are difficult to solve, into linear equations that can be solved by algebraic techniques.

* By comparing the original graph with direct form I graph:

(mem.)

original $\longrightarrow$ Requires one Multiplication and one delay

Direct form I $\longrightarrow$ Two Mult. and Two delays

" " II $\longrightarrow$ " " " one delay.

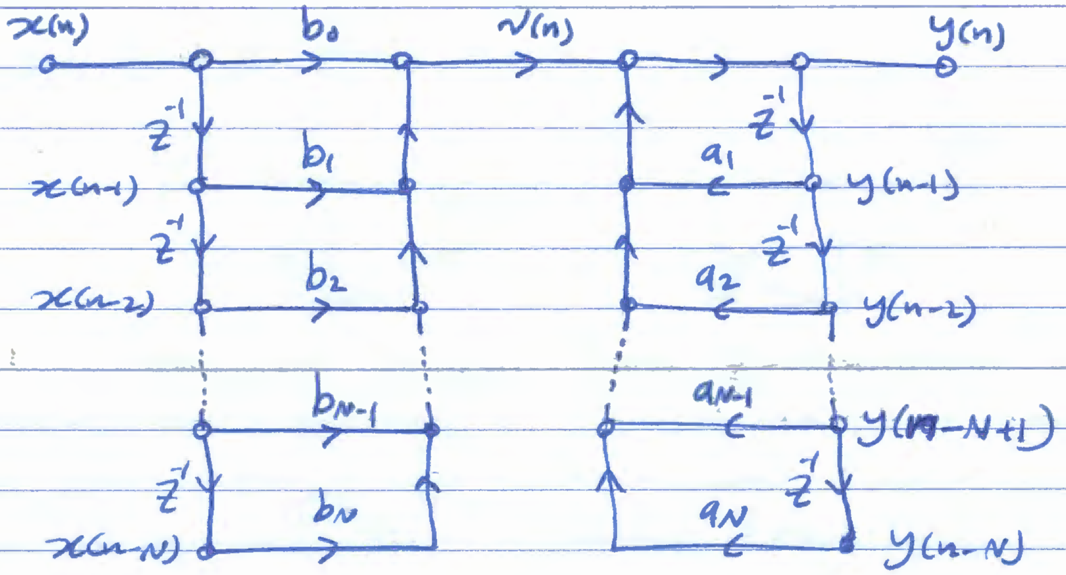
6.3 Basic Structures for IIR Systems:

* Direct Forms:

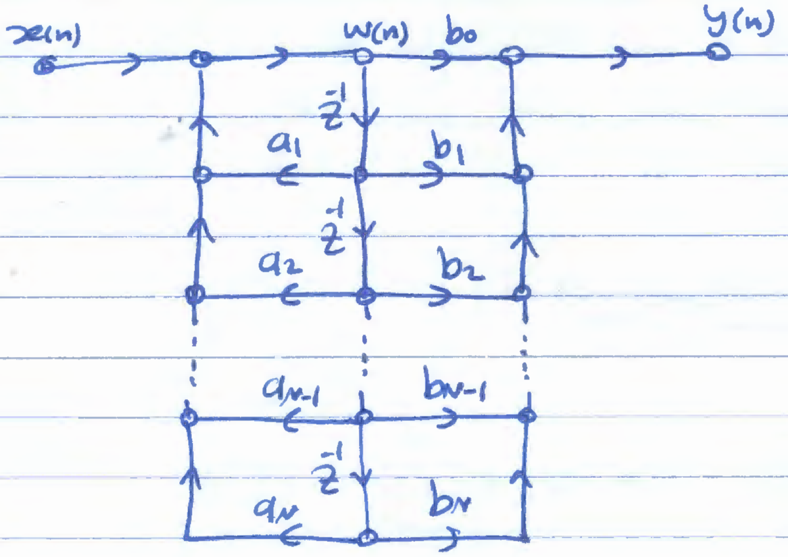
$$y(n) - \sum_{k=1}^{N} a_k\, y(n-k) = \sum_{k=0}^{M} b_k\, x(n-k)$$

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

Assuming $N = M$:



Signal flow graph of Direct Form I structure for an N^{th}-order sys.



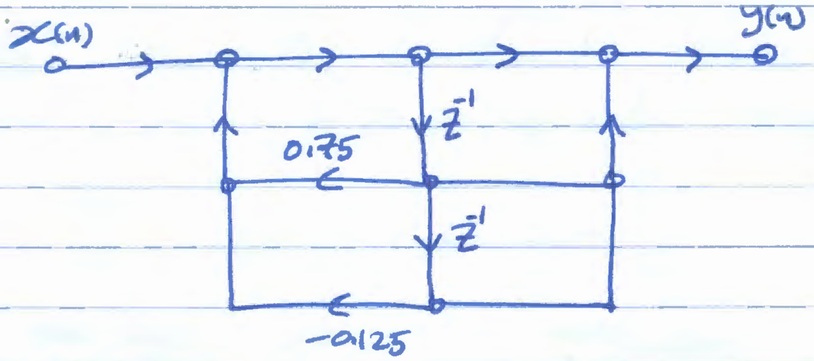
Direct Form II for an N^{th} order IIR system.

Example 1 $\quad H(z) = \dfrac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}}$

$b_0 = 1,\ b_1 = 2,\ b_3 = 1$

$a_1 = 0.75,\ a_2 = -0.125$

Direct Form I



Direct Form II

* Cascade form:

$$H(z) = A \frac{\prod_{k=1}^{M_1}(1-f_k z^{-1})}{\prod_{k=1}^{N_1}(1-c_k z^{-1})} \cdot \frac{\prod_{k=1}^{M_2}(1-g_k z^{-1})(1-g_k^* z^{-1})}{\prod_{k=1}^{N_2}(1-d_k z^{-1})(1-d_k^* z^{-1})}$$

* This is a Cascade of first-order and second-order systems.

$M = M_1 + 2M_2$

$N = N_1 + 2N_2$

First-order factors $\longrightarrow$ real zeros at f_k and real poles at c_k

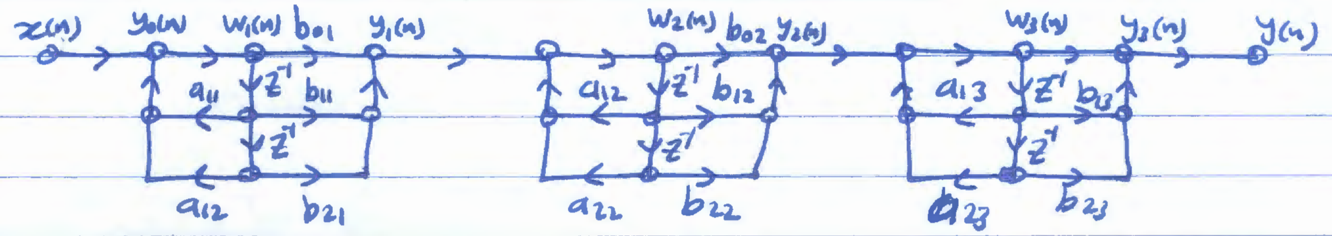
second-order " $\longrightarrow$ complex conjugate pairs of zeros at g_k and g_k^* and complex conjugate pairs of poles at d_k and d_k^*.

* By combining pairs of real factors and complex conjugate pairs into a second-order factors

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} \qquad \text{Assume } M \le N$$

$N_s = \lfloor (N+1)/2 \rfloor$ is largest Integer in $(N+1)/2$.

Example: Cascade structure for sixth-order system using three direct form II second-order sections.



$$y_0(n) = x(n)$$

$$W_k(n) = a_{1k} W_k(n-1) + a_{2k} W_k(n-2) + y_{k-1}(n), \quad k = 1, 2, \ldots, N_s$$

$$Y_k(n) = b_{0k} W_k(n) + b_{1k} W_k(n-1) + b_{2k} W_k(n-2), \quad k = 1, 2, \ldots, N_s$$

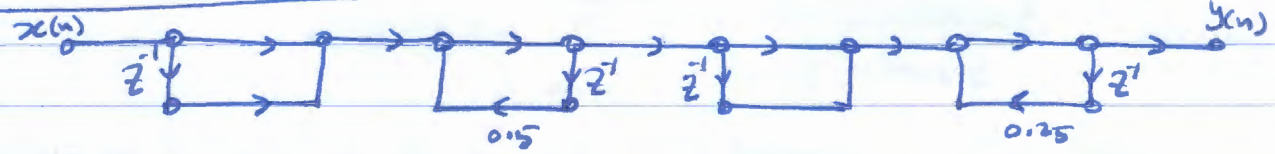
$$y(n) = y_{N_s}(n)$$

Example:

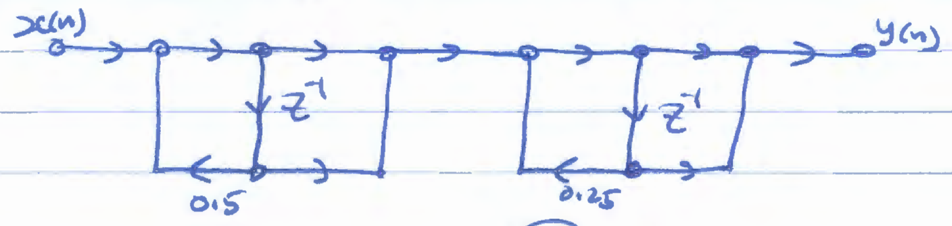
$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}} = \frac{(1+z^{-1})(1+z^{-1})}{(1-0.5z^{-1})(1-0.25z^{-1})}$$

Since all poles and zeros are real, a cascade structure with first-order secotions has real coefficients. But if poles and/or zeros were complex, only second-order sections would have real coefficients.

Cascade DF I:



D.F. II.



Parallel Form:

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_1} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \frac{B_k(1 - e_k z^{-1})}{(1 - d_k z^{-1})(1 - d_k^* z^{-1})}$$

$N = N_1 + 2N_2$. If $M \geq N \Rightarrow N_p = M - N$, otherwise first sum is not included. If a_k and b_k are real $\Rightarrow A_k, B_k, C_k, c_k$ and e_k are all real.

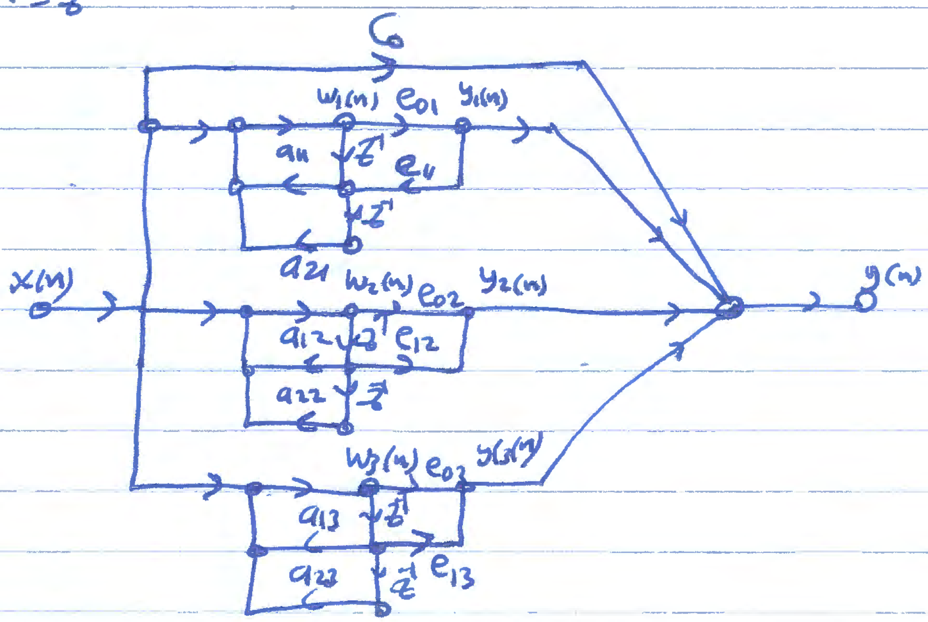
* Parallel combination of first and second-order IIR systems with possibly N_p scaled delay paths.

* Alternativily, we may group real pairs poles in pairs. $\Rightarrow$

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

$N_s = \lfloor (N+1)/2 \rfloor$ is largest integer in $(N+1)/2$.
If $N_p = M - N$ is negative, first sum disapear.

Example: $N = M = 6$



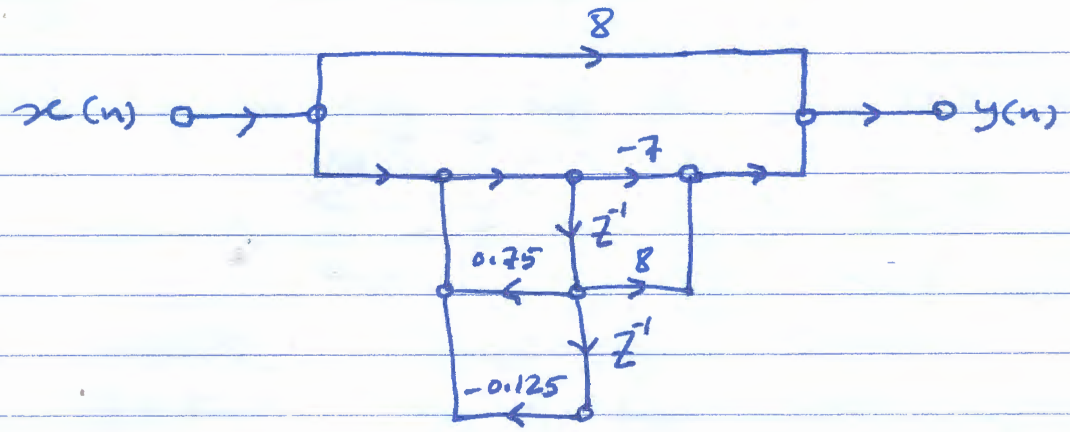
Example:

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}}$$

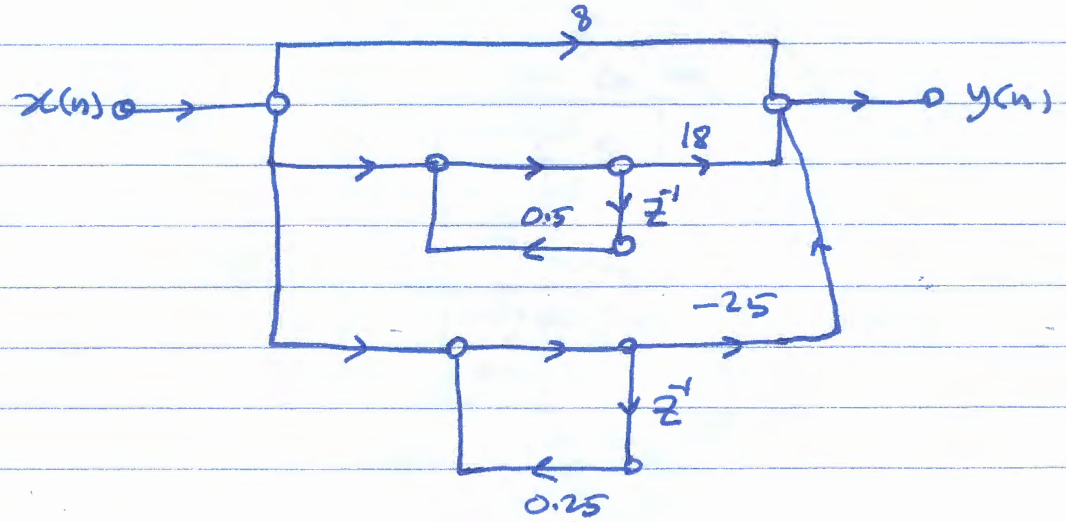
$$H(z) = 8 + \frac{-7 + 8z^{-1}}{1 - 0.75z^{-1} + 0.125z^{-2}}$$

* since all poles are real ⇒ we can obtain an alternative parallel realization by ~~executing~~ expanding $H(z)$ as

$$H(z) = 8 + \frac{18}{1 - 0.5z^{-1}} - \frac{25}{1 - 0.25z^{-1}}$$



Parallel Form Structure using second-order system.



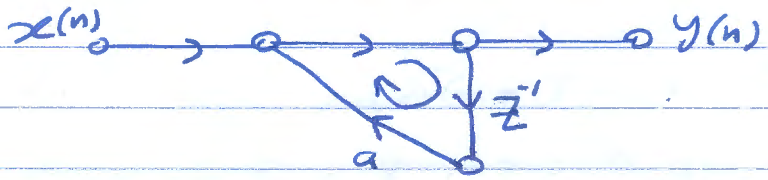
Parallel Form Structure using First-order systems.

Feedback in IIR systems:

- Feedback loops : closed paths begin at node and return to that node only in the direction of arrows (implies node variable depends directly or indirectly on itself).

E.G.

$$y(n) = a\,y(n-1) + x(n)$$



* If $x(n) = \delta(n)$, then the single sample continually circulates in the feedback loop with either increasing ($|a|>1$) or decreasing ($|a|<1$) amplitude $\Rightarrow h(n) = a^n u(n)$.

$\Rightarrow$ This is the way that feedback loop can create an infinitely long impulse response.

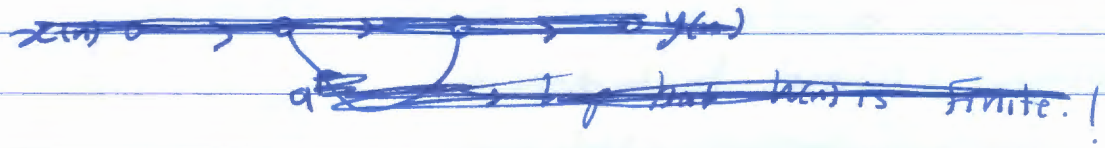
* Let's assume a network with no feedback loops. Therefore, longest delay between input and output occur for path that passes through all delay elements in the network.

$\Rightarrow$ For networks with no loops $\Rightarrow$ $H(z)$ has only zero's (except for poles at $z=0$) and # of zeros no more than # of delay elements.

* If $H(z)$ has poles $\Rightarrow$ Network will have feedback loops.

BUT Neither poles in $H(z)$ nor feedback loops in Network are sufficient for $h(n)$ to be infinitely long.

~~eg~~

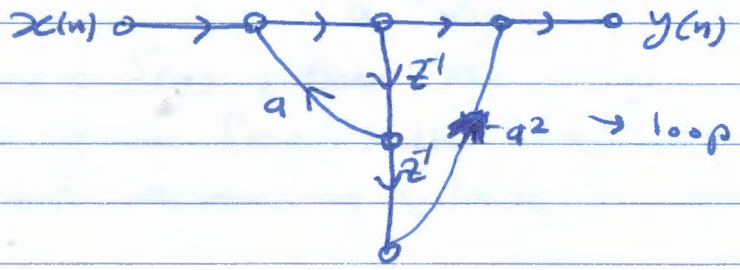


* This is because pole cancels with a zero.

e.g 2

$$H(z) = \frac{1 - a^2 z^{-2}}{1 - a z^{-1}} = \frac{(1 \cancel{- a z^{-1}})(1 + a z^{-1})}{(1 \cancel{- a z^{-1}})} = 1 + a z^{-1}$$

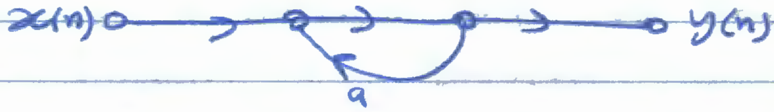
$h(n) = \delta(n) + a\,\delta(n-1) \longrightarrow$ FIR system.



→ loop but $h(n)$ is finite.

* Noncomputable Network

e.g. $y(n) = a\,y(n) + x(n)$



* This means that cannot represent a set of difference equations that can be solved successively for the node variable.

But it can be solved by

$$y(n) = \frac{x(n)}{1-a}$$

* The key to computability of graph is that all loops must contain at least one delay element
  * No delay-free loops

6.4 Transposed Forms

* Flow graph reversal or transposition

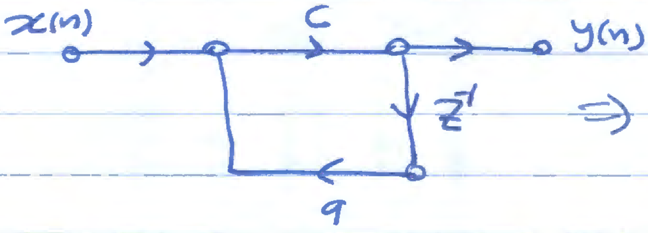
⇒ reverse directions of all branches while keeping branch transmittances as they were and reversing the roles of ~~sou~~ input and output, so that source node become sink node and vice versa.

* For single-input, single-output systems, resulting flow graph has system function $H(z)$ same as original graph if input and output node are interchanged.

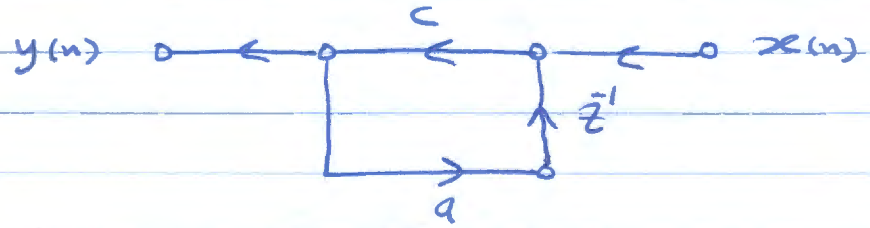
1. Reverse direction of all Branches
2. Interchange Input and output.

Transfer function remains the same.
(proof is not required!)

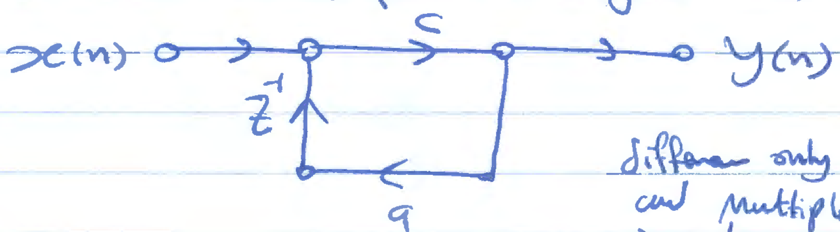
Example:



$$\Rightarrow \quad H(z) = \frac{c}{1 - a z^{-1}}$$

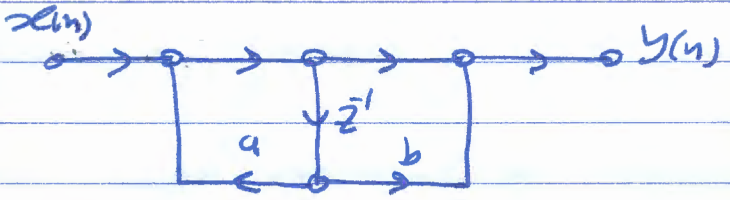


* Flip it over to have input at right side and output at left side.



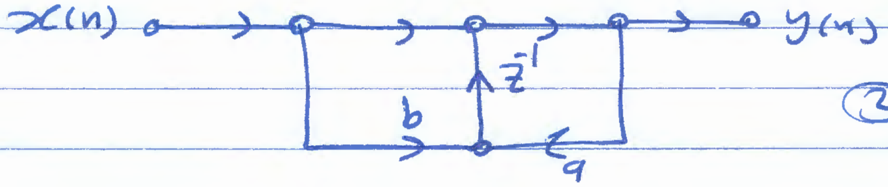
difference only on order of z^{-1} and multiplication by a.
⇒ order not important.

Ex. 2



①

Transposed form:



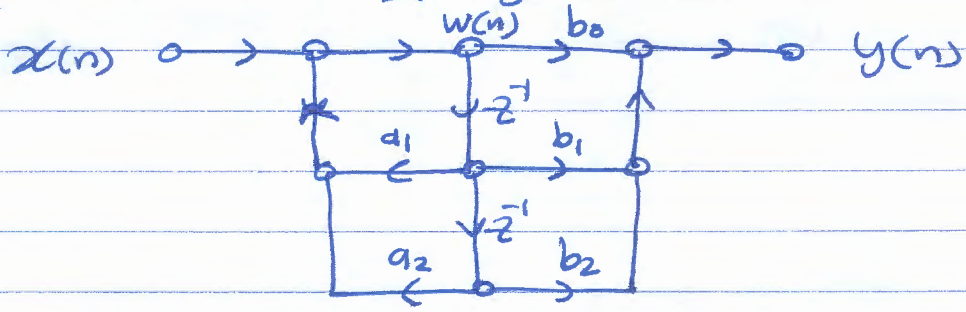
②

① and ② have the same transfer function H(z).

① Implements pole first, then zero. While in ② zero is implemented first, then pole.

(Verify this by calculating H(z) for ① and for ②).

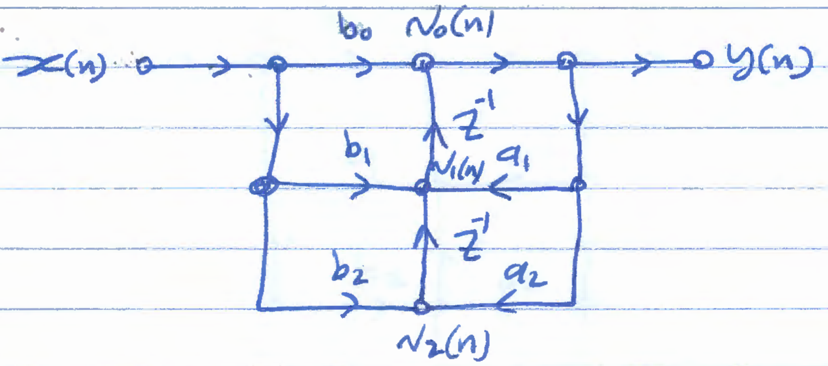
Ex. 3 Second-order IIR system.



The corresponding difference equations:

$$w(n) = a_1 w(n-1) + a_2 w(n-2) + x(n)$$
$$y(n) = b_0 w(n) + b_1 w(n-1) + b_2 w(n-2)$$

* The Transposed flow graph:



It's corresponding difference equations:

$$w_0(n) = b_0 x(n) + w_1(n-1)$$
$$w_1(n) = a_1 y(n) + b_1 x(n) + w_2(n-1)$$
$$w_2(n) = a_2 y(n) + b_2 x(n)$$
$$y(n) = w_0(n).$$

* These difference eq.'s are equivalent to difference equations of the original flow graph.

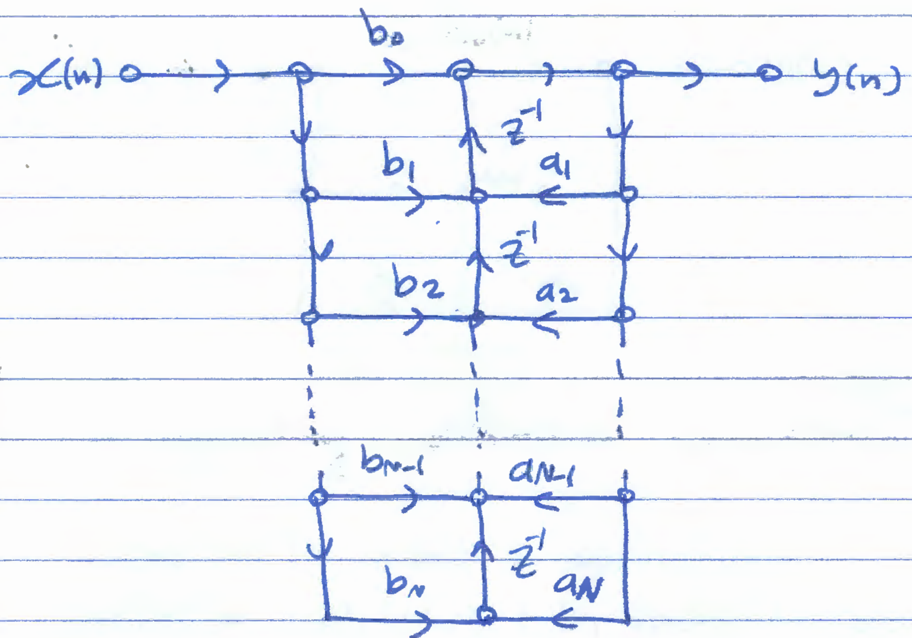
* To show that use z-transform to find $H(z) = \frac{Y(z)}{X(z)}$.

* Another way is find $y(n)$ from set of equations of the transposed graph.

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$$

$\Rightarrow$ Direct form II graph (original graph) satisfies this difference equation.

* Back to general Canonic structure of IIR system. we can get a second Canonic structure by applying transposition theorem.



(Transposed Nth order IIR Cannonic structure).

sec 6.5

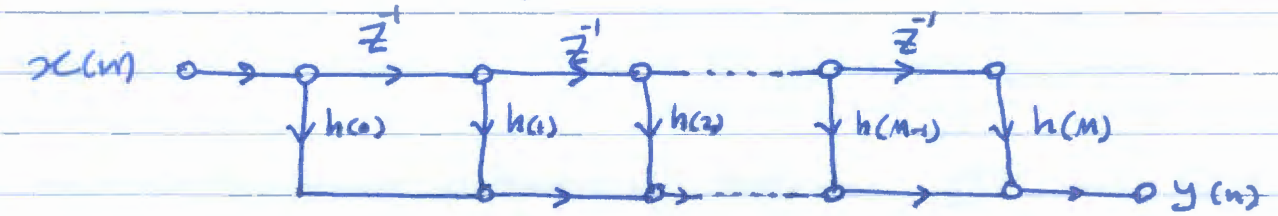
Basic Structures for FIR systems

* Direct Forms

Causal FIR systems have only zeros (except for poles at $z=0$), so diff. equation

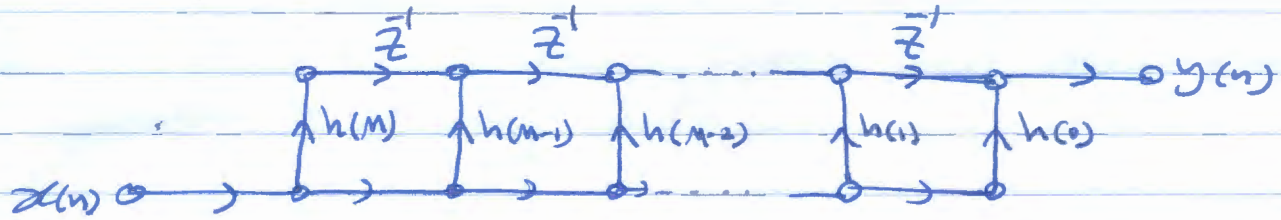
$$y(n) = \sum_{k=0}^{M} b_k \, x(n-k)$$

$$\Rightarrow \quad h(n) = \begin{cases} b_n , & n = 0, 1, \ldots, M \\ 0 , & \text{otherwise} \end{cases}$$



Direct form I realization of FIR system.

* This is called tapped-delay line structure or Transversal filter structure.

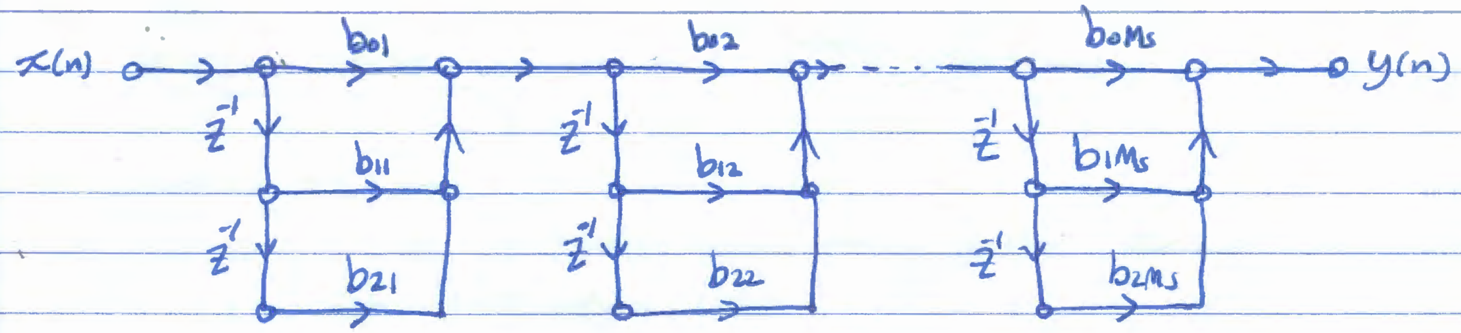


Transposed Direct form I.

* Cascade Forms

$$H(z) = \sum_{n=0}^{M} h(n) \, z^{-n} = \prod_{k=1}^{M_s} \left(b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2} \right)$$

$M_s = \lfloor (M+1)/2 \rfloor$, if M is odd, one of $b_{2k} = 0$



Cascade form realization of FIR sys.

* we can apply transposition theorem on this structure to get new implementation structure.

* Implementation of linear phase FIR systems

* one of the main differences between IIR and FIR systems that because $h(n)$ of FIR is finite, these systems can be designed to have a linear phase where the corresponding statement for IIR systems is not true.

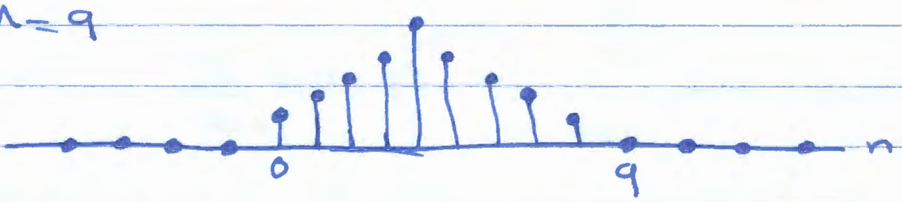
* To have linear phase FIR, $h(n)$ must satisfy ~~symmetry~~ symmetry condition:

$$h(n) = h(M-1-n) \quad , n=0,1,\dots M$$

or

$$h(n) = -h(M-1-n) \quad , n=0,1,\dots, M.$$

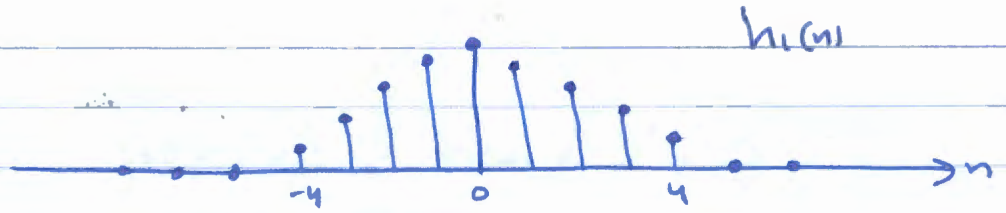
__e.g__ let $M=9$



$h(0)=h(8)$

$h(1)=h(7)$

$\vdots$



$$h(n) = h_1\left(n + \frac{M-1}{2}\right)$$

$$H(e^{jw}) = e^{jw\left(\frac{M-1}{2}\right)} H_1(e^{jw})$$

relation between H_1 and H is only shift.

$h_1(n)$ is even $\Rightarrow H_1(e^{jw})$ is real function of ω.

* If impulse response system, $h(n)$, has this symmetry property (i.e. $h(n) = \mp h(M-1-n)$), then the phase of the system is linear.

$$H(z) = \sum_{n=0}^{M-1} h(n) z^{-n}$$

Same coeff. but different index.

Assume. M even.

$$\Rightarrow H(z) = \sum_{n=0}^{\frac{M}{2}-1} h(n) z^{-n} + \underbrace{\sum_{n=\frac{M}{2}}^{M-1} h(n) z^{-n}}$$

let $r = (M-1)-n$ ~~$\Rightarrow$~~ M

$n = M-1-r$

Same but running in opposite direction

$$\Downarrow$$

$$\sum_{r=0}^{\frac{M}{2}-1} \underbrace{h(M-1-r)}_{h(r)} z^{-(M-1-r)}$$

at $n = \frac{M}{2} \Rightarrow r = M-1-\frac{M}{2} = \frac{M}{2}-1$

at $n = M-1 \Rightarrow r = M-1-(M-1) = 0$

$$= \sum_{n=0}^{\frac{M}{2}-1} h(n) z^{-n} + \sum_{n=0}^{\frac{M}{2}-1} h(n) z^{-(M-1-n)}$$

$$= \sum_{n=0}^{\frac{M}{2}-1} h(n) \left[z^{-n} + z^{-(M-1-n)} \right]$$

# of coeff. Multipliers is ~~[illegible]~~ halved

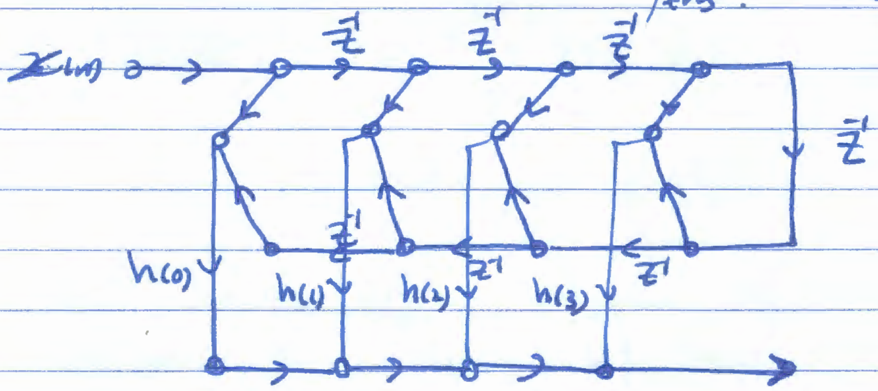
* requires $\frac{M}{2}$ different Coefficients instead of M.

Assume $M=8$.

$$H(z) = \sum_{n=0}^{3} h(n)\left[z^{-n} + z^{-(7-n)}\right]$$

* How many delay units required to implement this system?

Ans. 7 delays.



* Involve 7 delay units same as direct form I of FIR, but we use only half of multiplications.

* You can apply transposition theorem on this graph and get new implementation.