-Digital Signal Processing-
Representation of Linear Digital Networks

Lecture-11
03-May-16

What are Poles & Zeros?

- Let's say we have a transfer function defined as a ratio of two polynomials:

$$H(s) = \frac{N(s)}{D(s)}$$

- Where N(s) and D(s) are simple polynomials.
- Zeros are the roots of N(s) obtained by setting N(s)=0 and solving for s.
- Poles are the rots of D(s) obtained by setting D(s)=0 and solving for s.
- Note: a transfer function must not have more zeros than poles i.e., the polynomial order of D(s) must be greater than or equal to the polynomial order of N(s).
- The polynomial order of a function is the value of the highest exponent in the polynomial.

Example #0

- Consider the transfer function:

$$H(s) = \frac{s+2}{s^2+0.25}$$

- The numerator N(s) and denominator D(s) are:

$$N(s) = s+2$$

$$D(s) = s^2+0.25$$

- Set N(s) to zero and solve for s:

$$N(s) = s+2 = 0 \Rightarrow s = -2$$

- We have a zero at s=-2. now set D(s) to zero and solve for s to obtain the poles of the equation:

$$D(s) = s^2+0.25 = 0 \Rightarrow s = +i\sqrt{0.25}, -i\sqrt{0.25}$$

- Simplifying gives us the poles at : -i/2 and +i/2.
- Remember, s is a complex variable and it can therefore take imaginary and real values.

Digital Networks

- We discussed LTI systems that are representable by linear constant difference equations.
- As we defined previously the Nth order difference equation that corresponds to the linear combination of delayed output sequences equals to the linear combination of delayed input sequences.

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

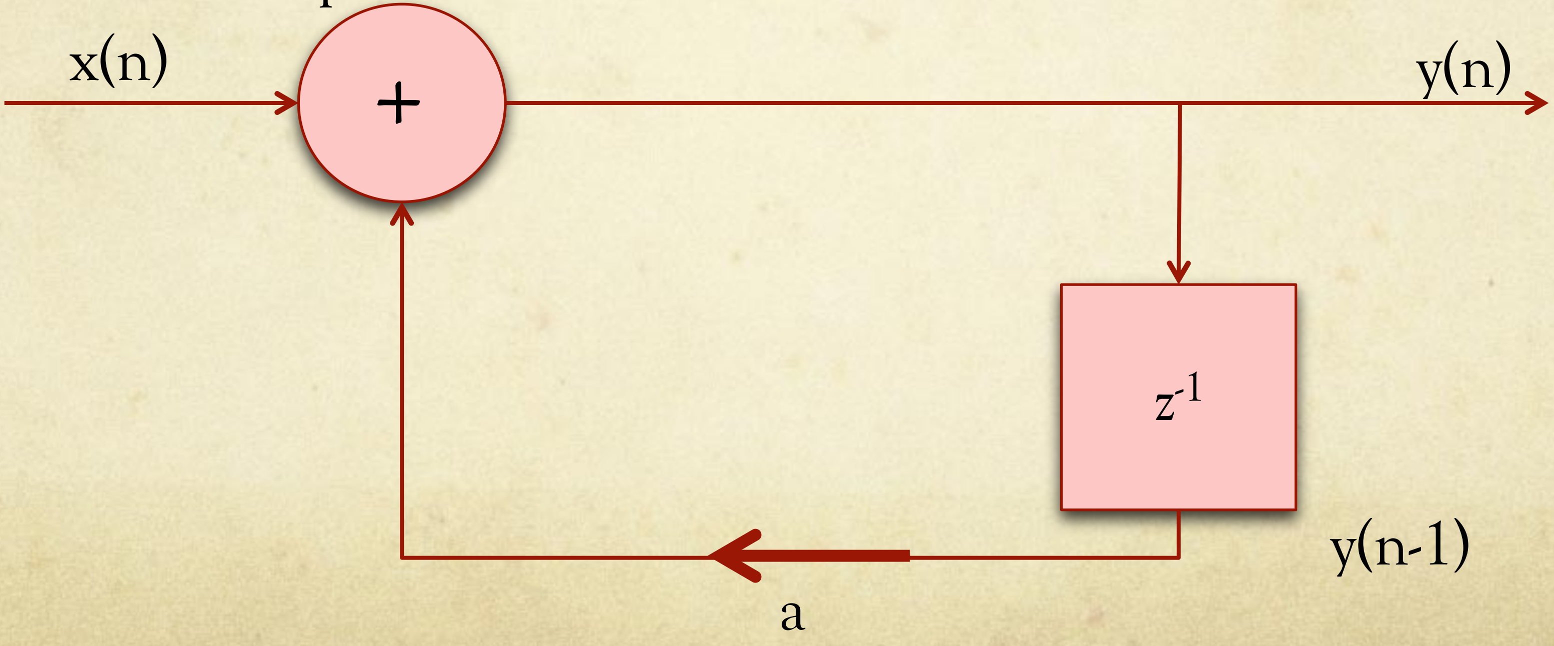
$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Digital Networks

- There are three basic operations are involved in above equations.
  - Delay
  - Multiplication
  - Addition
- We need graphical representation of above mentioned operations which will combine together into a digital network.

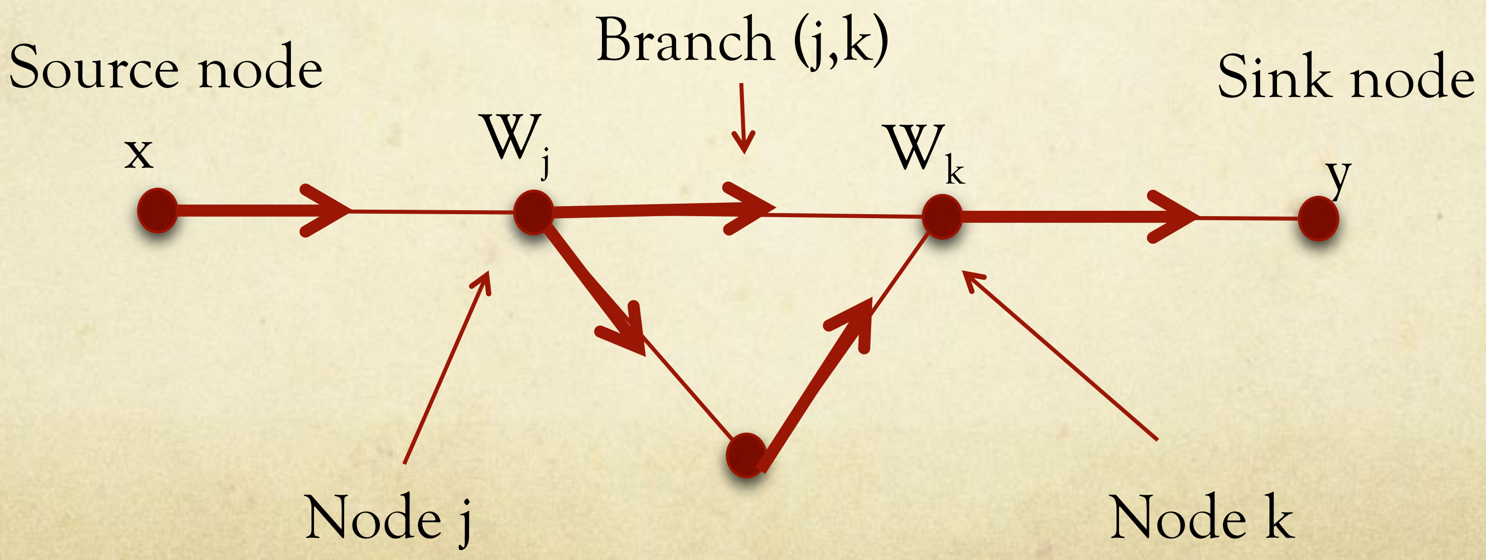
Example #1

- If we have first order differential equation: y(n)=ay(n-1)+x(n)
- Solution:
- The digital network in block diagram notation for the above difference equation is as follows:



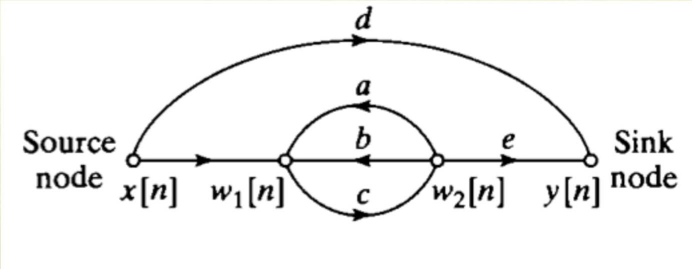
Signal Flow Graphs

- A network of directed branches which are connected in nodes.
- In a signal flow graph we have nodes & directed branches (with arrows on them).
- Nodes will be numbered and a variable will be associated with it.



Signal Flow Graphs

- Nodes are known as network nodes.
- There are also source nodes, which is a node that has no branches coming into it only have the branches leaving it.
- And we have opposite node that is sink node, which has no branches going out of it and has only branches coming to it.
- Source nodes represents the input to the network and sink node represents the output.
- Source nodes, sink nodes and simple branch gains in the signal flow graph are shown below:



Signal Flow Graphs

- The linear equations represented by the figure are as follows:

$$w_1[n] = x[n] + aw_2[n] + bw_2[n]$$

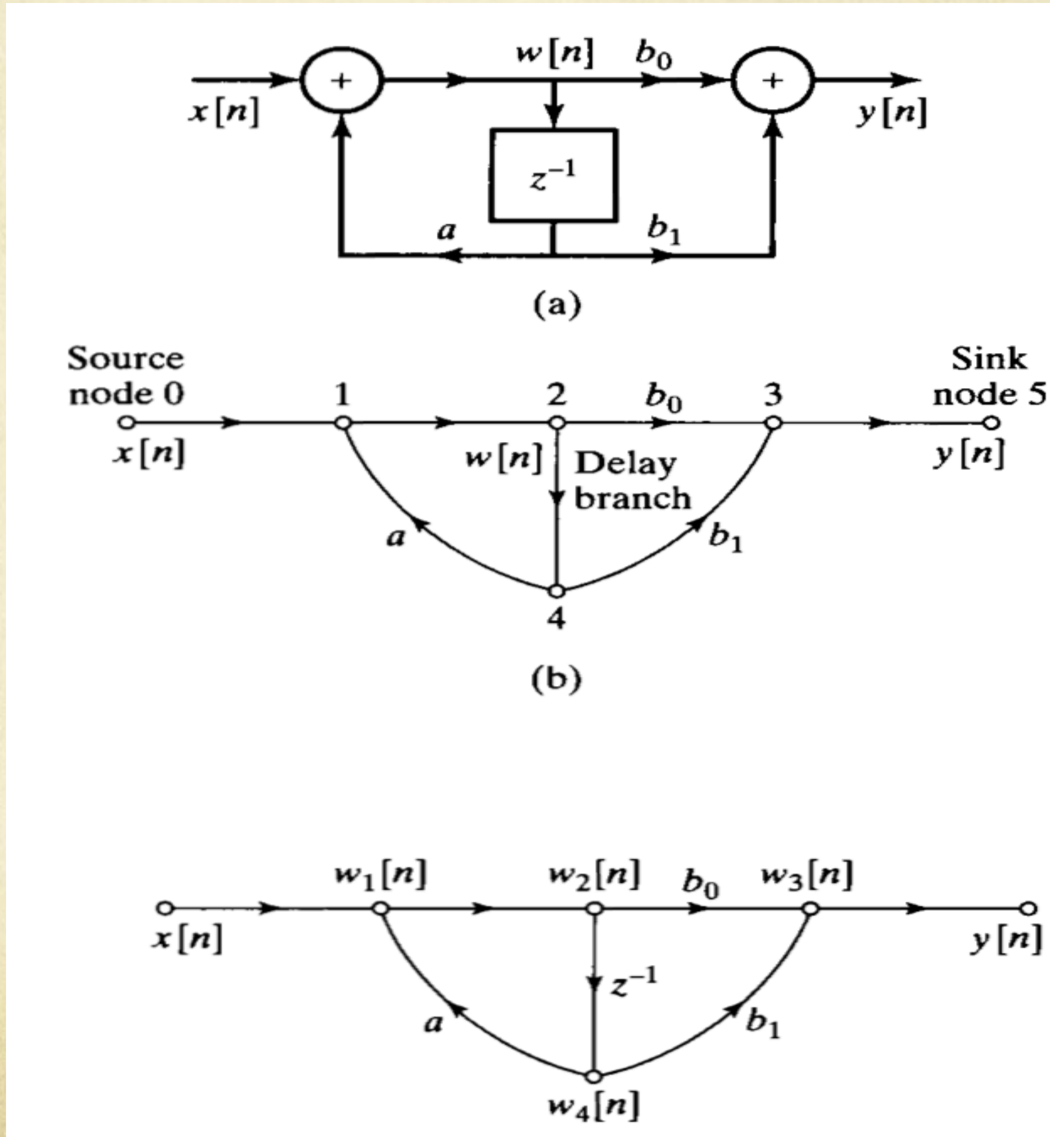
$$w_2[n] = cw_1[n]$$

$$y[n] = dx[n] + ew_2[n]$$

- Since these are all linear operations, it is possible to use signal flow graph notation to depict algorithms for implementing LTI discrete time systems.

Linear Signal Flow Graph

- Example:



Linear Signal Flow Graph (cont.)

- The equations represented by above figures are as follows:

$$w_1[n] = aw_4[n] + x[n] \rightarrow (1)$$

$$w_2[n] = w_1[n] \rightarrow (2)$$

$$w_3[n] = b_0 w_2[n] + b_1 w_4[n] \rightarrow (3)$$

$$w_4[n] = w_2[n-1] \rightarrow (4)$$

$$y[n] = w_3[n] \rightarrow (5)$$

- The comparison between the first figure and the third figure shows that there is a direct correspondence between branches in the block diagram and branches in the flow graph.

- The important difference b/w the two is that nodes in the flow graph represent both branching points and adders whereas in the block diagram a special symbol is used for adders.

Linear Signal Flow Graph (cont.)

- Above equations (1 to 5) define a multistep algorithm for computing the output of the LTI system from the input sequence x[n].
- They cannot be computed in arbitrary order.
- Equations (2) and (5) simply require renaming variables.
- Eq.(4) represents the updating of the memory of the system.
- It would be implemented simply by replacing the contents of the memory register representing $w_4[n]$ by the value of $w_2[n]$, but this would have to be done consistently either before and after the evaluation of all the other equations.

Linear Signal Flow Graph (cont.)

- Initial rest conditions would be imposed in this case by defining $w_2[-1]=0$ or $w_4[0]=0$.
- The flow graph represents a set of difference equations, with one equation being written at each node of the network.
- In the figure given in example we can eliminate some of the variables rather easily to obtain the pair of equations:

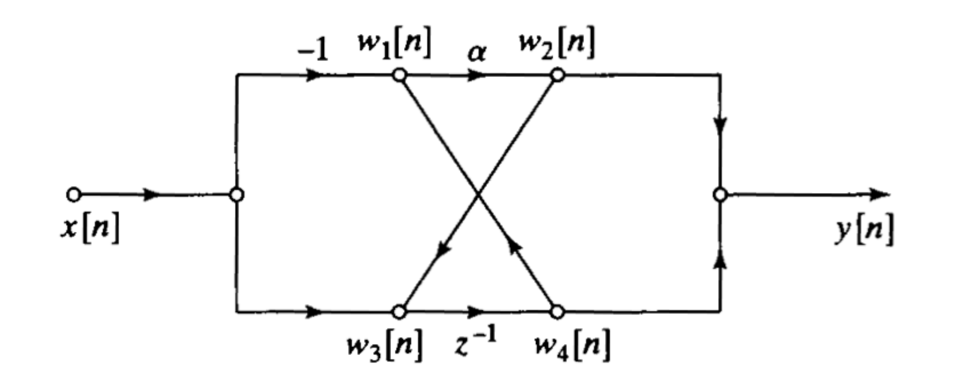
$$w_2[n] = aw_2[n-1] + x[n]$$

$$y[n] = b_0 w_2[n] + b_1 w_2[n-1]$$

- The manipulation of the difference equations of a flow graph is difficult when dealing with the time domain variables, due to feedback of delayed variables.
- In such cases it is always possible to work with the *z*-transform representation wherein all branches are simple gains.

Example #2

- Determination of the system function from a flow graph:
  - To illustrate the use of the *z*-transform in determining the system function from a flow graph consider the following figure:



$$w_1[n] = w_4[n] - x[n] \rightarrow eq.(2.1)$$

$$w_2[n] = aw_1[n] \rightarrow eq.(2.2)$$

$$w_3[n] = x[n] + w_2[n] \rightarrow eq.(2.3)$$

$$w_4[n] = w_3[n-1] \rightarrow eq.(2.4)$$

$$y[n] = w_4[n] + w_2[n] \rightarrow eq.(2.5)$$

Example #2 (cont.)

- These are the equations that would be used to implement the system in the form described by the flow graph.

- According to *z*-transform representation, above equations will become:

$$W_1[Z] = W_4[z] - X[z] \rightarrow eq.(2.1a)$$
$$W_2[z] = aW_1[z] \rightarrow eq.(2.2b)$$
$$W_3[z] = X[z] + W_2[z] \rightarrow eq.(2.3c)$$
$$W_4[z] = z^{-1}W_3[z] \rightarrow eq.(2.4d)$$
$$Y[z] = W_4[z] + W_2[z] \rightarrow eq.(2.5e)$$

- We can eliminate $W_1(z)$ and $W_3(z)$ from this set of equations by substituting eq.(2.1a) into eq.(2.2b) and eq.(2.3c) into eq.(2.4d), obtaining:

Example #2 (cont.)

$$W_2[z] = a(W_4[z] - X[z]) \rightarrow eq.(2.22a)$$

$$W_4[z] = z^{-1}(X[z] + W_2[z]) \rightarrow eq.(2.44b)$$

$$Y[z] = W_4[z] + W_2[z] \rightarrow eq.(2.55c)$$

- Eqs.(2.22a) and (2.44b) can be solved for $W_2(z)$ and $W_4(z)$, yielding:

$$W_2[z] = \frac{a(z^{-1} - 1)}{1 - az^{-1}} X(z), \rightarrow eq.(2.23a)$$

$$W_4[z] = \frac{z^{-1}(1 - a)}{1 - az^{-1}} X(z), \rightarrow eq.(2.23b)$$

- By substituting above equations in Y(z) leads to:

$$Y[z] = \left(\frac{a(z^{-1} - 1) + z^{-1}(1 - a)}{1 - az^{-1}}\right) X(z) = \left(\frac{z^{-1} - a}{1 - az^{-1}}\right) X(z) \rightarrow eq.(2.6)$$

Example #2 (cont.)

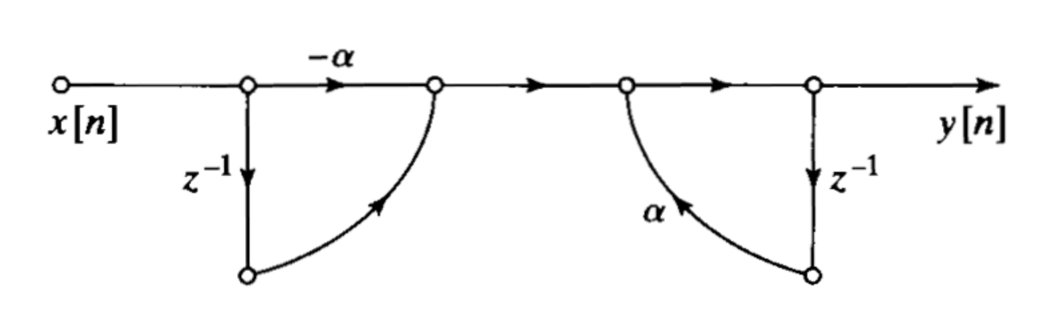
- Therefore the system function of the flow graph is:

$$H[z] = \frac{z^{-1} - a}{1 - az^{-1}}$$

- For which it follows that the impulse response of the system is:

$$h[n] = a^{n-1}u[n-1] - a^{n+1}u[n]$$

- The direct form I flow graph is as follows:



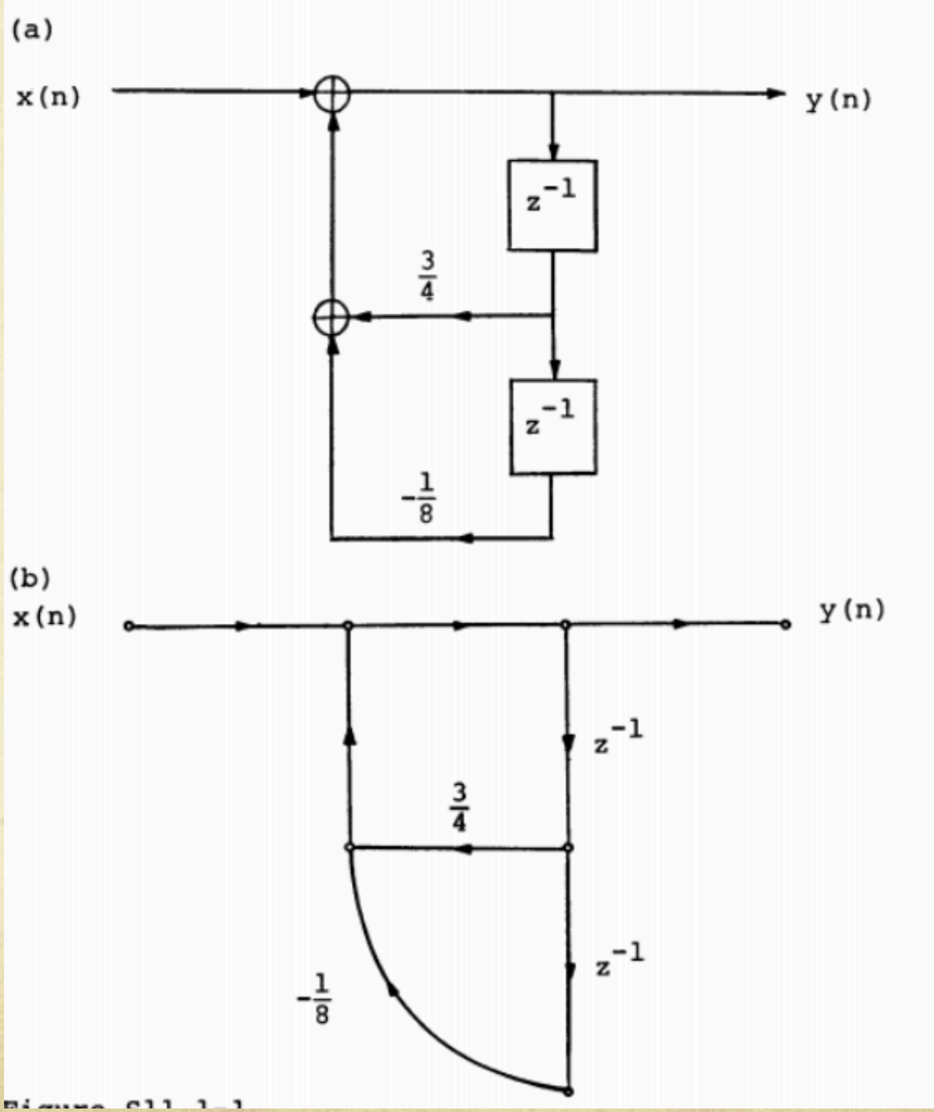
Example #3

- Consider the discrete time system represented by the linear constant coefficient difference equation:

$$y(n) - \frac{3}{4} y(n-1) + \frac{1}{8} y(n-2) = x(n)$$

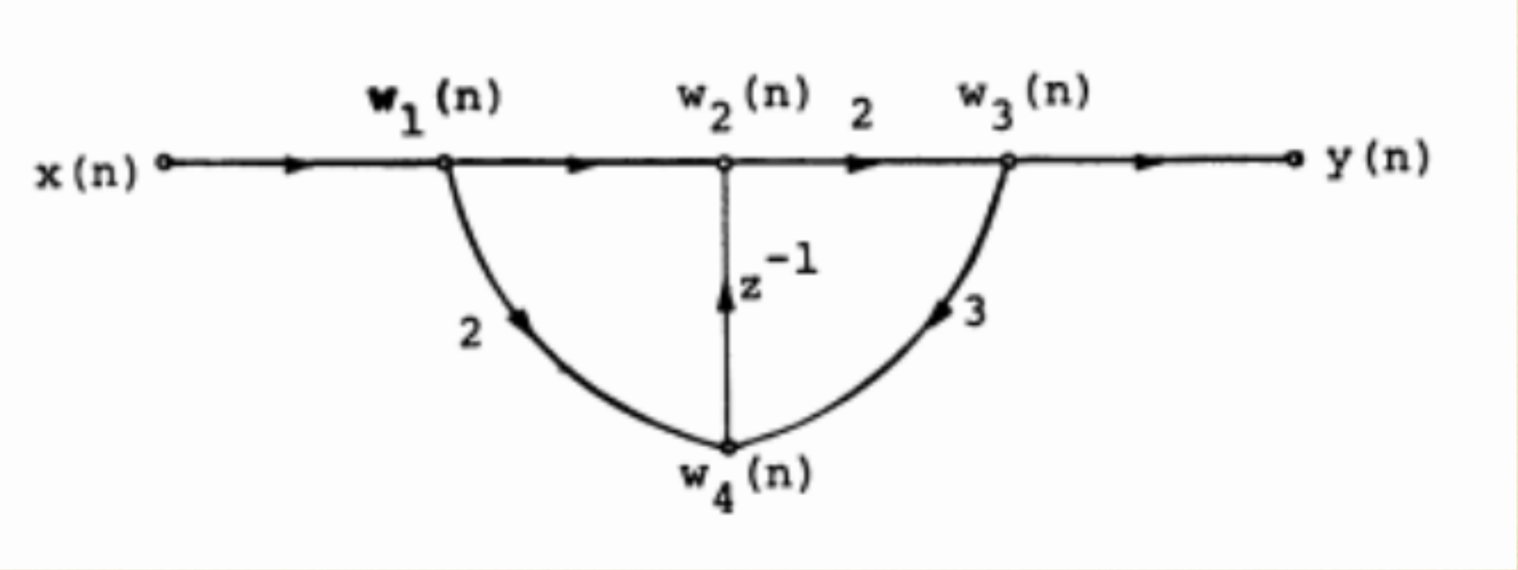
  - Draw a block diagram representation of the system in terms of adders and delay and coefficient multiplication branches.
  - Draw a linear signal flow graph representation of the system.

Example #3 Solution



Example #4

- Determine the corresponding equations for the digital network shown below:



Example #4 Solution

$$W_1(z) = X(z)$$

$$W_2(z) = W_1(z) + z^{-1}W_4(z)$$

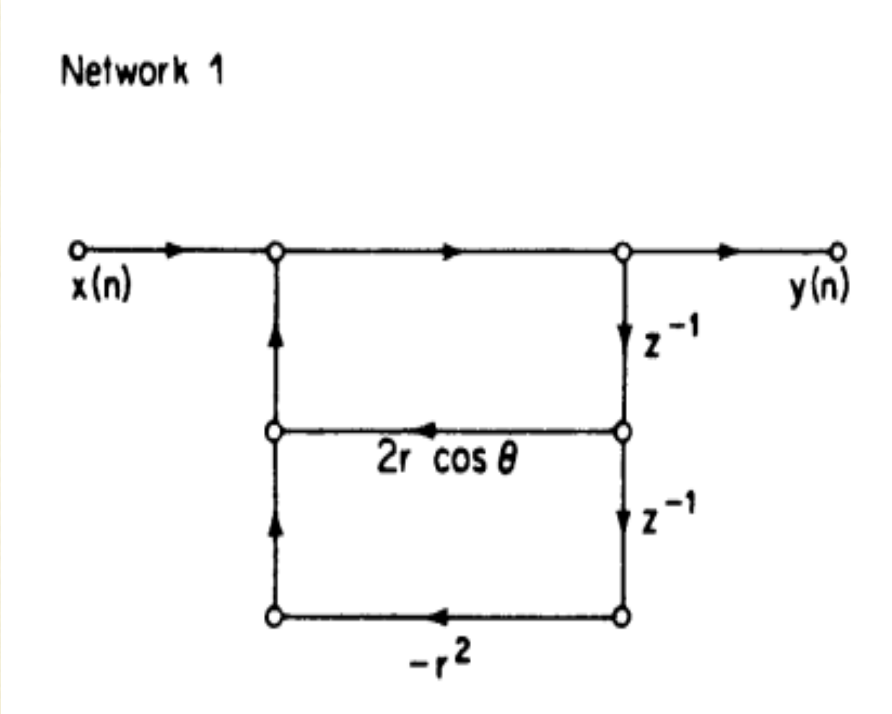
$$W_3(z) = 2W_2(z)$$

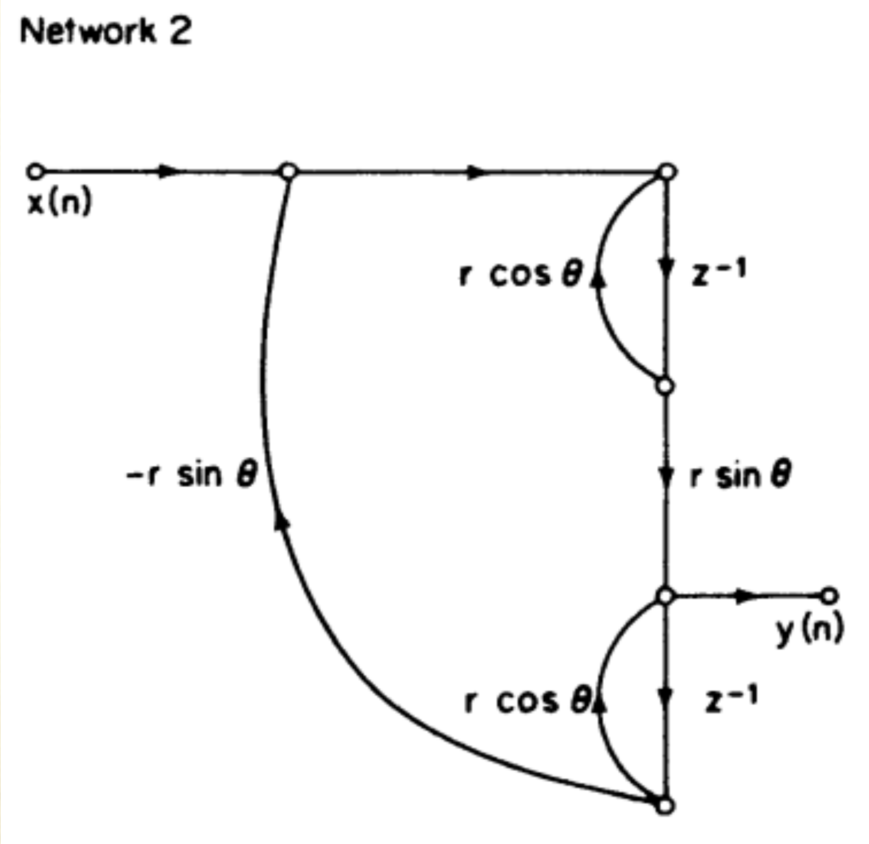
$$W_4(z) = 2W_1(z) + 3W_3(z)$$

$$Y(z) = W_3(z)$$

Example #5

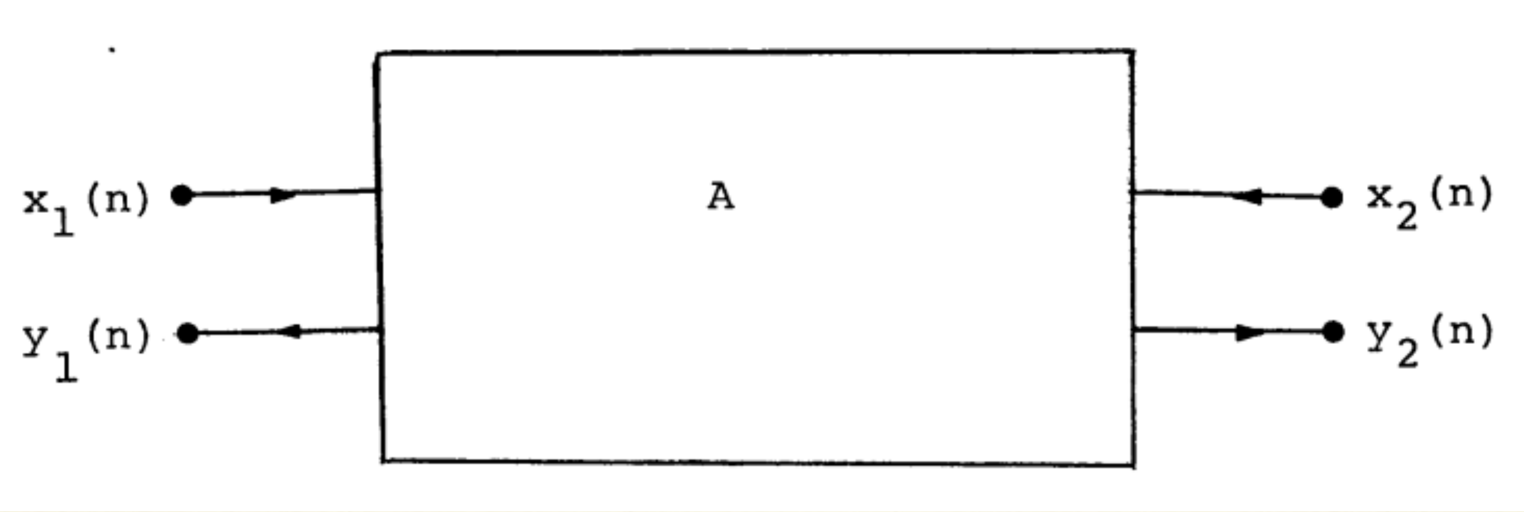
- Determine the system functions of the two networks shown below and show that they have the same poles.





Example #6

- In figure below is indicated a digital network A with two inputs, $x_1(n)$ and $x_2(n)$, and two outputs $y_1(n)$ and $y_2(n)$.



- The network A can be described in terms of the two part set of equations:

$$Y_1(z) = H_1(z) + H_2(z)X_2(z)$$

$$Y_2(z) = H_3(z) + H_4(z)X_2(z)$$

Example #6 (cont.)

- Where,

$$H_1(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

$$H_2(z) = 1$$

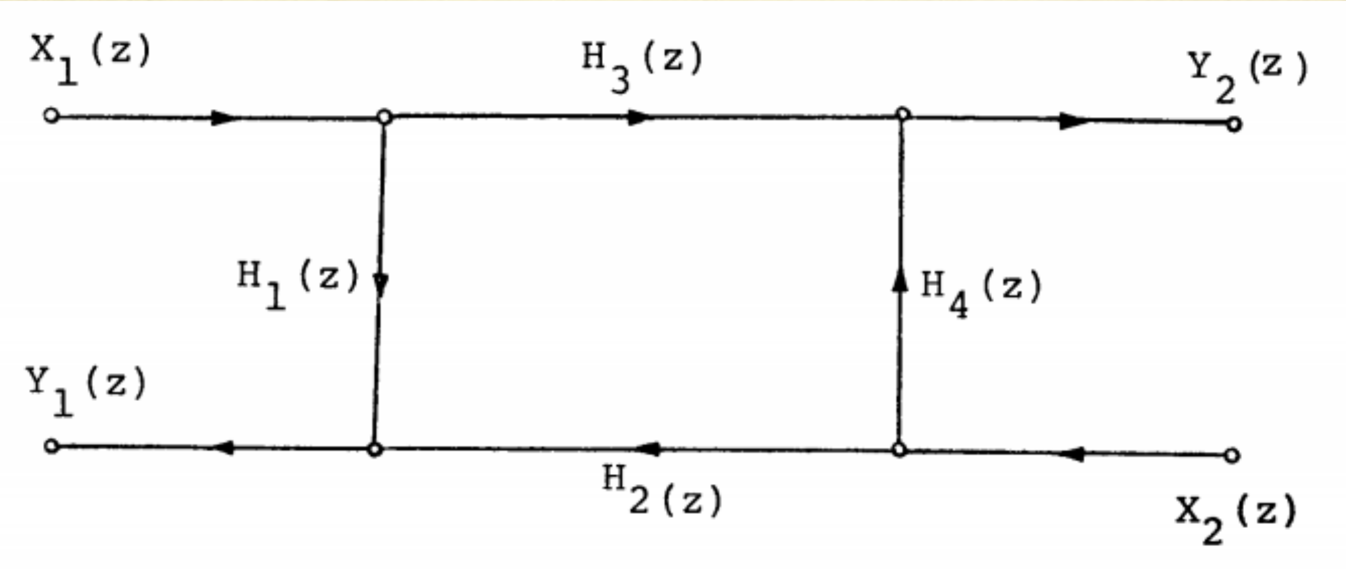
$$H_3(z) = \frac{1 + 2z^{-1}}{1 + \frac{1}{2}z^{-1}}$$

$$H_4(z) = \frac{1}{1 + \frac{1}{2}z^{-1}}$$

Example #6 (cont.)

- Draw a flow-graph implementation of the network. The transmittance of each branch must be a constant or a constant times z^{-1}. Higher-order functions of z^{-1} cannot be used as branch transmittances.

- Solution:
  - A flow graph in terms of H_1, H_2, H_3, and H_4 can be drawn as:



Example #6 (cont.)

- However we want to draw the flow-graph using branch transmittances which are constant or a constant times z^{-1}. Thus we replace H_1, H_2, H_3, and H_4 by their flow graph implementations to obtain: