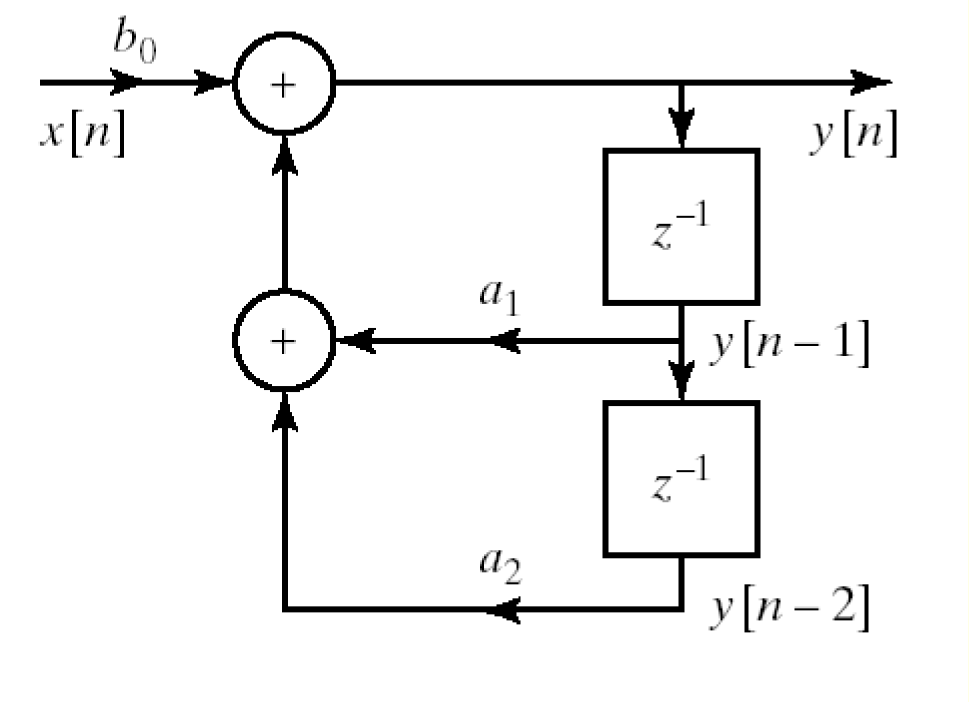
Discrete-Time System Functions

Example

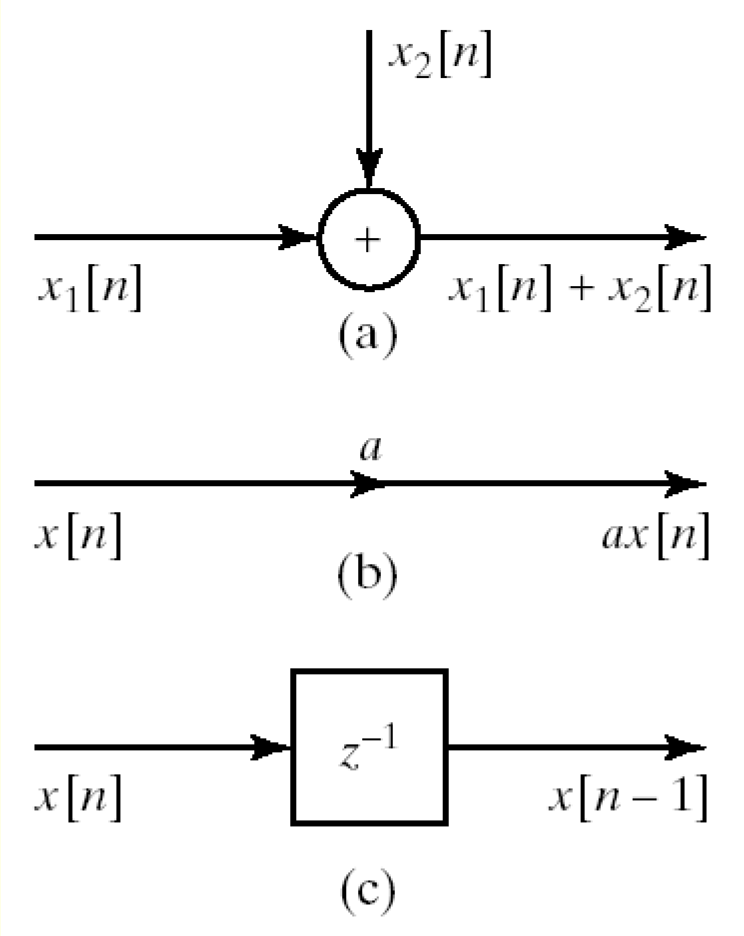
- Block diagram representation of

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n]$$



Block Diagram Representation

- LTI systems with rational system function can be represented as constant-coefficient difference equation
- The implementation of difference equations requires delayed values of the
  - input
  - output
  - intermediate results
- The requirement of delayed elements implies need for storage
- We also need means of
  - addition
  - multiplication



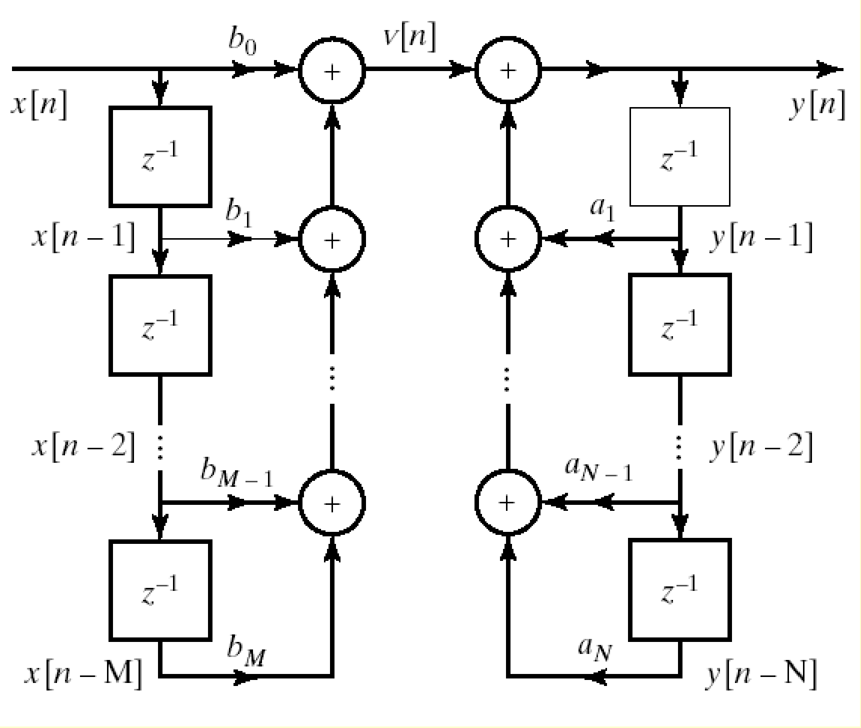
Direct Form I

- General form of difference equation

$$\sum_{k=0}^{N} \hat{a}_k y[n-k] = \sum_{k=0}^{M} \hat{b}_k x[n-k]$$

- Alternative equivalent form

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



Direct Form I

- Transfer function can be written as

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

- Direct Form I Represents

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

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

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

$$v[n] = \sum_{k=0}^{M} b_k x[n-k]$$

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + v[n]$$

Alternative Representation

- Replace order of cascade LTI systems

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

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

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

$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n]$$

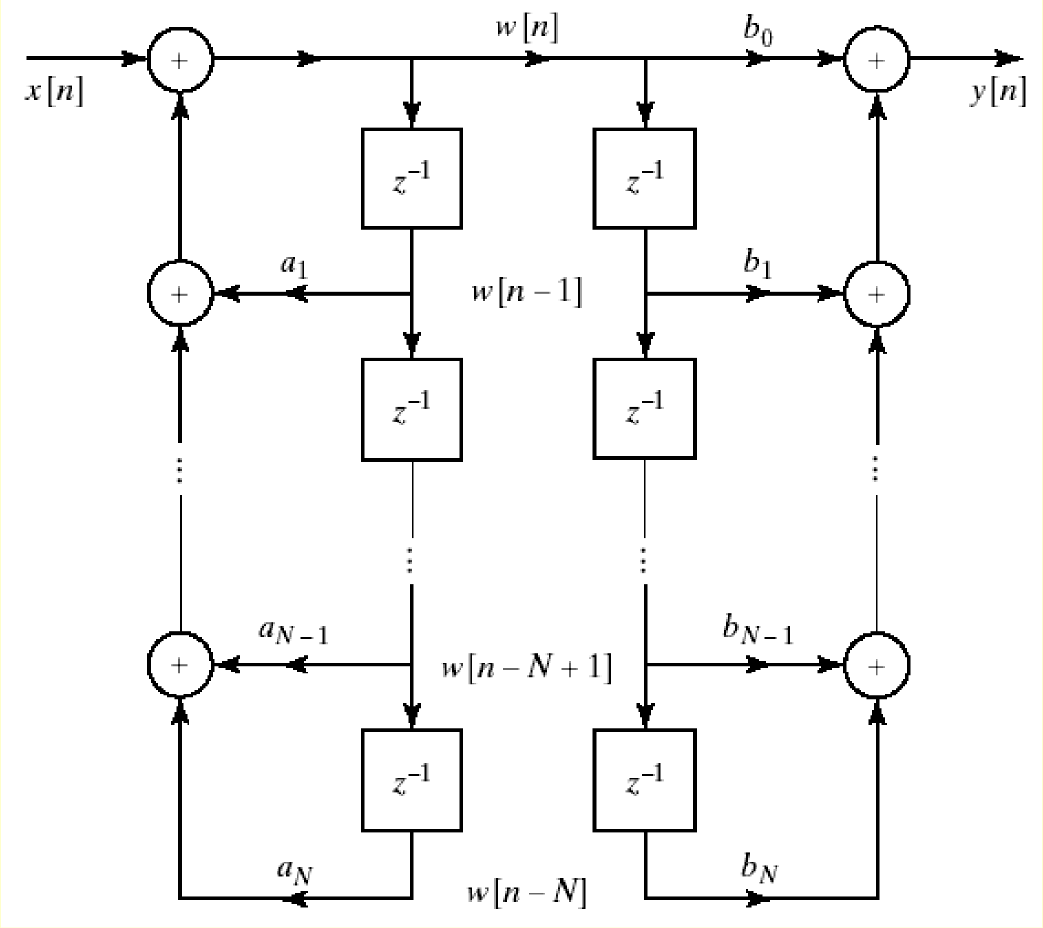
$$y[n] = \sum_{k=0}^{M} b_k w[n-k]$$

Alternative Block Diagram

- We can change the order of the cascade systems

$$w[n] = \sum_{k=1}^{N} a_k w[n-k] + x[n]$$

$$y[n] = \sum_{k=0}^{M} b_k w[n-k]$$



Direct Form II

- No need to store the same data twice in previous system
- So we can collapse the delay elements into one chain
- This is called Direct Form II or the Canonical Form
- Theoretically no difference between Direct Form I and II
- Implementation wise
  - Less memory in Direct II
  - Difference when using finite-precision arithmetic