Efficient Representation of Difference Equations

- Difference equation is a common way for realizing an LTI system (although not all LTI system is able to be implemented by difference equations)

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

- To represent a difference equation more efficiently, z-transform introduced below is widely used.

Z-transform

- Representing a discrete-time signal (or a sequence) as a polynomial.

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

- For CS students: It has the same form of the generating function for combinations in combinatorics.
  - But are **developed from different domains**.
  - In fact, **the difference equation is the same as the "recurrence relation" in combinatorics**.

Z-transform

- For a right-sided sequence ($x[n]$=0 for n<0), the z-transform is

$$X(z) = \sum_{n=0}^{\infty} x[n] z^{-n}$$

- **Example**: what is the **z-transform of delta-function**?

$$X(z) = \sum_{n=0}^{\infty} \delta[n] z^{-n} = 1$$

Example of z-transform for a finite sequence

n	$n \le -1$	0	1	2	3	4	5	$N > 5$
$x[n]$	0	2	4	6	4	2	1	0

$$X(z) = 2 + 4z^{-1} + 6z^{-2} + 4z^{-3} + 2z^{-4} + z^{-5}$$

Z-Transform vs. DTFT

- **Discrete-time Fourier Transform**

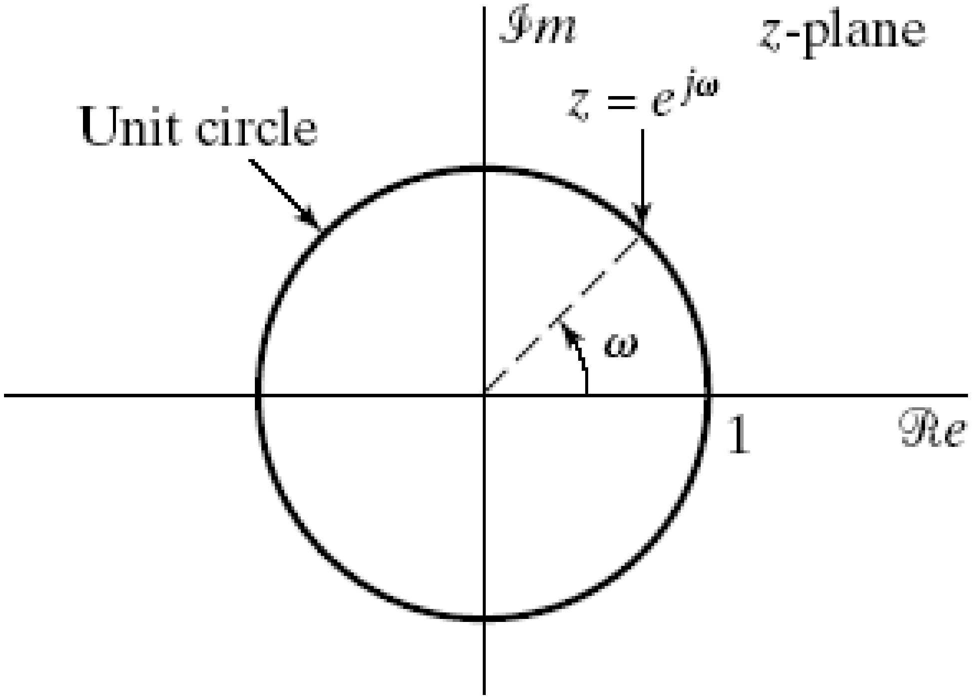
$$X\left(e^{jw}\right)=\sum_{n=-\infty}^{\infty}x[n]e^{-jwn}$$

- Hence, DTFT is equivalent to substituting $z = e^{j\omega}$ into the z-transform

$$X(z)=\sum_{n=-\infty}^{\infty}x[n]z^{-n}\Big|_{z=e^{jw}}$$

- More specifically, the z -transform is a generalization of the DTFT, where the DTFT evaluates the z-transform only on the complex unit circle ($|z| = 1$).

Z-Transform vs. DTFT



The unit circle in the complex z plane.
DTFT is only evaluated on this circle.

Z-transform vs. Convolution

- Remember that the convolution results can be seen as the coefficients of polynomial products.
- Since z-transform represents a sequence as a polynomial, it has the property that

Convolution in the n-domain corresponds to multiplication in the z-domain.

$$y[n] = h[n] * x[n] \quad \stackrel{z}{\longleftrightarrow} \quad Y(z) = H(z)X(z)$$

- Time domain convolution implies z-domain multiplication

Proof

(for the case of **causal systems** only, but the property holds for non-causal systems)

- Convolution $x(n) * h(n) = h(n) * x(n)$

$$y(n) = \sum_{k=0}^{\infty} x(k)h(n-k).$$

- Take the z-transform on both sides:

$$Y(z) = \sum_{n=0}^{\infty} y(n)z^{-n} = \sum_{n=0}^{\infty} \left[\sum_{k=0}^{\infty} x(k)h(n-k) \right] z^{-n}$$

Interchanging the order of summation, we obtain

$$Y(z) = \sum_{k=0}^{\infty} x(k) \sum_{n=0}^{\infty} h(n-k)z^{-n}$$

Proof (con't)

Let us make a substitution $m = n - k$, and now we have

$$Y(z) = \sum_{k=0}^{\infty} x(k) \sum_{m=-k}^{\infty} h(m) z^{-(m+k)}$$

$$= \sum_{k=0}^{\infty} x(k) z^{-k} \sum_{m=-k}^{\infty} h(m) z^{-m}$$

But $h(m) = 0$ for $-k \leq m \leq -1$, so that

$$Y(z) = \sum_{k=0}^{\infty} x(k) z^{-k} \sum_{m=0}^{\infty} h(m) z^{-m}$$

$$= X(z)H(z) = H(z)X(z)$$

Time domain convolution implies z-domain multiplication

Time Delay Property

- It can be easily shown that time delay of n_0 samples is equivalent to multiplying z^{-n_0} in the z-domain.

Time delay of n_0 samples multiplies the z-transform by z^{-n_0}.

$$x[n-n_0] \stackrel{z}{\longleftrightarrow} z^{-n_0}X(z)$$

Z-transform Applying to Systems

- In the above, z-transform is applied to a signal.
- Now, we apply it to the LTI system realized by a difference equation:

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

- Taking z-transforms for both sides, by using the time-delay property, we have

$$\sum_{k=0}^{N} a_k Y(z) z^{-k} = \sum_{m=0}^{M} b_m X(z) z^{-m}$$

Z-transform Applying to Systems

- Hence

$$\frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{\sum_{k=0}^{N} a_k z^{-k}}$$

- We call

$$\boxed{H(z) = \frac{Y(z)}{X(z)}}$$

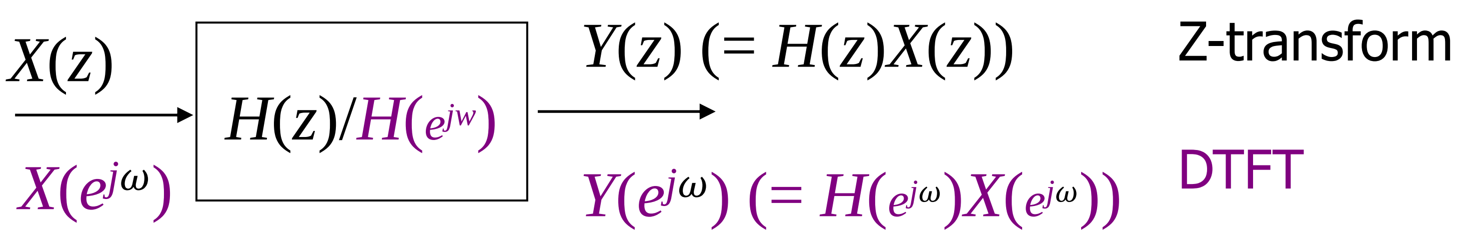
(which is a **fractional form**)

the **system function** of this LTI system

- General Definition of the system function: the **system function** of an LTI system is the z-transform of the output signal divided by the z-transform of the input signal.

System Function vs. Frequency Response

- Hence, when feeding an input signal $X(z)$ to an LTI system with the system function $H(z)$, the output is the product $Y(z) = X(z)H(z)$
- When $z = e^{j\omega}$, we obtain the output spectrum $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$.



Z-transform and Impulse Response

- In addition, the convolution in time domain implies multiplication in the z-domain (and also the frequency domain).
- So,

Time domain convolution

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n] = x[n] * h[n]$$

$\equiv$

$$\begin{array}{c} X(z) \\ X(e^{j\omega}) \end{array} \longrightarrow \boxed{H(z)/H(e^{j\omega})} \longrightarrow \begin{array}{c} Y(z)\ (= H(z)X(z)) \\ Y(e^{j\omega})\ (= H(e^{j\omega})X(e^{j\omega})) \end{array}$$

Z-transform

Fourier transform

System Function and Impulse Response

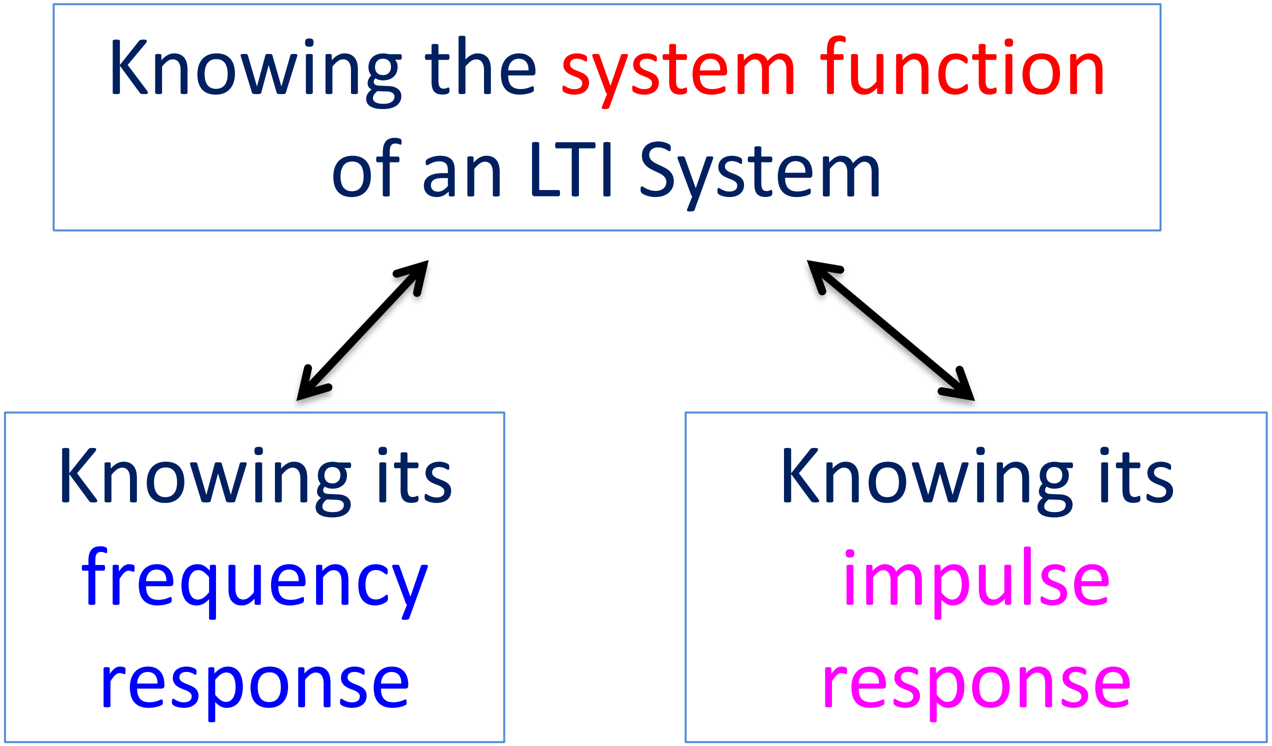
- When the input $x[n] = \delta[n]$, the z-transform of the impulse response should satisfy the following equation:

$$\mathrm{Z}\{h[n]\} = H(z)\mathrm{Z}\{\delta[n]\}.$$

- Since the z-transform of the unit impulse $\delta[n]$ is 1, we have

$$\mathrm{Z}\{h[n]\} = H(z)$$

- That is**, the system function $H(z)$ is the z-transform of the impulse response $h[n]$.**



- The system function $H(z)$ is the z-transform of the impulse response of the system.

Z-transform

- Hence, we can represent an LTI system by either $h[n]$ (impulse response), $H(e^{j\omega})$ (frequency response), or $H(z)$ (z-transform)
- In particular, we usually use $H(z)$ as a fractional form:

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

to represent a difference-equation LTI system.

- Although we use a causal system to define the system function in the above, the definition and properties are applicable to non-causal systems.

Z-transform of FIR system

- General form of causal FIR filter:

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

- So, the system function of a FIR filter contains only the denominator,

$$H(z) = \sum_{k=0}^{M} b[k]z^{-k}$$

Example

- Find the system function of the moving average filter,

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

Solution:

$$Y(z) = \frac{1}{3}(X(z) + X(z)z^{-1} + X(z)z^{-2}),$$

So, the system function is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{3}(1 + z^{-1} + z^{-2})$$

Frequency response is $H(\mathrm{e}^{j\omega}) = \frac{1}{3}(1 + e^{-j\omega} + e^{-j2\omega})$

Z-transform of IIR System

- General form of IIR filter

$$y[n] = \sum_{k=0}^{M} b[k]x[n-k] + \sum_{k=1}^{N} a[k]y[n-k]$$

- The system function is a fractional form containing both the denominator and nominator. In the z-domain,

$$Y(z) = \sum_{k=0}^{M} b[k]X(z)z^{-k} + \sum_{k=1}^{N} a[k]Y(z)z^{-k}$$

and thus

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=1}^{N} b[k]z^{-k}}{1 - \sum_{k=0}^{M} a[k]z^{-k}}$$

Example

- Find the system function of the IIR system,

$$y[n] = 2x[n] + 0.75y[n-1] - 0.125y[n-2]$$

Solution: in the z-domain,

$$Y(z) = 2X(z) + 0.75Y(z)z^{-1} - 0.125Y(z)z^{-2}$$

So, the system function is

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

Frequency response is $H\left(\mathrm{e}^{jw}\right) = \frac{2}{(1-0.75e^{-j\omega}+0.125e^{-j2\omega})}$

Example

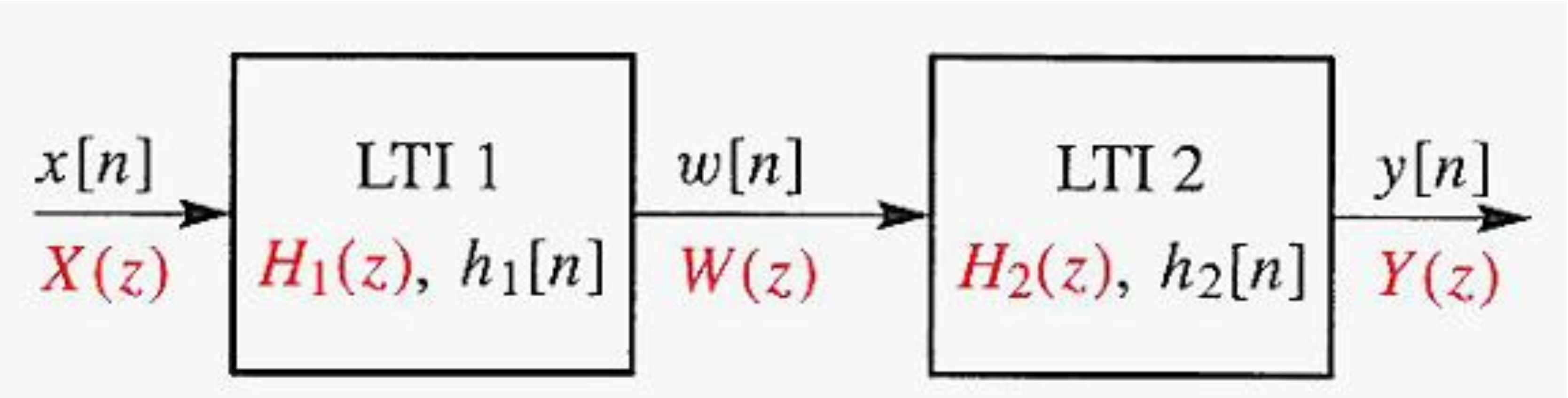
- Consider the FIR system $y[n] = 6x[n] - 5x[n-1] + x[n-2]$
- The z-transform system function is

$$H(z) = 6 - 5z^{-1} + z^{-2}$$

$$= (3 - z^{-1})(2 - z^{-1}) = 6\frac{\left(z - \frac{1}{3}\right)\left(z - \frac{1}{2}\right)}{z^2}$$

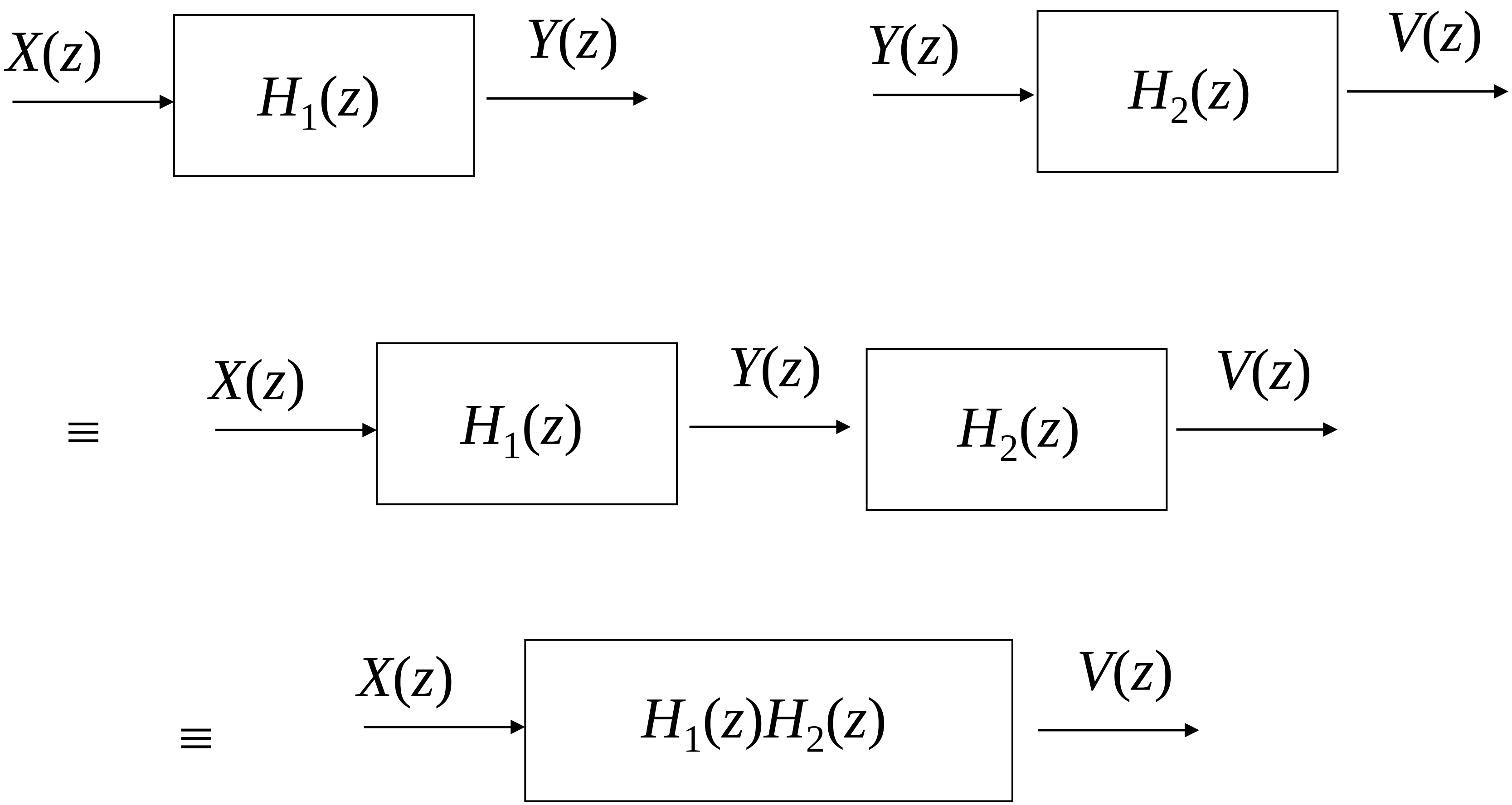
Cascading of LTI Systems

- Cascading of LTI systems can be described in the z-domain too.



- System function: $\dfrac{Y(z)}{X(z)} = H_2(z)H_1(z)$
- Hence, we have the following multiplication rule of a cascaded system (see next).

Multiplication Rule of Cascading System



Solving difference equations by Z-transform

- Difference equation:

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

- **Solving the difference equation**:
  - Giving an input signal $x[n]$, we want to find the signal $y[n]$ satisfying the above equation.

Solving difference equations by Z-transform (cont.)

- Solving it by z-transform: Taking z-transforms on both sides, we obtain

$$\sum_{k=0}^{N} a_k Y(z) z^{-k} = \sum_{m=0}^{M} b_m X(z) z^{-m}$$

so,

$$\frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{\sum_{k=0}^{N} a_k z^{-k}}$$

Solving difference equations by Z-transform (cont.)

- Remember that we have denoted

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

- Hence, given an input signal $x[n]$, we first find its Z-transform $X(z)$. Then, we find the z-transform of $Y(z)$ via $Y(z) = X(z)H(z)$. Finally, we find the **inverse z-transform** of $Y(z)$ to obtain $y[n]$.
  - **Note**: To find the inverse z-transform, we need to consider poles/zeros of an LTI system as well as the region of convergence of Z-transform, which will be introduced in the later course.

Z-transform vs. Laplace transform

Remark

- z-transform transfers a discrete-time signal to the z-domain.
- It is analogous to the Laplace transform for the continuous-time signal that is transformed to the s-domain.
- z-transform: for solving difference equations
- Laplace transform: for solving differential equations

Time Delay System

- Recall that time delay of n_0 samples is equivalent to convolving with $\delta(n-n_0)$ in time domain, or multiplying z^{-n_0} in the z-domain.

Time delay of n_0 samples multiplies the z-transform by z^{-n_0}.

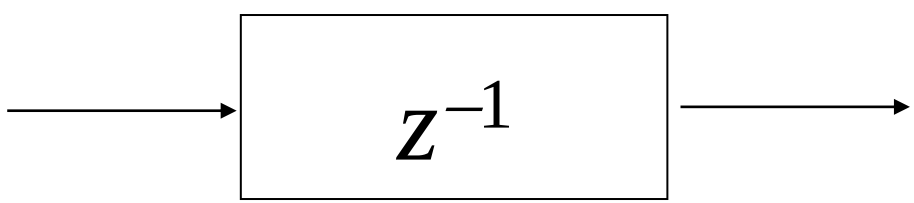
$$x[n-n_0] \quad \overset{z}{\longleftrightarrow} \quad z^{-n_0}X(z)$$

- We call z^{-1} the **unit-delay** system

Delay of one Sample

- More specifically, the system function of the FIR system $y[n] = x[n-1]$, (i.e., the one-sample-delay system), is

$$H(z) = z^{-1}$$



Delay of k Samples

- Similarly, the FIR system $y[n] = x[n-k]$, i.e., the k-sample-delay system, is the z-transform of the impulse response $\delta[n-k]$.

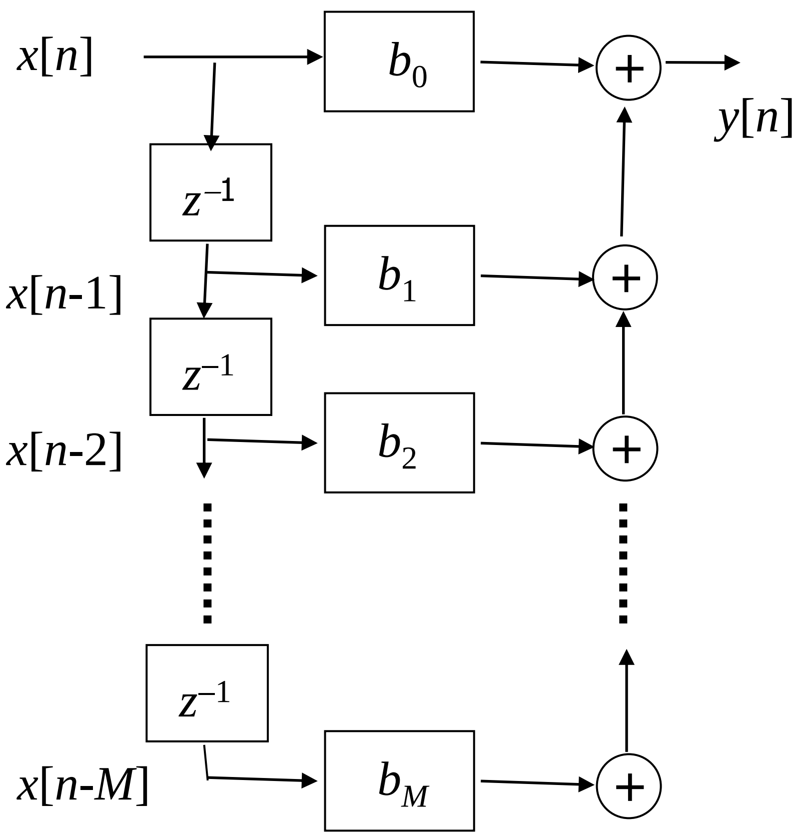
$$H(z) = z^{-k}$$

$$\longrightarrow \boxed{z^{-k}} \longrightarrow$$

System Diagram of A Causal FIR System

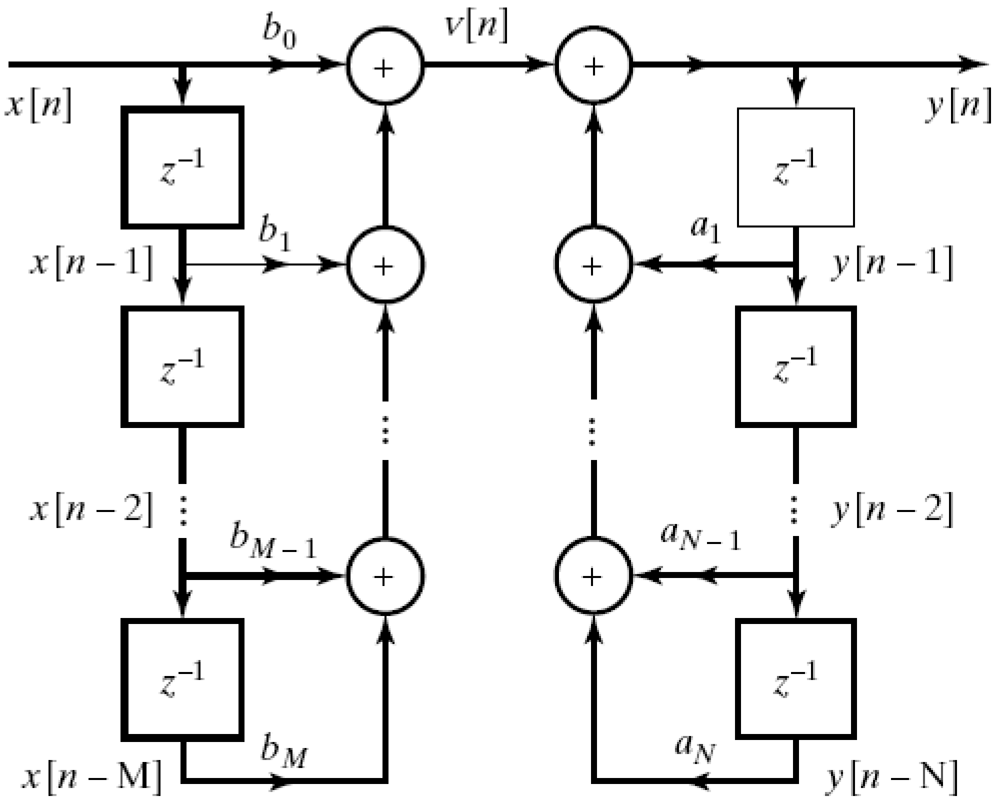
- The block diagram of a causal FIR system can be represented by z-transforms:

$$y[n] = \sum_{m=0}^{M} b_m x[n-m]$$



System Diagram of a Causal IIR Filter

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



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

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

Solution of Difference Equation

When will the difference equation system be LTI?

- Like differential equations for continuous-time systems, a linear constant-coefficient difference equation does not provide a unique solution if no additional constraints are provided.
- Additional constraints: consider the N auxiliary conditions that $y[-1], y[-2], \cdots, y[-N]$ are given.
  - The other values of $y[n]$ $(n \geq 0)$ can be generated by

$$y[n] = -\sum_{k=1}^{N} \frac{a_k}{a_0} y[n-k] + \sum_{m=0}^{M} \frac{b_m}{a_0} x[n-m]$$

when $x[n]$ is available, $y[0], y[1], \ldots\ y[n], \cdots$ can be computed recursively.

Example of the Solution

- Example: consider the difference equation

$$y[n] = a \cdot y[n-1] + x[n]$$

  - Assume the input is $x[n] = K\delta[n]$, and the auxiliary condition is $y[-1] = c$.
  - Hence, $y[0] = ac + K$,
    $y[1] = a \cdot y[0] + 0 = a^2c + aK, \cdots$
  - Recursively, we find that $y[n] = a^{n+1}c + a^n K$, for $n \geq 0$.

Example of the Solution

- Besides, we can also generate values of $y[n]$ for $n < -N$ recursively,

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

- Continue the above example:
  - For $n < -1$,

  $y[-2] = a^{-1}(y[-1] - x[-1]) = a^{-1}c$

  $y[-3] = a^{-1}y[-1] = a^{-2}c$, ..., and thus

  $y[n] = a^{n+1}c$ for $n < -1$.

- Combine the above solutions for $n \geq 0$ and $n < -1$, the entire solution of the example is

$$y[n] = a^{n+1}c + Ka^n u[n]$$

Example of the Solution (continue)

Discussion:

- The above solution system is non-linear:
  - When $K = 0$, i.e., the input is a zero sequence, the solution (system response) $y[n] = a^{n+1}c$.
  - Since a linear system requires that the output be zero when the input is a zero for all time. So, the system is **non-linear** when c is nonzero.
- The solution system is not time-invariant:
  - when input were shifted by n_0 samples, $x_1[n] = K\delta[n - n_0]$, the output is $y_1[n] = a^{n+1}c + Ka^{n-n_0}u[n - n_0]$. So, the system is not shift-invariant when c is nonzero.

LTI solution

- We are often interested in the systems that are linear and time invariant.
  - How to make the recursively-implemented solution system be LTI?
- Initial-rest condition: If the input $x[n]$ is zero for n less than some time n_0, the output $y[n]$ is also zero for n less than n_0.
  - The previous example does not satisfy this condition since $x[n] = 0$ for $n < 0$ but $y[-1] = c$.
- Property:
  - If the initial-rest condition is satisfied, then the difference equation system will be LTI and causal, and the solution can be obtained by using inverse Z-transform (which will be introduced in the future)