Convolution

EGC 113

LTI Systems

$x[n] = \Sigma\, x[k]\, \delta[n-k]$ $\rightarrow$ LTI $\rightarrow$ $y[n] = \Sigma\, x[k]\, h[n-k]$

Index $k = -\infty \; to \; +\infty$

Index $k = -\infty \; to \; +\infty$

Here, $y[n] = \Sigma\, x[k]\, h[n-k]$ is called the Convolution Sum

Index $k = -\infty \; to \; +\infty$

Convolution Operation, $x[n] * h[n]$

Operator is "$*$"

Defined as , $x[n] * h[n] = \Sigma\, x[k]\, h[n-k]$; Index $k = -\infty \; to \; +\infty$

In Summary, for an LTI System, to determine the O/P of the system to any arbitrary I/P, all we need to know is : Impulse Response, $h[n]$

i.e. LTI System is completely characterized by its Impulse Response

Convolution Sum

$$x[n] * h[n] = \Sigma\, x[k]\, h[n-k] \text{ ; Index } k = -\infty \; to \; +\infty$$

Steps Involved :

1) Time Reversal of h : $h[k]$ is time-reversed to obtain $h[-k]$

2) Shift it by “n” steps to obtain $h[n-k]$

3) $x[k]$ and $h[n-k]$ are point-wise multiplied for all values of “k” for a given value of “n”

4) Repeat for all values of “n”

Convolution Integral and Convolution Sum

- ***<u>What you often see:</u>***

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\,d\tau \qquad\qquad y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

Also represented as

$$y(t) = x(t) * h(t) \qquad\qquad y[n] = x[n] * h[n]$$

Properties

- The convolution operation is ***commutative***. That is, for any two functions x and h,

$$x * h = h * x.$$

- The convolution operation is ***associative***. That is, for any signals x, h_1, and h_2,

$$(x * h_1) * h_2 = x * (h_1 * h_2).$$

- The convolution operation is ***distributive*** with respect to addition. That is, for any signals x, h_1, and h_2,

$$x * (h_1 + h_2) = x * h_1 + x * h_2.$$

Example

- Determine the output y[n], given
  - x[n] = u[n]

$$h[n] = \left(\frac{1}{2}\right)^{n-3} u[n-3]$$

$$x[n] = \begin{cases} 1, & 0 \le n \le 4 \\ 0, & \text{otherwise} \end{cases}$$

$$h[n] = \begin{cases} \alpha^n, & 0 \le n \le 6 \\ 0, & \text{otherwise} \end{cases}.$$

Useful Mathematical Relationships

$$\sum_{n=0}^{\infty} \alpha^n = \frac{1}{1-\alpha}, \text{ which converges only for } |\alpha| < 1, \text{ and}$$

$$\sum_{n-0}^{N-1} \alpha^n = \frac{1-\alpha^N}{1-\alpha}, \text{ which is a finite sum and hence always converges}$$

Example

- Compute y[n], given

$$x[n] = 2^n u[-n],$$
$$h[n] = u[n].$$

$$x(t) = e^{2t} u(-t),$$
$$h(t) = u(t-3).$$

$$x[n] = \left(\frac{1}{2}\right)^n u[n] + 2^n u[-n],$$
$$h[n] = u[n].$$

Example

- Are the systems shown earlier:
- 1. Causal
- 2. BIBO Stable
- 3. Invertible
- 4. Memoryless

Example

- Consider a discrete LTI system whose input and output are related by

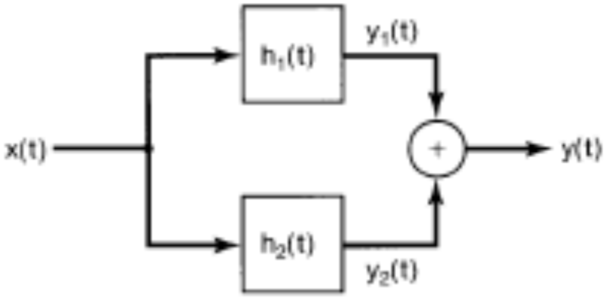
- $y[n] = \sum_{k=-\infty}^{n} 2^{k-n} x[k+1]$

- Determine the impulse response of the system and if it is stable, causal.

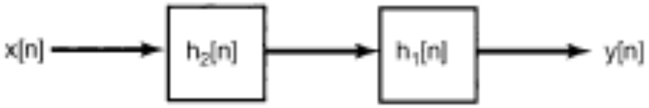
Example

- Consider a system below, where
- $h_1(t) = e^{-2t}u(t)\ h_2(t) = 2e^{-t}u(t)$

- Find the impulse response of the overall system.
- Is the system stable?



(a)



Example

- Consider a system below, where
- $h_1(t) = u(t)\ h_2(t) = \delta(t-1)$

- Find the impulse response of the overall system.
- Is the system stable?

Convolution

- LTI systems can be represented as a the convolution of the input with an impulse response.
- Convolution has many useful properties (associative, commutative, etc).

- These carry over to LTI systems
  - Composition of system blocks
  - Order of system blocks

- Useful both practically, and for understanding. While convolution is conceptually simple, it can be practically difficult. It can be tedious to convolve your way through a complex system.
- There has to be a better way ...

Example

- Suppose Ram's parents opens a savings account on his 1st birthday and deposit rupees 50 on the first of each month in the account. Determine the money in the account after 1 year and 10 years if the interest rate is 1% per month.

Example

- The interest calculation can be done easily using linear difference equation. Let y[n] be the amount in his account at nth month.
- $y[n] = y[n-1] + 0.01y[n-1] + x[n]$
- $y[n] = 1.01y[n-1] + x[n]$
- Assuming zero initial conditions, i.e. y[-1] = 0

- The amount after 1 year will be y[12]. Can you calculate that? What about y[120]?

Example

- Let's write equation in the form of a geometric series:
- $y[n] = \alpha y[n-1] + \beta x[n]$
- In our case, y[-1] = 0 and x[0]=x[1]=….=D (=50) and β =1.
- $y[0] = \beta D$
- $y[1] = \beta D(1+\alpha)$
- $y[2] = \alpha y[1] + \beta D = \beta D(\alpha^2 + \alpha + 1)$
- ….
- $y[N] = \beta D(\alpha^N + \alpha^{N-1} + \cdots + \alpha + 1) = \beta D \sum_{m=0}^{N} \alpha^m = \beta D \frac{1-\alpha^{N+1}}{1-\alpha}$

Example

- In our case, α = 1.01, b = 1, D =50, N = 12 and 120,
- The amount will be,
- $y[12] = 50\,\frac{1-1.01^{13}}{1-1.01}$= 690.47
- $y[120] = 50\,\frac{1-1.01^{121}}{1-1.01} = 1166.95$
- How will you calculations change if the interest was computed quarterly or annually?

Multiple Representations of Discrete-Time Systems

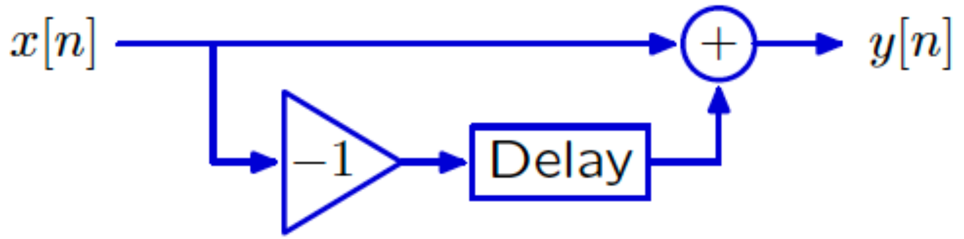
Systems can be represented in different ways to more easily address different types of issues.

Verbal description: 'To reduce the number of bits needed to store a sequence of large numbers that are nearly equal, record the first number, and then record successive differences.'

Difference equation:

$$y[n] = x[n] - x[n-1]$$

Block diagram:



We will exploit particular strengths of each of these representations.

Difference Equations

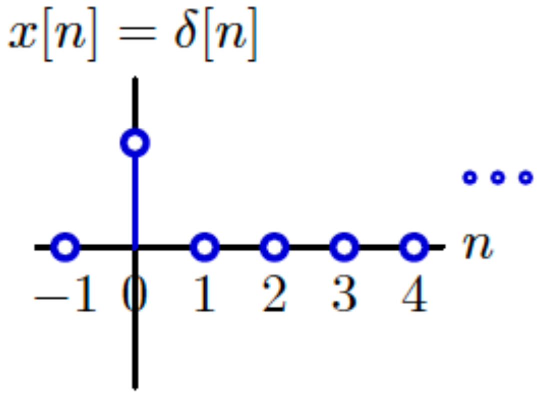
Difference equations are mathematically precise and compact.

Example:

$$y[n] = x[n] - x[n-1]$$

Let $x[n]$ equal the "unit sample" signal $\delta[n]$,

$$\delta[n] = \begin{cases} 1, & \text{if } n = 0; \\ 0, & \text{otherwise.} \end{cases}$$



We will use the unit sample as a "primitive" (building-block signal) to construct more complex signals.

Check Yourself

Solve

$$y[n] = x[n] - x[n-1]$$

given

$$x[n] = \delta[n]$$

How many of the following are true?

1. $y[2] > y[1]$
2. $y[3] > y[2]$
3. $y[2] = 0$
4. $y[n] - y[n-1] = x[n] - 2x[n-1] + x[n-2]$
5. $y[119] = 0$

In what ways are difference equations different from block diagrams?

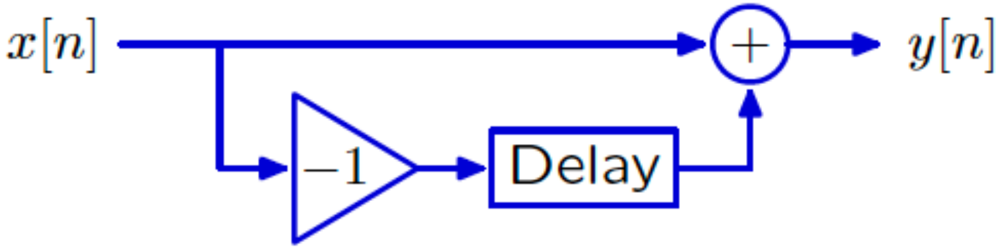
Difference equation:

$$y[n] = x[n] - x[n-1]$$

Difference equations are "declarative."

They tell you rules that the system obeys.

Block diagram:



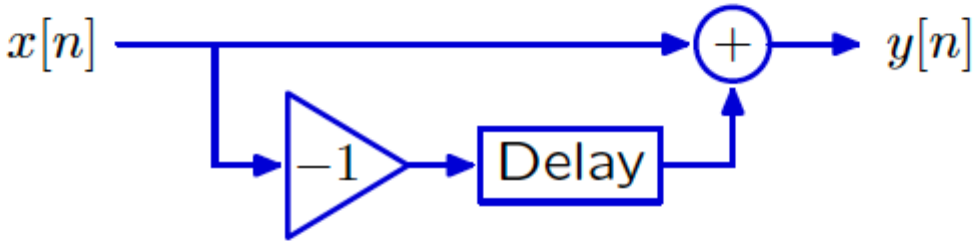
Block diagrams are "imperative."

They tell you what to do.

Block diagrams contain **more** information than the corresponding difference equation (e.g., what is the input? what is the output?)

From Samples to Signals

Operators manipulate signals rather than individual samples.



Nodes represent whole signals (e.g., X and Y).

The boxes **operate** on those signals:

- Delay = shift whole signal to right 1 time step
- Add = sum two signals
- -1: multiply by -1

Signals are the primitives.

Operators are the means of combination.

Operator Notation

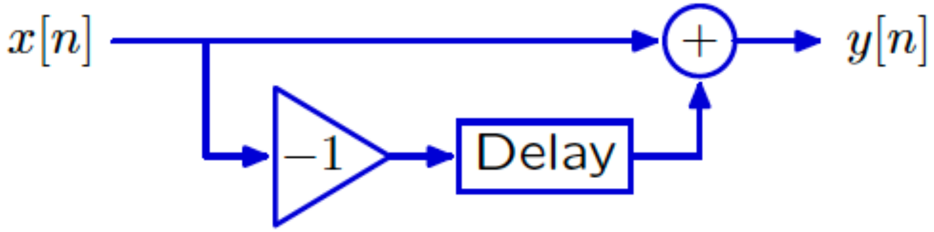
Symbols can now compactly represent diagrams.

Let $\mathcal{R}$ represent the right-shift **operator**:

$$Y = \mathcal{R}\{X\} \equiv \mathcal{R}X$$

where X represents the whole input signal ($x[n]$ for all n) and Y represents the whole output signal ($y[n]$ for all n)

Representing the difference machine



with $\mathcal{R}$ leads to the equivalent representation

$$Y = X - \mathcal{R}X = (1 - \mathcal{R})\, X$$

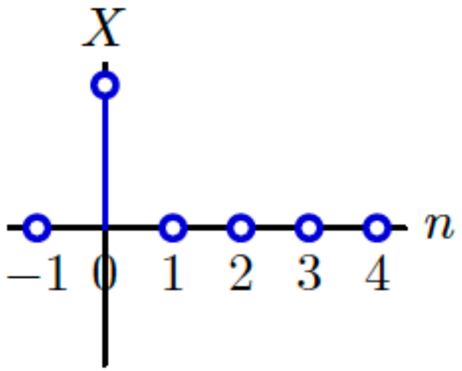
Operator Notation: Check Yourself

Let $Y = \mathcal{R}X$. Which of the following is/are true:

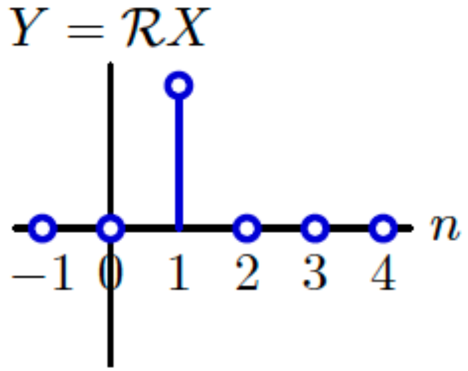
1. $y[n] = x[n]$ for all n
2. $y[n+1] = x[n]$ for all n
3. $y[n] = x[n+1]$ for all n
4. $y[n-1] = x[n]$ for all n
5. none of the above

Check Yourself

Consider a simple signal:



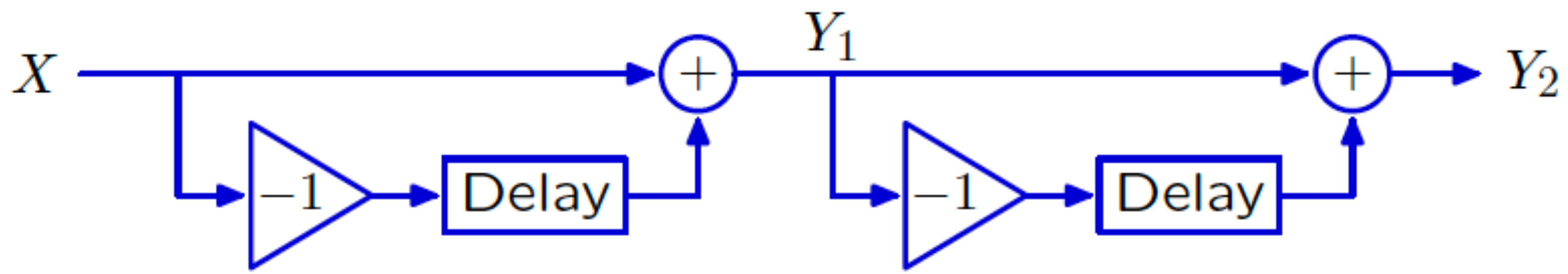
Then



Operator Representation of a Cascaded System

System operations have simple operator representations.

Cascade systems $\rightarrow$ multiply operator expressions.



Using operator notation:

$$Y_1 = (1 - \mathcal{R})\, X$$

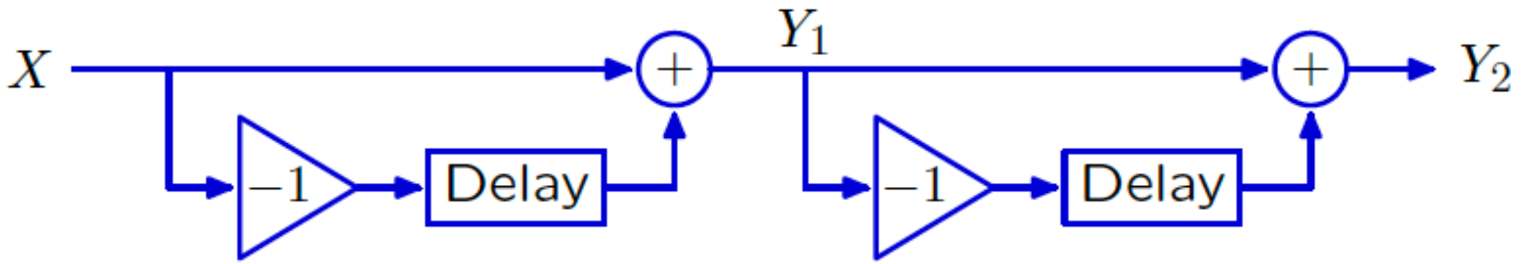
$$Y_2 = (1 - \mathcal{R})\, Y_1$$

Substituting for Y_1:

$$Y_2 = (1 - \mathcal{R})(1 - \mathcal{R})\, X$$

Operator Algebra

Operator expressions can be manipulated as polynomials.



Using difference equations:

$$
\begin{aligned}
y_2[n] &= y_1[n] - y_1[n-1] \\
&= (x[n] - x[n-1]) - (x[n-1] - x[n-2]) \\
&= x[n] - 2x[n-1] + x[n-2]
\end{aligned}
$$

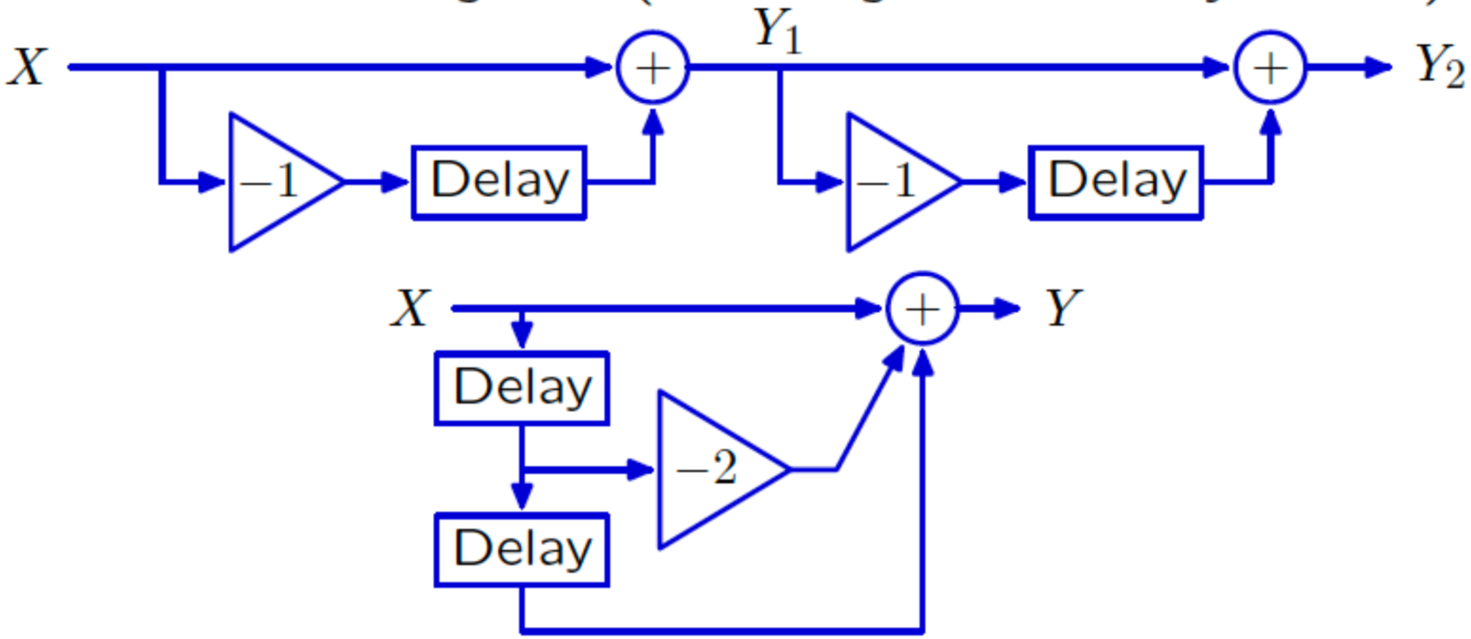
Using operator notation:

$$
\begin{aligned}
Y_2 &= (1-\mathcal{R})\,Y_1 = (1-\mathcal{R})(1-\mathcal{R})\,X \\
&= (1-\mathcal{R})^2 X \\
&= (1-2\mathcal{R}+\mathcal{R}^2)\,X
\end{aligned}
$$

Operator Algebra

Operator notation facilitates seeing relations among systems.

"Equivalent" block diagrams (assuming both initially at rest):



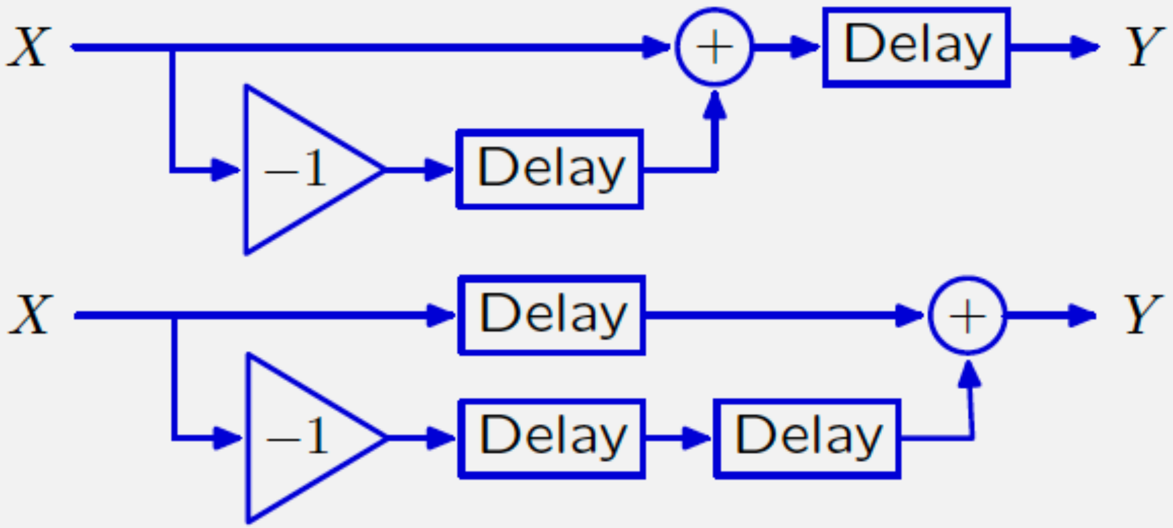
Equivalent operator expressions:

$$(1-\mathcal{R})(1-\mathcal{R}) = 1 - 2\mathcal{R} + \mathcal{R}^2$$

The operator equivalence is much easier to see.

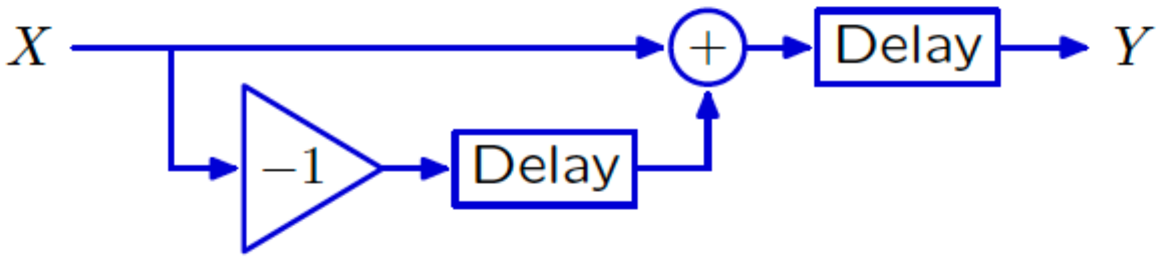
Check Yourself

Operator expressions for these "equivalent" systems (if started "at rest") obey what mathematical property?

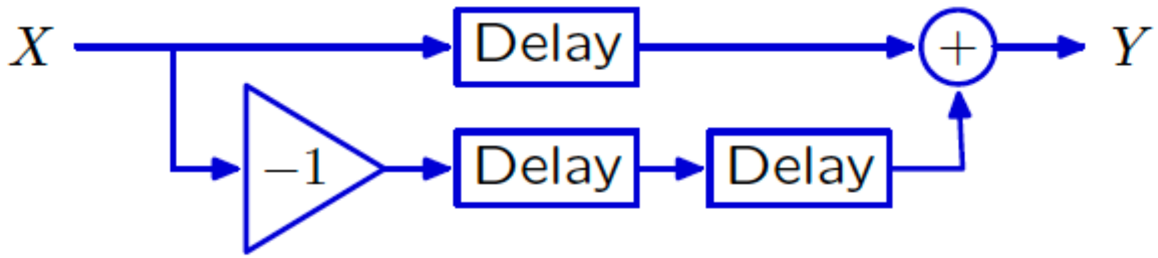


1. commutate
2. associative
3. distributive
4. transitive
5. none of the above

Check Yourself



$$Y = \mathcal{R}(1 - \mathcal{R})X$$



$$Y = (\mathcal{R} - \mathcal{R}^2)X$$

Multiplication by $\mathcal{R}$ distributes over addition.

Check Yourself

How many of the following systems are equivalent to

$$Y = (4\mathcal{R}^2 + 4\mathcal{R} + 1)\,X \quad ?$$