6.003: Signals and Systems

Discrete-Time Systems

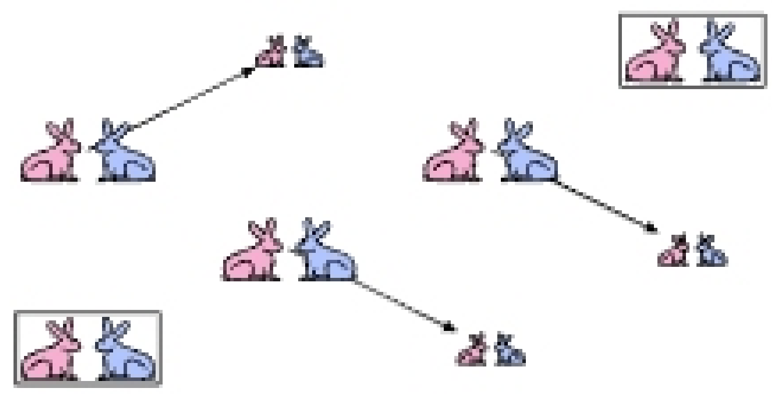
September 15, 2009

Discrete-Time Systems

We start with discrete-time systems because they

- are conceptually simpler than continuous-time systems
- illustrate the same important modes of thinking
- are increasingly important (digital electronics and computation)

Example: Population Growth



Multiple Representations of Discrete-Time Systems

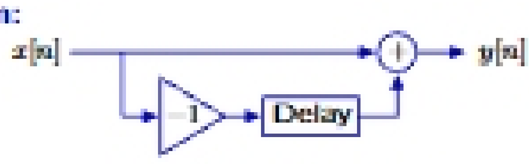
Systems can be represented in different ways to more easily address different types of issues.

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

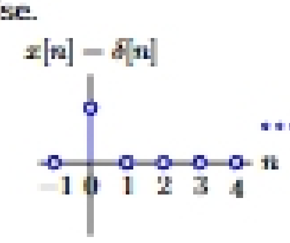
Difference equations are mathematically precise and compact.

Example:

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

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

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



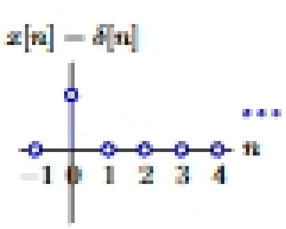
We will use the unit sample as a "primitive" (building-block signal) to construct more complex signals.

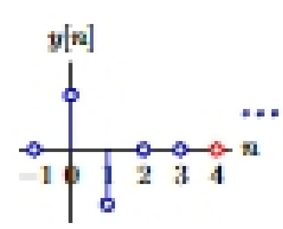
Step-By-Step Solutions

Difference equations are convenient for step-by-step analysis.

Find $y[n]$ given $x[n] = \delta[n]$: $y[n] = x[n] - x[n-1]$

$$\begin{aligned}
y[-1] &= x[-1] - x[-2] && = 0 - 0 = 0 \\
y[0] &= x[0] - x[-1] && = 1 - 0 = 1 \\
y[1] &= x[1] - x[0] && = 0 - 1 = -1 \\
y[2] &= x[2] - x[1] && = 0 - 0 = 0 \\
y[3] &= x[3] - x[2] && = 0 - 0 = 0 \\
&\ldots
\end{aligned}$$

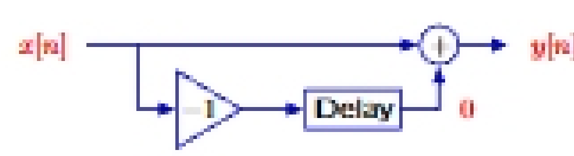


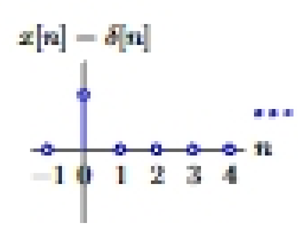


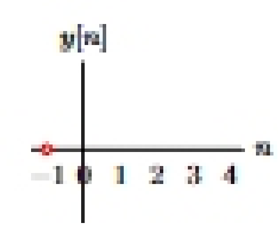
Step-By-Step Solutions

Block diagrams are also useful for step-by-step analysis.

Represent $y[n] = x[n] - x[n-1]$ with a block diagram: start "at rest"







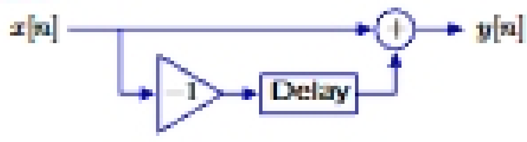
Check Yourself

DT systems can be described by difference equations and/or block diagrams.

Difference equation:

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

Block diagram:



In what ways are these representations different?

From Samples to Signals

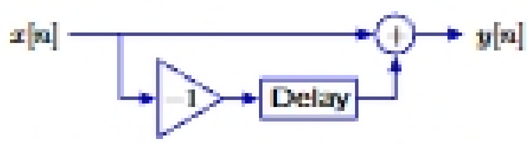
Lumping all of the (possibly infinite) samples into a single object — the signal — simplifies its manipulation.

This lumping is an **abstraction** that is analogous to

- representing coordinates in three-space as points
- representing lists of numbers as vectors in linear algebra
- creating an object in Python

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

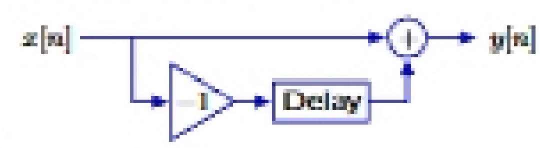
Symbols can now compactly represent diagrams.

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

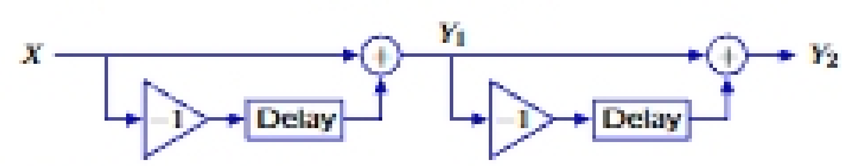
Let $Y = \mathcal{R}X$. Which of the following is/are true:

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

Operator Representation of a Cascaded System

System operations have simple operator representations.

Cascade systems → multiply operator expressions.



Using operator notation:

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

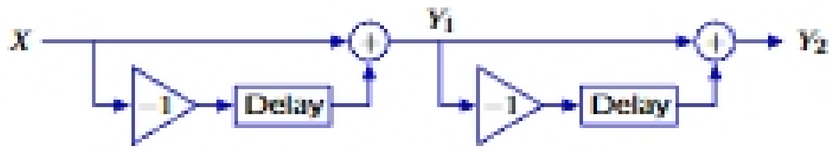
$Y_2 = (1 - \mathcal{R})\,Y_1$

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:

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

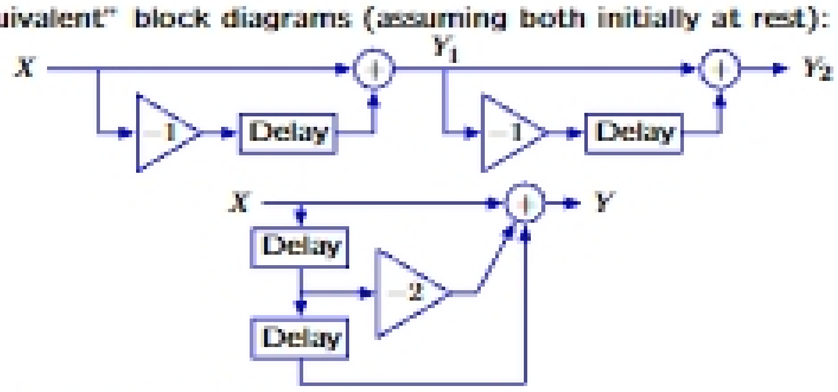
Operator Approach

Applies your existing expertise with polynomials to understand block diagrams, and thereby understand systems.

Operator Algebra

Operator notation facilitates seeing relations among systems.

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



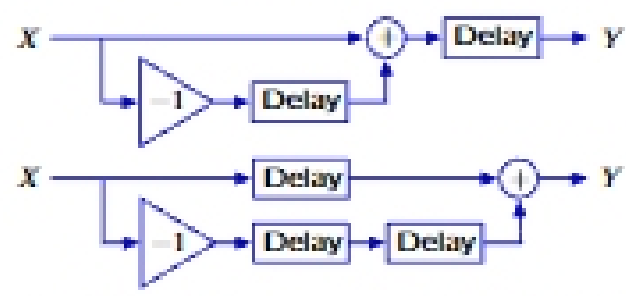
Equivalent operator expressions:

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

The operator equivalence is much easier to see.

Check Yourself

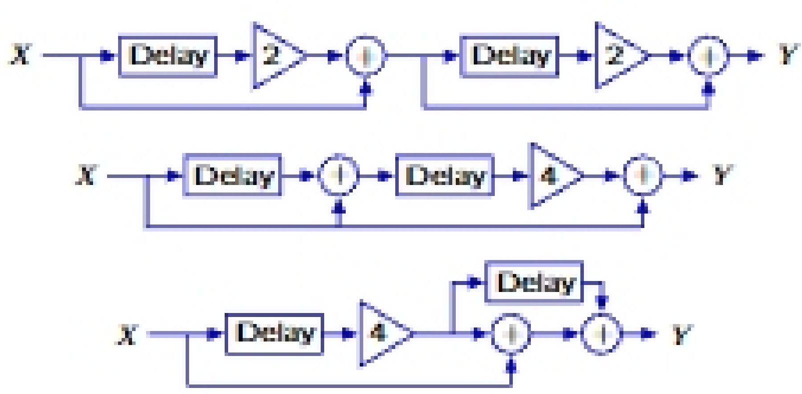
Operator expressions for these "equivalent" systems (if started "at rest") obey what mathematical property?



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

Check Yourself

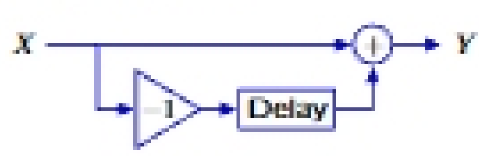
How many of the following systems are equivalent to $Y = (4\mathcal{R}^2 + 4\mathcal{R} + 1)\,X$?



Operator Algebra: Explicit and Implicit Rules

Recipes versus constraints.

Recipe: subtract a right-shifted version of the input signal from a copy of the input signal.



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

Constraint: the difference between Y and $\mathcal{R}Y$ is X.

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

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

But how does one solve such a constraint?