6.01: Introduction to EECS I

Signals and Systems

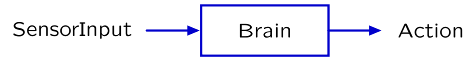
February 15, 2011

Module 1 Summary: Software Engineering

Focused on **abstraction** and **modularity** in software engineering.

Topics: procedures, data structures, objects, state machines

Lab Exercises: implementing robot controllers as state machines



Abstraction and Modularity: Combinators

- **`Cascade`**: make new SM by cascading two SM's
- **`Parallel`**: make new SM by running two SM's in parallel
- **`Select`**: combine two inputs to get one output

Themes: PCAP

Primitives – **C**ombination – **A**bstraction – **P**atterns

6.01: Introduction to EECS I

The **intellectual themes** in 6.01 are recurring themes in EECS:

- design of complex systems
- modeling and controlling physical systems
- augmenting physical systems with computation
- building systems that are robust to uncertainty

Intellectual themes are developed in context of a mobile robot.

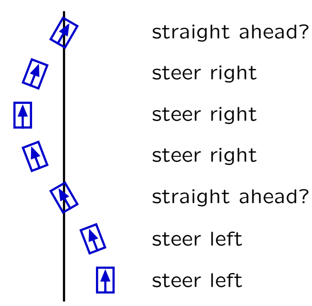
Goal is to convey a distinct perspective about engineering.

Module 2 Preview: Signals and Systems

Focus next on **analysis** of feedback and control systems.

Topics: difference equations, system functions, controllers.

Lab exercises: robotic steering

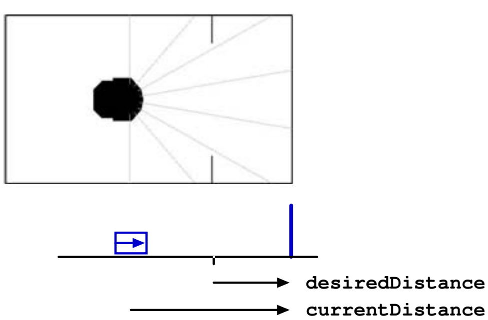


Themes: modeling complex systems, analyzing behaviors

Analyzing (and Predicting) Behavior

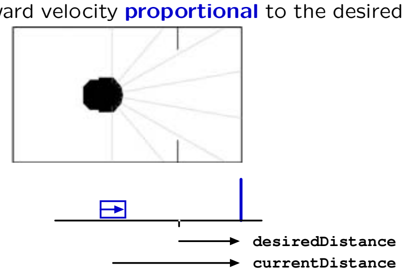
Today we will start to develop tools to **analyze** and predict behavior.

Example (design Lab 2): use sonar sensors (i.e., **`currentDistance`**) to move robot **`desiredDistance`** from wall.



Analyzing (and Predicting) Behavior

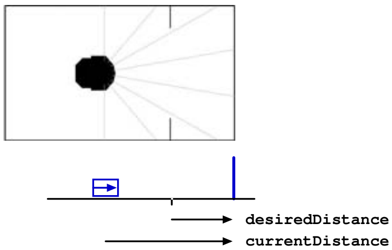
Make the forward velocity **proportional** to the desired displacement.




```
>>> class wallFinder(sm.SM):
...     startState = None
...     def getNextValues(self, state, inp):
...         desiredDistance = 0.5
...         currentDistance = inp.sonars[3]
...         return (state,io.Action(fvel=?,rvel=0))
```

Find an expression for **`fvel`**.

Check Yourself

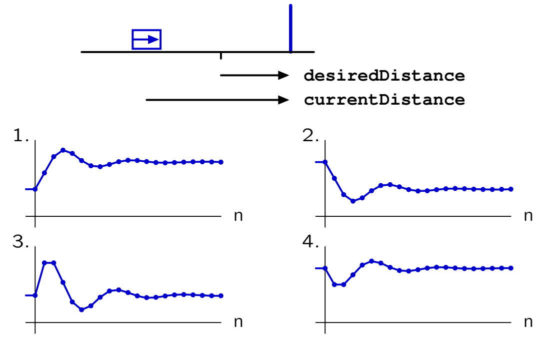


Which expression for **fvel** has the correct form?

1. **currentDistance**
2. **currentDistance-desiredDistance**
3. **desiredDistance**
4. **currentDistance/desiredDistance**
5. none of the above

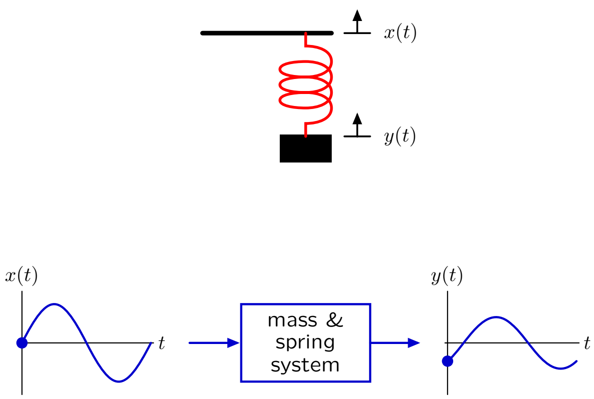
Check Yourself

Which plot best represents **currentDistance**?

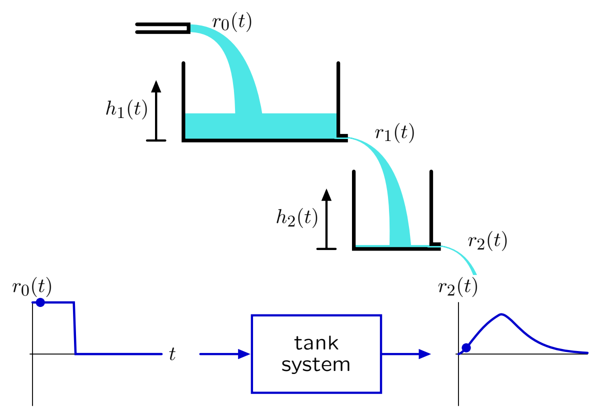


5. none of the above

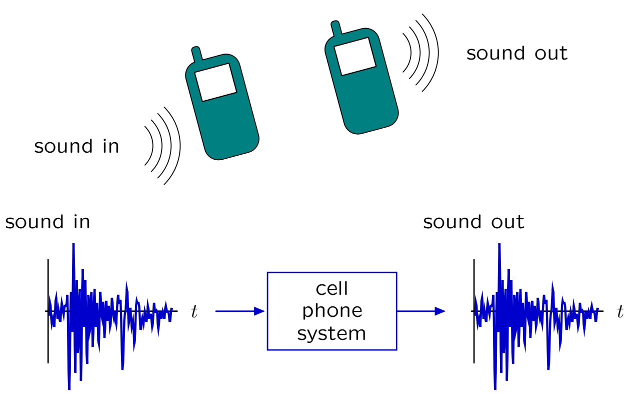
Example: Mass and Spring



Example: Tanks

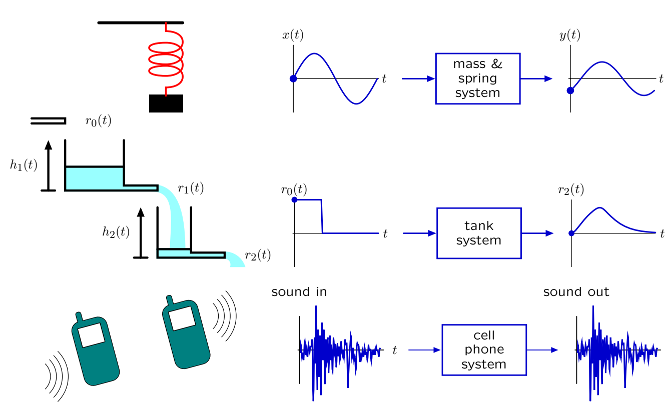


Example: Cell Phone System



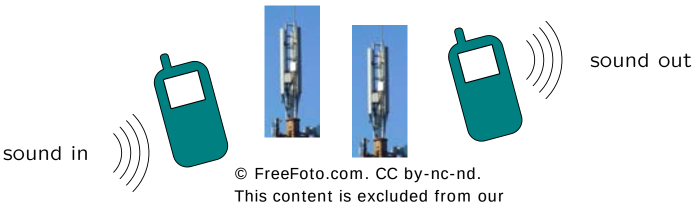
Signals and Systems: Widely Applicable

The Signals and Systems approach has broad application: electrical, mechanical, optical, acoustic, biological, financial, ...

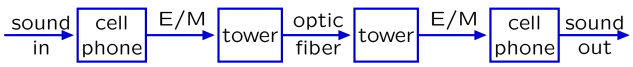


Signals and Systems: Modular

The representation does not depend upon the physical substrate.



© FreeFoto.com. CC by-nc-nd.
This content is excluded from our
Creative Commons license. For more
information, see http://ocw.mit.edu/fairuse.

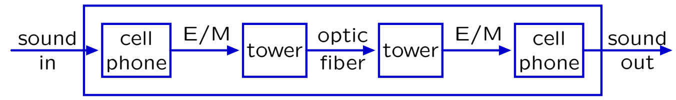


focuses on the flow of **information**, abstracts away everything else

Signals and Systems: Hierarchical

Representations of component systems are easily combined.

Example: cascade of component systems

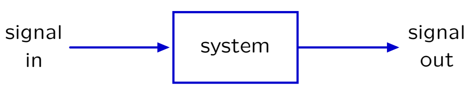


Composite system

Component and composite systems have the same form, and are analyzed with same methods.

The Signals and Systems Abstraction

Our goal is to develop representations for systems that facilitate analysis.

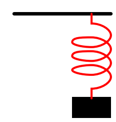


Examples:

- Does the output signal overshoot? If so, how much?
- How long does it take for the output signal to reach its final value?

Continuous and Discrete Time

Inputs and outputs of systems can be functions of continuous time



or discrete time.

We will focus on discrete-time systems.

Difference Equations

Difference equations are an excellent way to represent discrete-time systems.

Example:

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

Difference equations can be applied to any discrete-time system; they are mathematically precise and compact.

Difference Equations

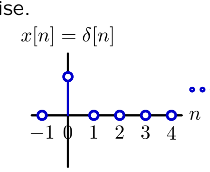
Difference equations are mathematically precise and compact.

Example:

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

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

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



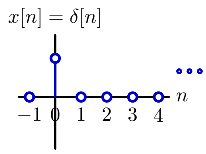
We will use the unit sample as a "primitive" (building-block signal) to construct more complex signals.

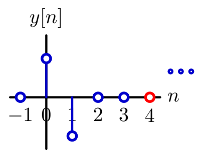
Step-By-Step Solutions

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

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

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





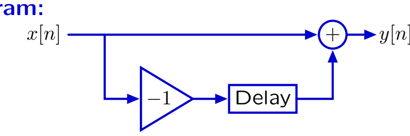
Multiple Representations of Discrete-Time Systems

Block diagrams are useful alternative representations that highlight visual/graphical patterns.

Difference equation:

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

Block diagram:



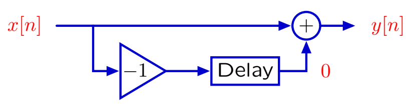
Same input-output behavior, different strengths/weaknesses:

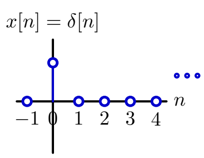
- **difference equations** are mathematically compact
- **block diagrams** illustrate signal flow paths

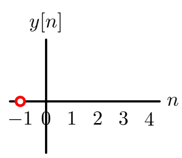
Step-By-Step Solutions

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"







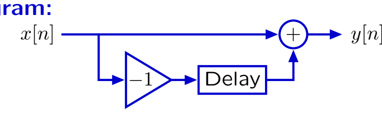
Check Yourself

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

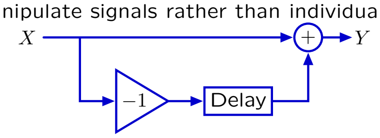
Lumping all of the (possibly infinite) samples into a **single object** – **the signal** – simplifies its manipulation.

This lumping is 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

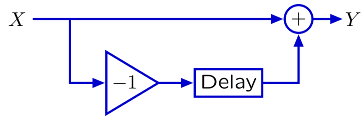
Symbols can now compactly represent diagrams.

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

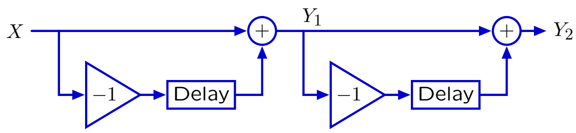
Let $Y = \mathcal{R}X$. Which of the following is/are true:

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:

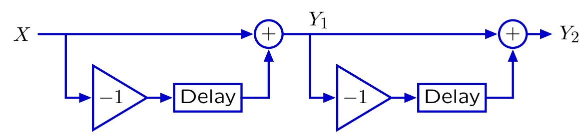
$$Y_1 = (1 - \mathcal{R})\,X$$
$$Y_2 = (1 - \mathcal{R})\,Y_1$$

Substituting for Y_1:

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

Operator Algebra

Operator expressions expand and reduce like 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:

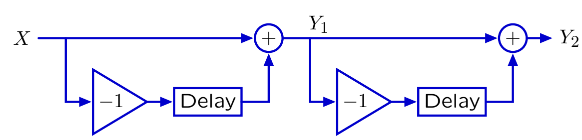
$$\begin{aligned} Y_2 &= (1 - \mathcal{R})\,Y_1 = (1 - \mathcal{R})(1 - \mathcal{R})\,X \\ &= (1 - \mathcal{R})^2 X \\ &= (1 - 2\mathcal{R} + \mathcal{R}^2)\,X \end{aligned}$$

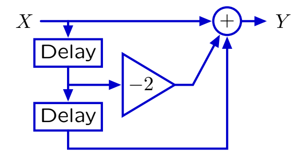
Operator Approach

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):



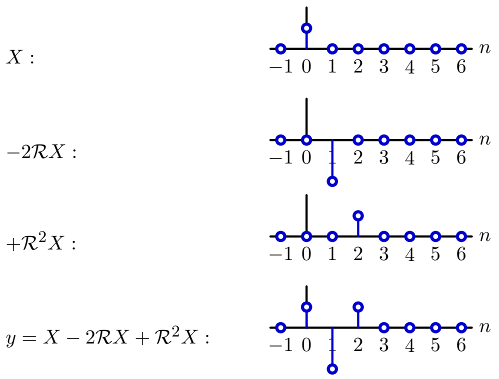


Equivalent operator expression:

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

Operator Algebra

Operator notation prescribes operations on signals, not samples: e.g., start with X, subtract 2 times a right-shifted version of X, and add a double-right-shifted version of X!

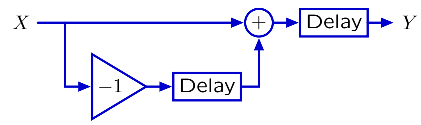


Operator Algebra

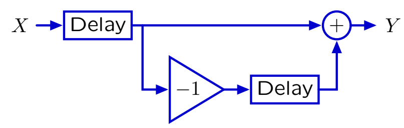
Expressions involving $\mathcal{R}$ obey many familiar laws of algebra, e.g., commutativity.

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

This is easily proved by the definition of $\mathcal{R}$, and it implies that cascaded systems commute (assuming initial rest)



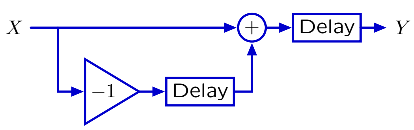
is equivalent to

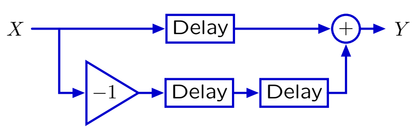


Operator Algebra

Multiplication distributes over addition.

Equivalent systems





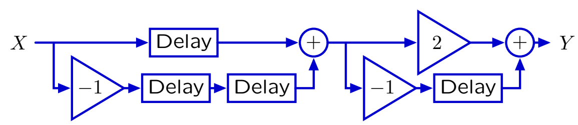
Equivalent operator expression:

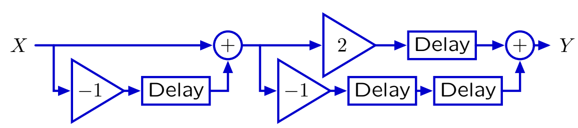
$$\mathcal{R}(1-\mathcal{R}) = \mathcal{R} - \mathcal{R}^2$$

Operator Algebra

The associative property similarly holds for operator expressions.

Equivalent systems



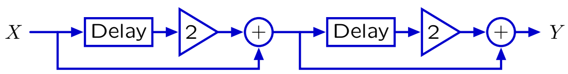


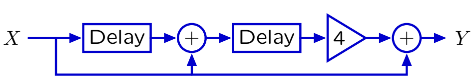
Equivalent operator expression:

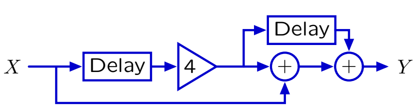
$$\Big((1-\mathcal{R})\mathcal{R}\Big)(2-\mathcal{R}) = (1-\mathcal{R})\Big(\mathcal{R}(2-\mathcal{R})\Big)$$

Check Yourself

How many of the following systems are equivalent?

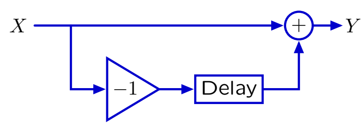






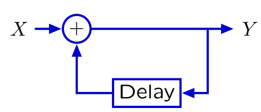
Explicit and Implicit Rules

Recipes versus constraints.



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

Recipe: output signal equals difference between input signal and right-shifted input signal.

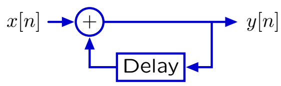


$$Y = \mathcal{R}Y + X$$
$$(1-\mathcal{R})\,Y = X$$

Constraints: find the signal Y such that the difference between Y and $\mathcal{R}Y$ is X. But how?

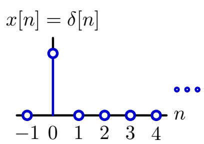
Example: Accumulator

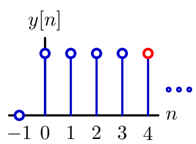
Try step-by-step analysis: it always works. Start "at rest."



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

$$y[0] = x[0] + y[-1] \quad = 1 + 0 = 1$$
$$y[1] = x[1] + y[0] \quad = 0 + 1 = 1$$
$$y[2] = x[2] + y[1] \quad = 0 + 1 = 1$$
$$\cdots$$

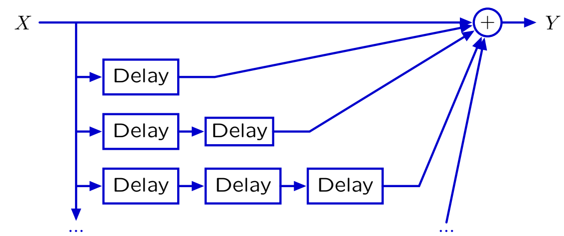




Persistent response to a transient input!

Example: Accumulator

The response of the accumulator system could also be generated by a system with infinitely many paths from input to output, each with one unit of delay more than the previous.



$$Y = (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\, X$$

Example: Accumulator

These systems are equivalent in the sense that if each is initially at rest, they will produce identical outputs from the same input.

$$(1-\mathcal{R})\,Y_1 = X_1 \quad \Leftrightarrow ? \quad Y_2 = (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\, X_2$$

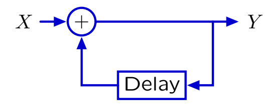
Proof: Assume $X_2 = X_1$:

$$\begin{aligned} Y_2 &= (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\, X_2 \\ &= (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\, X_1 \\ &= (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)\,(1-\mathcal{R})\, Y_1 \\ &= ((1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots) - (\mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots))\, Y_1 \\ &= Y_1 \end{aligned}$$

It follows that $Y_2 = Y_1$.

Example: Accumulator

The system functional for the accumulator is the reciprocal of a polynomial in $\mathcal{R}$.



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

The product $(1-\mathcal{R}) \times (1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)$ equals 1.

Therefore the terms $(1-\mathcal{R})$ and $(1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots)$ are reciprocals.

Thus we can write

$$\frac{Y}{X} = \frac{1}{1-\mathcal{R}} = 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \mathcal{R}^4 + \cdots$$

Example: Accumulator

The reciprocal of $1-\mathcal{R}$ can also be evaluated using synthetic division.

$$\begin{array}{r|l} & 1 \;+\mathcal{R}\; +\mathcal{R}^2\; +\mathcal{R}^3\; +\cdots \\ \hline 1-\mathcal{R} & 1 \\ & \underline{1 \;\; -\mathcal{R}} \\ & \quad \mathcal{R} \\ & \quad \underline{\mathcal{R} \;\; -\mathcal{R}^2} \\ & \qquad \mathcal{R}^2 \\ & \qquad \underline{\mathcal{R}^2 \;\; -\mathcal{R}^3} \\ & \qquad\quad \mathcal{R}^3 \\ & \qquad\quad \underline{\mathcal{R}^3 \;\; -\mathcal{R}^4} \\ & \qquad\qquad \cdots \end{array}$$

Therefore

$$\frac{1}{1-\mathcal{R}} = 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \mathcal{R}^4 + \cdots$$

Check Yourself

A system is described by the following operator expression:

$$\frac{Y}{X} = \frac{1}{1+2\mathcal{R}}.$$

Determine the output of the system when the input is a unit sample.

Linear Difference Equations with Constant Coefficients

Any system composed of adders, gains, and delays can be represented by a difference equation.

$$y[n] + a_1 y[n-1] + a_2 y[n-2] + a_3 y[n-3] + \cdots$$
$$= b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_3 x[n-3] + \cdots$$

Such a system can also be represented by an operator expression.

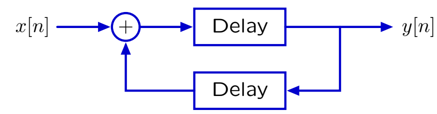
$$(1 + a_1\mathcal{R} + a_2\mathcal{R}^2 + a_3\mathcal{R}^3 + \cdots)\,Y = (b_0 + b_1\mathcal{R} + b_2\mathcal{R}^2 + b_3\mathcal{R}^3 + \cdots)\,X$$

We will see that this correspondence provides insight into behavior.

This correspondence also reduces algebraic tedium.

Check Yourself

Determine the difference equation that relates $x[\cdot]$ and $y[\cdot]$.



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

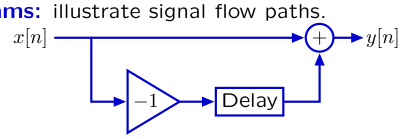
Signals and Systems

Multiple representations of discrete-time systems.

Difference equations: mathematically compact.

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

Block diagrams: illustrate signal flow paths.



Operator representations: analyze systems as polynomials.

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

Labs: representing **signals** in python
controlling robots and analyzing their behaviors.

MIT OpenCourseWare
http://ocw.mit.edu

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.