6.003: Signals and Systems

Discrete-Time Systems

September 13, 2011

Homework

Doing the homework is essential to understanding the content.

Weekly Homework Assigments

- tutor (exam-type) problems:
  answers are automatically checked to provide quick feedback
- engineering design (real-world) problems:
  graded by a human

Learning doesn't end when you have submitted your work!

- solutions will be posted on Wednesdays at 5pm
- read solutions to find errors and to see alternative approaches
- mark the errors in your previously submitted work
- submit the markup by Friday at 5pm
- **identify ALL errors and get back half of the points you lost!**

做作业对理解内容是必不可少的。
每周家庭作业导师(考试类型)问题:自动检查答案以提供快速反馈工程设计(现实世界)问题:由人工评分学习不会在您提交作业后结束!
解决方案将于周三下午5点发布阅读解决方案，找出错误并查看替代方法，标记之前提交的作业中的错误，并在周五下午5点之前提交标记，找出所有错误并获得一半的分数!

Discrete-Time Systems

We start with discrete-time (DT) systems because they

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

我们从离散时间(DT)系统开始,
因为它们在概念上比连续时间系统更简单,
说明了与连续时间系统同样重要的思维模式,
因为连续时间系统越来越重要(数字电子和计算)。

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.'

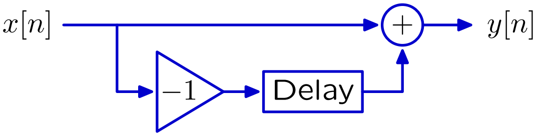
系统可以用不同的方式表示，
以便更容易地处理不同类型的问题。
口头描述:
为了减少存储一串几乎相等的大数所需的比特数，
记录第一个数字，然后记录连续的差异。
差分方程:y[n] = x[n] - x[n -1]

Difference equation:

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

Block diagram:

框图:



我们将利用每种表示的特定优势。

We will exploit particular strengths of each of these representations.

Difference Equations

Difference equations are mathematically precise and compact.

差分方程在数学上是精确而紧凑的。

Example:

例子:

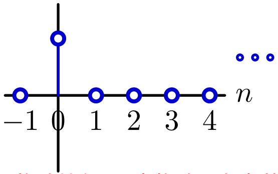
$$y[n] = x[n] - x[n-1]$$

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

令x[n]等于 " 单位样本 " 信号 δ [n],

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

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



我们将使用单元样本作为 " 原始 " (构建块信号)来构建更复杂的信号。

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. $y[119] = 0$

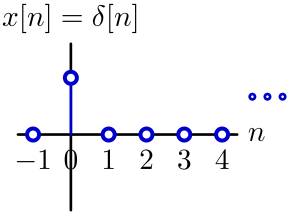
Step-By-Step Solutions

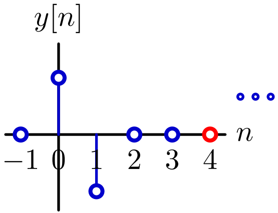
Difference equations are convenient for step-by-step analysis.

差分方程便于分步分析。

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

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





Check Yourself

Solve

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

given

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

How many of the following are true? 4

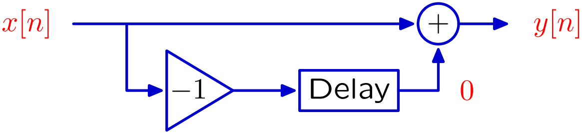
1. $y[2] > y[1]$ $\surd$
2. $y[3] > y[2]$ X
3. $y[2] = 0$ $\surd$
4. $y[n] - y[n-1] = x[n] - 2x[n-1] + x[n-2]$ $\surd$
5. $y[119] = 0$ $\surd$

Step-By-Step Solutions

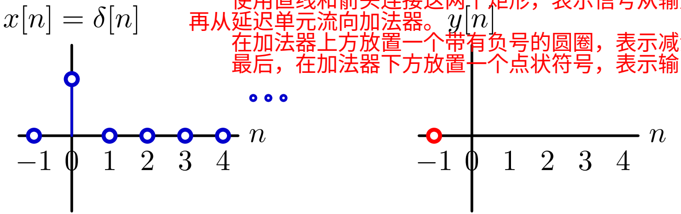
Block diagrams are also useful for step-by-step analysis.

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

框图对于逐步分析也很有用。
用框图表示y[n] = [n]- x[n -1]:



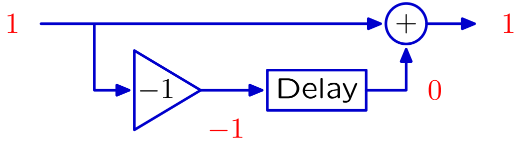
在画布上放置一个矩形，并在其内部写入“Delay”字样，表示延迟单元。
将另一个矩形放在第一个矩形旁边，并在其内部写入“+”符号，表示加法器。
使用直线和箭头连接这两个矩形，表示信号从输入端流向延迟单元，再从延迟单元流向加法器。
在加法器上方放置一个带有负号的圆圈，表示减法运算。
最后，在加法器下方放置一个点状符号，表示输出信号 y[n]。

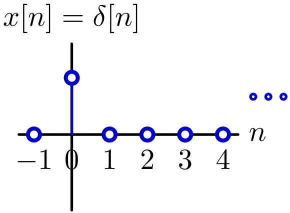


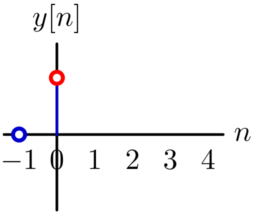
Step-By-Step Solutions

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"



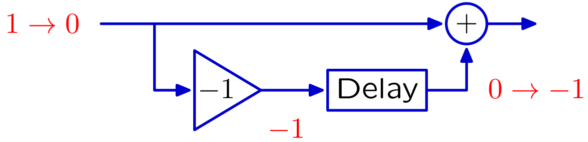


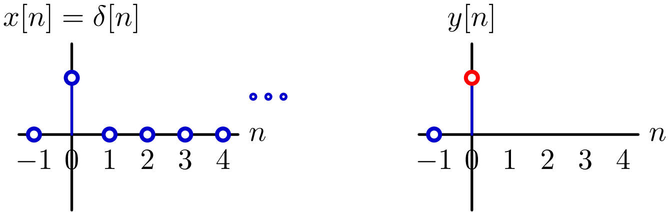


Step-By-Step Solutions

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"

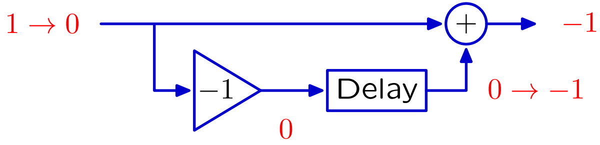


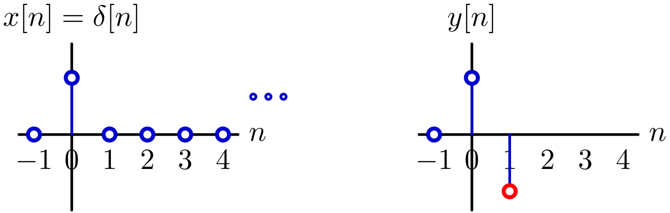


Step-By-Step Solutions

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"

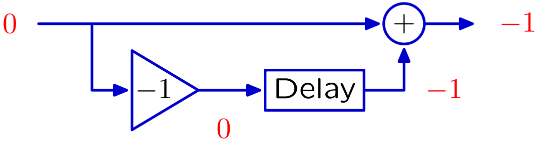


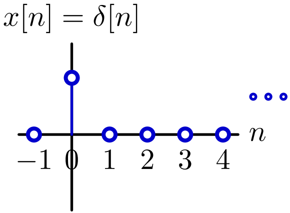


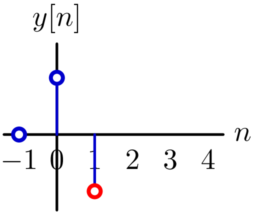
Step-By-Step Solutions

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"



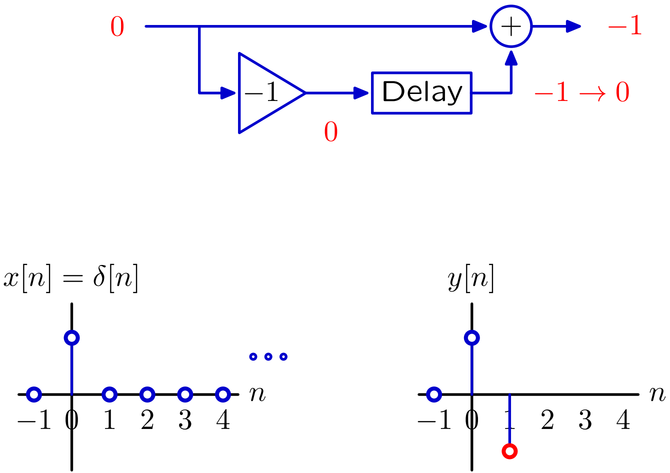




Step-By-Step Solutions

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

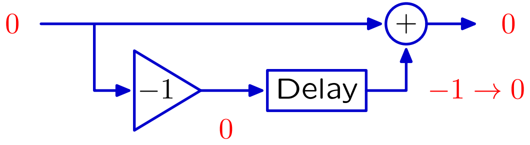
Represent $y[n] = x[n] - x[n-1]$ with a block diagram: start "at rest"

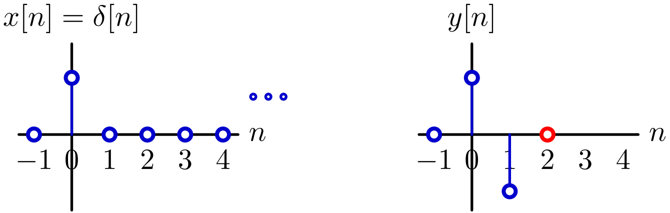


Step-By-Step Solutions

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"

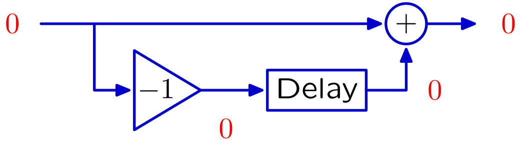


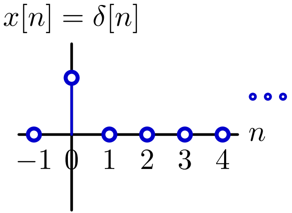


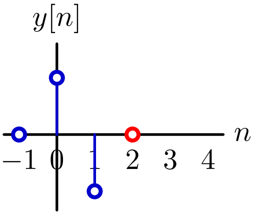
Step-By-Step Solutions

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"



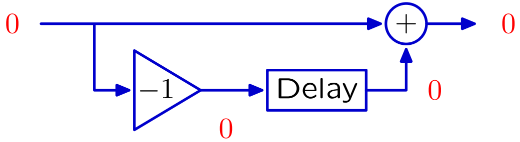


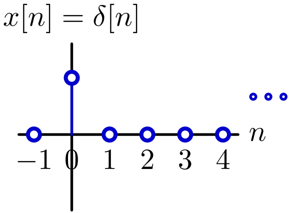


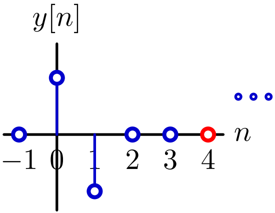
Step-By-Step Solutions

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"







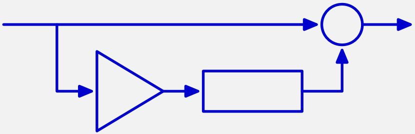
Check Yourself

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?

Check Yourself

In what ways are difference equations different from block diagrams?

Difference equation:

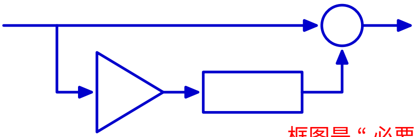
$$y[n] = x[n] - x[n-1]$$

Difference equations are "declarative."

They tell you rules that the system obeys.

差分方程是“ 陈述性的 ”。
他们告诉你系统遵守的规则。

Block diagram:



Block diagrams are "imperative."

They tell you what to do.

框图是“ 必要的 ”。
“ 他们告诉你该怎么做。
差分方程(例如，输入是什么?
输出是什么?)
框图比相应的框图包含更多的信息

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

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

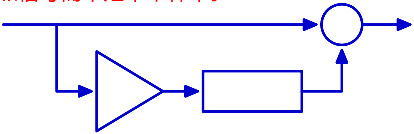
将所有(可能是无限的)样本集中到一个对象中，信号简化了其操作。

这种集总是一种抽象，
类似于将三维空间中的坐标表示为点，
将数字列表表示为线性代数中的向量，
从而在Python中创建对象

From Samples to Signals

Operators manipulate signals rather than individual samples.

操作员操纵信号而不是单个样本。



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

节点代表整个信号(例如，X和Y)

盒子对这些信号进行操作:
Delay =将整个信号向右移动1个时间步
Add =将两个信号相加
-1:乘以-1。

Signals are the primitives.

Operators are the means of combination.

信号是基元。
操作符是组合的手段。

Operator Notation

符号现在可以简洁地表示图表。
设R表示右移算子:Y=R{X} = rx,

Symbols can now compactly represent diagrams.

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

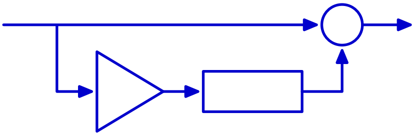
$$Y = \mathcal{R}\{X\} \equiv \mathcal{R}X$$

其中X表示整个输入信号([n]表示所有n),
Y表示整个输出信号(Y [n]表示所有n),

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

表示差分机



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

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

和R的表达式是等价的

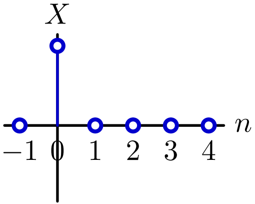
Operator Notation: Check Yourself

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

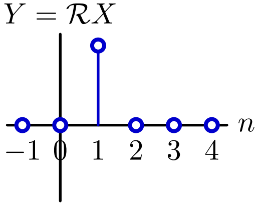
1. $y[n] = x[n]$ for all n
2. $y[n+1] = x[n]$ for all n
3. $y[n] = x[n+1]$ for all n
4. $y[n-1] = x[n]$ for all n
5. none of the above

Check Yourself

Consider a simple signal:



Then



Clearly $y[1] = x[0]$. Equivalently, if $n = 0$, then $y[n+1] = x[n]$.

The same sort of argument works for all other n.

Operator Notation: Check Yourself

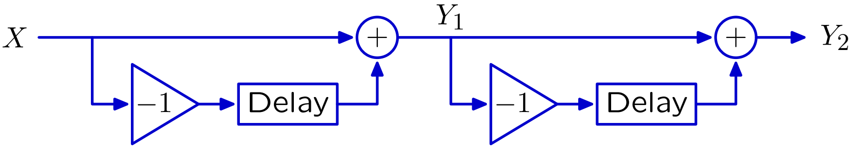
Let $Y = \mathcal{R}X$. Which of the following is/are true:

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 $\rightarrow$ multiply operator expressions.



级联系统的操作符表示
系统操作具有简单的操作符表示。
级联系统→乘法运算符表达式。

Using operator notation:

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

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

Substituting for Y_1:

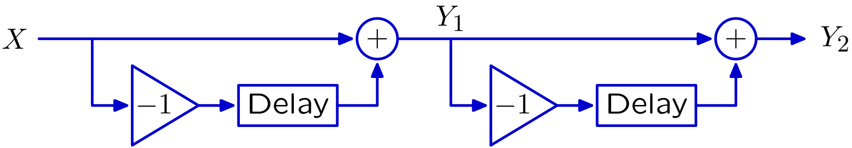
$$Y_2 = (1 - \mathcal{R})(1 - \mathcal{R})\, X$$

Operator Algebra

运算符

代数运算符表达式可以作为多项式来操作。

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.

运用你现有的多项式专业知识来理解方框图，从而理解系统。

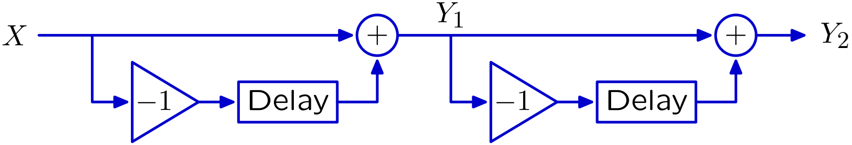
Operator Algebra

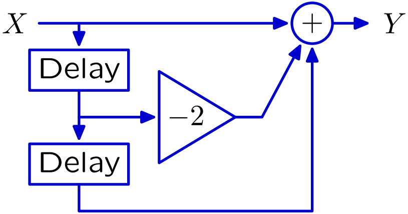
Operator notation facilitates seeing relations among systems.

运算符符号便于查看系统之间的关系。

“等效”框图(假设两者最初处于静止状态):

“Equivalent” block diagrams (assuming both initially at rest):





Equivalent operator expressions:

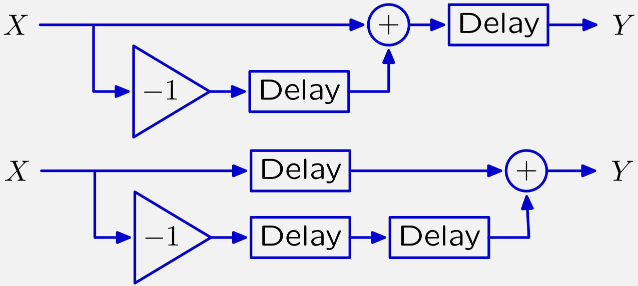
$$(1-\mathcal{R})(1-\mathcal{R}) = 1 - 2\mathcal{R} + \mathcal{R}^2$$

The operator equivalence is much easier to see.

Check Yourself

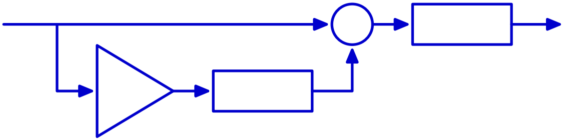
这些“等效”系统的算子表达式(如果开始时“静止”)服从什么数学性质?

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

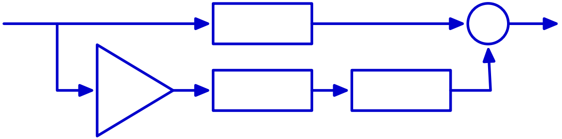


1. commutate
2. associative
3. distributive
4. transitive
5. none of the above

Check Yourself



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

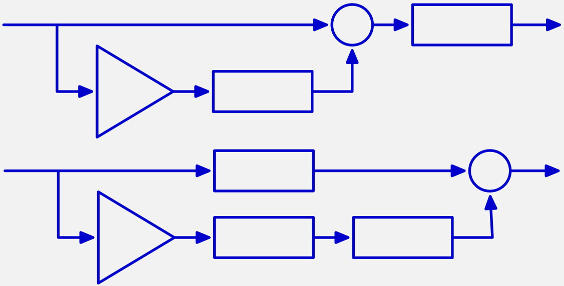


$$Y = (\mathcal{R} - \mathcal{R}^2)X$$

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

Check Yourself

Operator expressions for these "equivalent" systems (if started "at rest") obey what mathematical property? 3

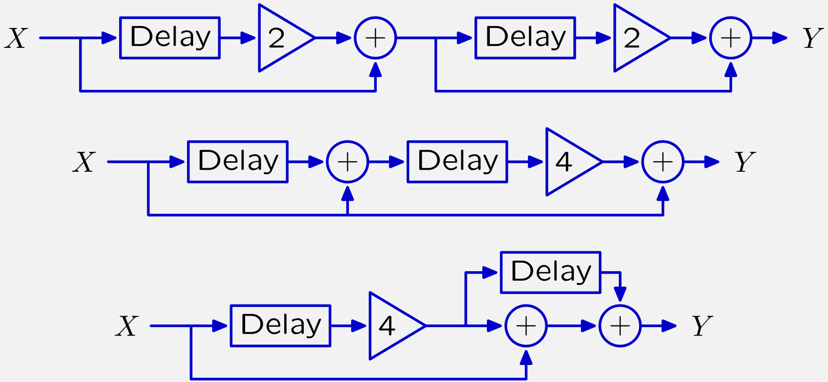


1. commutate
2. associative
3. distributive
4. transitive
5. none of the above

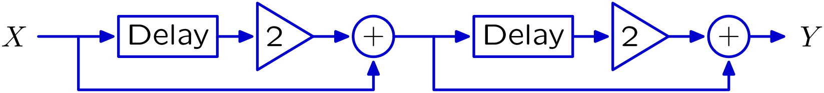
Check Yourself

How many of the following systems are equivalent to

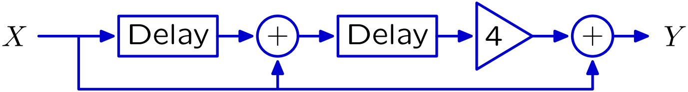
$$Y = (4\mathcal{R}^2 + 4\mathcal{R} + 1)\,X \quad ?$$



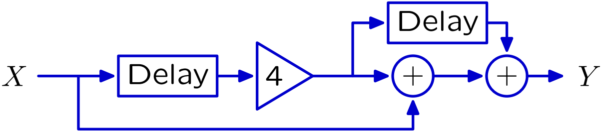
Check Yourself



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



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

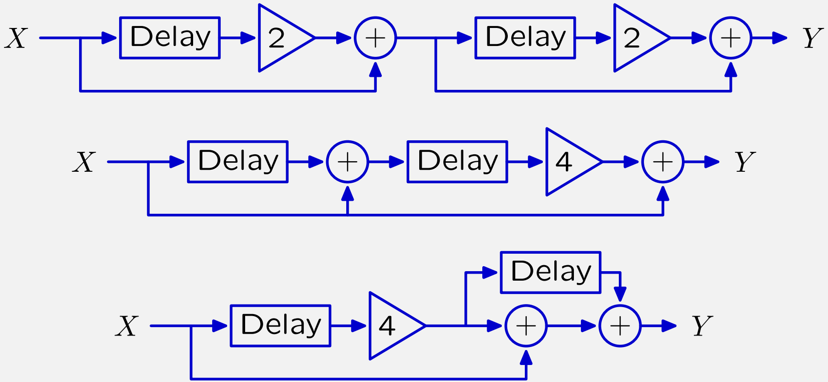


$$Y = (4\mathcal{R}^2+4\mathcal{R}+1)\,X$$

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

Check Yourself

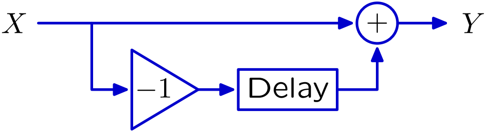
How many of the following systems are equivalent to
$$Y = \left(4\mathcal{R}^2 + 4\mathcal{R} + 1\right) X \quad ? \qquad 3$$



Operator Algebra: Explicit and Implicit Rules

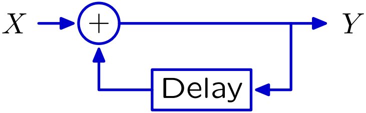
Recipes versus constraints.

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.



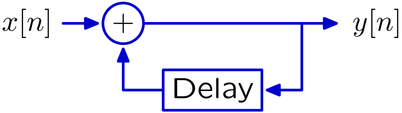
$$Y = \mathcal{R}Y + X$$

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

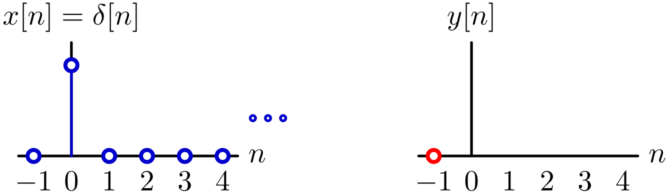
But how does one solve such a constraint?

Example: Accumulator

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

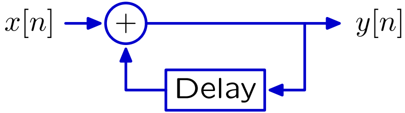


Find $y[n]$ given $x[n] = \delta[n]$: $\quad y[n] = x[n] + y[n-1]$



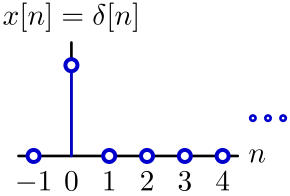
Example: Accumulator

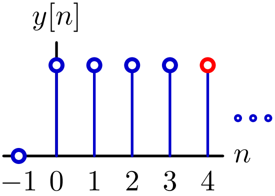
Try step-by-step analysis: it always works. Start "at rest."



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

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

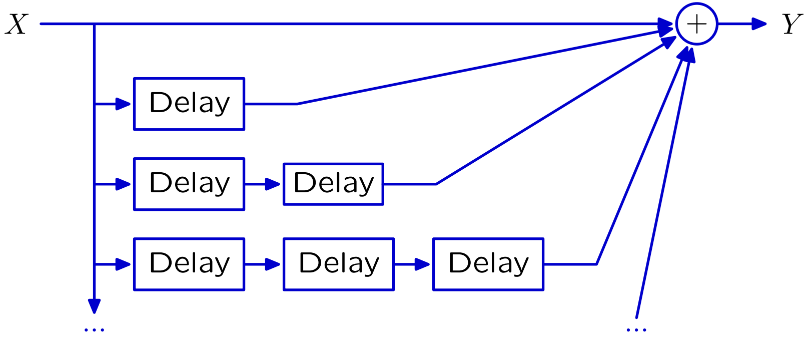




Persistent response to a transient input!

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.

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

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 $Y_2 = Y_1$.

It also follows that $(1-\mathcal{R})$ and $(1+\mathcal{R}+\mathcal{R}^2+\mathcal{R}^3+\cdots)$ are reciprocals.

Example: Accumulator

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

$$
\begin{array}{r|llllll}
 & 1 & +\mathcal{R} & +\mathcal{R}^2 & +\mathcal{R}^3 & +\cdots & \\
\cline{2-6}
1-\mathcal{R} & 1 & & & & & \\
 & \underline{1} & \underline{-\mathcal{R}} & & & & \\
 & & \mathcal{R} & & & & \\
 & & \underline{\mathcal{R}} & \underline{-\mathcal{R}^2} & & & \\
 & & & \mathcal{R}^2 & & & \\
 & & & \underline{\mathcal{R}^2} & \underline{-\mathcal{R}^3} & & \\
 & & & & \mathcal{R}^3 & & \\
 & & & & \underline{\mathcal{R}^3} & \underline{-\mathcal{R}^4} & \\
 & & & & & \cdots &
\end{array}
$$

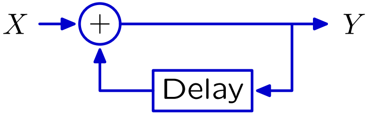
Therefore

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

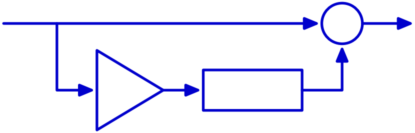
Feedback

Systems with signals that depend on previous values of the same signal are said to have **feedback**.

Example: The accumulator system has feedback.



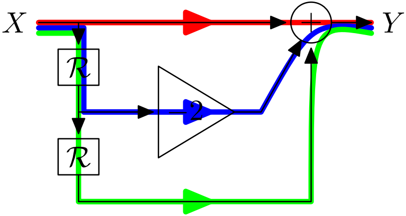
By contrast, the difference machine does not have feedback.



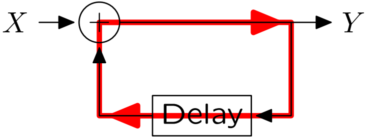
Cyclic Signal Paths, Feedback, and Modes

Block diagrams help visualize feedback.

Feedback occurs when there is a cyclic signal flow path.



acyclic



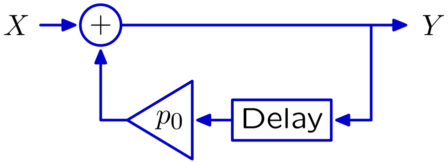
cyclic

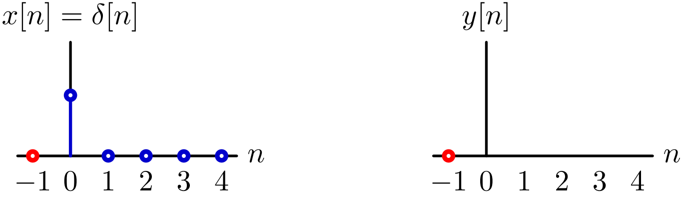
Acyclic: all paths through system go from input to output with no cycles.

Cyclic: at least one cycle.

Feedback, Cyclic Signal Paths, and Modes

The effect of feedback can be visualized by tracing each cycle through the cyclic signal paths.

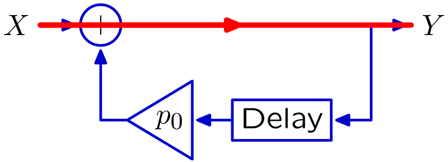


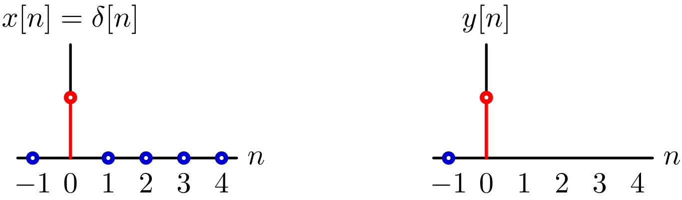


Each cycle creates another sample in the output.

Feedback, Cyclic Signal Paths, and Modes

The effect of feedback can be visualized by tracing each cycle through the cyclic signal paths.

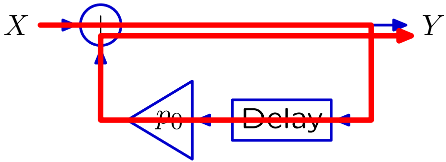


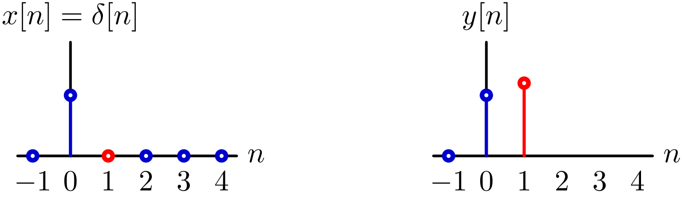


Each cycle creates another sample in the output.

Feedback, Cyclic Signal Paths, and Modes

The effect of feedback can be visualized by tracing each cycle through the cyclic signal paths.

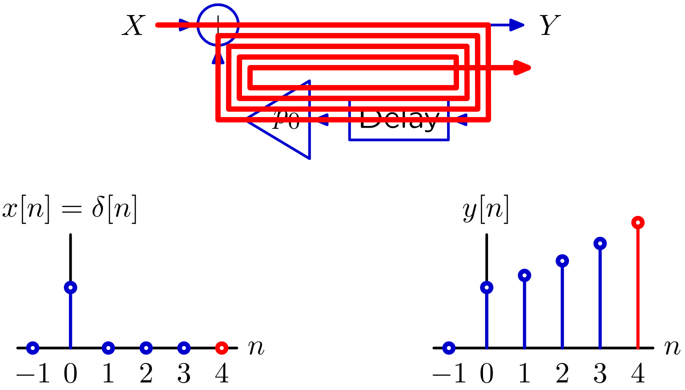




Each cycle creates another sample in the output.

Feedback, Cyclic Signal Paths, and Modes

The effect of feedback can be visualized by tracing each cycle through the cyclic signal paths.

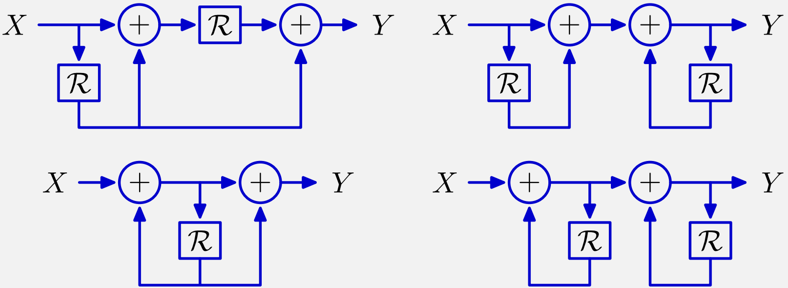


Each cycle creates another sample in the output.

The response will persist even though the input is transient.

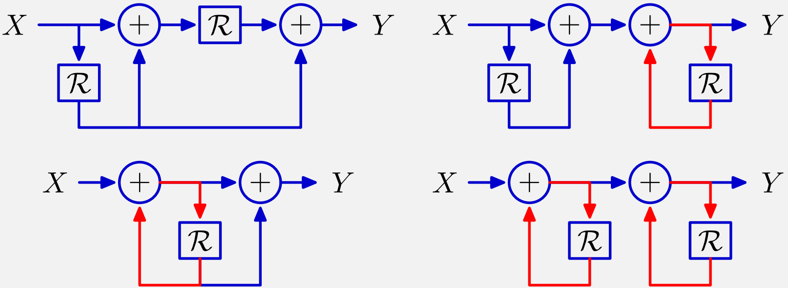
Check Yourself

How many of the following systems have cyclic signal paths?



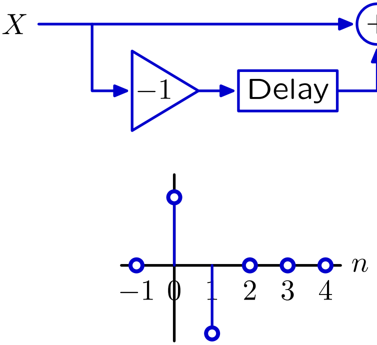
Check Yourself

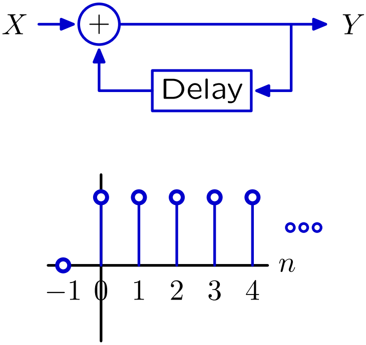
How many of the following systems have cyclic signal paths? 3



Finite and Infinite Impulse Responses

The impulse response of an acyclic system has finite duration, while that of a cyclic system can have infinite duration.





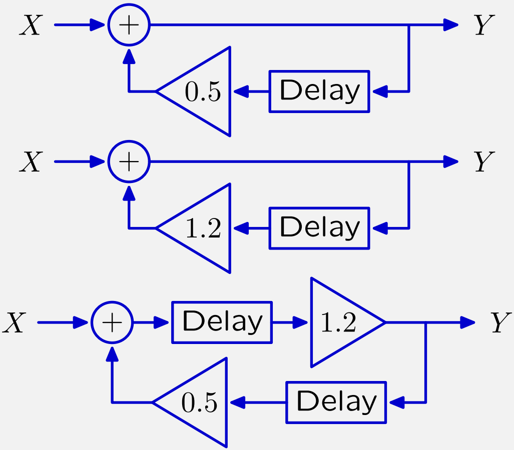
Analysis of Cyclic Systems: Geometric Growth

If traversing the cycle decreases or increases the magnitude of the signal, then the fundamental mode will decay or grow, respectively.

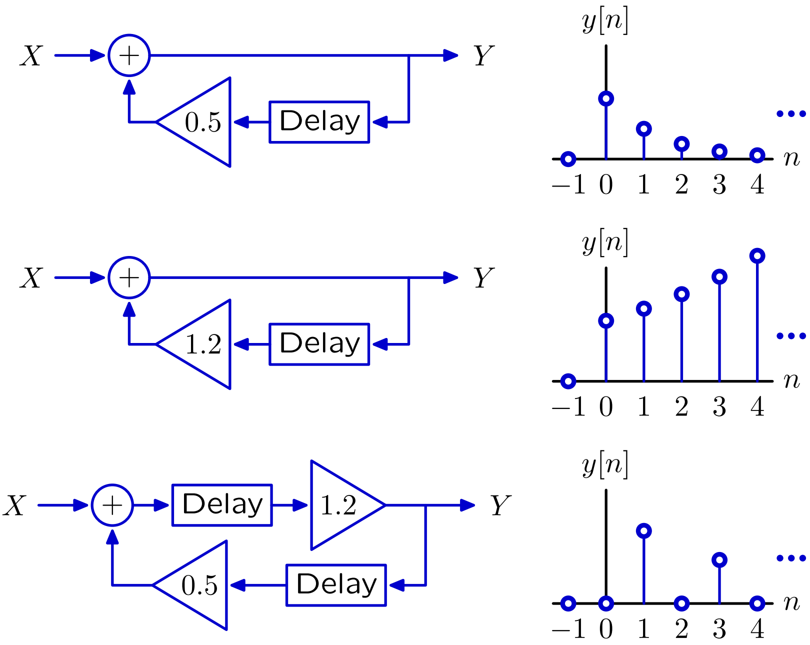
If the response decays toward zero, then we say that it **converges**.
Otherwise, we it **diverges**.

Check Yourself

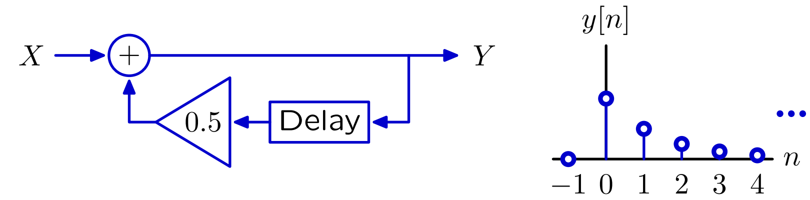
How many of these systems have divergent unit-sample responses?



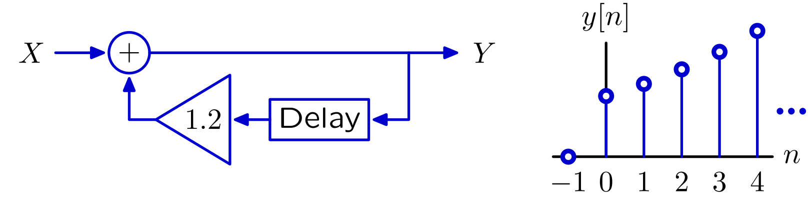
Check Yourself



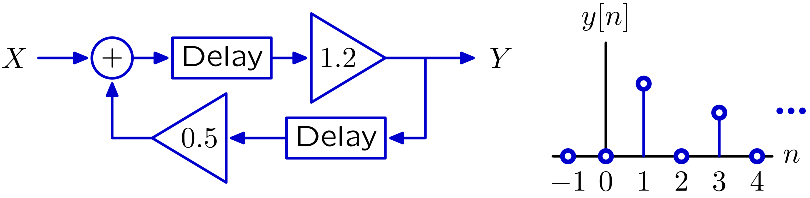
Check Yourself



×



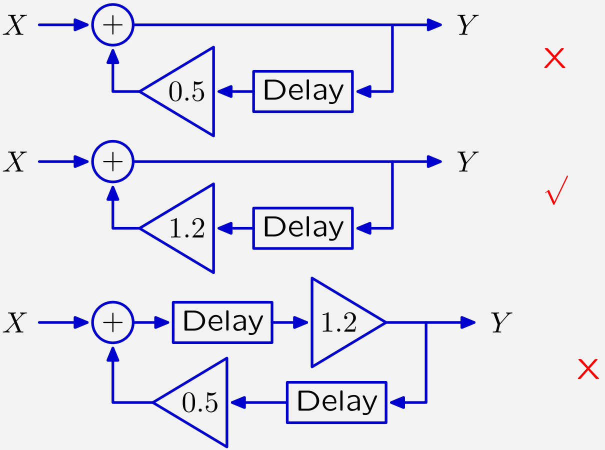
√



×

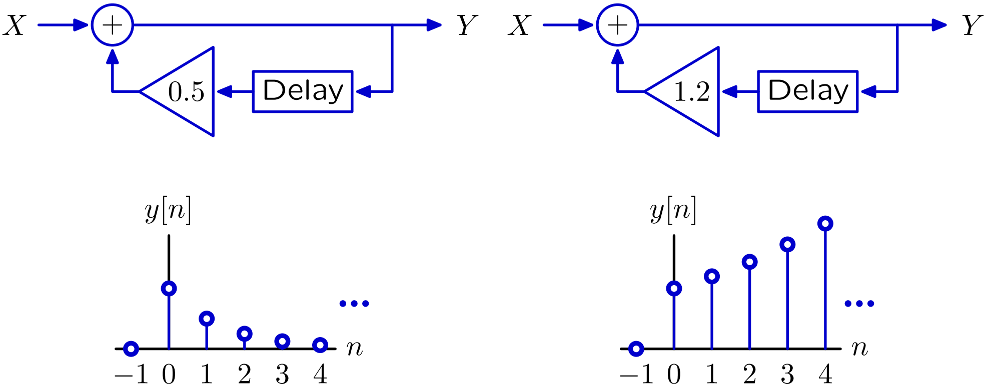
Check Yourself

How many of these systems have divergent unit-sample responses? 1



Cyclic Systems: Geometric Growth

If traversing the cycle decreases or increases the magnitude of the signal, then the fundamental mode will decay or grow, respectively.



These are geometric sequences: $y[n] = (0.5)^n$ and $(1.2)^n$ for $n \ge 0$.

These geometric sequences are called **fundamental modes**.

Multiple Representations of Discrete-Time Systems

Now you know four representations of discrete-time systems.

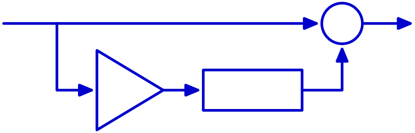
Verbal descriptions: preserve the rationale.

> "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 equations: mathematically compact.

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

Block diagrams: illustrate signal flow paths.



Operator representations: analyze systems as polynomials.

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

6.003 Signals and Systems
Fall 2011