Hardware Fundamentals [CESE4005]

Signals and Systems

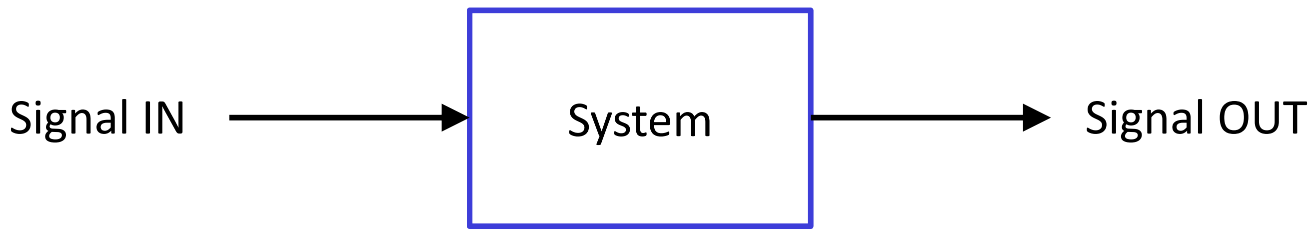
Lecturers: Anteneh Gebrgiorgious & Georgi Gaydadjiev
Teaching Assistant: Antonio Costa Bernardes

September 2024

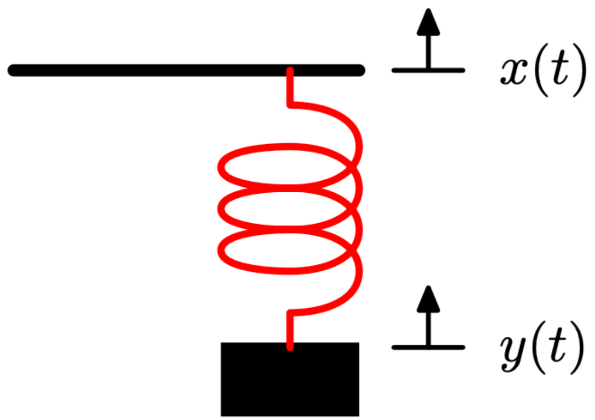
TU Delft

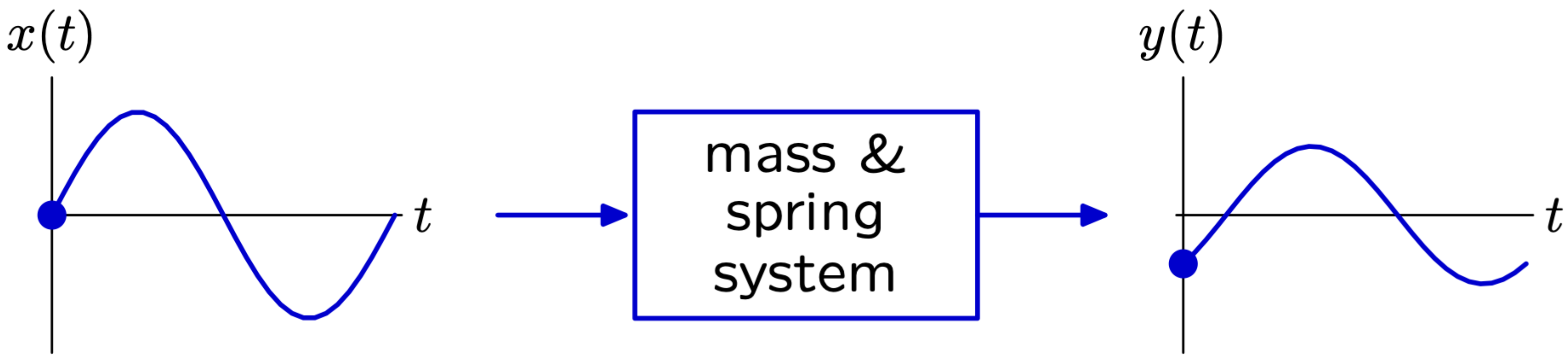
The Signals and Systems Abstraction

Describe a system (physical, mathematical, or computational) by the way it transforms an input signal into an output signal.

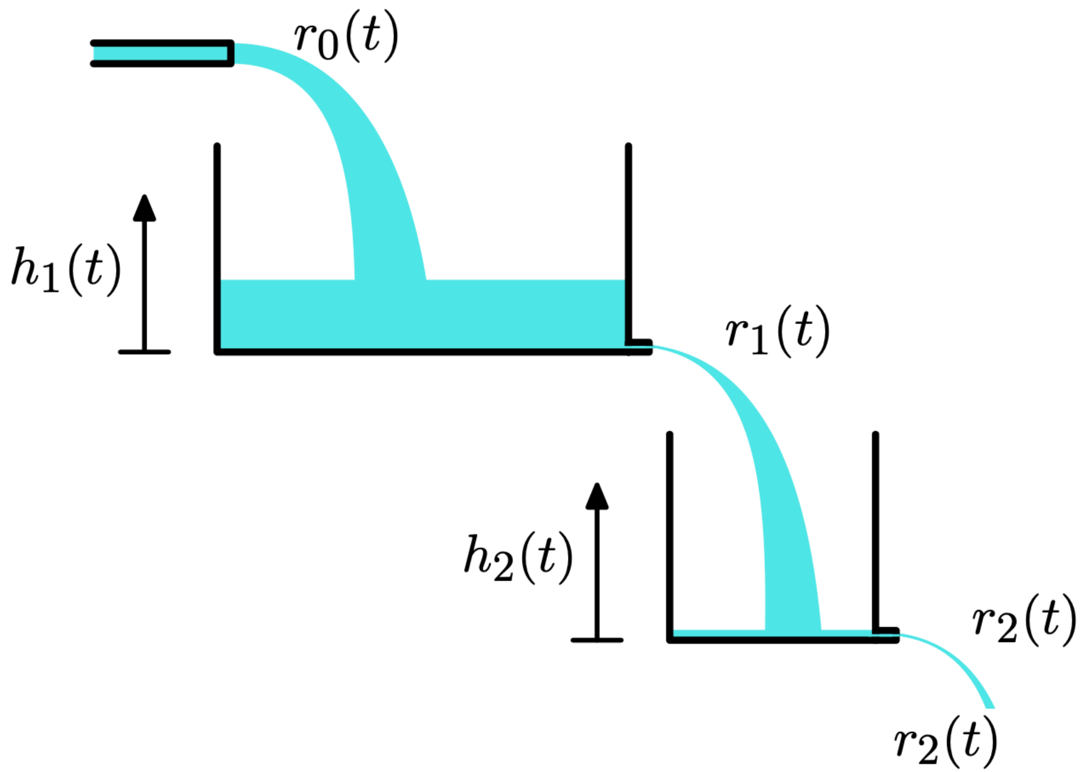


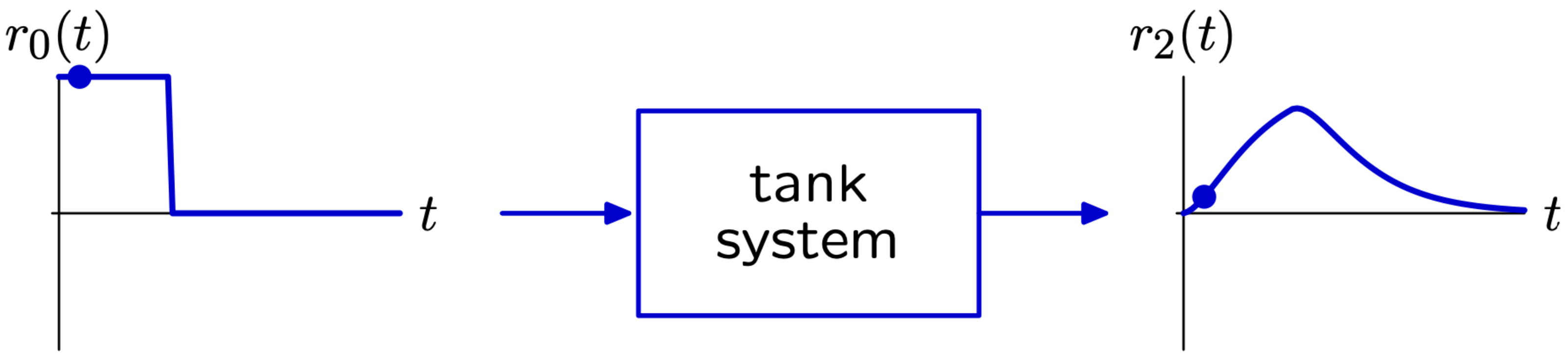
Example: Mass and Spring



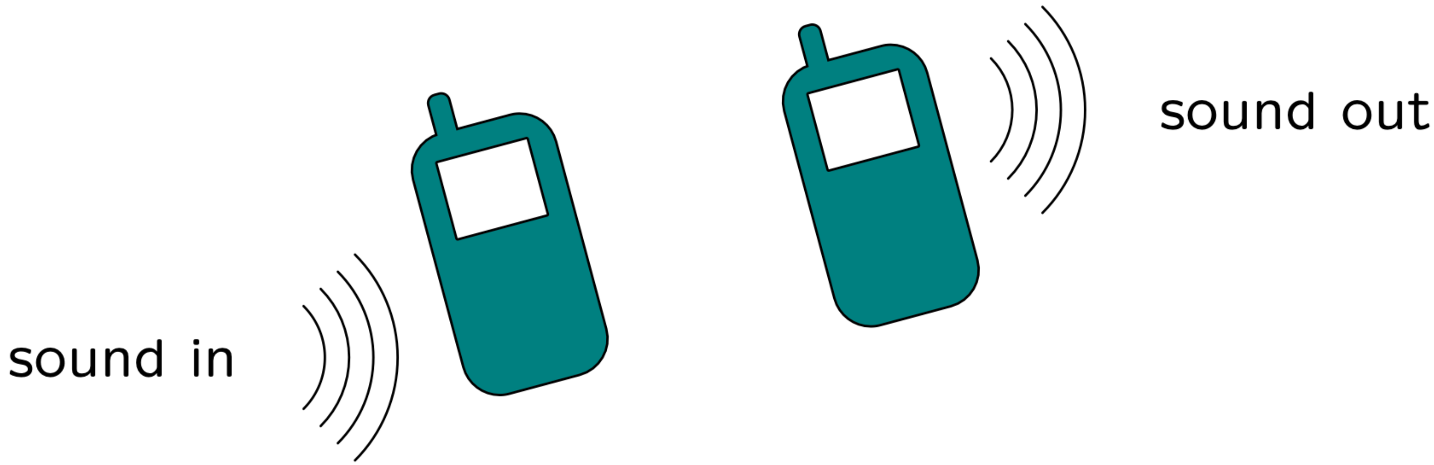


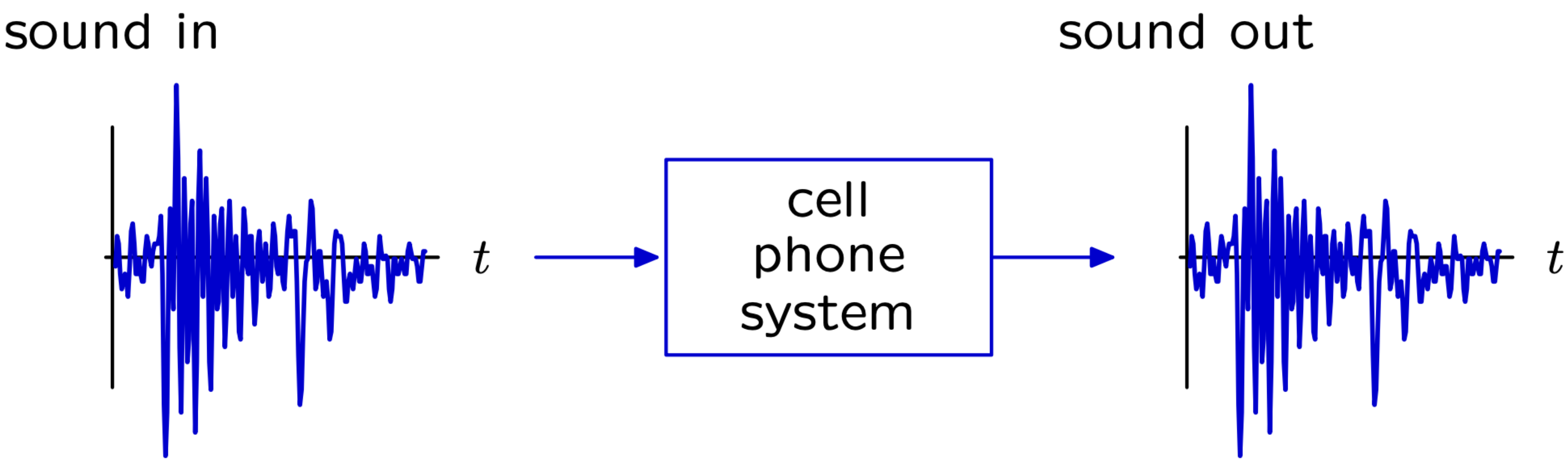
Example: Tanks





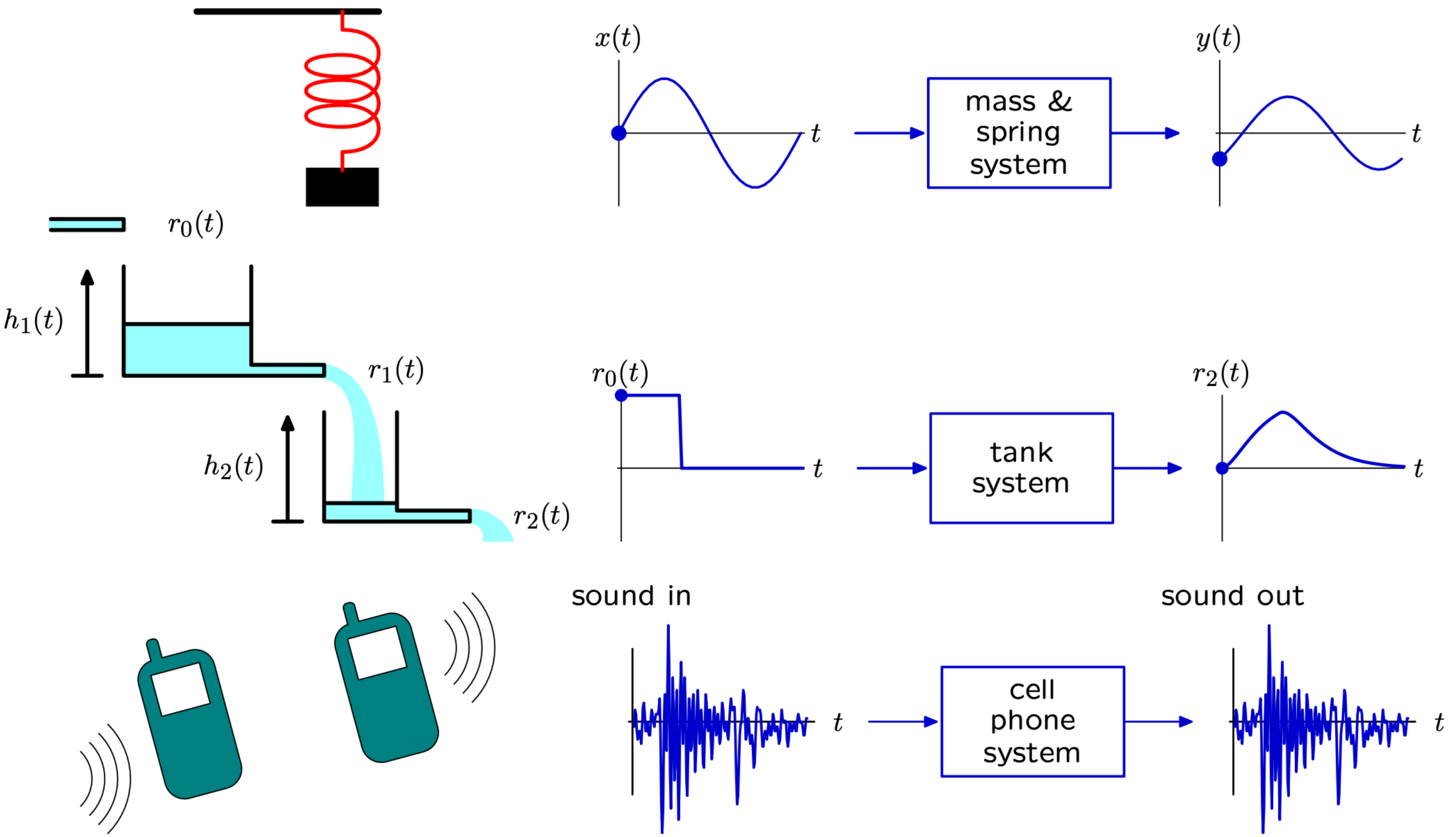
Example: Tanks





Signals and Systems: Widely Applicable

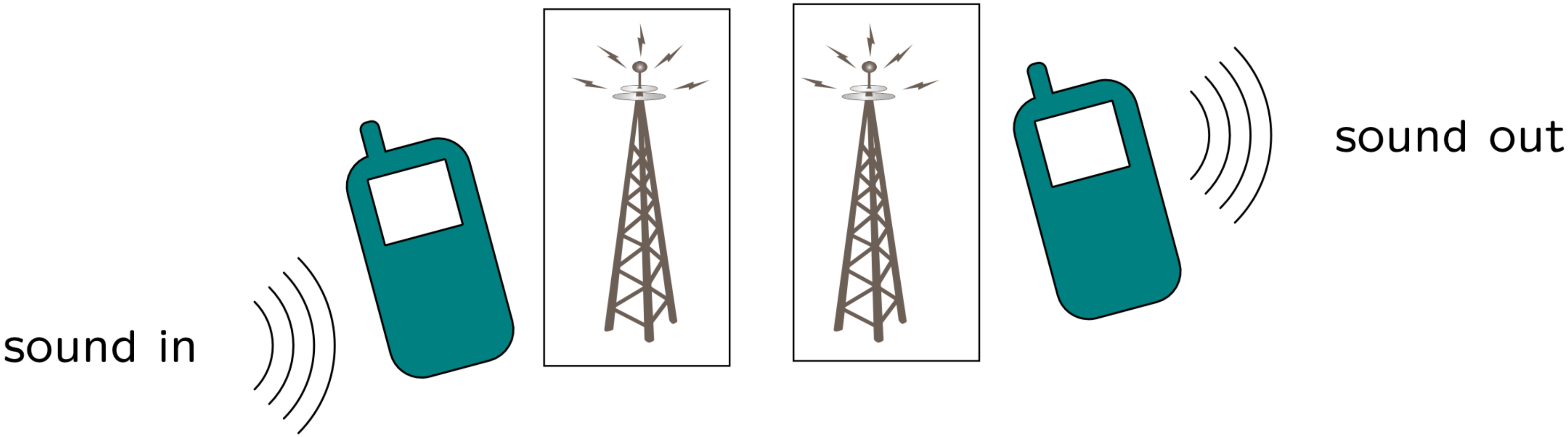
The Signals and Systems approach has broad application:
electrical, mechanical, optical, acoustic, biological, financial, ...

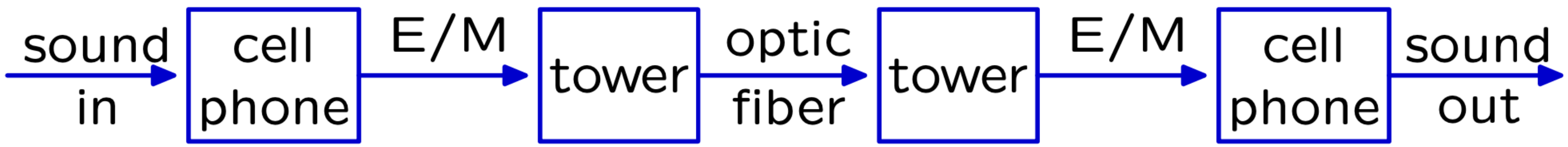


TU Delft

Signals and Systems: Modular

The representation does not depend upon the physical substrate.

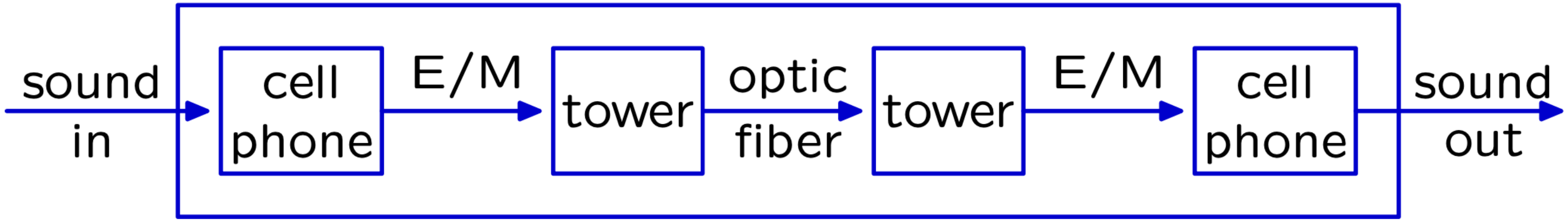




focuses on the flow of information, abstracts away everything else.

Signals and Systems: Hierarchical

Representations of component systems are easily combined.
Example: cascade of component systems



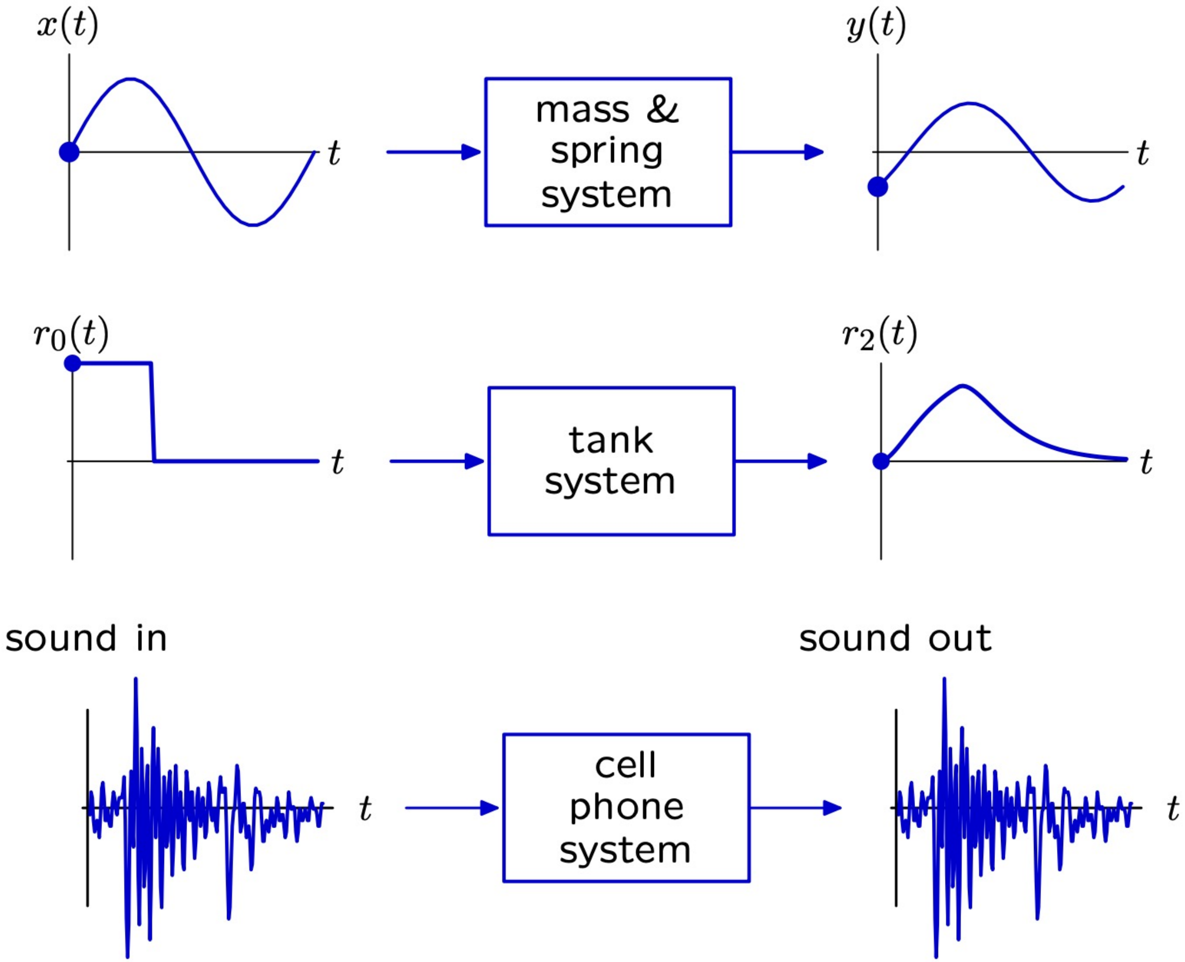
Compositional

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

Signals and Systems

Signals are mathematical functions.

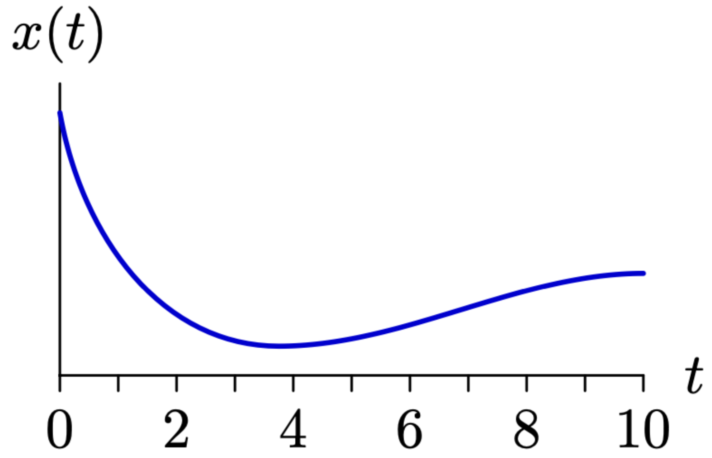
- independent variable = time
- dependent variable = voltage, flow rate, sound pressure

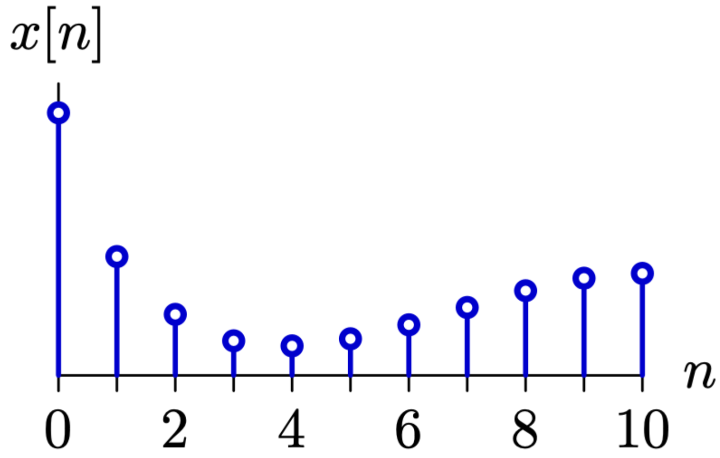


TU Delft

Signals and Systems

continuous "time" (CT) and discrete "time" (DT)





Signals from physical systems often functions of continuous time.

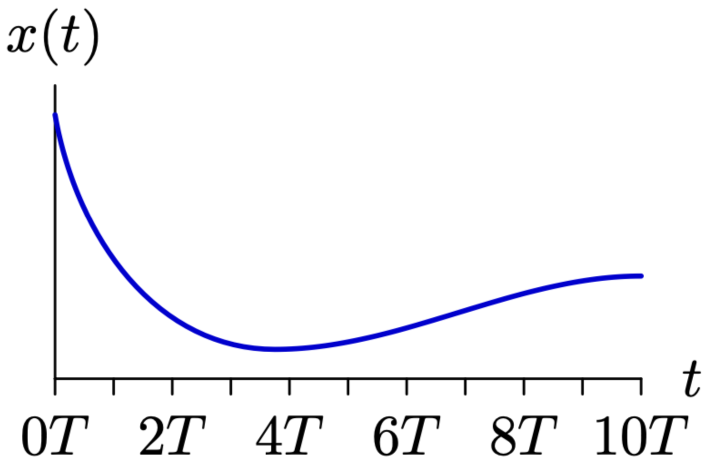
- mass and spring
- leaky tank

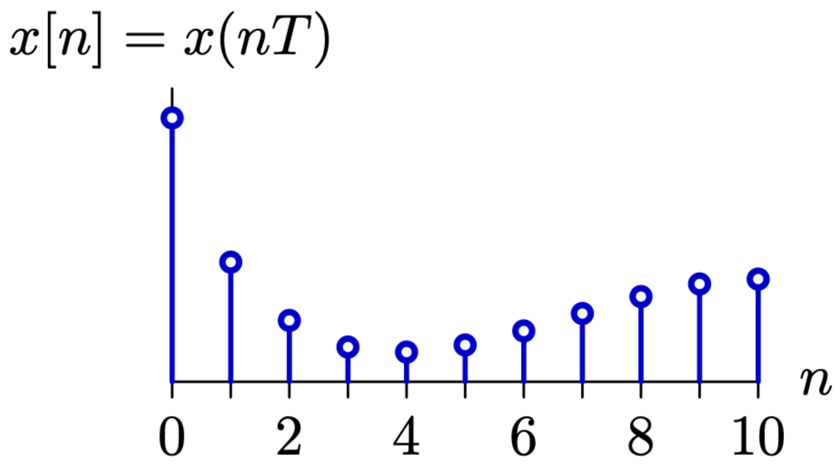
Signals from computation systems often functions of discrete time.

- state machines: given the current input and current state, what is the next output and next state.

Signals and Systems

Sampling: converting CT to DT (Analog to Digital conversion A/D)





T = sampling interval

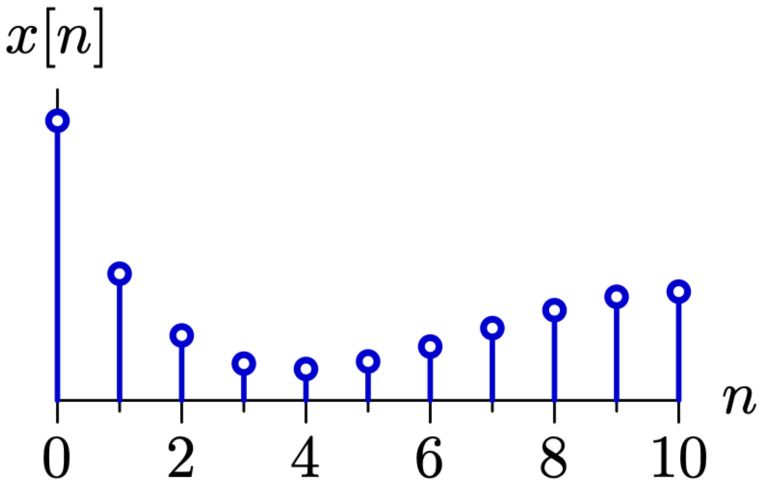
Important for computational manipulation of physical data.

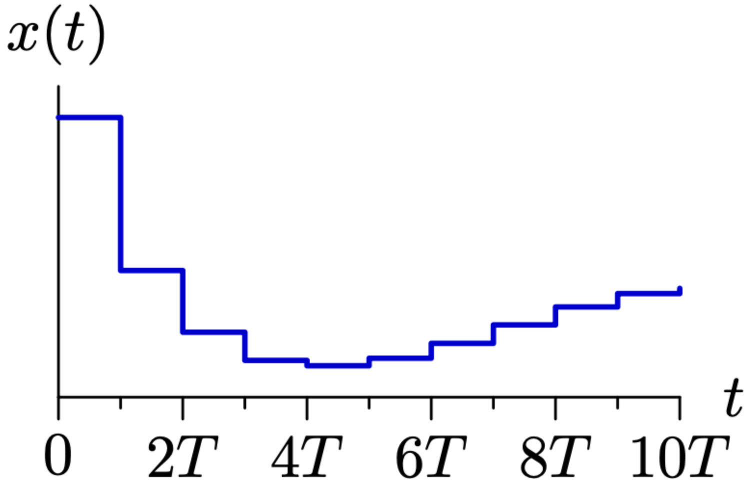
- digital representations of audio signals (e.g., MP3)
- digital representations of images (e.g., JPEG)

Signals and Systems

Reconstruction: converting DT signals to CT

zero-order hold





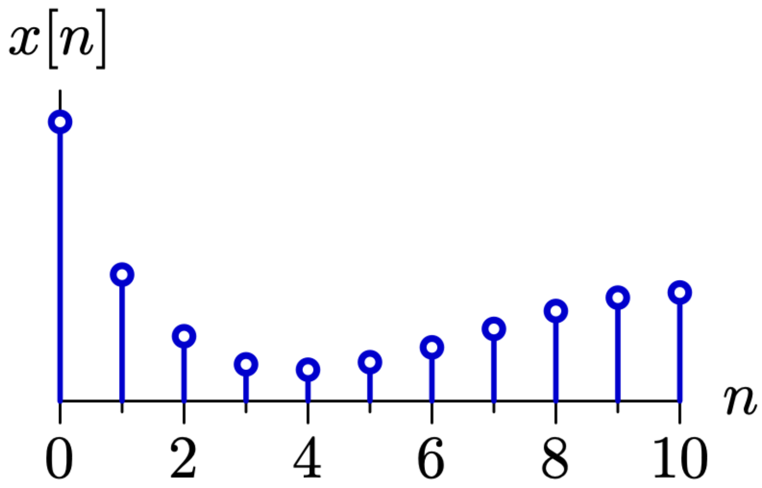
T = sampling interval

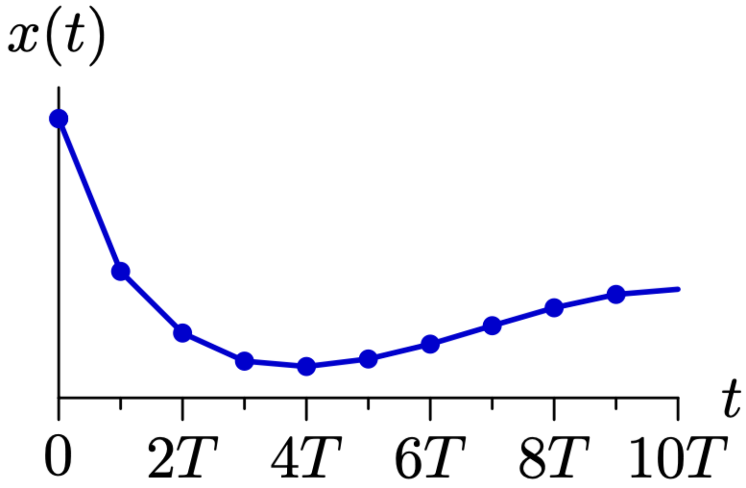
commonly used in audio output devices such as CD players

Signals and Systems

Reconstruction: converting DT signals to CT

piecewise linear

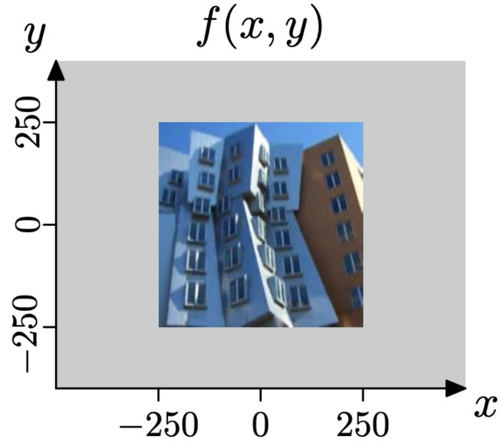




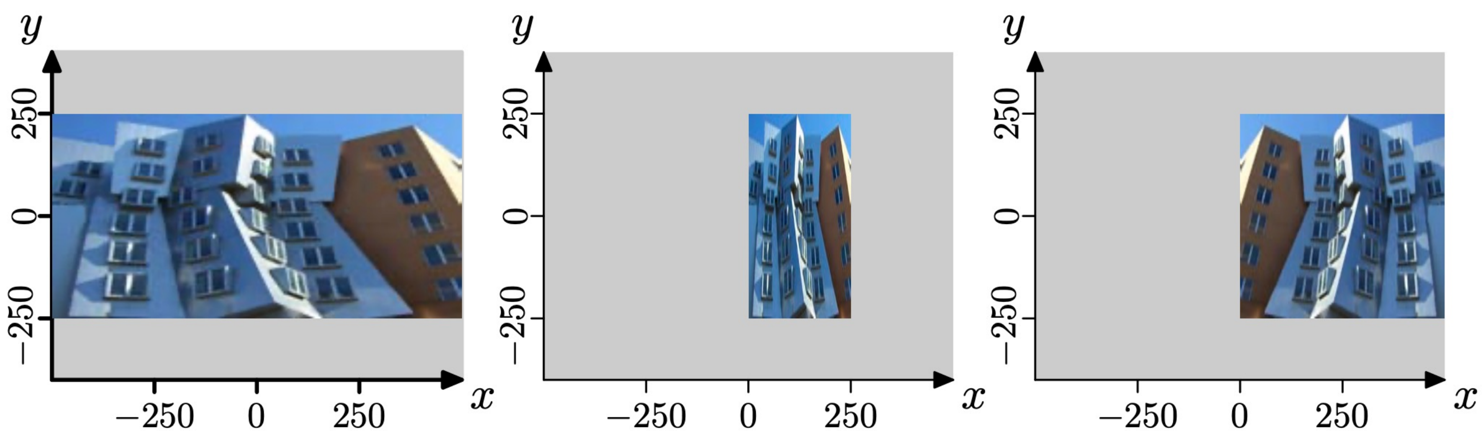
T = sampling interval

commonly used in rendering images

Check yourself (2D signal)



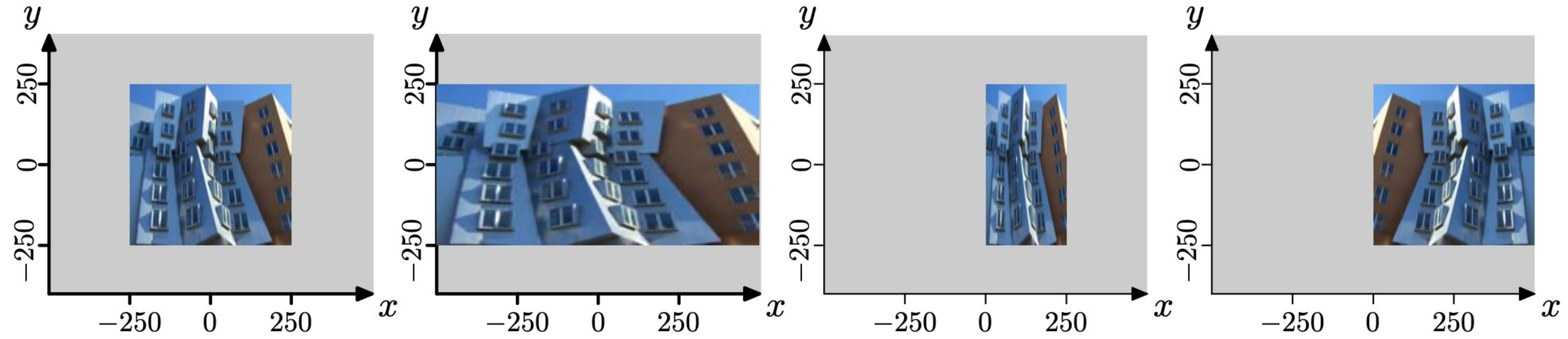
How many images match the expressions beneath them?



$f_1(x,y) = f(2x,y)$? $f_2(x,y) = f(2x-250,y)$? $f_3(x,y) = f(-x-250,y)$?

TUDelft

Check yourself (2D signal)



$f(x,y)$ $\quad$ $f_1(x,y) = f(2x,y)$? $\quad$ $f_2(x,y) = f(2x-250,y)$? $\quad$ $f_3(x,y) = f(-x-250,y)$?

$$x = 0 \quad \rightarrow f_1(0,y) = f(0,y) \qquad \surd$$

$$x = 250 \quad \rightarrow f_1(250,y) = f(500,y) \qquad \times$$

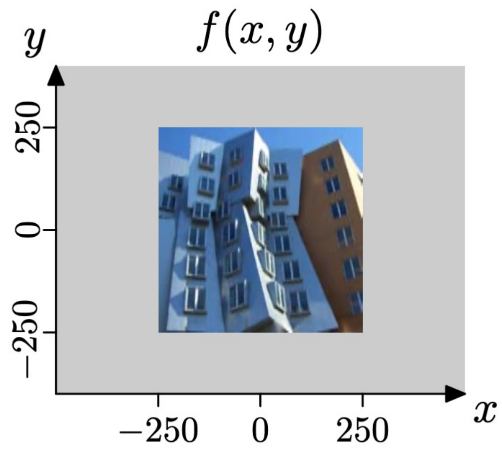
$$x = 0 \quad \rightarrow f_2(0,y) = f(-250,y) \qquad \surd$$

$$x = 250 \quad \rightarrow f_2(250,y) = f(250,y) \qquad \surd$$

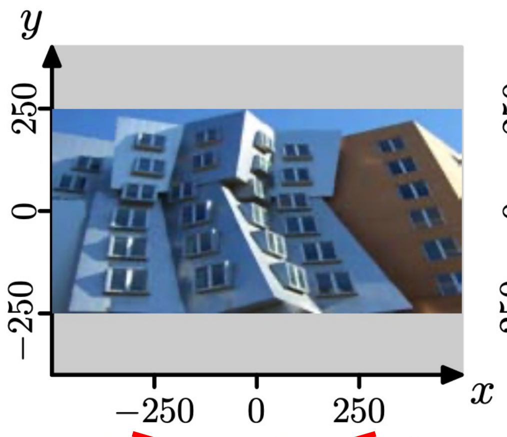
$$x = 0 \quad \rightarrow f_3(0,y) = f(-250,y) \qquad \times$$

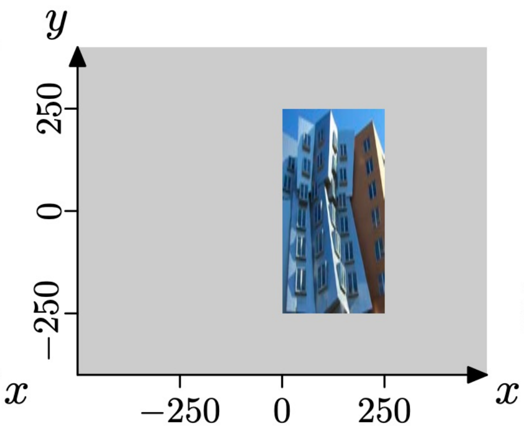
$$x = 250 \quad \rightarrow f_3(250,y) = f(-500,y) \qquad \times$$

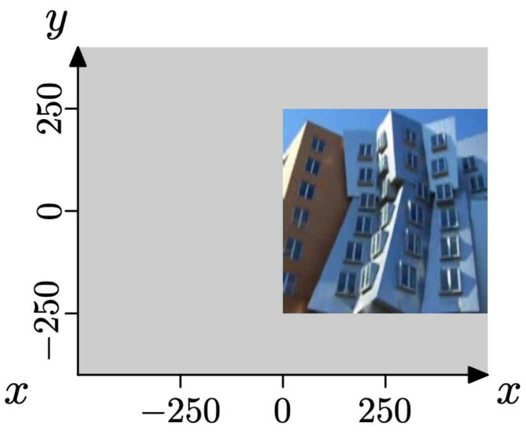
Check yourself (2D signal)



How many images match the expressions beneath them?







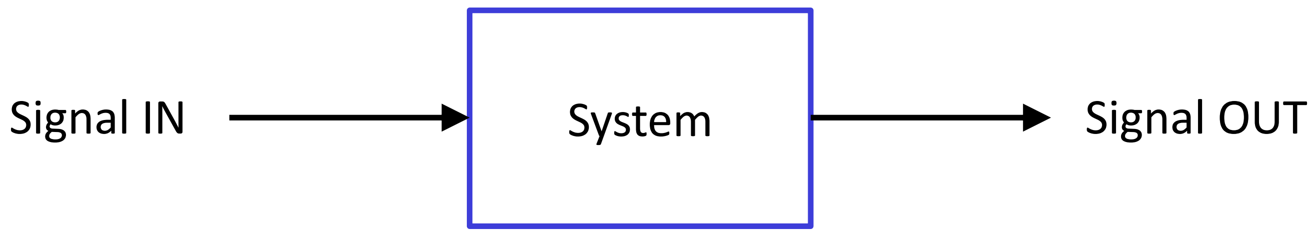
$\cancel{f_1(x,y)=f(2x,y)}$?

$f_2(x,y)=f(2x-250,y)$?

$\cancel{f_3(x,y)=f(-x-250,y)}$?

The Signals and Systems Abstraction

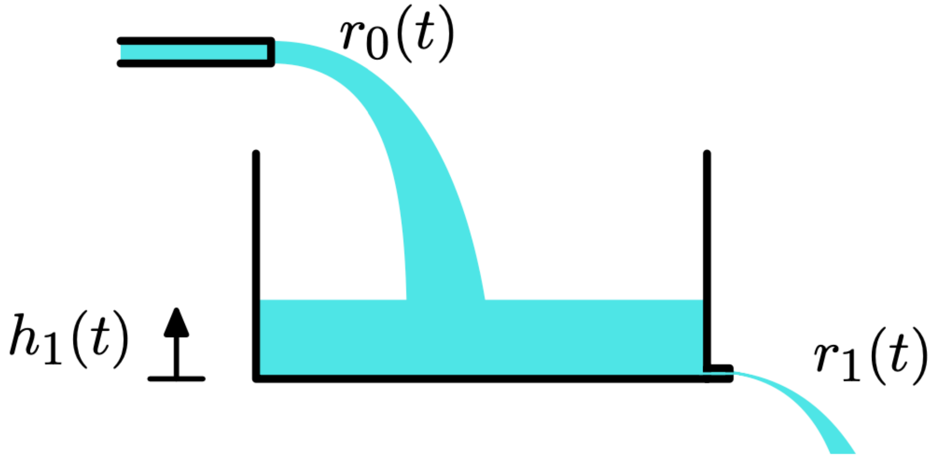
Describe a system (physical, mathematical, or computational) by the way it transforms an input signal into an output signal.



TU Delft

Example System: Leaky Tank

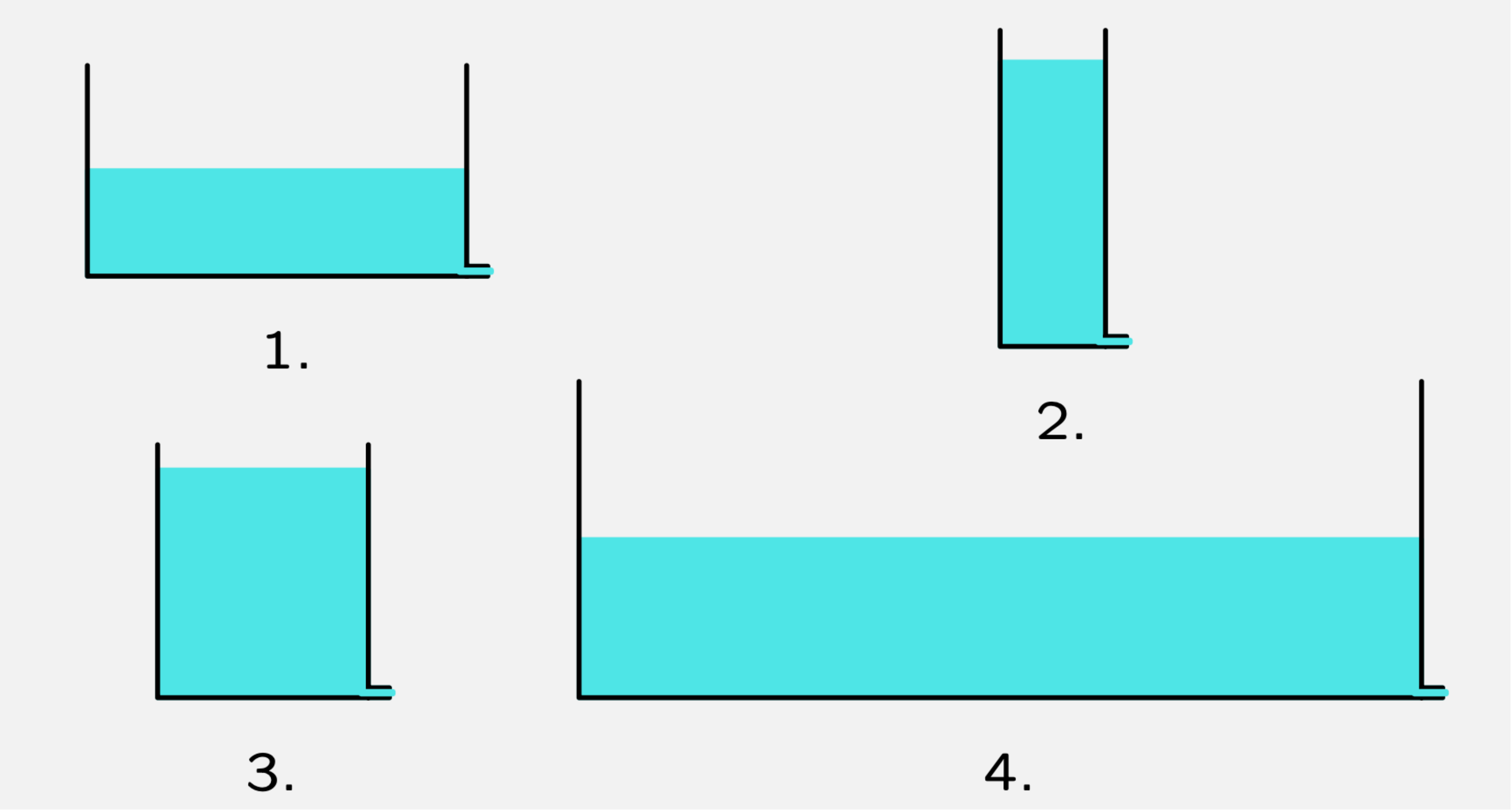
Formulate a mathematical description of this system.



What determines the leak rate?

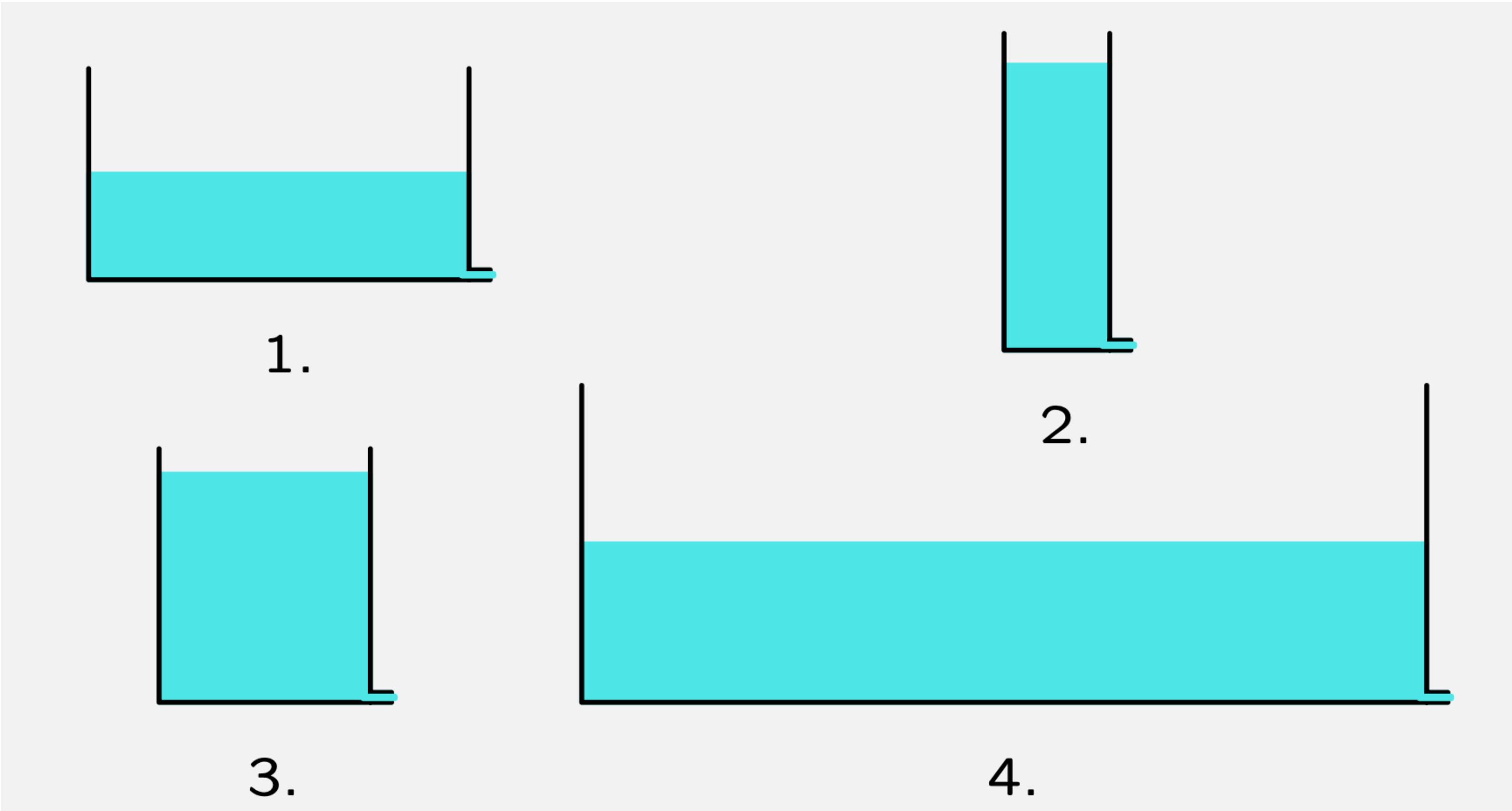
Check yourself

The holes in each of the following tanks have equal size. Which tank has the largest leak rate *r1(t)*?



Check yourself

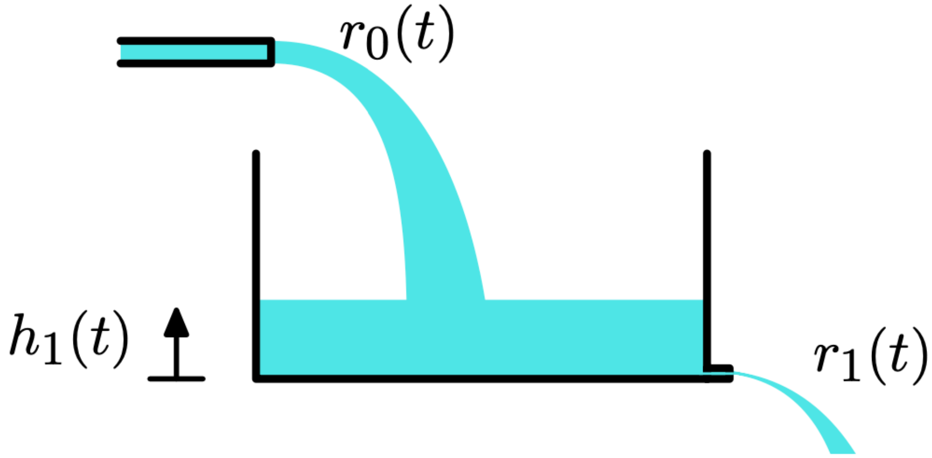
The holes in each of the following tanks have equal size. Which tank has the largest leak rate *r1(t)*?   **2**



TU Delft

Example System: Leaky Tank

Formulate a mathematical description of this system.

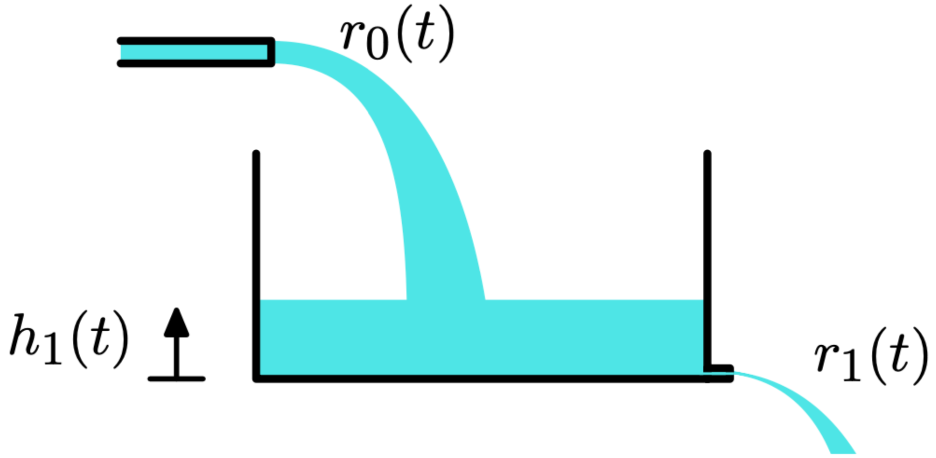


Assume linear leaking: *r1(t)* $\propto$ *h1(t)*

What determines the height *h1(t)*?

Example System: Leaky Tank

Formulate a mathematical description of this system.



Assume linear leaking: $r_1(t) \propto h_1(t)$

Assume water is conserved: $\frac{dh_1(t)}{dt} \propto r_0(t) - r_1(t)$

Solve: $\frac{dr_1(t)}{dt} \propto r_0(t) - r_1(t)$

Check yourself

What are the dimensions of constant of proportionality C?

$$\frac{dr_1(t)}{dt} = C\left(r_0(t) - r_1(t)\right)$$

Check yourself

What are the dimensions of constant of proportionality C?
inverse time (to match dimensions of *dt*)

$$\frac{dr_1(t)}{dt} = C(r_0(t) - r_1(t))$$

TU Delft

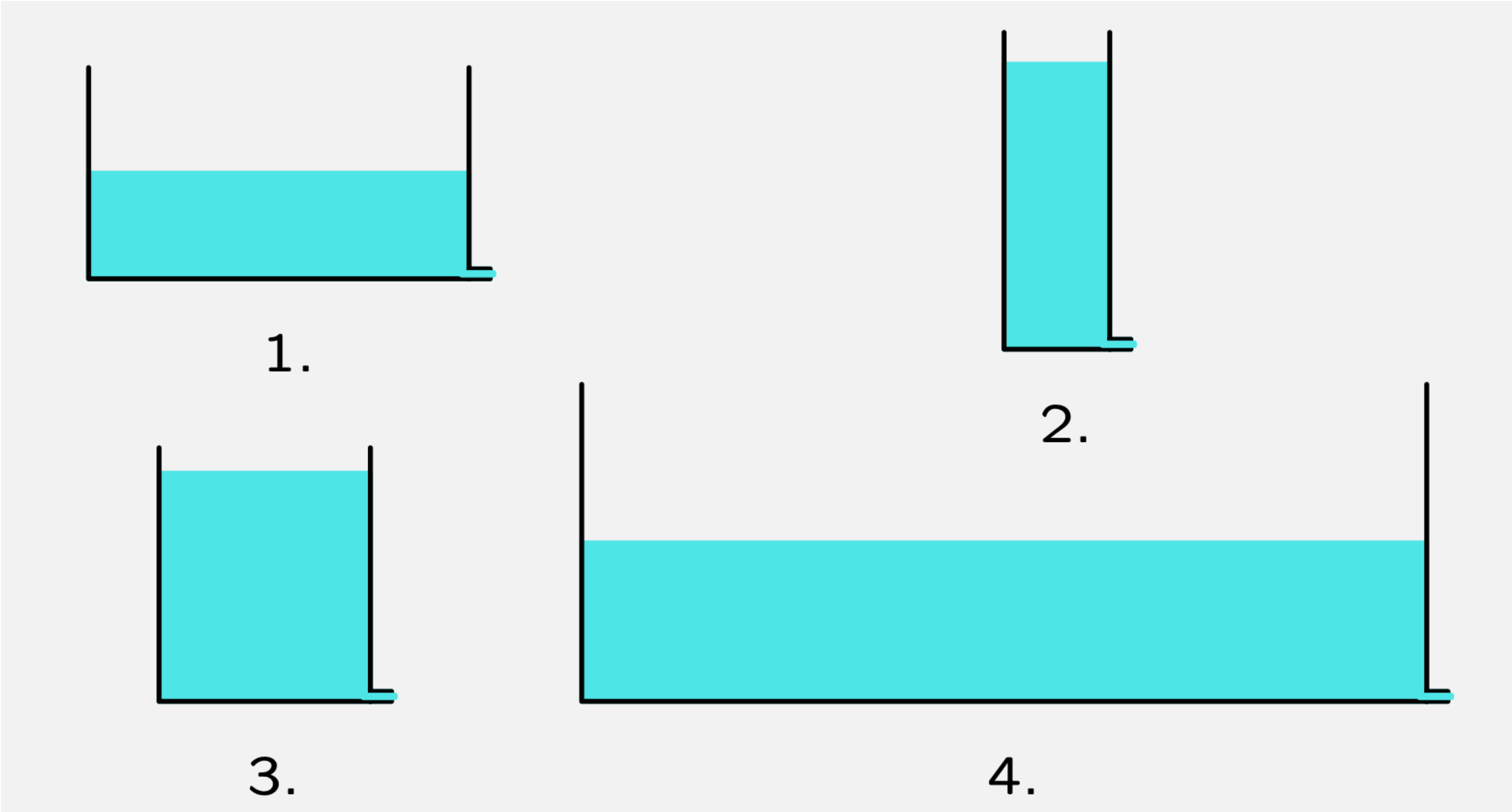
Analysis of the Leaky tank

Call the constant of proportionality $1/\tau$
Then τ is called the **time constant** of the system.

$$\frac{dr_1(t)}{dt} = \frac{r_0(t)}{\tau} - \frac{r_1(t)}{\tau}$$

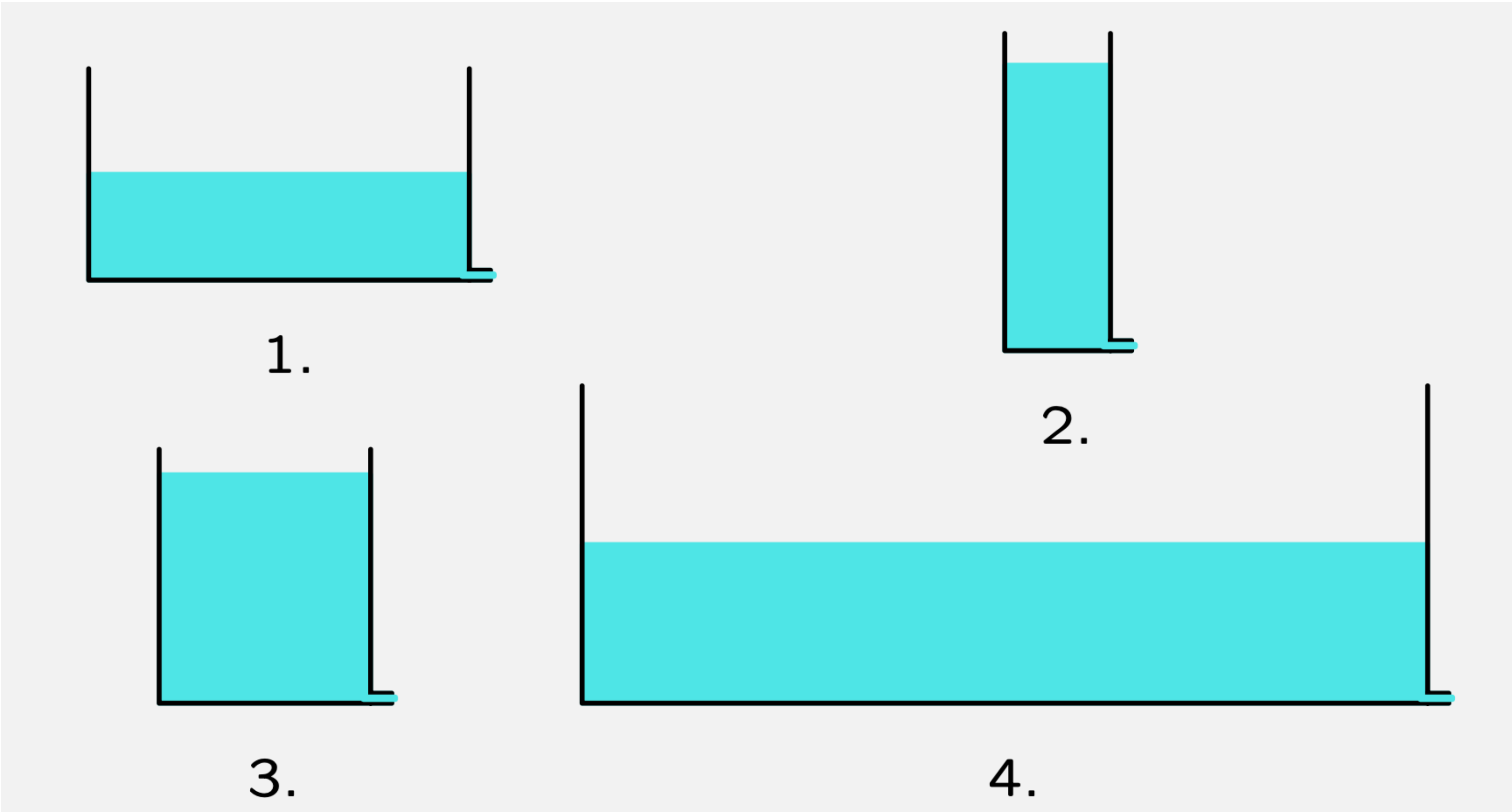
Check yourself

Which tank has the largest time constant τ ?



Check yourself

Which tank has the largest time constant τ ? **4**

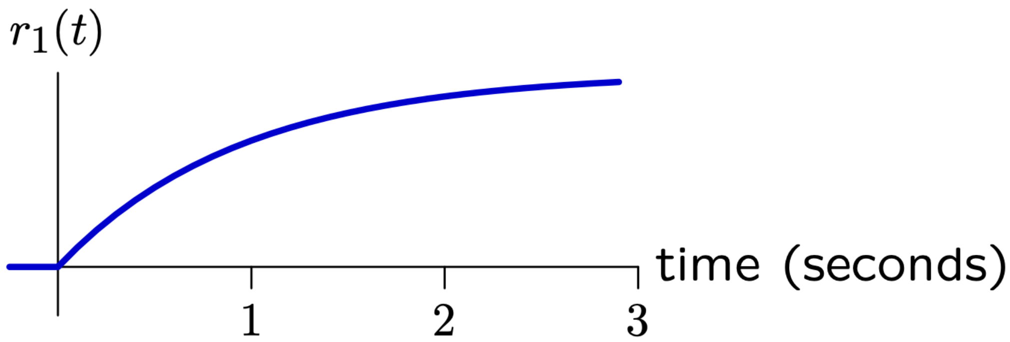


Analysis of the Leaky tank

Call the constant of proportionality $1/\tau$
Then τ is called the **time constant** of the system.

$$\frac{dr_1(t)}{dt} = \frac{r_0(t)}{\tau} - \frac{r_1(t)}{\tau}$$

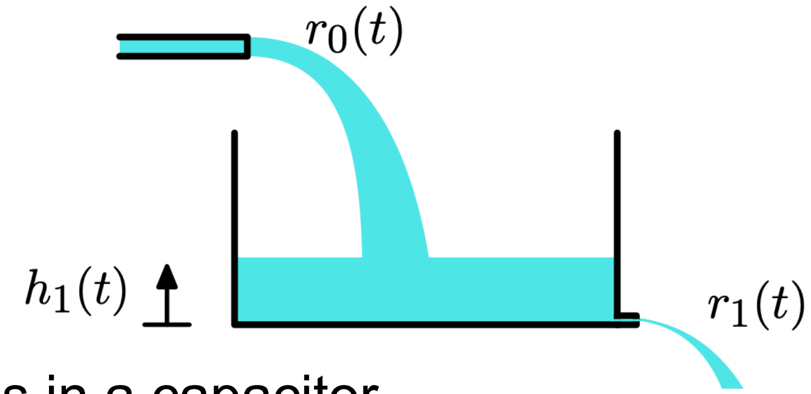
Assume that the tank is initially empty, and then water enters at a constant rate $r_0(t) = 1$. Determine the output rate $r_1(t)$.



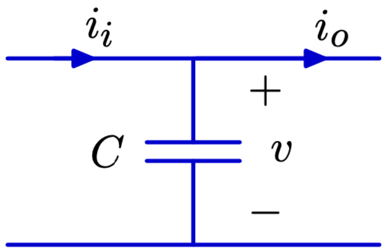
Explain the shape of this curve mathematically.
Explain the shape of this curve physically.

Leaky Tanks and Capacitors

Although derived for a leaky tank, this sort of model can be used to represent a variety of physical systems.
Water accumulates in a leaky tank.



Charge accumulates in a capacitor.



$$\frac{dv}{dt} = \frac{i_i(t) - i_o(t)}{C} \propto i_i - i_o \quad \text{analogous to} \quad \frac{dh}{dt} \propto r_0 - r_1$$

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)

But also:
Algebra is simpler than Calculus
Computers are Discrete Systems

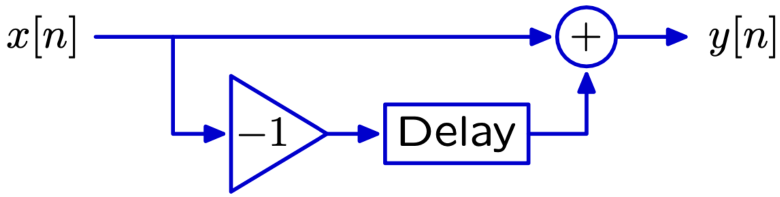
Discrete Time (DT) Systems

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.

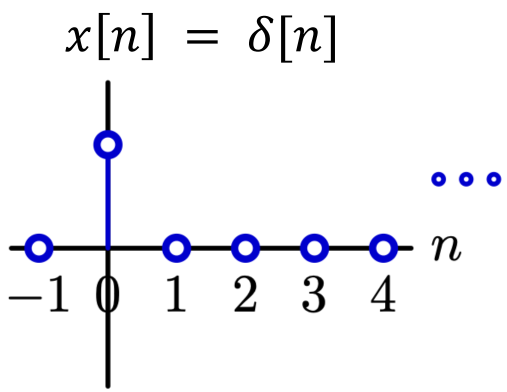
Difference Equations

Difference equations are mathematically precise and compact.

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

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

$$\delta[n] = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases}$$



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. $y[119] = 0$

TU Delft

Step-by-step solutions

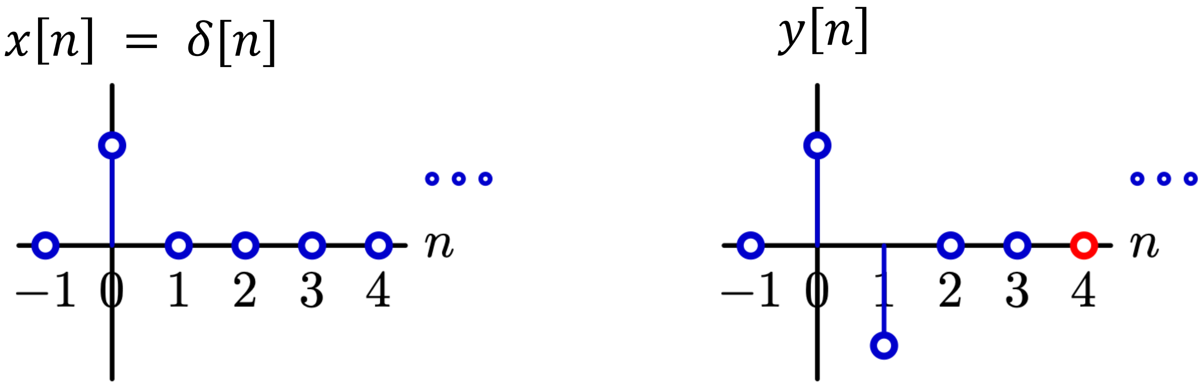
Difference equations are convenient for step-by-step analysis.

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

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

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

...



Check yourself

Solve: $y[n] = x[n] - x[n-1]$
Given: $x[n] = \delta[n]$

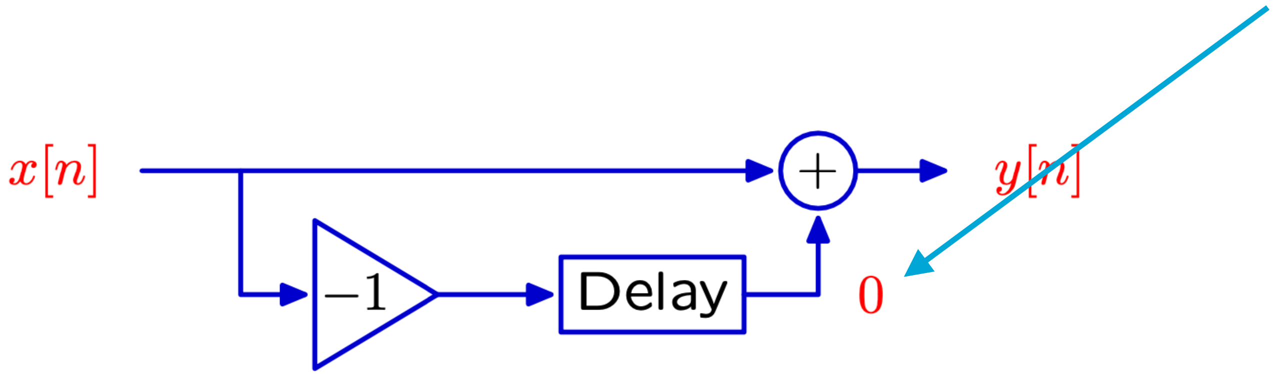
How many of the following are true? **4**

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

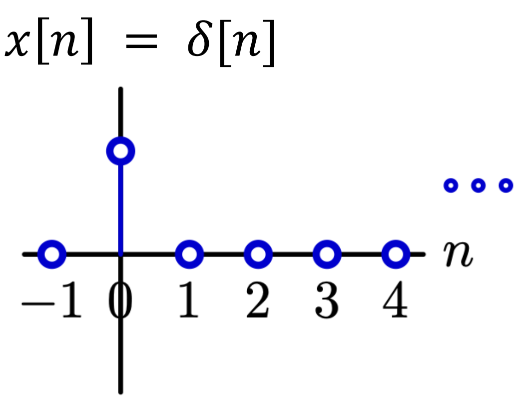
TUDelft

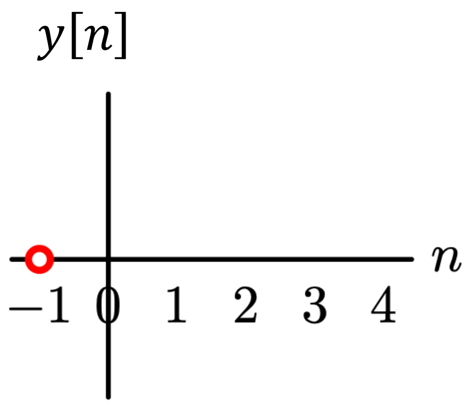
Step-by-step solutions

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"



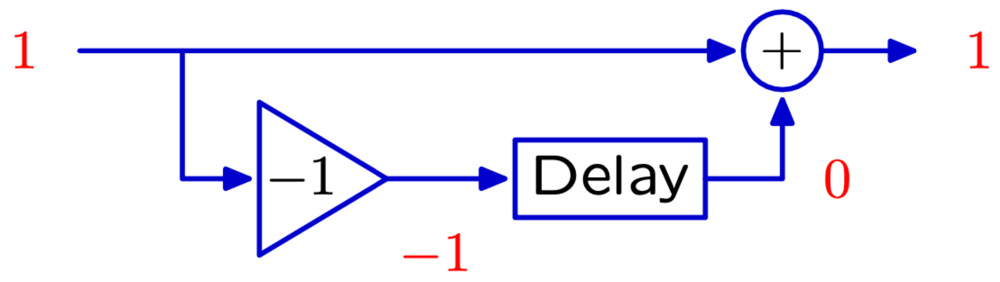
Can be directly implemented in HW

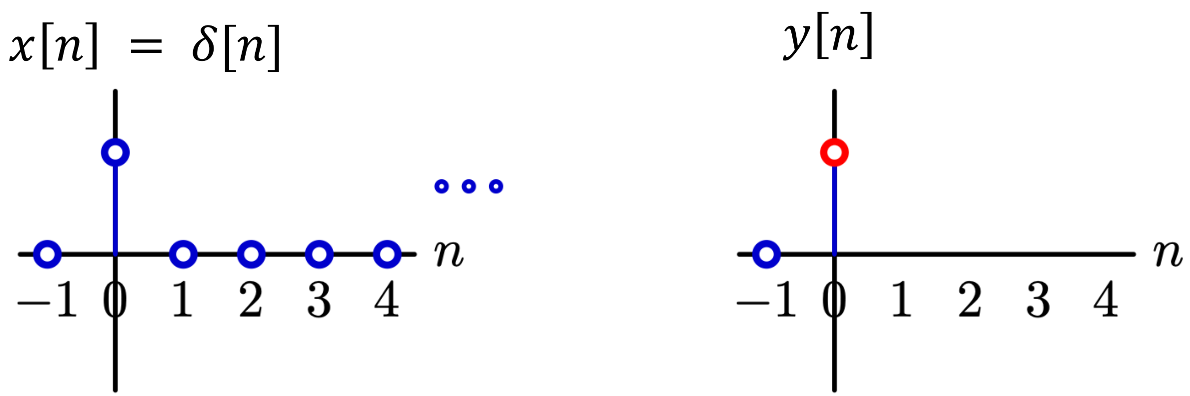




Step-by-step solutions

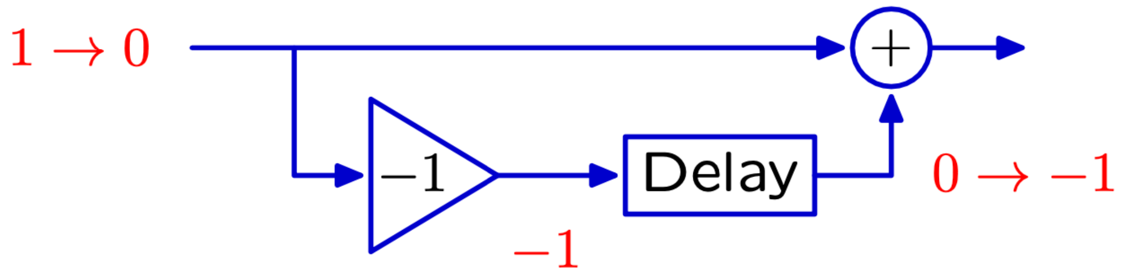
Block diagrams are also useful for step-by-step analysis.
Represent $y[n] = x[n] - x[n-1]$ with a block diagram: start "at rest"

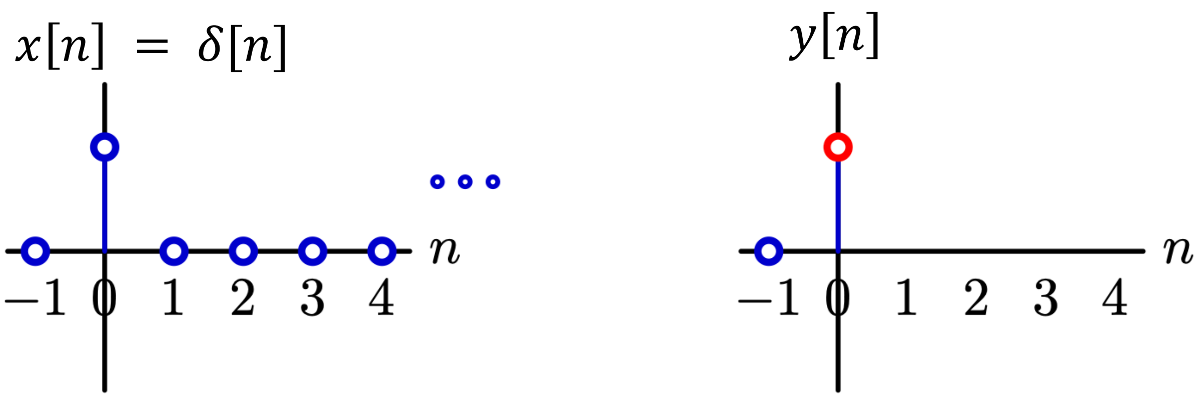




Step-by-step solutions

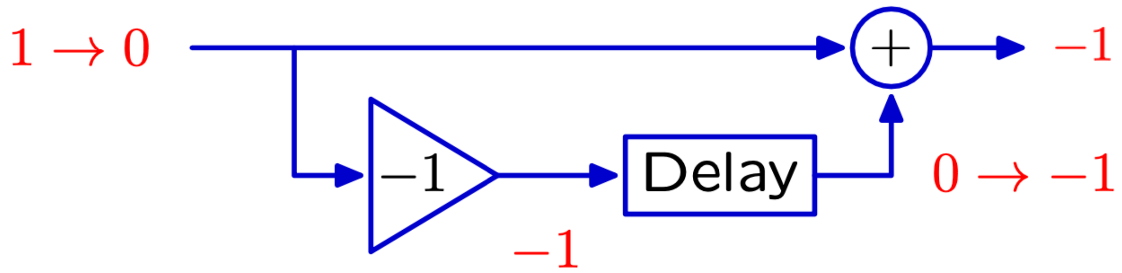
Block diagrams are also useful for step-by-step analysis.
Represent $y[n] = x[n] - x[n-1]$ with a block diagram: start "at rest"

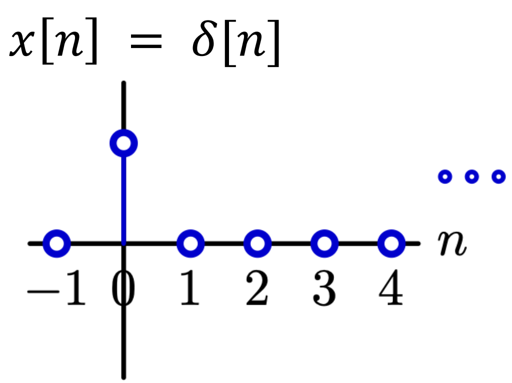


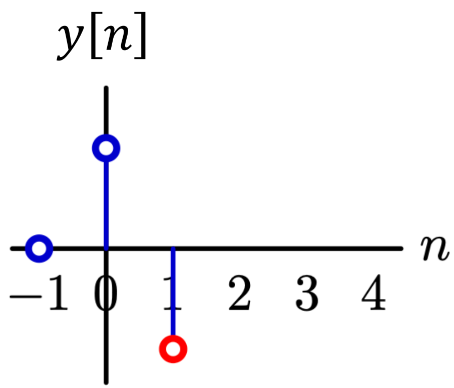


Step-by-step solutions

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"

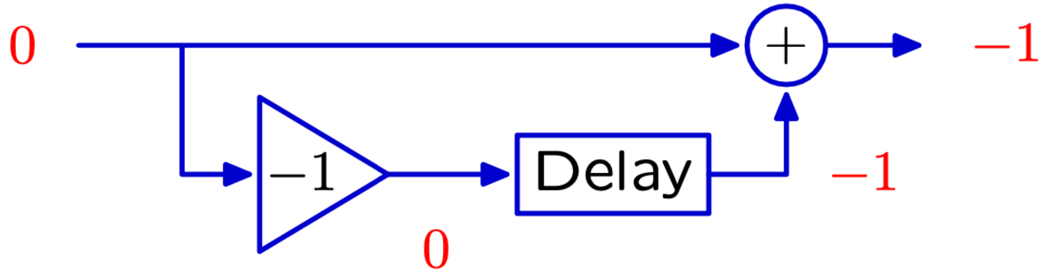


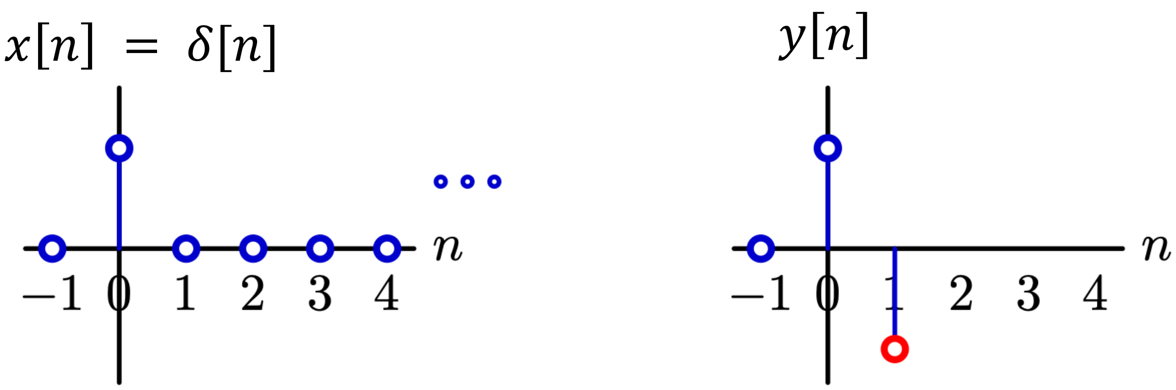




Step-by-step solutions

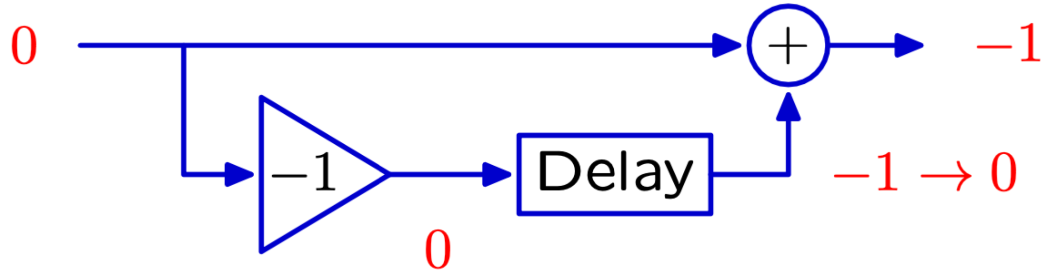
Block diagrams are also useful for step-by-step analysis.
Represent $y[n] = x[n] - x[n-1]$ with a block diagram: start "at rest"

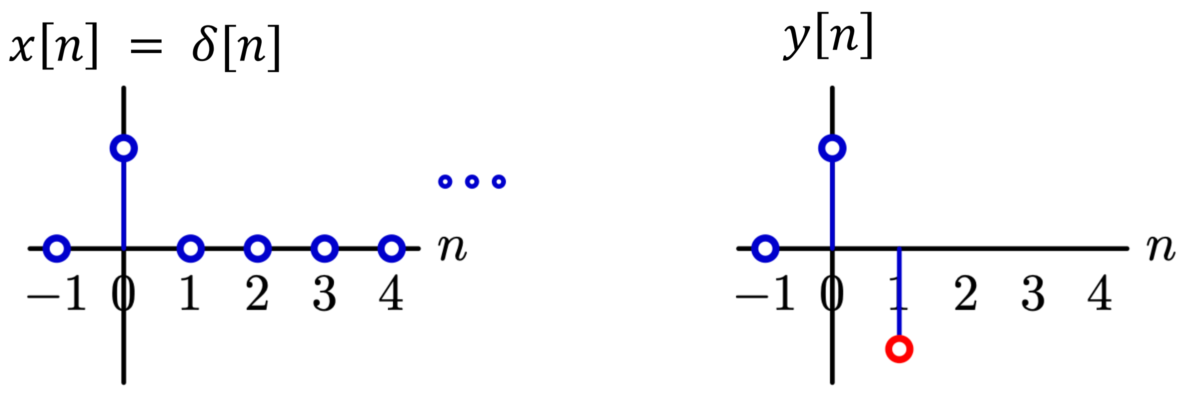




Step-by-step solutions

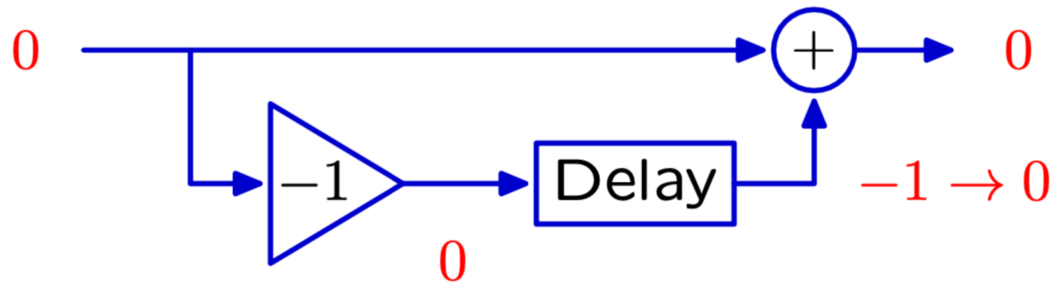
Block diagrams are also useful for step-by-step analysis.
Represent $y[n] = x[n] - x[n-1]$ with a block diagram: start "at rest"

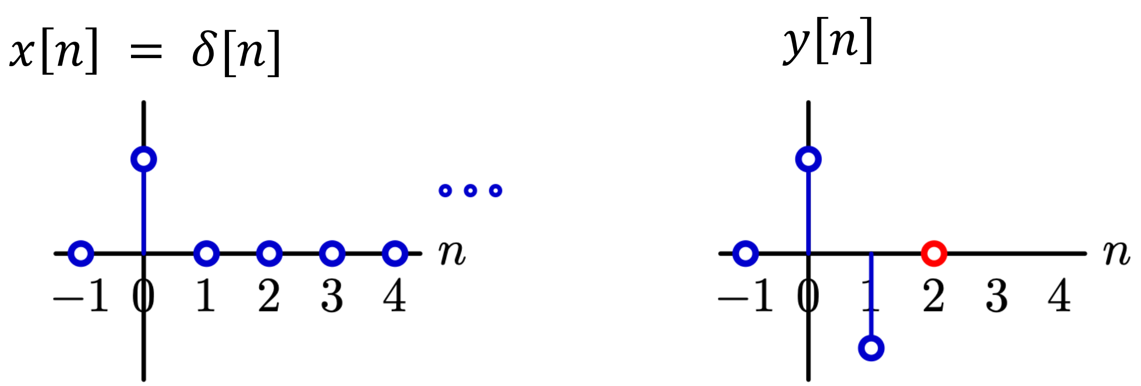




Step-by-step solutions

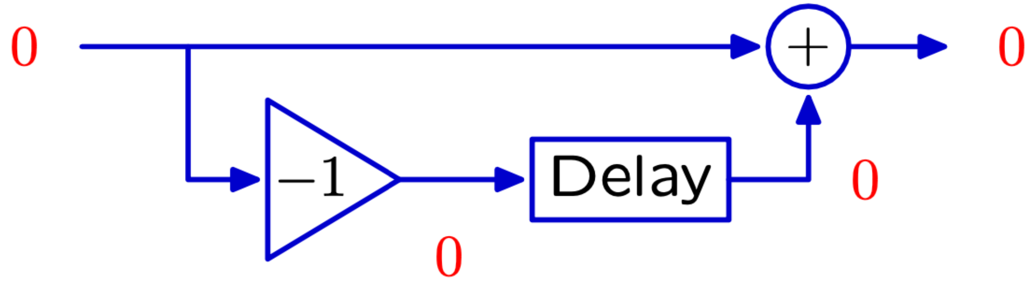
Block diagrams are also useful for step-by-step analysis.
Represent $y[n] = x[n] - x[n-1]$ with a block diagram: start "at rest"

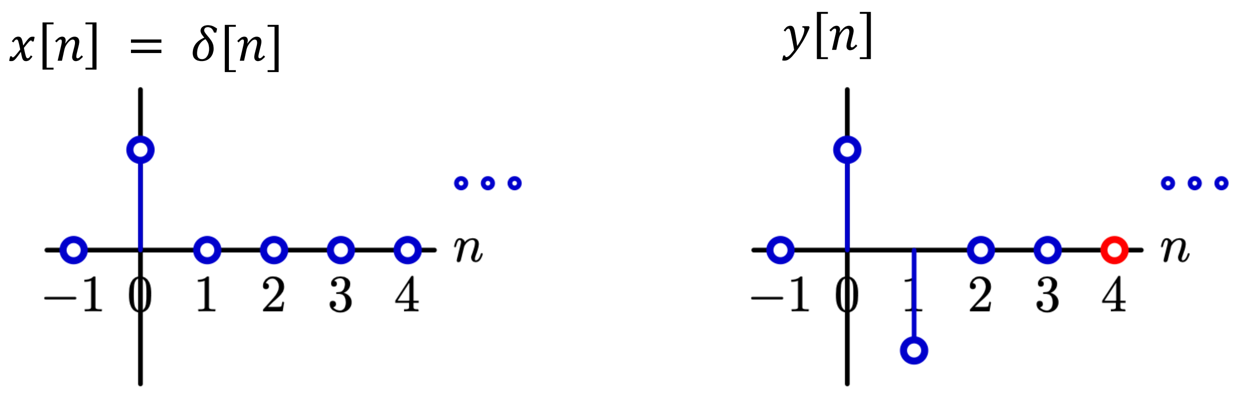




Step-by-step solutions

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"



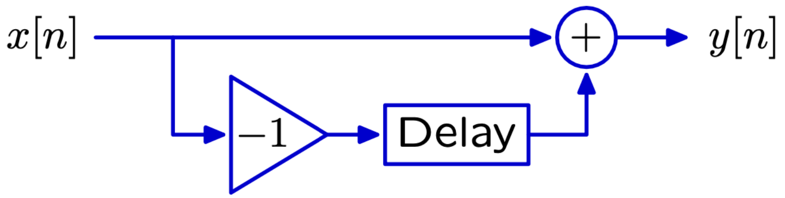


Check yourself

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?

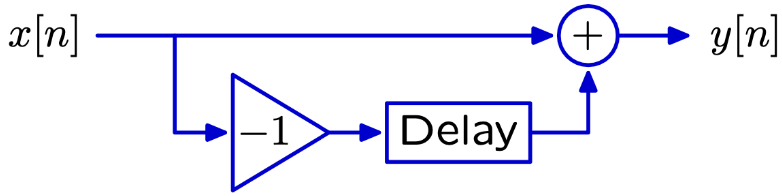
Check yourself

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.

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

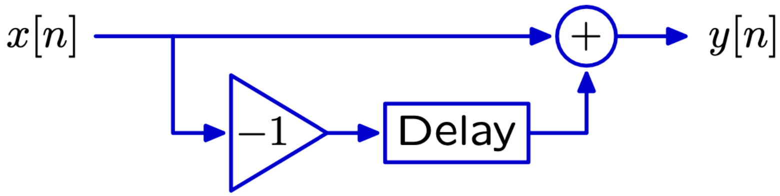
Lumping all of the (possibly infinite) samples into a single object — the signal — simplifies its manipulation.

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.

TU Delft

Operator Notation

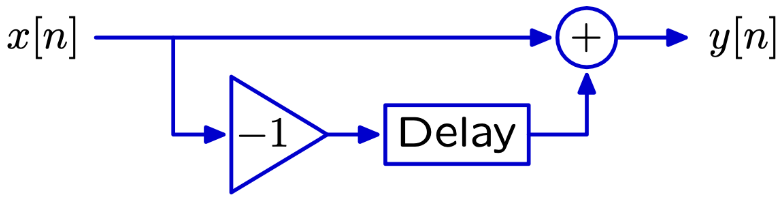
Symbols can now compactly represent diagrams.

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

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

As consize as difference equations but imperative

TU Delft

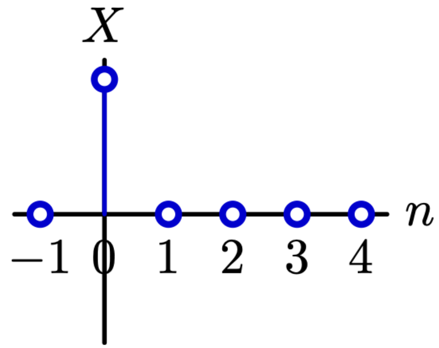
Check yourself (Operator Notation)

Let $Y = \mathcal{R}X$ how many of the following are true?

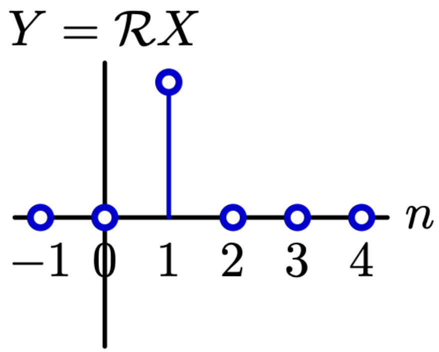
1. $y[n] = x[n] \;\; \forall\, n$
2. $y[n+1] = x[n] \; \forall\, n$
3. $y[n] = x[n+1] \; \forall\, n$
4. $y[n-1] = x[n] \forall\, n$
5. none of the above

Check yourself (Operator Notation)

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.

Check yourself (Operator Notation)

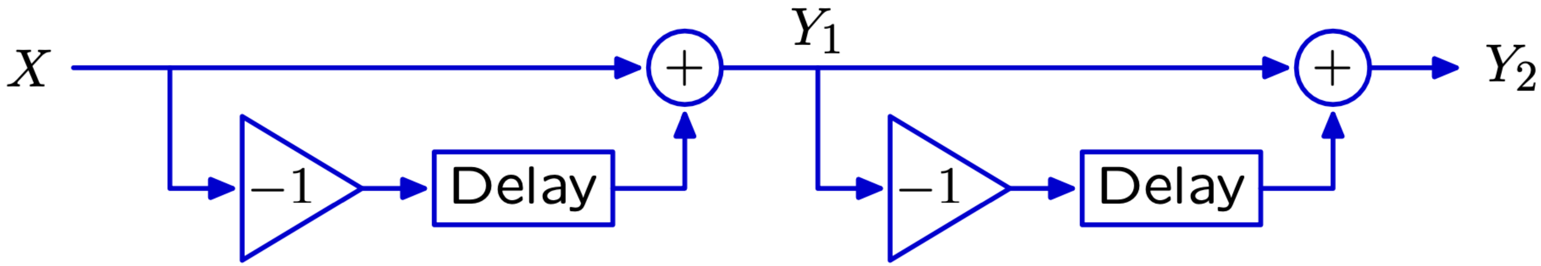
Let $Y = \mathcal{R}X$ how many of the following are true?

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

Operator Representation of a Cascaded System

System operations have simple operator representations.

Cascade systems $\rightarrow$ multiply operator expressions.



Using operator notation:

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

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

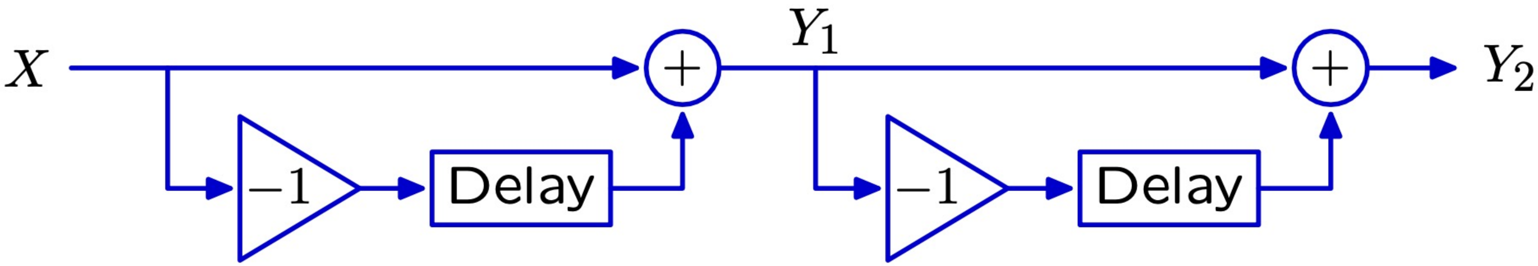
Substituting for Y_1:

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

TU Delft

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:

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

Isomorphism between
polinomials and DT systems

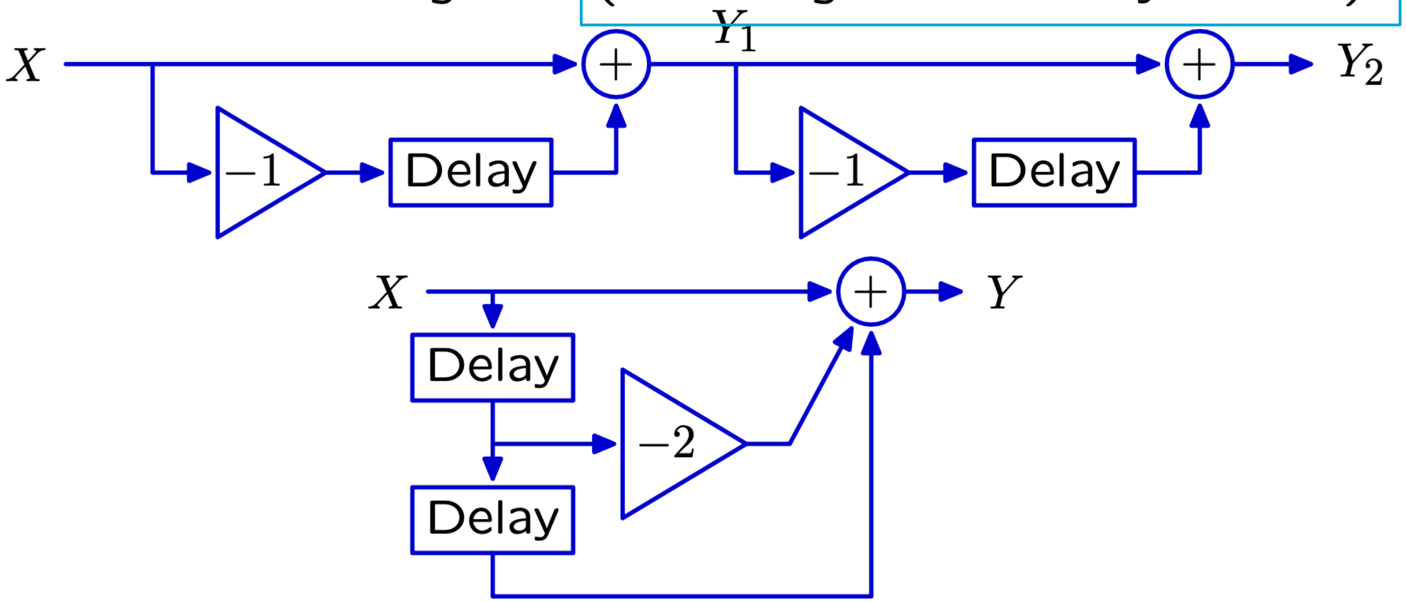
Operator Approach

Applies your 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:

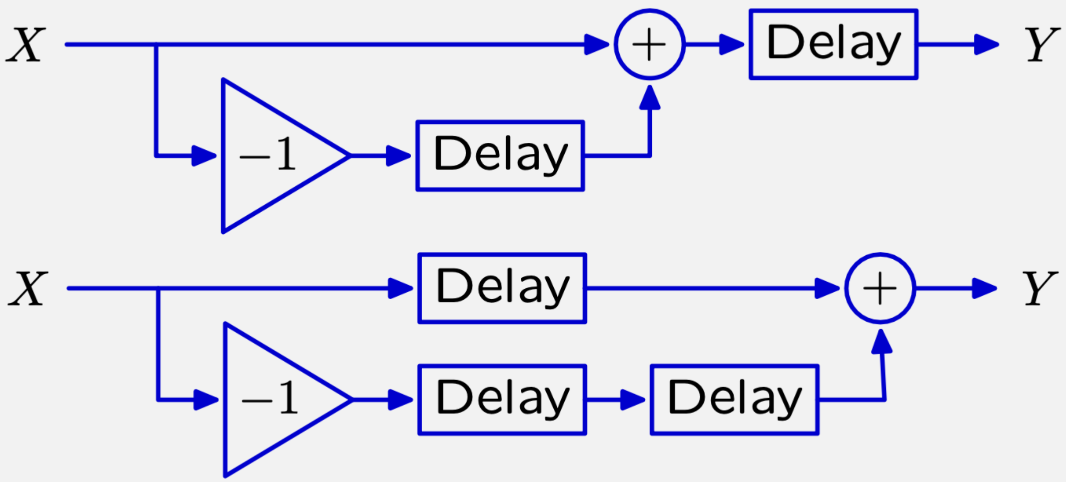
$$(1 - \mathcal{R})(1 - \mathcal{R}) = 1 - 2\mathcal{R} + \mathcal{R}^2$$

The operator equivalence is much easier to see.

TU Delft

Check yourself

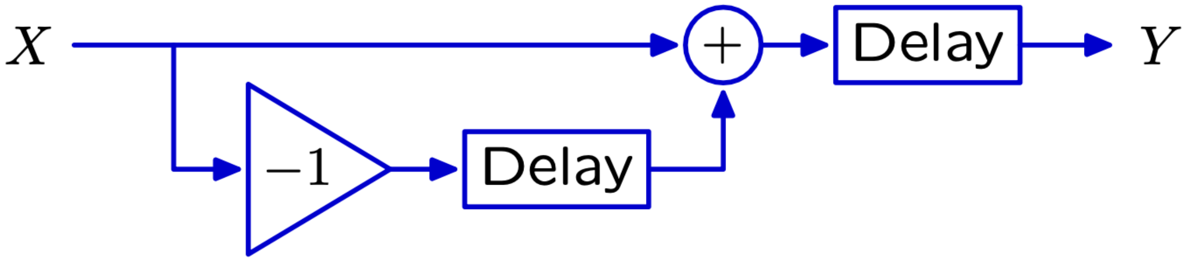
Operator expressions for these "equivalent" systems (if started "at rest") obey what mathematical property?



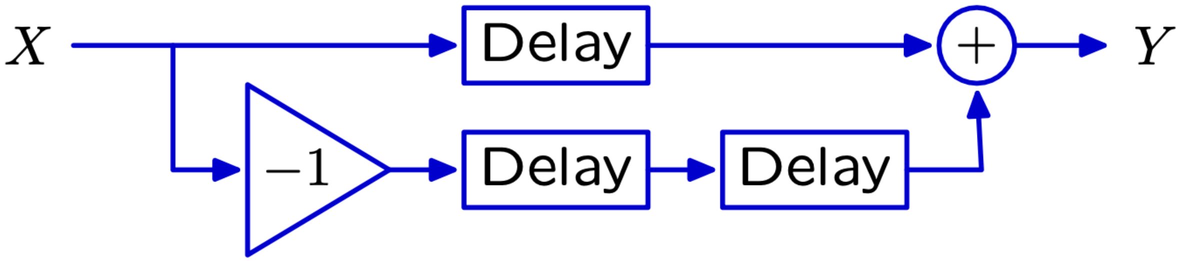
1. commutate
2. associative
3. distributive
4. transitive
5. none of the above

What property of polinomials is emphasized here?

Check yourself



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

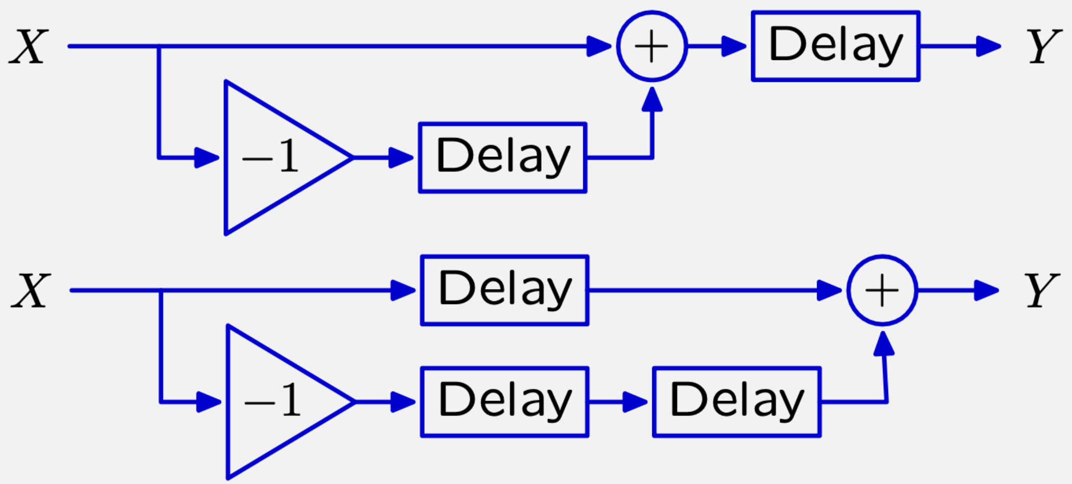


$$Y = (\mathcal{R} - \mathcal{R}^2)X$$

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

Check yourself

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

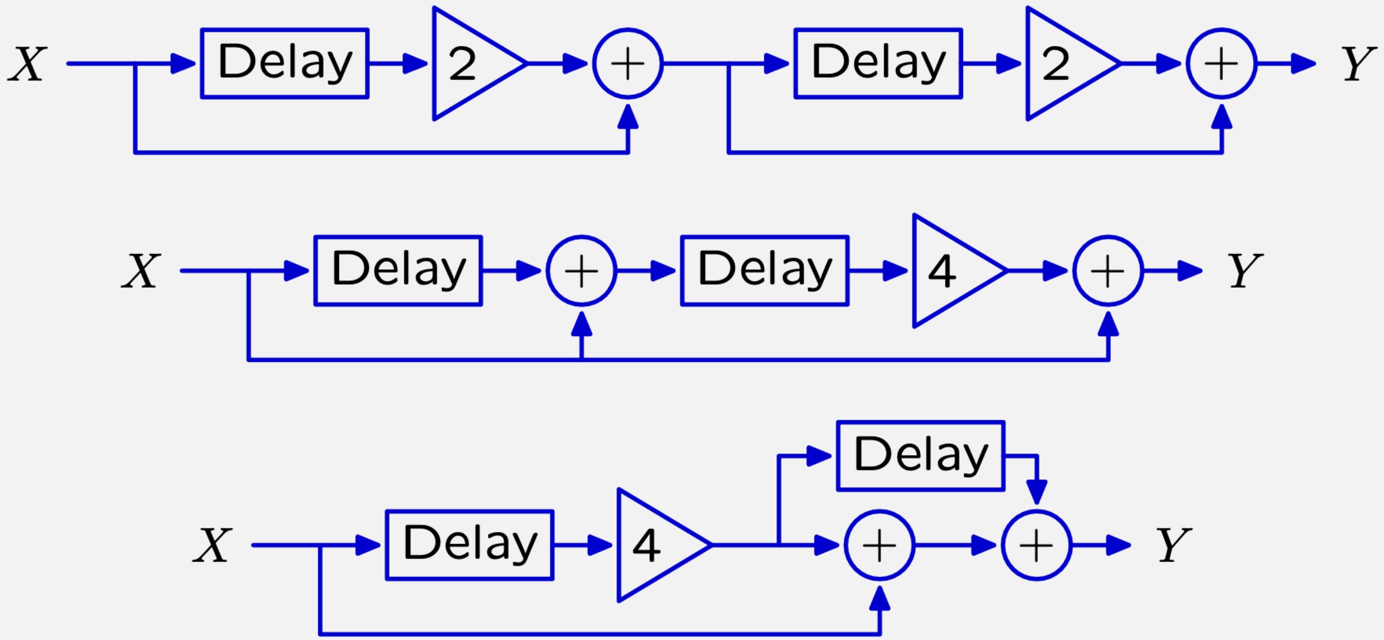


1. commutate
2. associative
3. distributive
4. transitive
5. none of the above

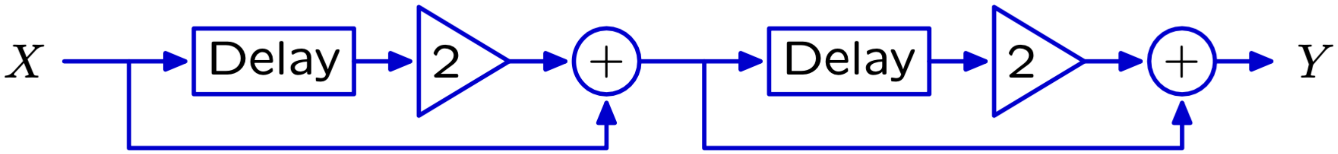
Check yourself

How many of the following systems are equivalent to

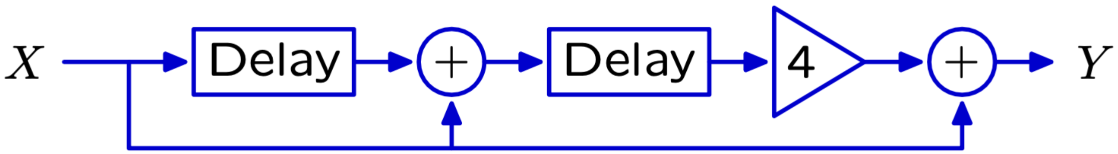
$$Y = (4\mathcal{R}^2 + 4\mathcal{R} + 1)\,X \quad ?$$



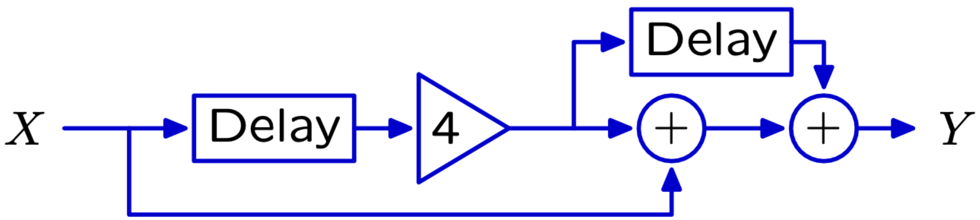
Check yourself



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



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

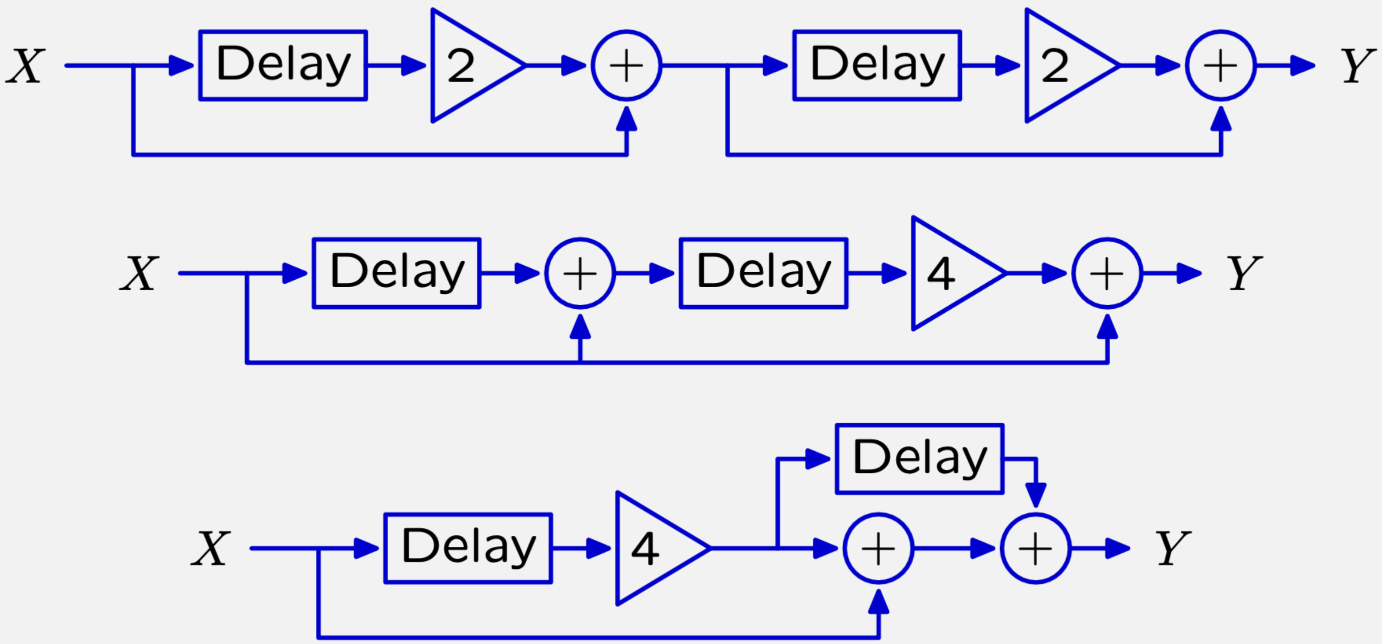


$$Y = (4\mathcal{R}^2+4\mathcal{R}+1)\,X$$

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

Check yourself

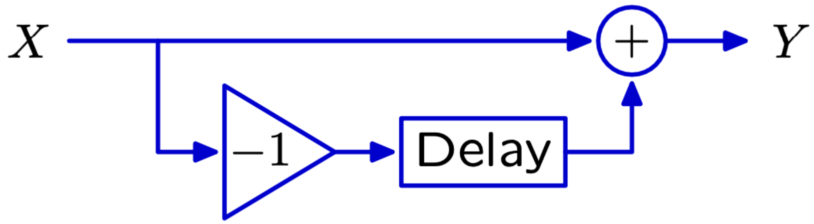
How many of the following systems are equivalent to
$Y = (4\mathcal{R}^2 + 4\mathcal{R} + 1)\,X$? 3



Operator Algebra: Explicit and Implicit Rules

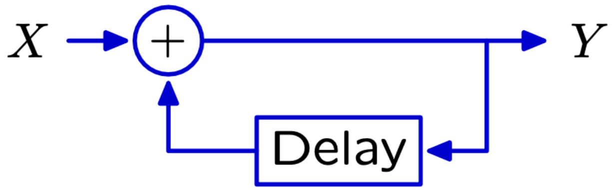
Recipes versus constraints.

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.



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

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

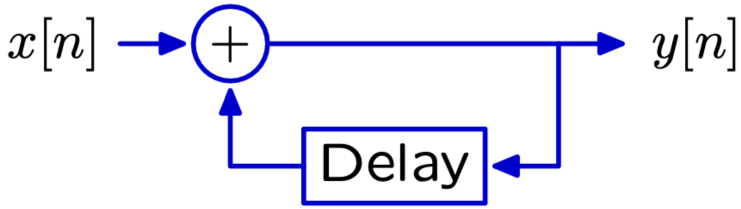
declarative!
(compute the input from the output)

But how does one solve such a constraint?

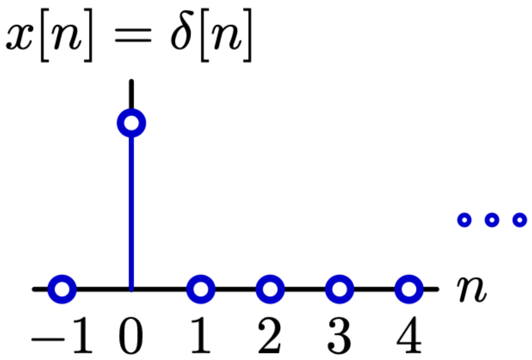
Example: Accumulator

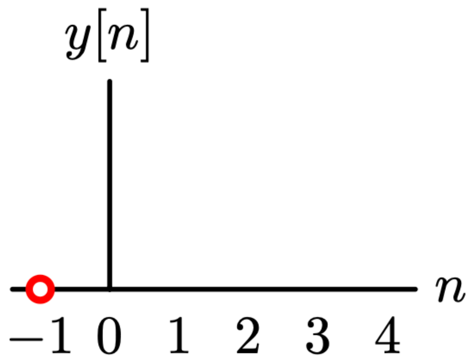
Try step-by-step analysis: it always works. Start "at rest."

(lower the abstraction)



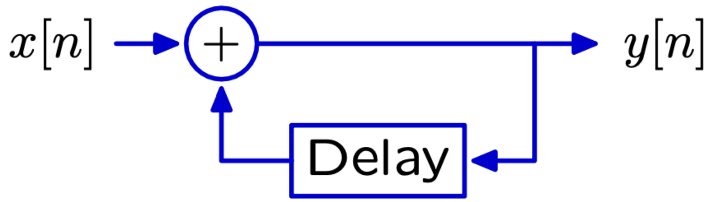
Find $y[n]$ given $x[n] = \delta[n]$: $\quad y[n] = x[n] + y[n-1]$





Example: Accumulator

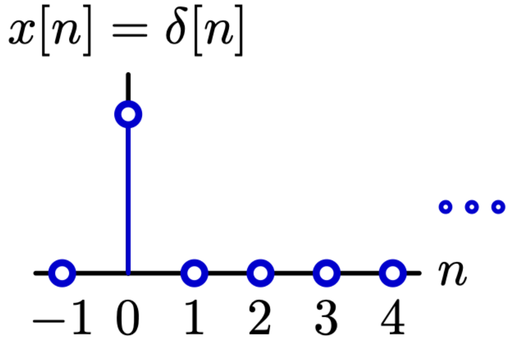
Try step-by-step analysis: it always works. Start "at rest."

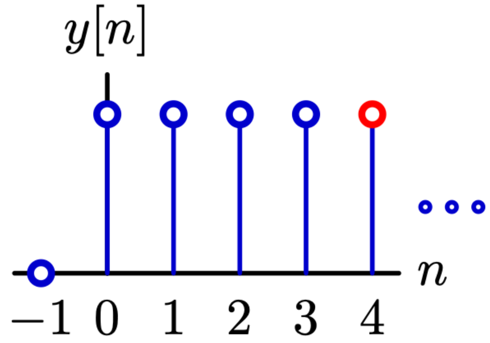


Find $y[n]$ given $x[n] = \delta[n]$:

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

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

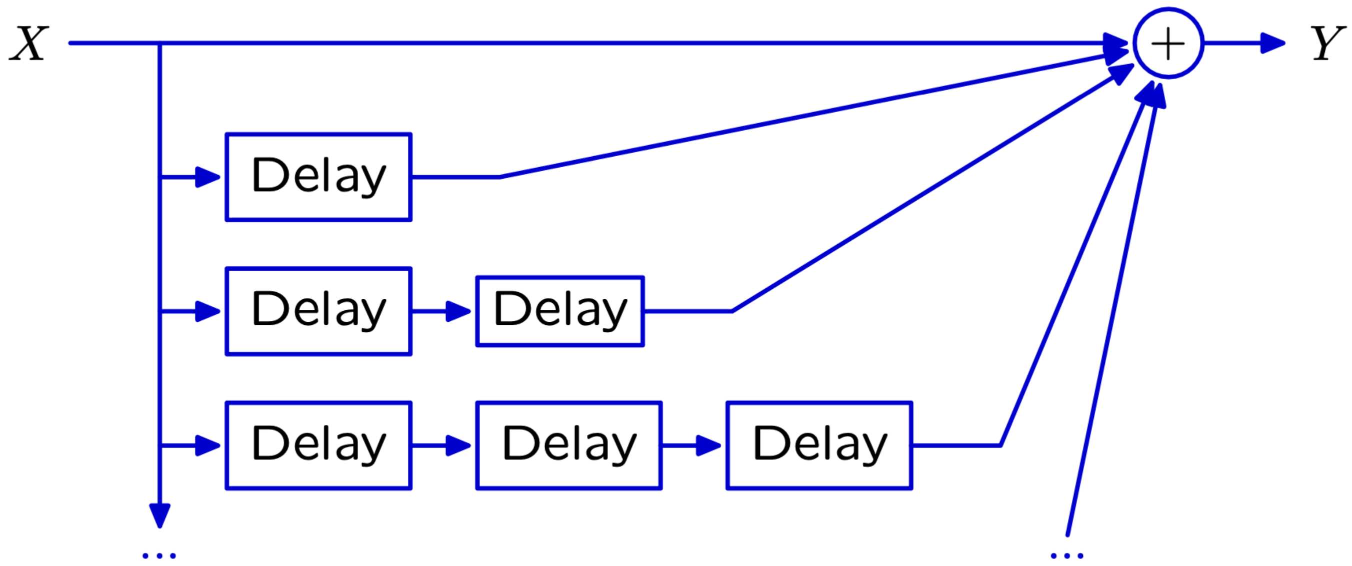




Persistent response to a transient input!

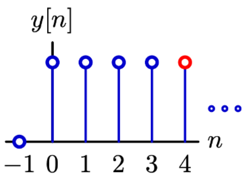
Example: Accumulator

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.



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

This is an imperative system



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.

$$
\begin{array}{r|lllll}
 & 1 & +\mathcal{R} & +\mathcal{R}^2 & +\mathcal{R}^3 & +\cdots \\
\hline
1-\mathcal{R} & 1 & & & & \\
 & 1 & -\mathcal{R} & & & \\
\cline{2-3}
 & & \mathcal{R} & & & \\
 & & \mathcal{R} & -\mathcal{R}^2 & & \\
\cline{3-4}
 & & & \mathcal{R}^2 & & \\
 & & & \mathcal{R}^2 & -\mathcal{R}^3 & \\
\cline{4-5}
 & & & & \mathcal{R}^3 & \\
 & & & & \mathcal{R}^3 & -\mathcal{R}^4 \\
\cline{5-6}
 & & & & & \cdots
\end{array}
$$

Therefore

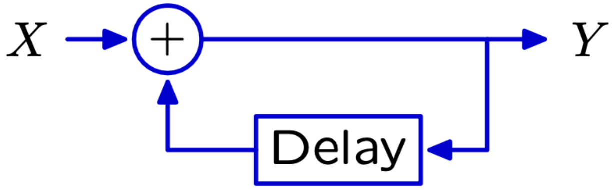
$$\frac{1}{1-\mathcal{R}} = 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \mathcal{R}^4 + \cdots$$

TUDelft

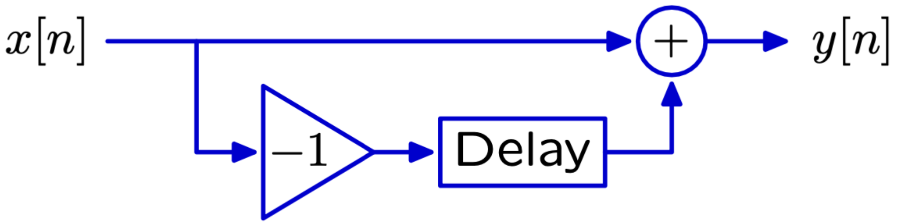
Feedback

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

Example: The accumulator system has feedback.



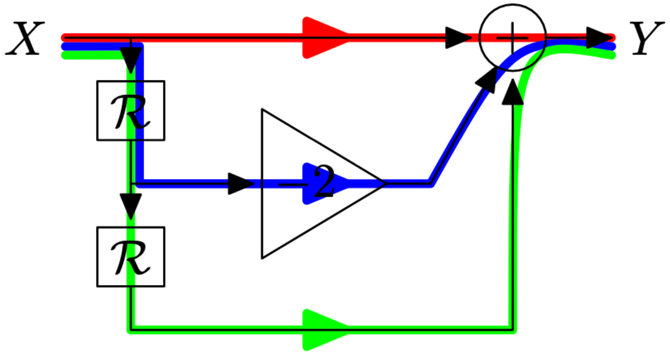
By contrast, the difference machine does not have feedback.



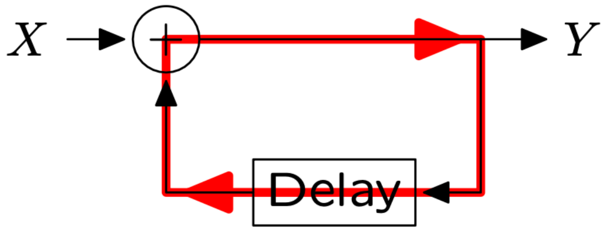
Cyclic Signal Paths, Feedback, and Modes

Block diagrams help visualize feedback.

Feedback occurs when there is a cyclic signal flow path.



acyclic



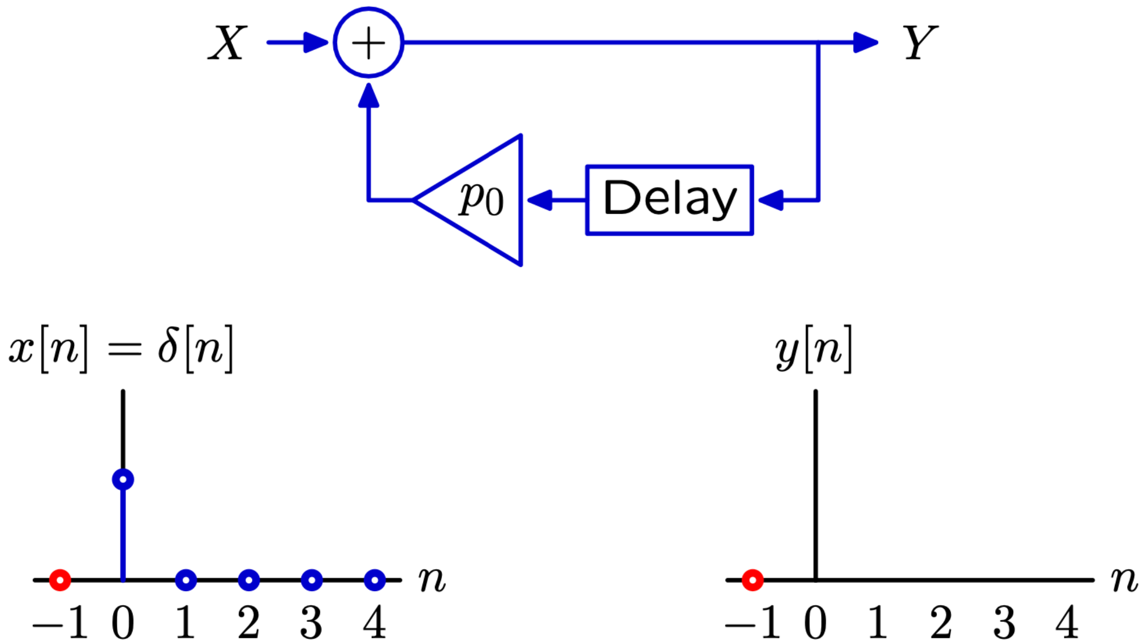
cyclic

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

Cyclic: at least one cycle.

Cyclic Signal Paths, Feedback, and Modes

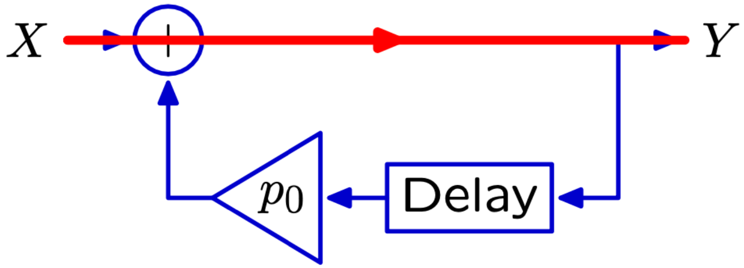
The effect of feedback can be visualized by tracing each cycle through the cyclic signal paths.

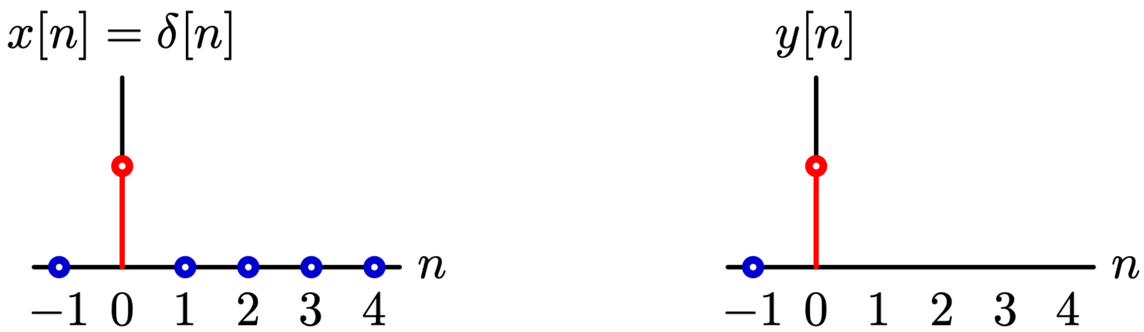


Each cycle creates another sample in the output.

Feedback, Cyclic Signal Paths and Modes

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

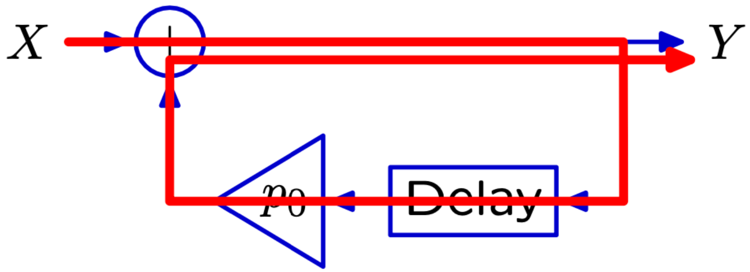


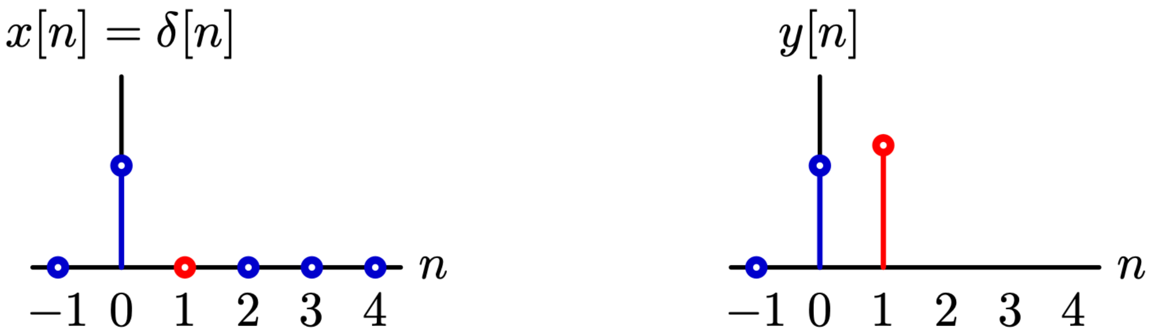


Each cycle creates another sample in the output.

Feedback, Cyclic Signal Paths and Modes

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

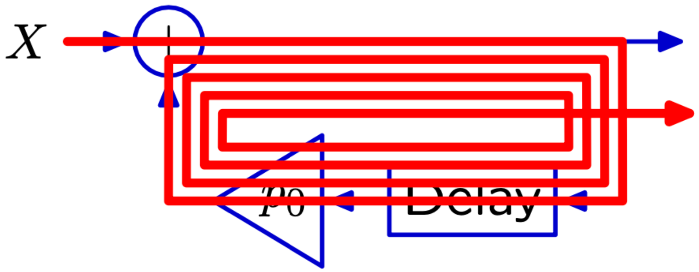


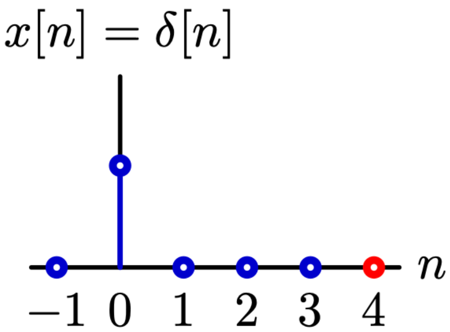


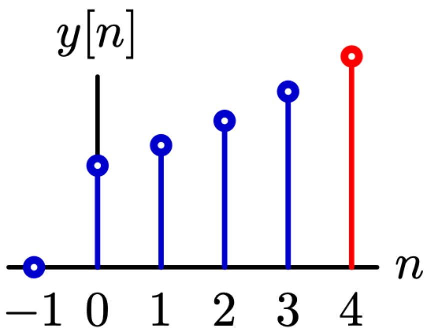
Each cycle creates another sample in the output.

Feedback, Cyclic Signal Paths and Modes

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





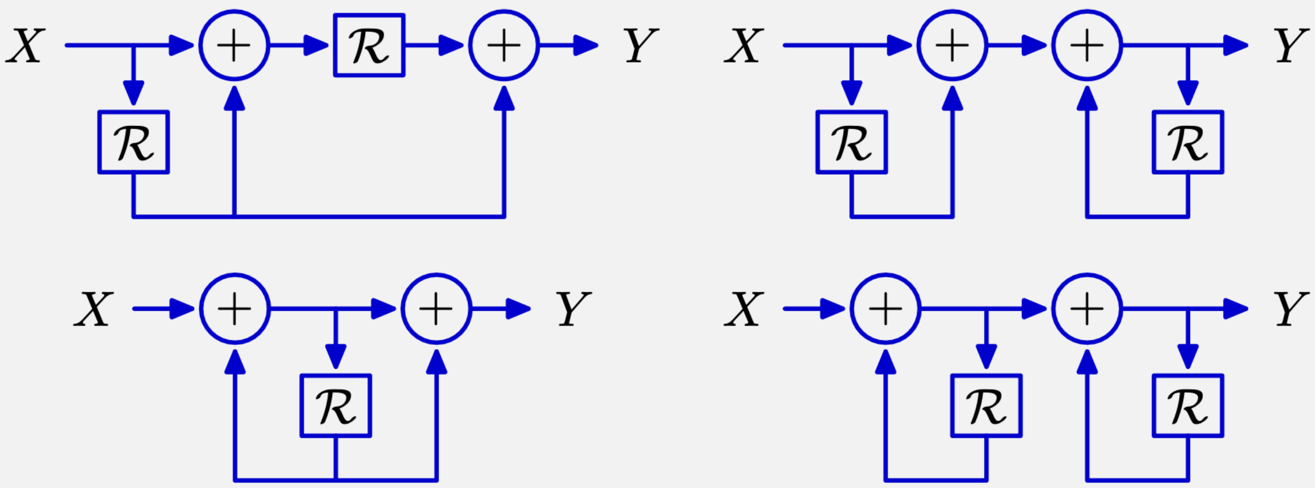


Each cycle creates another sample in the output.

The response will persist even though the input is transient.

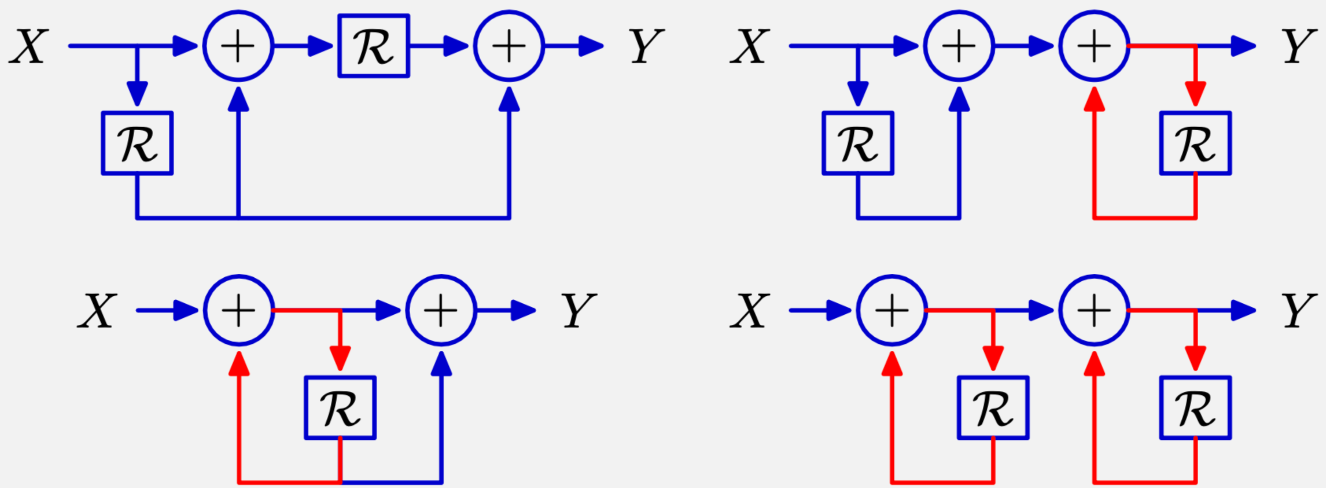
Check yourself

How many of the following systems have cyclic signal paths?



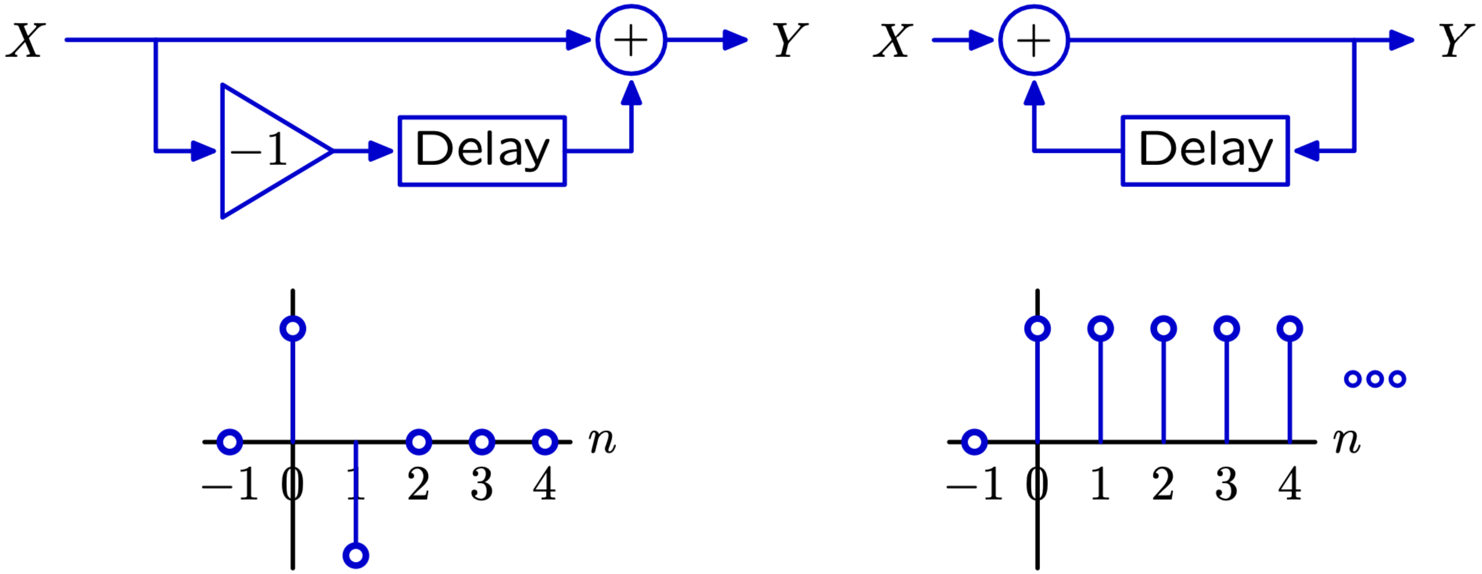
Check yourself

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

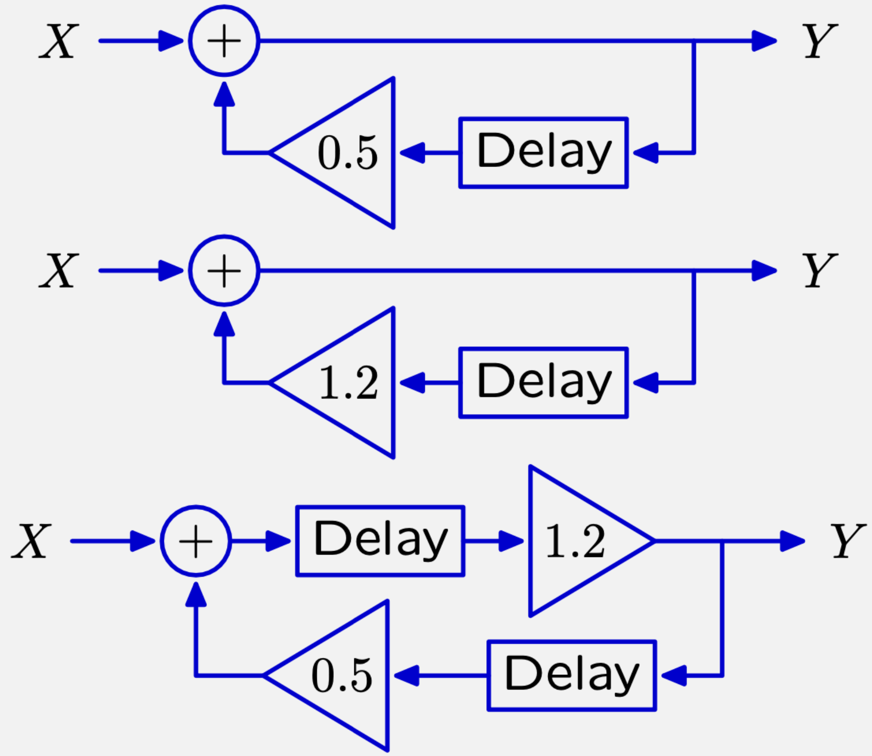
If traversing the cycle decreases or increases the magnitude of the signal, then the fundamental mode will decay or grow, respectively.

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

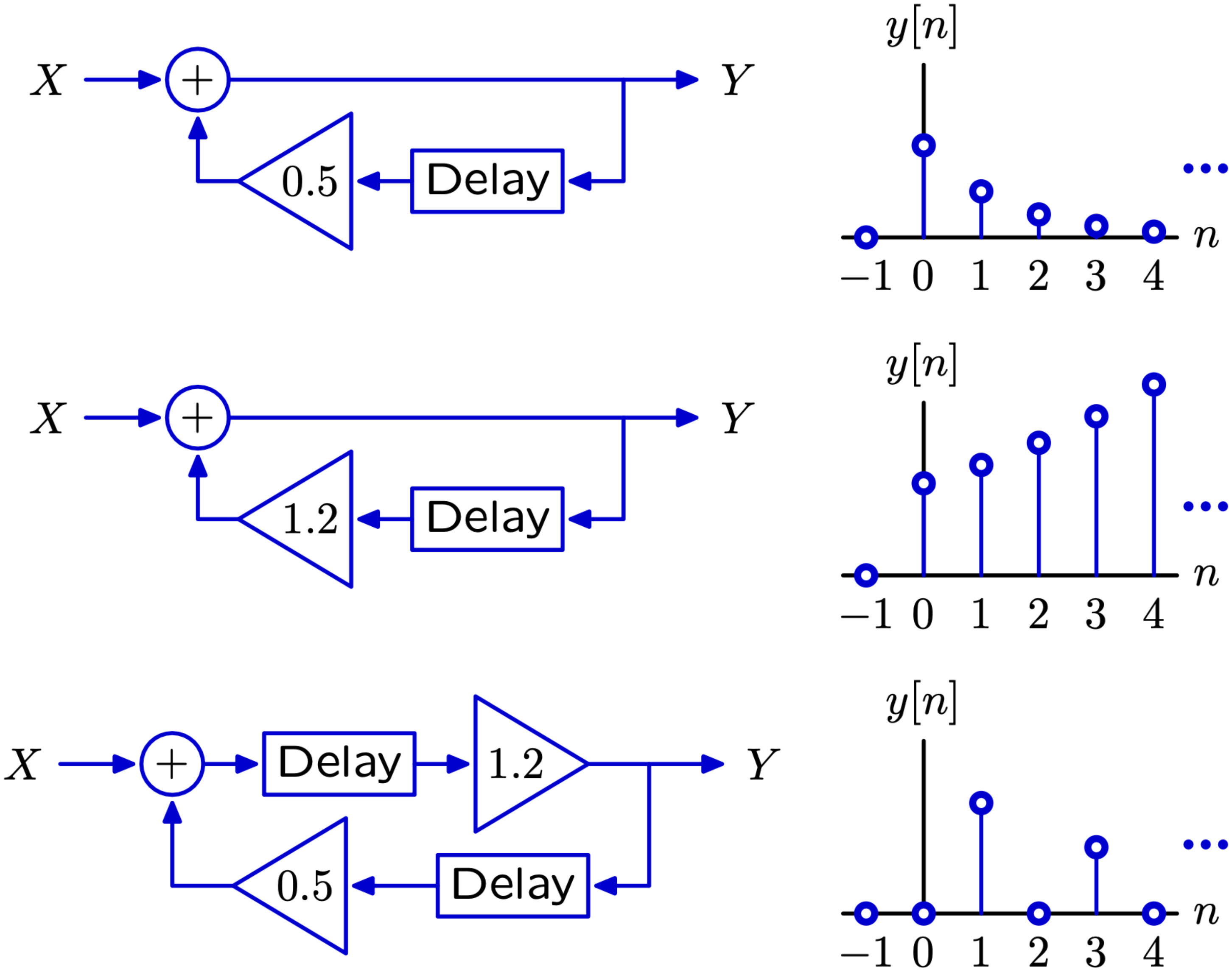
Otherwise, we it **diverges**.

Check yourself

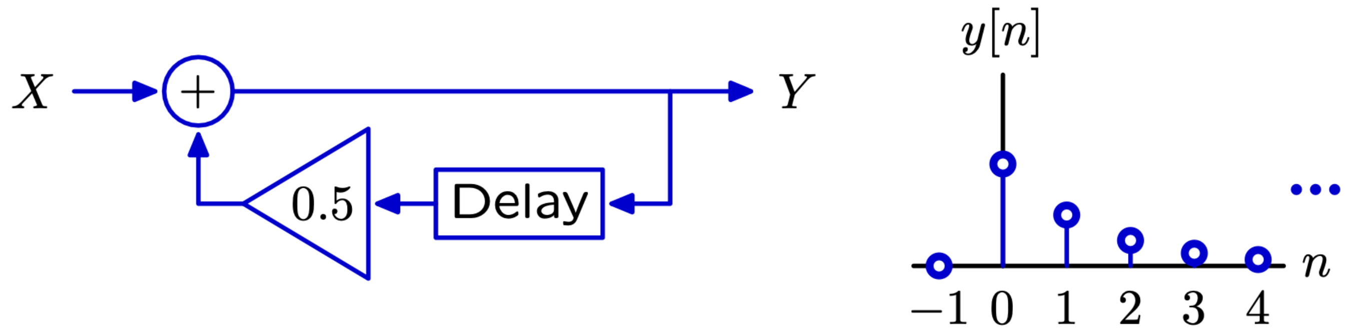
How many of these systems have divergent unit-sample responses?



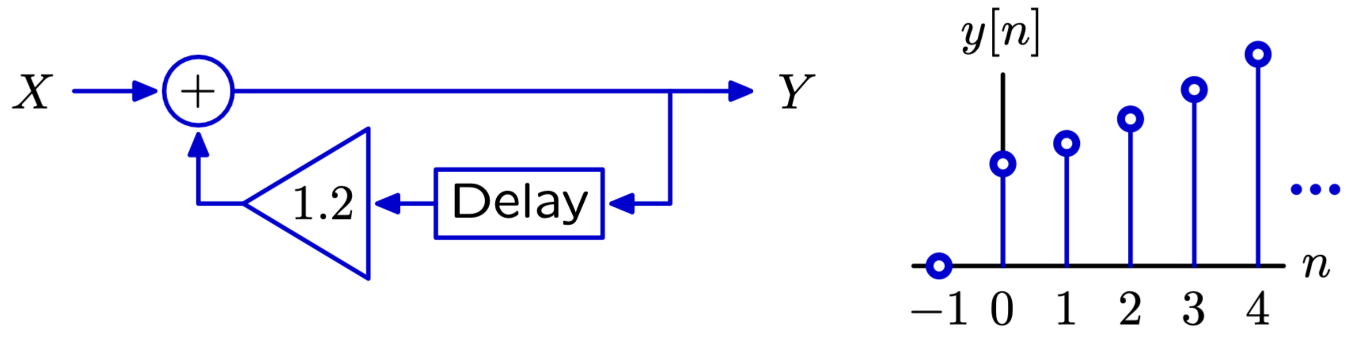
Check yourself



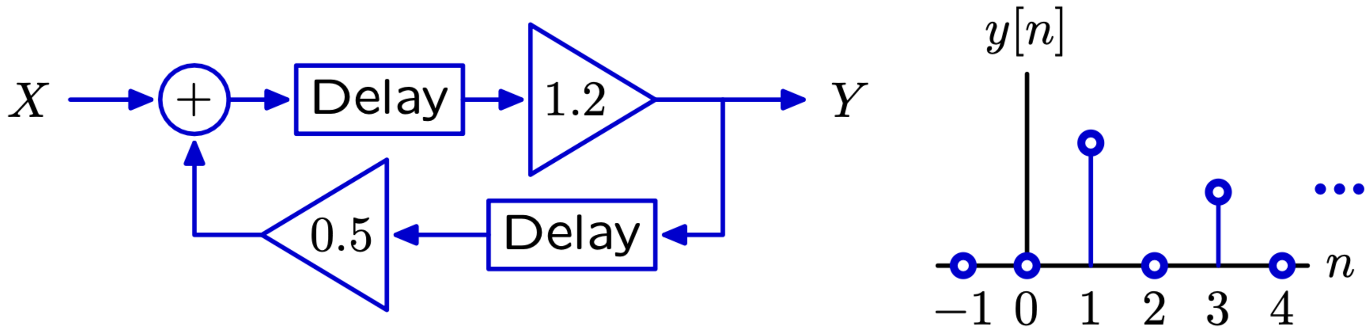
Check yourself



×



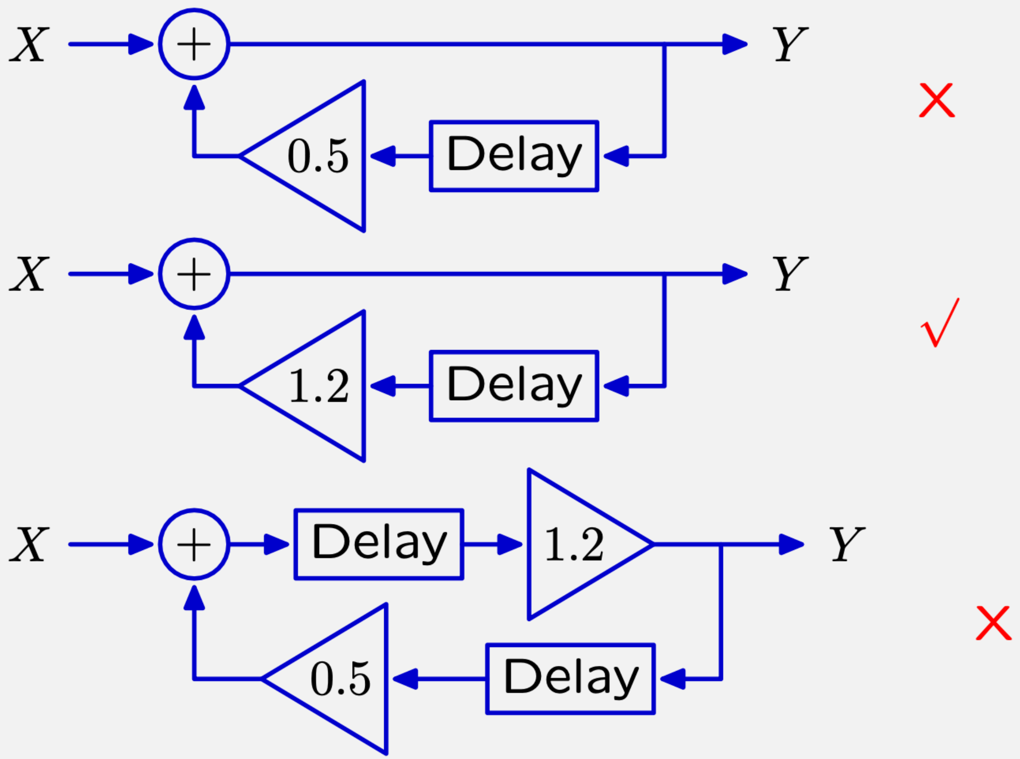
√



×

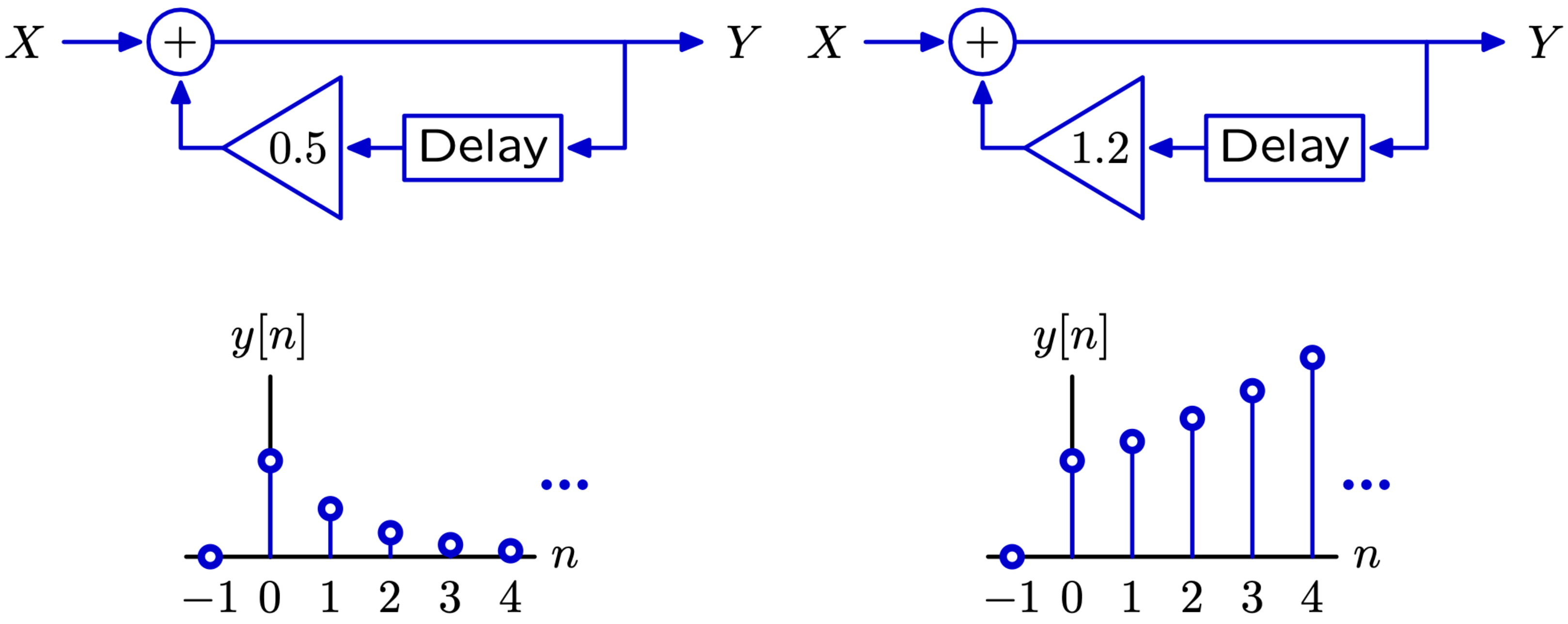
Check yourself

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 $n \geq 0$.

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

TU Delft

Multiple Representations of DTime Systems

Now you know four representations of discrete-time systems.

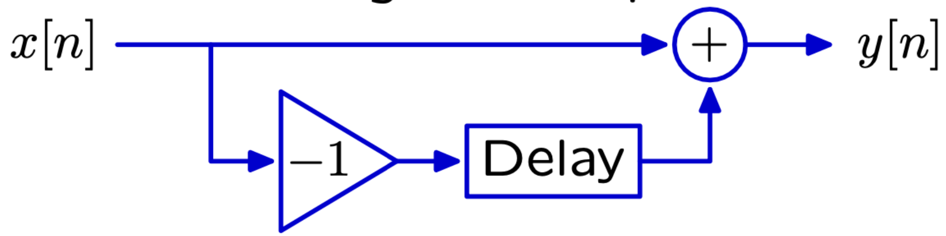
Verbal descriptions: preserve the rationale.

> "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.

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

TU Delft