EEO 401
Digital Signal Processing

Prof. Mark Fowler

Note Set #2

- DT Signals and Systems – Review
- Reading Assignment: Sect 2.1 & 2.2 of Proakis & Manolakis

Discrete-Time (D-T) Signals

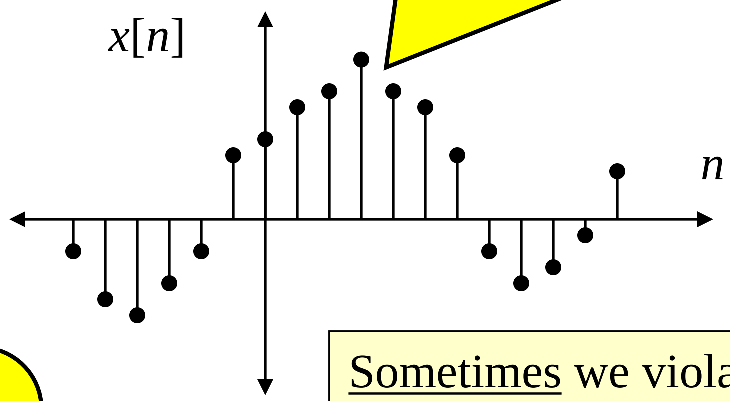
A discrete time signal is a sequence of numbers indexed by integers

Example: $x[n] \longrightarrow n = \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$

Brackets indicate discrete-time signal. Use parentheses to indicate a C-T signal. (Proakis & Manolakis don't do this!)

A stem plot emphasizes that the signal does not exist in-between integer n values

Remember: for our point of view, D-T signals are needed to allow us to process signals (i.e. information) using D-T systems rather than only Continuous-Time systems



Sometimes we violate this and plot with line segments connecting the dots.

See the book's description three alternative ways to represent a DT signal:

- Functional Representation
- Tabular Representation
- Sequence Representation

use ↑ to indicate which sample is at the time origin (n = 0)

Infinite Duration & Finite Duration D-T Signals

Most general case: the number of non-zero values in the signal is infinite.

"<u>Infinite Duration</u>" "Goes on forever"

"Doubly Infinite" $x[n] = \{\ldots, 4, 2, 0, 1, 3, \underset{\uparrow}{7}, 12, 5.3, 2.1, \ldots\}$

"Singly Infinite" $x[n] = \{\underset{\uparrow}{7}, 12, 5.3, 2.1, \ldots\}$ This means $x[n] = 0 \ \forall n < 0$

$x[n] = \{\ldots, 4, 2, 0, 1, 3, \underset{\uparrow}{7}\}$ This means $x[n] = 0 \ \forall n > 0$

Special case: the number of non-zero values in the signal is finite.

"**<u>Finite Duration</u>**" "Starts and Stops"

$x[n] = \{\underset{\uparrow}{7}, 12, 5.3, 2.1\}$ This means $x[n] = \begin{cases} 0 & \forall n < 0 \\ 0 & \forall n > 3 \end{cases}$

This is "a finite-duration sequence of length 4"

Some Elementary DT Signals

Most real-world signals are quite complicated looking… but there are some simple signals that are good for learning but also do show up sometimes in the real world!

"Unit Sample Sequence"

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

Also called "Unit Impulse"
or "Delta Function"
or "Delta Sequence"

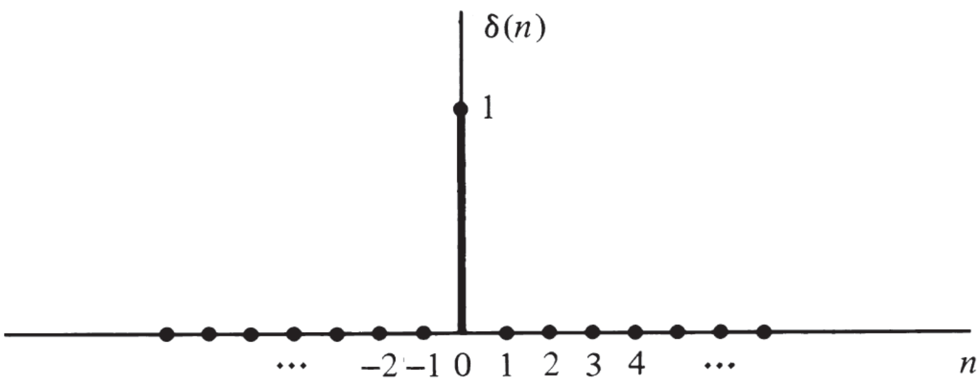


Figure 2.1.2 Graphical representation of the unit sample signal.

Two Important Properties

$$\sum_{n=-\infty}^{\infty} \delta[n] = 1$$

$$\sum_{n=-\infty}^{\infty} x[n]\delta[n-n_0] = x[n_0]$$

Sifting Property

"Unit Step Sequence"

$$u[n] = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$

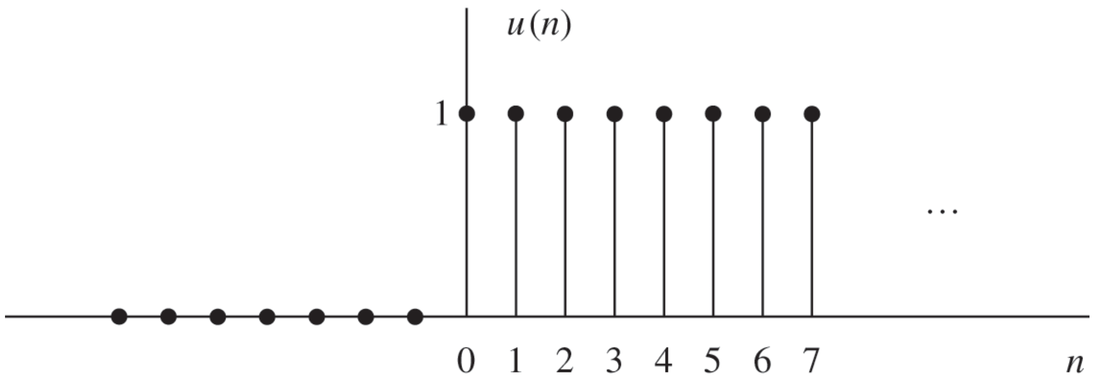


Figure 2.1.3 Graphical representation of the unit step signal.

"Unit Ramp Sequence"

$$u_r[n] \; or \; r[n]$$

$$u_r[n] = \begin{cases} 0, & n < 0 \\ n, & n \geq 0 \end{cases}$$

$$u_r[n] = nu[n]$$

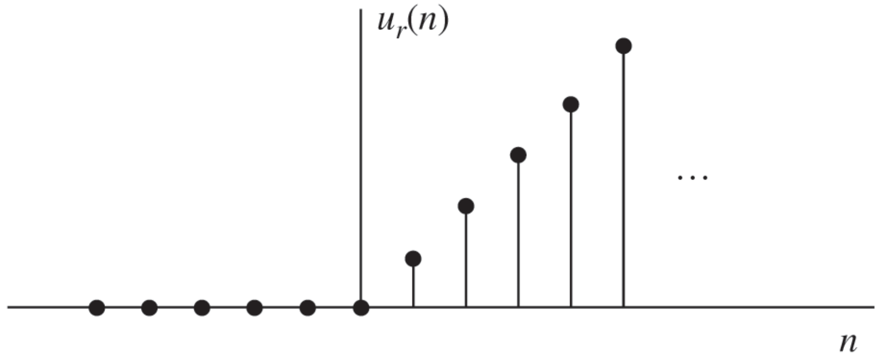


Figure 2.1.4 Graphical representation of the unit ramp signal.

"Real Exponential Signal"

$$x[n] = a^n \quad \forall n, a \in \mathbb{R}$$

Decays if $|a| < 1$ | Explodes if $|a| > 1$

Monotonic if $a > 0$

Oscillates if $a < 0$

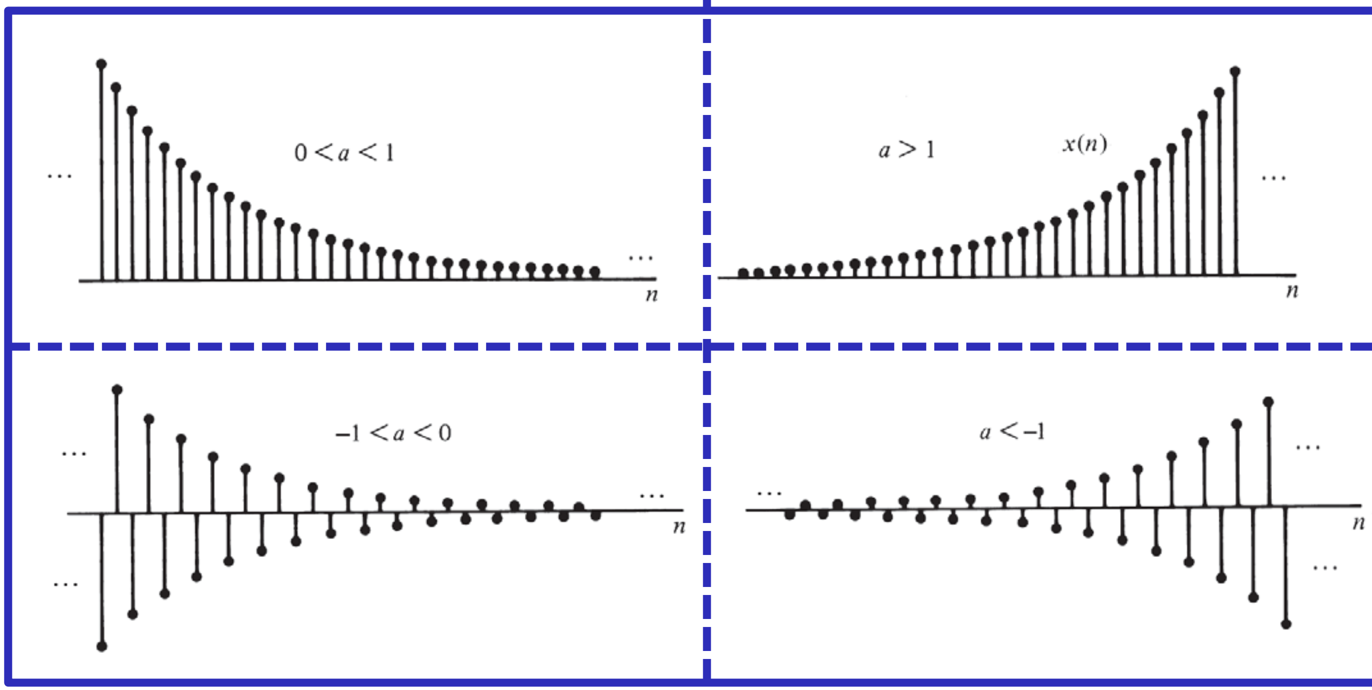


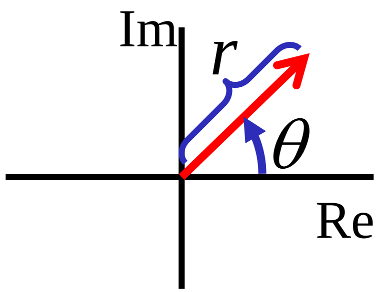
Figure 2.1.5 Graphical representation of exponential signals.

“Complex Exponential Signal”

Includes Complex Sinusoids ($r = 1$)

$$x[n] = a^n \quad \forall n, a \in \mathbb{C}$$

$$a = re^{j\theta}, r > 0$$



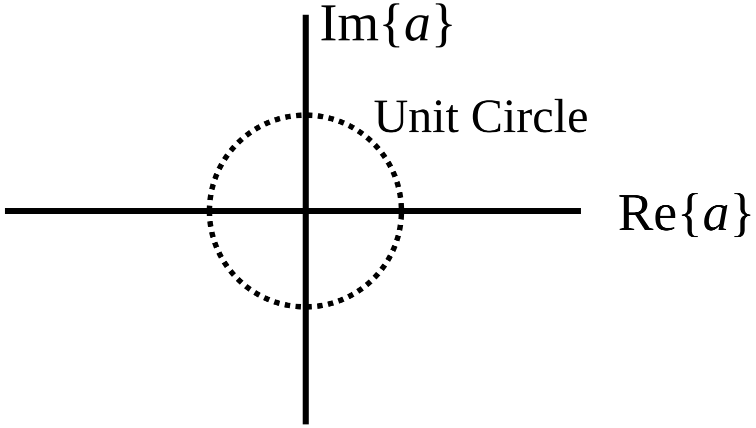
$$x[n] = a^n = \left(re^{j\theta}\right)^n = r^n e^{j\theta n} = r^n\left(\cos(\theta n) + j\sin(\theta n)\right)$$

Radius r controls “decay rate”

- $r < 1$ (inside UC) decays
- $r > 1$ (outside UC) “explodes”

Angle θ controls “oscillation rate”

- $\theta = 0$ (on positive Re) no oscillation
- $\theta \neq 0$ (off positive Re) oscillation



Classification of DT Signals

Many times what we can or can not do with a signal depends on some characteristic of that signal. Thus, it is helpful to classify DT signals into some typical subsets to help keep track of all this…

Some Categories of DT Signals

- **Infinite Duration vs Finite Duration**
  - We've seen that one already!
- **Energy Signals vs. Power Signals**
  - We'll discuss that next
- **Periodic Signals vs. Aperiodic Signals**
  - You should recall this from Signals & Systems (see our textbook)
- **Symmetric (even) Signals vs. Antisymmtric (odd) Signals**

Energy Signals vs. Power Signals

Physically, defining power and energy for DT signals is not that well defined… after all these are just a stream of numbers in a computer!

However, we can still do this from a mathematical sense that mimics the definitions used for physical CT signals.

Energy of DT Signal:

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

Obviously, $E > 0$ for any non-zero signal!

But, since we are adding (possibly) infinitely many positive values it is possible that for some signals the value of E is infinite.

This leads to our first classification:

A signal is said to be an ***Energy Signal*** if its energy is finite (i.e., $0 < E < \infty$)

Clearly for this to happen we need $|x[n]|$ to decay "fast enough" as $n \to \pm\infty$

Examples of Energy Signals
1. Any finite-duration signal
2. Decaying exponential: a^n, with $|a| < 1$

Examples of Non-Energy Signals
1. Sinusoid
2. Unit Step

For those signals that are NOT energy signals we can characterize their **Power**

Drawing from the *physical* idea that power is the rate of doing work (i.e., the amount of energy consumed/delivered per unit time) we can define similar ideas for DT signals.

Energy of DT Signal over finite interval of "length" $2N+1$: $E_N = \sum_{n=-N}^{N} |x[n]|^2$

Then… Power of a DT Signal is:

$$P = \lim_{N\to\infty} \frac{E_N}{2N+1} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2$$

Energy per 2N+1 samples

Note that what you get for P depends on a "race" between E_N and $2N+1$:

- If E_N grows but eventually "levels out" (then $x[n]$ is an energy signal) we get that $P = 0$. So only energy signals have $P = 0$.
- If E_N grows and never levels out, but does not grow faster than $2N+1$ then P is finite and nonzero. Not an energy signal, called power signal.
- If E_N grows and never levels out, but does grow faster than $2N+1$ then P is infinite. Neither energy nor power signal.

Classification of Signals on Energy & Power

$$E_N = \sum_{n=-N}^{N} |x[n]|^2$$

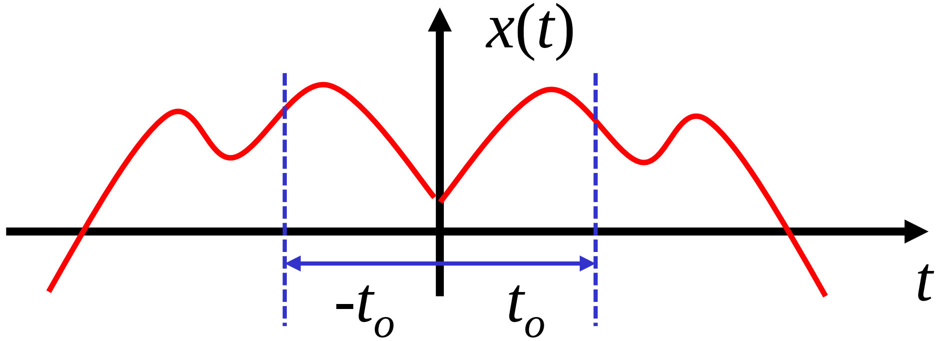
$$E = \lim_{N\to\infty} E_N$$

$$P = \lim_{N\to\infty} \frac{E_N}{2N+1}$$

Energy Signal	$E_N \rightarrow E < \infty$	$P = 0$
Power Signal	$E_N \rightarrow \infty$ but not faster than $2N+1$	$0 < P < \infty$
Neither	$E_N \rightarrow \infty$ but faster than $2N+1$	$P \rightarrow \infty$

Symmetric (Even) vs. Anti-Symmetric (Odd)

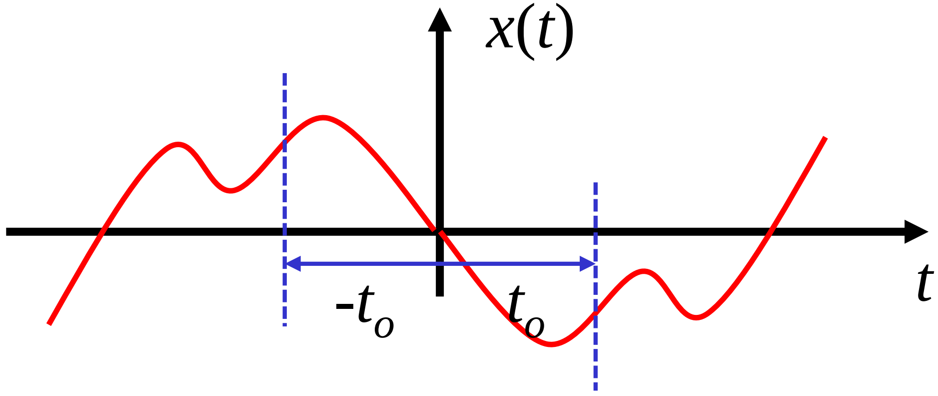
A DT signal is symmetric (even symmetric) if $x[-n] = x[n]$



Illustrated here using CT because it is easier to draw!

Same value for all t_o

A DT signal is anti-symmetric (odd symmetric) if $x[-n] = -x[n]$



Negative values for all t_o

Note what happens at $n = 0$…

$$x[-0] = -x[0]$$

$$\Rightarrow \quad x[0] = -x[0]$$

That can only be true if $x[0] = 0$!!!

Why do we care about symmetry??? Because in Fourier analysis of signals we are decomposing a signal into sines (odd) and cosines (even) and it turns out that if the signal is even (odd) then we can build it using only cosines (sines).

So if there is some form of symmetry we can exploit it to make our analysis job easier!!

Even we have a general signal we can sometimes make use of the simplifications that come from symmetry:

Any signal $x[n]$ can be broken into a sum of an even part % odd part like this

$$x[n] = x_e[n] + x_o[n]$$

$$x_e[n] = \frac{1}{2}\big[x[n] + x[-n]\big] \qquad\qquad x_o[n] = \frac{1}{2}\big[x[n] - x[-n]\big]$$

Clearly, these two add to $x[n]$… and it is easy to verify they are even and odd.

Simple Manipulation of DT Signals

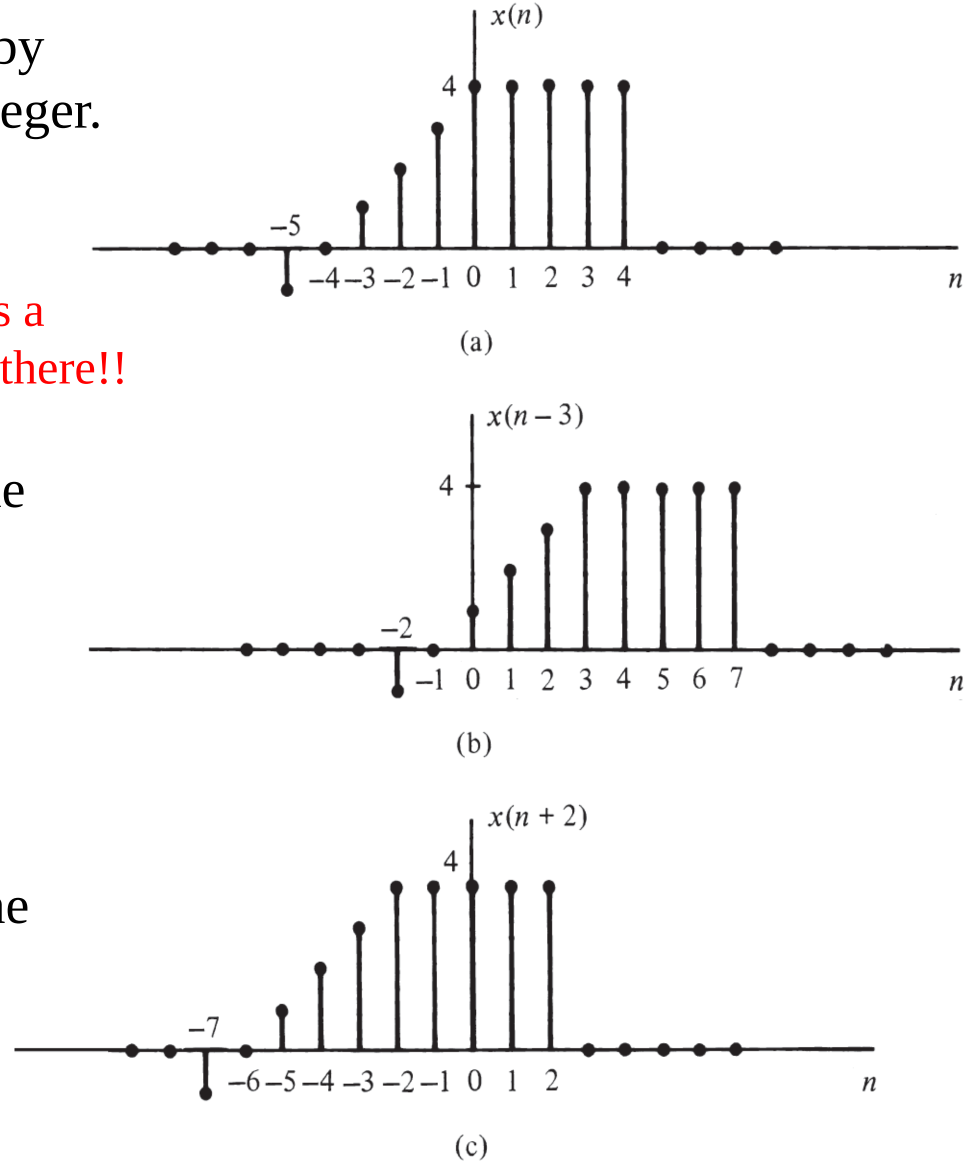
Time Shift

A given signal $x[n]$ is shifted in time by replacing n by $n - k$, where k is an integer.

Don't forget there is a negative already in there!!

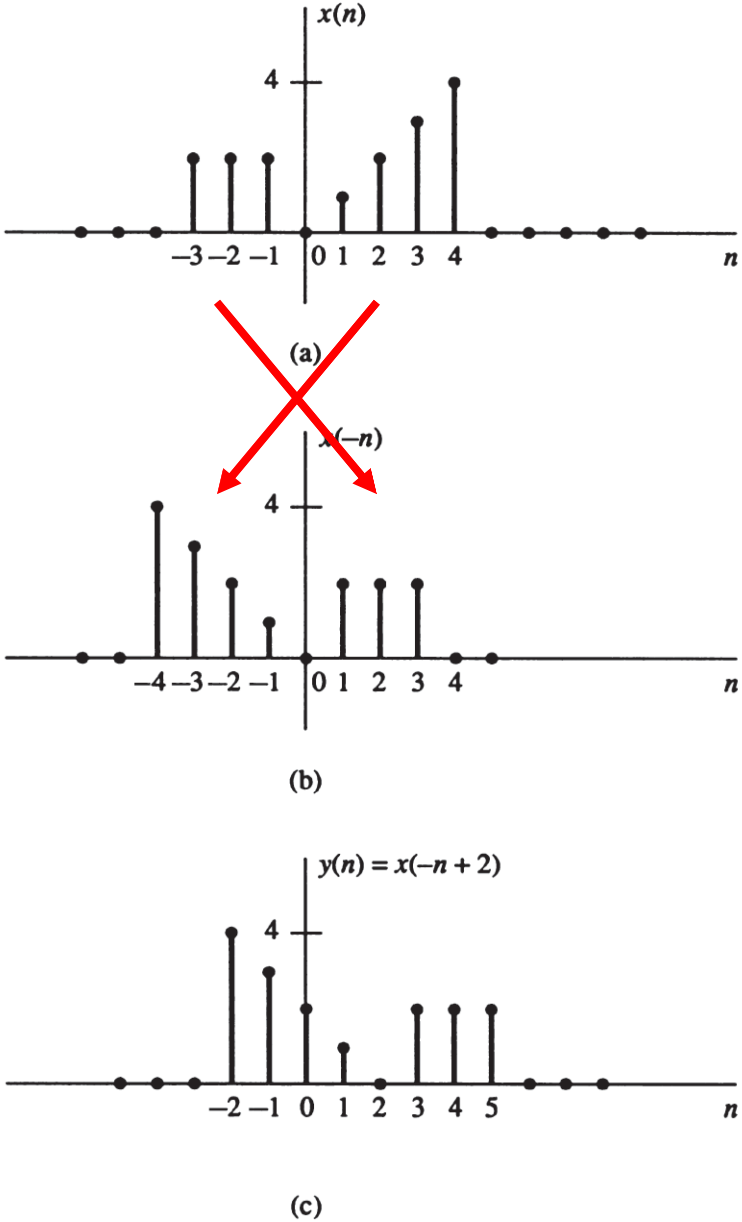
- If k is positive then the shift is to the right… which is a "delay"

- If k is negative then the shift is to the left… which is an "advance"



Time Reversal

A given signal $x[n]$ is reversed in time by replacing n by $-n$.



- Time reversal just "anchors" the signal at $n = 0$ and flips it

- Flipping and shifting can be combined… Careful here: Do the shift first then the flip.

$$x[n] \rightarrow \underbrace{x[n+2]}_{\text{Shift left by 2}} \rightarrow \underbrace{x[-n+2]}_{\text{Flip the shifted signal}}$$

Time Scaling

A given signal $x[n]$ is scaled in time by replacing n by μn, where μ must be an integer.

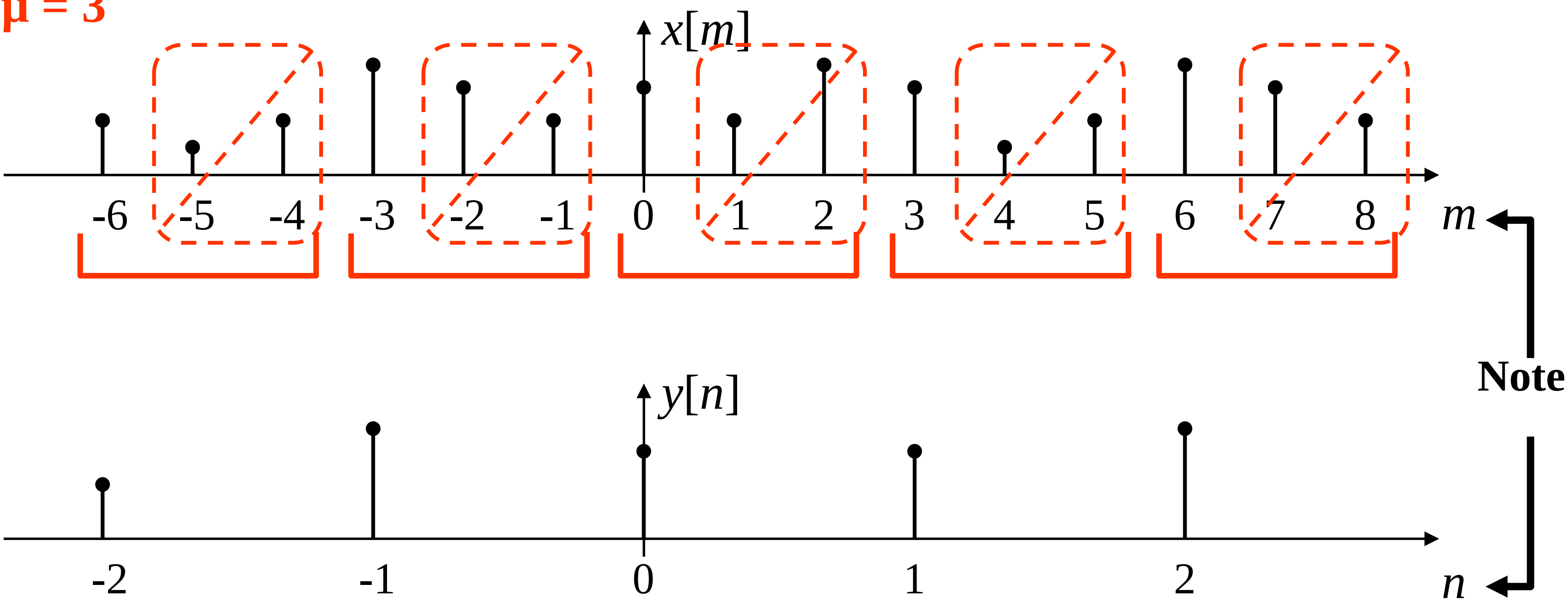
$$y[n] = x[\mu n]$$

An easy way to help visualize this is to let $m = \mu n$, let n run through the integers and find the resulting values of m… those are the indices of the samples you'll have "keep"

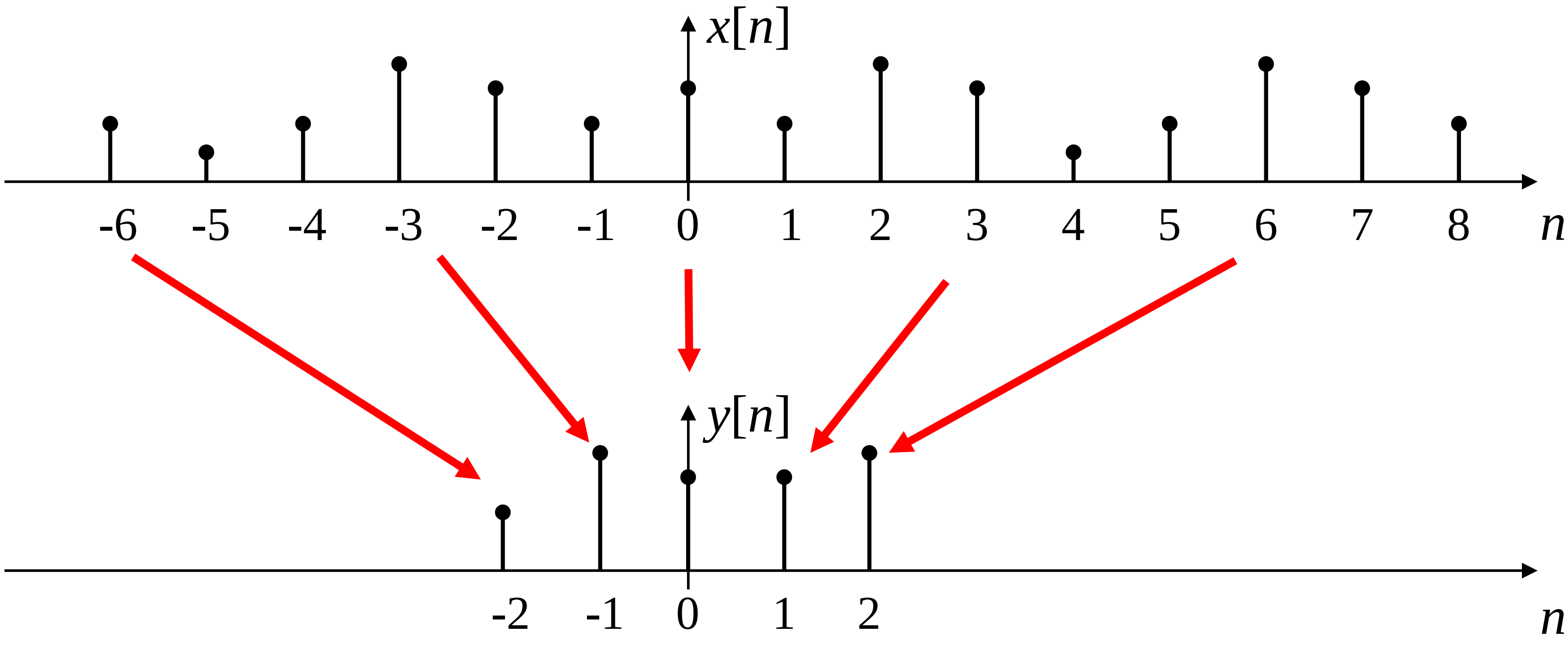
You can see why this process is also called "Down Sampling"!

n:	0			1			2			3	…
m:	0	1	2	3	4	5	6	7	8	9	…

μ = 3



Then…. We re-imagine the new signal on the same axis as the original:



Addition, Multiplication & Amplitude Scaling

Adding two sequences, multiplying two sequences, or multiplying a sequence by a number are defined in the obvious way.

Input-Output Description of Systems

Systems, by their very nature, simply map each given input sequence into a corresponding output sequence.

Thus… the simplest viewpoint of describing (or modeling) systems is to provide a mathematical description of how the system creates the output from the input.

Some Examples:

"Unit Delay System" $y[n] = x[n-1]$

"Moving Average Filter" $y[n] = \frac{1}{3}\left[x[n] + x[n-1] + x[n-2]\right]$

"Accumulator" $y[n] = \sum_{k=-\infty}^{n} x[k]$

Note that we can re-write the Accumulator as $y[n] = y[n-1] + x[n]$

which describes the system using a recursive equation (an example of the general class of difference equations).

Difference Equations

A general N^{th} order Difference Equations looks like this:

$$y[n] + a_1 y[n-1] + \ldots + a_N y[n-N] = b_0 x[n] + b_1 x[n-1] + \ldots + b_M x[n-M]$$

Most "Advanced" Output Sample

Least "Advanced" Output Sample

The difference between these two index values is the "order" of the difference eq. Here we have: $n - (n - N) = N$

Can Write As:

$$y[n] + \sum_{i=1}^{N} a_i y[n-i] = \sum_{i=0}^{M} b_i x[n-i]$$

Now… isolating the $y[n]$ term gives the "Recursive Form":

$$y[n] = -\sum_{i=1}^{N} a_i y[n-i] + \sum_{i=0}^{M} b_i x[n-i]$$

Given the input values and N "initial" output values we can solve this recursively.

"current" output value to be computed

Some "past" output values, with values already known

current & past input values already "received"

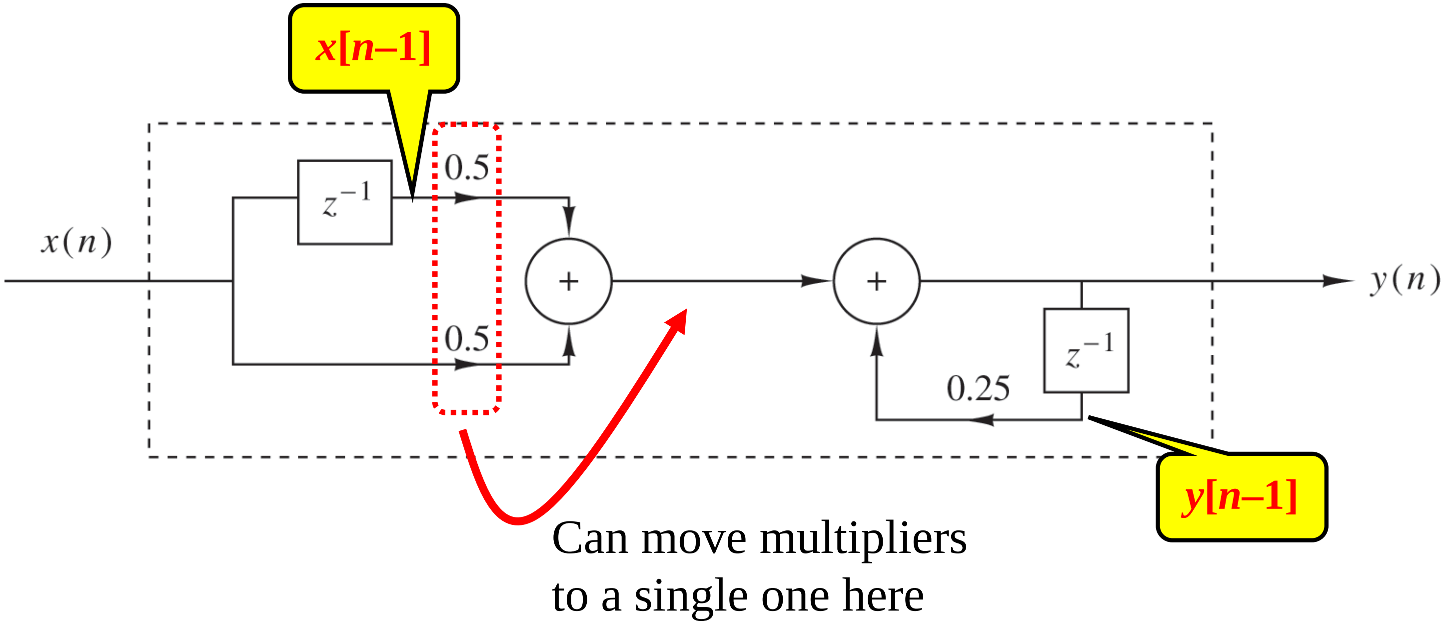
Block Diagrams for DT Systems

We use a few simple blocks to allow us to "build" representations of DT systems:

- Adder – adds two signals
- Constant Multiplier – multiplies signal by a constant
- Signals Multiplier – multiplies one signal by the other
- Unit Delay – Outputs a one-sample delayed version of its input
- Unit Advance – Outputs a one-sample advanced version of its input

Example: System described by Difference Equation

$$y[n] = 0.25y[n-1] + 0.5x[n] + 0.5x[n-1]$$



Classification of DT Systems

Static (memoryless) vs. Dynamic (has memory)

Static system's output at any time *n* depends at most on the input sample at that instant… but no input samples from past or future.

Otherwise it is a Dynamic system. Because its output depends on past input sample values the system in essence must have some sort of "memory" to store these past value.

Caution… If it depends on past *output samples* then it also depends on past input samples!

If you never need more than $N < \infty$ past input values... then it is Finite Memory.
If you need past inputs infinitely far into the past… then it is Infinite Memory.

Static

$$y[n] = ax[n]$$

$$y[n] = nx[n] + bx^3[n]$$

Dynamic

$$y[n] = x[n] + 3x[n-1]$$

Finite, Fixed Memory

$$y[n] = \sum_{k=0}^{n} x[n-k]$$

Finite, Growing Memory

$$y[n] = \sum_{k=0}^{\infty} x[n-k]$$

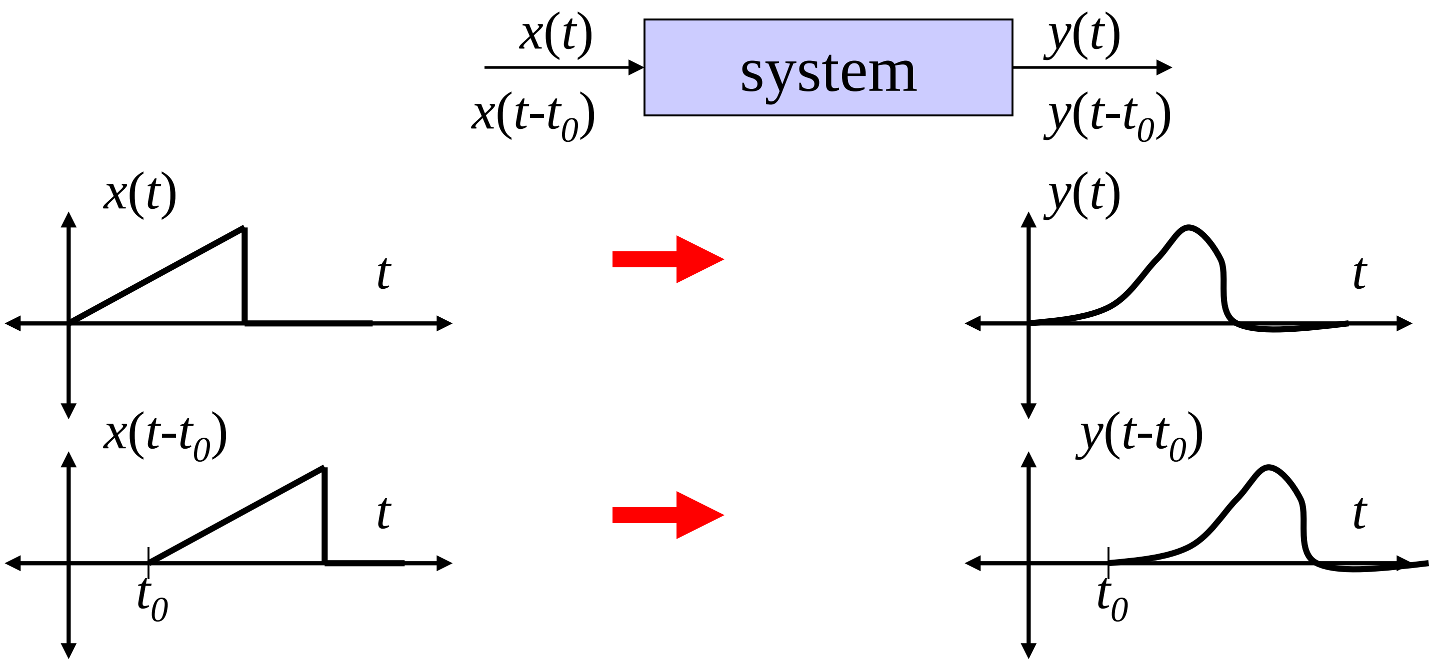
Infinite Memory

Time-Invariant vs. Time-Variant Systems

A Time-Invariant system is one for which the structure of its Input-Output relationship does not change with time.

That means that if you put a signal in later, the output will be the same shape just will come out later.

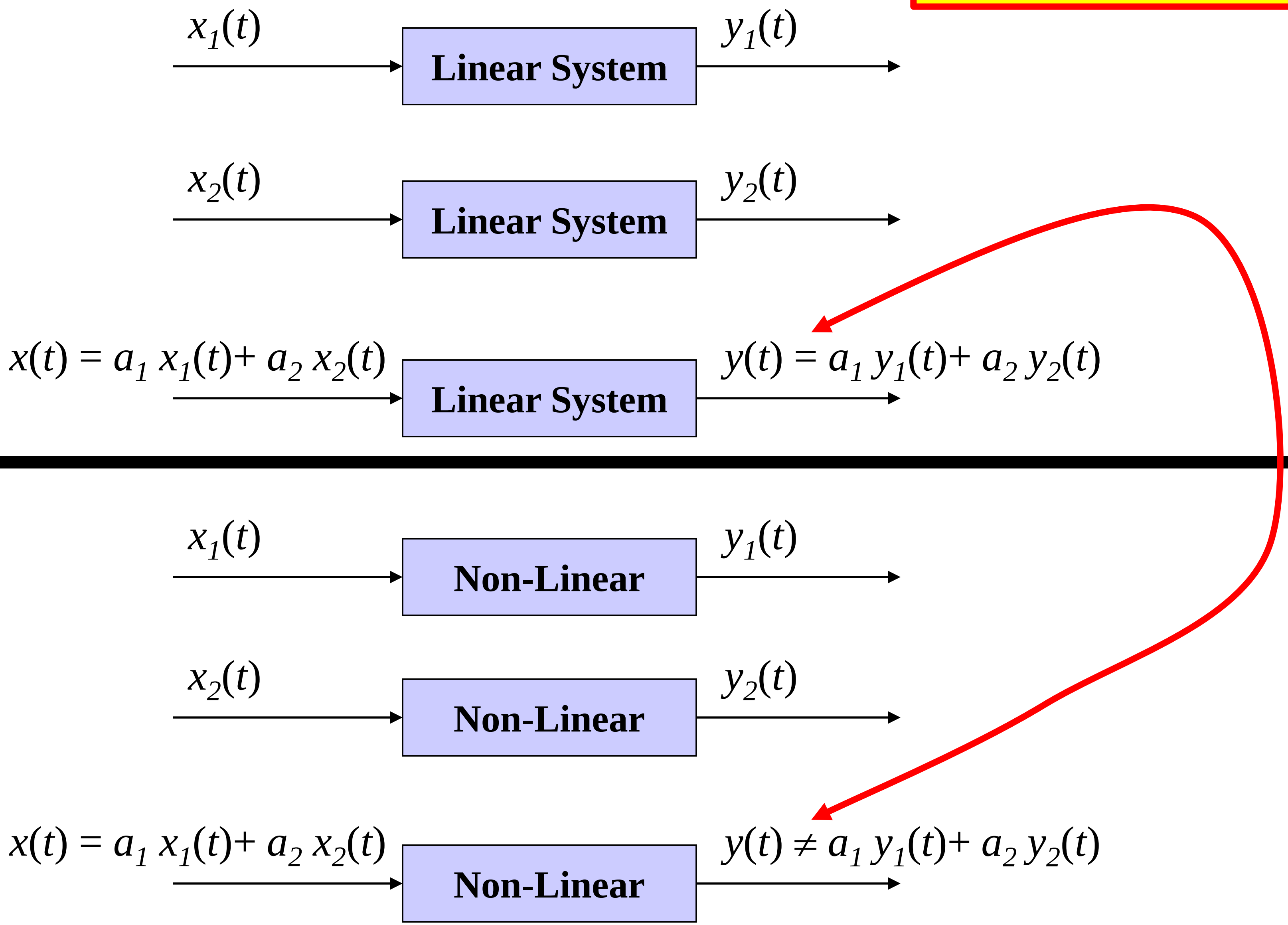
Shown here for CT… DT is similar!



Linear vs. Non-Linear Systems

A system is linear if superposition holds:

A system that is Linear & Time-Invariant is called an **LTI System**

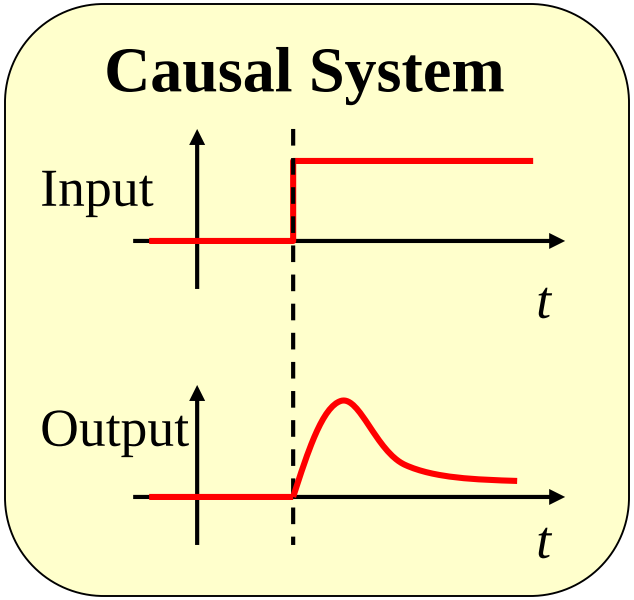


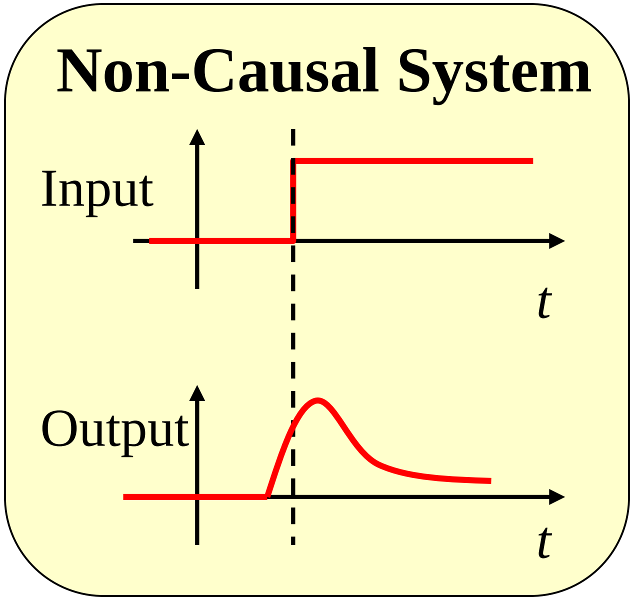
Causal vs. Non-Causal

A causal (or non-anticipatory) system's output at a time t_1 does not depend on values of the input $x(t)$ for $t > t_1$

The "future input" cannot impact the "now output"

⇒ A Causal system (with zero initial conditions) cannot have a non-zero output until a non-zero input is applied.





"Real-time" systems must be causal…. But time-signals recorded processed off-line can in essence be non-causal. Images can be processed "non-causally".

Stable vs. Unstable

A system is said to be bounded input – bounded output stable if and only if <u>every</u> bounded input produces a bounded output.

A bounded signal means that there exists some finite number M_x such that the signal's absolute value never exceeds it.

$$|x[n]| \le M_x < \infty \quad \forall n$$

<u>Example of an Unstable System</u>

The "accumulator" is an unstable system

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

Note: to prove that it is unstable we need only find *one* bounded input that gives an unbounded output. If the input is the unit step $u[n]$, which is a bounded input, then the output equals

$$y[n] = \sum_{k=0}^{n} 1 = (n+1)$$

which grows without bound so the output is unbounded.