Perception & Multimedia Computing

Week 16 – Implementing and Understanding effects, Using signals & systems

Michael Zbyszyński
Lecturer, Department of Computing
Goldsmiths University of London

Last time...

- Musical expectation
- Rhythm
- Why signals & systems?
- Linearity
- Time invariance

Today

Signals & Systems

- Reasoning about audio/image
- Applying operations
- What is a signal? System?
- Real-world types
- Special types
- Using impulse response to reason about LTI systems

What is a Signal?

- Any quantity varying over space or time
- Maths: *function* Physics: *field*
    - *Electrical current at a point in a circuit*
    - *Count of students in weekly lectures*
    - *Temperature at all points in room*

1-dimensional, discrete-time signals in Python

```
x = [1,2,3]
t = np.arange(0, 1, 1/44100)
s = sin(2*pi*100*t)
```

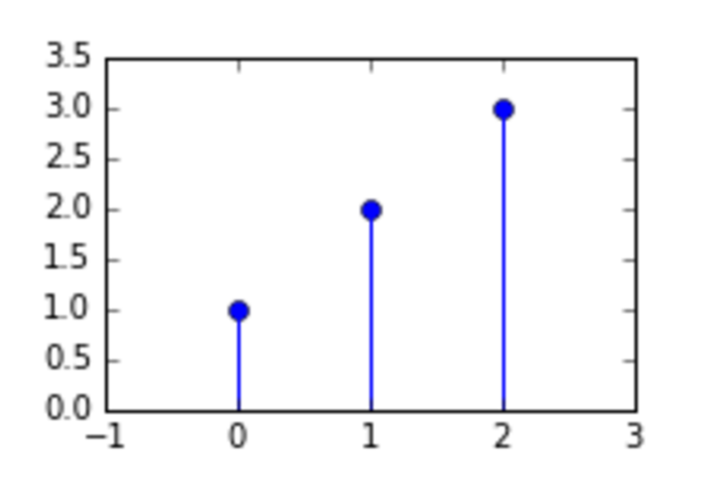
These are just arrays!

We interpret each element as happening at a certain time

e.g., s[10] happens 10 samples after “time 0”

```
x = [1,2,3]
print x
fig, axes = plt.subplots(figsize=(3,2))
axes.stem(x)
ylim([0, 3.5])
plt.xticks(np.arange(-1, 4, 1.0))
```

```
[1, 2, 3]
```



Signal notation

These are all equivalent:

- x[n] = [3, 5, 12]
- x[0] = 3, x[1] = 5, x[2] = 12
- x[n] = [3, 5, 12, 0, 0, 0, ….]

and x[-1] = 0, x[-100] = 0, etc.

A signal is like an array:

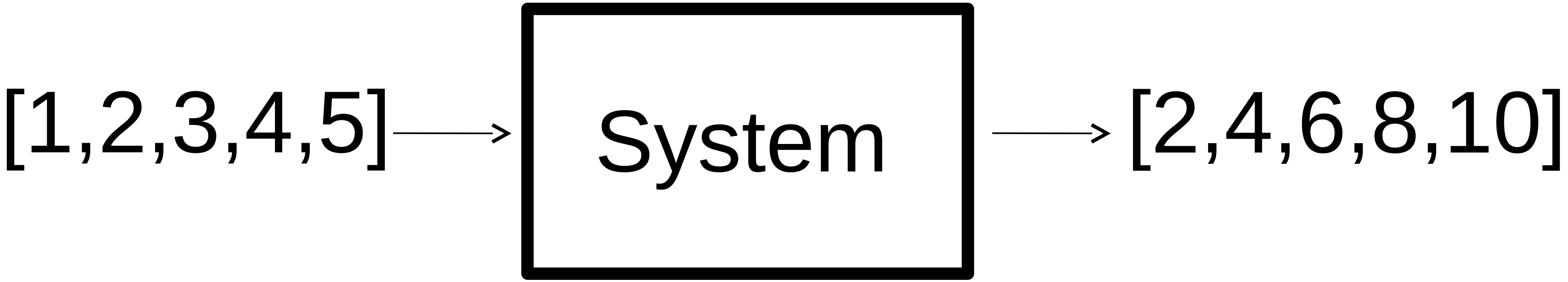
- It is an ordered list of values
- It specifies the value of a signal at sample points in time(starting from t=0, so time is like an index)

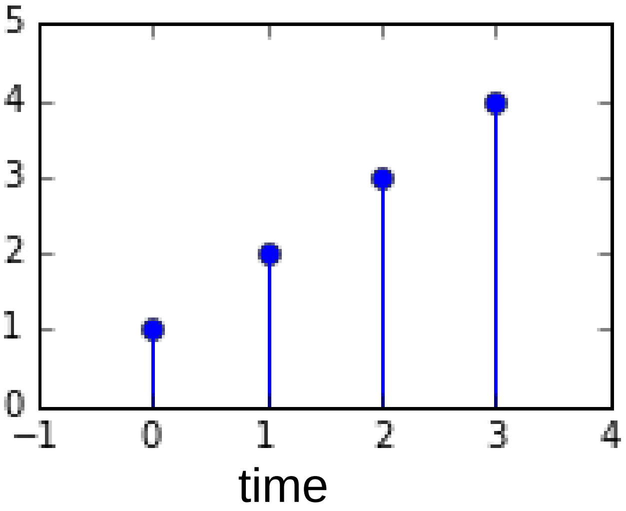
But different:

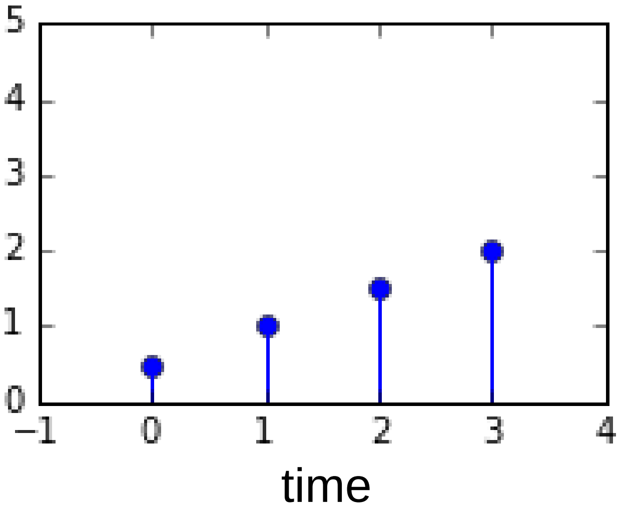
- It’s not code
- Its value is 0 at all points (including negative times.

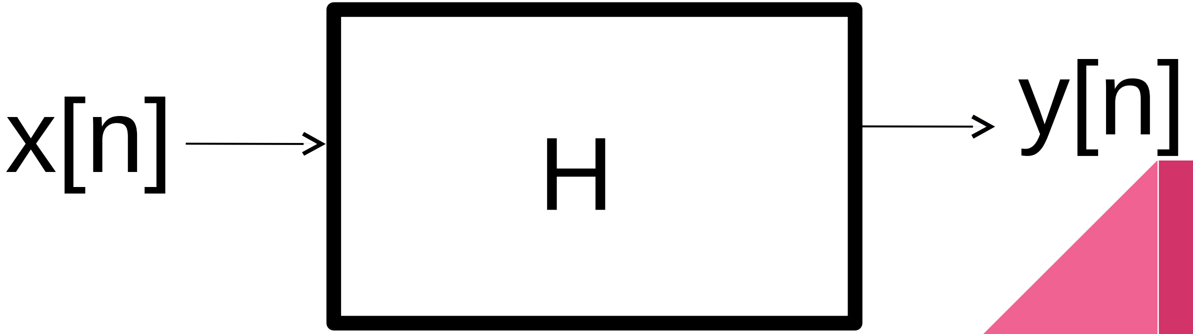
Systems

Compute output signal (response) from input




0
1
2
3
4
5
−1
0
1
2
3
4
time


0
1
2
3
4
5
−1
0
1
2
3
4
time


x[n]
H
y[n]

Examples of systems used in multimedia?

The magical system-building genie

I wish I could build a filter/effect that changes a sound in a specific way…

I wish I could replicate an existing effect in my own software...

I wish I could reason about how this particular effect will alter a new sound...

Done!!!

(If you meet just a few conditions, then follow my instructions)

What are the conditions?

Systems have to be

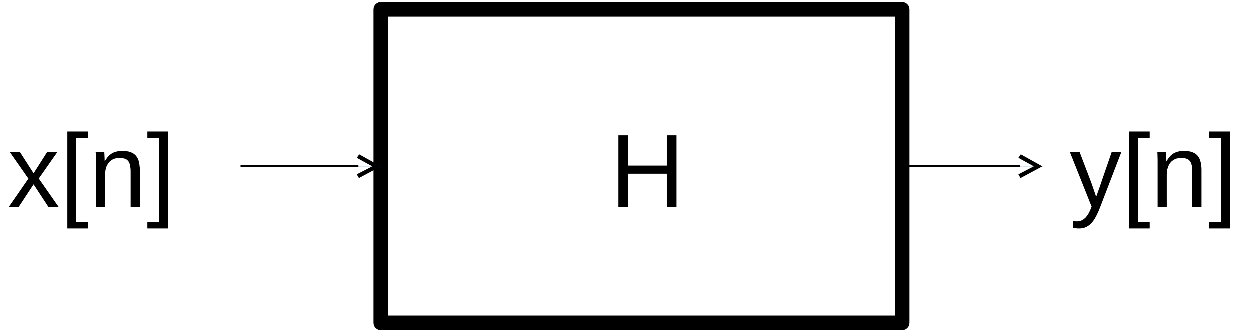
1. Linear
2. Time-invariant

(Or for images, shift-invariant)

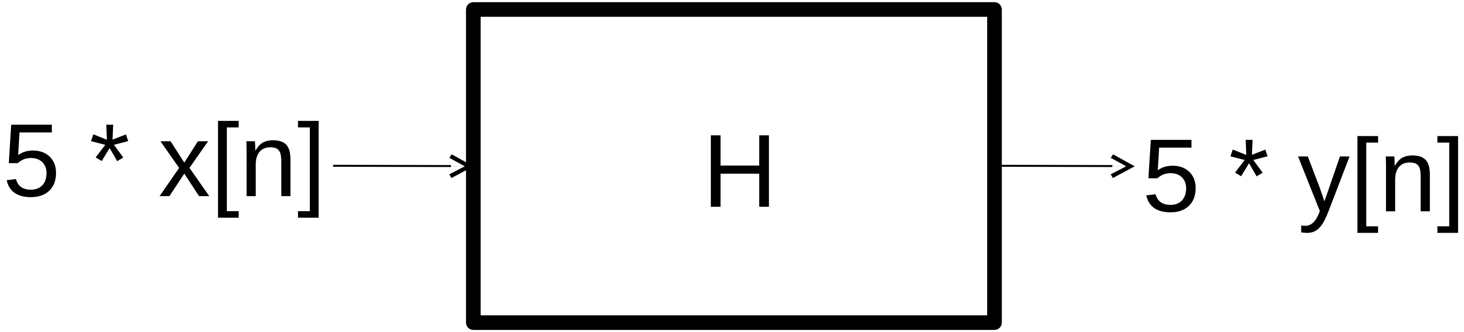
Linearity

Linear systems have the property that *scaling* and *superposition* of their input signals lead to a corresponding scaled superposition of their outputs.

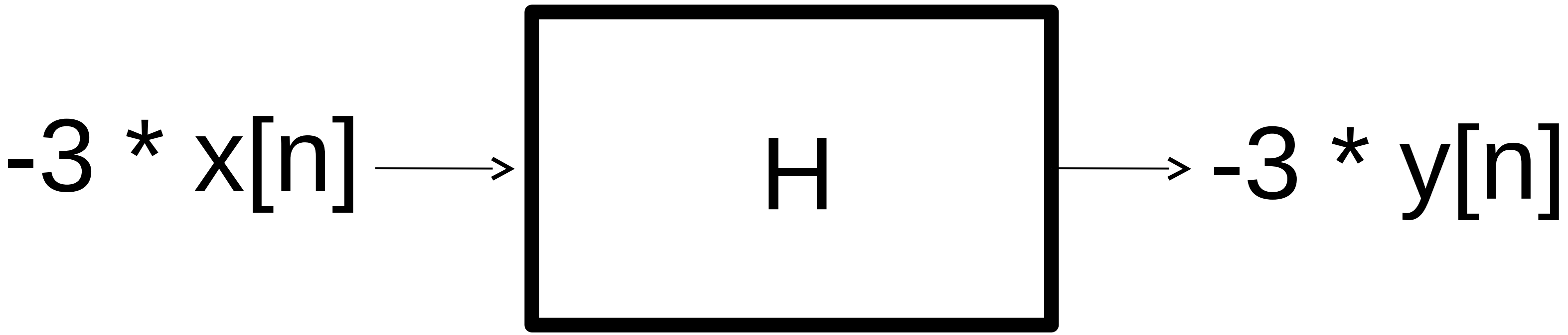
Scaling



Scaling



Scaling



Example

Scaling: for any input signal x_1 where $y_1 = H\{x_1\}$,

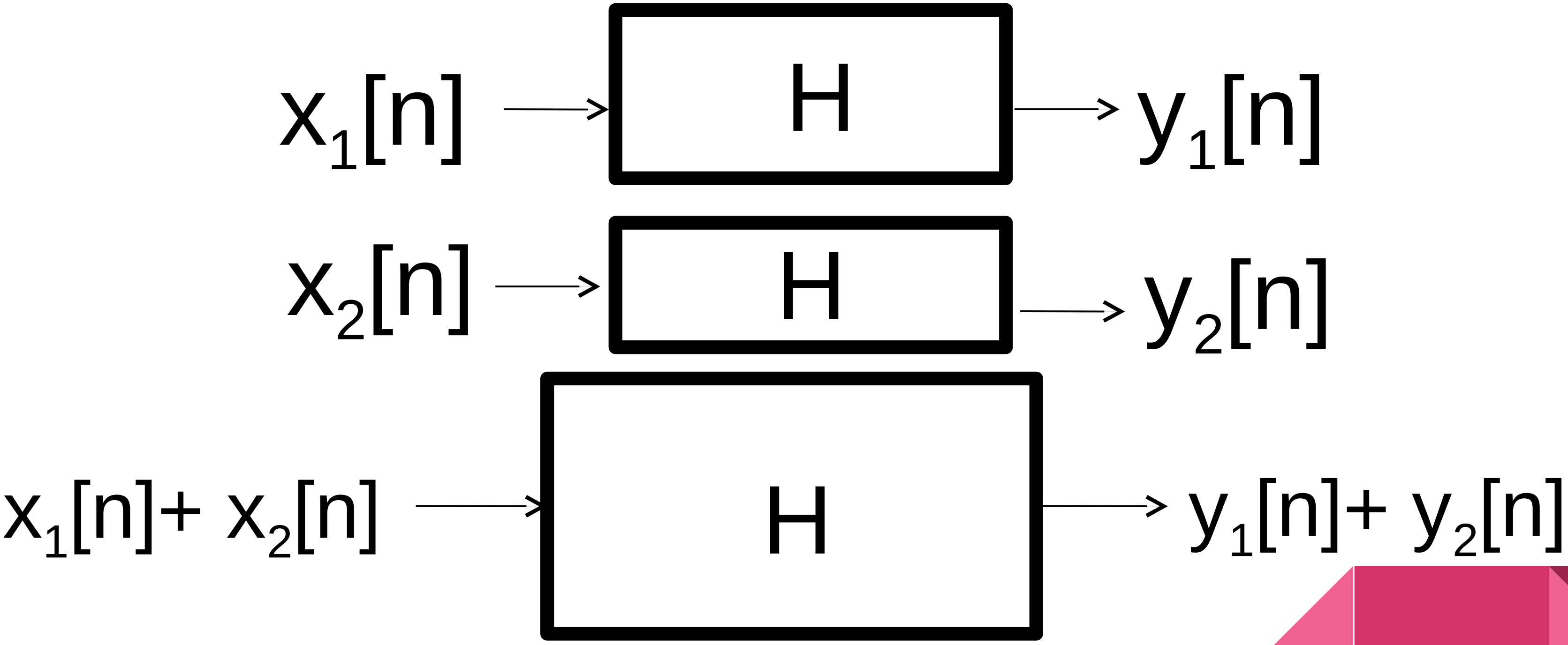
$H\{\alpha x_1\} = \alpha y_1$ for any scalar value α

x1 = [2, 4, 6]		y1 = H{x1} = [3, 18, 4]
x2 = [4, 8, 12]		y2 = H{x2} = ?
x1 = 2 * x1	so	y2 = 2* y1

Linearity

Linear systems have the property that *scaling* and *superposition* of their input signals lead to a corresponding scaled superposition of their outputs.

Superposition



Example

Superposition: for any two input signals x_1 and x_2

where $y_1 = H\{x_1\}$ and $y_2 = H\{x_2\}$, $H\{x_1 + x_2\} = y_1 + y_2$

x1 = [2, 4, 6]
x2 = [1, 5]
X3 = [5, 2, 4]
x3 = x1 + x2

y1 = H{x1} = [3, 18, 4]
y2 = H{x2} = [1, -3, 5]
y3 = ???
so y3 = y1 + y2

y3 = [4, 13, 7]

Linearity

Linear systems have the property that *scaling* and *superposition* of their input signals lead to a corresponding scaled superposition of their outputs.

Example

x1 = [5]
x2 = [0, 1, 2]
x3 = [5, 2, 4]

y1 = H{x1} = [-5]
y2 = H{x2} = [0, 6, 3]
y3 = ?

x3 = x1 + 2*x2 so y3 = y1 + 2*y2

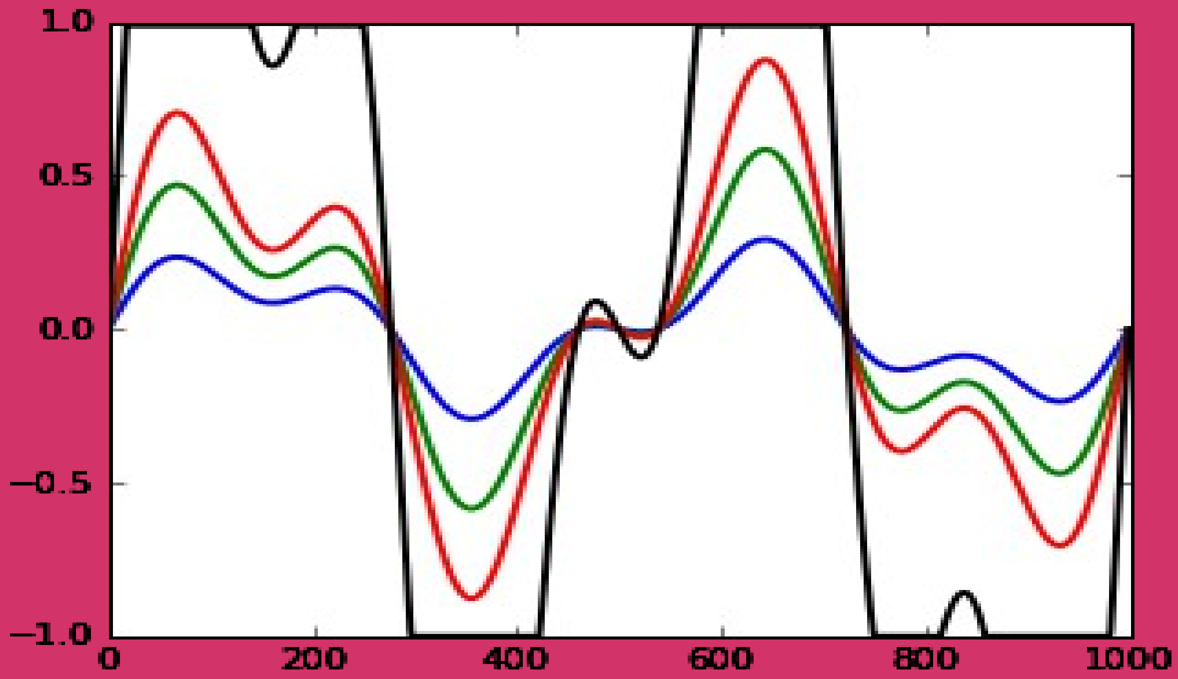
y3 = [-5, 12, 6]

Linearity

Almost all systems in the real world are linear systems for *small enough signals*.

e.g. audio signal that doesn’t clip

Nonlinear: Volume clipping



Time-invariance

Time-invariant systems have the property that the output signal of the system for a given input signal does not depend explicitly on absolute time.

Time-invariance

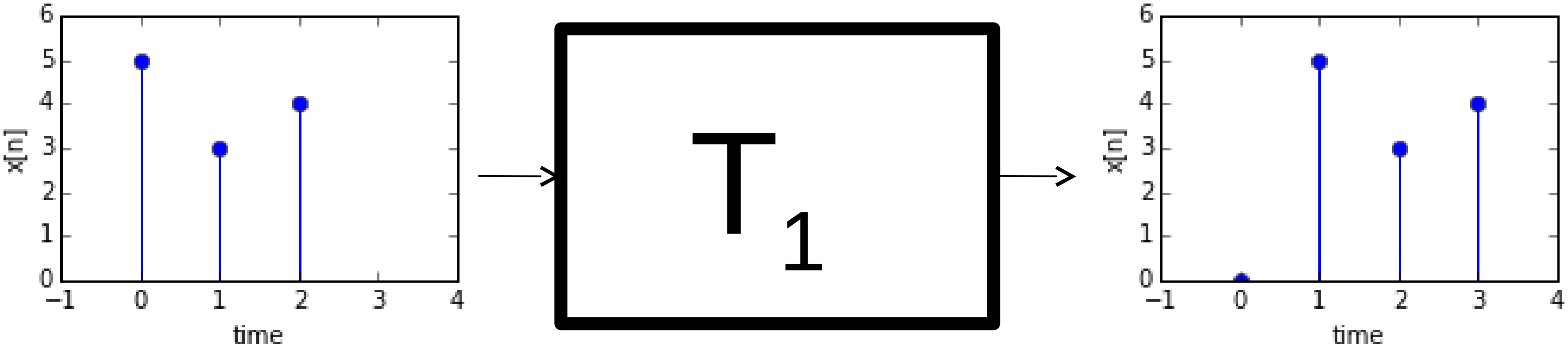
E.g., does your guitar amp affect your guitar differently on Monday than on Tuesday?

Does your image filter blur your image differently at 12:00 than at 11:00?

Probably not: These are time-invariant!

Unit Delay System

Just delays signal by 1 sample



The Unit Delay System

A special linear system T_1 whose output signal is the input signal, but delayed by one time unit.

$$y = T_1\{x\}$$

Or with discrete time explicitly represented:

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

This unit delay system is the building block of the systems we will cover in this course.

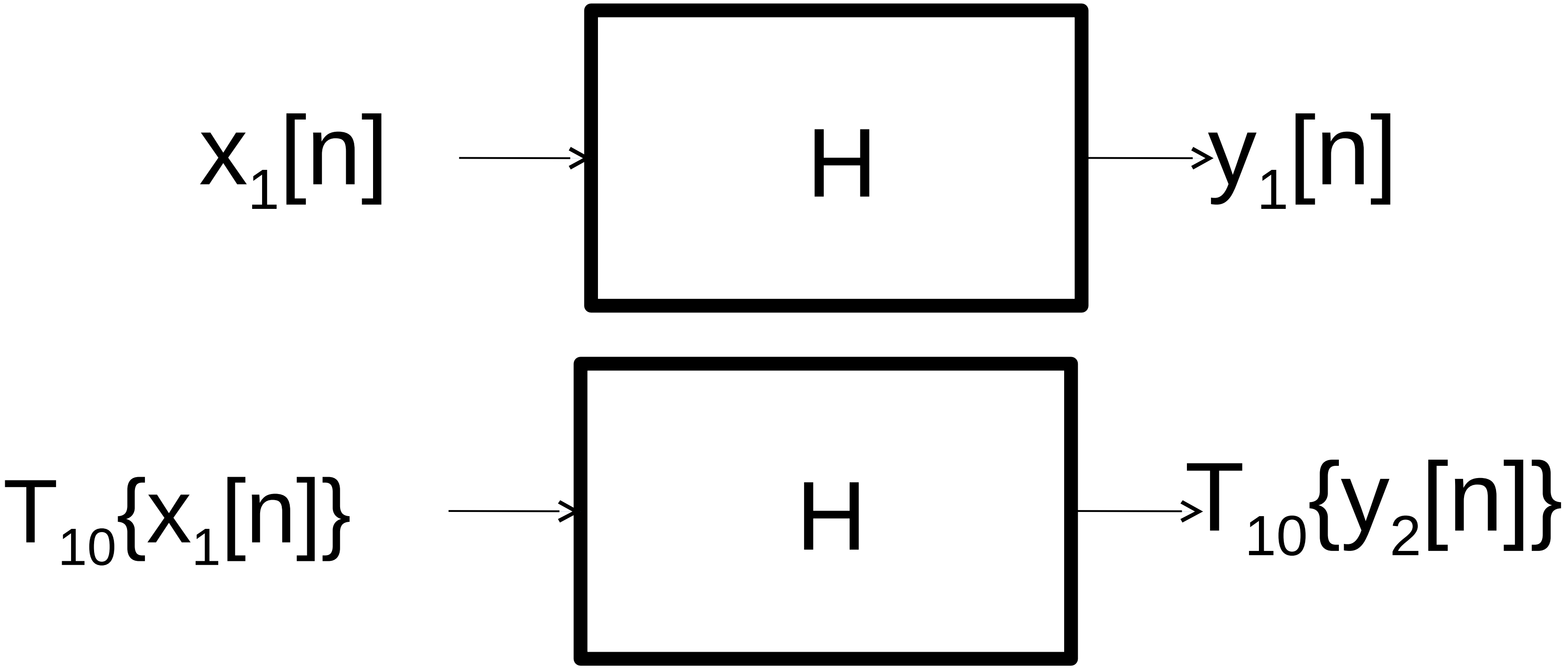
Time-invariant systems have the property that the output signal of the system for a given input signal does not depend explicitly on absolute time.
For any input signal x with y = H{x}, the system H is time-invariant if:

$$H\{T_\delta\{x\}\} = T_\delta\{y\}$$

Where T_δ is a delay system for arbitrary delay.

Passing a delayed sound through a time-invariant system gives the same result as passing the sound through the system first,
then delaying it.

Time invariance



Example

H is time-invariant

$x1 = [1, 2, 3]$

and $y1 = H\{x1\} = [3, 4, 5]$

$x2 = [0, 0, 1, 2, 3]$

$y2 = H\{x2\} = ?$

$x2 = T2\{x1\}$

so $y2 = T2\{y1\} = [0, 0, 3, 4, 5]$

Linear Time-Invariant or **LTI** Systems have both the linear property and the time-invariant property.

Why do we care?

Reason 1:

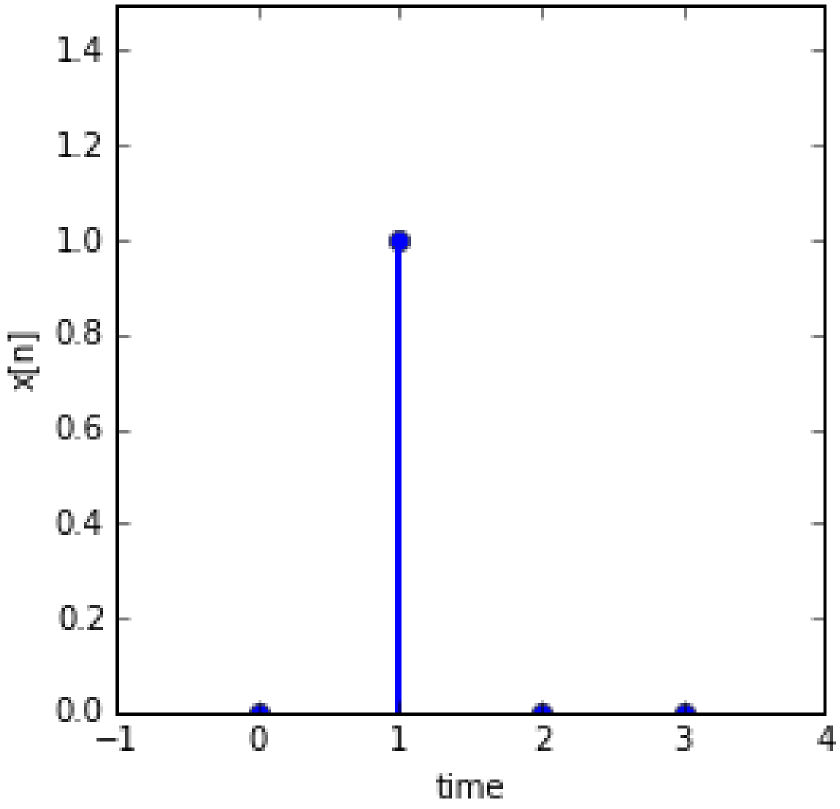
If we know the response of an LTI system for the signal below, we can compute the response to any signal whatsoever!

$$d[n] = [1, 0, 0, 0, \ldots] = [1]$$

The unit impulse signal

This special signal is called the unit impulse

d[n] = [1, 0, 0, 0, …] = [1]

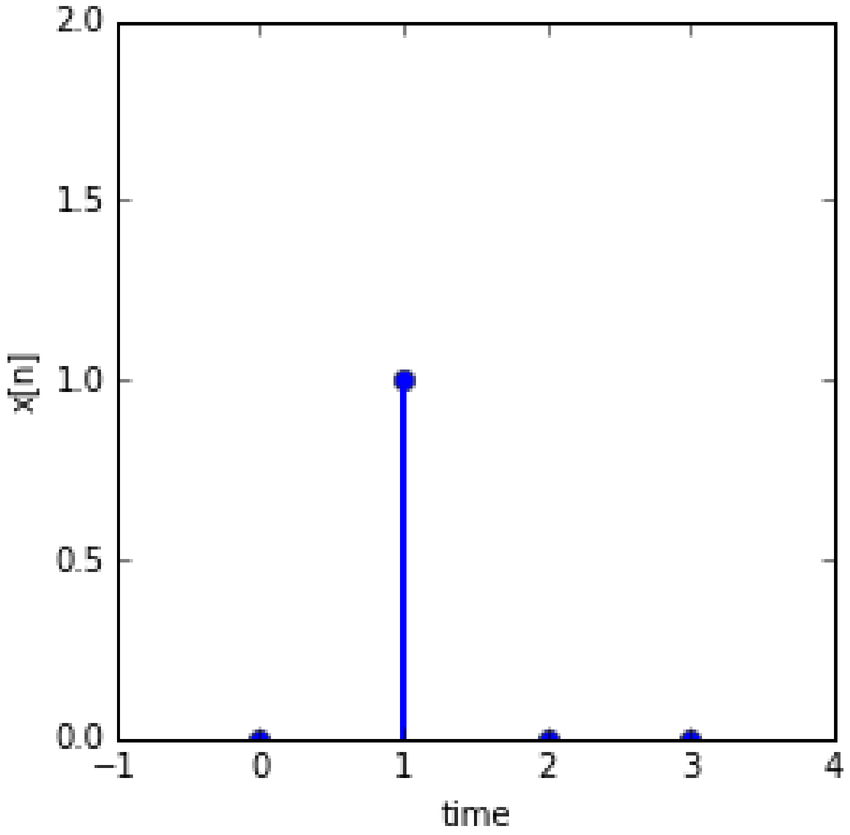


The unit impulse signal

d[n]

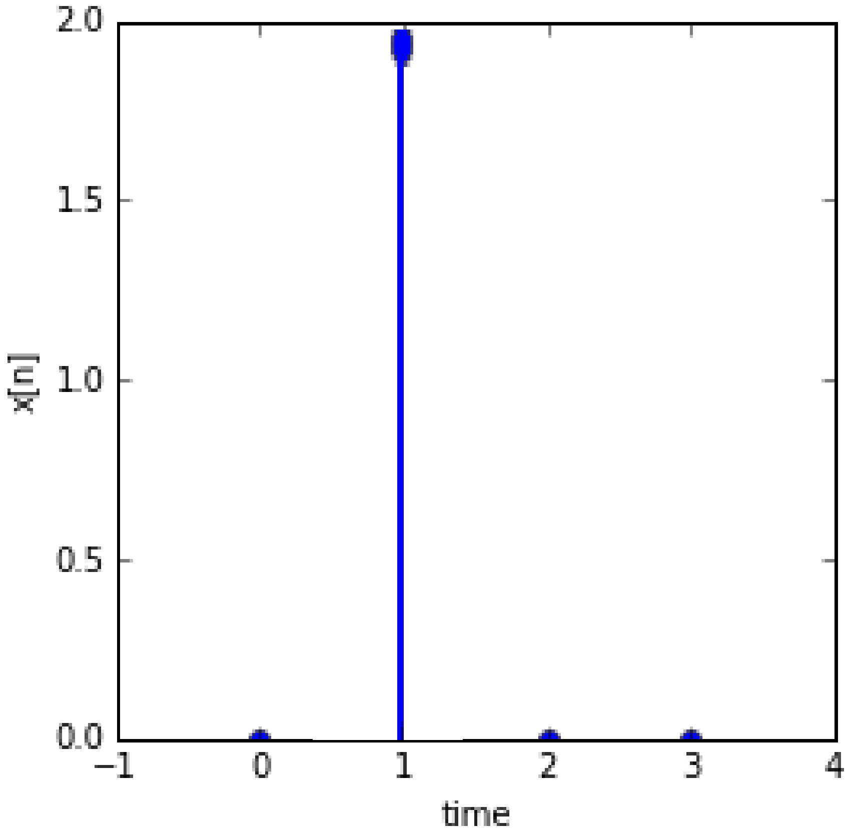
1 at time 0, 0 everywhere else

Any signal can be represented as weighted sum of delayed unit impulses.



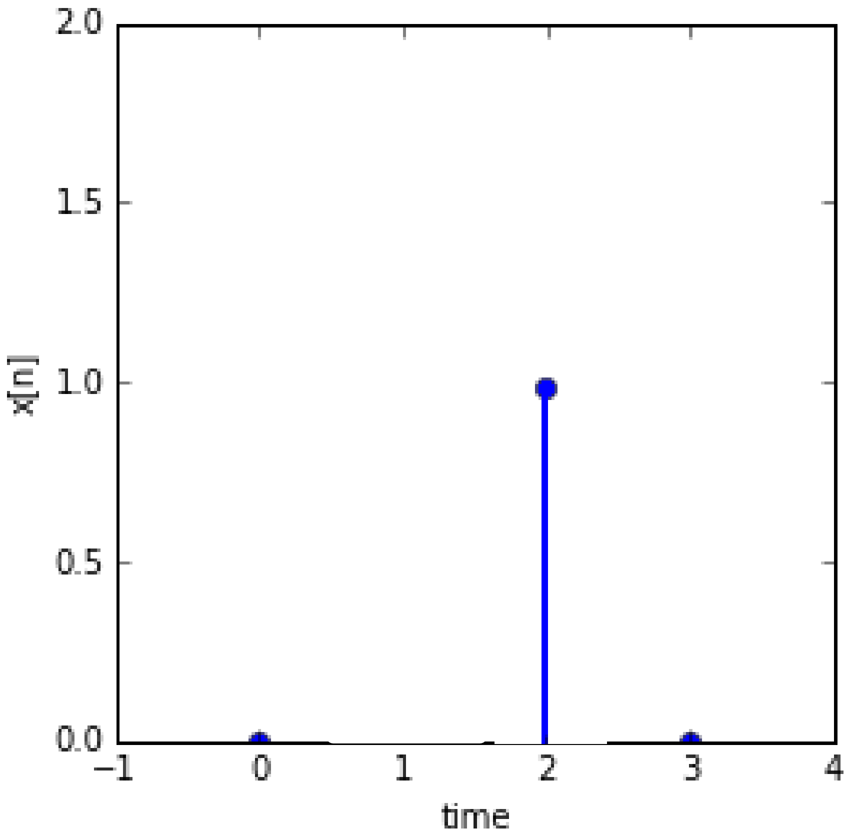
d[n]

Any signal can be represented as weighted sum of delayed unit impulses.



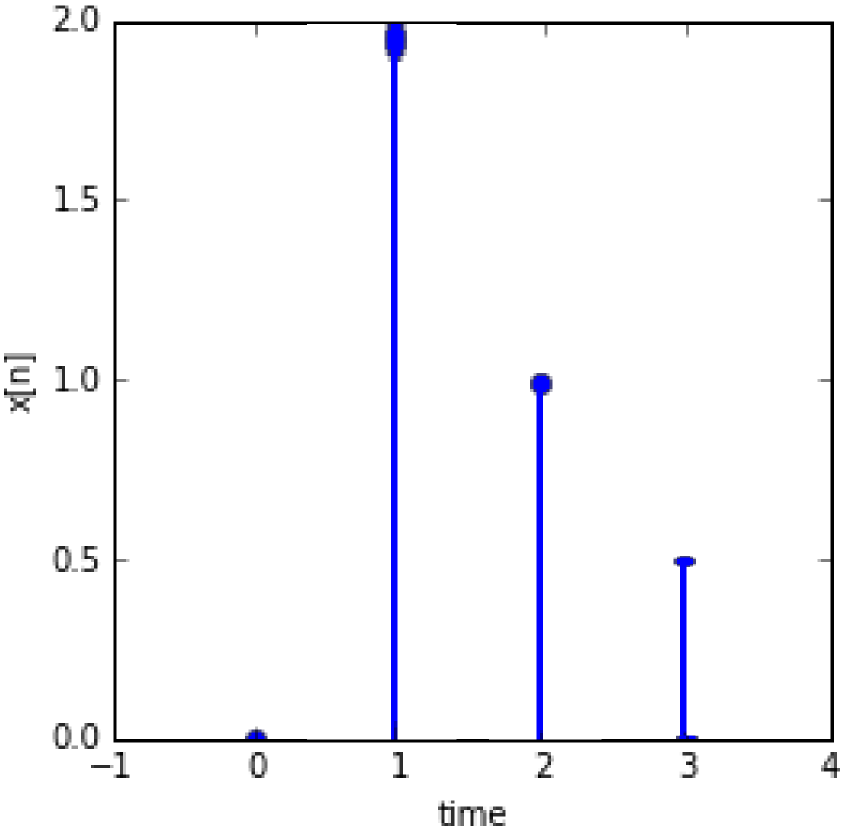
2d[n]

Any signal can be represented as weighted sum of delayed unit impulses.



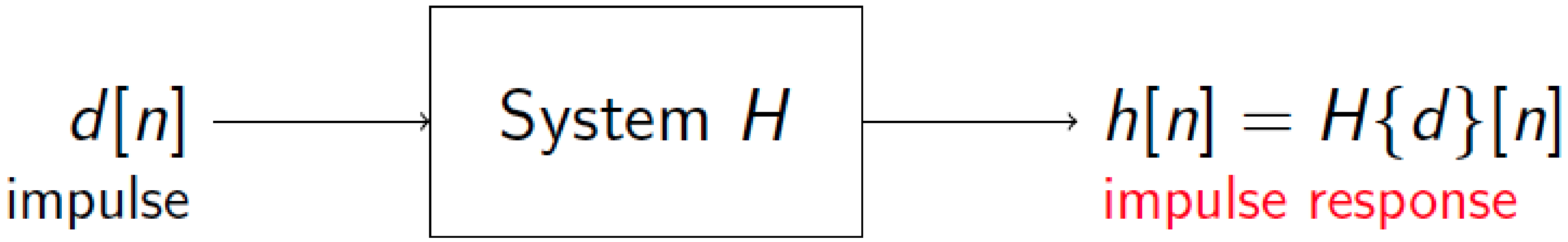
T1{d[n]}

Any signal can be represented as weighted sum of delayed unit impulses.



2*d[n] +
T1{d[n]} +
0.5 * T2{d[n]}

The response of an LTI system to the unit impulse input is called its “impulse response”



Any signal can be represented as weighted sum of delayed unit impulses.

The response of a LTI system for this signal is a weighted sum of delayed impulse responses.

Example 1

$H\{d[n]\} = 3$ impulse response

$x = [0, 1]$

$= T1\{d[n]\}$

$H\{x\} = H\{T_1\{d[n]\}\} = T_1\{H\{d[n]\}\}$

$=T_1\{[3]\} = [0, 3]$

Why? It's *time-invariant*

Example 1

$[1] = [1, 0, 0, \ldots] \rightarrow \boxed{H} \rightarrow [3] = [3, 0, 0, \ldots]$

d[n]: Impulse in

h[n]: Impulse response out

$x[n] \rightarrow \boxed{H} \rightarrow y[n]$

$x_1 = [0, 1]$

$y_1[n] = ?$

$x_1 = T_1\{d[n]\}$,

so $y = T_1\{h[n]\}$

Why?

It's *time-invariant*

Example 2

Why? It's linear;
this is *principle of scaling*

$H\{d[n]\} = 3$

$x = 5$

$= 5 * d[n]$

$H\{x[n]\} = H\{5 * d[n]\} = 5 * H\{d[n]\}$

$= 5 * 3 = 15$

Example 3

H{d[n]} = 3

x = [0, 5]

= T1{5*d[n]}

H{x[n]} = H{T1{5*d[n]}} = T1 {H{5*d[n]}} = 5*T1{H{d[n]}}

=5 * T1{[3]} = [0, 15]

Why? *Scaling* + *time-invariance*

Example 4

x[n] = [5, 1]

= T1{5*d[n]}

H{d[n]} = 3

H{x[n]} = H{T1{5*d[n]}} = T1 {H{5*d[n]}} = 5*T1{H{d[n]}}

=5 * T1{[3]} = [0, 15]

Why? *Scaling* + *superposition*

“I have a system here that I like. I wish I could re-create this system in my own software…”

e.g., “I like how music sounds when played in this giant cathedral. I wish I could make all my music sound like it’s recorded here.”

or

“I really like a ‘blur’ effect in this photo editing software. I wish I could make this effect in my own software.”

System-creation genie says:

Done!!!

(If you meet just a few conditions, then follow my instructions)

Systems must be
linear & time-invariant

System-creation genie says:

Done!!!

(If you meet just a few conditions, then follow my instructions)

1. Measure the impulse response of the system
2. Decompose your new signal into a set of weighted, delayed unit impulses, then figure out how to recombine the impulse responses to compute the new system response

Convolution

Why convolution?

For any linear, time-invariant system H, if we know the system’s impulse response:

$$[1] = [1, 0, 0, \ldots] \rightarrow \boxed{H} \rightarrow h[n]$$

Impulse in

Impulse response out

Why convolution?

Then we can compute the output of the system for any new input, by convolving that input with the impulse response.

$$x[n] \rightarrow \boxed{H} \rightarrow x[n] * h[n]$$

Arbitrary signal in

System's output for this signal where $*$ denotes convolution

What is convolution?

An operation on two signals (arrays)

Here, the *discrete-time* definition:

$$(h * x)[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

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

Note: Different from "*" operator in Python, Processing, etc.!

Applying convolution by hand

Example #1

H = [0.5]

x = [1, 2]

Applying convolution by hand

Example #2

$H = [0.5, 1]$

$x = [1, 2]$

$y[n] = x[n] * h[n]$ for any $x[n]$

This means that convolution with $h[n]$, the impulse response of H, is by definition equivalent to applying the system H to a signal!

Real-world example

I like how music sounds when played in this giant cathedral. I wish I could make all my music sound like it's recorded here.

What is system?

Cathedral

What is impulse response?

Live recording of cathedral when an impulse is played in cathedral

How to apply cathedral reverb to any new sound?

Convolve new sound with impulse response

Convolution Efficiency

Consider a linear, time-invariant system H with Finite Impulse Response

- Impulse Response of length L (in samples);
- All other values 0

Implementation of that system with direct convolution for an input signal X of length N:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=0}^{L-1} x[k]h[n-k]$$

Computational Complexity:

- $O(NL)$
- ... but we can do better ...

Exercises

Convolve:

1. $x = [1, 2, 3]$, $h = [2]$
2. $x = [2, 3, 4]$, $h = [0.5, 0.5]$
3. $x = [1, 1, 1]$, $h = [-1, 1]$

Coming up

This week's lab: Convolution

Next 2 weeks: Designing and reasoning about audio and image effects