6.300: Signal Processing

Unit-Sample Response and Convolution

DT: Convolution takes the form of a sum.

$(f*g)[n] \triangleq \sum_{m=-\infty}^{\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} g[m]f[n-m]$

CT: Convolution takes the form of an integral.

$(f*g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = \int_{-\infty}^{\infty} g(\tau)f(t-\tau)d\tau$

Convolution with impulses (i.e., $\delta[n]$ and $\delta(t)$) is easy!

- $f[n]*\delta[n-n_0] = f[n-n_0]$
- $f(t)*\delta(t-t_0) = f(t-t_0)$

March 12, 2026

Agenda for Recitation

- Terminology
- Convolution: Superposition (e.g., with a table)
- Convolution: Flip and shift (e.g., graphically)
- Example: Solera
- More on $u(t)$, $u[n]$, $\delta(t)$, and $\delta[n]$ (as time allows)

What questions do you have from lecture?

Agenda for Recitation

- Terminology
- Convolution: Superposition (e.g., with a table)
- Convolution: Flip and shift (e.g., graphically)
- Example: Solera
- More on $u(t)$, $u[n]$, $\delta(t)$, and $\delta[n]$ (as time allows)

Terminology

Let's start by clearing up the most common confusion.

Convolve means "compute the convolution $(f * g)$ of two signals f and g."

"To determine the output signal $y(t)$, convolve the input signal $x(t)$ with the system's impulse response $h(t)$."

Convolute means "make something complicated."
Convoluted means something is overly complicated.

"**@#$%&*!** This homework problem is so convoluted!"

-1000 points for saying "convolute" when you mean "convolve"! (Just kidding ... Well, maybe.)

Agenda for Recitation

- Terminology
- Convolution: Superposition (e.g., with a table)
- Convolution: Flip and shift (e.g., graphically)
- Example: Solera
- More on $u(t)$, $u[n]$, $\delta(t)$, and $\delta[n]$ (as time allows)

Convolution: Superposition

Superposition: $(f * g)[n]$ is

- a sum of scaled-by-$f[m]$, shifted-by-m copies of $g[n]$, or
- a sum of scaled-by-$g[m]$, shifted-by-m copies of $f[n]$.

Convolution: Superposition

Superposition: $(f * g)[n]$ is

- a sum of scaled-by-$f[m]$, shifted-by-m copies of $g[n]$, or
- a sum of scaled-by-$g[m]$, shifted-by-m copies of $f[n]$.

n	=	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$f[n]$	=	1	1	1	1	0	0	0	0
$g[n]$	=	1	2	3	0	0	0	0	0
$f[0]\,g[n-0]$	=	1	2	3	0	0	0	0	0
$f[1]\,g[n-1]$	=	0	1	2	3	0	0	0	0
$f[2]\,g[n-2]$	=	0	0	1	2	3	0	0	0
$f[3]\,g[n-3]$	=	0	0	0	1	2	3	0	0
$(f * g)[n]$	=	**1**	**3**	**6**	**6**	**5**	**3**	**0**	**0**

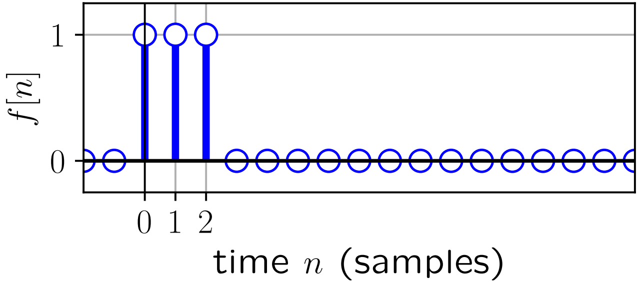
Convolution: Superposition

Superposition: $(f * g)[n]$ is

- a sum of scaled-by-$f[m]$, shifted-by-m copies of $g[n]$, or
- a sum of scaled-by-$g[m]$, shifted-by-m copies of $f[n]$.

n	=	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$f[n]$	=	1	1	1	1	0	0	0	0
$g[n]$	=	1	2	3	0	0	0	0	0
$g[0]\,f[n-0]$	=	1	1	1	1	0	0	0	0
$g[1]\,f[n-1]$	=	0	2	2	2	2	0	0	0
$g[2]\,f[n-2]$	=	0	0	3	3	3	3	0	0
$(f * g)[n]$	=	**1**	**3**	**6**	**6**	**5**	**3**	**0**	**0**

Convolution: Superposition



Sketch $(f * g_i)[n]$ for $i \in \{0, 1, 2, 3\}$.

- $g_0[n] = \delta[n]$
- $g_1[n] = \delta[n-3]$
- $g_2[n] = \delta[n-6]$
- $g_3[n] = \delta[n] + 2\delta[n-3] + 3\delta[n-6]$

Agenda for Recitation

- Terminology
- Convolution: Superposition (e.g., with a table)
- Convolution: Flip and shift (e.g., graphically)
- Example: Solera
- More on $u(t)$, $u[n]$, $\delta(t)$, and $\delta[n]$ (as time allows)

Convolution: Flip and Shift

"Flip and shift" is a four-step graphical method.

Flip: $g(\tau) \to g(-\tau)$

Shift: $g(-\tau) \to g(-(\tau - t)) = g(t - \tau)$

Multiply: $f(\tau)g(t - \tau)$

Add: $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$

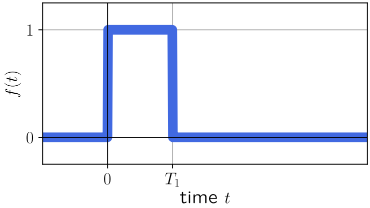
Maybe we should call it "flip-shift-multiply-add" instead.

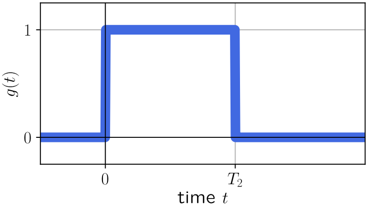
Convolution: Flip and Shift

Let’s convolve two “boxcars.”

from https://www.american-rails.com/box.html

Convolution: Flip and Shift





Compute $(f * g)(t)$. You may assume $0 < T_1 < T_2$.

Agenda for Recitation

- Terminology
- Convolution: Superposition (e.g., with a table)
- Convolution: Flip and shift (e.g., graphically)
- Example: Solera
- More on $u(t)$, $u[n]$, $\delta(t)$, and $\delta[n]$ (as time allows)

Solera

Solera is a process for aging and blending liquids (e.g., wine, beer, or vinegar) from different crops.[1]

- mixing produces a more uniform product
- mitigates worst-case results of one bad year
- blends wines from many previous years

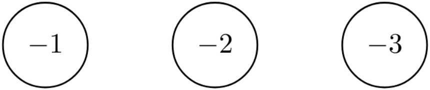
[1] Example by Professor Denny Freeman

Solera

Solera is a process for aging and blending liquids (e.g., wine, beer, or vinegar) from different crops.

- mixing produces a more uniform product
- mitigates worst-case results of one bad year
- blends wines from many previous years

Barrels of wine — newest at left, oldest at right:

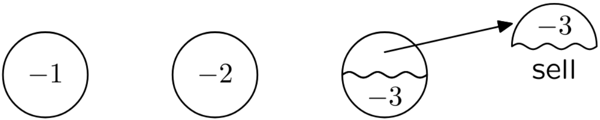


Solera

Solera is a process for aging and blending liquids (e.g., wine, beer, or vinegar) from different crops.

- mixing produces a more uniform product
- mitigates worst-case results of one bad year
- blends wines from many previous years

Sell half of the oldest stock.

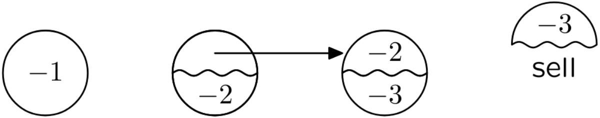


Solera

Solera is a process for aging and blending liquids (e.g., wine, beer, or vinegar) from different crops.

- mixing produces a more uniform product
- mitigates worst-case results of one bad year
- blends wines from many previous years

Refill oldest barrel from next-oldest barrel.

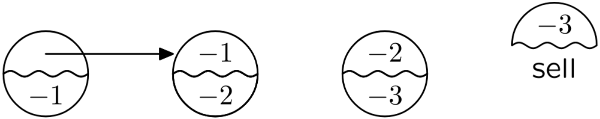


Solera

Solera is a process for aging and blending liquids (e.g., wine, beer, or vinegar) from different crops.

- mixing produces a more uniform product
- mitigates worst-case results of one bad year
- blends wines from many previous years

Refill next-oldest barrel from youngest barrel.

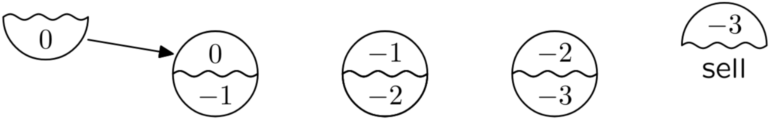


Solera

Solera is a process for aging and blending liquids (e.g., wine, beer, or vinegar) from different crops.

- mixing produces a more uniform product
- mitigates worst-case results of one bad year
- blends wines from many previous years

Refill youngest barrel with this year's harvest.

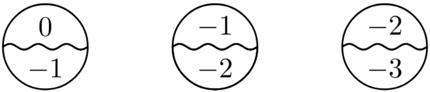


Solera

Solera is a process for aging and blending liquids (e.g., wine, beer, or vinegar) from different crops.

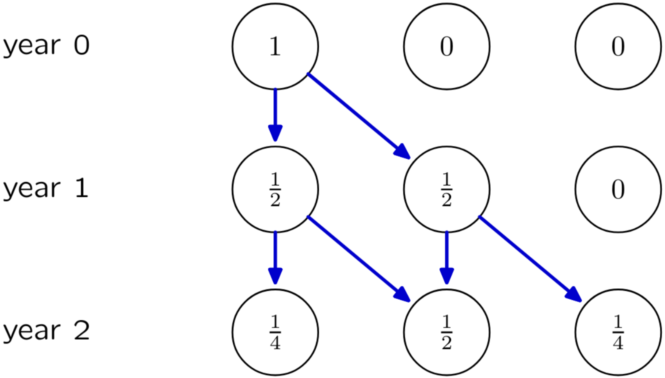
- mixing produces a more uniform product
- mitigates worst-case results of one bad year
- blends wines from many previous years

Old and new wines mix. Ready for next year!



Solera

Suppose we add one unit of "tracer" (like dye) to the new crop. Track the tracer through the barrels.



How much tracer will be in each barrel at the end of year 3?

Solera

Let n represent the year, $x[n]$ the tracer in, and $y[n]$ the tracer out. We can track the tracer through the process.

n	$x[n]$	Barrel #1	Barrel #2	Barrel #3	$y[n]$
0	1	1	0	0	0
1	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0
2	0	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{1}{4}$	0
3	0	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$
4	0	$\frac{1}{16}$	$\frac{4}{16}$	$\frac{6}{16}$	$\frac{3}{16}$
5	0	$\frac{1}{32}$	$\frac{5}{32}$	$\frac{10}{32}$	$\frac{6}{32}$
6	0	$\frac{1}{64}$	$\frac{6}{64}$	$\frac{15}{64}$	$\frac{10}{64}$

Suppose a unit of tracer is added in years 0, 1, and 2. How much tracer will be in the output of year 5?

1. $\frac{21}{32}$ **2.** $\frac{1}{2}$ **3.** $\frac{3}{16}$ **4.** $\frac{9}{16}$ **5.** none of these

Solera

Let n represent the year, $x[n]$ the tracer in, and $y[n]$ the tracer out. The solera process is an **LTI system!**

$$x[n] \rightarrow \boxed{\boxed{\text{barrel}} \rightarrow \boxed{\text{barrel}} \rightarrow \boxed{\text{barrel}}} \rightarrow y[n] = (x * h)[n]$$

To determine the system's **unit-sample response** $h[n]$, first determine the unit-sample response of a single barrel.

$$\delta[n] \rightarrow \boxed{\text{barrel}} \rightarrow h_1[n]$$

What is the **unit-sample response** $h_1[n]$ of the one-barrel solera process?

Solera

Next, determine the unit-sample response $h_2[n]$ of a **two-barrel** solera process.

$$\delta[n] \rightarrow \boxed{\boxed{\text{barrel}} \rightarrow h_1[n] \rightarrow \boxed{\text{barrel}}} \rightarrow h_2[n]$$

Solera

Finally, determine the **unit-sample response** $h_3[n] = h[n]$ of the **three-barrel** solera process.

$$\delta[n] \rightarrow \boxed{\boxed{\text{barrel}} \rightarrow \boxed{\text{barrel}} \rightarrow \boxed{\text{barrel}}} \rightarrow h_3[n] = h[n]$$

Agenda for Recitation

- Terminology
- Convolution: Superposition (e.g., with a table)
- Convolution: Flip and shift (e.g., graphically)
- Example: Solera
- More on $u(t)$, $u[n]$, $\delta(t)$, and $\delta[n]$ (as time allows)

The material presented after this point is **not necessary** for success in this subject; it is provided only for your enrichment. Perhaps you'll find it interesting.

Heaviside

$u(t)$ is sometimes called the Heaviside step function, after British electrical engineer **Oliver Heaviside** (1850–1925).

Wikipedia: "[Heaviside] invented a new technique for solving differential equations (equivalent to the Laplace transform), independently developed vector calculus, and rewrote Maxwell's equations in the form commonly used today."

Step Function

As usual, let $u(t)$ denote the unit step function.

It's interesting to note that

$$(f * u)(t) = \int_{-\infty}^{t} f(\tau)\,d\tau,$$

i.e., convolution with $u(t)$ represents **accumulation**.

Example: Consider charge accumulating on a capacitor.

- voltage: $v(t)$ volts
- current: $i(t)$ amps
- capacitance: C farads

$$v(t) = \frac{1}{C}\underbrace{\left(\int_{-\infty}^{t} i(\tau)\,d\tau\right)}_{\text{charge } q(t)} = \frac{\text{accumulation of current } i(t)}{C}$$

Step Response

It's also interesting to note that $u(t)$ is the integral of $\delta(t)$.

$$u(t) = \int_{-\infty}^{t} \delta(\tau)d\tau \iff \delta(t) = \frac{d}{dt}u(t)$$

Consequently, an LTI system's **impulse response**

$$\delta(t) \rightarrow \boxed{\textsf{LTI}} \rightarrow h(t) = \frac{d}{dt}h_{\textsf{step}}(t)$$

is the derivative of the system's **step response**!

$$u(t) \rightarrow \boxed{\textsf{LTI}} \rightarrow h_{\textsf{step}}(t) = \int_{-\infty}^{t} h(\tau)d\tau + C$$

Impulses

The **Dirac delta function** $\delta(t)$ is a **generalized function**. We use it to represent sudden "shocks" or "impulses." Think of $\delta(t)$ as a very tall, narrow pulse.

$$\delta(t) \triangleq \lim_{\Delta \to 0} p_\Delta(t) = \lim_{\Delta \to 0} \begin{cases} \frac{1}{\Delta} & -\frac{1}{2}\Delta \le t \le \frac{1}{2}\Delta \\ 0 & \text{otherwise} \end{cases}$$

You can also think of $\delta(t)$ as the probability density function of a **zero-variance Gaussian random variable**.

$$\delta(t) \triangleq \lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi}\sigma} e^{-t^2/2\sigma^2}$$

Perhaps the most important property of $\delta(t)$ (for us) is the **sifting property:**

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_0)\, dt = f(t - t_0).$$

Impulses

The Dirac delta function $\delta(t)$ is similar to (but distinct from) the **Kronecker delta function**, defined in 6.300 as

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

In some disciplines, like physics, notation for the Kronecker delta function looks like

$$\delta_{ij} \triangleq \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

because it often arises in sums over indices i and j. So,

$$\delta[n] = \delta_{n0} = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0. \end{cases}$$

Lessons Learned

Convolution is a mathematical operation that arises all the time in signal processing and adjacent disciplines.

DT: Convolution takes the form of a sum.

$$(f * g)[n] \triangleq \sum_{m=-\infty}^{\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} g[m]f[n-m]$$

CT: Convolution takes the form of an integral.

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = \int_{-\infty}^{\infty} g(\tau)f(t-\tau)d\tau$$

Convolution with impulses (i.e., $\delta[n]$ and $\delta(t)$) is easy!

- $f[n] * \delta[n-n_0] = f[n-n_0]$
- $f(t) * \delta(t-t_0) = f(t-t_0)$