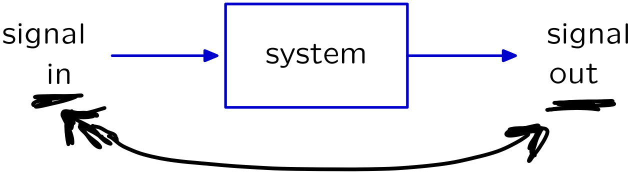
Systems

So far, we have focused our attention on signals and their mathematical representations. However, we are often also interested in *manipulating* signals. To this end, we introduce the notion of a **system** (or **filter**).

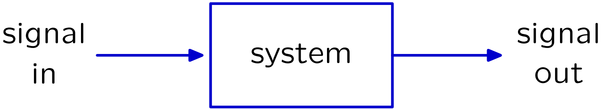


Examples:

- audio enhancement: equalization, noise reduction, reverberation, echo cancellation, pitch shift (auto-tune)
- image enhancement: smoothing, edge enhancement, unsharp masking, feature detection
- video enhancement: image stabilization, motion magnification

Representations of Signals

Characterize systems by their input/output relationships.



Difference Equation: represent system by algebraic constraints on samples

Convolution: represent system by its unit sample response

Filter: represent system as amplfication/attentuation of frequency components

Linearity

A system is linear if its response to a **weighted sum of inputs** is equal to the **weighted sum of its responses** to each of the inputs.

Given

$$x_1[n] \longrightarrow \boxed{\text{system}} \longrightarrow y_1[n]$$

and

$$x_2[n] \longrightarrow \boxed{\text{system}} \longrightarrow y_2[n]$$

the **system is linear** if

$$\alpha x_1[n] + \beta x_2[n] \longrightarrow \boxed{\text{system}} \longrightarrow \alpha y_1[n] + \beta y_2[n]$$

is true for all α and β and all possible inputs.

A system is linear if it is both additive and homogeneous.

Time-Invariance

A system is time-invariant if delaying the input to the system simply delays the output by the same amount of time.

Given

$$x[n] \longrightarrow \boxed{\text{system}} \longrightarrow y[n]$$

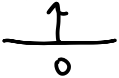
the **system is time invariant** if

$$x[n-n_0] \longrightarrow \boxed{\text{system}} \longrightarrow y[n-n_0]$$

is true for all n_0 and for all possible inputs.

Unit Sample Response

If a system is linear and time-invariant, its input-output relation is completely specified by the system's **unit sample response** $h[n]$.

$\delta(t) =$ 

Dirac

$$x[n] \longrightarrow \boxed{\text{system}} \longrightarrow y[n]$$

The unit sample response $h[n]$ is the output of the system when the input is the unit sample signal $\delta[n]$.

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \text{o.w} \end{cases}$$

Kroenecker

$$\delta[n] \longrightarrow \boxed{\text{system}} \longrightarrow h[n]$$

The output for more complicated inputs can be computed by summing scaled and shifted versions of the unit sample response.

Superposition

Consider the following signal:

$$x[n] = \begin{cases} 1 & \text{if } n = 0 \\ -1 & \text{if } n = 3 \\ -2 & \text{if } n = 4 \\ 0 & \text{otherwise} \end{cases}$$

This signal can be represented as:

$$x[n] = \delta[n] - \delta[n-3] - 2\delta[n-4]$$

In general, we can represent a signal as a sum of scaled, shifted deltas:

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

$$= \ldots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + \ldots$$

Superposition

In general, we can represent a signal as a sum of scaled, shifted deltas:

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

$$= \ldots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + \ldots$$

If $h[\cdot]$ is the unit sample response of an LTI system, then the output of that system in response to this arbitrary input $x[\cdot]$ can be viewed as a sum of scaled, shifted *unit sample responses*:

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

$$= \ldots + x[-1]h[n+1] + x[0]h[n] + x[1]h[n-1] + x[2]h[n-2] + \ldots$$

Structure of Superposition

If a system is linear and time-invariant (LTI) then its output is the sum of weighted and shifted unit sample responses.

$$\delta[n] \longrightarrow \boxed{\text{system}} \longrightarrow h[n]$$

$$\delta[n-m] \longrightarrow \boxed{\text{system}} \longrightarrow h[n-m]$$

$$x[m]\delta[n-m] \longrightarrow \boxed{\text{system}} \longrightarrow x[m]h[n-m]$$

$$x[n] = \sum_{m=-\infty}^{\infty} x[m]\delta[n-m] \longrightarrow \boxed{\text{system}} \longrightarrow y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

Superposition by Example

Consider an LTI system described by $y[n] = 3x[n] - x[n-1]$. Compute the response of this system to an input $x[\cdot]$ given by:

$$x[n] = 2\delta[n-1] + 5\delta[n-2] + 3\delta[n-4]$$

$$h[n] = 3\delta[n] - \delta[n-1]$$

$$y[n] = 2h[n-1] + 5h[n-2] + 3h[n-4]$$

n	-2	-1	0	1	2	3	4	5	6
$x[n]$	0	0	0	2	5	0	3	0	0
$2h[n-1]$	0	0	0	6	-2	0	0	0	0
$5h[n-2]$					15	-5			
$3h[n-4]$							9	-3	
$y[n]$	0	0	0	6	13	-5	9	-3	0

Convolution

Response of an LTI system to an arbitrary input.

$$x[n] \longrightarrow \boxed{\text{LTI}} \longrightarrow y[n]$$

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

This operation is called **convolution** (verb form: **convolve**).

Convolution

Convolution is represented with an asterisk.

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

$\cancel{x[n] * h[n]}$

Convolution operates on **signals**, not samples.

The symbols x and h represent DT signals.

Convolving x with h generates a new DT signal $x * h$.

DT Convolution: Summary

Unit sample response $h[n]$ is a complete description of an LTI system.

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

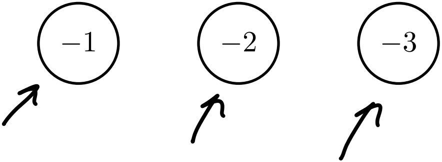
Given $h[\cdot]$, we can compute the response $y[\cdot]$ to any input $x[\cdot]$ by convolving $x[\cdot]$ and $h[\cdot]$:

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

Discrete-Time Example: Solera Process

Aging and blending wines from different crops.

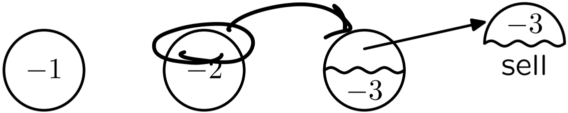
Start with 3 barrels of wine: newest at left, oldest at right.



Discrete-Time Example: Solera Process

Aging and blending wines from different crops.

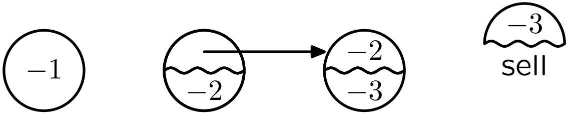
Sell half of the oldest stock.



Discrete-Time Example: Solera Process

Aging and blending wines from different crops.

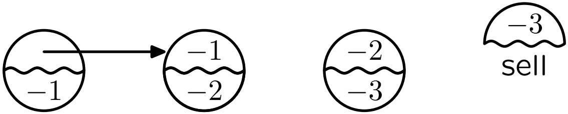
Refill oldest barrel from next-to-oldest barrel.



Discrete-Time Example: Solera Process

Aging and blending wines from different crops.

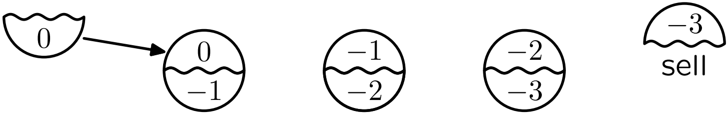
Refill next-to-oldest barrel from youngest barrel.



Discrete-Time Example: Solera Process

Aging and blending wines from different crops.

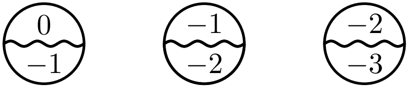
Refill youngest barrel with this year's harvest.



Discrete-Time Example: Solera Process

Aging and blending wines from different crops.

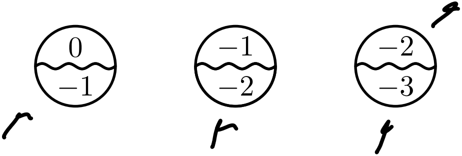
Old and new contents mix; ready for next year.



Discrete-Time Example: Solera Process

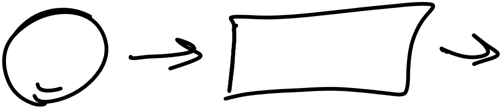
Aging and blending wines from different crops.

Old and new contents mix; ready for next year.



Properties of solera process:

- Mixing produces a more uniform product.
- Mitigates worst-case results of one bad year.
- Blends wines from MANY previous years.



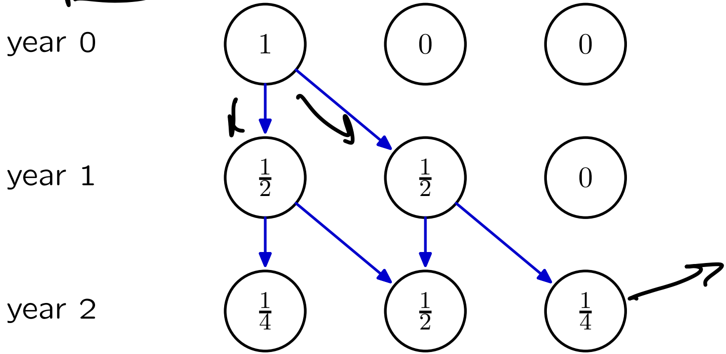
Solera Analysis

We can analyze these effects with a tracer experiment.

Add 1 unit of tracer to new crop; track tracer through the system.

amount of tracer added in year

amount of tracer in sold

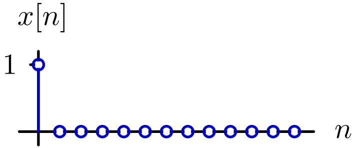


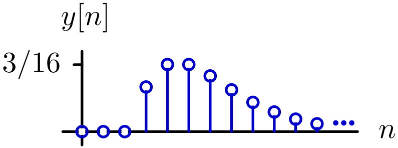
How much tracer will be in each barrel at the end of year 3?

Solera Analysis

Add 1 unit of tracer to new crop; track tracer through the system.

Year n	Tracer in $x[n]$	Barrel #1	Barrel #2	Barrel #3	Tracer out $y[n]$
0	1	1	0	0	0
1	0	1/2	1/2	0	0
2	0	1/4	2/4	1/4	0
3	0	1/8	3/8	3/8	1/8
4	0	1/16	4/16	6/16	3/16
5	0	1/32	5/32	10/32	6/32
6	0	1/64	6/64	15/64	10/64

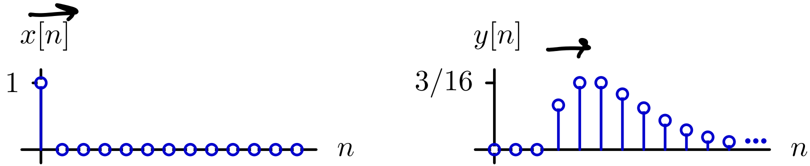




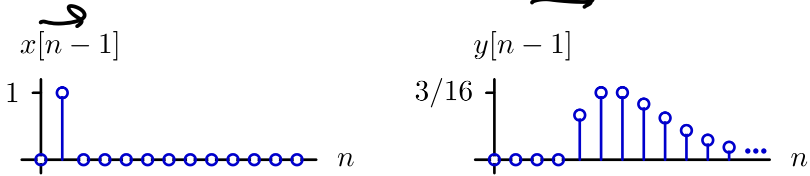
Solera Analysis

How would results change if tracer were added in year 1 (not 0)?

Original response:



Delayed input $\rightarrow$ delayed output:



Delaying the input by a year simply delays the outputs by one year.

Time-Invariance

A system is time-invariant if delaying the input to the system simply delays the output by the same amount of time.

Given

$$x[n] \longrightarrow \boxed{\text{system}} \longrightarrow y[n]$$

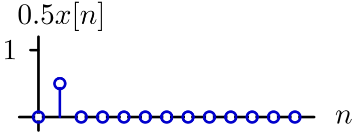
the system is **time invariant** if

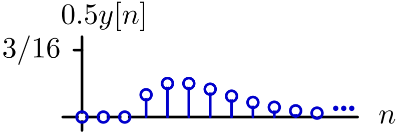
$$x[n-n_0] \longrightarrow \boxed{\text{system}} \longrightarrow y[n-n_0]$$

is true for all n_0.

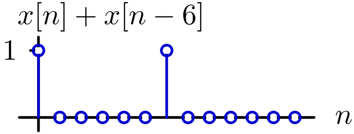
Solera Analysis

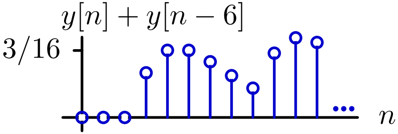
Scaling the input amplitudes:



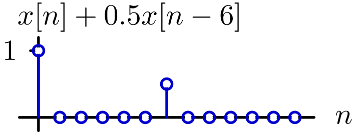


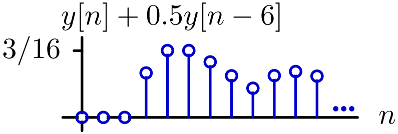
Adding two inputs:





Linearly combining two inputs:





Linearity

A system is linear if its response to a weighted sum of inputs is equal to the weighted sum of its responses to each of the inputs.

Given

$$x_1[n] \longrightarrow \boxed{\text{system}} \longrightarrow y_1[n]$$

and

$$x_2[n] \longrightarrow \boxed{\text{system}} \longrightarrow y_2[n]$$

the system is linear if

$$\alpha x_1[n] + \beta x_2[n] \longrightarrow \boxed{\text{system}} \longrightarrow \alpha y_1[n] + \beta y_2[n]$$

is true for all α and β.

Convolution

If a system is linear and time invariant, the response to an arbitrary input is the convolution of that input with the unit-sample response.

$$\delta[n] \longrightarrow \boxed{\text{system}} \longrightarrow h[n]$$

$$\delta[n-m] \longrightarrow \boxed{\text{system}} \longrightarrow h[n-m]$$

$$x[k]\delta[n-m] \longrightarrow \boxed{\text{system}} \longrightarrow x[k]h[n-m]$$

$$x[n] = \sum_{m=-\infty}^{\infty} x[k]\delta[n-m] \longrightarrow \boxed{\text{system}} \longrightarrow y[n] = \sum_{m=-\infty}^{\infty} x[k]h[n-m]$$

Check Yourself

For solera process ...

Year n	Tracer in $x[n]$		Barrel #1	Barrel #2	Barrel #3	Tracer out $y[n]$		
0	1	1	1	0	0	0	0	0
1	0	1	1/2	1/2	0	0	0	0
2	0	1	1/4	2/4	1/4	0	0	0
3	0		1/8	3/8	3/8	1/8	0	0
4	0		1/16	4/16	6/16	3/16	1/8	0
5	0		1/32	5/32	10/32	6/32	3/16	1/8
6	0		1/64	6/64	15/64	10/64	⋮	⋮

$$\frac{6}{32} + \frac{6}{32} + \frac{4}{32} = \frac{16}{32} = \frac{1}{2}$$

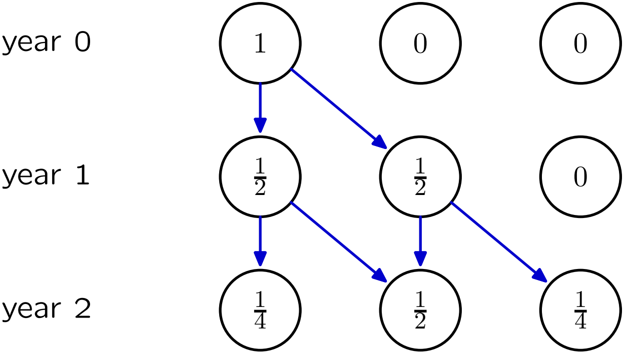
How much tracer would be found in output from year 5, if one unit of tracer is added in years 0, 1, and 2?

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

Divide and Conquer

The content of barrel #3 has no direct dependence on barrel #1.

The new content of barrel #3 depends only on itself and barrel #2. All dependence on barrel #1 is through barrel #2.



Since barrel #3 depends only on barrel #2, and barrel #2 depends only on barrel #1, the three barrel system is equivalent to the cascade of three one barrel systems!

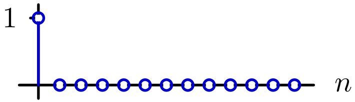
Divide and Conquer

Making a three-barrel system by cascading three one-barrel systems.

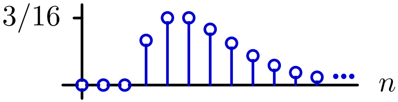
$$x[n] \longrightarrow \boxed{\text{3-barrel solera}} \longrightarrow y[n]$$

Year n	Tracer in $x[n] = \delta[n]$	Barrel #1	Barrel #2	Barrel #3	Tracer out $y[n] = h_3[n]$
0	1	1	0	0	0
1	0	1/2	1/2	0	0
2	0	1/4	2/4	1/4	0
3	0	1/8	3/8	3/8	1/8
4	0	1/16	4/16	6/16	3/16
5	0	1/32	5/32	10/32	6/32
6	0	1/64	6/64	15/64	10/64

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

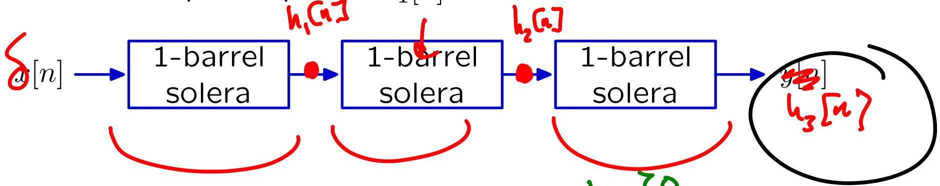


$y[n] = h_3[n]$

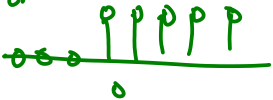


Divide and Conquer

Find the unit-sample response $h_1[n]$ for a 1-barrel solera.



$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & \text{o.w.} \end{cases}$$



Divide and Conquer

Find the unit-sample response $h_1[n]$ for a 1-barrel solera.

$$\delta[n] \rightarrow \boxed{\text{1-barrel solera}} \rightarrow h_1[n]$$

$\delta[n]$	$h_1[n]$
1	0
0	1/2
0	1/4
0	1/8
0	1/16
0	1/32

$$h_1[n] = \left(\tfrac{1}{2}\right)^n \overbrace{u[n-1]}^{\text{to account for } h_1[n\le 0]=0}$$

Divide and Conquer

How can we use this to find $h_2[n]$ (output of a two-barrel system)?

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

second barrel identical to first
(unit sample response = $h_1[n]$)

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

Compute response to arbitrary input by convolving input w/ USR!

$h_2[n] = (h_1 * h_1)[n]$

(work and answer on next page)

$$h_2[n] = (h_1 * h_1)[n] = \sum_{m=-\infty}^{\infty} h_1[m]\, h_1[n-m]$$ definition of convolution

plug in $h_1[\cdot]$

$$= \sum_{m=-\infty}^{\infty} \left(\tfrac{1}{2}\right)^m \underbrace{u[m-1]}_{\substack{0 \text{ unless} \\ m \geq 1}} \left(\tfrac{1}{2}\right)^{n-m} \underbrace{u[n-m-1]}_{\substack{0 \text{ unless} \\ m \leq n-1}}$$

these constrain our sum!
the terms in the sum are only nonzero when $1 \leq m \leq n-1$ (also implies $n \geq 2$)

$$= \sum_{m=1}^{n-1} \left(\tfrac{1}{2}\right)^m (1) \left(\tfrac{1}{2}\right)^{n-m} (1) = \sum_{m=1}^{n-1} \underbrace{\left(\tfrac{1}{2}\right)^n}$$

doesn't depend on m!

$$= \frac{1}{2}^n \sum_{m=1}^{n-1} (1) = (n-1)\left(\tfrac{1}{2}\right)^n \underbrace{u[n-2]}$$

to account for $h_2[n \leq 1] = 0$

Divide and Conquer

How can we use this to find $h_3[n]$ (output of a three-barrel system)?

$$\delta[n] \to \boxed{\text{barrel}} \xrightarrow{h_1} \boxed{\text{barrel}} \xrightarrow{h_2} \boxed{\text{barrel}} \to h_3[n]$$

notice:

$$h_2[n] \to \boxed{\text{barrel}} \to h_3[n]$$

our 1-barrel system from before!

So $h_3[n] = (h_2 * h_1)[n]$

(work and answer on following pages)

$$h_1[n] = \left(\tfrac{1}{2}\right)^n u[n-1] \qquad h_2[n] = (n-1)\left(\tfrac{1}{2}\right)^n u[n-2]$$

$$h_3[n] = (h_1 * h_2)[n] = \sum_{m=-\infty}^{\infty} h_1[m]\, h_2[n-m]$$

plug in h_1 and h_2

$$= \sum_{m=-\infty}^{\infty} \left(\tfrac{1}{2}\right)^m u[m-1]\, (n-m-1)\left(\tfrac{1}{2}\right)^{n-m} u[n-m-2]$$

terms are all 0, except when $m \geq 1$ and $n-m \geq 2$

(where $1 \leq m \leq n-2$, which also implies $n \geq 3$)

constrain the sum

$$= \sum_{m=1}^{n-2} (n-m-1)\left(\tfrac{1}{2}\right)^n = \left(\tfrac{1}{2}\right)^n \sum_{m=1}^{n-2} (n-m-1)$$

doesn't depend on m, pull outside

this looks ok, but how to simplify the sum? if we don't remember a formula for this, that's OK, we can think! (see next page)

$h_3[n] = \left(\frac{1}{2}\right)^n \sum_{m=1}^{n-2} (n-m-1)$, let's see how we can simplify $\sum_{m=1}^{n-2} (n-m-1)$

$$\sum_{m=1}^{n-2}(n-m-1) = n-2 + n-3 + n-4 + n-5 + \dots + \underbrace{n-(n-2)-1}_{=n-(n-1) \,=\, 1}$$

we have $n-2$ terms like n, so separate those out

$$= (n-2)n - \underbrace{\sum_{i=2}^{n-1} i}_{\text{how to think about this?}}$$

$$\sum_{i=2}^{n-1} i = 2+3+4+\dots+n-1$$

regroup to:

$$= 2+3+4+5+\dots$$
$$+\,n-1 + n-2 + n-3 + n-4 + \dots$$
$$n+1 + n+1 + n+1 + n+1 + \dots$$

and we have $\frac{n-2}{2}$ of these terms!

$$= \left(\frac{n-2}{2}\right)(n+1)$$

now plug in!

$$= (n-2)n - \left(\frac{n-2}{2}\right)(n+1)$$

$$= \frac{2n^2 - 4n - (n^2 - n - 2)}{2}$$

$$= \frac{n^2 - 3n + 2}{2} = \frac{(n-1)(n-2)}{2}$$ hooray!

(finish up on next slide)

$$h_3[n] = \left(\frac{1}{2}\right)^n \sum_{m=1}^{n-2} (n-m-1) = \left(\frac{1}{2}^n\right)\left(\frac{(n-1)(n-2)}{2}\right) u[n-3]$$

(matches graph on page 23, which we got via simulating the whole system)