Notes 2: Systems & System Properties

Objectives

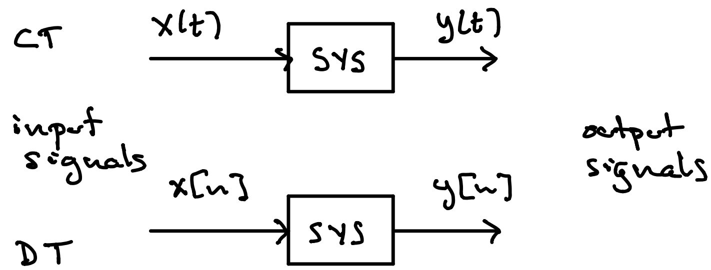
1) Understand how to use block diagrams to represent mathematical relationships between signals and how to turn a set of mathematical relationships between signals into a block diagram.

2) Check whether a system is memoryless or with memory, causal or noncausal or anticausal, invertible or non-invertible, stable or unstable, time invariant or time varying, and linear or nonlinear.

3) Come up with examples of systems which have a given set of the properties listed above.

Systems and system examples

Remember our "boxes and arrows" view:



We want to characterize different types of things that systems do. We saw a few examples earlier:

delay:

$$x(t) \longrightarrow \boxed{\text{delay}(\tau)} \longrightarrow y(t) = x(t-\tau)$$

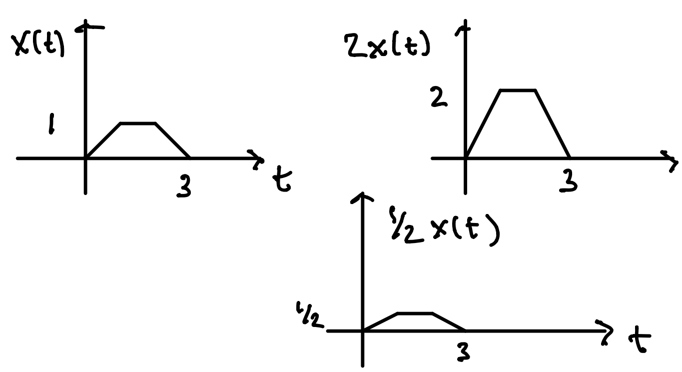
if $\tau > 0$ then this is a delay

if $\tau < 0$ then this is lookahead (!?)

amplification / attenuation:

$$y(t) = \alpha\, x(t)$$

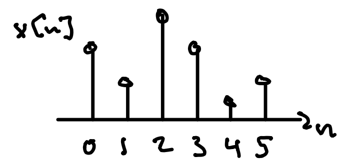
$\alpha > 1$ amplify
$\alpha < 1$ attenuate

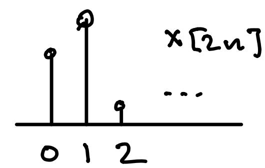


time dilation/contraction:

$$y(t) = x(\alpha t) \qquad \alpha > 1 \ \text{contraction}$$
$$\alpha < 1 \ \text{dilation}$$

$$y[n] = x[kn] \qquad k > 1 \ \text{downsampling}$$





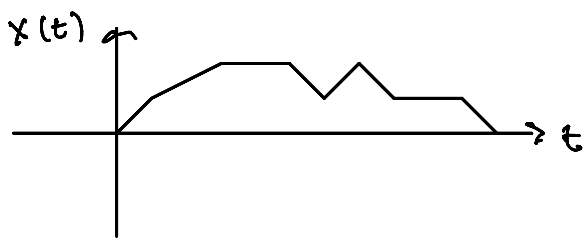
time reversal:

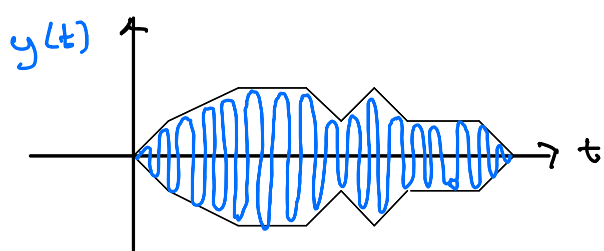
$$y(t) = x(-t)$$
$$y[n] = x[-n]$$

There are lots of other operations on signals that you can think of:

modulation:

$$y(t) = x(t) \cos(2\pi f_c t)$$





This is called amplitude modulation (AM) because the amplitude of x(t) gives the amplitude of the cosine wave. This is how AM radio works: an audio signal (like a song or a baseball game) is modulated onto a carrier frequency f_c (which is the station's frequency) and the signal transmitted over the antenna is $y(t)$.

Phil

Actually, there is a little more stuff going on in real AM radio systems, but this is basically how they work (and how early AM radios worked).

In general, in this class we will be learning about the core principles behind technologies that use systems to process signals. Real deployed systems have a lot of tweaks & hacks to make them more robust/efficient/"usable."

For example, the core ideas behind JPEG image compression are straightforward, but the actual JPEG standard is more complex — try to take a look if you are interested!

Main

Frequency modulation (FM) uses $x(t)$ to modulate the frequency (duh):

$$y(t) = \cos\left(2\pi f_c t + 2\pi K \int_0^t x(\tau)\,d\tau\right)$$

If you're interested in AM/FM radio and how it works, some basics are covered in ECE 322 (Principles of Comm Systems).

!!!

FM is beyond the scope of this class but we will see AM as an example later on...

Main

The book has many examples of physics and circuit based systems (1.8, 1.9) and systems based on difference equations (1.10, 1.11).

The main goal is to figure out how to group (or classify) systems into categories so that we can analyze all systems in a particular category/type using a common approach.

Phil

There is a famous quote from the statistician George Box that

"all models are wrong but some are useful"

That is exactly what engineering analysis is about: building a model for the problem you want to solve that is "good enough" to capture the relevant behavior of the system in the real world but also be "simple enough" that we can design solutions.

Main

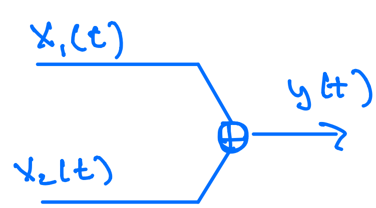
Boxes and Arrows revisited

We want to be able to use block diagrams to understand how systems connect to each other. These block diagrams will help us visualize the interactions between different systems.

Ex

Examples galore:

1) $y(t) = x_1(t) + x_2(t)$



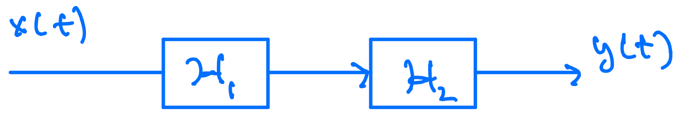
2) If we think of systems as maps:

$$y(t) = \mathcal{H}(x(t))$$

Then we can apply systems one after the other (sometimes called a cascade:

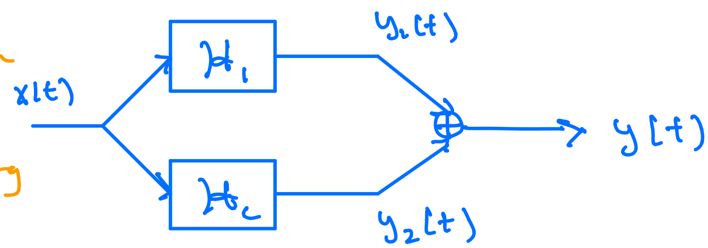
$$y(t) = \mathcal{H}_2(\mathcal{H}_1(x(t)))$$

Multi-stage processing



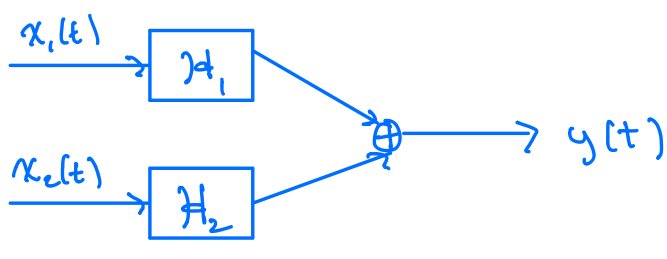
$$y(t) = \mathcal{H}_1(x(t)) + \mathcal{H}_2(x(t))$$

Filterbank or parallel processing



$$y(t) = \mathcal{H}_1(x_1(t)) + \mathcal{H}_2(x_2(t))$$

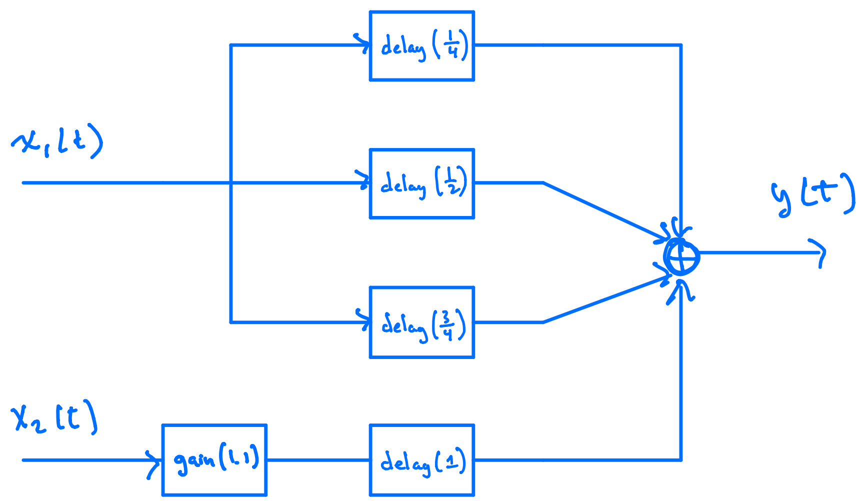
A model for analog synthesizers



!!!

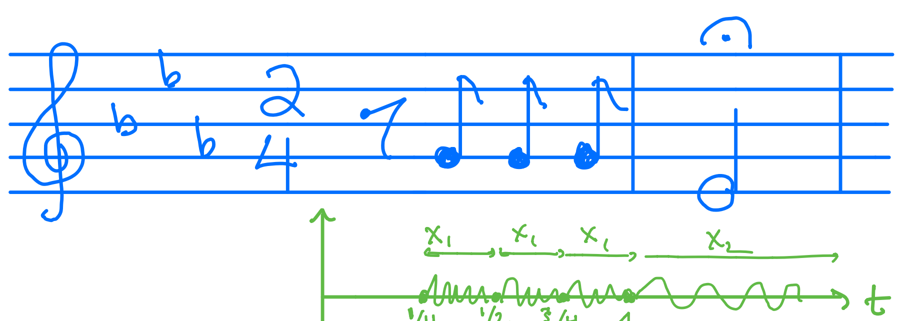
All of these examples are described for CT signals but you can get DT examples by just substituting (t) for [n].

3) We can get more elaborate:



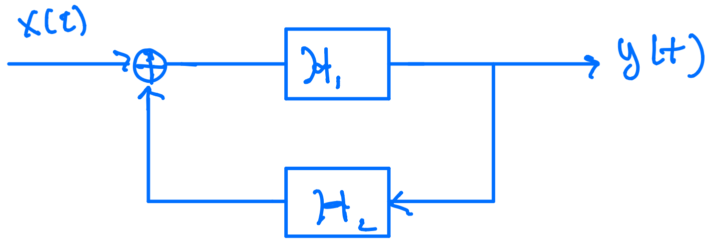
$$y(t) = x_1(t-1/4) + x_1(t-1/2) + x_1(t-3/4) + 1.1\, x_2(t-1)$$

This could be a model for the opening to Beethoven's 5th Symphony (see YouTube if you haven't heard it):



!!! Knowledge of Western classical music is <u>not</u> required for this course — I am just trying to share some fun examples I can think up — if you have others, please do let me know!

4) Feedback!



This is a bit more complicated:

It helps to define a signal

$$z(t) = \mathcal{H}_2(y(t))$$

!!! <u>Protip:</u> If you are faced with some complicated problem it often helps to name some of the intermediate quantities to help sort out what is going on...

Ex

input to $\mathcal{H}_1$: $x(t) + z(t)$

$$y(t) = \mathcal{H}_1(x(t) + z(t))$$

$$z(t) = \mathcal{H}_2(y(t))$$

$$\Rightarrow y(t) = \mathcal{H}_1(x(t) + \mathcal{H}_2(y(t)))$$

Let's look at a DT version where $\mathcal{H}_2$ is a <u>unit delay</u>: $\mathcal{H}_2(s[n]) = s[n-1]$

and $\mathcal{H}_1(s[n]) = 2s[n] + s[n-1]$

Then

$$y[n] = \mathcal{H}_1(x[n] + \mathcal{H}_2(y[n]))$$
$$= \mathcal{H}_1(x[n] + y[n-1])$$
$$= 2x[n] + 2y[n-1] + x[n-1] + y[n-2]$$

or $$y[n] - 2y[n-1] - y[n-2] = 2x[n] + x[n-1]$$

This is a somewhat complicated relationship between input and output! How should we understand it? More on that coming soon...

Main

System Properties

One way to classify systems is in terms of the kinds of transformations they use. A related way is to look at the properties they have:

Causality: $y[n]$ or $y(t)$ only depends on current/past inputs $x[n]$ or $x(t)$

Invertibility: can we get $x[n]$ (or $x(t)$) from $y[n]$ (or $y(t)$)?

Memory: Does $y[n]$ depend only on the current $x[n]$?

Stability: Can $y[n] \longrightarrow \infty$ as $n \to \infty$?

Time-invariance: If $y[n] = \mathcal{H}(x[n])$ and I delay the input, do we have

$$\mathcal{H}(x[n-m]) = y[n-m] \; ?$$

Linearity: if $y_1[n] = \mathcal{H}(x_1[n])$ and $y_2[n] = \mathcal{H}(x_2[n])$,

is $\mathcal{H}(\alpha_1 x_1[n] + \alpha_2 x_2[n])$

$$= \alpha_1 y_1[n] + \alpha_2 y_2[n] \; ?$$

Def

<u>Def</u>. A system is <u>memoryless</u> if the output $y(t)$ (or $y[n]$) at time t (or n) only depends on the input $x(t)$ (or $x[n]$) at time t (or n).

We call a system that is not memoryless a system <u>with memory</u>.

Ex

<u>Examples</u>:

i) $y[n] = |x[n]| - 3\cos(2\pi x[n])$

Memoryless

2) $y(t) = x(t) + 3x(t)^2 - x(t-1)$

<u>Not</u> memoryless

3) $y(t) = x(t)\cos(2\pi f_0 t)$

Memoryless

4) $y[n] = x[n] + \frac{1}{2}x[n-1] + \frac{1}{4}x[n-2] + \dots$

<u>Not</u> memoryless

Try

Try to come up with examples of systems that are real / complex, DT/CT, and memoryless / with memory. 8 combinations in all!

Def

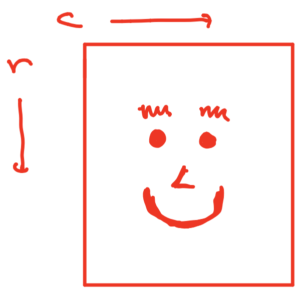
<u>Def</u>. A system is <u>causal</u> if the output at t $y(t)$ (or $y[n]$) only depends on the current and past inputs. A system is <u>noncausal</u> if it is not causal. A system is <u>anticausal</u> if the output at time t depends only on current and future inputs.

!!!

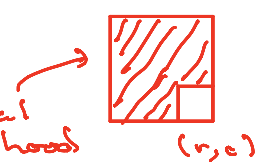
Sometimes it gets a bit confusing thinking about time like this. When would a system be anticausal? This is largely due to a

Sort of "mixed metaphors" problem.

Remember that signals are not always functions over time — we are calling the independent variable time just so we can be concrete. Lots of signals vary over space — images, for example — so we might define the coordinates of a pixel in terms of rows & columns, with the top left being (0,0). Then an anticausal system would use only pixels to the left & up from (r,c) to compute the output at (r,c)



an image



A second thing to think about is that not all systems operate in "real time". If you have an audio recording you can apply systems where $y[n]$ is a function of $x[n]$, $x[n+1]$, ... etc. The same holds for all sorts of offline signal processing.

Ex

<u>Examples</u> : $y[n] = x[n] + x[n-1] + x[n-2]$

Causal

$$y(t) = \int_{-\infty}^{t} x(\tau)\, d\tau \qquad \text{Causal}$$

called an "integrator"

Any memoryless system is causal (why?)

$$y(t) = x(t) \cos(2\pi f_c (t+\tau))$$

This is causal because we only care about how $y(t)$ depends on $x(t)$, not on this cosine term.

$$y[n] = \frac{1}{2n+1} \sum_{m=-n}^{n} |x[m]|^2$$

Where have we seen this before?
(Power calculations)

This is not causal. To see this it's important to check the output for all n (or t):

$$y[3] = \frac{1}{7}\left(x[-3] + x[-2] + \cdots + x[3] \right)$$

looks causal, right? For $n > 0$, $y[n]$ depends only on $x[-n], \dots x[n]$.

But:

$$y[-3] = \frac{1}{7}\left(x[3] + x[2] + \cdots + x[-3] \right)$$

$$= y[3]$$

depends on $x[3]$, $x[2]$, ...

For $n<0$, $y[n]$ depends on future $\{x[k]\}$ values so it cannot be causal.

!!!

Protip: check what happens for $n \geq 0$ and $n < 0$ ... sometimes things that look causal are not.

Try

Try the same 8 combinations from the previous self-check, but replace memoryless/with memory with causal vs. noncausal.

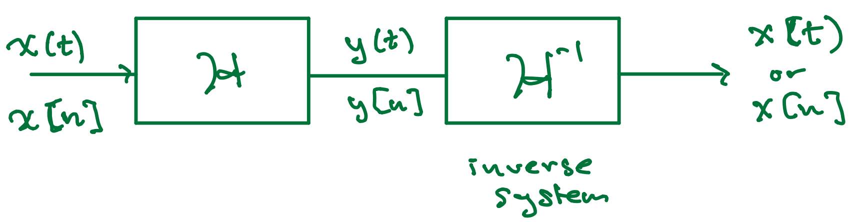
Def

Def. A system $\mathcal{H}$ is invertible if there is a system $\mathcal{H}^{-1}$ such that

$$\mathcal{H}^{-1}(\mathcal{H}(x(t))) = x(t)$$

or

$$\mathcal{H}^{-1}(\mathcal{H}(x[n])) = x[n]$$



That is, different inputs $x_1(t)$, $x_2(t)$ lead to different outputs. $y_1(t)$, $y_2(t)$

$$x_1(t) \neq x_2(t) \implies y_1(t) \neq y_2(t)$$

Ex

Examples:

Invertible systems:

1) $y[n] = \alpha\, x[n]$ system

$x[n] = \frac{1}{\alpha} y[n]$ inverse

2) $y[n] = \sum_{m=-\infty}^{n} x[m]$ (accumulator)

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

3) $y(t) = x(t-3)$ (delay by 3)

$x(t) = y(t+3)$ (lookahead by 3)

note this is anticausal!

Non-invertible systems

1) $y(t) = |x(t)|^2$

many inputs have the same output:

for real $x(t)$, if

$x_1(t) = x(t)$

$x_2(t) = -x(t)$

then $y_1(t) = y_2(t)$

for complex $x(t)$, if

$x_\omega(t) = e^{j\omega t}$ then

$y_\omega(t) = 1$ for any ω.

2) $y[n] = x[2n]$ (generally non-invertible)

we are throwing away all $x[k]$

for odd k — can't get them back!

Def

Def. A system is bounded-input bounded-output (BIBO)* stable if for any input $x(t)$ (or $x[n]$) such that $|x(t)| \leq B$ (or $|x[n]| \leq B$), the output $|y(t)| \leq f(B)$ (or $|y[n]| \leq f(B)$).

Not

* Some people say "bee-boh" and others say "bye-boh." I have never heard "bee-bow" but you never know..

Main

BIBO systems have bounded outputs when the input is bounded.

Ex

Examples:

1) $y[n] = |x[n]|^2$ calculates the instantaneous power

if $|x[n]| \leq B$ then

$|y[n]| \leq B^2$ so $f(B) = B$

This system is BIBO stable.

2) $y(t) = \int_{-\infty}^{t} x(\tau)\, d\tau$ integrator

this system is not stable. To see why it helps to come up with a counterexample:

try $x(t) = u(t)$

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

$$= \begin{cases} 0 & t < 0 \\ t & t > 0 \end{cases}$$

but then

$$|y(t)| = t \longrightarrow \infty \quad \text{as } t \to \infty$$

so $y(t)$ "blows up" and this system is <u>not</u> BIBO stable.

Main

Another way to look at the accumulator system

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

is recursively:

$$y[n-1] = \sum_{m=-\infty}^{n-1} x[m]$$

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

this will often be a more useful/compact way to think about this system.

The two most important/useful properties that systems can have are <u>linearity</u> and <u>time-invariance</u>. Linear time-invariant (LTI) systems will be the main type of systems we will discuss in this class.

Def

<u>Def.</u> A system is <u>linear</u> if for any complex a_1, a_2:

$$\mathcal{H}\left(a_1 x_1(t) + a_2 x_2(t)\right) = a_1 \mathcal{H}(x_1(t)) + a_2 \mathcal{H}(x_2(t))$$

$$\mathcal{H}\left(a_1 x_1[n] + a_2 x_2[n]\right) = a_1 \mathcal{H}(x_1[n]) + a_2 \mathcal{H}(x_2[n])$$

Main

Linear systems are important because they let us analyze what happens to complicated signals by analyzing what happens to the components of the signal separately.

It's usually possible to check whether a system is linear but sometimes the math can get a little confusing. So it's important to practice.

Try

For any of the systems considered earlier in these notes or in Section 1.5 of the book, check if they are linear.

Ex

Examples:

$y(t) = t\, x(t)$

To check: set $x(t) = a_1 x_1(t) + a_2 x_2(t)$

$$
\begin{aligned}
y(t) &= t\,(a_1 x_1(t) + a_2 x_2(t)) \\
&= a_1\, t\, x_1(t) + a_2\, t\, x_2(t) \\
&= a\, y_1(t) + a_2\, y_2(t)
\end{aligned}
$$

if we define $y_1(t)$ and $y_2(t)$ to be the system outputs for $x_1(t)$ and $x_2(t)$.

So yes, LINEAR

$y(t) = x(t)^3$

Check:

$$
\begin{aligned}
y(t) &= (a_1 x_1(t) + a_2 x_2(t))^3 \\
&= a_1^3 x_1(t)^3 + 3 a_1^2 a_2 x_1(t)^2 x_2(t) \\
&\quad + 3 a_1 a_2^2 x_1(t) x_2(t)^2 \\
&\quad + a_2^3 x_2(t)^3 \\
&\neq a_1 x_1(t)^3 + a_2 x_2(t)^3
\end{aligned}
$$

NOT LINEAR ... maybe this was obvious?

See Examples 1.19 and 1.20 for other cases — be careful!

Def

Def. A system is time-invariant if for all signals $x(t)$ (or $x[n]$), if

$$y(t) = \mathcal{H}(x(t))$$

$$(\text{or } \ y[n] = \mathcal{H}(x[n]))$$

then

$$\mathcal{H}(x(t-t_0)) = y(t-t_0)$$

$$(\text{or } \ \mathcal{H}(x[n-m]) = y[n-m])$$

The meaning is this: shifting the input by a delay / lookahead t_0 (or m) shifts the output by the same amount.

Phil

This notion of time-invariance is very important and also kind of special. It means the system's response is sort of "on demand": if you delay the start of the input, the start of the output is also delayed. By contrast, time varying systems have different reactions / outputs depending on the <u>absolute</u> time.

Ex

Examples:

1) $y(t) = x(t)^2$

Check: $x(t-t_0)^2 = y(t-t_0)$

time invariant! But not linear...

2) $y[n] = n\, x[n]$

Check: input $z[n] = x[n-n_0]$

output $n z[n] = n\, x[n-n_0]$

$\neq y[n-n_0]$

$= (n-n_0)\, x[n-n_0]$

!!!

Protip: use the definitions but be careful to follow all the steps! Cutting corners when you are first learning leads to big errors!

Try

Try doing the example 1.16 in the book. Be sure you understand what is going on. Try to make up other examples of non-time invariant systems that are similar & dissimilar to this example.

Go through the earlier examples here and in the book — which systems are time-invariant?