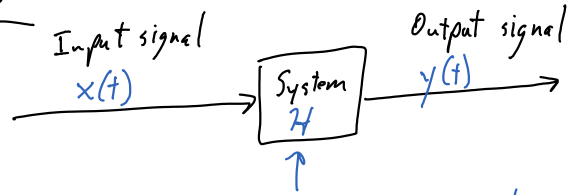
Lecture 11

Wednesday, March 04, 2015 8:46 AM

Systems:



- signal processing algorithm
- circuit
- mechanical device
- communication network
- computer system.

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

This is casual notation like our convolution notation.

System Properties:

For each property statement, assume $y = \mathcal{H}(x)$, $y_1 = \mathcal{H}(x_1)$, $y_2 = \mathcal{H}(x_2)$

Memoryless: $\mathcal{H}$ is memoryless if $y(t)$ depends only on input $x(t)$ at current time t.
(i.e. $x_1(t) = x_2(t) \Rightarrow y_1(t) = y_2(t)$ for any specific t.)

Causal: $\mathcal{H}$ is causal if $y(t)$ depends only on $x(\tau)$ for $\tau \le t$, $\forall t$
(i.e. $x_1(\tau) = x_2(\tau)\ \forall \tau \le t \Rightarrow y_1(t) = y_2(t)$ for any specific t.)

Note: Memoryless $\Rightarrow$ causal
Real-time systems are causal

Time-invariant: ("shift-invariant") $\mathcal{H}$ is time invariant if $y(t-t_0) = \mathcal{H}(x(t-t_0))\quad \forall t, t_0$
(i.e. $x_2(t) = x_1(t-t_0)\ \forall t \Rightarrow y_2(t) = y_1(t-t_0)\ \forall t$ for any t_0.)

Linear: $\mathcal{H}$ is linear if $\mathcal{H}(a x_1 + x_2) = a\,\mathcal{H}(x_1) + \mathcal{H}(x_2)$

Invertible: $\mathcal{H}$ is invertible if x can be recovered from y.
- No loss of information
- $\mathcal{H}^{-1}$ exists where $\mathcal{H}^{-1}(\mathcal{H}(x)) = x$ for any x.

Stable: $\mathcal{H}$ is stable if a bounded input always produces a bounded output.

Stable: $\mathcal{H}$ is stable if a bounded input always produces a bounded output.

BIBO

(i.e. $|x(t)| \le B_x \ \forall t$ for $B_x \Rightarrow |y(t)| \le B_y \ \forall t$ for some B_y)

Examples:

1.) $y(t) = \sin(t+1) + x(t)$

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

3.) $y(t) = \int_0^t x(s)\,ds$

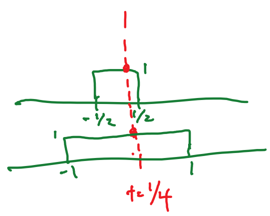
4.) $y(t) = t\,x(t) + x(t-1)$

	Memoryless	Causal	T.I	Linear	Invertible	Stable
1.)	✓ →	✓	×	×	✓	✓
2.)	✓ →	✓	✓	×	× (sign)	✓
3.)	×	✓	✓	✓	✓	×
4.)	×	✓	×	✓	×	×

4) is not memoryless:

$x_1(t) = \mathrm{rect}(t)$

$x_2(t) = \mathrm{rect}(t/2)$



$$y_1\left(\tfrac{1}{4}\right) = \left(\tfrac{1}{4}\right) \underbrace{x_1\left(\tfrac{1}{4}\right)}_{1} + \underbrace{x_1\left(-\tfrac{3}{4}\right)}_{0}$$

$$y_2\left(\tfrac{1}{4}\right) = \left(\tfrac{1}{4}\right) \underbrace{x_2\left(\tfrac{1}{4}\right)}_{1} + \underbrace{x_2\left(-\tfrac{3}{4}\right)}_{1}$$

1) is not T.I.

$x_1(t) = 0$

$x_2(t) = x_1(t-1) = 0$

$y_1(t) = \sin(t+1)$

$y_2(t) = \sin(t+1)$

$y_2(t) \neq y_1(t-1) = \sin(t)$

Linear Time-Invariant Systems:

(LTI Systems)

- Occur frequently in nature
  - Physical laws are time-invariant and often linear (e.g. waves, sound, light)

  When not linear, it can be useful to make linear approximation
  "linearize" using Taylor approximation

- Shift-invariance in optics, convolutional neural networks, etc.

- LTI signal processing is very popular

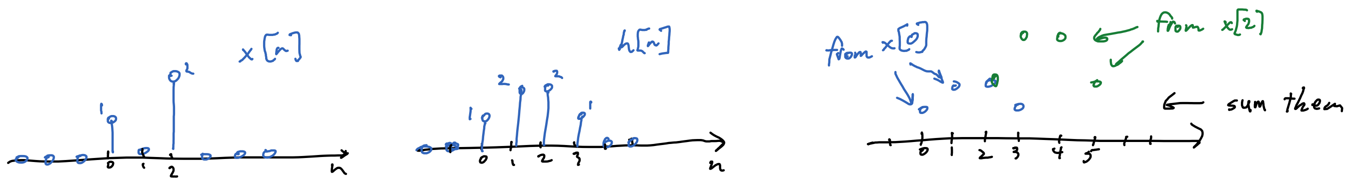
Impulse response:

C.T. $h(t) = \mathcal{H}(\delta(t))$

D.T. $h[n] = \mathcal{H}(\delta[n])$

$h(t)$ is the output when the input is tapped at $t=0$.

If $\mathcal{H}$ is LTI, the impulse response is a fingerprint of the system.



Represent $x[n]$ as sequence of delta functions:

$$
\begin{aligned}
y[n] &= \mathcal{H}(x[n]) \\
&= \mathcal{H}\Big(\sum_k x[k]\,\delta[n-k]\Big) && \text{sifting (even works in C.T.)} \\
&= \sum_k x[k]\,\mathcal{H}(\delta[n-k]) && \text{linearity} \\
&= \sum_k x[k]\,h[n-k] && \text{T.I.} \\
&= x[n] * h[n]
\end{aligned}
$$

LTI = Convolution!

with respect to one input

1.) Convolution is linear and T.I.

2.) Any LTI system does convolution: $\mathcal{H}(x(t)) = x(t) * h(t)$

To understand an LTI system, we only need $h(t)$, how it responds to an impulse. Not true for other systems.