Elementary Functions – P+M 2.1

A lot of signal processing involves re-writing ~~#~~ signals as a ~~functio~~ sum of more easily-manipulated signals (or basis functions)

$$y(n) = \sum_{k=1}^{K} a_k \, x_k(n)$$

↑ coefficient ↑ basis function

WHY?

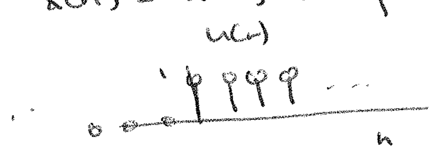
A1) analysis → expand signal as, say, sum of ~~exponentials of~~ sinusoids ~~#~~ at different freq

A2) data compression
pick an expansion so that $y(n)$ can be well approximated by just a few terms; then just keep those.

Choice of ~~exp~~ basis functions / "elementary signals"

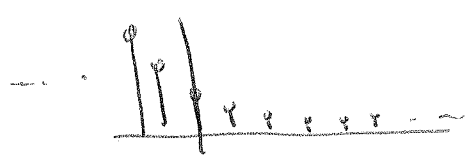
a) unit impulse (delta fcn) $x(n) = \delta(n) = \{\dots, 0, 0, \overset{\downarrow}{1}, 0, 0, \dots\}$

b) unit step (switch turns on) $x(n) = u(n) = \{\dots, 0, 0, \overset{\downarrow}{1}, 1, 1, \dots\}$



c) exponential $x(n) = a^n = \{\dots, a^{-3}, a^{-2}, a^{-1}, \overset{\downarrow}{1}, a, a^2, \dots\}$

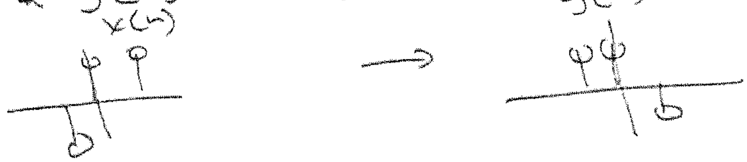
if a is real and $0<a<1$, then



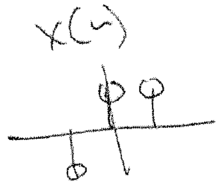
2

ipulation of signals

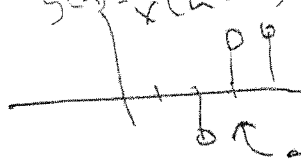
Folding: $y(n) = x(-n)$ (flip in time)



shifting $y(n) = x(n-k)$

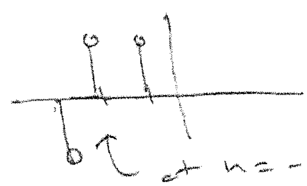


$y(n) = x(n-3)$



at $n=3$, same as "old" $n=0$

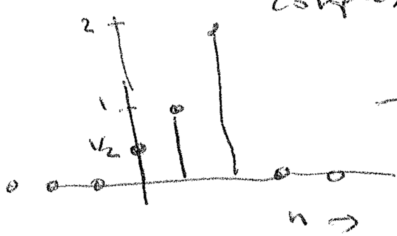
$y(n) = x(n+2)$



at $n=-2$, $n+2=0$ like "old" $x(0)$

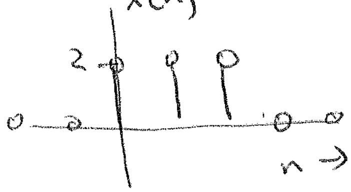
By combining elementary signals w/ folding / shifting (mostly shifting), we can get more complex signals

ex1)



$\rightarrow \{0, 0, \tfrac{1}{2}\delta(n), 1 * \delta(n-1), 2\delta(n-2), 0, 0, \ldots\}$

ex2) What are ways to express the following?
what is most compact



answer 1: $\{\ldots, 0, 0, 2\delta(n), 2\delta(n-1), 2\delta(n-2), \ldots\}$

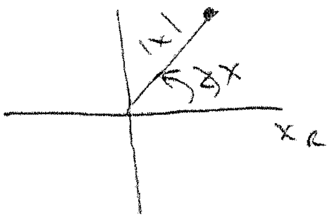
answer 2 (compact)

$$x(n) = 2u(n) - 2u(n-3)$$

usually take a to be complex $\rightarrow$ $a = e^{\ }$, etc.

Quick review of complex #'s

$$x = x_R + j\,\mathcal{X}_I \qquad \text{cartesian} \Rightarrow \text{real + imag}$$



$$= |x|\, e^{j\measuredangle x} \qquad \text{polar} \Rightarrow \text{mag / phase}$$

the phase is 2π periodic, ie

$$e^{j[\measuredangle x]} = e^{j[\measuredangle x + 2\pi]}$$

thus we can define phase 2 ways

'wrapped' phase $\quad ARG[x] \in [-\pi, \pi]$

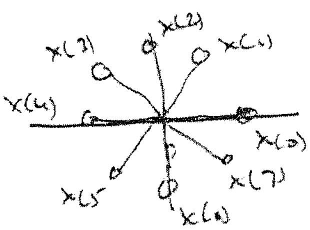
'unwrapped' phase: let go outside $[-\pi, \pi)$ $\quad \arg[x] \in (-\infty, \infty)$

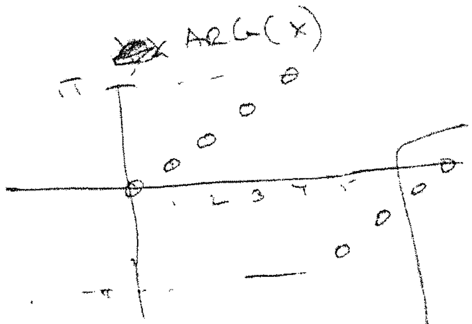
so $\arg = ARG + 2\pi\, r[n]$

? phase "unwrapping" gets from one to another.

ex: $x(n) = a^n = r^n e^{j\theta n}$

$r = 1, \quad \theta = \pi/4$





polar to cartesian / vice versa

$$|x| = \sqrt{x_R^2 + \mathcal{X}_I^2}$$

$$ARG[x] = \arctan\left(\frac{\mathcal{X}_I}{\mathcal{X}_R}\right)$$

why does this work? remember Euler:

$$a^n = r^n e^{j n\theta} = r^n(\cos\theta n + j\sin n\theta)$$

$$\mathcal{X}_I = r^n \sin n\theta$$

$$\mathcal{X}_R = r^n \cos n\theta$$

so $ARG(x(n)) = \arctan\left(\dfrac{\sin n\theta}{\cos n\theta}\right)$

$$= \arctan(\tan(n\theta)) = n\theta$$

Section 2.2 DT System properties

$$x(n) \longrightarrow \boxed{H} \longrightarrow y(n) = H[x(n)]$$

input system output

1) **Linear**: if $x_1 \to y_1$ $x_2 \to y_2$

then $x_3 = a x_1 + b x_2 \Rightarrow y_3 = a y_1 + b y_2$

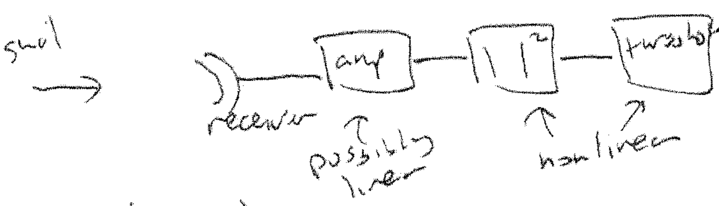
scale/add input $\Rightarrow$ scaled/added outputs

ex nonlinear: threshold, squaring

2) shift-invariant
or time-invariant

delay input = delay output

if $x(n) \to y(n)$, then $x(n-k) \to y(n-k)$



Causal : output depends on past + current inputs

$$y(n) = F(x(n), x(n-1), \ldots)$$

BIBO stable : bounded input → bounded output

i.e. if $|x(n)| \leq M_x < \infty \quad \forall n$

then $|y(n)| < M_y < \infty \quad \forall n$

concept: impulse response $h[n]$. Response to impulse $\delta(n)$

☆ thought experiment : hit a bell w/ a hammer. which of these properties does it exhibit

---

Convolution $x \rightarrow \boxed{h} \rightarrow y$

$\boxed{\text{Key result for LTI systems Linear Time-invariant}}$

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

"shorthand"

What this says: Knowing the impulse response, and knowing the system is LTI, we can calculate response to <u>any</u> input.

kind of remarkable.

We'll look at this ~~two~~ 3 ways:
- graphical "derivation" for insight
- calculation method
- math derivation

First, note the sum above is $\sum_{-\infty}^{\infty}$

→ if $h(n)$ really non-zero for ∞ time we can't store on computer (IIR)

→ if $h(n)$ is finite length (FIR), we <u>can</u> do calculation