Obj

Discrete Time Signals

- Represented ✓
- Sampling
- Operations
- MATLAB

Mitra Text

2 Time domain
3 Freq domain
4 Filters what they do
5 Sampling
6,7 Filters how to make ⇒MATLAB
8 Algorithms ⇒MATLAB
9-12 Graduate level applications

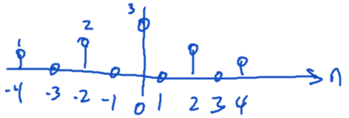
Representation

$x(t)$

1. Notational $x[n]$ $h[n]$ square brackets, n integer

$\underbrace{\sum_{n=-\infty}^{\infty} h[n] < \infty}_{\text{stable}}$ $\underbrace{\int_{-\infty}^{\infty} h(t)\,dt < \infty}_{\text{stable}}$

2. Graphical - stem



MATLAB: >> x = [1 0 2 0 3 0 2 0 1]
>> n = ~~[-4 -3~~ -2
>> n = -4:4;
>> stem(n, x)

what is x[1.2] ?= Does not exist

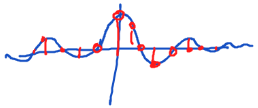
3. Sequence

[1 0 2 0 3 0 2 0 1]
↑ corresponds to n = 0

Sequence Lengths

If $x[n]$ is zero except for $N_1 \le n \le N_2$ then

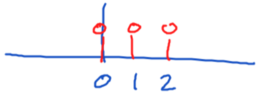
a) $N_1 = -\infty$ $N_2 = \infty$ two-sided signal
must define as eqn $x[n] = \begin{cases} 1 & n = 0 \\ \frac{\sin(n)}{n} & n \neq 0 \end{cases}$ "sinc"



b) if N_1 finite right-sided signal

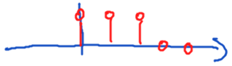
c) if N_2 finite left-sided signal

d) if N_1 & N_2 finite finite length sequence
ex x = [1 1 1]
n = [0 1 2]
Length N = 3



? Length 3

e) zero padding
explicitly add zeros (right) side of finite length sequence



x = [1 1 1 0 0]
n = [0 1 2 3 4]
Length 5

x = [0 0 1 1 1]
n = [0 1 2 3 4]

n = [-2 -1 0 1 2]

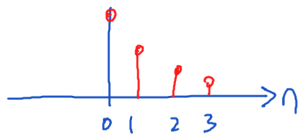
f) Causal $N_1 \ge 0$
"zero before zero"

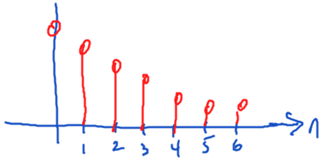
g) Anticausal $N_2 \le 0$

Sampling CT → DT

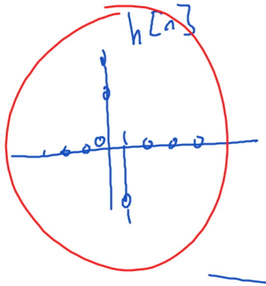
Sample $T_s = \frac{1}{2}$ s

Sample $T_s = 1$



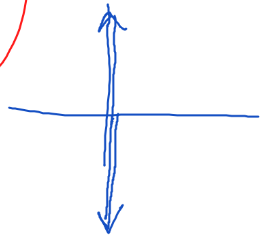


faster sampling → "stretches" more
(more detail / "zoom in")



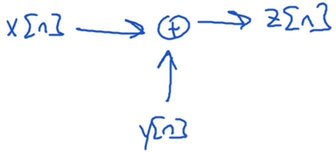
$\frac{d}{dt} \quad \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}$

$h(t) = \dot{\delta}(t)$



Operations

a. add



ex $x = [4\ 2\ 9\ 8]$

$y = [6\ 2\ -1\ -1]$

$z = [6\ 6\ 1\ 8\ 8]$

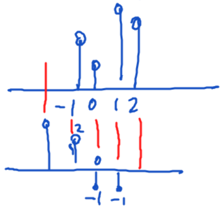
n=0

x   4 2 9 8

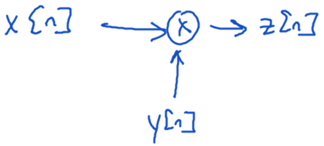
y 6 2 -1 -1

6 6 1 8 8

```
>> x = [0 4 2 9 8]
>> y = [6 2 -1 -1 0]
>> n = -2:2
>> z = x+y
>> stem(n,z)
```



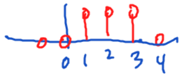
b. multiplication "modulation" 2 signals



c. mult. coef.

d. delay $x[n] \to \boxed{z^{-1}} \to y[n]$

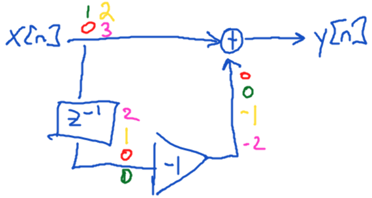
current output = previous input



e. advance $x[n] \to \boxed{z} \to y[n]$

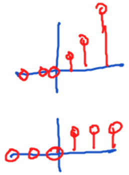
Example

Differentiator



$x = [0\ 1\ 2\ 3\ 4\ 5\ \dots]$

$y = [0\ 1\ 1\ 1\ 1\ 1\ 1\ \dots]$



Down sampling

$x[n] \to \boxed{\downarrow M} \to y[n]$