
ELEN E4810: Digital Signal Processing

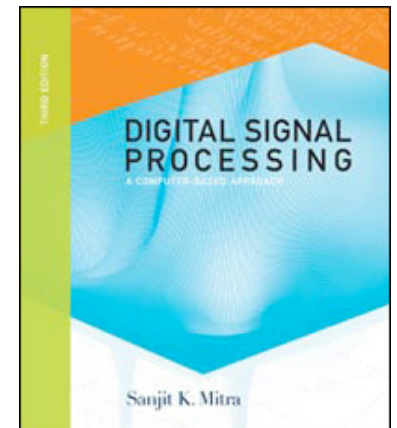
Topic 1: Introduction

1. Course overview
2. Digital Signal Processing
3. Basic operations & block diagrams
4. Classes of sequences



1. Course overview

- **Digital signal processing:**
Modifying signals with computers
- Web site:
<http://www.ee.columbia.edu/~dpwe/e4810/>
- Book:
Mitra “Digital Signal Processing”
(3rd ed., 2005)
- Instructor: dpwe@ee.columbia.edu



Grading structure

- Homeworks: 20%
 - Mainly from Mitra
 - Tuesday-tuesday schedule
 - Collaborate, don't copy
- Midterm: 20%
 - One session
- Final exam: 30%
- Project: 30%



Course project

- Goal: hands-on experience with DSP
- Practical implementation
- Work in pairs or alone
- Brief report, optional presentation
- Recommend MATLAB
- Ideas on website



Example past projects

on web site

- Solo Singing Detection
- Guitar Chord Classifier
- Speech/Music Discrimination
- DTMF decoder
- Reverb algorithms
- Compression algorithms

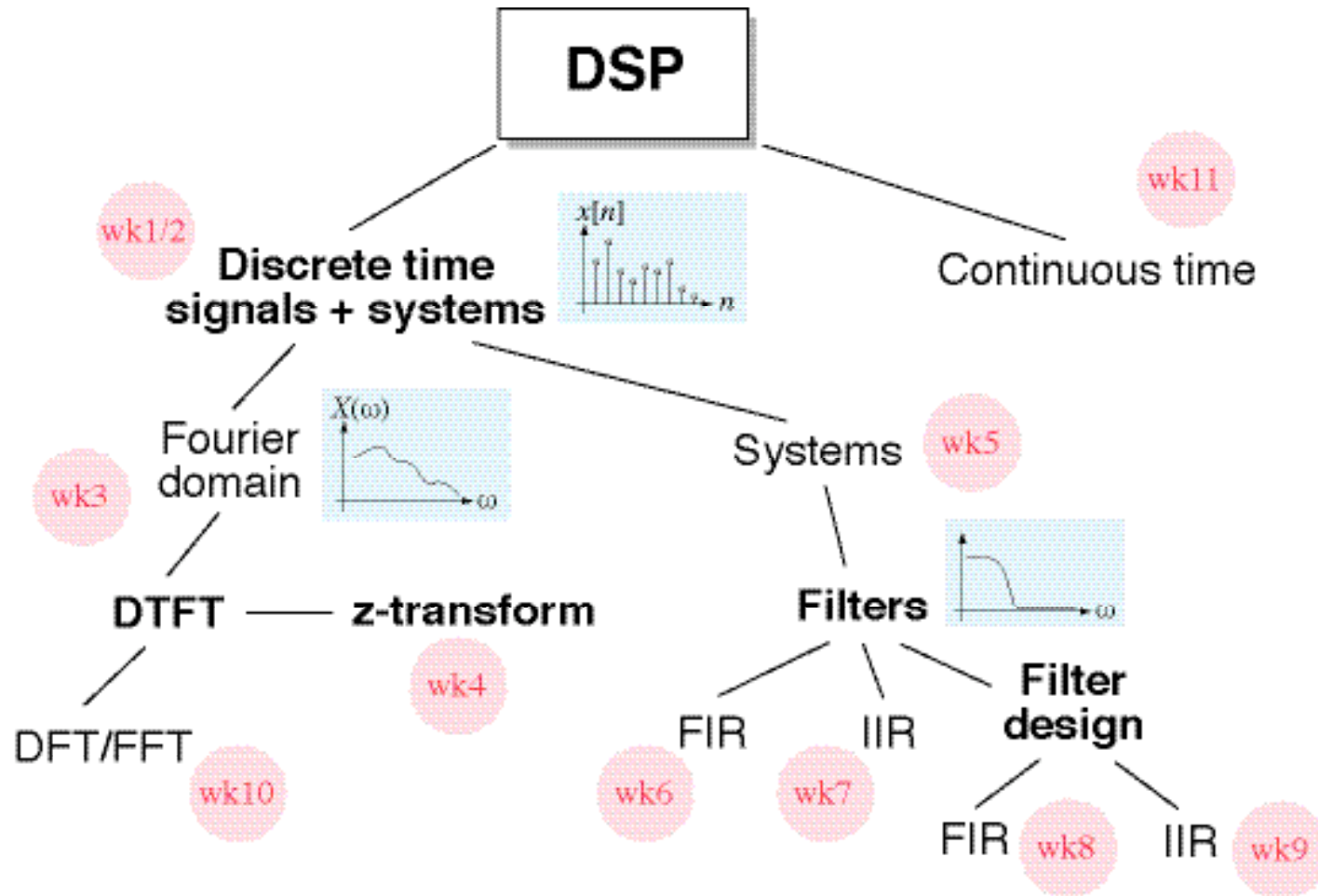


MATLAB

- Interactive system for numerical computation
- Extensive signal processing library
- Focus on **algorithm**, not implementation
- Access:
 - Engineering Terrace 251 computer lab
 - Student Version (need Sig. Proc. toolbox)
 - ILAB, 1235 Mudd (email TA)



Course at a glance



2. Digital Signal Processing

- Signals:
Information-bearing function
- E.g. sound: air pressure variation at a point as a function of time $p(t)$
- Dimensionality:
Sound: 1-Dimension
Greyscale image $i(x,y)$: 2-D
Video: 3 x 3-D: $\{r(x,y,t) \ g(x,y,t) \ b(x,y,t)\}$



Example signals

- Noise - all domains
- Spread-spectrum phone - radio
- ECG - biological
- Music
- Image/video - compression
-



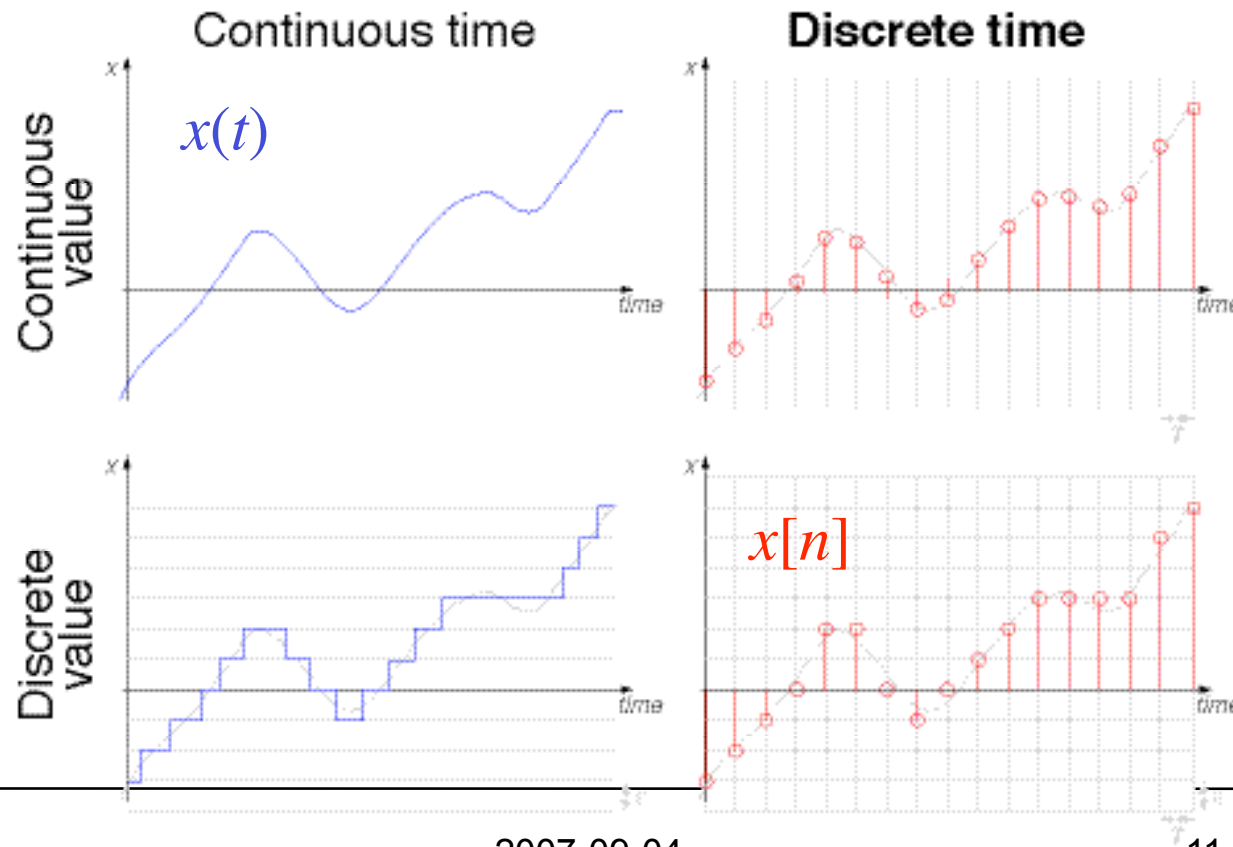
Signal processing

- Modify a signal to extract/enhance/rearrange the information
- Origin in analog electronics e.g. radar
- Examples...
 - Noise reduction
 - Data compression
 - Representation for recognition/classification...



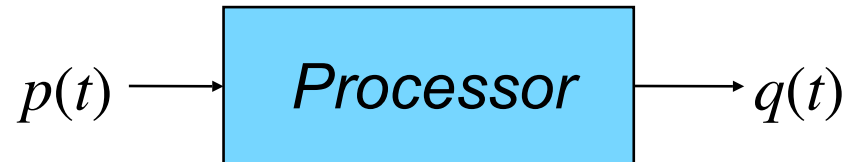
Digital Signal Processing

- DSP = signal processing on a computer
- Two effects: discrete-time, discrete level



DSP vs. analog SP

- Conventional signal processing:



- Digital SP system:



Digital vs. analog

■ Pros

- Noise performance - quantized signal
- Use a general computer - flexibility, upgrade
- Stability/duplicability
- Novelty

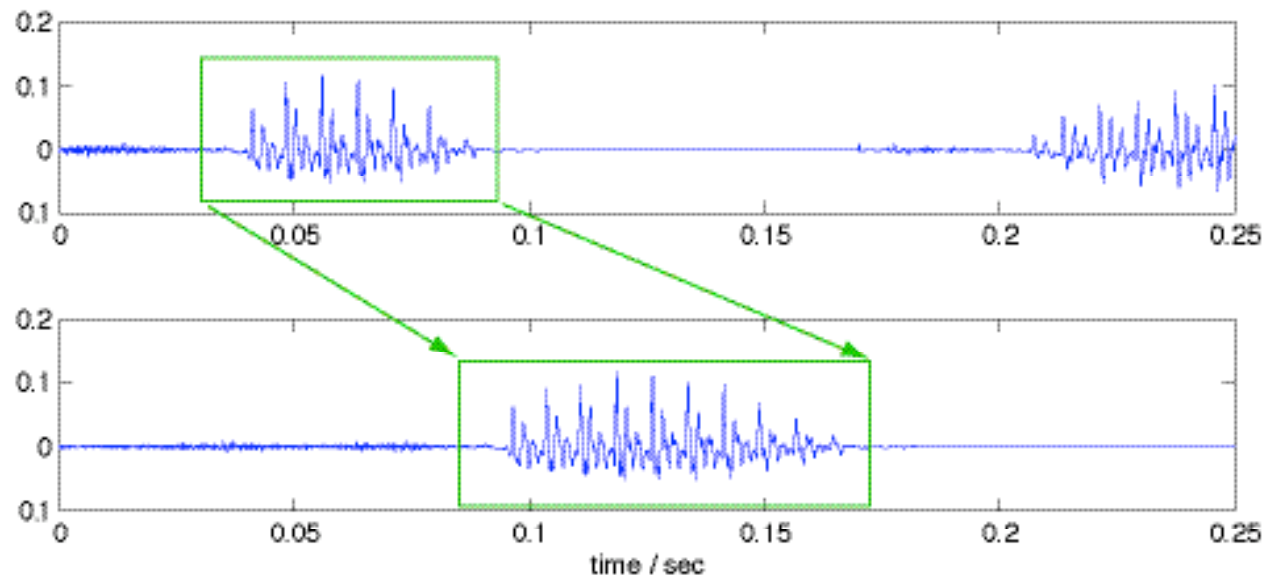
■ Cons

- Limitations of A/D & D/A
- Baseline complexity / power consumption



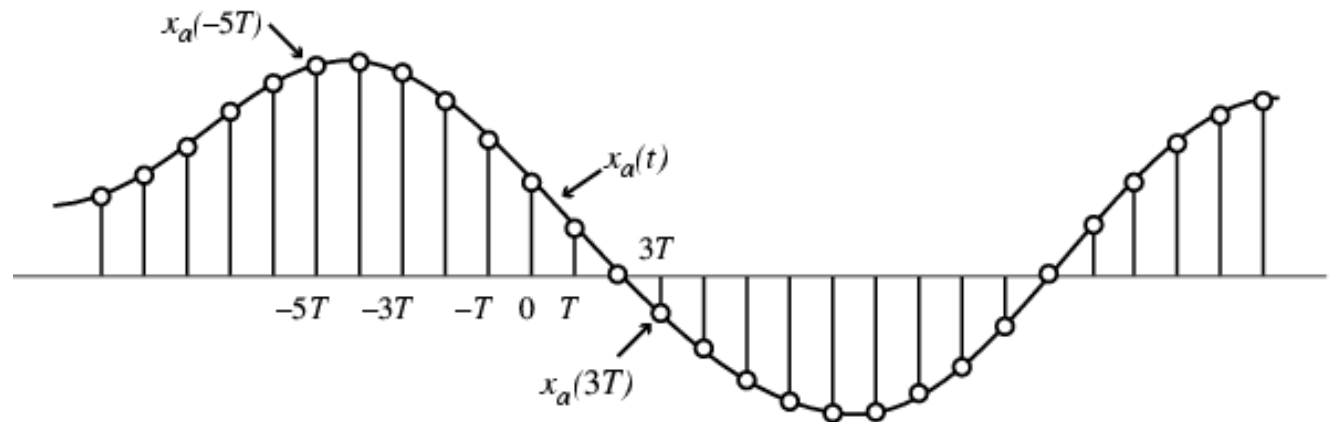
DSP example

- Speech time-scale modification: extend duration without altering pitch



3. Operations on signals

- Discrete time signal often obtained by **sampling** a continuous-time signal



- Sequence $\{x[n]\} = x_a(nT)$, $n = \dots -1, 0, 1, 2, \dots$
- $T =$ samp. period; $1/T =$ samp. frequency



Sequences

- Can write a sequence by listing values:

$$\{x[n]\} = \{K, -0.2, 2.2, 1.1, 0.2, -3.7, 2.9, K\}$$

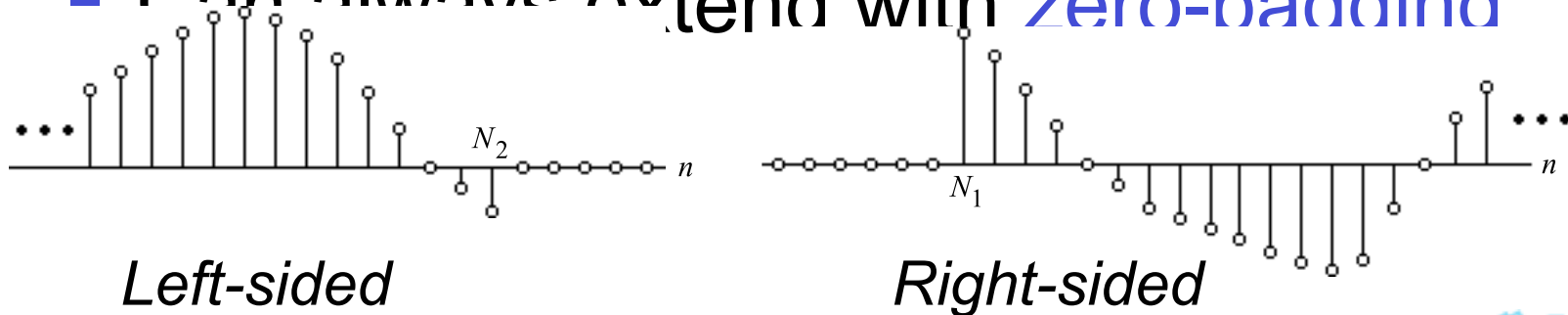
↑

- Arrow indicates where $n=0$
- Thus, $x[-1] = -0.2$, $x[0] = 2.2$, $x[1] = 1.1$,



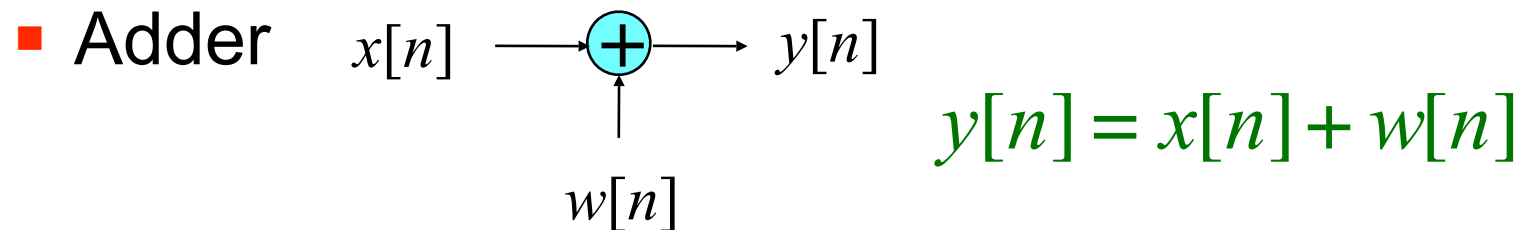
Left- and right-sided

- $x[n]$ may be defined **only** for certain n :
 - $N_1 \leq n \leq N_2$: **Finite length** (length = ...)
 - $N_1 \leq n$: **Right-sided** (**Causal** if $N_1 \geq 0$)
 - $n \leq N_2$: **Left-sided** (Anticausal)
- Can always extend with **zero-padding**

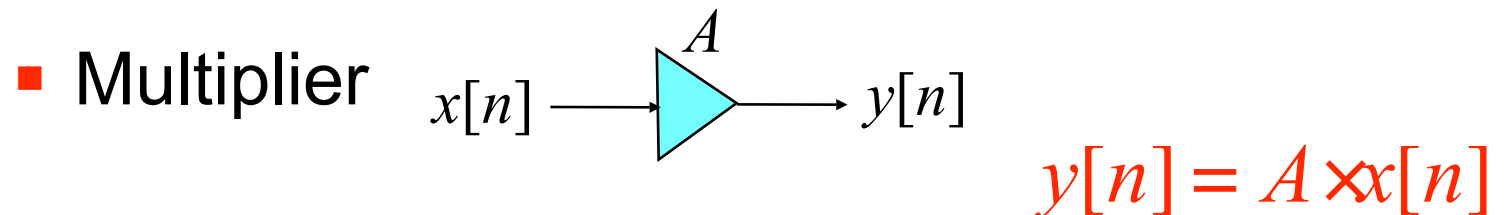


Operations on sequences

- **Addition operation:**



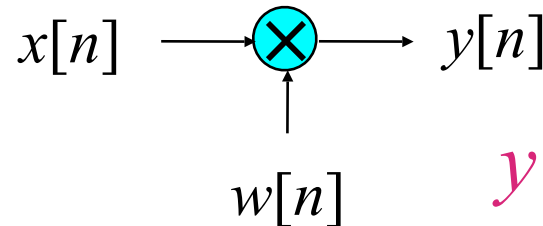
- **Multiplication operation**



More operations

- **Product (modulation) operation:**

- Modulator



$$y[n] = x[n] \times w[n]$$

- E.g. **Windowing**: multiplying an infinite-length sequence by a finite-length **window** sequence to extract a region



Time shifting

- **Time-shifting** operation: $y[n] = x[n - N]$
where N is an integer

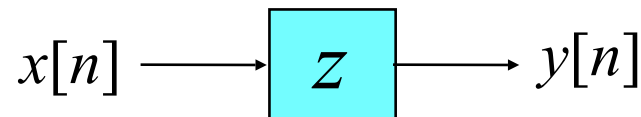
- If $N > 0$, it is **delaying** operation

- Unit delay $x[n] \longrightarrow \boxed{z^{-1}} \longrightarrow y[n]$

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

- If $N < 0$, it is an **advance** operation

- Unit advance

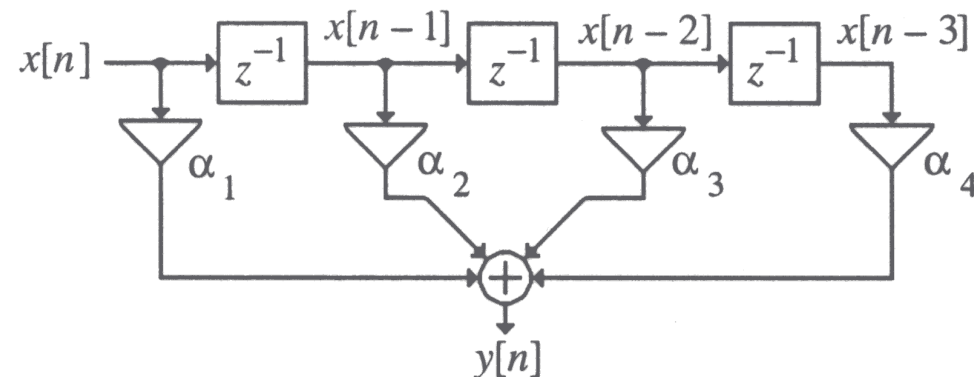


$$y[n] = x[n + 1]$$



Combination of basic operations

- Example



$$y[n] = \alpha_1 x[n] + \alpha_2 x[n-1] + \alpha_3 x[n-2] + \alpha_4 x[n-3]$$



Up- and down-sampling

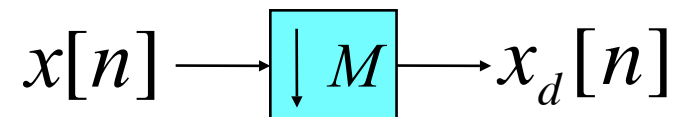
- Certain operations change the effective **sampling rate** of sequences by adding or removing samples
- Up-sampling = adding more samples
= **interpolation**
- Down-sampling = discarding samples
= **decimation**



Down-sampling

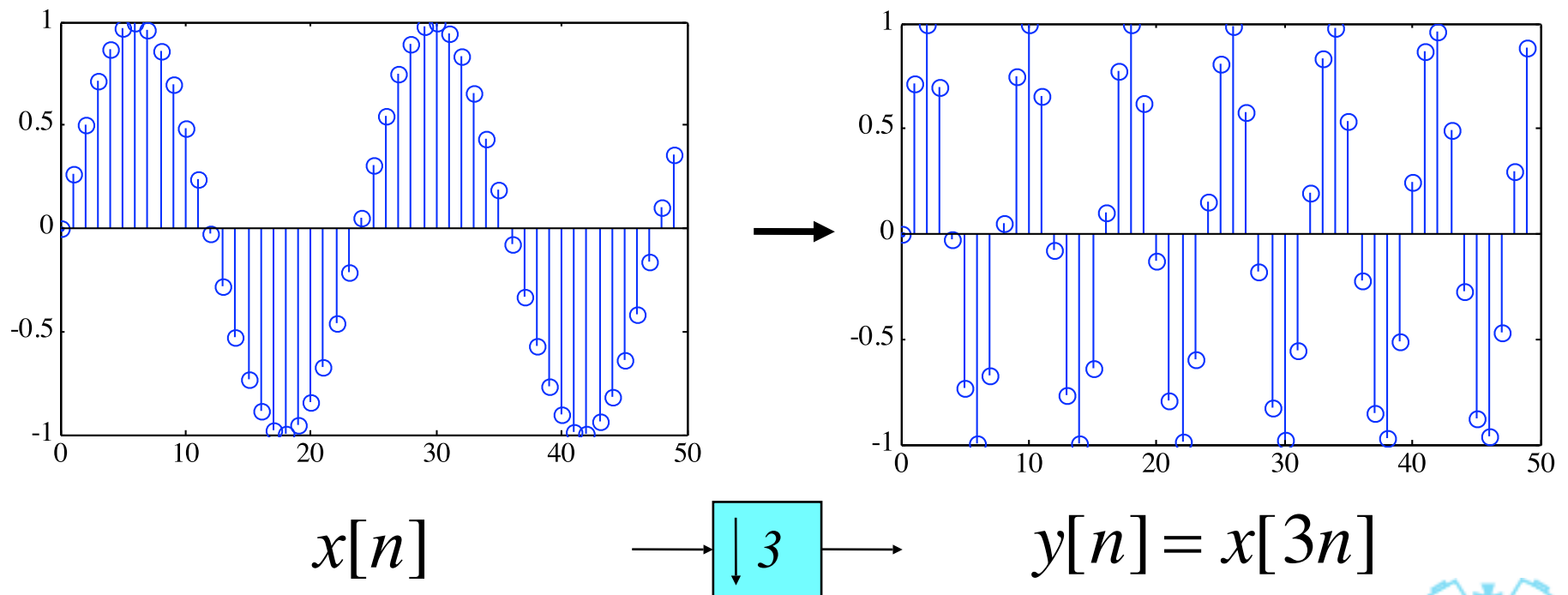
- In **down-sampling** by an integer factor $M > 1$, every M -th sample of the input sequence is kept and $M - 1$ in-between samples are removed:

$$x_d[n] = x[nM]$$



Down-sampling

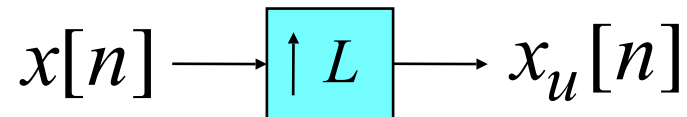
- An example of down-sampling



Up-sampling

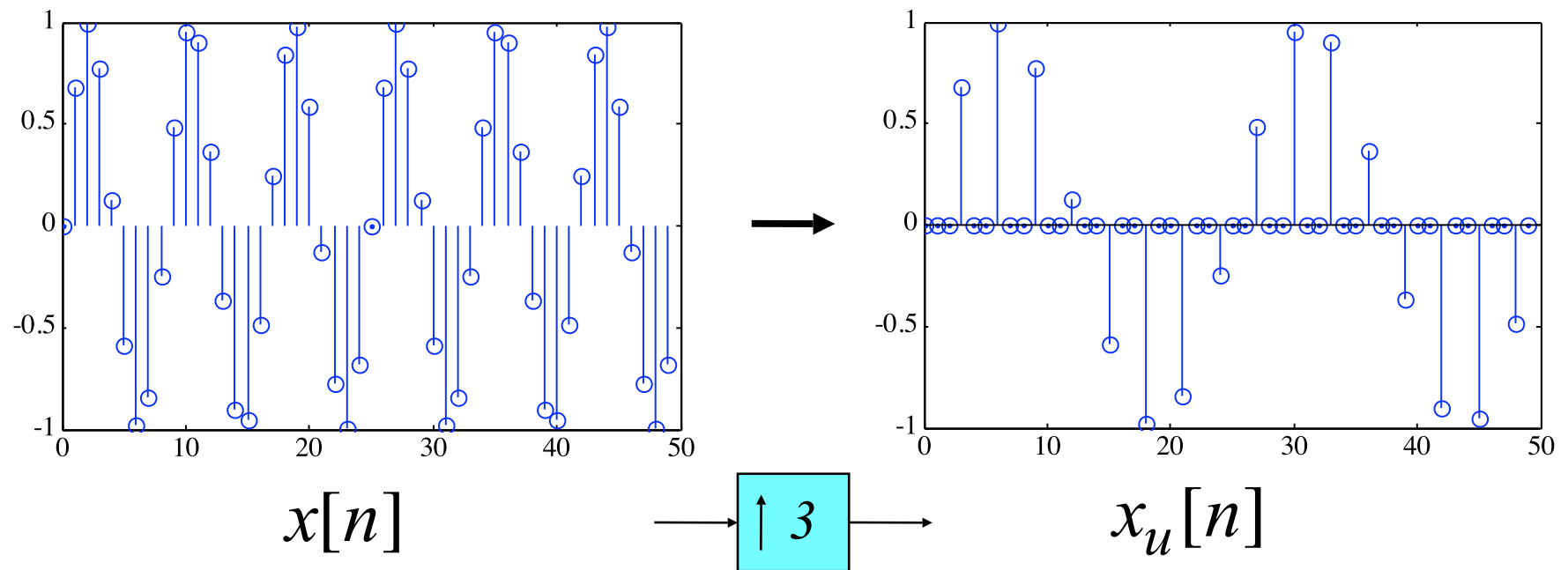
- Up-sampling is the converse of down-sampling: $L-1$ zero values are inserted between each pair of original values.

$$x_u[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \dots \\ 0, & \text{otherwise} \end{cases}$$



Up-sampling

- An example of up-sampling



not inverse of downsampling!

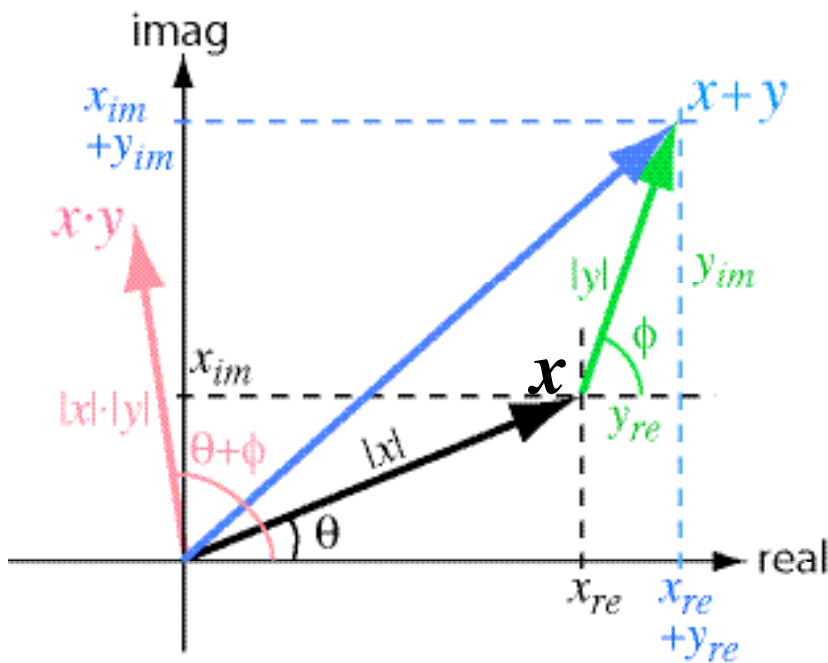


Complex numbers

- .. a mathematical convenience that lead to simple expressions
- A second “imaginary” dimension ($j \equiv \sqrt{-1}$) is added to all values.
- **Rectangular** form: $x = x_{re} + j \cdot x_{im}$
where *magnitude* $|x| = \sqrt{(x_{re}^2 + x_{im}^2)}$
and *phase* $\theta = \tan^{-1}(x_{im}/x_{re})$
- **Polar** form: $x = |x| e^{j\theta} = |x| \cos \theta + j \cdot |x| \sin \theta$
($e^{j\theta} = \cos \theta + j \sin \theta$)



Complex math



- When **adding**, real and imaginary parts add: $(a+jb) + (c+jd) = (a+c) + j(b+d)$
- When **multiplying**, magnitudes multiply and phases add: $re^{j\theta} \cdot se^{j\phi} = rse^{j(\theta+\phi)}$
- Phases modulo 2π

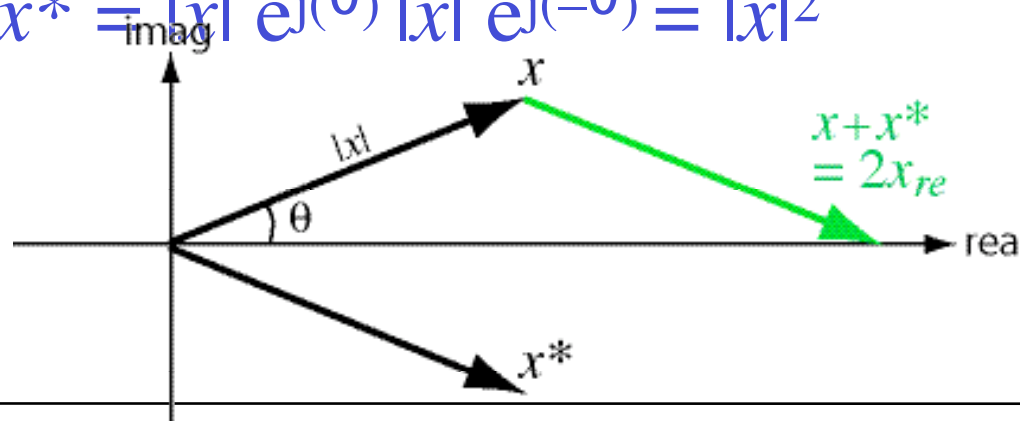


Complex conjugate

- Flips imaginary part / negates phase:
conjugate $x^* = x_{re} - j \cdot x_{im} = |x| e^{j(-\theta)}$
- Useful in resolving to real quantities:

$$x + x^* = x_{re} + j \cdot x_{im} + x_{re} - j \cdot x_{im} = 2x_{re}$$

$$x \cdot x^* = |x| e^{j(\theta)} |x| e^{j(-\theta)} = |x|^2$$



Classes of sequences

- Useful to define broad categories...
- Finite/infinite (extent in n)
- Real/complex:

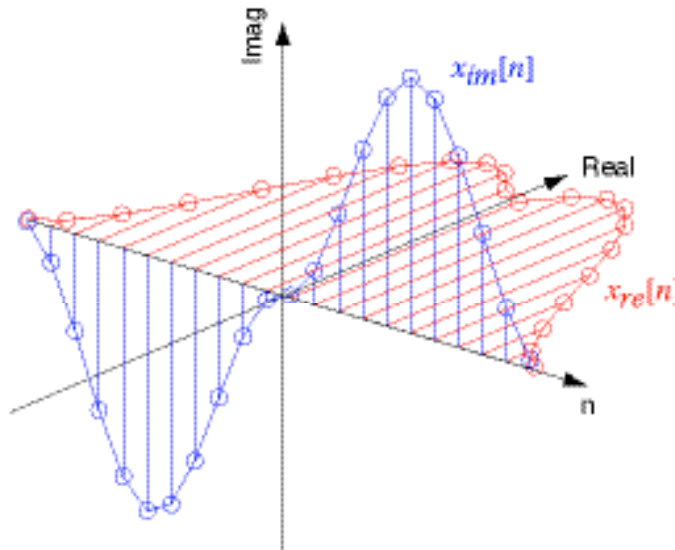
$$x[n] = x_{re}[n] + j \cdot x_{im}[n]$$



Classification by symmetry

- **Conjugate symmetric** sequence:

$$x_{cs}[n] = x_{cs}^*[-n] = x_{re}[-n] - j \cdot x_{im}[-n]$$



- **Conjugate antisymmetric**:

$$x_{ca}[n] = -x_{ca}^*[-n] = -x_{re}[-n] + j \cdot x_{im}[-n]$$



Conjugate symmetric decomposition

- Any sequence can be expressed as conjugate symmetric (CS) / antisymmetric (CA) parts:

$$x[n] = x_{cs}[n] + x_{ca}[n]$$

where:

$$x_{cs}[n] = 1/2(x[n] + x^*[-n]) = x_{cs}^*[-n]$$

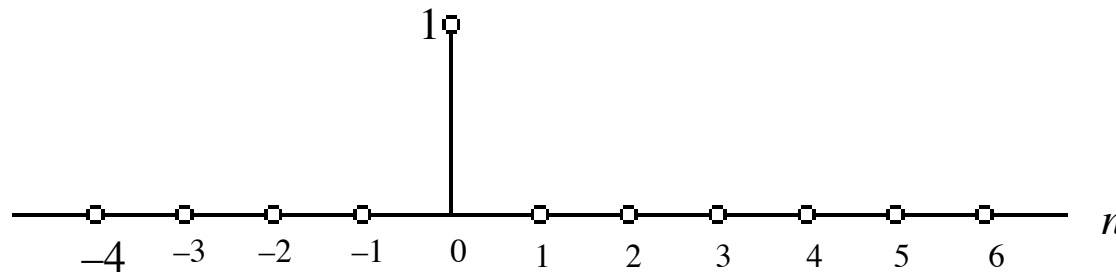
$$x_{ca}[n] = 1/2(x[n] - x^*[-n]) = -x_{ca}^*[-n]$$

- When signals are **real**,
CS $\rightarrow$ Even ($x_{re}[n] = x_{re}[-n]$), CA $\rightarrow$ Odd

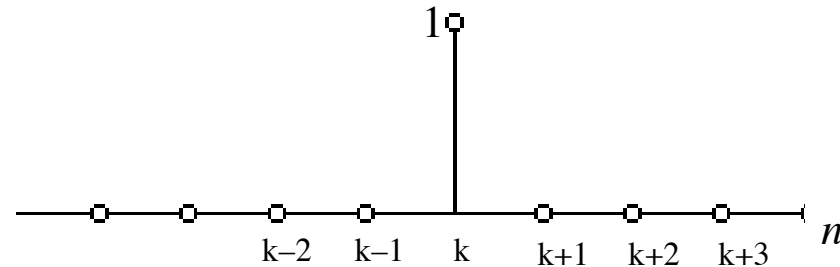


Basic sequences

- **Unit sample** sequence: $\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$



- Shift in time:
 $\delta[n - k]$



- Can express any sequence with δ :

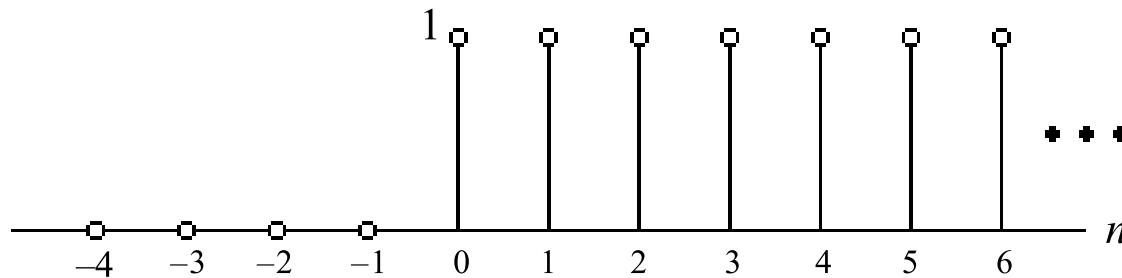
$$\{\alpha_0, \alpha_1, \alpha_2, \dots\} = \alpha_0 \delta[n] + \alpha_1 \delta[n-1] + \alpha_2 \delta[n-2] \dots$$

$\uparrow$



More basic sequences

- **Unit step** sequence: $\mu[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$



- Relate to unit sample:

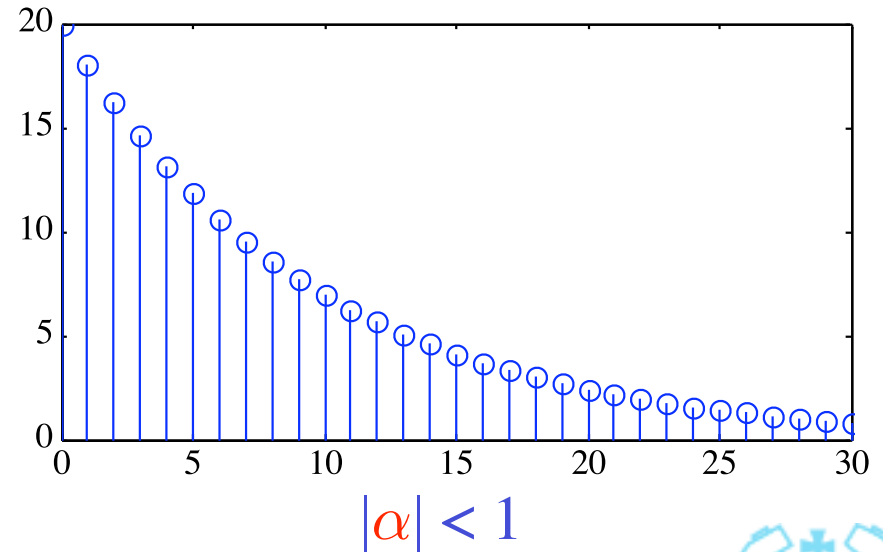
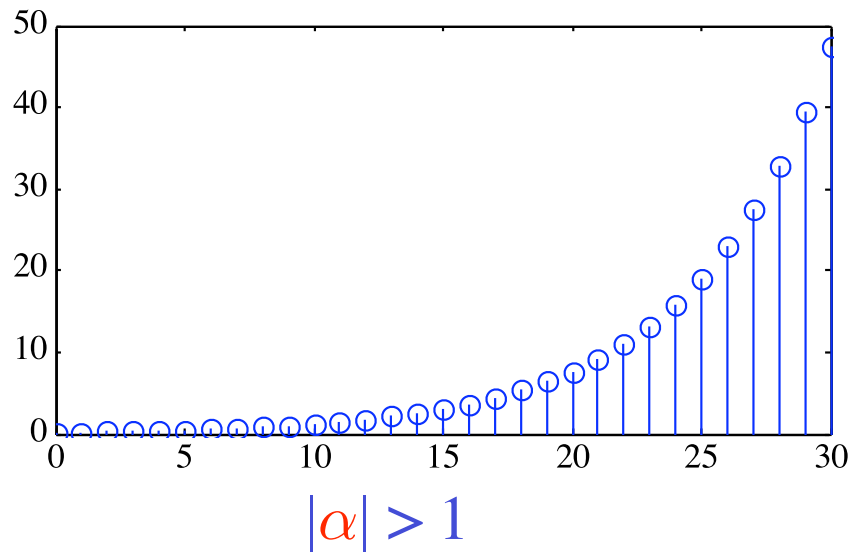
$$\delta[n] = \mu[n] - \mu[n-1]$$

$$\mu[n] = \sum_{k=-\infty}^n \delta[k]$$



Exponential sequences

- Exponential sequences= *eigenfunctions*
- General form: $x[n] = A \cdot \alpha^n$
- If A and α are *real* (and positive):



Complex exponentials

$$x[n] = A \cdot \alpha^n$$

- Constants A , α can be complex :

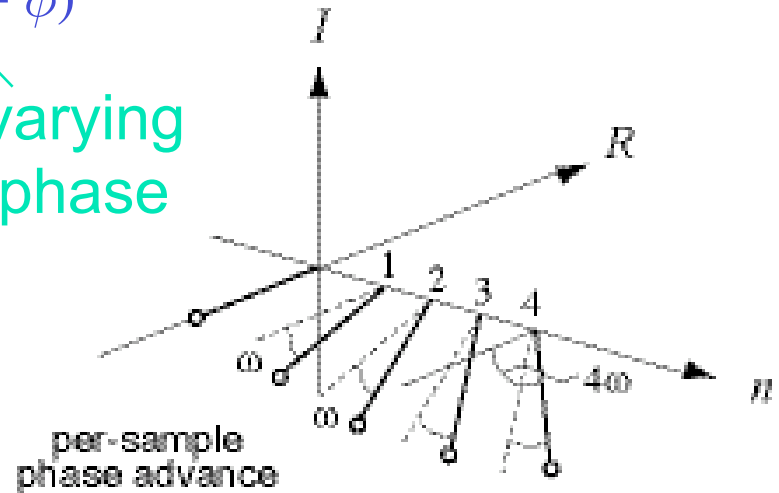
$$A = |A|e^{j\phi} ; \quad \alpha = e^{(\sigma + j\omega)}$$

$$\rightarrow x[n] = |A| e^{\sigma n} e^{j(\omega n + \phi)}$$

scale

varying
magnitude

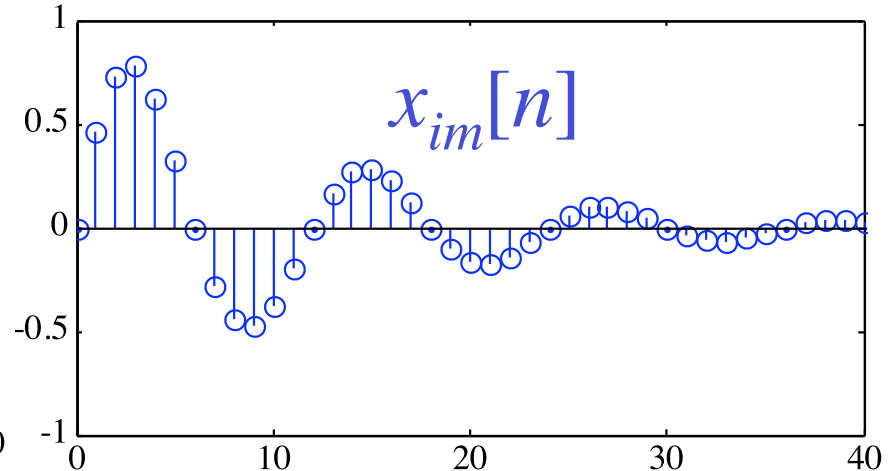
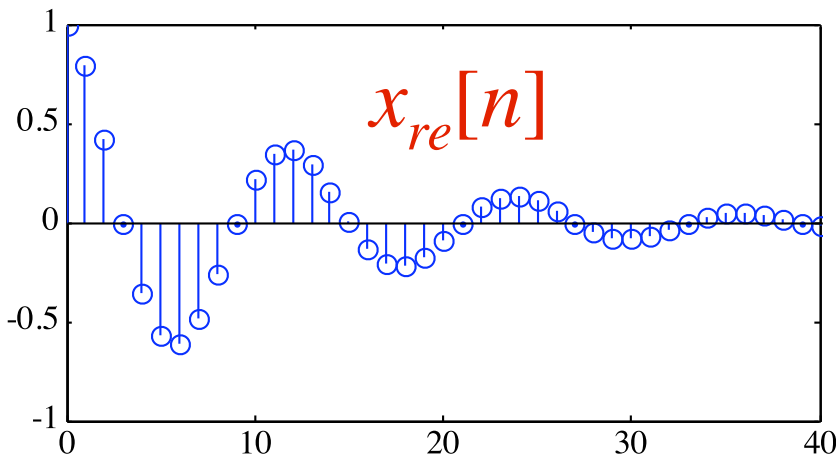
varying
phase



Complex exponentials

- Complex exponential sequence can 'project down' onto real & imaginary axes to give sinusoidal sequences

$$x[n] = \exp\left\{\left(-\frac{1}{12} + j\frac{\pi}{6}\right)n\right\} \quad e^{j\theta} = \cos\theta + j\sin\theta$$



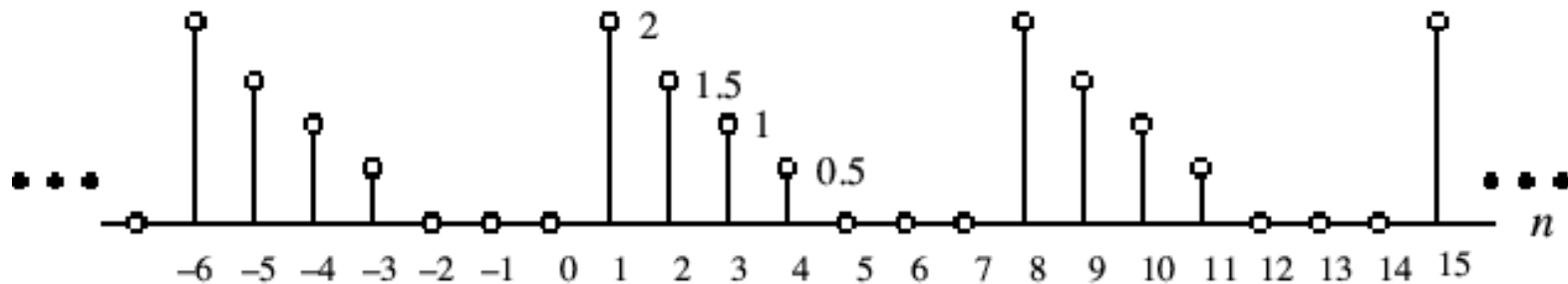
$$x_{re}[n] = e^{-n/12} \cos(\pi n/6) \quad x_{im}[n] = e^{-n/12} \sin(\pi n/6)$$



Periodic sequences

- A sequence $\tilde{x}[n]$ satisfying $\tilde{x}[n] = \tilde{x}[n + kN]$, is called a **periodic sequence** with a **period** N where N is a positive integer and k is any integer.

Smallest value of N satisfying $\tilde{x}[n] = \tilde{x}[n + kN]$ is called the **fundamental period**



Periodic exponentials

- Sinusoidal sequence $A \cos(\omega_o n + \phi)$ and complex exponential sequence $B \exp(j\omega_o n)$ are periodic sequences of period N **only if**
 $\omega_o N = 2\pi r$ with N & r positive **integers**
- Smallest value of N satisfying $\omega_o N = 2\pi r$ is the **fundamental period** of the sequence
- $r = 1 \rightarrow$ one sinusoid cycle per N samples
 $r > 1 \rightarrow r$ cycles per N samples



Symmetry of periodic sequences

- An N -point finite-length sequence $x_f[n]$ defines a periodic sequence:

$$x[n] = x_f[\langle n \rangle_N] \quad \text{“}n \text{ modulo } N\text{”}$$

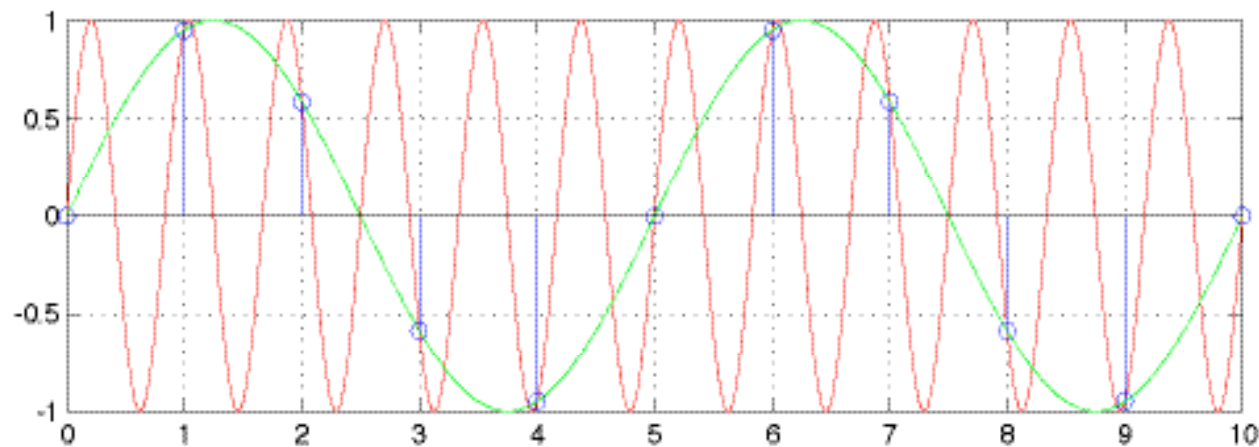
- Symmetry of $x_f[n]$ is not defined because $x_f[n]$ is undefined for $n < 0$
- Define **Periodic Conjugate Symmetric**:

$$\begin{aligned} x_{pcs}[n] &= 1/2(x[n] + x^*[\langle -n \rangle_N]) \\ &= 1/2(x_f[n] + x_f^*[N - n]) \quad 1 \leq n \leq N \end{aligned}$$



Sampling sinusoids

- Sampling a sinusoid is *ambiguous*:



$$x_1[n] = \sin(\omega_0 n)$$

$$x_2[n] = \sin((\omega_0 + 2\pi r)n) = \sin(\omega_0 n) = x_1[n]$$



Aliasing

- E.g. for $\cos(\omega n)$, $\omega = 2\pi r \pm \omega_0$
all r appear the same after sampling
- We say that a larger ω appears **aliased** to a lower frequency
- **Principal value** for discrete-time frequency: $0 \leq \omega_0 \leq \pi$ (i.e. less than one-half cycle per sample)

