Summary (1/2)

- *Characterization and representation of discrete signals*
  - *Types of signals*
  - *Discrete-time signals*
    - *representation of a discrete-time signal*
    - *basic discrete-time signals*

- *Characterization and representation of discrete systems*
  - *Properties of discrete-time systems*
  - *Linear time-invariant systems* (LTI)
    - *response to a discrete-time input*
    - *Discrete-time convolution*
    - *properties of LTI systems*
    - *FIR and IIR systems*
    - *(linear) difference equations with constant coefficients*

Summary (2/2)

- *Basic signal properties*
  - *Continuous versus discrete*
  - *Periodic versus aperiodic*
  - *Deterministic versus random*
  - *Energy versus power*
  - *Sinusoidal sequences with a prescribed SNR*

- *the auto-correlation and the cross-correlation*
  - *concept and meaning*
  - *definition of the auto-correlation*
  - *definition of the cross-correlation*
  - *auto-correlation and cross-correlation examples*
  - *auto-correlation and cross-correlation properties*

Characterization and representation of discrete signals

- types of signals and systems

  - **continuous-time signal** (or analog) = continuous function of independent continuous-time variables

  - **continuous-time systems**: those whose inputs and outputs are continuous

  - **discrete-time signal** = continuous function of independent discrete-time variables

  - **digital signal**: discrete function of independent discrete-time variables

  - **digital systems**: those whose inputs and outputs are discrete

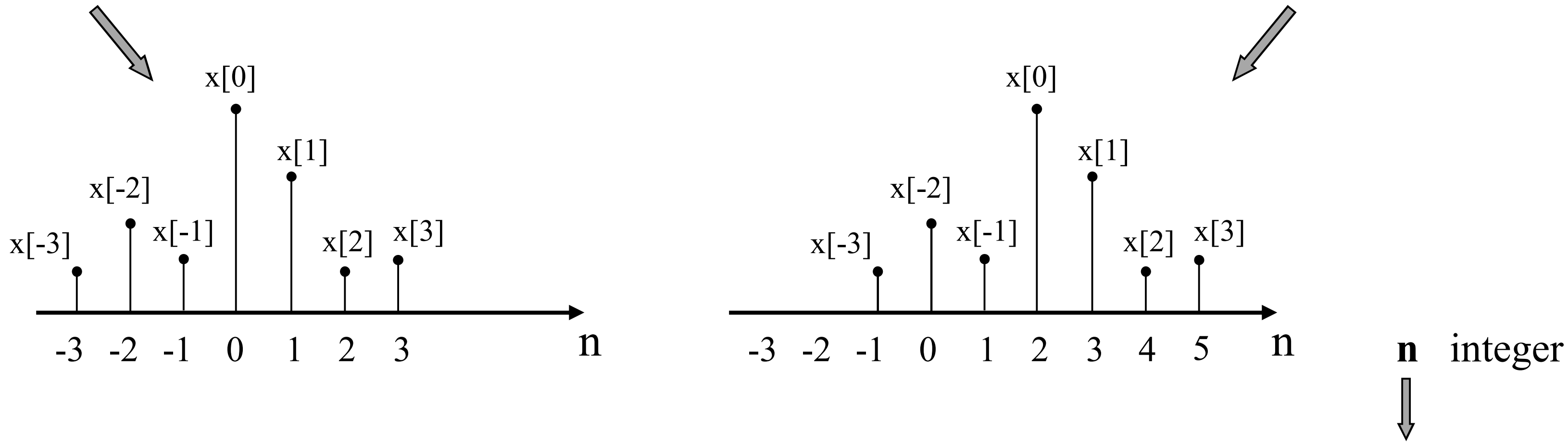
Despite the fact that digital signal processing presumes digital (*i.e.,* quantized) signals, we will focus mainly on discrete signals and systems and will address quantization separately, when that is required.

Characterization and representation of discrete signals

- representation of a discrete-time signal

$x[n], \quad n \in \;]-\infty, \dots, -2, -1, 0, 1, 2, \dots, +\infty[$

x[n-2]



x[n] represents symbolically the ENTIRE discrete signal for $-\infty < n < +\infty$

taking a specific value of n, for example n=2, then x[2] represents the magnitude of the sample of x[n] at position 2 in the sequence of numbers

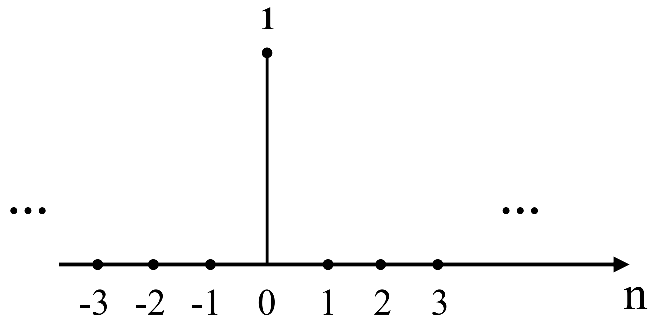
$x[n-n_0]$ represents x[n] delayed by n_0 samples

Characterization and representation of discrete signals

- basic discrete-time signals

<u>unit impulse</u> (and not “Dirac” impulse! – what is the difference ?)

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$



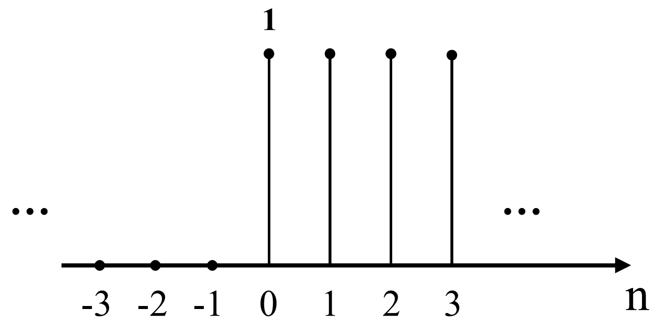
– this signal is very important for various reasons, including the possibility to express any discrete-time sequence as a sum of scaled, delayed impulses :

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

Characterization and representation of discrete signals

unit step

$$u[n]=\begin{cases}1, & n\geq 0\\ 0, & n<0\end{cases}$$



sum of delayed impulses

– it is possible to write:

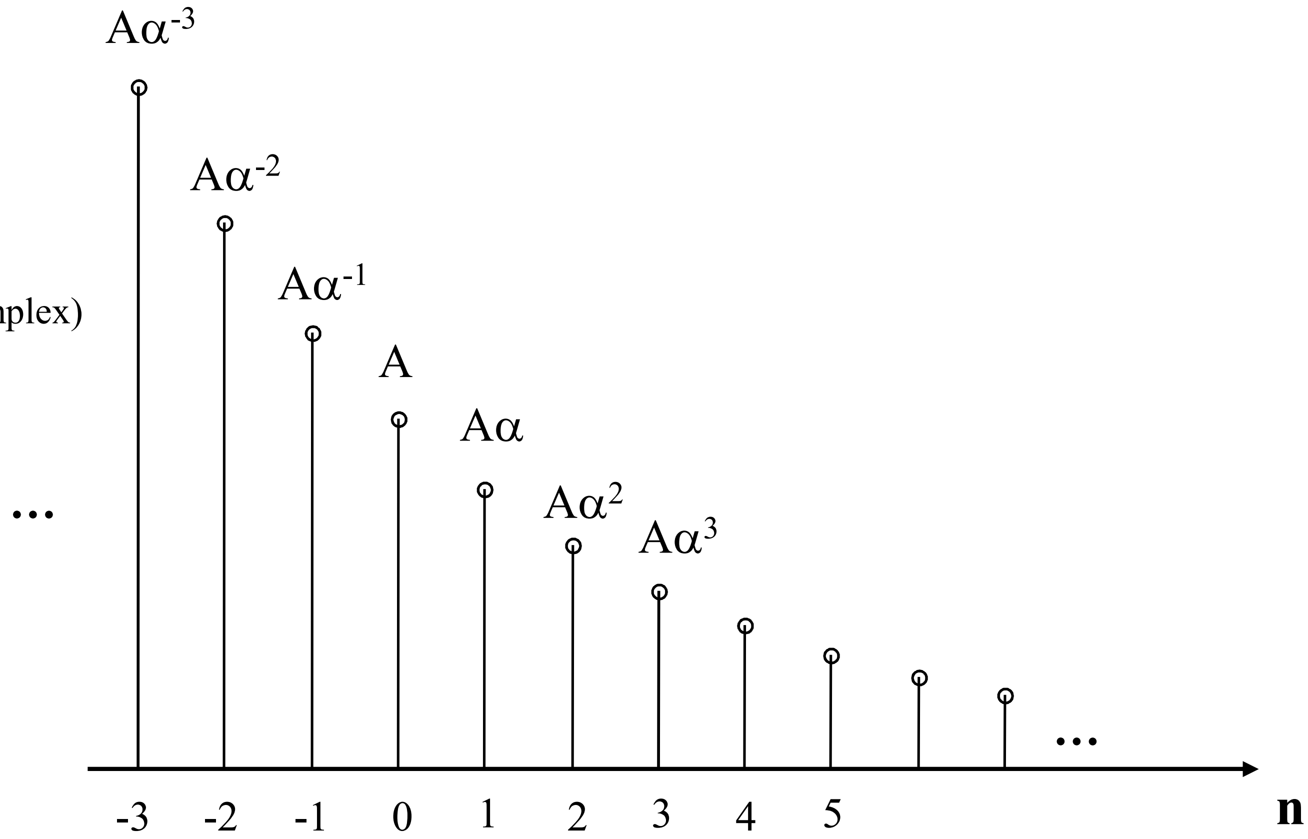
$$u[n]=\sum_{k=-\infty}^{n}\delta[k]=\sum_{k=0}^{+\infty}\delta[n-k]$$

accumulated sum of impulses till position n

NOTE: it is also possible to write δ[n]=u[n]-u[n-1] exponential sequences

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

(A and/or α may be complex)

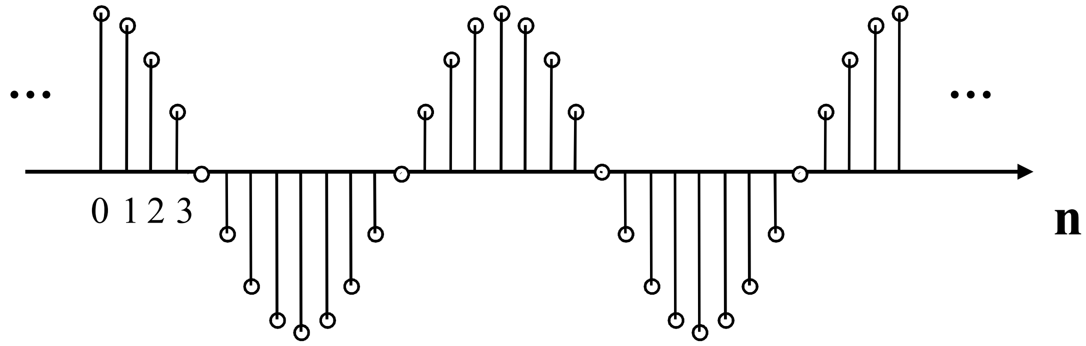


important particular case: A=1 and $\alpha=e^{j\omega}$ (unit complex exponential)

Characterization and representation of discrete signals

<u>sinusoidal sequences</u>

$$x[n] = A\cos(n\omega_0 + \phi)$$



ω_0 - frequency of the sinusoidal sequence [radians]

ϕ - phase of the sinusoidal sequence [radians]

NOTE: since $A.\cos[n(\omega_0+k2\pi)+\phi] = A.\cos[n\omega_0+\phi]$ with k integer, frequency ω_0 is only defined, for example, in the range $]-\pi, +\pi]$ or $[0, 2\pi[$

<u>combination of sequences</u>

It is common to combine basic discrete-time sequences to express a large variety of signals, e.g. :

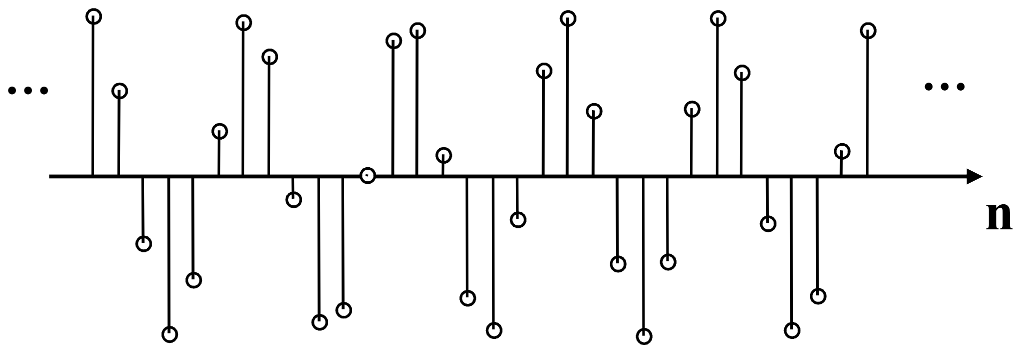
$$x[n] = A\alpha^n u[n]$$

$$x[n] = A(u[n] - u[n-n_0])\alpha^n$$

periodic sequences

those verifying $x[n]=x[n+N]$, for a given integer N and for any value of n: $-\infty < n < +\infty$

NOTE: since the n index may only take integer values, the counter-intuitive case may occur of a function, such as $\cos(n\omega_0+\phi)$, not showing periodicity in n, for a given ω_0, as for example cos(n) :



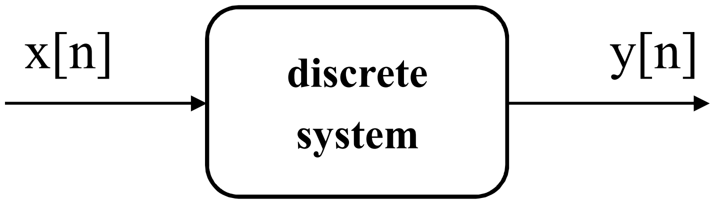
QUESTION: Under what condition is that ω_0 insures periodicity in n ?

A : $\cos(n\omega_0+\phi) = \cos(n\omega_0+N\omega_0+\phi)$, $\Rightarrow N\omega_0 = k2\pi \Leftrightarrow \omega_0=2\pi k/N$

Characterization and representation of discrete systems

- Discrete-time systems

  – they transform an input discrete-time sequence into an output discrete-time sequence



  – Example 1 (delay) : $y[n]=x[n-n_d]$, n_d positive integer = system delay

  – Example 2 (moving average) : $$y[n]=\frac{1}{N_1+N_2+1}\sum_{k=-N_1}^{N_2}x[n-k]$$

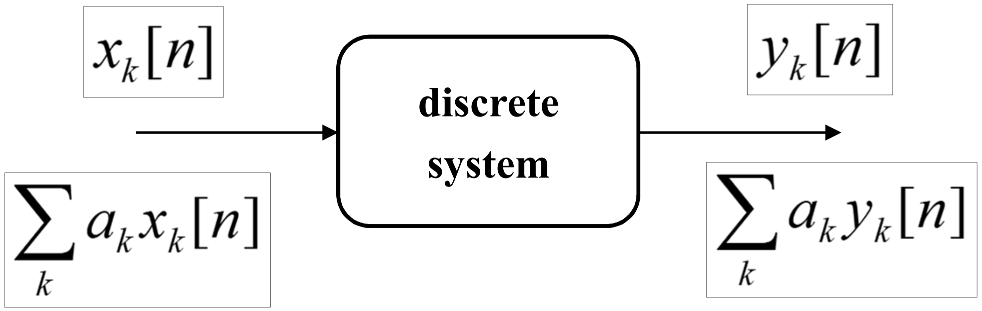
  the output at position n is the average of the (N_1+N_2+1) input samples between position $n-N_2$ and position $n+N_1$

- Properties of discrete-time systems

memory **-** a memoryless system depends only on the input sample at position n to generate an output sample at the same position

- example: $y[n]=\left(x[n]\right)^2$

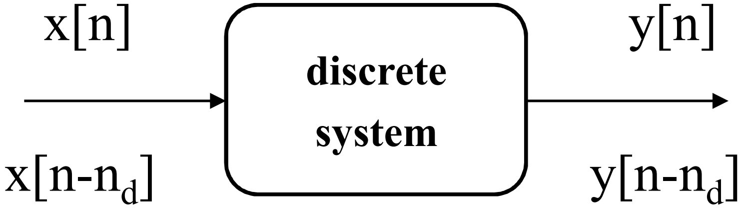
linearity **–** a linear system complies with the superposition principle



- example: the “accumulator” function $y[n]=\sum_{k=-\infty}^{n} x[k]$ is linear
- example: function $y[n]=\log_{10}|x[n]|$ is non-linear

Characterization and representation of discrete systems

<u>time invariance</u> – a system is time-invariant if a delay of the input sequence gives rise to the same delay of the output sequence



- example: the "accumulator" $y[n] = \sum_{k=-\infty}^{n} x[k]$ is time-invariant because if the input samples are delayed by n_0, so are the output samples: $y[n-n_0] = \sum_{k=-\infty}^{n-n_0} x[k]$

- example: the decimator system *y[n]=x[nM]* is not time-invariant

<u>causality</u> – a system is causal (*i.e., it is non-anticipative*) if for any n_0, the output of the system at position $n=n_0$ depends uniquely on the input samples for $n \leq n_0$

- example: *y[n]=x[n]-x[n-1]* is causal
- example: *y[n]=x[n+1]-x[n]* is not causal

Characterization and representation of discrete systems

<u>stability</u> – a system is stable if any bounded input sequence gives rise to an output sequence that is also bounded

$$|x[n]| \le B_x < \infty, \ \forall n \quad \longrightarrow \quad \textbf{discrete system} \quad \longrightarrow \quad |y[n]| \le B_y < \infty, \ \forall n$$

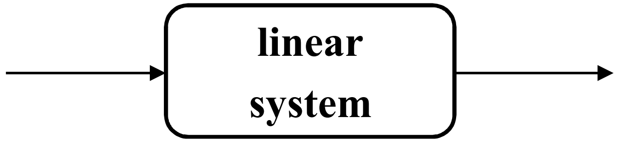
- example: the system $y[n] = \left(x[n]\right)^2$ is stable
- example: the system $y[n] = \log_{10}|x[n]|$ is not stable (*e.g.,* for x[n]=0)
- example: the system $y[n] = \sum_{k=-\infty}^{n} x[k]$ is not stable (*e.g.,* for x[n]=u[n])

NOTE: a sufficient proof of instability is to find/show a case not complying with the stability condition

LTI systems: response to a discrete input

- response to a discrete-time input
  - if the impulse response is not time-invariant, we have:

$$\left.\begin{array}{l}\delta[n-k] \\ x[n]=\sum_{k=-\infty}^{+\infty} x[k]\delta[n-k]\end{array}\right\}$$

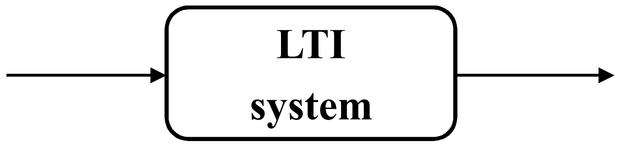


$$\left\{\begin{array}{l}h_k[n] \\ y[n]=\sum_{k=-\infty}^{+\infty} x[k]h_k[n]\end{array}\right.$$

this result is of limited usefulness since the response of the system to a linear combination of input impulses is the same combination of the individual responses to the input impulses which may depend on the position of the impulses (time variance)

  - now, if we consider time-invariance:

$$\left.\begin{array}{l}\delta[n-k] \\ x[n]=\sum_{k=-\infty}^{+\infty} x[k]\delta[n-k]\end{array}\right\}$$



$$\left\{\begin{array}{l}h[n-k] \\ y[n]=\sum_{k=-\infty}^{+\infty} x[k]h[n-k]\end{array}\right.$$

the output of an LTI system is expressed as a function of a single impulse response !

Linear time-invariant systems (LTI)

– we may thus say that an LTI system is completely characterized by its impulse response ⇔ given *h[n]*, it is possible to know the response of the LTI system to *any* input:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k] = \sum_{k=-\infty}^{+\infty} h[k]x[n-k]$$

this equation consists in the <u>discrete-time convolution</u> or, in other words, the convolution sum which is reminiscent of the familiar convolution integral for continuous-time signals:

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau$$

- discrete-time convolution

independent variable

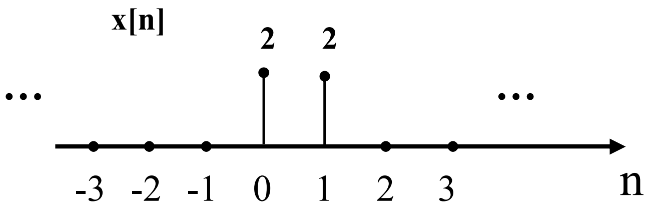
$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

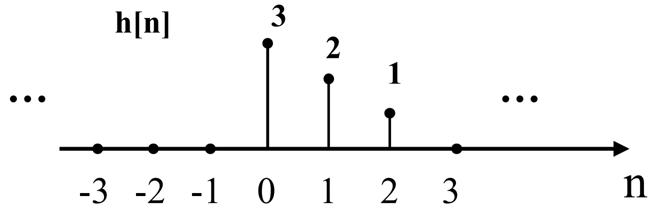
independent variable

parameter (shift in k) !

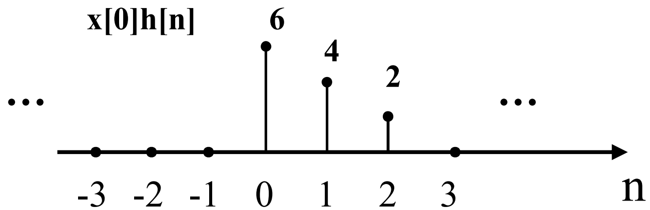
LTI systems: the discrete convolution

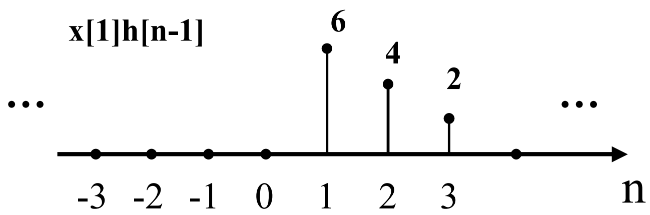
example of the discrete convolution between two sequences:

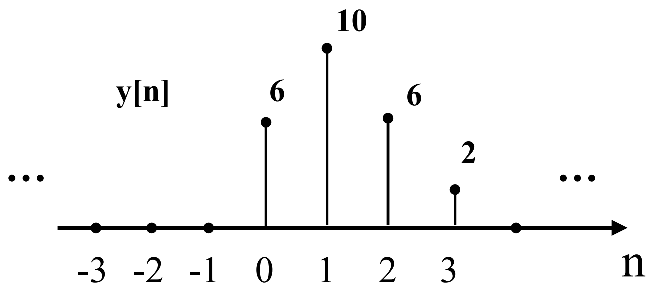




– Method 1: solving by realizing k, we obtain a weighted sum of impulse responses: *y[n]=x[0]h[n]+x[1]h[n-1]*



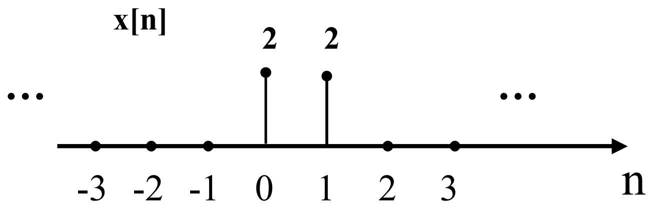


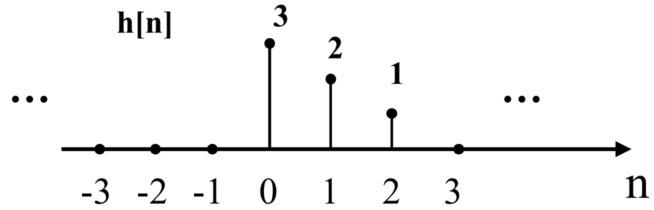


$y[n]=6\delta[n]+10\delta[n-1]+6\delta[n-2]+ 2\delta[n-3]$

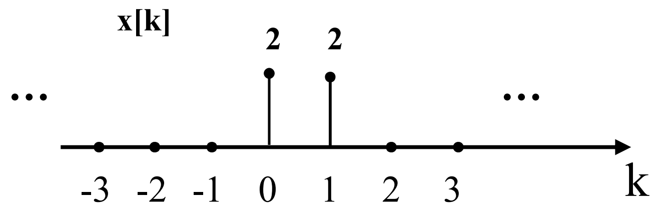
LTI systems: the discrete-time convolution

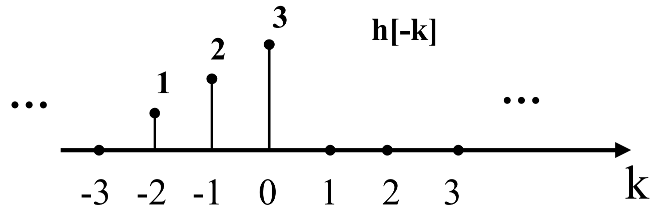
– Method 2: solving by realizing *n*, we follow a computational procedure similar to the convolution between two continuous-time signals (one of discrete sequences is time-reversed and is shifted from -∞ till +∞, and for each value of the shift, the accumulation of the sample-to-sample product between this sequence and the other –frozen- signal is computed). Using our example:



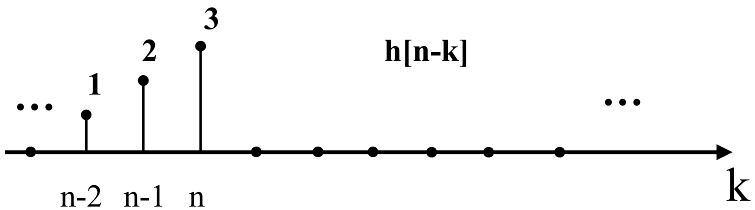


we have:

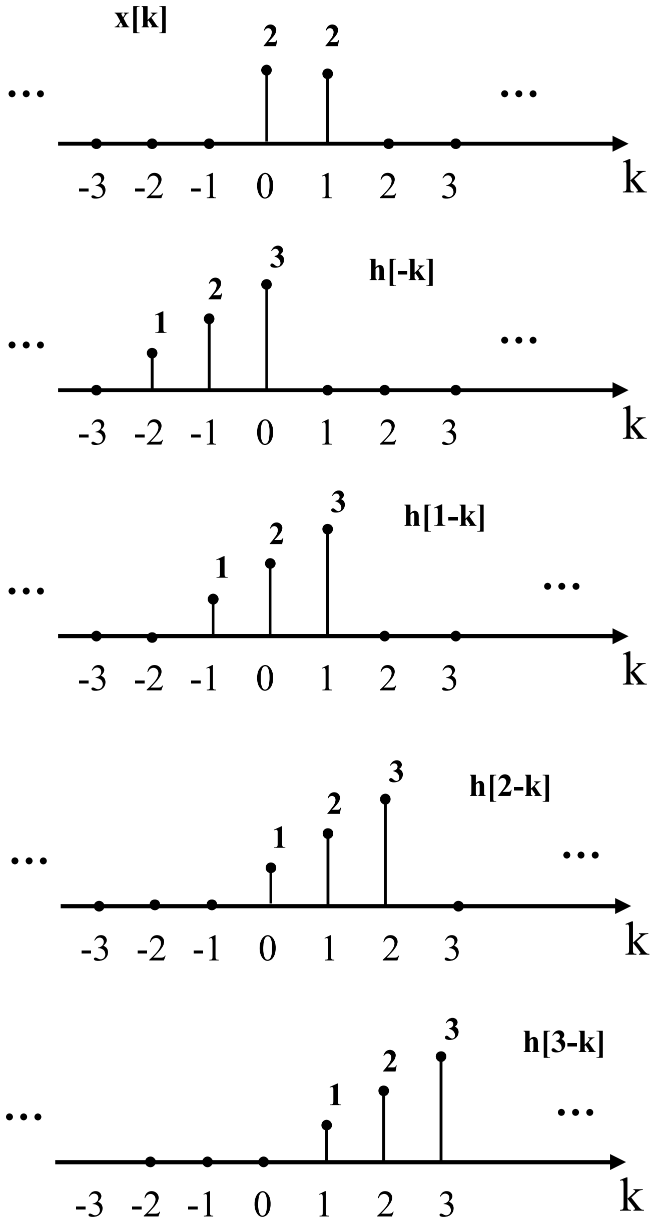




it is clear that
n<0, y[n]=0. For
example, for n=-3 we have:
y[-3]=x[0]h[-3]+x[1]h[-2]=0



LTI systems: the discrete convolution



y[0]=Σ x[k]h[-k]=x[0]h[0]=6

y[1]= Σ x[k]h[1-k]= x[0]h[1]+x[1]h[0]=10

y[2]= Σ x[k]h[2-k]= x[0]h[2]+x[1]h[1]=6

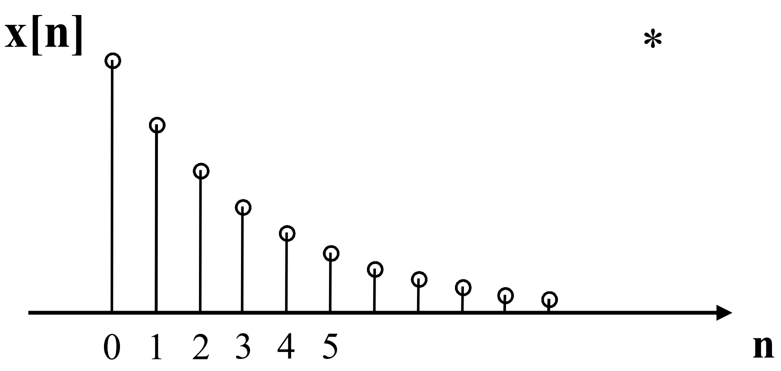
y[3]=Σ x[k]h[3-k]=x[1]h[2]=2

LTI systems: the discrete convolution

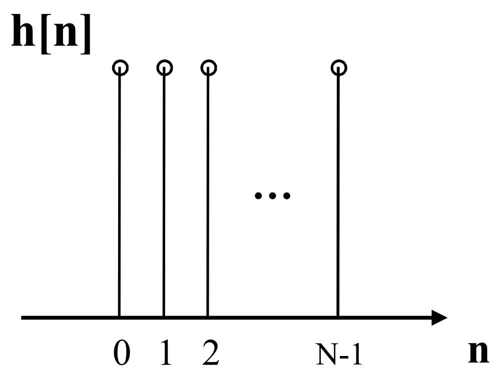
it should be noted that $y[n]=0$ for $n>3$ since $x[k]$ and $h[n-k]$ are not simultaneously different from zero for any value of k.

In summary: $y[n]=6\delta[n]+10\delta[n-1]+6\delta[n-2]+2\delta[n-3]$

Another example: $h[n]=u[n]-u[n-N]$ and $x[n]=\alpha^n u[n]$, with $|\alpha|<1$



$*$

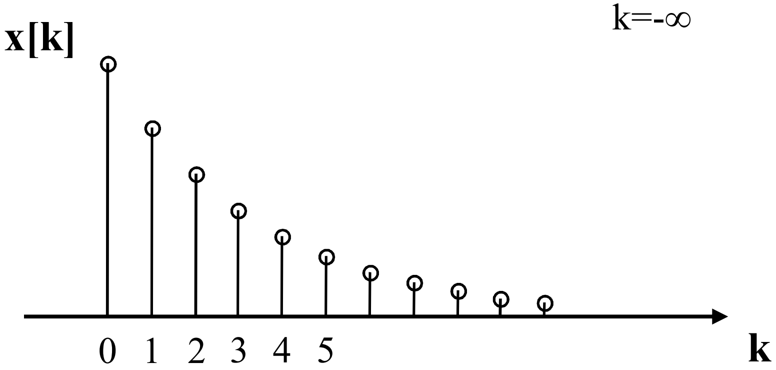


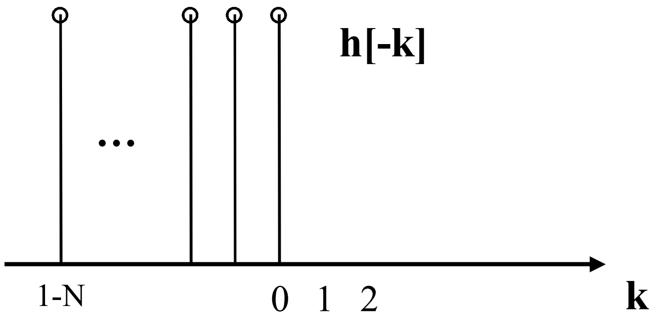
$= ?$

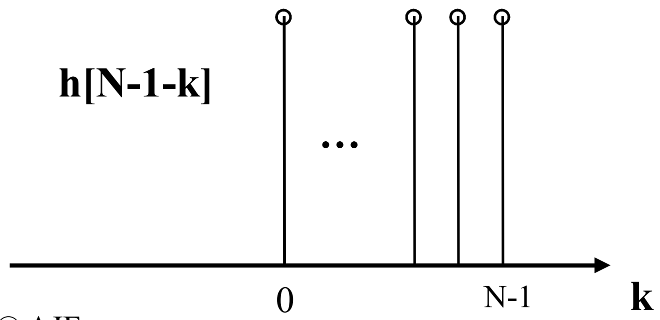
QUESTION: which of the previous two methods is more convenient ?

LTI systems: the discrete convolution

$$y[n]=\sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$







Graphically, it is apparent that we have three intervals for which the y[n] result may be given by a single expression that is valid for all the values of n inside each interval:

Interval 1: $n<0 \;\rightarrow\; y[n]=0$

Interval 2: $0\leq n \leq N-1$

Interval 3: $N-1\leq n$

LTI systems: the discrete convolution

– Interval 1: n < 0, y[n]=0

– Interval 2: 0≤ n ≤ N-1

u[n-k] ≡ 1 for n- k ≥ 0 ⇔ k ≤ n

u[n-k-N] ≡ 1 for n- k-N ≥ 0 ⇔ k ≤ n-N

$$y[n]=\sum_{k=-\infty}^{+\infty}x[k]h[n-k]=\sum_{k=0}^{n}\alpha^k u[k]\big(u[n-k]-u[n-k-N]\big)=\sum_{k=0}^{n}\alpha^k$$

final: k ≥ 0 && n-N+1 ≤ k ≤ n ≡ 0 ≤ k ≤ n
(it is easier to conclude graphically)

– Interval 3: n ≥ N-1

u[k] ≡ 1 for k ≥ 0

≡ 1 for n-N+1 ≤ k ≤ n

$$y[n]=\sum_{k=-\infty}^{+\infty}x[k]h[n-k]=\sum_{k=n-N+1}^{n}\alpha^k u[k]\big(u[n-k]-u[n-k-N]\big)=\sum_{k=n-N+1}^{n}\alpha^k$$

final: k ≥ 0 && n-N+1 ≤ k ≤ n ≡ n-N-1 ≤ k ≤ n
(it is easier to conclude graphically)

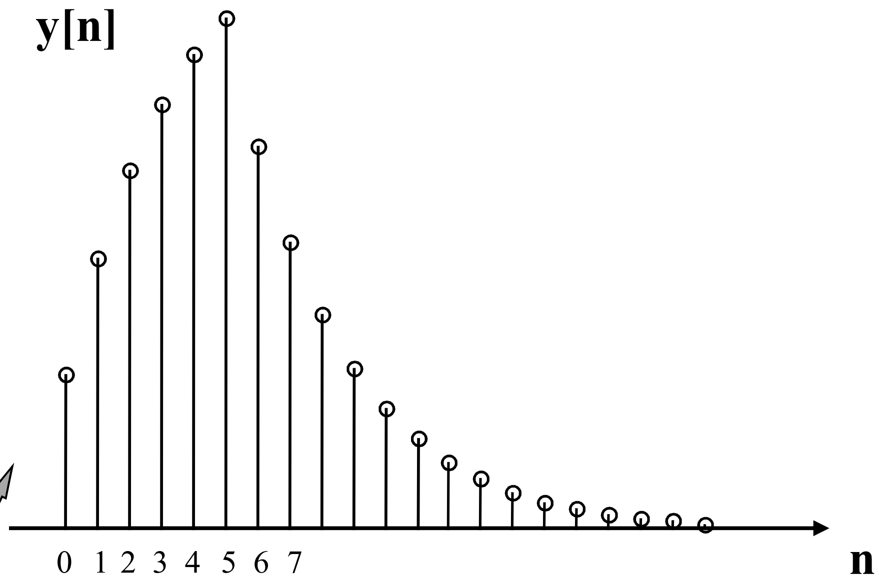
LTI systems: the discrete convolution

– Since the sum of M terms of a geometric series is given by the expression: $G_M = \sum_{k=0}^{M} \alpha^k = \dfrac{1-\alpha^{M+1}}{1-\alpha}$ (for an arithmetic series it would be:)

$$A_M = \sum_{k=1}^{M} k = \frac{M(M+1)}{2}$$

the final result is:

$$y[n] = \begin{cases} 0, & n < 0 \\ \dfrac{1-\alpha^{n+1}}{1-\alpha}, & 0 \le n \le N-1 \\ \dfrac{\alpha^{n-N+1} - \alpha^{n+1}}{1-\alpha}, & n \ge N-1 \end{cases}$$



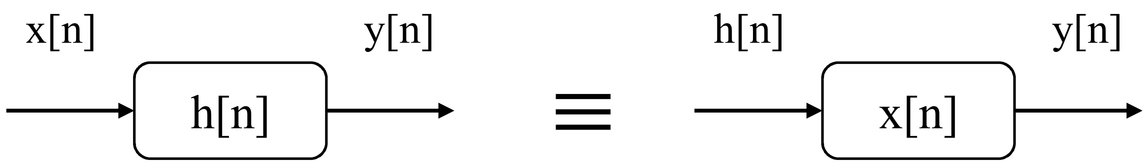
In this illustrative case, what is the value of N ?

Properties of LTI systems

- Since an LTI system is completely characterized by its impulse response, its properties follow those of the discrete-time convolution

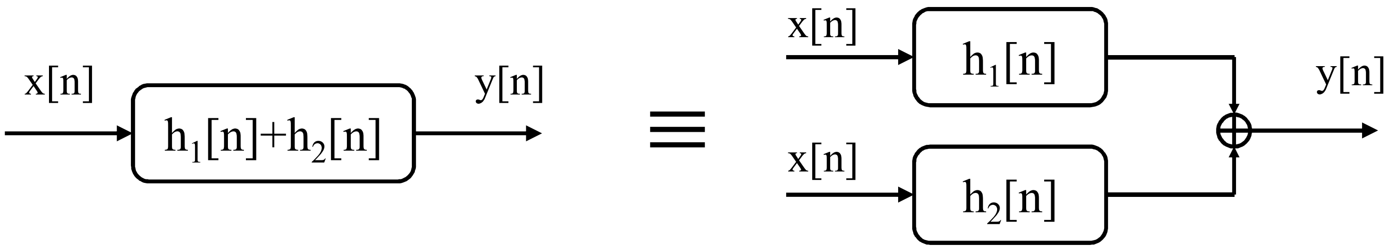
  – <u>commutative property:</u>

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k] = \sum_{k=-\infty}^{+\infty} h[k]x[n-k] = h[n] * x[n]$$



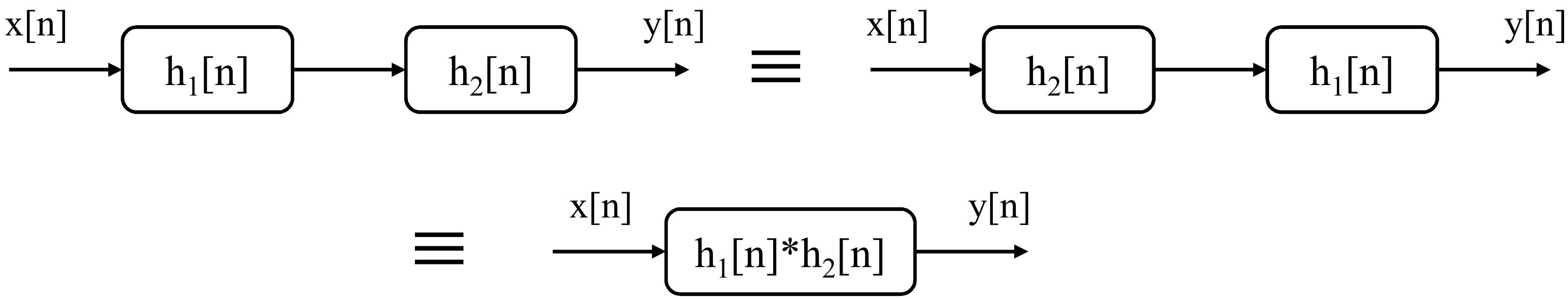
  – <u>distributive property</u> (of the convolution relative to the sum):

$$y[n] = x[n] * \left(h_1[n] + h_2[n]\right) = x[n] * h_1[n] + x[n] * h_2[n]$$

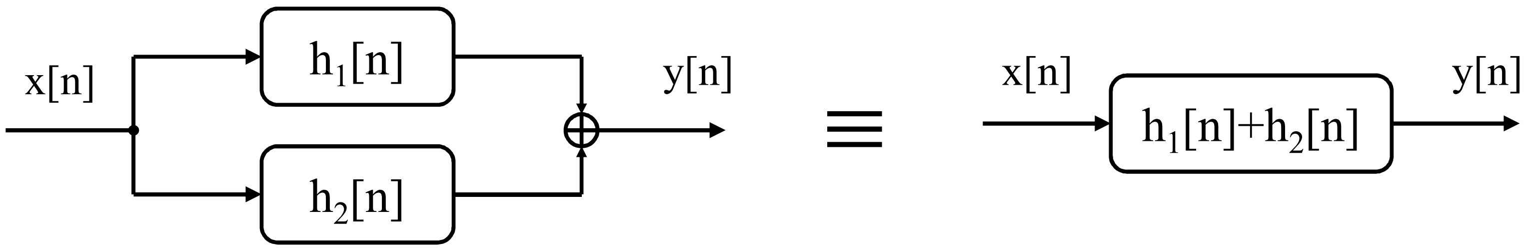


Properties of LTI systems

– series of systems:



– parallel of systems:



Properties of LTI systems

– <u>condition for the stability of an LTI system</u>: if and only if its impulse response is absolutely summable:

$$S = \sum_{k=-\infty}^{+\infty} |h[k]| < \infty$$ (necessary and sufficient condition)

– <u>condition for the causality of an LTI system:</u>

$$h[n] = 0, \quad n < 0$$

example: what is the impulse response of each one of the following LTI systems ?

$$y_1[n] = x[n-n_d]$$

$$y_2[n] = \frac{1}{N_1+N_2+1}\sum_{k=-N_1}^{N_2} x[n-k]$$

$$y_3[n] = \sum_{k=-\infty}^{n} x[k]$$

since h[n] = y[n] $\big|_{x[n]=\delta[n]}$ we have:

$$h_1[n] = \delta[n-n_d]$$

$$h_2[n] = \frac{1}{N_1+N_2+1}\sum_{k=-N_1}^{N_2} \delta[n-k] = \begin{cases} \frac{1}{N_1+N_2+1}, & -N_1 \le n \le N_2 \\ 0, & other \end{cases}$$

$$h_3[n] = \sum_{k=-\infty}^{n} \delta[k] = u[n] = \begin{cases} 1, & n \ge 0 \\ 0 & n < 0 \end{cases}$$

LTI systems: FIR and IIR systems

- According to the number of non-zero samples of its impulse response, an LTI system may be classified as:
  - FIR (finite-duration impulse response): if h[n] has a finite number of non-zero samples
    - NOTE 1: FIR systems are always stable
    - NOTE 2: a non-causal FIR system may be converted into a causal FIR system by adding a suitable delay
  - IIR (infinite-duration impulse response): if h[n] has an infinite number of non-zero samples
    - NOTE: IIR systems may be stable, for example:

$$h[n] = \alpha^n u[n], \quad |\alpha| < 1 \quad \rightarrow \quad S = \sum_{n=0}^{\infty} |\alpha|^n = \frac{1}{1-\alpha}$$

    - NOTE: IIR systems may also be unstable, for example:

$$h[n] = u[n] \quad \rightarrow \quad S = \sum_{n=0}^{\infty} 1 = \infty$$

Constant-coefficient difference equations

- Consists in an alternative way (relative to the impulse response) to characterize (although not completely) a sub-class of LTI systems (the characterization is only complete if it is added, for example, that the system is causal and starts from rest) by relating a combination of delayed inputs with a combination of delayed outputs, which describes (a realization of) the system:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m] \quad \Leftrightarrow \quad y[n] = -\sum_{k=1}^{N} \frac{a_k}{a_0} y[n-k] + \sum_{m=0}^{M} \frac{b_m}{a_0} x[n-m]$$

  – NOTE 1: this form emphasizes the recursive nature of the relation: the output is obtained after the input sequence is known and after the previous values of the output sequence are known.

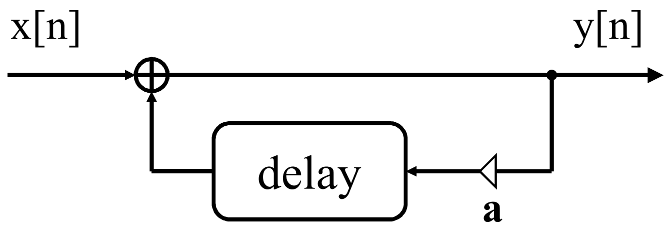
  – NOTE 2: if N=0, we have:

$$y[n] = \sum_{m=0}^{M} \frac{b_m}{a_0} x[n-m] \quad \Leftrightarrow \quad h[n] = \sum_{m=0}^{M} \frac{b_m}{a_0} \delta[n-m]$$

  which reveals we are dealing with an FIR system, while the more general equation describes an IIR system.

Constant-coefficient difference equations

Example: what is the impulse response of the causal system described by the difference equation: **y[n]=ay[n-1]+x[n]**



A: considering x[n]=kδ[n] e admitting that the system starts from rest:

n	x[n]	y[n-1]	y[n]
-1	0	0	0
0	k	0	k
1	0	k	ak
2	0	ak	a^2k
3	0	a^2k	a^3k
4	0	a^3k	a^4k
:	:	:	:
n	0	$a^{n-1}k$	a^nk

this means: $h[n]=a^nu[n]$

NOTE: the same system may be described by different difference equations; a specific difference equation is indicative of a specific realization of a discrete system among several possible alternatives (topic to be detailed later on).

Basic signal properties

- Continuous *versus* discrete
  - a continuous-time signal is a real or complex function of one or more independent variables that, most often, are real-valued, e.g. $x_c(t)$
    - the round brackets reinforce that the independent variable is continuous-time
  - a discrete-time signal is a real or complex function of one or more independent variables that can take on integer values only, e.g. $x[n]$
    - the square brackets denote that the enclosed variable is discrete-time

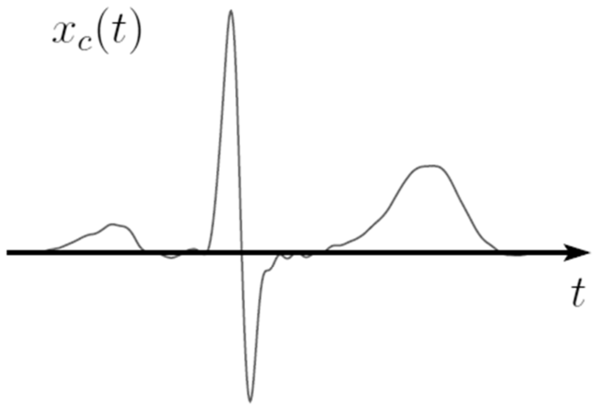


Fig. 1.16 A continuous-time signal representing the bioelectric voltage due to one cycle of the heart beat.

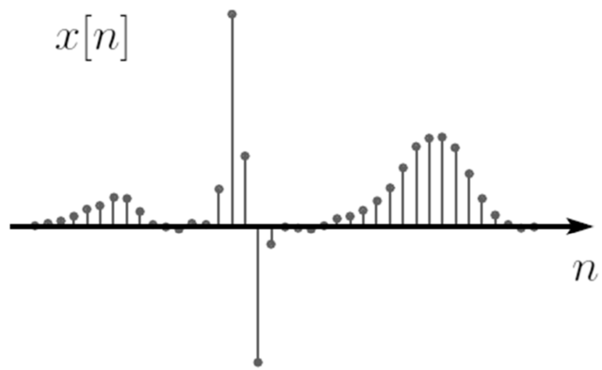


Fig. 1.17 A discrete-time version of the continuous-time signal in Fig. 1.16

Basic signal properties

- Periodic *versus* aperiodic

  – a periodic discrete-time signal is one whose structure or pattern repeats in n for some finite period N, i.e.

$$x[n] = x[n+N], \qquad N \in \mathbb{Z}\backslash\{0\}, \qquad \forall\, n \in \mathbb{Z}$$

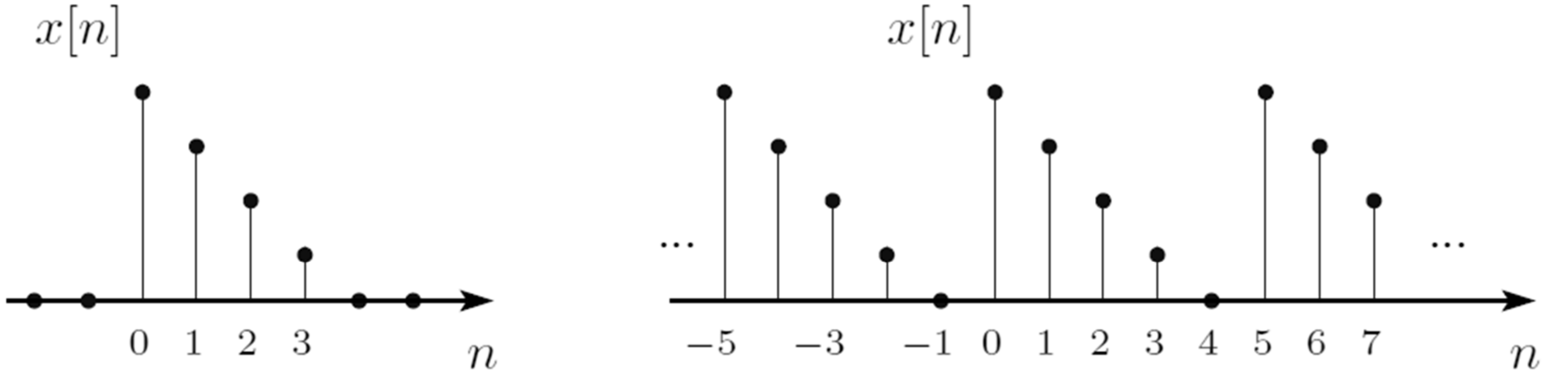


Fig. 1.20 Illustration of an aperiodic discrete sequence (figure on the left) and a periodic discrete sequence whose period is $N = 5$ (figure on the right).

Question: if a periodic signal is obtained by repeating periodically an aperiodic one, what condition makes that the aperiodic signal is recognizable in the periodic signal ?

- Deterministic *versus* random
  - a deterministic signal is specified by a mathematical function, an algorithmic procedure, or computational method, which completely determines the signal value at any point in the discrete-time domain n
  - quite often, these deterministic signal representation alternatives are replaced by a lookup table (LUT) when trading computation for memory is desired
  - a notable feature of a deterministic signal is that it is predictable
  - if a signal is deterministic, so are its statistical properties, for example, the probability density function (PDF) of the signal amplitudes

  - a random signal is characterized by unpredictability and uncertainty concerning the value of each sample in a discrete-time sequence
  - instead of being governed by a deterministic rule, the realization of each sample in a random sequence is governed by a probabilistic model underlying the stochastic process that generates that sequence
  - although a random signal is unpredictable at sample level, certain practical assumptions, such as stationarity, make that 'latent' probabilistic attributes underlying a random signal, are predictable

- Deterministic *versus* random
  - a stationary random signal may exhibit a specific PDF, for example, uniform, of Gaussian
  - these possibilities are illustrated next and have been created using the Matlab functions `rand()` and `randn()`

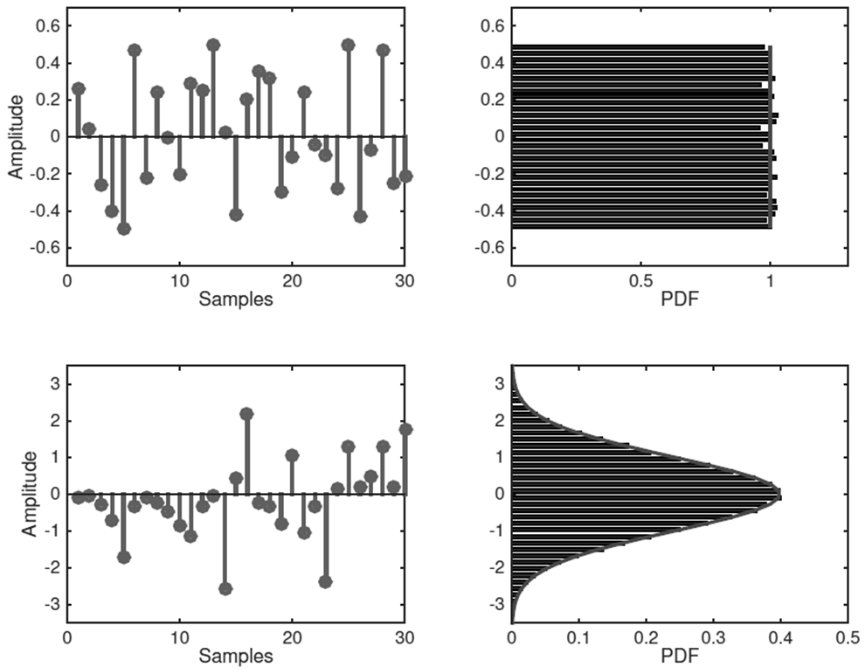


Fig. 1.23 Representation of the first 30 samples of two random sequences, one having a uniform PDF between -0.5 and 0.5 (top left) and another having a Gaussian PDF with unit variance (bottom left). The corresponding PDF are inferred on the right-hand side by means of histograms. The exact PDF models are also represented as a solid (magenta) lines.

- Energy *versus* power
  - a discrete-time signal $x[n]$ is classified as an energy signal if its energy (E) is finite, i.e. if

$$E = \sum_{n=-\infty}^{+\infty} |x[n]|^2 < \infty$$

  - any finite-length sequence is always classified as an energy signal as long as its samples have a finite magnitude
  - a discrete-time signal $x[n]$ is classified as a power signal if its energy is infinite but a finite result is obtained when the energy of an arbitrary large number of samples is divided by the number of samples

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{+N} |x[n]|^2 < \infty$$

  - in the case of an *N*-periodic signal

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2 < \infty$$

- Sinusoidal sequences with a prescribed SNR
  - frequently, discrete-time test signals need to be generated that have a specific level of noise contamination; most often, these test signals consist of a single complex exponential, or a single sinusoidal sequence, and the noise consists of complex or real-valued Gaussian noise
  - the severity of the noise contamination is objectively determined by the ratio between the average power of the signal, which we represent as P_S, and the average power of the noise, which we represent as P_N
  - that ratio is typically evaluated on a logarithmic scale in tenths of a unit called Bel, which is usually abbreviated to deciBel, or dB; thus, this scale expresses a proportion which is called the Signal-to-Noise Ratio (SNR) and is defined as

$$SNR = 10 \log_{10} \frac{P_S}{P_N}$$

- Sinusoidal sequences with a prescribed SNR
  - the following example illustrates the case of a real sinusoid that is contaminated by Gaussian noise according to a prescribed SNR; the result is confirmed numerically in Matlab

Example 1.12. Create in Matlab a real-valued sinusoidal sequence having frequency $\omega_0 = 0.0123$ rad. and magnitude $A = 1$. Create a noise vector such that when it is added to the sinusoidal sequence the resulting SNR is 30 dB. Validate numerically the Matlab code using 10^4 samples.

Examples 1.8 and 1.9 have shown that the average power of a real-valued sinusoid (or co-sinusoid) having magnitude A is given by $A^2/2$. Example 1.11 has also shown that the average power of (zero-mean) Gaussian noise is given by σ^2. Generating a noise vector complying with a desired SNR just involves using the `randn(·)` Matlab function with the appropriate σ parameter. Thus, we find first σ based on (1.55):

$$\sigma = \frac{A}{10^{SNR/20}\sqrt{2}} \ .$$

After both discrete sequences are created, a numerical validation is easily achieved by estimating the corresponding average power according to Eq. (1.16.4) or Eq. (1.52). A possible Matlab implementation is listed next.

```
N=1E4; n=[0:N-1].';
omega=0.0123; A=1; signal=A*sin(omega*n);
SNR=30; sigma=A/(sqrt(2)*10^(SNR/20));
noise=sigma*randn(N,1);
Ps=mean((abs(signal)).^2);
Pn=mean((abs(noise)).^2);
10*log10(Ps/Pn)

ans = 30.06
```

The auto-correlation and the cross-correlation

- concept and meaning
  - the auto-correlation and the cross-correlation are two important discrete-time signal processing functions that consist in specific forms of the discrete-time convolution
  - they are designed to evaluate the similarity between two discrete-time signals, or waveforms
  - if the two waveforms are based on the same discrete-time signal, then the function is called auto-correlation whereas if the two waveforms are based on different discrete-time signals, then the function is called cross-correlation

- definition of the auto-correlation
  - the auto-correlation ($r_x[\ell]$) assesses the similarity between between a finite-energy reference signal, $x[k]$, and the conjugate of its shifted version $x^*[k-\ell]$, where ℓ represents a discrete-time shift, also commonly referred to as lag
  - it is obtained as the discrete-time convolution between $x[\ell]$ and the conjugate of its time-reversed version, $x^*[-\ell]$:

$$r_x[\ell] \triangleq x[\ell] * x^*[-\ell] = \sum_{k=-\infty}^{+\infty} x[k]x^*[-(\ell-k)] = \sum_{k=-\infty}^{+\infty} x[k]x^*[k-\ell]$$

  - The auto-correlation may be regarded as a self-similarity measure for a given discrete-time shift
  - if the self-similarity is strong, then the auto-correlation exhibits a high absolute value; conversely, if the self-similarity is weak, then the auto-correlation exhibits a small absolute value, nearing zero in the case of strong dissimilarity

- definition of the cross-correlation
  - the cross-correlation ($r_{xy}[\ell]$) assesses the similarity between between a finite-energy reference signal, $x[k]$, and $y^*[k-\ell]$ that represents the conjugate of another discrete-time signal, y$[k]$, that is also affected by a lag ℓ
  - it is obtained as the discrete convolution between $x[\ell]$ and $y^*[-\ell]$:

$$r_{xy}[\ell] \triangleq x[\ell] * y^*[-\ell] = \sum_{k=-\infty}^{+\infty} x[k]y^*[-(\ell-k)] = \sum_{k=-\infty}^{+\infty} x[k]y^*[k-\ell]$$

  - if the similarity between the two waveforms is strong for a given lag, then the cross-correlation function exhibits a high absolute value for that lag, however, if the similarity is weak, then the cross-correlation exhibits a small absolute value; when it nears zero, one may say the signals are approximately uncorrelated

The auto-correlation and the cross-correlation

- auto-correlation and cross-correlation examples
  - a special waveform $x[n]$ is designed such that its auto-correlation consists of a single impulse, i.e. $r_x[\ell] = \delta[\ell]$
  - a second waveform y$[n]$ is generated that consists of a noisy version of $x[n+3]$, i.e. $y[n] = x[n+3] + v[n]$, where $v[n]$ is a random sequence that is not correlated to $x[n]$
  - both $r_x[\ell]$ and $r_{xy}[\ell]$ are represented in the following figure

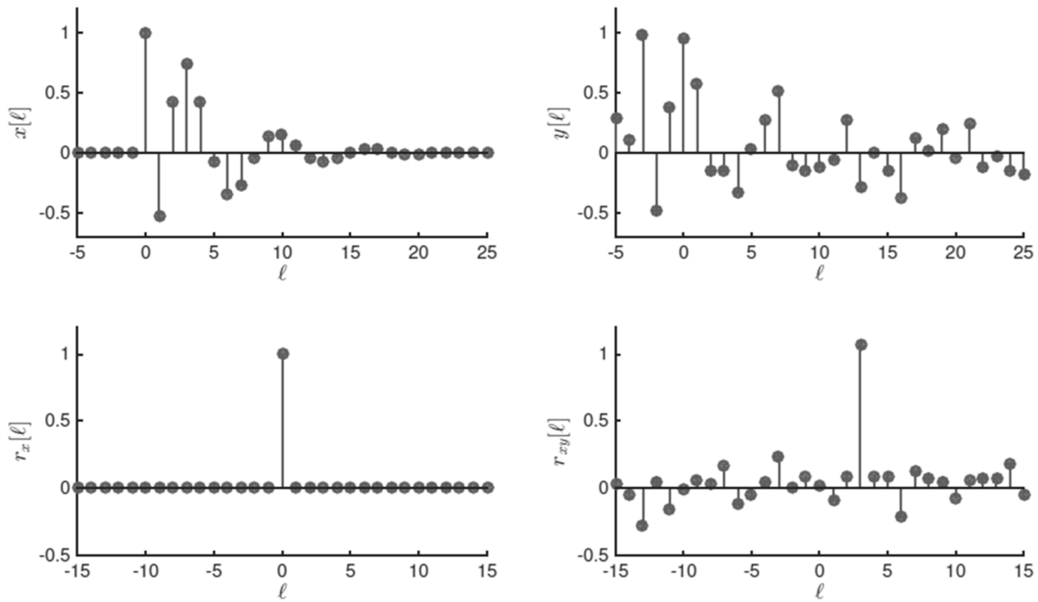


Fig. 1.76 Example of a waveform $x[\ell]$ (top left figure), its auto-correlation function $r_x[\ell]$ (bottom left figure), a waveform $y[\ell]$ consisting of noisy version of $x[\ell+3]$ (top right figure), and the cross-correlation function $r_{xy}[\ell]$ (bottom right figure).

The auto-correlation and the cross-correlation

- auto-correlation and cross-correlation properties
  - in the case of the auto-correlation, it can be easily shown that if the lag is zero, then we obtain the energy of the signal

$$r_x[0] = \sum_{k=-\infty}^{+\infty} |x[k]|^2 = E_x$$

  - the auto-correlation is conjugate-symmetric, i.e. $r_x[\ell] = r_x^*[-\ell]$
  - the cross-correlation verifies $r_{xy}[\ell] = r_{yx}^*[-\ell]$
    - which does ***not*** mean conjugate-symmetry
  - it can also be shown that $|r_{xy}[\ell]| \leq \sqrt{\mathsf{E}_x \mathsf{E}_y} = \sqrt{r_x[0] r_y[0]}$
  - and, as a particular case, the auto-correlation is upper bounded by the signal energy:

$$|r_x[\ell]| \leq r_x[0] = E_x$$

- auto-correlation and cross-correlation properties
  - the results in the previous slide can be used to normalize both auto-correlation and cross-correlation functions, which leads to $\rho_x[\ell]$ and $\rho_{xy}[\ell]$:

$$\rho_x[\ell] = \frac{r_x[\ell]}{r_x[0]} \, , \qquad -1 < \rho_x[\ell] \leq 1, \ \forall \ \ell \in \mathbb{Z}$$

$$\rho_{xy}[\ell] = \frac{r_{xy}[\ell]}{\sqrt{r_x[0] r_y[0]}} \, , \qquad -1 \leq \rho_{xy}[\ell] \leq 1, \ \forall \ \ell \in \mathbb{Z}$$

Question: the previous results have been developed for energy signals, how should they be adapted to power signals ?