PROCESSAMENTO DE SINAL E IMAGEM EM FÍSICA MÉDICA

2019/2020 – 2º Semestre (F4012)

Discrete-Time Systems

Classification of systems
Impulse and Step Responses
Time-Domain Characterization of LTI DT Systems
Convolution Sum
Stability and Causality of LTI Systems
LTI systems with linear constant coefficient difference equation
Finite (FIR) and Infinite (IIR) Impulse Response Systems
Recursive LTI Discrete-Time Systems
Correlation of Signals

Discrete-Time Systems

There are various types of classifications that can be used for discrete-time systems, such as:

- Linear System (Sistema linear)
- Time-Invariant System (Invariante no tempo)
- Causal System (Causal)
- Stable System (Estável)
- Passive and Lossless System (Passivo / Sem perdas)

Linear Discrete-Time Systems

DEFINITION

Given a system which has $y_1[n]$ as the output for the input $x_1[n]$, and $y_2[n]$ as the output for the input $x_2[n]$, then

for an input $x[n] = \alpha\, x_1[n] + \beta\, x_2[n]$

the output is $y[n] = \alpha\, y_1[n] + \beta\, y_2[n]$

This property must hold for any arbitrary constants α and β, and for all possible inputs $x_1[n]$ and $x_2[n]$.

Some examples in the following slides.

Linear Discrete-Time Systems

Ex.1 – Accumulator $y_1[n] = \sum_{k=-\infty}^{n} x_1[k]$

$y_2[n] = \sum_{k=-\infty}^{n} x_2[k]$

For an input $x[n] = \alpha\, x_1[n] + \beta\, x_2[n]$

the output is $y[n] = \sum_{k=-\infty}^{n} (\alpha\, x_1[k] + \beta\, x_2[k])$

$= \alpha \sum_{k=-\infty}^{n} x_1[k] + \beta \sum_{k=-\infty}^{n} x_2[k]$

$= \alpha\, y_1[n] + \beta\, y_2[n]$

Hence, the system is **linear**.

Linear Discrete-Time Systems

Ex.2 – Causal accumulator $y_1[n] = y_1[-1] + \sum_{k=0}^{n} x_1[k]$

$y_2[n] = y_2[-1] + \sum_{k=0}^{n} x_2[k]$

The output y[n] for an input $x[n] = \alpha\, x_1[n] + \beta\, x_2[n]$ is given by

$$y[n] = y[-1] + \sum_{k=-\infty}^{n} (\alpha\, x_1[k] + \beta\, x_2[k])$$

As $\alpha y_1[n]+\beta y_2[n] = \alpha\,(y_1[-1] + \sum_{k=-\infty}^{n} x_1[k]) + \beta\,(y_2[-1] + \sum_{k=-\infty}^{n} x_2[k])$

$= \alpha y_1[-1] + \beta y_2[-1] + \alpha \sum_{k=-\infty}^{n} x_1[k] + \beta \sum_{k=-\infty}^{n} x_2[k])$

Thus $y[n] = \alpha y_1[n]+\beta y_2[n]$ if $y[-1] = \alpha y_1[-1]+\beta y_2[-1]$ for all initial conditions $y_1[-1]$, $y_2[-1]$, and all values of α and β.

This condition cannot be satisfied unless the acumulator is initially at rest, that is $y_1[-1]=y_2[-1]=0$.

For non-zero initial conditions, the system is **non-linear**.

Non-Linear Discrete-Time Systems

Median Filter

The median filter is a non-linear discrete-time system.

Consider the length 3 median filter and the following inputs:

$x_1[n] = \{ 3, 4, 5 \}$ and $x_2[n] = \{ 2, -1, -1 \}$ for $0 \le n \le 2$

The outputs for these inputs are (using 0-padding):

$y_1[n] = \{ 3, 4, 4 \}$ and $y_2[n] = \{ 0, -1, -1 \}$ for $0 \le n \le 2$

However, the output for the input $x[n] = x_1[n] + x_2[n] = \{ 5, 3, 4 \}$ is:

$y[n] = \{ 3, 4, 3\}$, which is different from $y_1[n] + y_2[n] = \{ 3, 3, 3 \}$

Time-Invariant (or shift-invariant) System

DEFINITION

For a time-invariant system, if $y_1[n]$ is the system response (output) for an input $x_1[n]$, then the response to an input

$x[n] = x_1[n-n_0]$ is $y[n] = y_1[n-n_0]$

where n_0 is any positive or negative integer.

This relation must hold for any arbitrary input and corresponding output.

Time-invariance property, in the case of sequences and systems with indices n related to discrete instants of time.

This property ensures that for a specified input, the output is independent of the time the input is being applied.

Time-Invariant System

Consider the up-sampler with an input-output relation given by

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

For an input $x_1[n] = x[n-n_o]$ the output $x_{1,u}[n]$ is given by

$$x_{1,u}[n] = \begin{cases} x_1[n/L], & n = 0, \pm L, \pm 2L, \ldots.. \\ 0, & \text{otherwise} \end{cases}$$
$$= \begin{cases} x[(n-Ln_o)/L], & n = 0, \pm L, \pm 2L, \ldots.. \\ 0, & \text{otherwise} \end{cases}$$

However from the definition of the up-sampler

$$x_u[n-n_o] = \begin{cases} x[(n-n_o)/L], & n = n_o, n_o \pm L, n_o \pm 2L, \ldots.. \\ 0, & \text{otherwise} \end{cases} \neq x_{1,u}[n]$$

- Hence, the up-sampler is a time-varying system

Linear Time-Invariant (LTI) System

DEFINITION

A system is Linear Time-Invariant (LTI) if it satisfies both the linearity and time-invariance properties (Sistema linear e invariante no tempo).

LTI systems are mathematically easy to analyse and characterise, and consequently, easy to design.

Highly useful signal processing algorithms have been developed using this class of systems

Causal System

DEFINITION

In a causal system, the n_0-th output sample $y[n_0]$ depends only on input samples $x[n]$ for $n \leq n_0$, and does not depend on input samples for $n > n_0$.

Let $y_1[n]$ and $y_2[n]$ be the responses of a causal discrete-time system to the inputs $x_1[n]$ and $x_2[n]$, respectively.

Then, $x_1[n] = x_2[n]$ for $n<N$ implies also that $y_1[n] = y_2[n]$ for $n<N$.

For a causal system, changes in output samples do not precede changes in the input samples.

Causal System

Examples of causal systems:

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

$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + a_1 y[n-1] + a_2 y[n-2]$

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

Examples of non-causal systems:

(*) $y[n] = x[n] + 0.5 x[n-1] + 0.5 x[n+1]$

$y[n] = x[n] + 0.3 x[n-2] + 0.7 x[n-1] + 0.7 x[n+1] + 0.3 x[n+2]$

A non-causal system can be implemented as a causal system by delaying the output.

e.g. (*) $y[n] = x[n-1] + 0.5 x[n-2] + 0.5 x[n]$

Stable System

There are various possible definitions of stability. Here the **bounded-input, bounded-output (BIBO)** stability condition is considered.

If y[n] is the response of a system to an input x[n],

and if $|x[n]| \leq B_x$ for all values of n

then $|y[n]| \leq B_y$ for all values of n

<u>Ex. M-point moving average</u> $y[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k]$

For a bounded input $|x[n]| \leq B_x$, the output will be

$|y[n]| = \left|\frac{1}{M}\sum_{k=0}^{M-1} x[n-k]\right| \leq \frac{1}{M}\sum_{k=0}^{M-1} |x[n-k]| \leq \frac{1}{M}(MB_x) \leq B_x$

The system is BIBO stable.

Passive and Lossless Systems

A discrete-time system is defined to be **passive** (passivo) if, for every finite-energy input x[n], the output y[n] has, at most, the same energy as the input.

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

For a **lossless system** (sistema sem perdas), the above inequality is satisfied with na equal sign for every input.

Example – Consider the discrete-time system defined by

$y[n] = \alpha\, x[n-N]$, where N is a positive integer.

The output energy is given by $\sum_{n=-\infty}^{\infty} |y[n]|^2 = |\alpha|^2 \sum_{n=-\infty}^{\infty} |x[n]|^2$

It is a passive system if $|\alpha| \leq 1$ and a lossless system if $|\alpha| = 1$.

Impulse and Step Response

The response of a discrete-time system to a unit sample sequence $\delta[n]$ is called the unit sample response, or simply the **impulse response** (Resposta impulsional), and is denoted by **h[n]**.

The response of a discrete-time system to a unit step sequence $\mu[n]$ is called the unit step response, or simply the **step response**, and is denoted by **s[n]**.

Ex. 1 The impulse response of the system

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

is obtained by setting $x[n]=\delta[n]$, resulting in

$h[n] = \alpha_1\, \delta[n] + \alpha_2\, \delta[n-1] + \alpha_3\, \delta[n-2] + \alpha_4\, \delta[n-3]$

h[n] is thus a 4-length sequence – $h[n] = \{ \underset{\uparrow}{\alpha_1} , \alpha_2 , \alpha_3 , \alpha_4 \}$

Impulse and Step Response

Ex. 2 The impulse response h[n] of the discrete-time accumulator

$y[n] = \sum_{k=-\infty}^{n} x[k]$ is obtained by setting $x[n]=\delta[n]$

The result is $h[n] = \sum_{k=-\infty}^{n} \delta[k] = \mu[n]$

Ex. 3 The impulse response h[n] of the factor-of-2 interpolator

$$y[n] = x[n] + 0.5\, x[n-1] + 0.5\, x[n+1]$$

is obtained by setting $x[n]=\delta[n]$, resulting in

$$h[n] = \delta[n] + 0.5\, \delta[n-1] + 0.5\, \delta[n+1]$$

The impulse response is thus a finite length sequence (length 3)

$$h[n] = \{\, 0.5,\ \underset{\uparrow}{1},\ 0.5 \,\}$$

LTI Discrete-Time System

A consequence of the linearity and time invariance properties, is that a Linear Time-Invariant (LTI) system is completely characterised by its impulse response.

Input-Output Relationship – By knowing the impulse response, the output of a LTI system can be computed for any arbitrary input.

Example: Let h[n] be the impulse response of a LTI discrete-time system, x[n] the input and y[n] the output. Consider the following input:

$$x[n] = 0.5\, \delta[n+2] + 1.5\, \delta[n-1] - \delta[n-2] + 0.75\, \delta[n-5]$$

As the system is linear, we can compute its output for each member of the input separately, and add the individual outputs to determine y[n].

LTI Discrete-Time System

Input: x[n] = 0.5 δ[n+2] + 1.5 δ[n-1] - δ[n-2] + 0.75 δ[n-5]

Since the system is time-invariant,

input	output	input	output
δ[n+2] →	h[n+2]	0.5δ[n+2] →	0.5h[n+2]
δ[n-1] →	h[n-1]	1.5δ[n-1] →	1.5h[n-1]
δ[n-2] →	h[n-2]	-δ[n-2] →	-h[n-2]
δ[n-5] →	h[n-5]	0.75δ[n-5] →	0.75h[n-5]

As the system is linear,

y[n] = 0.5 h[n+2] + 1.5 h[n-1] - h[n-2] + 0.75 h[n-5]

The output is then a linear combination of the impulse response h[n] displaced in time (delayed and advanced versions of h[n]).

LTI Discrete-Time System

Any arbitrary input sequence x[n] can be expressed as a linear combination of delayed and advanced unit sample sequences, in the form

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

The response of the LTI system to an input $x[k]\, \delta[n-k]$

will be $x[k]\, h[n-k]$

Hence, the response y[n] to an input $x[n] = \sum_{k=-\infty}^{\infty} x[k]\, \delta[n-k]$

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

which can be alternately written as $y[n] = \sum_{k=-\infty}^{\infty} x[n-k]\, h[k]$

Convolution Sum

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

is called the **convolution sum** of the sequence x[n] and h[n]

and represented compactly as $y[n] = x[n] \otimes h[n]$ (convolução)

Properties:

Commutative property: $x[n] \otimes h[n] = h[n] \otimes x[n]$

Associative property: $(x[n] \otimes h[n]) \otimes y[n] = x[n] \otimes (h[n] \otimes y[n])$

Distributive property: $x[n] \otimes (h[n] \otimes y[n]) = x[n] \otimes h[n] + x[n] \otimes y[n]$

Convolution Sum

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

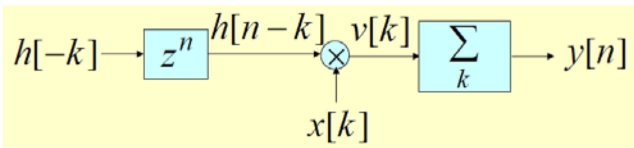
1) Time reversal h[k] to form h[-k]

2) Shift h[-k] n sampling elements to the right (if n>0) or left (if n<0), to form h[n-k]

3) Form the product $v[k] = x[k]\, h[n-k]$

4) Sum all samples of $v[k]=x[k]h[n-k]$ to compute the n^{th} sample of y[n]

Schematic Representation:



Convolution Sum

The computation of the convolution sum is simply a sum of products.

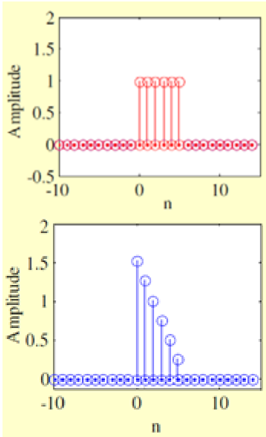
$y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

An example is presented next, for the following sequences x[n] and h[n]:

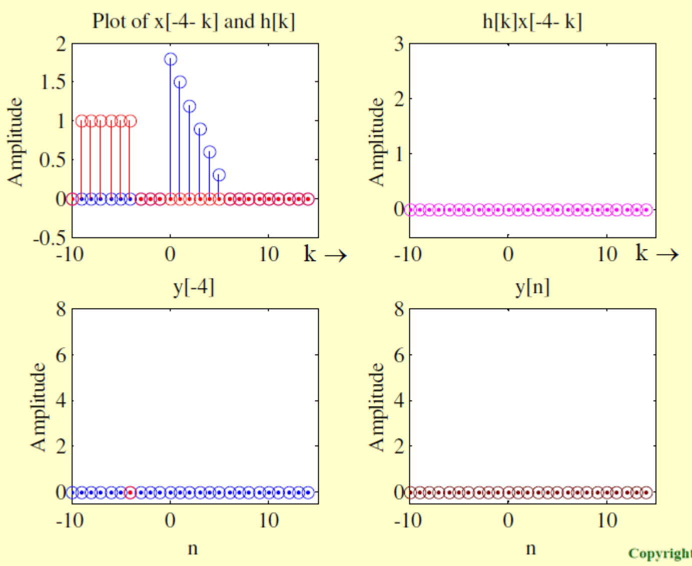
$$x[n] = \begin{cases} 1\,, & 0 \leq n \leq 5 \\ 0\,, & otherwise \end{cases}$$

$$h[n] = \begin{cases} 1.8 - 0.3n\,, & 0 \leq n \leq 5 \\ 0\,, & otherwise \end{cases}$$

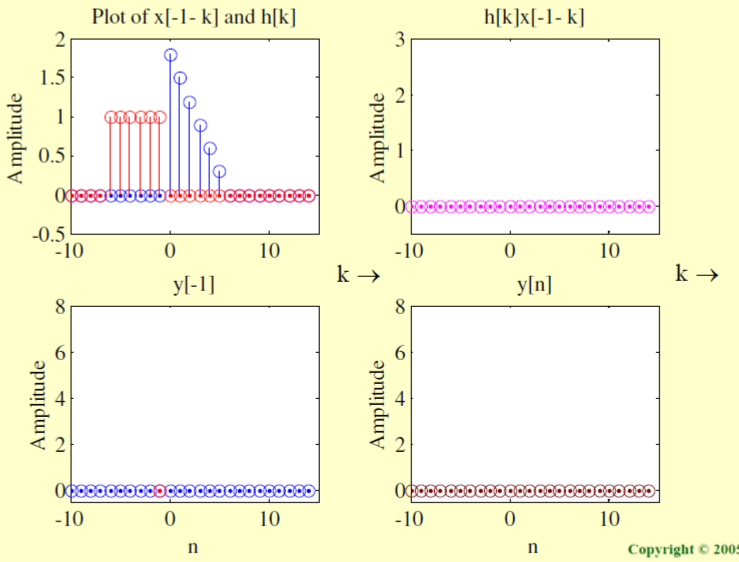
The figures in the next slides illustrate the steps involved in the computation of y[n] = x[n] ⊗ h[n]



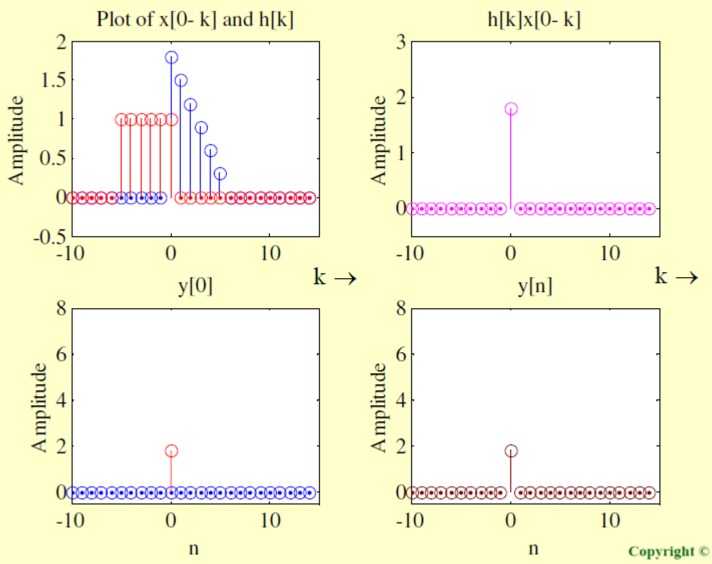
Convolution Sum



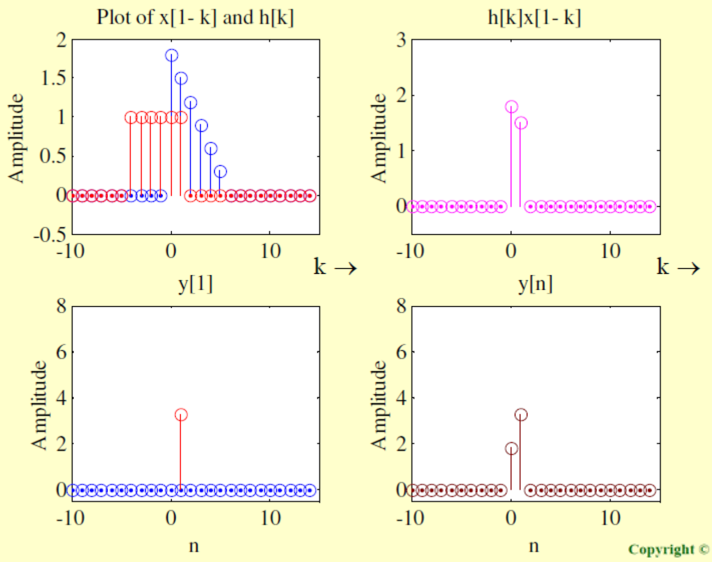
Convolution Sum



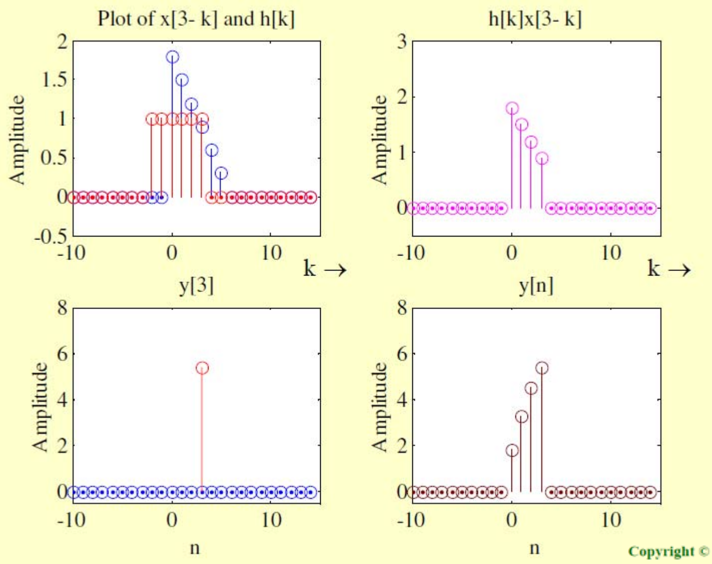
Convolution Sum



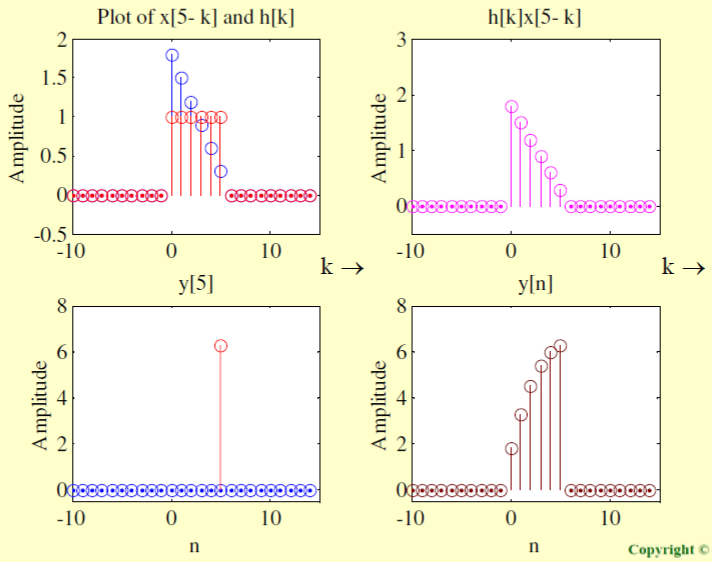
Convolution Sum



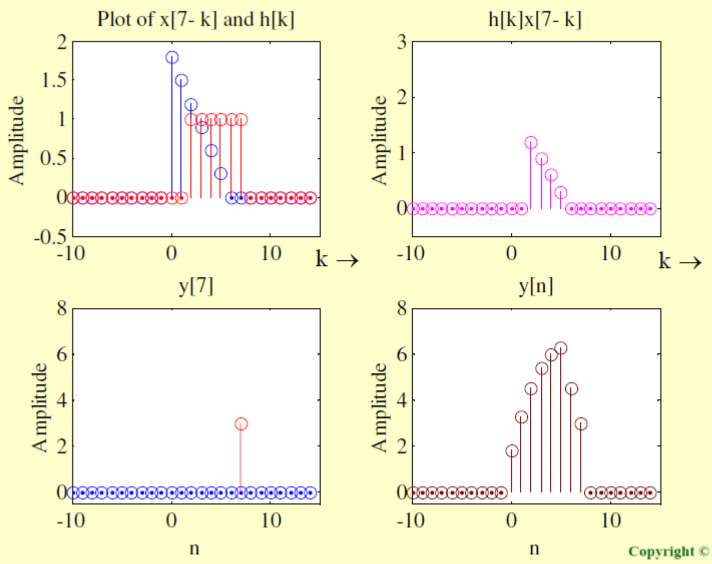
Convolution Sum



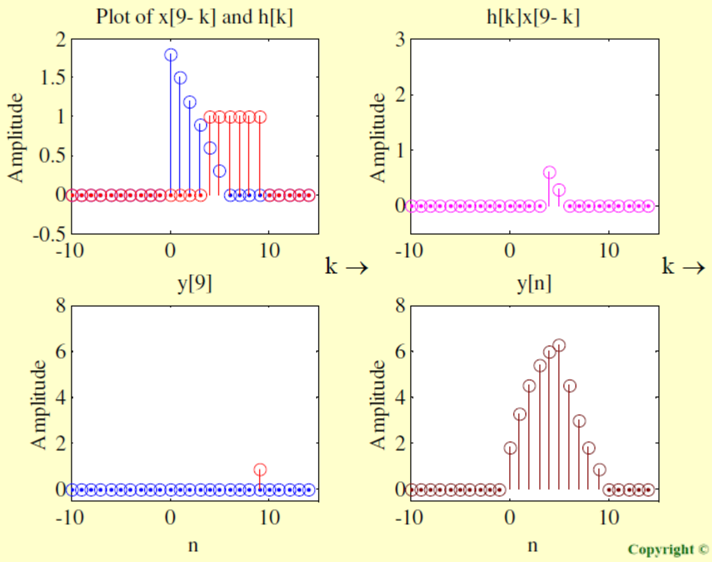
Convolution Sum



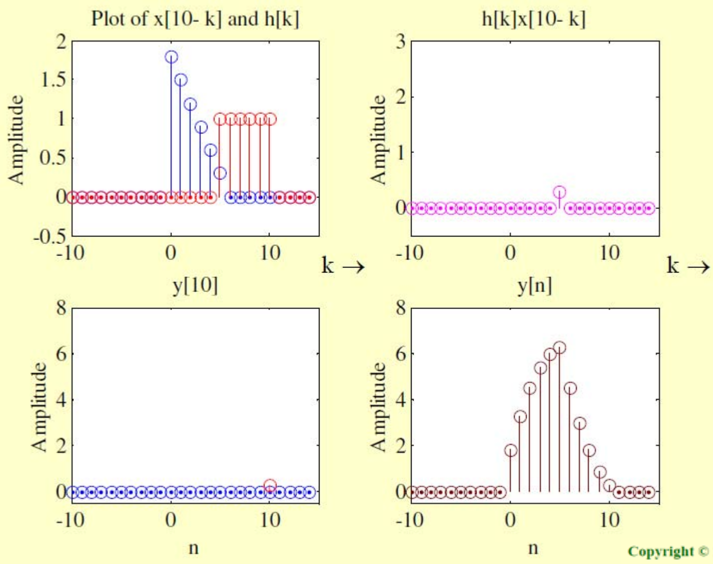
Convolution Sum



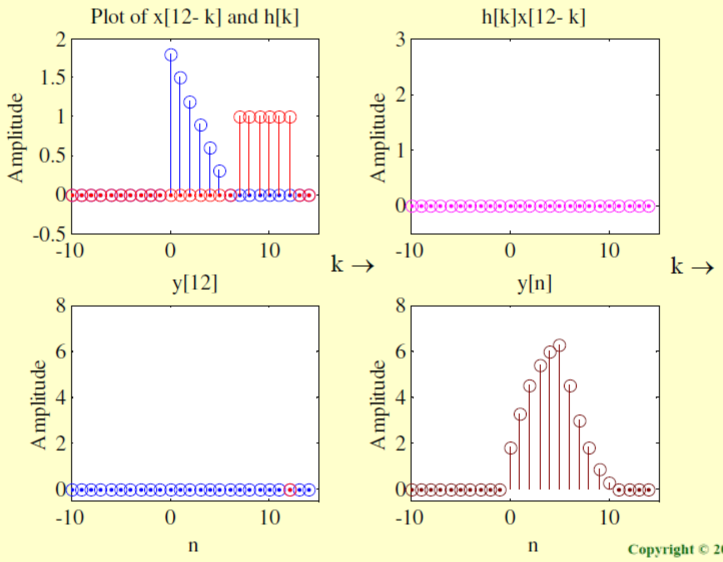
Convolution Sum



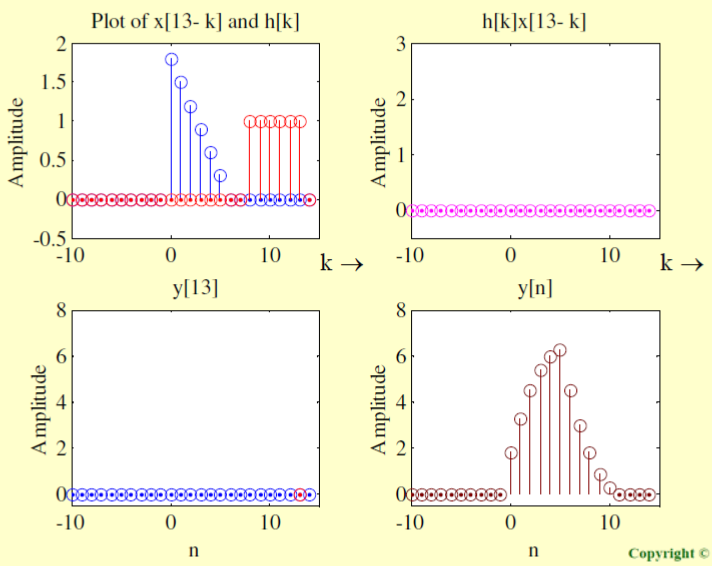
Convolution Sum



Convolution Sum



Convolution Sum



LTI Discrete-Time System

In practice, if either the input or the impulse response is of finite length, the convolution sum can be used to compute the output sample, as it involves a finite sum of products.

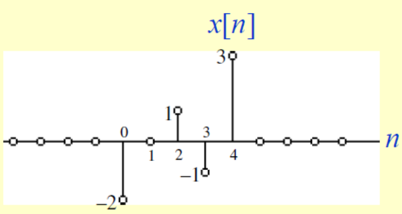
y[n] = x[n] ⊗ h[n]

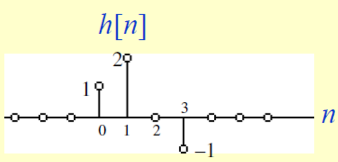
If both the input sequence (x[n]) and the impulse response (h[n]) are of **finite length**, the output sequence (y[n]) is also of finite length.

If both the input sequence and the impulse response are of **infinite length**, the convolution sum cannot be used to compute the output.

Time-Domain Characterisation of LTI System

Ex. The aim is to compute the output sequence y[n] of a system, given the input sequence x[n] and the impulse response h[n].

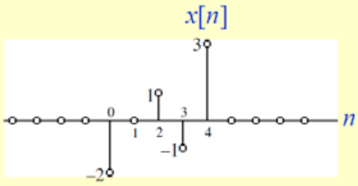


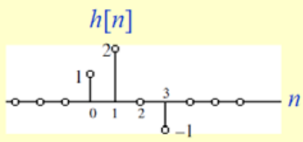


y[n] = x[n] ⊗ h[n]

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

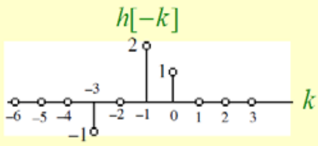
Time-Domain Characterisation of LTI System



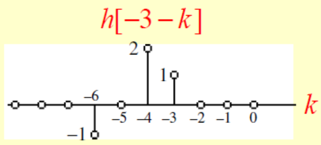


$y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

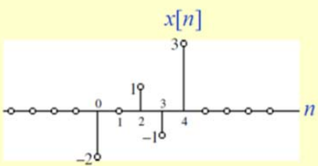
The initial step is to compute a time reversed version of the impulse response h[-k]

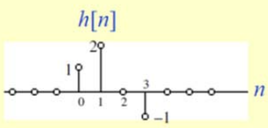


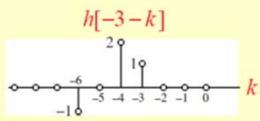
and then to time-shift it n elements e.g. for n=-3, h[-3-k]



Time-Domain Characterisation of LTI System

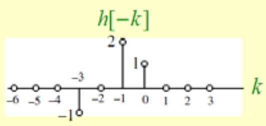




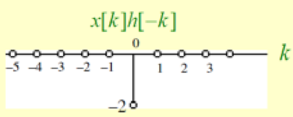


In this case (n=-3), the product of the k-th sample of x[k] and h[n-k] is always zero, thus y[-3]=0

In fact, y[n]=0 for all values of n<0

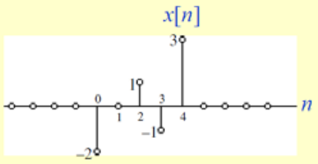


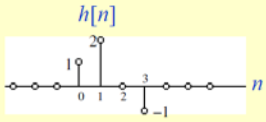
For n=0, h[-k] is used to compute the product sequence x[k]h[-k], which has a single non-zero element, for k=0.

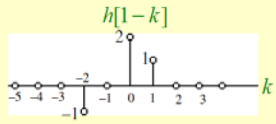


Thus, y[0] = x[0] h[0] = -2

Time-Domain Characterisation of LTI System

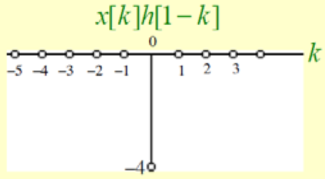






For the computation of y[1], the sequence h[-k] is shifted 1 sample to the right, to form h[1-k]

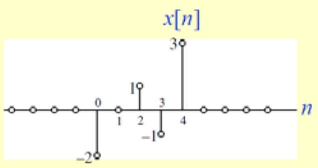
and then the product sequence x[k] h[1-k] is computed.

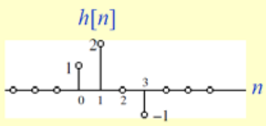


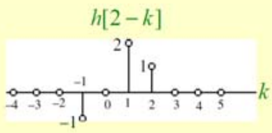
Hence,

y[1] = x[0] h[1] + x[1] h[0] = -4 + 0 = -4

Time-Domain Characterisation of LTI System

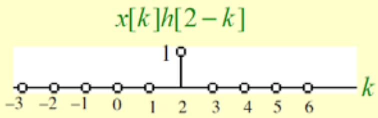






To calculate y[2], the sequence h[-k] is shifted 2 samples to the right, to form h[2-k]

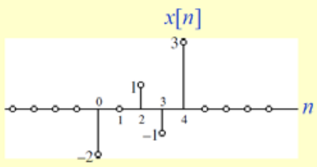
and then the product sequence x[k] h[2-k] is computed

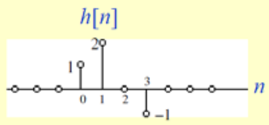


Hence,

y[2] = x[0] h[2] + x[1] h[1] + x[2] h[0] = 0 + 0 + 1 = 1

Time-Domain Characterisation of LTI System





Continuing the process, we get:

y[3] = x[0] h[3] + x[1] h[2] + x[2] h[1] + x[3] h[0] = 2 + 0 + 0 + 1 = 3

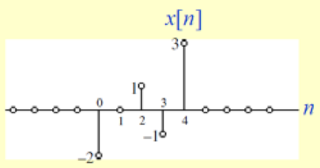
y[4] = x[1] h[3] + x[2] h[2] + x[3] h[1] + x[4] h[0] = 0 + 0 - 2 + 3 = 1

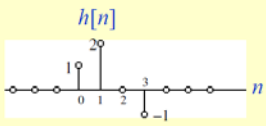
y[5] = x[2] h[3] + x[3] h[2] + x[4] h[1] + x[4] h[0] = -1 + 0 + 6 = 5

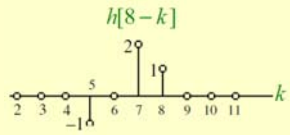
y[6] = x[3] h[3] + x[4] h[2] = 1 + 0 = 1

y[7] = x[4] h[3] = -3

Time-Domain Characterisation of LTI System

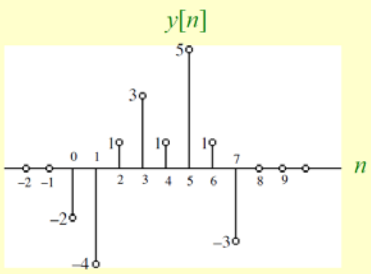






An inspection of the plots for x[k] and h[n-k] for n>7 (e.g. n=8, figure), indicates that there are no common non-zero values in these sequences.

As a result, y[n]=0 for n>7



The sequence y[n], generated by the convolution sum of x[n] and h[n] is the output signal of the system.

y[n] = x[n] ⊗ h[n]

Time-Domain Characterisation of LTI System

The sum of indices of each sample product inside the convolution sum is equal to the index of the sample being generated.

For example, y[3] = x[0] h[3] + x[1] h[2] + x[2] h[1] + x[3] h[0]

In general, if the 2 sequences being convolved are of lengths M and N, then the sequence generated by the convolution is of length M+N-1.

In the previous example, the convolution of

x[n] of length 5, with

h[n] of length 4, resulted in a sequence

y[n] of length 8 (=5+4-1)

Convolution Sum Computation – Tabular Method

The tabular method can be used to compute the convolution between 2 finite-length sequences.

Consider the convolution of g[n], 0≤n≤3, with h[n], 0≤n≤2, which generates a sequence y[n] = g[n] ⊗ h[n] of length 6 (4+3-1).

n:	0	1	2	3	4	5
$g[n]$:	$g[0]$	$g[1]$	$g[2]$	$g[3]$		
$h[n]$:	$h[0]$	$h[1]$	$h[2]$			
	$g[0]h[0]$	$g[1]h[0]$	$g[2]h[0]$	$g[3]h[0]$		
		$g[0]h[1]$	$g[1]h[1]$	$g[2]h[1]$	$g[3]h[1]$	
			$g[0]h[2]$	$g[1]h[2]$	$g[2]h[2]$	$g[3]h[2]$
$y[n]$:	$y[0]$	$y[1]$	$y[2]$	$y[3]$	$y[4]$	$y[5]$

Convolution Sum Computation – Tabular Method

The samples of y[n] are :

y[0] = g[0] h[0]
y[1] = g[1] h[0] + g[0] h[1]
y[2] = g[2] h[0] + g[1] h[1] + g[0] h[2]
y[3] = g[3] h[0] + g[2] h[1] + g[1] h[2]
y[4] = g[3] h[1] + g[2] h[2]
y[5] = g[3] h[2]

The method can be also applied to compute the convolution of 2 finite-lenght two-sided sequences.

In this case, a decimal point is first placed to the right of the sample with the time index (n=0), for each sequence. Then, the tabular method is applied ignoring the decimal points.

The decimal point is then inserted, using the convention multiplication rules.

Convolution Sum Computation – Using MATLAB

The MATLAB function `conv` implements the convolution sum of 2 finite-lenght sequences.

For example,

```
a = [-2 0 1 -1 3]

b = [1 2 0 -1]

y = conv(a,b)
```

Exercício: Fazer com lápis e papel…

```
[-2 -4 1 3 1 5 1 -3]
```

Stability Condition of a LTI System

A discrete-time system is **BIBO stable** (bounded-input, bounded-output) if and only if the output y[n] remains bounded for all bounded inputs x[n].

An LTI discrete-time system is **BIBO stable** if and only if its impulse response h[n] is absolutely summable, i.e. $S = \sum_{n=0}^{\infty} | h[n] | < \infty$

Proof: Since x[n] is bounded, we have $| x[n] | \le B_x < \infty$. Therefore,

$$| y[n] | = | \sum_{k=0}^{M-1} h[k]x[n-k] | \le \sum_{k=0}^{M-1} |h[n]|\,|x[n-k]| \le B_x \sum_{k=0}^{M-1} |h[n]| \le B_x S$$

Thus, $S<\infty \Rightarrow |y[n]| \le B_y < \infty$, which indicates that y[n] is also bounded.

Now, let us assume that y[n] is bounded, i.e. $|y[n]| \le B_y < \infty$, and x[n] as

$$x[n] = \begin{cases} h[-n]/|h[-n]|, & if\ h[-n] \neq 0 \\ K, & if\ h[-n] = 0 \end{cases} \quad \text{with } |K| \le 1.$$

x[n] is bounded, since $|x[n]| \le 1$.

For this input, $y[0] = \sum_{k=0}^{M-1} (h[k]/|h[k]|).h[k] = S \le B_y < \infty$

Therefore, $|y[n]| \le B_y \Rightarrow S<\infty$

Stability Condition of a LTI System

Example

Consider a LTI discrete-time system, with the following impulse response h[n]

$$h[n] = \alpha^n \mu[n]$$

For this system,

$$S = \sum_{n=-}^{\infty} |\alpha^n \mu[n]| < \sum_{n=0}^{\infty} |\alpha|^n = \frac{1}{1-|\alpha|} \quad \text{if } |\alpha|<1$$

Therefore $S<\infty$ if $|\alpha|<1$, in which case the system is BIBO stable.

If $|\alpha| \ge 1$, the system is not BIBO stable.

Causality Condition of a LTI System

Let $x_1[n]$ and $x_2[n]$ be two input sequences, with $x_1[n]=x_2[n]$ for $n \le n_0$.

The corresponding output samples of an LTI system with impulse response h[n], at $n=n_0$, are:

$y_1[n_0] = \sum_{k=-\infty}^{\infty} h[k]\, x_1[n_0-k] = \sum_{k=0}^{\infty} h[k]\, x_1[n_0-k] + \sum_{k=-\infty}^{-1} h[k]\, x_1[n_0-k]$

$y_2[n_0] = \sum_{k=-\infty}^{\infty} h[k]\, x_2[n_0-k] = \sum_{k=0}^{\infty} h[k]\, x_2[n_0-k] + \sum_{k=-\infty}^{-1} h[k]\, x_2[n_0-k]$

If the LTI system is causal, then $y_1[n_0]$ must be equal to $y_2[n_0]$.

Once $x_1[n]=x_2[n]$, we have $\sum_{k=0}^{\infty} h[k]\, x_1[n_0-k] = \sum_{k=0}^{\infty} h[k]\, x_2[n_0-k]$

thus $y_1[n]=y_2[n]$ implies that $\sum_{k=-\infty}^{-1} h[k]\, x_1[n_0-k] = \sum_{k=-\infty}^{-1} h[k]\, x_2[n_0-k]$

As $x_1[n]$ is not equal to $x_2[n]$ for $n>n_0$, $h[k]=0$ for $k<0$

As a result, **an LTI system is causal if and only if h[n] is causal**.

Causality Condition of a LTI System

An LTI discrete-time system is **causal** if and only if its impulse response h[n] is a causal sequence.

Example: The discrete-time system defined by

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

is a causal system, as it has a causal impulse response $h[n] = \{ \underset{\uparrow}{\alpha_1}, \alpha_2, \alpha_3, \alpha_4 \}$

Example: The discrete-time accumulator, defined by

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

is a causal system, as it has a causal impulse response

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

Causality Condition of a LTI System

Example:

A factor-of-2 interpolator, defined by

y[n] = x[n] + 0.5 (x[n-1] + x[n+1]) is non-causal,

as its impulse response h[n] = { 0.5 , 1 , 0.5 } is non-causal

↑

A non-causal system can be implemented as a causal system by delaying the output.

In this example, by delaying the output by one sample, a causal version of the factor-of-2 interpolator is obtained

y[n] = x[n-1] + 0.5 (x[n-2] + x[n])

LTI system / constant coefficient difference equation

An important subclass of LTI discrete-time systems is characterized by a **linear constant coefficient difference equation**, of the form

$$\sum_{k=0}^{N} d_k\, y[n-k] = \sum_{k=0}^{M} p_k\, x[n-k]$$

where x[n] and y[n] are the input and output sequences, and $\{d_k\}$ and $\{p_k\}$ are constants characterizing the system.

The **order** of the system is given by **max(N,M)**, which is the order of the difference equation.

If the system is causal, then the output y[n] can be recursively computed by

$$y[n] = -\sum_{k=1}^{N} \frac{d_k}{d_0} y[n-k] + \sum_{k=0}^{M} \frac{p_k}{d_0} x[n-k] \text{ , for } d_0 \neq 0$$

y[n] can thus be computed for all $n \geq n_0$, knowing x[n] and the initial conditions $y[n_0-1]$, $y[n_0-2]$, …, $y[n_0-N]$

LTI system / constant coefficient difference equation

A LTI discrete-time system characterized by a **linear constant coefficient difference equation**, can be implemented in MATLAB using the function `filter`.

$$\sum_{k=0}^{N} d_k\, y[n-k] = \sum_{k=0}^{M} p_k\, x[n-k]$$

The standard syntax is `y=filter(b,a,x)`, where x and y are the input and output sequences, a is a vector with the coefficients p_k, and b is a vector with the coefficients d_k.

Example: An average filter of window size 3, corresponds to the equation

$$y[n] = \frac{1}{3}\, x[n] + \frac{1}{3}\, x[n-1] + \frac{1}{3}\, x[n-2]$$

which can be implemented in MATLAB as:

```
y=filter([1/3,1/3,1/3],1,x)
```


LTI system / constant coefficient difference equation

Example: An average filter of window size 5, corresponds to the equation

$$y[n] = \frac{1}{5}\, x[n] + \frac{1}{5}\, x[n-1] + \frac{1}{5}\, x[n-2] + \frac{1}{5}\, x[n-3] + \frac{1}{5}\, x[n-4]$$

which can be implemented in MATLAB as:

```
b=(1/5)*ones(1,5);
y=filter(b,1,x);
```

More generally, for an average filter of window size n:

```
b=(1/n)*ones(1,n);
y=filter(b,1,x);
```

Classification of LTI Discrete-Time Systems

Based on the Impulse Response Length:

- **Finite Impulse Response (FIR)** – If the impulse response h[n] is of finite length, i.e. h[n]=0 for $n<N_1$ and $n>N_2$, with $N_1<N_2$.
- **Infinite Impulse Response (IIR)** – If h[n] is of infinite length.

Examples:

- The moving-average system and the linear interpolators are examples of FIR systems.
- The discrete-time accumulator, defined by

$$y[n] = \sum_{k=-\infty}^{n} x[k] = y[n-1] + x[n]$$

is an IIR system.

Classification of LTI Discrete-Time Systems

Based on the Output Calculation Process:

- **Non-recursive System** – The output can be calculated sequentially, knowing only the present and past input samples.
- **Recursive System** – The output computation involves past output samples, in addition to present and past input samples.

Based on the Coefficients:

- **Real Discrete-Time System** – The impulse response samples are real valued.
- **Complex Discrete-Time System** – The impulse response samples are complex valued.

Recursive System

Example

This recursive system can be used to compute $\sqrt{A}$, for $A \geq 0$

$$y[n] = \frac{1}{2}\left(y[n-1] + \frac{x[n]}{y[n-1]} \right) \quad , \quad n=0,1,2,\ldots$$

with y[-1] being the initial estimate for $\sqrt{A}$ and $x[n]=A\mu[n]$.

Using an input signal $x[n]=A\mu[n]$ and an initial estimate y[-1], the system output y[n] will converge no $\sqrt{A}$.

-1	2	1	
n	x[n]	y[n]	y[n]-y[n-1]
0	2	1,5	0,5
1	2	1,4166667	-0,0833333
2	2	1,4142157	-0,002451
3	2	1,4142136	-2,124E-06
4	2	1,4142136	-1,595E-12
5	2	1,4142136	0
6	2	1,4142136	0
7	2	1,4142136	0
8	2	1,4142136	0

-1	2	7	
n	x[n]	y[n]	y[n]-y[n-1]
0	2	3,6428571	-3,3571429
1	2	2,0959384	-1,5469188
2	2	1,5250825	-0,5708559
3	2	1,4182435	-0,106839
4	2	1,4142193	-0,0040242
5	2	1,4142136	-5,725E-06
6	2	1,4142136	-1,159E-11
7	2	1,4142136	0
8	2	1,4142136	0

Correlation of Signals

In many applications it might be necessary to compare one signal with 1 or more reference signals, to determine their similarity.

For example, in digital communications, a set of data symbols (numbers or other characters) are represented by a set of unique discrete-time sequences.

When a sequence is transmitted, the receiver has to determine which particular sequence it is, by comparing the signal received with every possible sequence from the set.

The detection or recognition problem gets more complicated in practice, as the signal is often corrupted by additive random noise.

Correlation of Signals

A measure of similarity between a pair of energy signals, x[n] and y[n], is given by the **cross-correlation** sequence $r_{xy}[l]$, defined by

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} x[n]\, y[n-l] \quad \text{, with } l = 0, \pm 1, \pm 2, \ldots$$

The parameter l (lag) indicates the time-shift between the signals

- The ordering of the subscripts in $r_{xy}[l]$ specifies that x[n] is the reference sequence, which remains fixed in time.
- y[n] is said to be shifted by l samples to the right (left) with respect to the reference x[n] for positive (negative) values of l.

If y[n] is used as reference, the cross-correlation sequence $r_{yx}[l]$ is

$$r_{yx}[l] = \sum_{n=-\infty}^{\infty} y[n]\, x[n-l] = \sum_{m=-\infty}^{\infty} y[m+l]\, x[m] = r_{xy}[-l]$$

thus $r_{yx}[l]$ is obtained by time-reversing $r_{xy}[l]$.

Correlation of Signals

The **auto-correlation** sequence $r_{xx}[l]$, for signal x[n], is given by

$$r_{xx}[l] = \sum_{n=-\infty}^{\infty} x[n]\, x[n-l] \quad \text{, with } l = 0, \pm 1, \pm 2, \ldots$$

The value of $r_{xx}[0]$ is the energy of the signal x[n].

$$r_{xx}[0] = \sum_{n=-\infty}^{\infty} x[n]\, x[n-0] = \sum_{n=-}^{\infty} x^2[n] = E_x$$

Considering that $r_{yx}[l] = r_{xy}[-l]$, it follows that $r_{xx}[l] = r_{xx}[-l]$, which implies that $r_{xx}[l]$ is an even function for real x[n].

Correlation of Signals

An examination of $r_{xy}[l] = \sum_{n=-}^{\infty}$ x[n] y[n−l] reveals that the expression of r_{xy} looks quite similar to that of the linear convolution

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} x[n]\, y[n-l] = \sum_{n=-}^{\infty} x[n]\, y[-(l-n)] = x[l] \otimes y[-l]$$

The cross-correlation of y[n] with the reference x[n] can be computed by processing x[n] with na LTI system of impulse response h[n]=y[-n]

$$x[n] \longrightarrow \boxed{y[-n]} \longrightarrow r_{xy}[n]$$

Similarly, the auto-correlation of x[n] can be computed by processing x[n] with na LTI system of impulse response h[n]=x[-n]

$$x[n] \longrightarrow \boxed{x[-n]} \longrightarrow r_{xx}[n]$$

Properties of Auto-correlation and Cross-correlation

- Consider two finite-energy sequences $x[n]$ and $y[n]$
- The energy of the combined sequence $a\,x[n] + y[n-\ell]$ is also finite and nonnegative, i.e.,

$$\sum_{n=-\infty}^{\infty} (a\,x[n] + y[n-\ell])^2 = a^2 \sum_{n=-\infty}^{\infty} x^2[n]$$

$$+ 2a \sum_{n=-\infty}^{\infty} x[n]y[n-\ell] + \sum_{n=-\infty}^{\infty} y^2[n-\ell] \geq 0$$

- Thus $a^2 r_{xx}[0] + 2a\, r_{xy}[\ell] + r_{yy}[0] \geq 0$

  where $r_{xx}[0] = \mathcal{E}_x > 0$ and $r_{yy}[0] = \mathcal{E}_y > 0$

Properties of Auto-correlation and Cross-correlation

- We can rewrite the equation on the previous slide as

$$[a \quad 1]\begin{bmatrix} r_{xx}[0] & r_{xy}[\ell] \\ r_{xy}[\ell] & r_{yy}[0] \end{bmatrix}\begin{bmatrix} a \\ 1 \end{bmatrix} \geq 0$$

for any finite value of a

- Or, in other words, the matrix $\begin{bmatrix} r_{xx}[0] & r_{xy}[\ell] \\ r_{xy}[\ell] & r_{yy}[0] \end{bmatrix}$ is positive semidefinite

$\Rightarrow$ $r_{xx}[0]r_{yy}[0] - r_{xy}^2[\ell] \geq 0$

or, equivalently,

$$|r_{xy}[\ell]| \leq \sqrt{r_{xx}[0]r_{yy}[0]} = \sqrt{\mathcal{E}_x\mathcal{E}_y}$$

Properties of Auto-correlation and Cross-correlation

- The last inequality on the previous slide provides an upper bound for the cross-correlation samples

$$|r_{xy}[\ell]| \leq \sqrt{r_{xx}[0]r_{yy}[0]} = \sqrt{\mathcal{E}_x\mathcal{E}_y}$$

- If we set $y[n] = x[n]$, then the inequality reduces to

$$|r_{xy}[\ell]| \leq r_{xx}[0] = \mathcal{E}_x$$

- Thus, at zero lag ($\ell = 0$), the sample value of the autocorrelation sequence has its maximum value

Properties of Auto-correlation and Cross-correlation

- Now consider the case $y[n] = \pm b\,x[n-N]$ where N is an integer and $b > 0$ is an arbitrary number
- In this case $\mathcal{E}_y = b^2 \mathcal{E}_x$
- Therefore $\sqrt{\mathcal{E}_x \mathcal{E}_y} = \sqrt{b^2 \mathcal{E}_x^2} = b\mathcal{E}_x$
- Using the above result in

$$|r_{xy}[\ell]| \le \sqrt{r_{xx}[0]r_{yy}[0]} = \sqrt{\mathcal{E}_x \mathcal{E}_y}$$

we get

$$-b\,r_{xx}[0] \le r_{xy}[\ell] \le b\,r_{xx}[0]$$

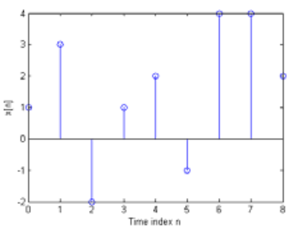
Correlation computation using MATLAB

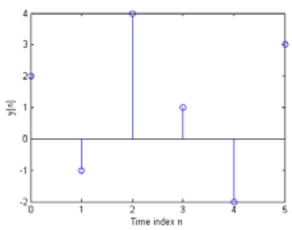
The cross-correlation and auto-correlation sequences can be easily computed using MATLAB – `corr`

Example: Consider two finite-length sequences

x[n] = {1, 3, -2, 1, 2, -1, 4, 4, 2}

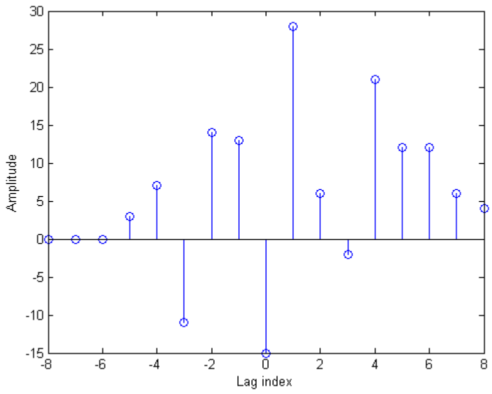
y[n] = {2, -1, 4, 1, -2, 3}





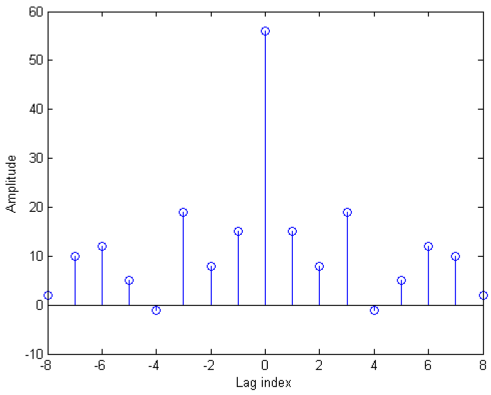
Correlation computation using MATLAB

The cross-correlation sequence $r_{xy}[n]$



Correlation computation using MATLAB

The auto-correlation sequence $r_{xx}[n]$.

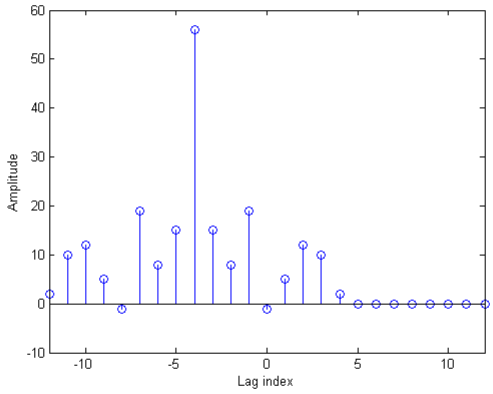


At zero lag,

$r_{xx}[n]$ is max.

Correlation computation using MATLAB

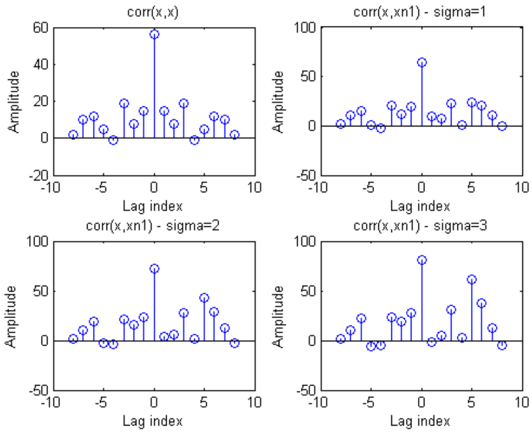
cross-correlation of x[n] and y[n]=x[n-N], for N=4,



The peak of the cross- correlation is precisely the value of the delay N.

Correlation computation using MATLAB

Cross-correlation of x[n] and x'[n], corrupted with noise,



Additive random noise

Zero-Mean Gaussian

σ=1,2,3

Normalized Forms of Correlation

- Normalized forms of autocorrelation and cross-correlation are given by

$$\rho_{xx}[\ell] = \frac{r_{xx}[\ell]}{r_{xx}[0]} \qquad \rho_{xy}[\ell] = \frac{r_{xy}[\ell]}{\sqrt{r_{xx}[0] r_{yy}[0]}}$$

- They are often used for convenience in comparing and displaying

Note: $|\rho_{xx}[\ell]| \leq 1$ and $|\rho_{xy}[\ell]| \leq 1$ independent of the range of values of $x[n]$ and $y[n]$

Correlation Computation for Power Signals

- The cross-correlation sequence for a pair of power signals, $x[n]$ and $y[n]$, is defined as

$$r_{xy}[\ell] = \lim_{K \to \infty} \frac{1}{2K+1} \sum_{n=-K}^{K} x[n] y[n-\ell]$$

- The autocorrelation sequence of a power signal $x[n]$ is given by

$$r_{xx}[\ell] = \lim_{K \to \infty} \frac{1}{2K+1} \sum_{n=-K}^{K} x[n] x[n-\ell]$$

Correlation Computation for Periodic Signals

- The cross-correlation sequence for a pair of periodic signals of period N, $\tilde{x}[n]$ and $\tilde{y}[n]$, is defined as
$$r_{\tilde{x}\tilde{y}}[\ell] = \frac{1}{N}\sum_{n=0}^{N-1}\tilde{x}[n]\tilde{y}[n-\ell]$$
- The autocorrelation sequence of a periodic signal $\tilde{x}[n]$ of period N is given by
$$r_{\tilde{x}\tilde{x}}[\ell] = \frac{1}{N}\sum_{n=0}^{N-1}\tilde{x}[n]\tilde{x}[n-\ell]$$

Note: Both $r_{\tilde{x}\tilde{y}}[\ell]$ and $r_{\tilde{x}\tilde{x}}[\ell]$ are also periodic signals with a period N

Correlation Computation for Periodic Signals

- The periodicity property of the autocorrelation sequence can be exploited to determine the period of a periodic signal that may have been corrupted by an additive random disturbance
- Let $\tilde{x}[n]$ be a periodic signal corrupted by the random noise $d[n]$ resulting in the signal
$$w[n] = \tilde{x}[n] + d[n]$$
which is observed for $0 \le n \le M-1$ where $M \gg N$

Correlation Computation for Periodic Signals

- The autocorrelation of $w[n]$ is given by

$$r_{ww}[\ell] = \frac{1}{M}\sum_{n=0}^{M-1} w[n]w[n-\ell]$$

$$= \frac{1}{M}\sum_{n=0}^{M-1} (\tilde{x}[n]+d[n])(\tilde{x}[n-\ell]+d[n-\ell])$$

$$= \frac{1}{M}\sum_{n=0}^{M-1} \tilde{x}[n]\tilde{x}[n-\ell] + \frac{1}{M}\sum_{n=0}^{M-1} d[n]d[n-\ell]$$

$$+ \frac{1}{M}\sum_{n=0}^{M-1} \tilde{x}[n]d[n-\ell] + \frac{1}{M}\sum_{n=0}^{M-1} d[n]\tilde{x}[n-\ell]$$

$$= r_{\tilde{x}\tilde{x}}[\ell] + r_{dd}[\ell] + r_{\tilde{x}d}[\ell] + r_{d\tilde{x}}[\ell]$$

$r_{\tilde{x}\tilde{x}}[\ell]$ is a periodic sequence with a period N and hence will have peaks at $\ell = 0, N, 2N, \ldots$ with the same amplitudes as ℓ approaches M

Correlation Computation for Periodic Signals

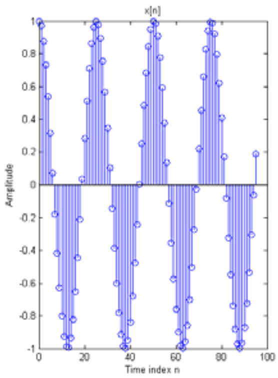
- As $\tilde{x}[n]$ and $d[n]$ are not correlated, samples of cross-correlation sequences $r_{\tilde{x}d}[\ell]$ and $r_{d\tilde{x}}[\ell]$ are likely to be very small relative to the amplitudes of $r_{\tilde{x}\tilde{x}}[\ell]$
- The autocorrelation $r_{dd}[\ell]$ of $d[n]$ will show a peak at $\ell = 0$ with other samples having rapidly decreasing amplitudes with increasing values of $|\ell|$
- Hence, peaks of $r_{ww}[\ell]$ for $\ell > 0$ are essentially due to the peaks of $r_{\tilde{x}\tilde{x}}[\ell]$ and can be used to determine whether $\tilde{x}[n]$ is a periodic sequence and also its period N if the peaks occur at periodic intervals

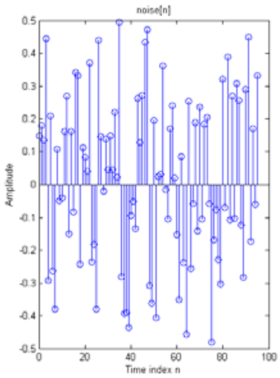
Correlation Computation of a periodic signal (MATLAB)

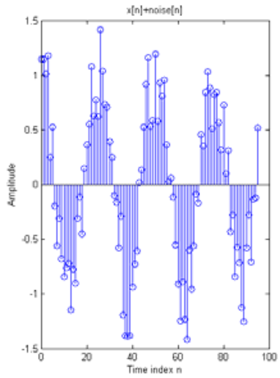
Example: Determine the period of the sinusoidal sequence

$x[n]=\cos(0.25n)$, $0 \le n \le 95$

corrupted by an additive, uniformly distributed, random noise of amplitude in the range [-0.5,0.5].

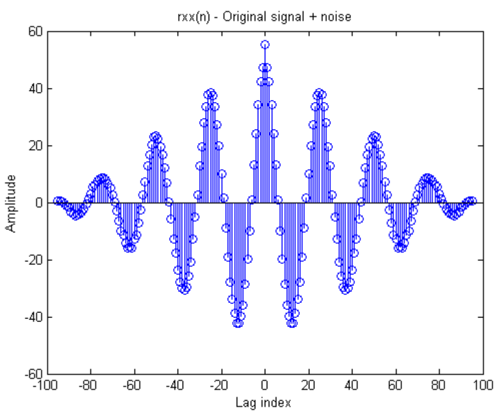






Correlation Computation of a periodic signal (MATLAB)

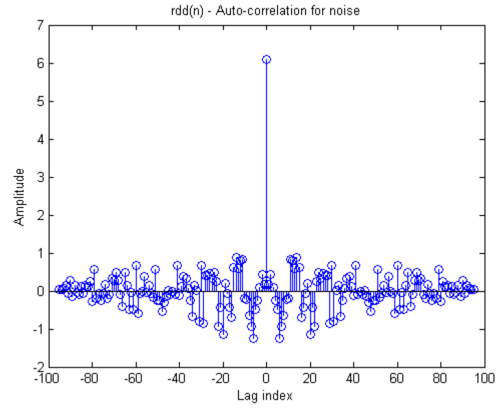
The plot shows $r_{xx}[n]$, the autocorrelation of the signal corrupted by additive noise, as a function of the time lag.



- There is a strong peak at zero lag.
- There are distinct peaks at intervals of 8π, which is the period of the sequence.

Correlation Computation of a periodic signal (MATLAB)

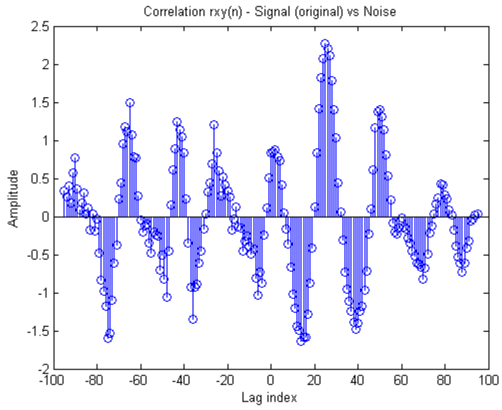
The plot shows $r_{dd}[n]$ the autocorrelation of the noise component only, as a function of the time lag.



- As expected, there is a single strong peak at zero lag.
- The remaining values are low.

Correlation Computation of a periodic signal (MATLAB)

The plot shows $r_{dd}[n]$ the correlation between the signal (original) and the noise component, as a function of the time lag.



- The values are much lower than those for the autocorrelation of the signal + noise