Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP
Signal Processing

Prof. Dr.-Ing. Timo Gerkmann

Digital Media Signal Processing

2. Discrete-Time Signals and Systems

Outline

Motivation

- While sinusoids are very elementary signals, here we introduce more elementary signals that facilitate signal processing
  - elementary signals are used as building blocks for more complex signals
- Introduce and analyze the concept of Discrete-Time Systems
- What are the advantages of linear time-invariant (LTI) systems?
  - Large collection of mathematical techniques to elegantly describe LTI systems
  - Many practical systems are (approximately) LTI
    - Rooms that introduce reverberation
    - Many filters (lowpass, highpass, bandpass)
- Introduction of measures for signal correlation
  - autocorrelation, crosscorrelation

Outline

1. The *unit sample sequence* / *unit impulse* is denoted as $\delta(n)$ and is defined as

$$\delta(n) \equiv \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{for } n \neq 0 \end{cases}$$

related to time domain unit impulse $\delta(t)$ (but much simpler).

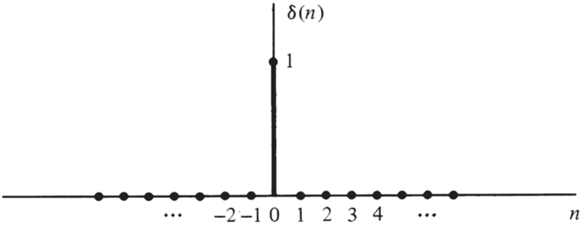


Figure 2.1.2 Graphical representation of the unit sample signal.

2. The *unit step signal*

$$u(n) \equiv \begin{cases} 1, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases}$$

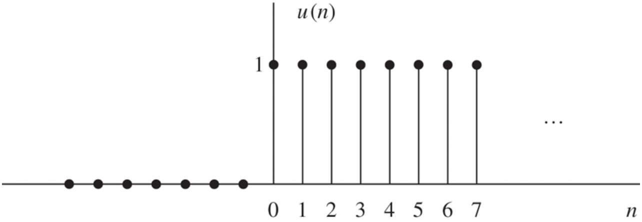


Figure 2.1.3 Graphical representation of the unit step signal.

3. The *unit ramp* signal

$$u_{\mathrm{r}}(n) \equiv \begin{cases} n, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases}$$

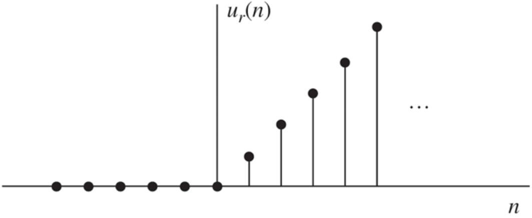


Figure 2.1.4 Graphical representation of the unit ramp signal.

4. The *exponential signal* has the form

$$x(n) \equiv a^n \quad \text{for all } n$$

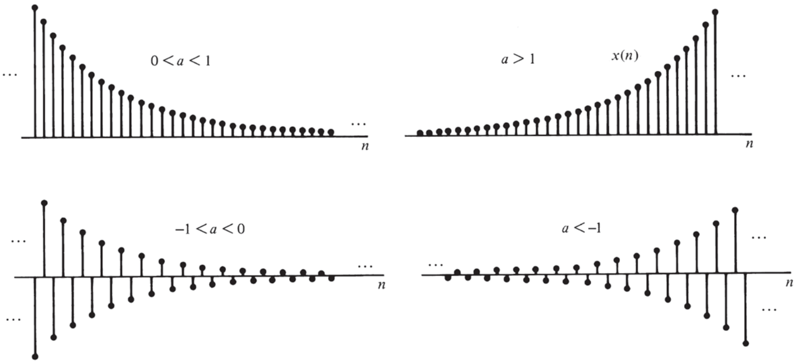


Figure 2.1.5 Graphical representation of exponential signals.

Elementary Discrete-Time Signals V

4. The *exponential signal* has the form

$$x(n) \equiv a^n \quad \text{for all } n$$

- if a is real-valued $\rightarrow x(n)$ is real-valued
- when complex-valued we have $a \equiv re^{j\theta}$, and

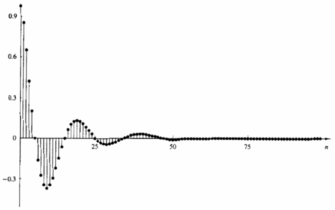
$$\begin{aligned} x(n) &= r^n e^{j\theta n} \\ x(n) &= r^n (\cos\theta n + j \sin\theta n) \end{aligned}$$

Representation of complex numbers

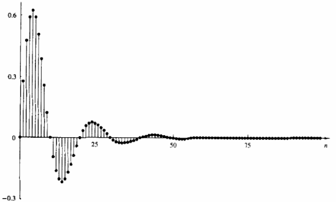
Cartesian $x(n) = x_R(n) + jx_I(n)$ $\quad (= r^n \cos\theta n + jr^n \sin\theta n)$

Polar $x(n) = |x(n)| e^{j\angle x(n)}$ $\quad (= r^n e^{j\theta n})$

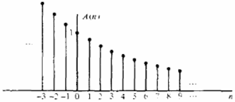
- cartesian representation (left) and polar representation (right)
  $x(n) = (0.9 e^{j\frac{\pi}{10}})^n$



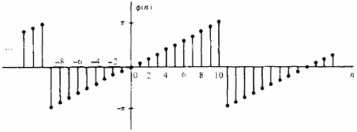
(a) Graph of $x_R(n) = (0.9)^n \cos \frac{\pi n}{10}$



(b) Graph of $x_I(n) = (0.9)^n \sin \frac{\pi n}{10}$



(a) Graph of $A(n) = r^n$, $r = 0.9$



(b) Graph of $\phi(n) = \frac{\pi}{10} n$, modulo 2π plotted in the range $(-\pi, \pi)$

Energy and power signals

- The energy E of a signal $x(n)$ is defined as

$$E \equiv \sum_{n=-\infty}^{\infty} |x(n)|^2$$

  ➡ if $0 < E < \infty$ (i.e. E is finite), $x(n)$ is called **energy signal**

- The average power of a signal is defined as

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

  - if $0 < E < \infty$ (i.e. E is finite), then $P = 0$
  - if E is infinite, P can either be finite or infinite

  ➡ if P is finite, $x(n)$ is called **power signal**

Periodic and aperiodic signals

- A signal $x(n)$ is peridodic with period N $(N > 0)$, iff

$$x(n+N) = x(n) \quad \text{for all } n \tag{1}$$

- the smallest N for which (1) holds is called **fundamental period**
- if there is no N for which (1), $x(n)$ is called **nonperiodic**
- Periodic signals with $x(n) < \infty$ $(\forall n)$, are power signals, i.e. they have finite power

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2$$

Even and Odd Signals

- A real-valued signal $x(n)$ is even if

$$x(-n) \equiv x(n)$$

- A real-valued signal $x(n)$ is odd if

$$x(-n) \equiv -x(n)$$

  - implies $x(0) = 0$

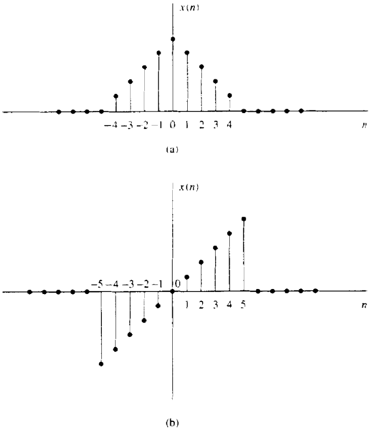


Fig.: even (a) and odd (b) signals

Even and Odd Signals

In general, a signal $x(n)$ can be decomposed

- into an even part $x_\mathrm{e}(-n) = x_\mathrm{e}(n)$, as

$$x_\mathrm{e}(n) = \frac{1}{2}\left[x(n) + x(-n)\right]$$

- and an odd part $x_\mathrm{o}(-n) = -x_\mathrm{o}(n)$, as

$$x_\mathrm{o}(n) = \frac{1}{2}\left[x(n) - x(-n)\right]$$

such that $x(n) = x_\mathrm{e}(n) + x_\mathrm{o}(n)$

SP Signal Processing

Time shift A signal delay by $k > 0$ samples is obtained by replacing n by $n - k$,

$$\mathrm{TD}_k\left[x(n)\right] = x(n-k)$$

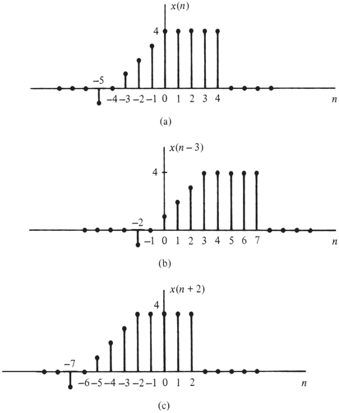


Figure 2.1.9 Graphical representation of a signal, and its delayed and advanced versions.

Folding A folding at origin $n = 0$ is obtained as

$$\mathrm{FD}\left[x(n)\right] = x(-n)$$

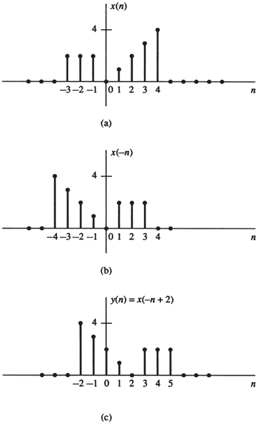


Figure 2.1.10 Graphical illustration of the folding and shifting operations.

- Note that the time dependent operations of folding and delaying are not commutative. I.e.

$$\mathrm{TD}_k \left\{\mathrm{FD}\left[x(n)\right]\right\} = \mathrm{TD}_k \left[x(-n)\right] = x(-n+k)$$

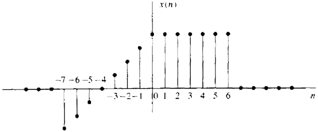
while

$$\mathrm{FD}\left\{\mathrm{TD}_k \left[x(n)\right]\right\} = \mathrm{FD}\left[x(n-k)\right] = x(-n-k)$$

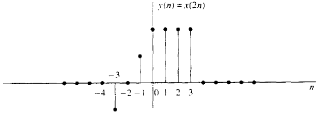
SP Signal Processing

Down-sampling A down-sampling is obtained by

$$\mathrm{TS}_\mu\left[x(n)\right] = x(\mu n), \quad \mu = 2, 3, 4, \ldots$$



(a)



(b)

➡ Be careful with sampling theorem!

Simple Modifications of Signals V

Amplitude Scaling of a signal with constant A

$$y(n) = Ax(n), \quad -\infty < n < \infty$$

Summation of two signals

$$y(n) = x_1(n) + x_2(n), \quad -\infty < n < \infty$$

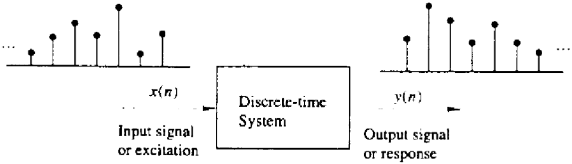
Product of two signals is defined on a sample by sample basis

$$y(n) = x_1(n)x_2(n), \quad -\infty < n < \infty$$

Outline

Signal Processing

- A system is a device or an algorithm that performs some prescribed operation on a signal
- The system processes the system input signal (excitation) $x(n)$ to obtain the processed output signal (response) $y(n)$



- This relation is described as

$$y(n) \equiv \mathcal{T}\left[x(n)\right]$$

- The magic trick is, to treat the system as a black box, but to still find a mathematical rule which explicitly defines the relation between the input and the output

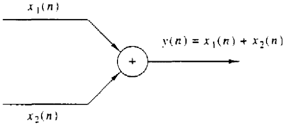
$$x(n) \xrightarrow{\mathcal{T}} y(n)$$

Exercise: determine the output of the following systems given the input signal

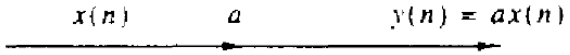
$$x(n) = \begin{cases} |n|, & -3 \leq n \leq 3 \\ 0, & \text{else} \end{cases}$$

1. $y(n) = x(n)$ (identity)
2. $y(n) = x(n-1)$ (unit delay system)
3. $y(n) = \frac{1}{3}\left[x(n+1) + x(n) + x(n-1)\right]$ (moving average filter)
4. $y(n) = \sum_{k=-\infty}^{n} x(k) = x(n) + x(n-1) + x(n-2) + \ldots$ (accumulator)

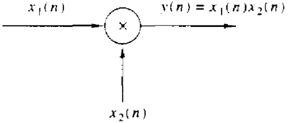
Adder No storing is necessary. The addition is *memoryless*



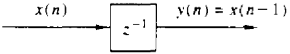
Constant Multiplier (memoryless)



Signal Multiplier (memoryless)



unit delay requires a memory. The notation z^{-1} becomes clear when discussing the z-Transform



unit advance Moves the input ahead by one sample $\rightarrow$ impossible to realize in real-time

Exercise

- Sketch the block diagram of the system

$$y(n) = \frac{1}{4}y(n-1) + \frac{1}{2}x(n) + \frac{1}{2}x(n-1)$$

- Sketch the block diagram of the system

$$y(n) = \frac{1}{4}y(n-1) + \frac{1}{2}x(n) + \frac{1}{2}x(n-1)$$

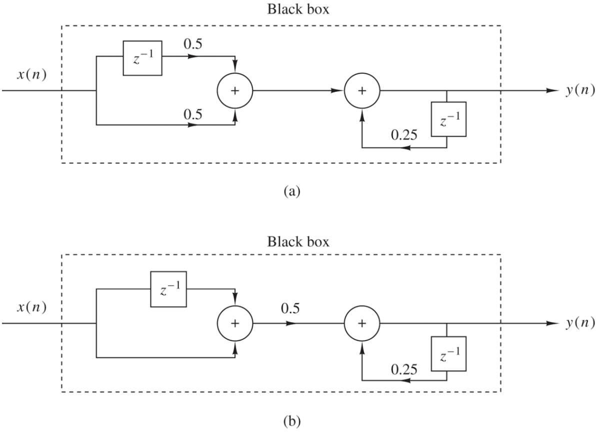


Figure 2.2.7 Block diagram realizations of the system $y(n) = 0.25y(n-1) + 0.5x(n) + 0.5x(n-1)$.

Classification of Discrete-Time Systems

Static versus dynamic systems

Static: The system is memoryless and depends only on the current sample (but not past or future samples). Can be described by

$$y(n) = \mathcal{T}\left[x(n), n\right]$$

Example:

$$y(n) = nx(n) + bx^3(n)$$

Dynamic: A system with memory. Example:

$$y(n) = x(n) + 2x(n-1)$$

Relaxed: A dynamic system is relaxed, if it received no prior excitation, i.e. the output signal depends only on the input signal

Time-variant versus time-invariant systems

Time Invariant System

A relaxed system $\mathcal{T}$ is **time invariant** or **shift invariant** iff

$$x(n) \xrightarrow{\mathcal{T}} y(n)$$

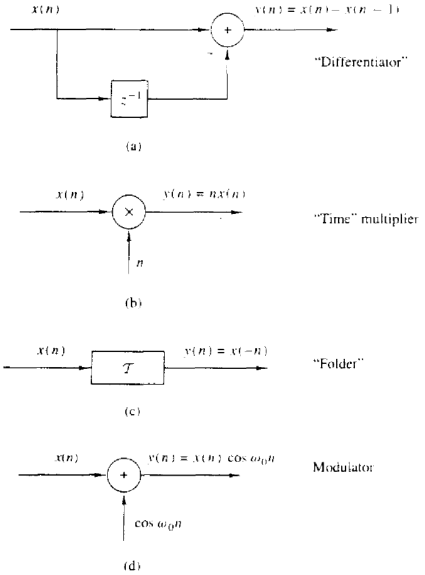
implies that

$$x(n-k) \xrightarrow{\mathcal{T}} y(n-k)$$

for every input signal $x(n)$ and every time shift k.

To check, we excite a signal with input sequence $x(n)$ which produces $y(n)$. Next, we delay the input signal by k samples $x_2(n) = x(n-k)$ and compute the output $y_2(n)$. If $y_2(n) = y(n-k)$, the system is shift invariant.

Determine if the following systems are time invariant



Linear versus nonlinear systems

Linear System

A system is linear iff

$$\mathcal{T}\left[a_1 x_1(n) + a_2 x_2(n)\right] = a_1 \mathcal{T}\left[x_1(n)\right] + a_2 \mathcal{T}\left[x_2(n)\right]$$

Example:

1. $y(n) = nx(n)$
2. $y(n) = x(n^2)$
3. $y(n) = x^2(n)$

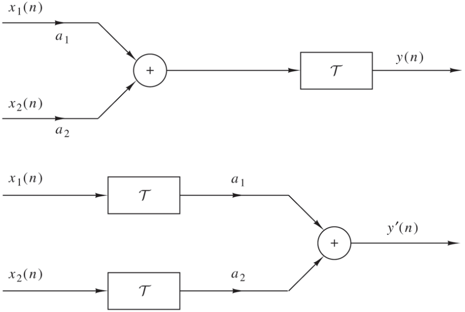


Figure 2.2.9 Graphical representation of the superposition principle. $\mathcal{T}$ is linear if and only if $y(n) = y'(n)$.

Causal versus noncausal systems

Causal System

A system is **causal** if the output of the system at any time n does not depend on future samples $x(n+k)$ with $k > 0$.

- Noncausal Systems cannot be physically realized (we cannot look into the future)
- Noncausal processing is possible on pre-recorded data.

Example:

1. $y(n) = x(n) - x(n-1)$
2. $y(n) = \sum_{k=-\infty}^{n} x(k)$
3. $y(n) = x(n+2)$
4. $y(n) = x(2n)$
5. $y(n) = x(-n)$

Stable versus unstable systems

Bounded-input bounded output (BIBO) stability

An arbitrary relaxed system is said to be BIBO stable, iff every bounded input $|x(n)| < \infty$ results in a bounded output $|y(n)| < \infty$ for all n.
If for any n the output is infinite, the system is classified unstable

Example:

- System: $y(n) = y^2(n-1) + x(n)$
- Input: $x(n) = C\delta(n)$

Outline

- While previously we classified systems w.r.t. linearity, causality, stability, and time-invariance, now we particular consider the important class of systems that are at the same time linear and time-invariant, so-called LTI-systems
- We will learn that
  - LTI-systems are fully characterized by their response to a unit sample sequence $\delta(n)$ (for continuous systems: the response to an impulse $\delta(t)$)
  - an arbitrary signal can be represented as a weighted sum of unit sample sequences
  - ➡ the response of a system to any arbitrary input signal can be expressed in terms of the unit sample response (impulse response) of the system

Resolve Discrete-Time Signal into Impulses

- ➡ An arbitrary signal $x(n)$ can be represented as a weighted sum of unit sample sequences

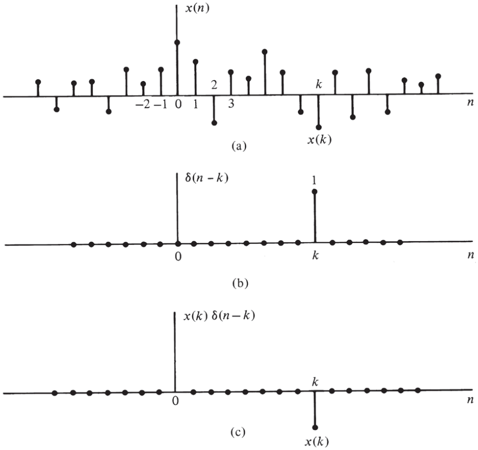


Figure 2.3.1 Multiplication of a signal $x(n)$ with a shifted unit sample sequence.

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

Example 2.3.1

- Consider the special case of a finite-duration sequence given as

$$x(n) = \left\{ 2, \underset{\uparrow}{4}, 0, 3 \right\}$$

Resolve the sequence $x(n)$ into a sum of weighted impulse sequences

Resolve Discrete-Time Signal into Impulses

Example 2.3.1

- Consider the special case of a finite-duration sequence given as

$$x(n) = \left\{ 2, \underset{\uparrow}{4}, 0, 3 \right\}$$

Resolve the sequence $x(n)$ into a sum of weighted impulse sequences

- Solution:

$$x(n) = 2\delta(n+1) + 4\delta(n) + 3\delta(n-2)$$

Response of Linear Systems to Arbitrary Inputs

- Unit sample response:

$$y(n,k) \equiv h(n,k) = \mathcal{T}\left[\delta(n-k)\right]$$

- Decompose input into sum of unit samples:

$$x(n) = \textstyle\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

- Response to arbitrary signal

$$y(n) = \mathcal{T}\left[x(n)\right] = \mathcal{T}\left[\textstyle\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right]$$

- For a relaxed **linear system**:

$$\begin{aligned} y(n) &= \mathcal{T}\left[x(n)\right] = \sum_{k=-\infty}^{\infty} x(k)\mathcal{T}\left[\delta(n-k)\right] \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n,k) \end{aligned}$$

Response of LTI Systems to Arbitrary Inputs I

The Convolution Sum

- For *time-invariant* systems we have:

$$\mathcal{T}\left[\delta(n-k)\right] = h(n,k) = h(n-k)$$

Response of an LTI-System to Arbitrary Inputs

$$y(n) = \mathcal{T}\left[x(n)\right] = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.3.17}$$

- System response: $y(n) = \mathcal{T}\left[x(n)\right]$
- Unit sequence response (impulse response): $h(n) \equiv y(n) = \mathcal{T}\left[\delta(n)\right]$

- A relaxed LTI system is completely characterized by a single function $h(n)$, which is the response to the unit sample sequence $h(n)$
- The input-output relation (2.3.17) is called the **convolution sum**

The Convolution Sum

- The convolution sum $y(n_0) = \sum_{k=-\infty}^{\infty} x(k)h(n_0 - k)$ at time n_0 is obtained by
  1. *Folding.* Fold $h(k)$ about $k = 0$ to obtain $h - k)$
  2. *Shifting.* Shift $h(-k)$ by n_0 (to the right if n_0 is positive) to obtain $h(n_0 - k)$
  3. *Multiplication.* Multiply $x(k)$ by $h(n_0 - k)$ to obtain $v_{n_0} \equiv x(k)h(n_0 - k)$
  4. *Summation.* Sum all the values of the product sequence v_{n_0} to obtain $y(n_0)$

Example 2.3.2

- The impulse response of an LTI system is

$$h(n) = \{1, \underset{\uparrow}{2}, 1, -1\}$$

Determine the response of the system to the input signal

$$x(n) = \{\underset{\uparrow}{1}, 2, 3, 1\}$$

Example 2.3.2

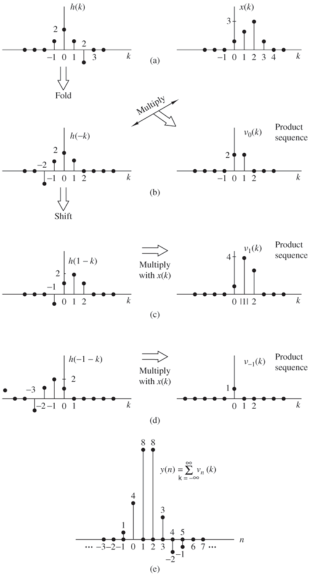


Figure 2.3.2 Graphical computation of convolution.

Properties of Convolution I

Notation The convolution is denoted by an asterisk "$*$"

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

Identity and shifting The unit sample sequence $\delta(n)$ is the identity element of convolution, i.e.

- $x(n) * \delta(n) = x(n)$,
- $x(n) * \delta(n-k) = x(n-k)$

Commutative With a change of variable $k \leftarrow n-k$, we see that

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

Associative

$$[x(n) * h_1(n)] * h_2(n)) = x(n) * [h_1(n) * h_2(n))]$$

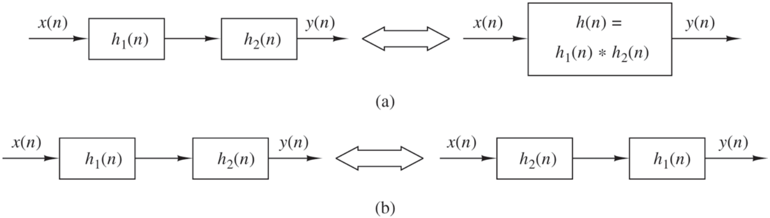


Figure 2.3.5 Implications of the associative (a) and the associative and commutative (b) properties of convolution.

Distributive

$$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)$$

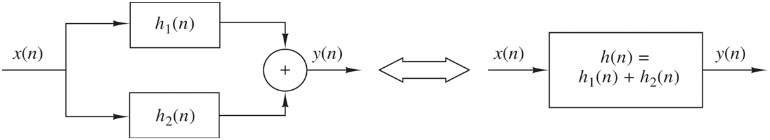


Figure 2.3.6 Interpretation of the distributive property of convolution: two LTI systems connected in parallel can be replaced by a single system with $h(n) = h_1(n) + h_2(n)$.

➡ Conversely, also means that: any LTI system can be decomposed into a parallel interconnection of subsystems

Causal Linear Time-Invariant Systems

- For a causal system, the output at time n_0 is independent of future observations $n > n_0$
- From the convolutional sum,

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

we follow that for an LTI-system, this means that

$$h(n) = 0, \quad n < 0$$

An LTI system is causal, iff its impulse response is zero for negative values of n

- if the input sequence is also causal, i.e. $x(n) = 0, \text{for } n < 0$, then

$$y(n) = \sum_{k=0}^{n} x(k)h(n-k) = \sum_{k=0}^{n} h(k)x(n-k)$$

Conditions for BIBO stability of LTI Systems

- For LTI-systems we have

$$
\begin{aligned}
y(n) &= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \\
|y(n)| &= \left| \sum_{k=-\infty}^{\infty} x(k)h(n-k) \right| \\
|y(n)| &\leq \sum_{k=-\infty}^{\infty} \underbrace{|x(k)|}_{\text{bounded}} \, |h(n-k)|
\end{aligned}
$$

- Consequently, an LTI-system is BIBO stable, if $\sum_{k=-\infty}^{\infty} |h(n-k)| < \infty$

An LTI-system is stable if its impulse response is absolutely summable

FIR and IIR Systems

It will be convenient to distinguish between systems with Finite Impulse Responses (**FIR**) and Infinite Impulse Responses (**IIR**).

- For causal FIR Systems, we have $h(n) = 0$, for $n < 0$ and $n \geq M$

$$y(n) = \sum_{k=0}^{M-1} x(k)h(n-k)$$

  - Here the systems acts as a windows which only views the most recent M samples.
  - finite memory of M samples

- For causal IIR Systems, we have

$$y(n) = \sum_{k=0}^{\infty} x(k)h(n-k)$$

Outline

- For an FIR filter it is clear that it can be realized by means of a finite amount of memory, additions and multiplications

$$y(n) = \sum_{k=0}^{M-1} x(k)h(n-k)$$

- But is it even possible to realize an IIR filter in practice?

$$y(n) = \sum_{k=0}^{\infty} x(k)h(n-k)$$

Systems Described by Difference Equations

- For an FIR filter it is clear that it can be realized by means of a finite amount of memory, additions and multiplications

$$y(n) = \sum_{k=0}^{M-1} x(k)h(n-k)$$

- But is it even possible to realize an IIR filter in practice?

$$y(n) = \sum_{k=0}^{\infty} x(k)h(n-k)$$

  - Certainly not using a convolution
  - But are there other means?

Example

- Take the cumulative average

$$y(n) = \frac{1}{n+1} \sum_{k=0}^{n} x(k), \quad n = 0, 1, \ldots$$

➡ requires storage of all input samples

- Instead, we can compute $y(n)$ more efficiently in a recursive manner
- with a simple rearrangement we obtain

$$\begin{aligned}(n+1)y(n) &= \sum_{k=0}^{n-1} x(k) + x(n) \\ &= ny(n-1) + x(n) \\ y(n) &= \frac{n}{n+1} y(n-1) + \frac{1}{n+1} x(n)\end{aligned}$$

➡ in a recursive implementation, only one memory element is needed

- Infinite impulse responses result by adding a recursive component, e.g.

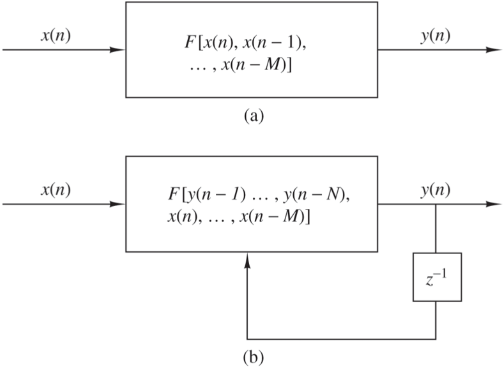


Figure 2.4.3 Basic form for a causal and realizable (a) nonrecursive and (b) recursive system.

Systems Characterized by Difference Equations

- Systems containing recursive and non-recursive parts are described by **difference equations** with **finite** N, M

$$y(n) = \underbrace{-\sum_{k=1}^{N} a_k y(n-k)}_{\text{recursive part}} + \underbrace{\sum_{k=0}^{M} b_k x(n-k)}_{\text{nonrecursive part}}, \qquad (2.5.6)$$

- N is called the **order** of the difference equation / system
- Solving the difference equations (determining the output for a given input) is similar to solving differential equations (see Sec. 2.4.3 in Proakis' Book)
  - ➡ As we will see, solving the difference equations in a spectral domain (the z-domain) is much simpler!

➡ The impulse response of such a system is exponential, i.e. **IIR**

$$h(n) = \sum_{k=1}^{N} C_k \lambda_k^n$$

➡ The impulse response of such a system is exponential, i.e. **IIR**

$$h(n) = \sum_{k=1}^{N} C_k \lambda_k^n$$

- To be BIBO stable, it must hold that $\sum_{n=0}^{\infty} |h(n)| < \infty$
  - if $|\lambda_k| < 1$ for all k then $\sum_{n=0}^{\infty} |\lambda_k|^n < \infty$ and the system is stable
  - if one or more $|\lambda_k|^n \geq 1$, the system is not BIBO stable

Outline

Direct Form I and Direct Form II

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$$

Direct Form I and Direct Form II

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$$

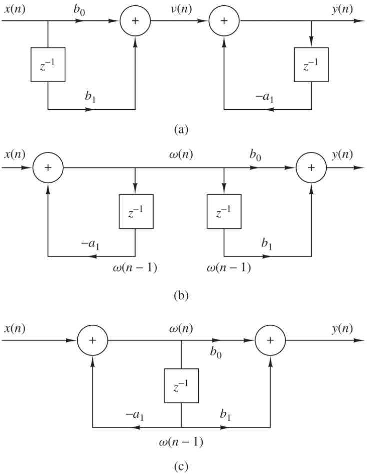


Figure 2.5.1 Steps in converting from the direct form I realization in (a) to the direct form II realization in (c).

Direct Form I for Difference Equation (2.5.6)

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

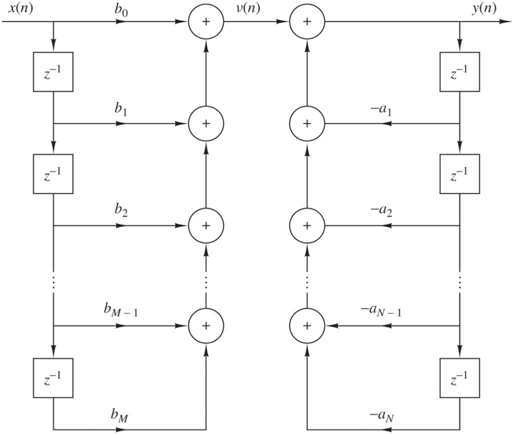


Figure 2.5.2 Direct form I structure of the system described by (2.5.6).

Direct Form II (canonic form) for Difference Equation (2.5.6)

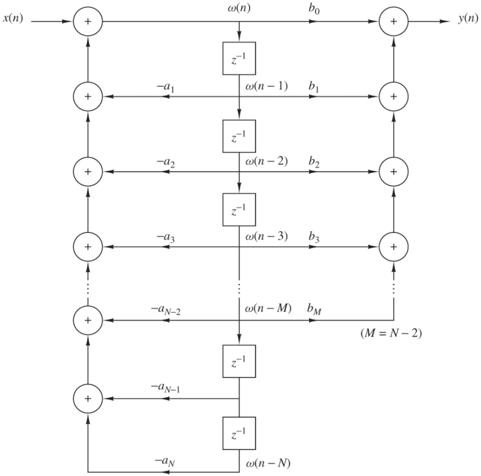


Figure 2.5.3 Direct form II structure for the system described by (2.5.6).

Examples of second order systems

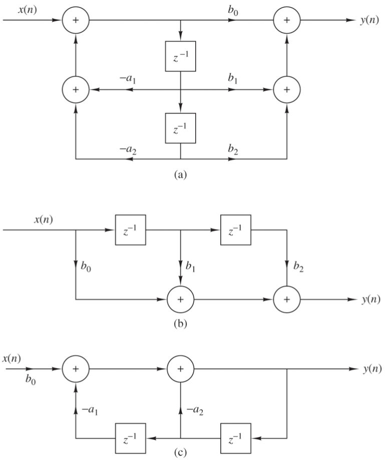


Figure 2.5.4 Structures for the realization of second-order systems: (a) general second-order system; (b) FIR system; (c) "purely recursive system."

Systems Described By Difference Equations

Special cases

- Recall that the difference equation is given by

$$y(n) = \underbrace{-\sum_{k=1}^{N} a_k y(n-k)}_{\text{recursive part}} + \underbrace{\sum_{k=0}^{M} b_k x(n-k)}_{\text{nonrecursive part}}, \tag{2.5.6}$$

  - If $a_k = 0$ for all k, we only have a non-recursive filter with finite impulse response (FIR) $h(k) = b_k$. Such as system is also referred to as **Moving Average (MA)** system
  - For $M = 0$ we have a purely recursive filter with infinite impulse response (IIR). Such as system is also referred to as **Auto Regressive (AR)** system

Outline

Correlation of Discrete-Time Signals

- Correlation measures the *similarity* between two signals
  - e.g. to find objects in images (e.g. faces)
- Correlation is mathematically similar to convolution, the difference is that for correlation, the signals are not folded

Convolution

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

Correlation

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) = x(l) * y(-l)$$

- l is called the signal **lag**
- If a convolution module is available, it can also be used to compute correlation (and vice versa)

- While the cross correlation $r_{xy}(l)$ measures the similarity between signals $x(n)$ and $y(n)$, the *autocorrelation* $r_{xx}(l)$ measures the self-similarity of a signal

  Autocorrelation

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) = x(l) * x(-l)$$

- $r_{xx}(0)$ is the energy of signal $x(n)$
- For signals periodic in N we have

$$r_{xx}(0) = r_{xx}(N) = \max_{l} r_{xx}(l)$$

  - Can be used to find the fundamental frequency in speech or music signals

Example of Autocorrelation

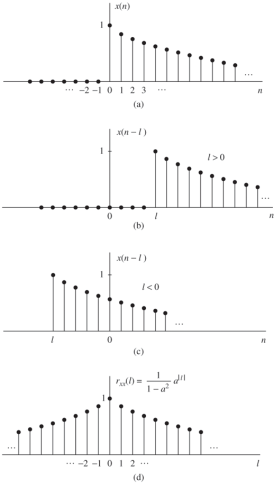
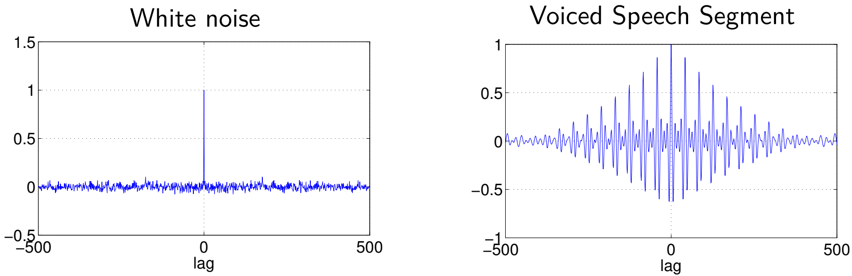


Figure 2.6.2 Computation of the autocorrelation of the signal $x(n) = a^n$, $0 < a < 1$.

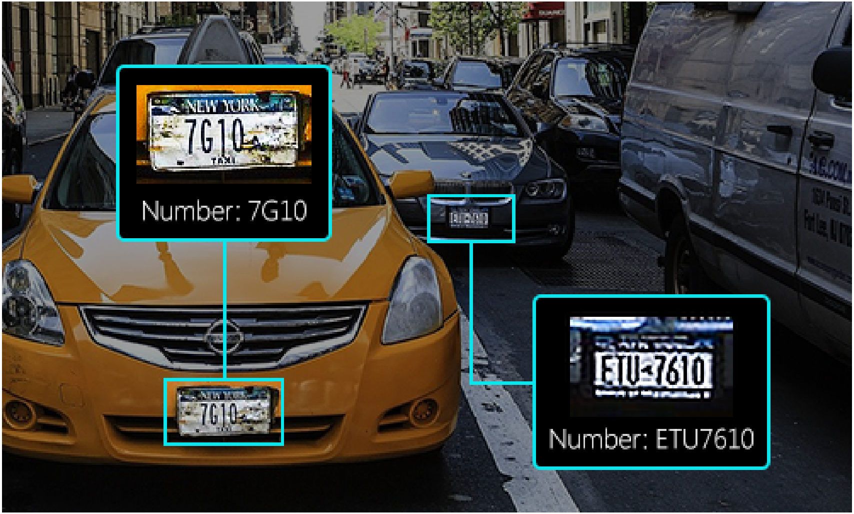
Examples of Autocorrelation of Signal Segments



White noise $r_{xx}(l) = E_x\delta(n)$: Successive samples are uncorrelated

Voiced speech The peak next to the lag $l = 0$ of the autocorrelation function corresponds to the fundamental period N.

Crosscorrelation for Object Detection



Source: http://rnd.azoft.com/convolutional-neural-networks-object-detection/

Content