Discrete-Time Systems - I
Time-Domain Representation

CHAPTER 4

Introduction

- A discrete-time system processes a given input sequence \( x[n] \) to generate an output sequence \( y[n] \) with more desirable properties.

- In most applications, the discrete-time system is a single-input, single-output system.

Examples

- A more complex example of a one-input, one-output discrete-time system is shown below.

Examples

- Accumulator:

\[
\begin{align*}
y[n] &= \sum_{\ell=-\infty}^{n} x[\ell] \\
&= \sum_{\ell=1}^{n} x[\ell] + x[n] \\
&= y[n-1] + x[n]
\end{align*}
\]

- The output \( y[n] \) at time instant \( n \) is the sum of the input sample \( x[n] \) at time instant \( n \) and the previous output \( y[n-1] \) at time instant \( n-1 \), which is the sum of all previous input sample values from \( -\infty \) to \( n-1 \).

- The system cumulatively adds, i.e., it accumulates all input sample values.
Input-output relation can also be written in the form
\[ y[n] = \sum_{k=-\infty}^{\infty} x[k] + \sum_{k=0}^{n} x[k] \]
\[ = y[-1] + \sum_{k=0}^{n} x[k], \quad n \geq 0 \]

The second form is used for a causal input sequence, in which case \( y[-1] \) is called the initial condition.

A more efficient implementation is developed next
\[ y[n] = \frac{1}{M} \left( \sum_{k=0}^{M-1} x[n-k] + x[n] - x[n-M] \right) \]
\[ = \frac{1}{M} \left( \sum_{k=0}^{M-1} x[n-1-k] + x[n] - x[n-M] \right) \]
\[ = \frac{1}{M} \left( \sum_{k=0}^{M-1} x[n-1-k] \right) + \frac{1}{M} (x[n] - x[n-M]) \]
Hence
\[ y[n] = y[n-1] + \frac{1}{M} (x[n] - x[n-M]) \]

Example: \( s[n] = 2(0.9)^n \) and \( d[n] \) is a random signal

Exponentially Weighted Running Average Filter:
\[ y[n] = \alpha y[n-1] + x[n], \quad 0 < \alpha < 1 \]

Computation of the running average requires only 1 addition, 1 multiplication and storage of the previous running average. Does not require storage of past input data samples.

For \( 0 < \alpha < 1 \), the exponentially weighted average filter places more emphasis on current data samples and less emphasis on past data samples as illustrated below.

\[ y[n] = \alpha y[n-1] + x[n] \]
\[ = \alpha^2 y[n-2] + \alpha x[n-1] + x[n] \]
\[ = \alpha^3 y[n-3] + \alpha^2 x[n-2] + \alpha x[n-1] + x[n] \]
\[ = \alpha^3 y[n-3] + \alpha^2 x[n-2] + \alpha x[n-1] + x[n] \]
Linear interpolation: Employed to estimate sample values between pairs of adjacent sample values of a discrete-time sequence.

Factor-of-4 interpolation

\[ y[n] = x_u[n] + \frac{1}{2}(x_u[n-1] + x_u[n+1]) \]

Factor-of-3 linear interpolator

\[ y[n] = x_u[n] + \frac{1}{3}(x_u[n-1] + x_u[n+2]) + \frac{2}{3}(x_u[n-2] + x_u[n+1]) \]

Factor-of-2 linear 2-D interpolator

Median Filter: The median of a set of \((2K+1)\) numbers is the number such that \(K\) numbers from the set have values greater than this number and the other \(K\) numbers have values smaller.

Median can be determined by rank-ordering the numbers in the set by their values and choosing the number at the middle.

Example: Consider the set of numbers \(\{2, -3, 10, 5, 1\}\).

Rank-ordered set is given by \(\{-3, -1, 2, 5, 10\}\).

Hence, \(\text{med}\{2, -3, 10, 5, 1\} = 2\).

Example:

- Original data
- Impulse noise corrupted data
- Median filtered noisy data
Classification

- Linear System
- Shift-Invariant System
- Causal System
- Stable System
- Passive and Lossless Systems

Linear System

- Definition: If \( y_1[n] \) is the output due to one input \( x_1[n] \) and \( y_2[n] \) is the output due to another input \( x_2[n] \) then for an input
  \[
  x[n] = \alpha x_1[n] + \beta x_2[n]
  \]
  the output is given by
  \[
  y[n] = \alpha y_1[n] + \beta y_2[n]
  \]
- Above property must hold for any arbitrary constants \( \alpha \) and \( \beta \) and for all possible inputs \( x_1[n] \) and \( x_2[n] \)

Example: Consider two accumulators with

\[
\begin{align*}
y_1[n] &= \sum_{\ell=-\infty}^{n} x_1[\ell] \\
y_2[n] &= \sum_{\ell=-\infty}^{n} x_2[\ell]
\end{align*}
\]

For an input

\[
x[n] = \alpha x_1[n] + \beta x_2[n]
\]

the output is

\[
y[n] = \sum_{\ell=-\infty}^{n} (\alpha x_1[\ell] + \beta x_2[\ell])
\]

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

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

Hence, the above system is linear.

Shift-Invariant System

- For a shift-invariant system, if \( y_1[n] \) is the response to an input \( x_1[n] \), then the response to an input
  \[
  x[n] = x_1[n - n_0]
  \]
  is simply
  \[
  y[n] = y_1[n - n_0]
  \]
  where \( n_0 \) is any positive or negative integer.
- The above relation must hold for any arbitrary input and its corresponding output.

Example: The median filter described earlier is a non-linear discrete-time system.

To show this, consider a median filter with a window of length 3.

Output \( y_1[n] \) of the filter for an input \( x_1[n] \),

\[
(x_1[n]) = (3, 4, 5), \quad 0 \leq n \leq 2
\]

is

\[
(y_1[n]) = (3, 4, 4), \quad 0 \leq n \leq 2
\]

Output \( y_2[n] \) of the filter for another input \( x_2[n] \),

\[
(x_2[n]) = (2, -1, -1), \quad 0 \leq n \leq 2
\]

is

\[
(y_2[n]) = (0, -1, -1), \quad 0 \leq n \leq 2
\]
In the case of sequences and systems with indices \( n \) related to discrete instants of time, the above property is called the **time-invariance property**.

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

**Example:**
- Consider the upsampler
  \[
  x[n] \uparrow L \rightarrow x_u[n]
  \]
  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 \( x_1[n] = x[n-n_0] \) the output \( x_{1,u}[n] \) is given by
    \[
    x_{1,u}[n] = \begin{cases} 
    x_1[n/L], & n = 0, \pm L, \pm 2L, \ldots \\
    0, & \text{otherwise}
    \end{cases}
    = \begin{cases} 
    x[n/L-n_0], & n = 0, \pm L, \pm 2L, \ldots \\
    0, & \text{otherwise}
    \end{cases}
    \]

However from the definition of the upsampler
\[
    x_u[n-n_0] = \begin{cases} 
    x_1[(n-n_0)/L], & n = 0, \pm L, \pm 2L, \ldots \\
    0, & \text{otherwise}
    \end{cases}
    \neq x_1[u[n]
\]

- Hence, the upsampler is a **time-varying system**.

**Linear Time-Invariant (LTI) System**

- **Linear Time-Invariant (LTI) system** is a system satisfying both the linearity and the time-invariance property.
- LTI systems are mathematically easy to analyze and characterize, and consequently, easy to design.
- Highly useful signal processing algorithms have been developed utilizing this class of systems over the last several decades.

**Causal System**

- In a **causal system**, the \( n \)-th output sample \( y[n] \) 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
  \[
  x_1[n] = x_2[n], \quad \text{for } n < N
  \]
  implies also that
  \[
  y_1[n] = y_2[n], \quad \text{for } n < N
  \]
  - For a causal system, changes in output samples do not precede changes in the input samples.

**Examples:**

- **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:**
  \[
  y[n] = x_u[n] + \frac{1}{2} (x_u[n-1] + x_u[n+1]) \\
  y[n] = x_u[n] + \frac{1}{3} (x_u[n-1] + x_u[n+2]) + \frac{2}{3} (x_u[n-2] + x_u[n+1])
  \]
A noncausal system can be implemented as a causal system by delaying the output by an appropriate number of samples.

For example, a causal implementation of the factor-of-2 interpolator is given by:

\[ y[n] = x_u[n-1] + \frac{1}{2}(x_u[n-2] + x_u[n]) \]

There are various definitions of stability.

We consider here the bounded-input, bounded-output (BIBO) stability, i.e.,

If \( y[n] \) is the response to an input \( x[n] \) and if \( |x| \leq B_x \) for all values of \( n \)

then \( |y| \leq B_y \) for all values of \( n \)

where \( B_x < \infty \) and \( B_y < \infty \)

Example: The \( M \)-point moving average filter is BIBO stable.

For a bounded input we have:

\[ |y[n]| = \left| \frac{1}{M} \sum_{k=0}^{M-1} x[n-k] \right| \]

\[ \leq \frac{1}{M} \sum_{k=0}^{M-1} |x[n-k]| \]

\[ \leq \frac{1}{M} (M B_x) \]

\[ \leq B_x \]

Example: Consider the discrete-time system defined by \( y[n] = \alpha x[n-N] \) with \( N \) a positive integer.

Its output energy is given by:

\[ \sum_{n=-\infty}^{\infty} |y[n]|^2 = |\alpha|^2 \sum_{n=-\infty}^{\infty} |x[n]|^2 \]

Hence, it is a passive system if \( |\alpha| \leq 1 \) and a lossless system if \( |\alpha| = 1 \)

There are various definitions of stability.

We consider here the bounded-input, bounded-output (BIBO) stability, i.e.,

If \( y[n] \) is the response to an input \( x[n] \) and if \( |x| \leq B_x \) for all values of \( n \)

then \( |y| \leq B_y \) for all values of \( n \)

where \( B_x < \infty \) and \( B_y < \infty \)

Example: The \( M \)-point moving average filter is BIBO stable.

For a bounded input we have:

\[ |y[n]| = \left| \frac{1}{M} \sum_{k=0}^{M-1} x[n-k] \right| \]

\[ \leq \frac{1}{M} \sum_{k=0}^{M-1} |x[n-k]| \]

\[ \leq \frac{1}{M} (M B_x) \]

\[ \leq B_x \]

Example: Consider the discrete-time system defined by \( y[n] = \alpha x[n-N] \) with \( N \) a positive integer.

Its output energy is given by:

\[ \sum_{n=-\infty}^{\infty} |y[n]|^2 = |\alpha|^2 \sum_{n=-\infty}^{\infty} |x[n]|^2 \]

Hence, it is a passive system if \( |\alpha| \leq 1 \) and a lossless system if \( |\alpha| = 1 \)

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, 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 \( \{s[n]\} \)
Thus, knowing the impulse response one can compute the output of the system for any arbitrary input.

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

resulting in

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

Likewise, as the system is linear, the output sequence can be expressed as a linear combination of delayed and advanced unit sample sequences in the form

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

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

which can be alternately written as

\[ y[n] = \sum_{k=-\infty}^{\infty} x[n-k] h[k] \]
The samples of $x[n]$ and $h[n]$ are then multiplied using the conventional multiplication method without any carry operation as shown on the next slide.

**Convolution Sum Properties**

- **Commutative property:**
  $$x[n] \ast h[n] = h[n] \ast x[n]$$

- **Associative property:**
  $$(x[n] \ast h[n]) \ast y[n] = x[n] \ast (h[n] \ast y[n])$$

- **Distributive property:**
  $$x[n] \ast (h[n] + y[n]) = x[n] \ast h[n] + x[n] \ast y[n]$$

*Example:*

Consider an LTI discrete-time system with an impulse response $h[n]$ generating an output $y[n]$ for an input $x[n]$:

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

Let us determine the output $y_1[n]$ of an LTI discrete-time system with an impulse response $h[n-N_0]$ for the same input $x[n]$.

Now

$$y_1[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k-N_0] = x[n] \ast h[n-N_0]$$

Hence,

$$y_1[n] = y[n-N_0]$$
The method can also be applied to convolve any two finite-length two-sided sequences as explained below.

Consider two sequences \( x_1[n] \) with \( N_1 \leq n \leq N_2 \) and \( x_2[n] \) with \( N_3 \leq n \leq N_4 \) and we are asked to compute the convolution \( y_1[n] = x_1[n] \ast x_2[n] \) of these two sequences of size \( N_2 - N_1 + N_3 - N_4 + 1 \).

1. Create two causal sequences \( y[n] = x_1[n + N_1] \) with \( 0 \leq n \leq N_2 - N_1 + N_3 - N_4 \) and \( h[n] = x_2[n + N_3] \) with \( 0 \leq n \leq N_3 - N_4 \), where both have their first elements at \( n = 0 \).
2. Compute the convolution \( y[n] = y[n] \ast h[n] \) using the table method explained on the previous slide. Here \( [y[n]] \) is defined for \( 0 \leq n \leq N_2 - N_1 + N_3 - N_4 \).
3. Then, obtain the real convolution \( y_1[n] \) as

\[
\begin{align*}
\{y_1[n]\} &= y[n - N_1 - N_3] \\
&\text{for } n \geq N_1 + N_3 - 1.
\end{align*}
\]

Convolution Using MATLAB

The MATLAB `conv` command implements the convolution sum of two finite-length sequences.

If

\[
a = [-2 \ 0 \ 1 \ -1] \\
\]

then `conv(a, b)` yields \([-2 \ -4 \ 1 \ 3 \ 1 \ 5 \ 1 \ -3]\).

Stability Condition of an LTI Discrete-Time System

- **BIBO Stability Condition**: A discrete-time system is BIBO stable if and only if the output sequence \( [y[n]] \) remains bounded for all bounded input sequence \( [x[n]] \).
- An LTI discrete-time system is BIBO stable if and only if its impulse response sequence \( [h[n]] \) is absolutely summable, i.e.

\[
S = \sum_{n=-\infty}^{\infty} |h[n]| < \infty
\]

- Proof can be found in the textbook.

Causality Condition of an LTI Discrete-Time System

- Let \( x_1[n] \) and \( x_2[n] \) be two input sequences with

\[
x_1[n] = x_2[n] \quad \text{for } n \leq n_0 \\
x_1[n] \neq x_2[n] \quad \text{for } n > n_0
\]

then the system is causal if the corresponding outputs \( y_1[n] \) and \( y_2[n] \) are also given by

\[
y_1[n] = y_2[n] \quad \text{for } n \leq n_0 \\
y_1[n] \neq y_2[n] \quad \text{for } n > n_0
\]

- An LTI discrete-time system is causal if and only if its impulse response \( [h[n]] \) is a causal sequence.
- Proof can be found in the textbook.
Examples:
- The factor-of-2 interpolator defined by
  \[ y[n] = x[n] + \frac{1}{2}(x_{n-1} + x_{n+1}) \]
  is noncausal as it has a noncausal impulse response given by
  \[ h[n] = \{0.5, 1, 0.5\}, \quad -1 \leq n \leq 1 \]

Note: A noncausal LTI discrete-time system with a finite-length impulse response can often be realized as a causal system by inserting an appropriate amount of delay.

For example, a causal version of the factor-of-2 interpolator is obtained by delaying the input by one sample period
\[ y[n] = x_{n-1} + \frac{1}{2}(x_{n-2} + x_{n}) \]

Simple Interconnection Schemes

Two simple interconnection schemes are:
- Cascade Connection
- Parallel Connection

Cascade Connection

\[ x[n] \rightarrow h_1[n] \rightarrow h_2[n] \rightarrow y[n] \equiv x[n] \rightarrow h_1[n] \star h_2[n] \rightarrow y[n] \]

- Impulse response \( h[n] \) of the cascade of two LTI discrete-time systems with impulse responses \( h_1[n] \) and \( h_2[n] \) is given by
  \[ h[n] = h_1[n] \star h_2[n] \]

- Note: The ordering of the systems in the cascade has no effect on the overall impulse response because of the commutative property of convolution.
- A cascade connection of two stable systems is stable.
- A cascade connection of two passive (lossless) systems is also passive (lossless).

An application is in the development of an inverse system as explained below:
If the cascade connection satisfies the relation
\[ h_1[n] \ast h_2[n] = \delta[n] \]
then the LTI system \( h_1[n] \) is said to be the inverse of \( h_2[n] \) and vice-versa.

An application of the inverse system concept is in the recovery of a signal \( x[n] \) from its distorted version \( \hat{x}[n] \) appearing at the output of a transmission channel
If the impulse response of the channel is known, then \( x[n] \) can be recovered by designing an inverse system of the channel

\[ x[n] \rightarrow h_1[n] \rightarrow h_2[n] \rightarrow \hat{x}[n] \]

\[ h_1[n] \ast h_2[n] = \delta[n] \]
Example:
- Consider the discrete-time accumulator with an impulse response $\mu[n]$. Its inverse system satisfy the condition

\[ \mu[n] \ast h_{2}[n] = \delta[n] \]

- It follows from the above that $h_{2}[n] = 0$ for $n < 0$ and $h_{2}[0] = 1$

\[ \sum_{\ell=0}^{n} h_{2}[\ell] = 0 \quad \text{for } n \geq 1 \]

- Thus the impulse response of the inverse system of the discrete-time accumulator is given by

\[ h_{3}[n] = \delta[n] - \delta[n-1] \]

which is called a backward difference system.

Example:
- Consider the discrete-time system shown in the figure below.

\[ h_{1}[n] = \delta[n] + 0.5\delta[n-1] \]
\[ h_{2}[n] = 0.5\delta[n] - 0.25\delta[n-1] \]
\[ h_{3}[n] = 2\delta[n] \]
\[ h_{4}[n] = -2(0.5)^{n}\mu[n] \]

Parallel Connection

\[ x[n] \quad \begin{array}{c} h_{1}[n] \oplus \quad h_{2}[n] \oplus \quad h_{3}[n] \oplus \quad h_{4}[n] \end{array} \quad y[n] = x[n] - h_{1}[n] + h_{2}[n] - h_{3}[n] + h_{4}[n] \]

Impulse response $h[n]$ of the parallel connection of two LTI discrete-time systems with impulse responses $h_{1}[n]$ and $h_{2}[n]$ is given by

\[ h[n] = h_{1}[n] + h_{2}[n] \]

Simplifying the block-diagram we obtain

Overall impulse response $h[n]$ is given by

\[ h[n] = h_{1}[n] + h_{2}[n] \ast (h_{3}[n] + h_{4}[n]) \]
\[ = h_{1}[n] + h_{2}[n] \ast h_{3}[n] + h_{2}[n] \ast h_{4}[n] \]

Now,

\[ h_{2}[n] \ast h_{3}[n] = \left( \frac{1}{2} \delta[n] - \frac{1}{4} \delta[n-1] \right) \ast 2\delta[n] \]
\[ = \delta[n] - \frac{1}{2} \delta[n-1] \]

Finite-Dimensional LTI Discrete-Time Systems

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

\[ \sum_{k=0}^{N} d_{k} y[n-k] = M \sum_{k=0}^{M} p_{k} x[n-k] \]

where $x[n]$ and $y[n]$ are, respectively, the input and the output of the system, 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.

- It is possible to implement an LTI system characterized by a constant coefficient difference equation as here the computation involves two finite sums of products.
If we assume the system to be causal, then the output $y[n]$ can be recursively computed using

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

where $d_0 \neq 0$.

$y[n]$ can be computed for all $n \geq n_0$, knowing $x[n]$ and the initial conditions $y[n_0], y[n_1], \ldots, y[n-N]$.

The convolution sum description here is

$$y[n] = \sum_{k=N_1}^{N_2} h[k] x[n-k]$$

Based on Impulse Response Length:

- If the impulse response $h[n]$ is of finite length, i.e.,
  $$h[n] = 0 \text{ for } n < N_1 \text{ and } n > N_2, N_1 < N_2$$
  then it is known as a finite impulse response (FIR) discrete-time system.
- The convolution sum description here is
  $$y[n] = \sum_{k=N_1}^{N_2} h[k] x[n-k]$$

The output $y[n]$ of an FIR LTI discrete-time system can be computed directly from the convolution sum as it is a finite sum of products.

Examples of FIR LTI discrete-time systems are the moving-average system and the linear interpolators.

The output $y[n]$ of an FIR LTI discrete-time system can be computed directly from the convolution sum as it is a finite sum of products.

Examples of FIR LTI discrete-time systems are the moving-average system and the linear interpolators.

The infinite impulse response (IIR) discrete-time systems are characterized by linear constant coefficient difference equations.

Examples:

- Example: The discrete-time accumulator defined by
  $$y[n] = y[n-1] + x[n]$$
  is seen to be an IIR system.
- Example: The familiar numerical integration formulas that are used to numerically solve integrals of the form
  $$y(t) = \int_{0}^{t} x(\tau) d\tau$$
  can be shown to be characterized by linear constant coefficient difference equations, and hence, are examples of IIR systems.

If we divide the interval of integration into $n$ equal parts of length $T$, then the previous integral can be rewritten as

$$y(nT) = y((n-1)T) + \int_{(n-1)T}^{nT} x(\tau) d\tau$$

where we have set $t = nT$ and used the notation

$$y(nT) = \int_{0}^{nT} x(\tau) d\tau$$

Using the trapezoidal method we can write

$$\int_{(n-1)T}^{nT} x(\tau) d\tau = \frac{T}{2} \left( x((n-1)T) + x(nT) \right)$$
Hence, a numerical representation of the definite integral is given by

\[ y(nT) = y((n-1)T) + \frac{T}{2} \{ x((n-1)T) + x(nT) \} \]

Let \( y[n] = y(nT) \) and \( x[n] = x(nT) \)

Then

\[ y(nT) = y((n-1)T) + \frac{T}{2} \{ x((n-1)T) + x(nT) \} \]

reduces to

\[ y[n] = y[n-1] + \frac{T}{2} \{ x[n-1] + x[n] \} \]

which is recognized as the difference equation representation of a first-order IIR discrete-time system.

**Nonrecursive and Recursive Systems**

Based on the Output Calculation Process:
- **Nonrecursive System:** Here the output can be calculated sequentially, knowing only the present and past input samples.
- **Recursive System:** Here the output computation involves past output samples in addition to the present and past input samples.

**Real and Complex Systems**

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.