Discrete-Time Systems

- 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:
  
  \[
  x[n] \xrightarrow{\text{Discrete-time System}} y[n]
  \]

Discrete-Time Systems: Examples

- **Accumulator** - Input-output relation can also be written in the form
  
  \[
  y[n] = \sum_{\ell=-\infty}^{n} x[\ell]
  \]

  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.

- **M-point moving-average system** -
  
  \[
  y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]
  \]

  Used in smoothing random variations in data.
  - In most applications, the data $x[n]$ is a bounded sequence.
  - $M$-point average $y[n]$ is also a bounded sequence.

- If there is no bias in the measurements, an improved estimate of the noisy data is obtained by simply increasing $M$.
- A direct implementation of the $M$-point moving average system requires $M-1$ additions, 1 division, and storage of $M-1$ past input data samples.
- A more efficient implementation is developed next.
Discrete-Time Systems: Examples

**Computation of the modified \( M \)-point moving average system using the recursive equation now requires 2 additions and 1 division**

An application: Consider

\[
x[n] = s[n] + d[n],
\]

where \( s[n] \) is the signal corrupted by a noise \( d[n] \).

Discrete-Time Systems: Examples

**Exponentially Weighted Running Average Filter**

Computation of the running average requires only 2 additions, 1 multiplication and storage of the previous running average.

Does not require storage of past input data samples.

Discrete-Time Systems: Examples

**Linear interpolation** - Employed to estimate sample values between pairs of adjacent sample values of a discrete-time sequence.

**Factor-of-4 interpolation**
Discrete-Time Systems: Examples

• Factor-of-2 interpolator -
  \[ y[n] = x_n[n] + \frac{1}{2}(x_n[n-1] + x_n[n+1]) \]

• Factor-of-3 interpolator -
  \[ y[n] = x_n[n] + \frac{1}{3}(x_n[n-1] + x_n[n+2]) \]
  \[ + \frac{2}{3}(x_n[n-2] + x_n[n+1]) \]

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

Median Filter –
• Example: Consider the set of numbers \(\{2, -3, 10, 5, -1\}\)
  • Rank-order set is given by \(\{-3, -1, 2, 5, 10\}\)
  • Hence, \(\text{med}\{2, -3, 10, 5, -1\} = 2\)

Median Filter –
• Implemented by sliding a window of odd length over the input sequence \(\{x[n]\}\) one sample at a time
• Output \(y[n]\) at instant \(n\) is the median value of the samples inside the window centered at \(n\)
• Finds applications in removing additive random noise, which shows up as sudden large errors in the corrupted signal
• Usually used for the smoothing of signals corrupted by impulse noise
Discrete-Time Systems: Examples

Median Filtering Example –

Linear Discrete-Time Systems

• Definition - If $y_1[n]$ is the output due to an input $x_1[n]$ and $y_2[n]$ is the output due to an 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 $x_2[n]$.

Linear Discrete-Time Systems

• The outputs $y_1[n]$ and $y_2[n]$ for inputs $x_1[n]$ and $x_2[n]$ are given by

$$y_1[n] = y_1[-1] + \sum_{\ell=0}^{n} x_1[\ell]$$

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

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

$$y[n] = y[-1] + \sum_{\ell=0}^{n} (\alpha x_1[\ell] + \beta x_2[\ell])$$

Linear Discrete-Time Systems

• Now $\alpha y_1[n] + \beta y_2[n]$

$$= \alpha(y_1[-1] + \sum_{\ell=0}^{n} x_1[\ell]) + \beta(y_2[-1] + \sum_{\ell=0}^{n} x_2[\ell])$$

$$= (\alpha y_1[-1] + \beta y_2[-1]) + (\alpha \sum_{\ell=0}^{n} x_1[\ell] + \beta \sum_{\ell=0}^{n} x_2[\ell])$$

• Thus $y[n] = \alpha y_1[n] + \beta y_2[n]$ if

$$y[-1] = \alpha y_1[-1] + \beta y_2[-1]$$

Discrete-Time Systems: Classification

• Linear System
• Shift-Invariant System
• Causal System
• Stable System
• Passive and Lossless Systems
**Linear Discrete-Time System**

- For the causal accumulator to be linear, the condition \( y[-1] = \alpha y_1[-1] + \beta y_2[-1] \) must hold for all initial conditions \( y_1[-1], y_2[-1] \), and all constants \( \alpha \) and \( \beta \).
- This condition cannot be satisfied unless the accumulator is initially at rest with zero initial condition.
- For nonzero initial condition, the system is nonlinear.

**Nonlinear Discrete-Time System**

- The median filter described earlier is a nonlinear discrete-time system.
- To show this, consider a median filter with a window of length 3.
- Output of the filter for an input \( \{x_1[n]\} = \{3, 4, 5\}, 0 \leq n \leq 2 \)
  is \( \{y_1[n]\} = \{3, 4, 4\}, 0 \leq n \leq 2 \).
- Output for an input \( \{x_2[n]\} = \{2, -1, -1\}, 0 \leq n \leq 2 \)
  is \( \{y_2[n]\} = \{0, -1, -1\}, 0 \leq n \leq 2 \).
- However, the output for an input \( \{x[n]\} = \{x_1[n] + x_2[n]\} \)
  is \( \{y[n]\} = \{3, 4, 3\} \).
- Hence, the median filter is a nonlinear discrete-time system.

**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.
- 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.
**Shift-Invariant System**

- **Example** - 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 \( x_{1u}[n] \) is given by
  \[ x_{1u}[n] = x[n/L], \quad n = 0, \pm L, \pm 2L, \ldots \\
  = x(n - Ln_o)/L, \quad n = 0, \pm L, \pm 2L, \ldots \quad \text{otherwise} \]

- However from the definition of the up-sampler
  \[ x_u[n-n_o] = \begin{cases} 
  x(n-n_o)/L, & n = n_o, n_o \pm L, n_o \pm 2L, \ldots \\
  0, & \text{otherwise} 
  \end{cases} \]

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

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

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

- **Causal System**
  - Then \( x_1[n] = x_2[n] \) for \( n < N \)
  - \( y_1[n] = y_2[n] \) for \( n < N \)
  - For a causal system, changes in output samples do not precede changes in the input samples
  - Examples of causal systems:
    \[
    y[n] = \alpha_1 x[n] + \alpha_2 x[n-1] + \alpha_3 x[n-2] + \alpha_4 x[n-3] \\
    y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_3 x[n-3] \\
    y[n] = y[n-1] + x[n] \\
    \]
  - Examples of noncausal systems:
    \[
    y[n] = x_u[n] + \frac{1}{3} (x_u[n-1] + x_u[n+1]) \\
    y[n] = x_u[n] + \frac{2}{3} (x_u[n-1] + x_u[n+1]) \\
    + \frac{2}{3} (x_u[n-2] + x_u[n+2]) \\
    \]
Causal System

- 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:

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

Stable System

- There are various definitions of stability.
- We consider here the bounded-input, bounded-output (BIBO) stability.
- If \( y[n] \) is the response 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 \( n \).

Passive and Lossless Systems

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

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

- For a lossless system, the above inequality is satisfied with an equal sign for every input.

Impulse and Step Responses

- 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 \( \{u[n]\} \) is called the unit step response or simply, the step response, and is denoted by \( \{s[n]\} \).
**Impulse Response**

- **Example** - 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
  \[ h[n] = \alpha_1 \delta[n] + \alpha_2 \delta[n-1] + \alpha_3 \delta[n-2] + \alpha_4 \delta[n-3] \]
  - The impulse response is thus a finite-length sequence of length 4 given by
    \( \{h[n]\} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} \)

---

**Time-Domain Characterization of LTI Discrete-Time System**

- **Let** \( h[n] \) **denote the impulse response of a LTI discrete-time system**
- **We compute its output** \( y[n] \) **for the 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 outputs for each member of the input separately and add the individual outputs to determine \( y[n] \)
Time-Domain Characterization of LTI Discrete-Time System

- Likewise, as the system is linear input
  
  \[ 0.5\delta[n+2] \rightarrow 0.5h[n+2] \]
  
  \[ 1.5\delta[n-1] \rightarrow 1.5h[n-1] \]
  
  \[ -\delta[n-2] \rightarrow -h[n-2] \]
  
  \[ 0.75\delta[n-5] \rightarrow 0.75h[n-5] \]
  
- Hence because of the linearity property we get
  
  \[ y[n] = 0.5h[n + 2] + 1.5h[n - 1] \]
  
  \[ -h[n - 2] + 0.75h[n - 5] \]

Time-Domain Characterization of LTI Discrete-Time System

- Now, 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 \( x[k]h[n - k] \)

Convolution Sum

- The summation
  
  \[ y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n - k] = \sum_{k=-\infty}^{\infty} x[n - k]h[n] \]
  
  is called the convolution sum of the sequences \( x[n] \) and \( h[n] \) and represented compactly as
  
  \[ y[n] = x[n] \otimes h[n] \]

Convolution Sum

- 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:
  
  \[ x[n] \otimes (h[n] + y[n]) = x[n] \otimes h[n] + x[n] \otimes y[n] \]
Convolution Sum

- Schematic Representation:
  \[ h[-k] \ast x[k] \sum_{k} y[n] \]

- The computation of an output sample using the convolution sum is simply a sum of products.
- Involves fairly simple operations such as additions, multiplications, and delays.

We illustrate the convolution operation for the following two sequences:

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

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

- Figures on the next several slides the steps involved in the computation of
  \[ y[n] = x[n] \ast h[n] \]
Time-Domain Characterization of 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.
- If both the input sequence and the impulse response sequence are of finite length, the output sequence is also of finite length.

**Example**

Develop the sequence $y[n]$ generated by the convolution of the sequences $x[n]$ and $h[n]$ shown below.
Time-Domain Characterization of LTI Discrete-Time System

- As a result, for \( n < 0 \), the product of the \( k \)-th samples of \( \{x[k]\} \) and \( \{h[n-k]\} \) is always zero, and hence \( y[n] = 0 \) for \( n < 0 \)
- Consider now the computation of \( y[0] \)
- The sequence \( \{h[-k]\} \) is shown on the right

\[ h[-k] \]

\[ 0\quad 0\quad 0\quad 0\quad -1\quad -2\quad -3\quad -4 \]

\( k \)

\[ -4\quad -3\quad -2\quad -1\quad 0\quad 1\quad 2\quad 3 \]

73

Copyright © 2005, S. K. Mitra

Time-Domain Characterization of LTI Discrete-Time System

- The product sequence \( \{x[k]\} \{h[-k]\} \) is plotted below which has a single nonzero sample \( x[0]h[0] \) for \( k = 0 \)

\[ x[k]\{h[-k]\} \]

\[ 0\quad 0\quad 0\quad 0\quad -1\quad -2\quad -3\quad -4 \]

\( k \)

\[ -4\quad -3\quad -2\quad -1\quad 0\quad 1\quad 2\quad 3 \]

74

Copyright © 2005, S. K. Mitra

Time-Domain Characterization of LTI Discrete-Time System

- For the computation of \( y[1] \), we shift \( \{h[-k]\} \) to the right by one sample period to form \( \{h[1-k]\} \) as shown below on the left
- The product sequence \( \{x[k]\} \{h[1-k]\} \) is shown below on the right

\[ x[k]\{h[1-k]\} \]

\[ 0\quad 0\quad 0\quad 0\quad 1\quad 2\quad 3\quad 4 \]

\( k \)

\[ 0\quad 0\quad 0\quad 0\quad -1\quad -2\quad -3\quad -4 \]

75

Copyright © 2005, S. K. Mitra

- Hence, \( y[1] = x[0]h[1] + x[1]h[0] = -4 + 0 = -4 \)

Time-Domain Characterization of LTI Discrete-Time System

- To calculate \( y[2] \), we form \( \{h[2-k]\} \) as shown below on the left
- The product sequence \( \{x[k]\} \{h[2-k]\} \) is plotted below on the right

\[ x[k]\{h[2-k]\} \]

\[ 0\quad 0\quad 0\quad 0\quad 1\quad 2\quad 3\quad 4 \]

\( k \)

\[ 0\quad 0\quad 0\quad 0\quad -1\quad -2\quad -3\quad -4 \]

76

Copyright © 2005, S. K. Mitra


Time-Domain Characterization of LTI Discrete-Time System

- Continuing the process we get
  \[
  
  
  
  
  
77

Copyright © 2005, S. K. Mitra

- From the plot of \( \{h[n-k]\} \) for \( n > 7 \) and the plot of \( \{x[k]\} \) as shown below, it can be seen that there is no overlap between these two sequences
- As a result \( y[n] = 0 \) for \( n > 7 \)

\[ x[k] \]

\[ 0\quad 0\quad 0\quad 0\quad 1\quad 2\quad 3\quad 4 \]

\( k \)

\[ 0\quad 0\quad 0\quad 0\quad -1\quad -2\quad -3\quad -4 \]

78

Copyright © 2005, S. K. Mitra

- As a result \( y[0] = x[0]h[0] = -2 \)
Time-Domain Characterization of LTI Discrete-Time System

- The sequence \( \{y[n]\} \) generated by the convolution sum is shown below.

\[
\begin{array}{cccccccc}
\text{Unit} & 0 & 1 & 2 & 3 & 4 & 5 \\
\{y[n]\} & 70 & 36 & 30 & 14 & 6 & 0 \\
\{h[n]\} & 30 & 14 & 6 & 0 & -6 & -14 & -30 \\
\{g[n]\} & 5 & 4 & 3 & 2 & 1 & 0 & -1 \\
\end{array}
\]

- Note: The sum of indices of each sample product inside the convolution sum is equal to the index of the sample being generated by the convolution operation.

- For example, the computation of \( y[3] \) in the previous example involves the products \( x[0]h[3], x[1]h[2], x[2]h[1], \) and \( x[3]h[0] \).

- The sum of indices in each of these products is equal to 3.

Tabular Method of Convolution Sum Computation

- Can be used to convolve two finite-length sequences.

- Consider the convolution of \( \{g[n]\}, 0 \leq n \leq 3, \) with \( \{h[n]\}, 0 \leq n \leq 2, \) generating the sequence \( y[n] = g[n] \circledast h[n] \).

- Samples of \( \{g[n]\} \) and \( \{h[n]\} \) are then multiplied using the conventional multiplication method without any carry operation.

- The samples \( y[n] \) generated by the convolution sum are obtained by adding the entries in the column above each sample.

\[
\begin{array}{cccccccc}
\text{Sample} & 0 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\text{Sample} & 0 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

- The samples \( y[n] \) are given by:

\[
\begin{align*}
y[0] &= g[0]h[0] \\
y[1] &= g[1]h[1] + g[0]h[0] \\
\end{align*}
\]
**Tabular Method of Convolution Sum Computation**

- The method can also be applied to convolve two finite-length 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
- Next, convolution is computed ignoring the location of the decimal point

**Convolution Using MATLAB**

- The M-file `conv` implements the convolution sum of two finite-length sequences
- If $a = [-2 0 1 -1 3]$  
  $b = [1 2 0 -1]$  
  then `conv(a, b)` yields $[-2 -4 1 3 1 5 1 -3]$

**Simple Interconnection Schemes**

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

**Cascade Connection**

- 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 $h[n] = h_1[n] \odot h_2[n]$
- 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 passive (lossless)
Cascade Connection

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

Example - Consider the discrete-time accumulator with an impulse response \( \mu[n] \).
• Its inverse system satisfy the condition
  \[ \mu[n] \otimes h_2[n] = \delta[n] \]
• It follows from the above that \( h_2[n] = 0 \) for \( n < 0 \) and
  \[ h_2[0] = 1 \]
  \[ \sum_{\ell=0}^{n} h_2[\ell] = 0 \] for \( n \geq 1 \).

Parallel Connection

- 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
  \[ h[n] = h_1[n] + h_2[n] \]

Simple Interconnection Schemes

- Consider the discrete-time system where
  \[ h_1[n] = \delta[n] + 0.5\delta[n - 1], \]
  \[ h_2[n] = 0.5\delta[n] - 0.25\delta[n - 1], \]
  \[ h_3[n] = 2\delta[n], \]
  \[ h_4[n] = -2(0.5)^n \mu[n] \]

Cascade Connection

- An application of the inverse system concept is in the recovery of a signal \( x[n] \) from its distorted version \( \tilde{x}[n] \) 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

\[
\begin{align*}
& x[n] \rightarrow \text{channel} \\
& \quad \downarrow h_1[n] \otimes h_2[n] \equiv \text{inverse system} \\
& \quad \downarrow \tilde{x}[n] \rightarrow x[n]
\end{align*}
\]

Thus the impulse response of the inverse system of the discrete-time accumulator is given by
\[ h_2[n] = \delta[n] - \delta[n - 1] \]
which is called a backward difference system.
Simple Interconnection Schemes

• Simplifying the block-diagram we obtain

\[
\begin{align*}
    h_2[n] & \quad \oplus \quad h_3[n] + h_4[n] \\
    h_2[n] & = h_3[n] \oplus (h_3[n] + h_4[n]) \\
    h[n] & = (h_3[n] + h_4[n]) \\
\end{align*}
\]

• Overall impulse response \( h[n] \) is given by

\[
    h[n] = h_1[n] + h_2[n] \oplus (h_3[n] + h_4[n]) \\
    = h_1[n] + h_2[n] \oplus h_3[n] + h_2[n] \oplus h_4[n] \\
\]

• Now,

\[
    h_2[n] \oplus h_3[n] = (\frac{1}{2} \delta[n] - \frac{1}{4} \delta[n-1]) \oplus 2 \delta[n] \\
    = \delta[n] - \frac{1}{2} \delta[n-1]
\]

Decreasing order

\[
    h_2[n] \oplus h_3[n] = \frac{1}{2} \delta[n] - \frac{1}{4} \delta[n-1] \oplus \frac{1}{2} \delta[n] + \frac{1}{2} \delta[n] + \frac{1}{4} \delta[n+1] \\
    = \frac{1}{2} \delta[n] - \frac{1}{2} \delta[n] - \frac{1}{4} \delta[n-1] + \frac{1}{2} \delta[n-1] \\
    = \frac{1}{2} \delta[n] - \frac{1}{4} \delta[n]-\frac{1}{2} \delta[n-1] \\
    = \frac{1}{2} \delta[n] - \frac{1}{4} \delta[n-1] - \delta[n-1] = \delta[n]
\]

• Therefore

\[
    h[n] = \delta[n] + \frac{1}{2} \delta[n-1] + \frac{1}{2} \delta[n-1] - \delta[n] = \delta[n]
\]