Analog vs. discrete signals

- Continuous-time signals are also known as *analog signals* because their amplitude is “analogous” (i.e., proportional) to the physical quantity they represent.

- *Discrete-time signals* are defined only at discrete times, that is, at a discrete set of values of an independent variable.

- Frequently, discrete signals arise from sampling their continuous counterparts. The result is a sequence of numbers defined by

\[ s[n] = s(t) \bigg|_{t=nT} = s(nT), \]

where \( n \in \mathbb{Z} \) and \( T \) is a *sampling period*. The quantity \( F_s = 1/T \) is known as *sampling frequency*.

- A discrete-time signal \( s[n] \) whose amplitude takes values from a finite set of \( K \) numbers \( \{a_k\}_{k=1}^K \) is known as a *digital signal*. 
Different signal formats

(a) $s(t)$

(b) $s[n]$

(c) $s(t/T)$

(d) $s[n]$
A continuous-time system (CTS) is a system which transforms a continuous-time input signal $x(t)$ into a continuous-time output signal $y(t)$. For example:

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

Symbolically, the input-output relation of a continuous-time system can be represented by

$$\mathcal{H} : x(t) \mapsto y(t) \quad \text{or} \quad \mathcal{H}\{x(t)\} = y(t).$$

In real-life, CTS are implemented using analog electronic circuits. The physical implementation of a CTS is called an analog system.
Discrete-time systems

- A system that transforms a discrete-time input signal $x[n]$ into a discrete-time output signal $y[n]$, is called a discrete-time system.

- Symbolically, the input-output relation of a discrete-time system is represented by

  $$
  \mathcal{H}: x[n] \mapsto y[n] \quad \text{or} \quad \mathcal{H}\{x[n]\} = y[n].
  $$

- The physical implementation of discrete-time systems can be done either in software or hardware.

- In both cases, the underlying physical systems consist of digital electronic circuits designed to manipulate logical information or physical quantities represented in digital form by binary electronic signals.
Quantization converts a continuous-amplitude signal $x(t)$ into a discrete-amplitude signal $x_d[n]$.

In theory, we are dealing with discrete-time signals; in practice, we are dealing with digital signals.
The conversion of a discrete-time signal into continuous time form is done using a *digital-to-analog (D/A) converter*.

An ideal D/A converter or interpolator essentially “fills the gaps” between the samples of a sequence of numbers to create a continuous-time function.
Analog, digital, and mixed signal processing

Typical implementation:

- Analog Systems
- Interface Systems
- Digital Systems

- Continuous-time Signals and Systems
- Analytical techniques
- Analog electronics

- Discrete-time Signals and Systems
- Numerical techniques
- Digital electronics

Diagram:

- Analog input
- Sensor
- Analog pre-filter
- ADC
- Digital output from ADC
- Digital input to DAC
- DSP
- DAC
- Analog output
- Analog post-filter

Prof. O. Michailovich, Dept of ECE, Spring 2019
ECE 413: Digital Signal Processing / Section 1
<table>
<thead>
<tr>
<th><strong>Applications of DSP</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Audio processing</strong></td>
<td>compression and decompression, equalization, mixing and editing, artificial reverberation, sound synthesis, stereo and surround sound, and noise cancelation</td>
</tr>
<tr>
<td><strong>Speech processing</strong></td>
<td>speech synthesis, compression and decompression, speech recognition, speaker identification, and speech enhancement</td>
</tr>
<tr>
<td><strong>Image and video processing</strong></td>
<td>image compression and decompression, image enhancement, geometric transformations, feature extraction, video coding, motion detection, and tomographic image reconstruction</td>
</tr>
<tr>
<td><strong>Telecommunications (transmission of audio, video, and data)</strong></td>
<td>modulation and demodulation, error detection and correction coding, encryption and decryption, acoustic echo cancelation, multipath equalization, computer networks, radio and television, and cellular telephony</td>
</tr>
<tr>
<td><strong>Computer systems</strong></td>
<td>sound and video processing, disk control, printer control, modems, Internet phone, radio, and television</td>
</tr>
<tr>
<td><strong>Military systems</strong></td>
<td>guidance and navigation, beamforming, radar and sonar processing, hyperspectral image processing, and software radio</td>
</tr>
</tbody>
</table>
Discrete-time signals

- We will use the following notation:
  - $x[n]$ – to represent the $n$-th sample of the sequence;
  - $\{x[n]\}_{n=N_1}^{N_2}$ – to represent the samples in the range $N_1 \leq n \leq N_2$;
  - $\{x[n]\}$ – to represent the entire sequence.

- There is a number of different ways to represent discrete signals:
  
  **Functional**
  $$x[n] = \begin{cases} 
  \left(\frac{1}{2}\right)^n, & n \geq 0 \\
  0, & n < 0 
  \end{cases}$$

  **Tabular**
  
  \[
  \begin{array}{cccccccc}
  n & \ldots & -2 & -1 & 0 & 1 & 2 & 3 & \ldots \\
  x[n] & \ldots & 0 & 0 & 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \ldots
  \end{array}
  \]

  **Sequence**
  $$x[n] = \{ \ldots, 0, 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots \}$$

  **Pictorial**
  
  ![Graphical representation of the sequence]
Discrete-time signals (continued)

- The **energy** of a sequence $x[n]$ is defined as
  \[
  E_x := \sum_n |x[n]|^2 \quad \left[= \text{sum}(x.*\text{conj}(x)) = Ex \right].
  \]

- The **power** of a sequence $x[n]$ is defined as the average energy per sample
  \[
  P_x := \lim_{L \to \infty} \left\{ \frac{1}{2L+1} \sum_{n=-L}^{L} |x[n]|^2 \right\} \quad \left[= \frac{Ex}{\text{length}(x)} = Px \right].
  \]

- When signal $x[n]$ represents a physical signal, both quantities are directly related to the energy and power of the signal.

- Note that when the duration of $\{x[n]\}$ is finite, its $E_x$ is always finite, while $P_x$ is equal to zero (assuming that the signal values outside of its support are equal to zero).
Some elementary signals

- **Unit sample sequence (aka “discrete Dirac”)**
  \[
  \delta[n] = \begin{cases} 
  1, & n = 0 \\
  0, & n \neq 0 
  \end{cases}.
  \]

- **Unit step sequence (aka “discrete Heavyside”)**
  \[
  u[n] = \begin{cases} 
  1, & n \geq 0 \\
  0, & n < 0 
  \end{cases} = \sum_{k=-\infty}^{n} \delta[k].
  \]

- **Sinusoidal sequence**
  \[
  x[n] = A \cos(\omega_0 n + \varphi), \quad \text{with } [\omega_0] = \frac{\text{radians}}{\text{sample}}.
  \]

- **Exponential sequence**
  \[
  x[n] = A a^n, \quad A, a \in \mathbb{C}.
  \]

- **Complex exponential sequence**
  \[
  x[n] = A e^{j\omega_0 n} = A \cos(\omega_0 n) + jA \sin(\omega_0 n)
  \]
A sequence $x[n]$ is called periodic if $x[n] = x[n + N]$, $\forall n \in \mathbb{Z}$.

The smallest $N$ for which it holds is called the fundamental period of $\{x[n]\}$.

A sinusoidal sequence is periodic if

$$
\cos(\omega_0 n + \varphi) = \cos(\omega_0 (n + N) + \varphi) = \cos(\omega_0 n + \omega_0 N + \varphi),
$$

which necessitates

$$
\omega_0 N = 2\pi k, \quad k \in \mathbb{Z}.
$$

Thus, for a given $N$, there are exactly $N$ distinct sinusoidal sequences with frequencies

$$
\omega_k = \frac{2\pi k}{N}, \quad \text{with } k = 0, 1, \ldots, N - 1.
$$
Causality

\( \mathcal{H} \) is called \textit{causal} if, for any \( n_0 \), \( y[n_0] \) is determined by the values of \( x[n] \) for \( n \leq n_0 \) only.

- \( y[n] = 1/3 \left( x[n] + x[n - 1] + x[n - 2] \right) \)  \hspace{1cm} \text{(causal)}
- \( y[n] = \text{median} \left( x[n - 1], x[n], x[n + 1] \right) \)  \hspace{1cm} \text{(non-causal)}

BIBO stability

\( \mathcal{H} \) is said to be \textit{stable} in the Bounded-Input Bounded-Output (BIBO) sense, if \( |x[n]| \leq M_x < \infty \) implies \( |y[n]| \leq M_y < \infty \) for some positive constants \( M_x \) and \( M_y \).

- \( y[n] = \frac{1}{L} \sum_{k=0}^{L-1} x[n - k] \)  \hspace{1cm} \text{(stable)}
- \( y[n] = \sum_{k=-\infty}^{n} x[k] \)  \hspace{1cm} \text{(not stable)}
Discrete-time systems (continued)

**Linearity**

\( \mathcal{H} \) is called *linear* if for every \( a_1 \) and \( a_2 \) (either real or complex) and every input signals \( x_1[n] \) and \( x_2[n] \) it holds that

\[
\mathcal{H}\{a_1 x_1[n] + a_2 x_2[n]\} = a_1 \mathcal{H}\{x_1\} + a_2 \mathcal{H}\{x_2\}
\]

- \( y[n] = x[n] - x[n - 1] \) (linear)
- \( y[n] = (x[n])^2 \) (non-linear)

**Time-invariance (TI)**

\( \mathcal{H} \) is called *time-invariant* if and only if

\[
y[n] = \mathcal{H}\{x[n]\} \implies y[n - n_0] = \mathcal{H}\{x[n - n_0]\}
\]

- \( y[n] = \cos(x[n]) \) (TI)
- \( y[n] = \sum_n x[n]w[n] \), for some fixed \( \{w[n]\} \) (not TI)
Block diagrams and signal flow graphs

Block Diagram Elements

- **Adder**: $y[n] = x_1[n] + x_2[n]$  
- **Multiplier**: $y[n] = ax[n]$  
- **Unit delay**: $y[n] = x[n-1]$  
- **Splitter**: $w[n] ightarrow w[n]$  

Signal Flow Graph Elements

- **Summing node**: $y[n] = x_1[n] + x_2[n]$  
- **Gain branch**: $y[n] = ax[n]$  
- **Unit delay branch**: $y[n] = x[n-1]$  
- **Pick-off node**: $w[n] ightarrow w[n]$
In the above example, we have

\[ w[n] = x[n] + a w[n - 1] \] (input node)

\[ y[n] = w[n] + b w[n - 1] \] (output node)

After a simple manipulation to eliminate \( w[n] \), we obtain

\[ y[n] = x[n] + b x[n - 1] + a y[n - 1]. \]
The operations of shifting and folding are not commutative. Indeed:

\[ x[n] \xrightarrow{\text{shift}} x[n-n_0] \xrightarrow{\text{fold}} x[-n-n_0] \neq x[n] \xrightarrow{\text{fold}} x[-n] \xrightarrow{\text{shift}} x[-n+n_0] \]

In MATLAB, discrete sequences can be plotted using

```matlab
stem(n, x, 'fill')
xlabel('n')
ylabel('x[n]')
```
Signal decomposition into impulses

- A time-shifted version \( \delta[n - k] \) of \( \delta[n] \) is defined as
  \[
  \delta[n - k] = \begin{cases} 
  1, & n = k \\
  0, & n \neq k 
  \end{cases}
  \]

- Using this sequence, one can redefine \( \{x[n]\} \) as given by
  \[
  x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n - k]
  \]

For example:
Impulse response

The impulse response $h[n]$ of a system $y[n] = \mathcal{H}\{x[n]\}$ is defined as

$$h[n] = \mathcal{H}\{\delta[n]\}$$

Now, if $\mathcal{H}$ is linear time-invariant (LTI), then $\mathcal{H}\{\delta[n - k]\} = h[n - k]$ and, consequently,

$$\mathcal{H}\{x[n]\} = \mathcal{H}\left\{\sum_k x[k]\delta[n - k]\right\} = \sum_k x[k]\mathcal{H}\{\delta[n - k]\} = \sum_k x[k]h[n - k]$$

Discrete convolution

Thus, the response of an LTI system can be described as

$$y[n] = \mathcal{H}\{x[n]\} = \sum_{k=-\infty}^{\infty} x[k]h[n - k] = x[n] * h[n]$$
Understanding the discrete convolution

Let us consider a case with $x[n] = \{1 \ 2 \ 3 \ 4 \ 5\}$ and $h[n] = \{-1 \ 2 \ 1\}$. Our task is to compute $y[3] = \sum_k x[k]h[3 - k]$.

Above, both $\{x[n]\}$ and $\{h[n]\}$ were converted to infinite-length sequences by means of **zero-padding**.

This type of convolution is called **full**.
Suppose we want to convolve \( \{h[n]\}_{n=0}^{M-1} \) and \( \{x[n]\}_{n=0}^{N-1} \), with \( M = 3 \) and \( N = 6 \). In this case, we have:

\[
\begin{array}{l}
y[-1] = h[0]0 + h[1]0 + h[2]0 & \text{No overlap} \\
y[0] = h[0]x[0] + h[1]0 + h[2]0 & \text{Partial overlap} \\
y[8] = h[0]0 + h[1]0 + h[2]0 & \text{No overlap.}
\end{array}
\]

One can see that:
- \( y[n] = 0 \) for \( n < 0 \)
- \( y[n] = 0 \) for \( n \geq N + M - 1 \)
The operation of convolution can be expressed in the form of vector-matrix multiplication. Specifically,

\[
\begin{bmatrix}
  y[0] \\
  y[1] \\
  y[2] \\
  y[3] \\
  y[4] \\
  y[5] \\
  y[6] \\
  y[7]
\end{bmatrix} =
\begin{bmatrix}
  x[0] & 0 & 0 \\
  x[1] & x[0] & 0 \\
  0 & 0 & x[5]
\end{bmatrix}
\begin{bmatrix}
  h[0] \\
  h[1] \\
  h[2]
\end{bmatrix} = h[0] + \ldots + h[2]
\]

\[
X =
\begin{bmatrix}
  x[0] \\
  x[1] \\
  x[2] \\
  x[3] \\
  x[4] \\
  x[5] \\
  0 \\
  0
\end{bmatrix}
\]

The convolution matrix \( X \) is a Toeplitz matrix (can be built using the m-function \texttt{toeplitz}).

The m-function \texttt{conv} can be used to compute linear convolution

- in the same range \( 0 \leq n \leq N - 1 \) (using \texttt{conv(x,h,'same')})
- in the full overlap range (using \texttt{conv(x,h,'valid')})
- in the entire range (using \texttt{conv(x,h)})
Properties of LTI systems

**Identity**

\[ x[n] * \delta[n] = x[n] \]

**Delay**

\[ x[n] * \delta[n - n_0] = x[n - n_0] \]

**Commutative**

\[ x[n] * h[n] = h[n] * x[n] \]

**Associative**

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

**Distributive**

\[ x[n] * (h_1[n] + h_2[n]) = x[n] * h_1[n] + x[n] * h_2[n] \]
Causality

An LTI system is causal if $h[n] = 0$ for $n < 0$.

Indeed, $y[n] = \cdots + h[-1]x[n + 1] + h[0]x[n] + h[1]x[n - 1] + \cdots$. So, $y[n]$ won’t depend on the future values $x[n + 1], x[n + 2], x[n + 3], \ldots$, if $h[n] = 0$ for $n < 0$.

Stability

An LTI system is BIBO stable if and only if its impulse response is \textit{absolutely summable}, that is, if $\sum_n |h[n]| < \infty$.

Indeed, we have

$$|y[n]| = \left| \sum_k h[k]x[n - k] \right| \leq \sum_k |h[k]| |x[n - k]| \leq M_x \sum_k |h[k]|.$$

Thus, if $S_h = \sum_n |h[n]| < \infty$ and $M_y = S_h M_x$, then $y[n] \leq M_y$. 
Stability (example)

Is an LTI system with $h[n] = b \, a^n \, u[n]$ BIBO stable?

$$S_h = \sum_{n} |h[n]| = |b| \sum_{n=0}^{\infty} |a|^n$$

Recall that the sum of a geometric series $\{1 \, r \, r^2 \, r^3 \ldots\}$ is given by

$$\sum_{n=0}^{\infty} r^n = \frac{1}{1 - r}, \text{ if } |r| < 1$$

Therefore, if $|a| < 1$, then

$$S_h = |b| \sum_{n=0}^{\infty} |a|^n = \frac{|b|}{1 - |a|} < \infty$$

Thus, the system if stable only if $|a| < 1$. 
An LTI system $\mathcal{H}$ with $h[n] = b a^n u[n]$ is called an infinite impulse response (IIR) filter. In this case,

$$y[n] = \sum_{k=0}^{\infty} h[k] x[n - k] = b x[n] + b a x[n - 1] + b a^2 x[n - 2] + \ldots = b x[n] + a (b x[n - 1] + b a x[n - 2] + \ldots) = b x[n] + a y[n - 1].$$

In this case, we said that $y[n]$ is computed recursively.
In general, LTI systems can be described by *linear constant-coefficient difference equations* (LCCDE) of the following form.

**General recursive systems**

The IO relation of a general LTI system is given by

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

where \(\{a_k\}_{k=1}^{N}\) and \(\{b_k\}_{k=1}^{M}\) are *feedback* and *feedforward* coefficients, respectively.

- \(N\) is known as the *order* of the system.
- If \(N = 0\), then \(y[n] = (x * h)[n]\), with \(h[n] = b_n, n = 0, 1, \ldots, M\).
- In the above case, the system is causal and said to have a *finite impulse response* (FIR).
Real-time implementation of FIR filters

- In most real-time applications, convolution is computed using *stream processing*.

- The delay, $\tau < T$, between the arrival of an input sample and the generation of the associated output sample is known as *latency*. 
Recursive FIR systems

- The *moving average* filter

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

is an FIR system with impulse response

\[ h[n] = \begin{cases} (M + 1)^{-1}, & 0 \leq n \leq M \\ 0, & \text{otherwise} \end{cases} \]

- The system can be implemented as

\[ y[n] = y[n - 1] + \frac{1}{M + 1} (x[n] - x[n - M - 1]) \]

which leads to a recursive implementation.
Echo generation and reverberation

The sound may consist of several components: direct sound, early reflections, and reverberations.

Each component requires appropriate modelling.
A single echo can be modelled using an FIR filter of the form

\[ y[n] = x[n] + a x[n - D]. \]

A multiple echo can be modelled as an IIR system given by

\[ y[n] = x[n] + a x[n - D] + a^2 x[n - 2D] + a^3 x[n - 3D] + \ldots = a y[n - D] + x[n]. \]

Thus, the problem of dereverberation amounts to finding an inverse filter to cancel the effect of the direct (model) filter.
Main questions

Given a system described by an LCCDE of the form

\[ y[n] = - \sum_{k=1}^{N} a_k y[n - k] + \sum_{k=0}^{M} b_k x[n - k] \]

we wish to be able to:

1. Prove that the system is LTI.
2. Analytically determine its \( h[n] \).
3. Given an analytical expression for the input \( x[n] \), find an analytical expression for the output \( y[n] \).
4. Given \( \{a_k, b_k\} \), determine whether or not the system is stable.

Z-transform

All the above questions can be conveniently addressed using the tool of \( z \)-transform.