Ve 216: Introduction to Signals and Systems

Yong Long

The University of Michigan- Shanghai Jiao Tong University Joint Institute
Shanghai Jiao Tong University

December 26, 2017

Based on Lecture Notes by Prof. Jeffrey A. Fessler

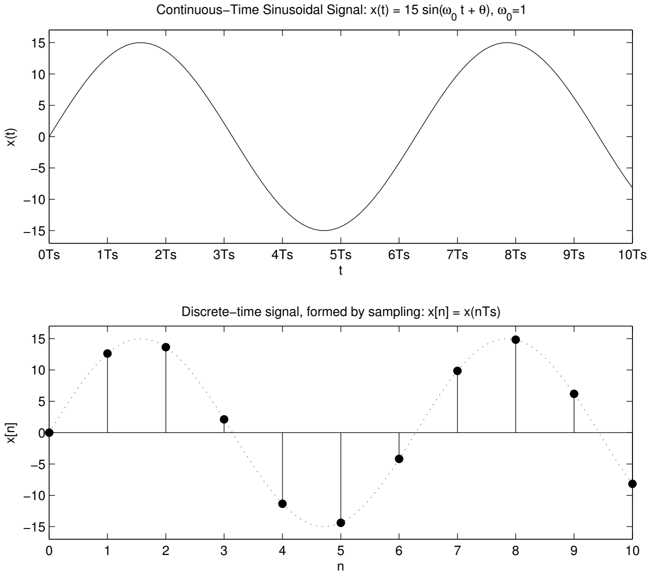
Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
     - Some elementary discrete-time signals
     - Signal notation
     - Signal support characteristics
     - Classification of discrete-time signals
     - Simple manipulations of discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Discrete-time signals (1)



Discrete-time signals (2)

- Defined only at certain specific values of time.
- Typically use t_n, $n = 0, \pm 1, \pm 2, \ldots$ to denote time instants where signal is defined.
- In this course we focus on uniformly spaced time samples

$$t_n = nT,$$

where T denotes the time-spacing between samples.
- In this case we can (and will) use the short hand $x[n]$ as follows:

$$x[n] = x(t_n) = x(nT).$$

Some authors use the notation x_n or $x(n)$.

Discrete-time signals (2)

- Defined only at certain specific values of time.
- Typically use t_n, $n = 0, \pm 1, \pm 2, \ldots$ to denote time instants where signal is defined.
- In this course we focus on uniformly spaced time samples

$$t_n = nT,$$

where T denotes the time-spacing between samples.
- In this case we can (and will) use the short hand $x[n]$ as follows:

$$x[n] = x(t_n) = x(nT).$$

Some authors use the notation x_n or $x(n)$.

Discrete-time signals (2)

- Defined only at certain specific values of time.
- Typically use t_n, $n = 0, \pm 1, \pm 2, \ldots$ to denote time instants where signal is defined.
- In this course we focus on uniformly spaced time samples

$$t_n = nT,$$

where T denotes the time-spacing between samples.
- In this case we can (and will) use the short hand $x[n]$ as follows:

$$x[n] = x(t_n) = x(nT).$$

Some authors use the notation x_n or $x(n)$.

Discrete-time signals (2)

- Defined only at certain specific values of time.
- Typically use t_n, $n = 0, \pm 1, \pm 2, \ldots$ to denote time instants where signal is defined.
- In this course we focus on uniformly spaced time samples

$$t_n = nT,$$

where T denotes the time-spacing between samples.
- In this case we can (and will) use the short hand $x[n]$ as follows:

$$x[n] = x(t_n) = x(nT).$$

Some authors use the notation x_n or $x(n)$.

2D discrete-time signals

For images, we refer to **continuous-space** functions $f(x, y)$ and **discrete-space** functions $x[n, m]$.

Example

When a "black and white" photograph with intensity $f(x, y)$ is scanned by a digital scanner, the output of the scanner is a digital image $x[n, m]$ consisting of uniformly-spaced samples of the original image:

$$x[n, m] = f(n\Delta, m\Delta),$$

where Δ denotes the sample spacing, *e.g.*, "72 dots per inch" means $\Delta = 1/72$ inches.
(PictureMIT, Lecture 17.2 and 17.9)

2D discrete-time signals

For images, we refer to **continuous-space** functions $f(x, y)$ and **discrete-space** functions $x[n, m]$.

Example

When a "black and white" photograph with intensity $f(x, y)$ is scanned by a digital scanner, the output of the scanner is a digital image $x[n, m]$ consisting of uniformly-spaced samples of the original image:

$$x[n, m] = f(n\Delta, m\Delta),$$

where Δ denotes the sample spacing, *e.g.*, "72 dots per inch" means $\Delta = 1/72$ inches.

(PictureMIT, Lecture 17.2 and 17.9)

Digital signal

Definition

A **digital signal** is a discrete-time signal that is also discrete-valued.

If the input to a DSP system is originally an analog signal (*e.g.*, an acoustic voice signal that has been converted to a voltage signal by a microphone), then the A/D converter will convert the analog signal to digital form by **quantizing** its values to a finite discrete set of values.

Digital signal

Definition

A **digital signal** is a discrete-time signal that is also discrete-valued.

If the input to a DSP system is originally an analog signal (*e.g.*, an acoustic voice signal that has been converted to a voltage signal by a microphone), then the A/D converter will convert the analog signal to digital form by **quantizing** its values to a finite discrete set of values.

Digital signal: example

Example

An 8-bit A/D converter can represent $2^8 = 256$ different values. Each value of the input signal must be rounded to the nearest of the 256 output values.

One focus

Our focus: single-channel, continuous-valued signals, namely 1D discrete-time signals $x[n]$.

In mathematical notation we write

$$x : \mathbb{Z} \to \mathbb{R} \quad \text{or} \quad x : \mathbb{Z} \to \mathbb{C}$$

- $x[n]$ can be represented graphically by `stem` plot in MATLAB.
- $x[n]$ is not defined for non-integer n. (It is not "zero" despite appearance of stem plot.)
- We call $x[n]$ the n**th sample** of the signal.

One focus

Our focus: single-channel, continuous-valued signals, namely 1D discrete-time signals $x[n]$.

In mathematical notation we write

$$x : \mathbb{Z} \to \mathbb{R} \quad \text{or} \quad x : \mathbb{Z} \to \mathbb{C}$$

- $x[n]$ can be represented graphically by `stem` plot in MATLAB.
- $x[n]$ is not defined for non-integer n. (It is not "zero" despite appearance of stem plot.)
- We call $x[n]$ the ***n*th sample** of the signal.

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
     - Some elementary discrete-time signals
     - Signal notation
     - Signal support characteristics
     - Classification of discrete-time signals
     - Simple manipulations of discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Unit impulse signal

unit sample sequence or **unit impulse** or **Kronecker delta function** (much simpler than the **Dirac impulse**)

$$\text{Centered: } \delta[n] \triangleq \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

$$\text{Shifted: } \delta[n-k] \triangleq \begin{cases} 1, & n = k \\ 0, & n \neq k \end{cases}$$

(Picture MIT, Lecture 3.1)

Unit impulse signal properties

Property

- *Scaling property:* $\delta[an] = \delta[n]$
  Note that this property is quite different from that of the Dirac impulse.
- *Unity sum:* $\sum_{n=-\infty}^{\infty} \delta[n] = 1$
- *Sampling property:* $\delta[n - n_0]\, f[n] = \delta[n - n_0]\, f[n_0]$
- *Sifting property:* $\sum_{n=-\infty}^{\infty} \delta[n - n_0]\, f[n] = f[n_0]$
- *Symmetry property* $\delta[-n] = \delta[n]$

Unit step signal

unit step signal

$$u[n] \triangleq \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} = \{\ldots, 0, 0, \underline{1}, 1, \ldots\}$$

Useful relationship:

$$\delta[n] = u[n] - u[n-1]$$

. This is the discrete-time analog of the continuous-time property of Dirac impulses:

$$\delta(t) = \frac{\mathrm{d}}{\mathrm{d}t} u(t)$$

(Picture MIT, Lecture 3.1)

Exponential signal

exponential signal or **geometric progression** (discrete-time analog of continuous-time e^{at})

$$x[n] = a^n$$

(Picture MIT, Lecture 2.15)

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
     - Some elementary discrete-time signals
     - Signal notation
     - Signal support characteristics
     - Classification of discrete-time signals
     - Simple manipulations of discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Signal notation

There are several ways to represent discrete-time signals.

- Graphically.
- Set notation:

$$x[n] = \{\ldots, 0, 0, \underline{2}, 1, 1, \ldots\}$$

- In terms of other functions

$$\begin{aligned} x[n] &= u[n] + \delta[n] = 2\,\delta[n] + \delta[n-1] + \delta[n-2] + \cdots \\ &= 2\,\delta[n] + \sum_{k=1}^{\infty} \delta[n-k] \end{aligned}$$

- Braces or piecewise notation:

$$x[n] = \begin{cases} 2, & n = 0, \\ 1, & n \geq 1, \\ 0, & n < 0. \end{cases}$$

- Formula

$$x[n] = a^n$$

Example

Skill: *Convert between different discrete-time signal representations.*
Skill: *Choose representation most appropriate for a given problem.* (There are perhaps more viable options than for CT signals.)

Example

$$
\begin{aligned}
x[n] &= \{\underline{1}, 0, 0, 1/2, 0, 0, 1/4, 0, \ldots\} \\
&= \delta[n-0] + \frac{1}{2}\delta[n-3] + \frac{1}{4}\delta[n-6] + \cdots \\
&= \sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k \delta[n-3k].
\end{aligned}
$$

Example

Skill: ***Convert between different discrete-time signal representations.***
Skill: ***Choose representation most appropriate for a given problem.*** (There are perhaps more viable options than for CT signals.)

Example

$$\begin{aligned} x[n] &= \{\underline{1}, 0, 0, 1/2, 0, 0, 1/4, 0, \ldots\} \\ &= \delta[n-0] + \frac{1}{2}\,\delta[n-3] + \frac{1}{4}\,\delta[n-6] + \cdots \\ &= \sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^{k} \delta[n-3k]\,. \end{aligned}$$

MATLAB implementation

In MATLAB you have two basic choices.

1. Enumeration: `xn = [0 0 1 0 3];` which typically means $x[n] = \delta[n-2] + 3\,\delta[n-5]$
2. Signal synthesis: `n = [-5:4]; x = cos(n);` which means $x[n] = \cos(n)$ for $-5 \leq n \leq 4$ (and $x[n]$ is unspecified outside that range).

The `function_handle` is also useful, *e.g.*, the unit impulse is:
`imp = @(n) n == 0`

Skill: ***Efficiently synthesize simple signals in MATLAB.***

MATLAB implementation

In MATLAB you have two basic choices.

1. Enumeration: `xn = [0 0 1 0 3];` which typically means $x[n] = \delta[n-2] + 3\,\delta[n-5]$
2. Signal synthesis: `n = [-5:4]; x = cos(n);` which means $x[n] = \cos(n)$ for $-5 \leq n \leq 4$ (and $x[n]$ is unspecified outside that range).

The `function_handle` is also useful, *e.g.*, the unit impulse is:

```
imp = @(n) n == 0
```

Skill: ***Efficiently synthesize simple signals in MATLAB.***

MATLAB implementation

In MATLAB you have two basic choices.

1. Enumeration: `xn = [0 0 1 0 3];` which typically means $x[n] = \delta[n-2] + 3\,\delta[n-5]$
2. Signal synthesis: `n = [-5:4]; x = cos(n);` which means $x[n] = \cos(n)$ for $-5 \leq n \leq 4$ (and $x[n]$ is unspecified outside that range).

The `function_handle` is also useful, *e.g.*, the unit impulse is:

```
imp = @(n) n == 0
```

Skill: *Efficiently synthesize simple signals in MATLAB.*

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
     - Some elementary discrete-time signals
     - Signal notation
     - Signal support characteristics
     - Classification of discrete-time signals
     - Simple manipulations of discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Support interval

These are signal characteristics related to the time axis.

Definition

Roughly speaking the **support interval** of a signal is the set of times such that the signal is not zero.

We often abbreviate and say simply **support** or **interval** instead of support interval.

Support interval

These are signal characteristics related to the time axis.

Definition

Roughly speaking the **support interval** of a signal is the set of times such that the signal is not zero.

We often abbreviate and say simply **support** or **interval** instead of support interval.

Support interval: DT signals

1. More precisely the support interval of a continuous-time signal $x(t)$ is the smallest time interval[1] $[t_1, t_2]$ such that the signal is zero outside this interval.
2. For a discrete-time signal $x[n]$, the support interval is a set of consecutive integers: $\{n_1, n_1 + 1, n_1 + 2, \ldots, n_2\}$. Specifically, n_1 is the largest integer such that $x[n] = 0$ for all $n < n_1$, and n_2 is the smallest integer such that $x[n] = 0$ for all $n > n_2$.

[1] Intervals can be open as in (a, b), closed as in $[a, b]$, or half-open, half-closed as in $(a, b]$ and $[a, b)$. For continuous-time signals, in almost all cases of practical interest, it is not necessary to distinguish the support interval as being of one type or the other.

Support interval: DT signals

1. More precisely the support interval of a continuous-time signal $x(t)$ is the smallest time interval[1] $[t_1, t_2]$ such that the signal is zero outside this interval.
2. For a discrete-time signal $x[n]$, the support interval is a set of consecutive integers: $\{n_1, n_1 + 1, n_1 + 2, \ldots, n_2\}$. Specifically, n_1 is the largest integer such that $x[n] = 0$ for all $n < n_1$, and n_2 is the smallest integer such that $x[n] = 0$ for all $n > n_2$.

[1] Intervals can be open as in (a, b), closed as in $[a, b]$, or half-open, half-closed as in $(a, b]$ and $[a, b)$. For continuous-time signals, in almost all cases of practical interest, it is not necessary to distinguish the support interval as being of one type or the other.

Duration

Definition

The **duration** or **length** of a signal is the length of its support interval.

For continuous-time signals, duration = $t_2 - t_1$.

Question

What is the duration of a discrete-time signal?

Duration

Definition

The **duration** or **length** of a signal is the length of its support interval.

For continuous-time signals, duration = $t_2 - t_1$.

Question

What is the duration of a discrete-time signal?

Duration

Definition

The **duration** or **length** of a signal is the length of its support interval.

For continuous-time signals, duration = $t_2 - t_1$.

Question

What is the duration of a discrete-time signal?

Duration

Definition

The **duration** or **length** of a signal is the length of its support interval.

For continuous-time signals, duration = $t_2 - t_1$.

Question

What is the duration of a discrete-time signal?
duration = $n_2 - n_1 + 1$.

Duration: example

Some signals have finite duration and others have infinite duration.

Example

Find the support and duration of the signal

$$x[n] = u[n-3] - u[n-7] + \delta[n-5] + \delta[n-9]$$

Duration: example

Some signals have finite duration and others have infinite duration.

Example

Find the support and duration of the signal

$$x[n] = u[n-3] - u[n-7] + \delta[n-5] + \delta[n-9]$$

Duration: example

Some signals have finite duration and others have infinite duration.

Example

Find the support and duration of the signal

$$x[n] = u[n-3] - u[n-7] + \delta[n-5] + \delta[n-9]$$

support is $\{3, 4, \ldots, 9\}$
duration is 7.

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
     - Some elementary discrete-time signals
     - Signal notation
     - Signal support characteristics
     - Classification of discrete-time signals
     - Simple manipulations of discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Energy and Power

Definition

The **energy** of a signal $x[n]$ is defined as

$$E_x \triangleq \sum_{n=-\infty}^{\infty} |x[n]|^2 .$$

Definition

The **average power** of a signal $x[n]$ is defined as

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2 .$$

- If $E_x < \infty$, we say $x[n]$ is called an **energy signal** and $P = 0$.
- If E_x is infinite, then P can be either finite or infinite. If P is finite and nonzero, then $x[n]$ is called a **power signal**.

Energy and Power

Definition

The **energy** of a signal $x[n]$ is defined as

$$E_x \triangleq \sum_{n=-\infty}^{\infty} |x[n]|^2 .$$

Definition

The **average power** of a signal $x[n]$ is defined as

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2 .$$

- If $E_x < \infty$, we say $x[n]$ is called an **energy signal** and $P = 0$.
- If E_x is infinite, then P can be either finite or infinite. If P is finite and nonzero, then $x[n]$ is called a **power signal**.

Energy and Power

Definition

The **energy** of a signal $x[n]$ is defined as

$$E_x \triangleq \sum_{n=-\infty}^{\infty} |x[n]|^2 .$$

Definition

The **average power** of a signal $x[n]$ is defined as

$$P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x[n]|^2 .$$

- If $E_x < \infty$, we say $x[n]$ is called an **energy signal** and $P = 0$.
- If E_x is infinite, then P can be either finite or infinite. If P is finite and nonzero, then $x[n]$ is called a **power signal**.

Example

Example

Consider $x[n] = 5$ (a constant signal). Is it an energy signal? Is it a power signal?

Example

Example

Consider $x[n] = 5$ (a constant signal). Is it an energy signal? Is it a power signal?

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} 5^2 = \lim_{N\to\infty} 5^2 = 25.$$

So $x[n]$ is a power signal.

Example

Example

Consider $x[n] = 5$ (a constant signal). Is it an energy signal? Is it a power signal?

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} 5^2 = \lim_{N\to\infty} 5^2 = 25.$$

So $x[n]$ is a power signal.
Since P is nonzero, E is infinite.

Periodicity

Definition

A discrete-time signal $x[n]$ is called **periodic** with period $N \in \mathbb{N}$, or N-periodic, if and only if

$$x[n] = x[n + N], \ \forall n \in \mathbb{Z}.$$

The smallest such N is called the **fundamental period** of the signal.

If no such N exists, the signal is called **aperiodic**.

Periodicity

Definition

A discrete-time signal $x[n]$ is called **periodic** with period $N \in \mathbb{N}$, or N-periodic, if and only if

$$x[n] = x[n + N], \ \forall n \in \mathbb{Z}.$$

The smallest such N is called the **fundamental period** of the signal.

If no such N exists, the signal is called **aperiodic**.

Frequency

There are similarities and differences between the meaning of frequency for continuous-time and discrete-time signals.

Consider an analog continuous-time sinusoidal oscillation

$$x(t) = A\cos(2\pi F_0 t + \phi), \; -\infty < t < \infty$$

- A is the **amplitude**
- ϕ is the **phase** in radians
- F_0 is the **frequency** in Hz (cycles per second)
  Some books express the frequency in radians per second using $\boxed{\Omega = 2\pi F_0.}$

Frequency

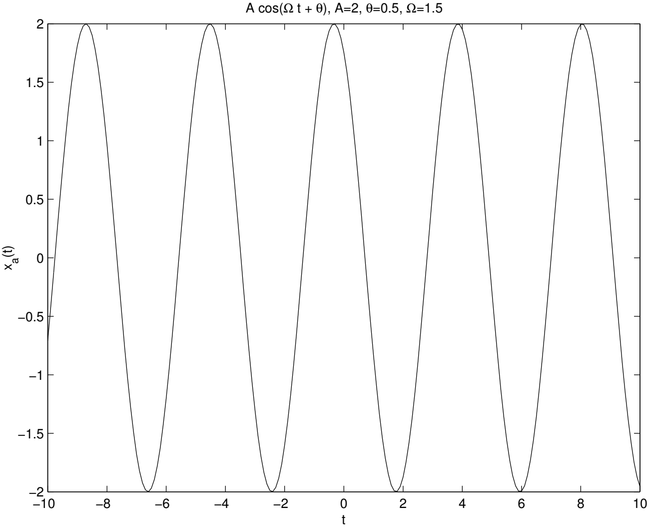
There are similarities and differences between the meaning of frequency for continuous-time and discrete-time signals.

Consider an analog continuous-time sinusoidal oscillation

$$x(t) = A\cos(2\pi F_0 t + \phi), \; -\infty < t < \infty$$

- A is the **amplitude**
- ϕ is the **phase** in radians
- F_0 is the **frequency** in Hz (cycles per second)
  Some books express the frequency in radians per second using $\boxed{\Omega = 2\pi F_0.}$

Continuous-Time Sinusoidal Signals



Properties of analog sinusoidal signals

Property

1. *The signal $x(t)$ is periodic for any fixed value of F_0, i.e.,*

$$x(t+T_0) = x(t)$$

*where $T_0 = 1/F_0$ is the **fundamental period** of the signal.*

2. *Continuous-time sinusoidal signals with distinct (different) frequencies are distinct. If $F_1 \neq F_2$ (and both have the same sign) then $\exists t_0$ s.t. $A\cos(2\pi F_1 t_0 + \phi) \neq A\cos(2\pi F_2 t_0 + \phi)$.*

3. *Increasing the frequency F_0 increases the **rate of oscillation**, i.e., there will be more periods in a given time interval.*

Properties of analog sinusoidal signals

Property

1. *The signal $x(t)$ is periodic for any fixed value of F_0, i.e.,*

$$x(t + T_0) = x(t)$$

   *where $T_0 = 1/F_0$ is the **fundamental period** of the signal.*
2. *Continuous-time sinusoidal signals with distinct (different) frequencies are distinct. If $F_1 \neq F_2$ (and both have the same sign) then $\exists t_0$ s.t. $A\cos(2\pi F_1 t_0 + \phi) \neq A\cos(2\pi F_2 t_0 + \phi)$.*
3. *Increasing the frequency F_0 increases the **rate of oscillation**, i.e., there will be more periods in a given time interval.*

Properties of analog sinusoidal signals

Property

1. *The signal $x(t)$ is periodic for any fixed value of F_0, i.e.,*

   $$x(t+T_0) = x(t)$$

   *where $T_0 = 1/F_0$ is the **fundamental period** of the signal.*

2. *Continuous-time sinusoidal signals with distinct (different) frequencies are distinct. If $F_1 \neq F_2$ (and both have the same sign) then $\exists t_0$ s.t.* $A\cos(2\pi F_1 t_0 + \phi) \neq A\cos(2\pi F_2 t_0 + \phi)$.

3. *Increasing the frequency F_0 increases the **rate of oscillation**, i.e., there will be more periods in a given time interval.*

Complex exponential signals

The above relationships also hold for **complex exponential** signals

$$x(t) = Ae^{j(2\pi Ft+\phi)}$$

where by the **Euler identity**

$$e^{\pm jS\theta} = \cos\theta \pm \jmath \sin\theta.$$

For sinusoidal signals, the frequency F is usually taken to be nonnegative.

Complex exponential signals

The above relationships also hold for **complex exponential** signals

$$x(t) = Ae^{j(2\pi Ft+\phi)}$$

where by the **Euler identity**

$$e^{\pm jS\theta} = \cos\theta \pm \jmath \sin\theta.$$

For sinusoidal signals, the frequency F is usually taken to be nonnegative.

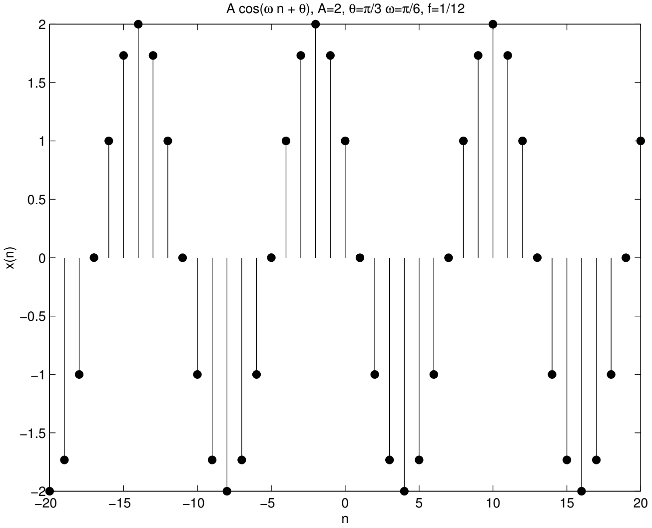
Discrete-Time Sinusoidal Signals

A discrete-time sinusoidal signal may be expressed

$$x[n] = A\cos(\omega_0 n + \phi),\ n = 0, \pm 1, \pm 2, \ldots$$

- n is an integer variable called the **sample number**
- A is the **amplitude**
- ϕ is the **phase** in radians
- ω_0 is the **frequency** in radians per sample
  Sometimes we express the frequency in cycles per sample using f, where $\boxed{\omega_0 = 2\pi f.}$

Picture of discrete-time sinusoid



Properties of discrete-time sinusoidal signals (1)

Property

*A DT sinusoidal signal $x[n]$ is **periodic** if and only if its frequency ω is 2π times a **rational number**, i.e.,*

$$\omega = 2\pi\frac{M}{N} \quad \textit{for} \quad M, N \in \mathbb{Z}.$$

Question

Why periodic only for rational frequencies?

Properties of discrete-time sinusoidal signals (1)

Property

A DT sinusoidal signal $x[n]$ is ***periodic*** *if and only if its frequency ω is 2π times a* ***rational number****, i.e.,*

$$\omega = 2\pi\frac{M}{N} \quad \textit{for} \quad M, N \in \mathbb{Z}.$$

Question

Why periodic only for rational frequencies?

Proof

Sinusoidal discrete-time signal with frequency ω_0 *is periodic iff* $\exists N$ *s.t.*

$$\cos(\omega_0 n + \phi) = \cos(\omega_0 (n + N) + \phi) = \cos(\omega_0 n + \phi + \omega_0 N)\,.$$

Since cos *is periodic with fundamental period* 2π, *the above relationship holds iff* $\omega_0 N$ *is an integer multiple of* 2π, *i.e.,*

exists an integer M s.t. $\omega_0 N = 2\pi M$, *or equivalently* $\omega_0 = 2\pi \frac{M}{N}$.

Thus the frequency must be a ratio of two integers, and hence rational.

Fundamental period of a DT sinusoidal signal

Skill: *Finding the fundamental period of a DT sinusoidal signal.*

1. Express $\omega_0 = 2\pi M/N$, where $M \in \mathbb{Z}$ and $N \in \mathbb{N}$ and M and N have no common divisors.
2. Then N will be the fundamental period (in samples).
3. If no such ratio, then $\frac{\omega_0}{2\pi}$ is irrational and the DT sinusoidal signal is aperiodic.

To convert a quantity, such as the period N, from units "samples" to time units (*e.g.*, seconds), multiply by the sampling rate $T_s = 1/F(s)$.

Fundamental period of a DT sinusoidal signal

Skill: *Finding the fundamental period of a DT sinusoidal signal.*

1. Express $\omega_0 = 2\pi M/N$, where $M \in \mathbb{Z}$ and $N \in \mathbb{N}$ and M and N have no common divisors.
2. Then N will be the fundamental period (in samples).
3. If no such ratio, then $\frac{\omega_0}{2\pi}$ is irrational and the DT sinusoidal signal is aperiodic.

To convert a quantity, such as the period N, from units "samples" to time units (*e.g.*, seconds), multiply by the sampling rate $T_s = 1/F(s)$.

Properties of discrete-time sinusoidal signals (2)

Property

DT sinusoidal signals with frequencies separated by an integer multiple of 2π are identical (indistinguishable).

If $\omega_2 = \omega_1 + k2\pi$ then

$$\begin{aligned}\cos(\omega_2 n + \phi) &= \cos((\omega_1 + 2\pi k)n + \phi) = \cos(\omega_1 n + \phi + 2\pi kn) \\ &= \cos(\omega_1 n + \phi)\end{aligned}$$

since cos is 2π periodic.

Properties of discrete-time sinusoidal signals (2)

Property

DT sinusoidal signals with frequencies separated by an integer multiple of 2π are identical (indistinguishable).

If $\omega_2 = \omega_1 + k2\pi$ then

$$\begin{aligned}\cos(\omega_2 n + \phi) &= \cos((\omega_1 + 2\pi k)n + \phi) = \cos(\omega_1 n + \phi + 2\pi kn) \\ &= \cos(\omega_1 n + \phi)\end{aligned}$$

since cos is 2π periodic.

Properties of discrete-time sinusoidal signals (3)

Property

*The highest **rate of oscillation** of a discrete-time sinusoid is attained when $\omega = \pi$.*

- Discrete-time sinusoidal signals with frequencies in the range $0 \leq \omega \leq \pi$ are distinct.
- Any DT sinusoidal signal with frequency outside the range $0 \leq \omega \leq \pi$ is identical to a DT sinusoidal signal with a frequency within that range (possibly with the negative phase). Hence we refer to such signals as **aliases**, since they are "the same signal with a different name."

Properties of discrete-time sinusoidal signals (3)

Property

The highest ***rate of oscillation*** *of a discrete-time sinusoid is attained when* $\omega = \pi$.

- Discrete-time sinusoidal signals with frequencies in the range $0 \leq \omega \leq \pi$ are distinct.
- Any DT sinusoidal signal with frequency outside the range $0 \leq \omega \leq \pi$ is identical to a DT sinusoidal signal with a frequency within that range (possibly with the negative phase). Hence we refer to such signals as **aliases**, since they are "the same signal with a different name."

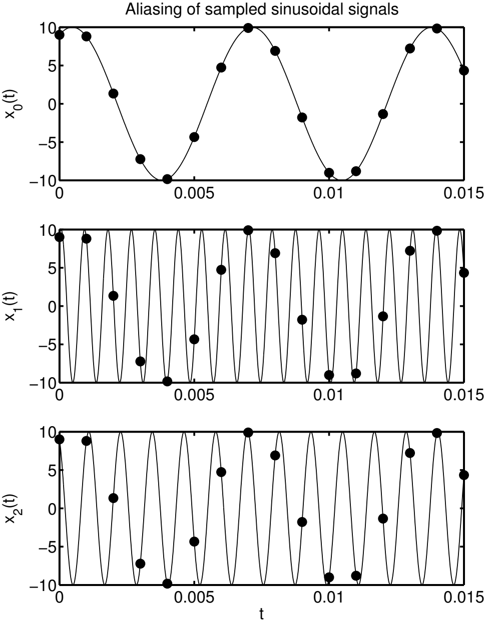
Example

Example

$$\cos\left(\frac{19\pi}{7}n+\frac{\pi}{3}\right) = \cos\left(\frac{5\pi}{7}n+2\pi n+\frac{\pi}{3}\right) = \cos\left(\frac{5\pi}{7}n+\frac{\pi}{3}\right)$$

$$\begin{aligned}
\cos\left(\frac{9\pi}{7}n+\frac{\pi}{3}\right) &= \cos\left(\frac{-9\pi}{7}n-\frac{\pi}{3}\right) \\
&= \cos\left(\frac{5\pi}{7}n-2\pi n-\frac{\pi}{3}\right) \\
&= \cos\left(\frac{5\pi}{7}n-\frac{\pi}{3}\right)
\end{aligned}$$

Aliasing illustrated



Aliasing illustrated

Three distinct CT sinusoidal signals whose DT samples are identical.

DT **complex exponential** signals:

$$x[n] = Ae^{j(\omega n+\phi)}$$

For sinusoidal signals, we usually take the frequency to be nonnegative.

Symmetry

Definition

- $x[n]$is **symmetric** or **even** iff

$$x[-n] = x[n]$$

- $x[n]$is **antisymmetric** or **odd** iff

$$x[-n] = -x[n]$$

We can decompose any signal into even and odd components:

$$\begin{aligned} x[n] &= x[n]_e + x[n]_o \\ x[n]_e &\triangleq \frac{1}{2}(x[n] + x[-n]) \\ x[n]_o &\triangleq \frac{1}{2}(x[n] - x[-n]) \end{aligned}$$

Symmetry

Definition

- $x[n]$is **symmetric** or **even** iff

$$x[-n] = x[n]$$

- $x[n]$is **antisymmetric** or **odd** iff

$$x[-n] = -x[n]$$

We can decompose any signal into even and odd components:

$$\begin{aligned} x[n] &= x[n]_e + x[n]_o \\ x[n]_e &\triangleq \frac{1}{2}(x[n] + x[-n]) \\ x[n]_o &\triangleq \frac{1}{2}(x[n] - x[-n]) \end{aligned}$$

Symmetry: example

Example

Find the even and odd component of the signal

$$x[n] = 2\,u[n]$$

Symmetry: example

Example

Find the even and odd component of the signal

$$x[n] = 2\,u[n]$$

$$x[n]_e = \frac{1}{2}\,(2\,u[n] + 2\,u[-n]) = 1 + \delta[n]$$

$$x[n]_o = \frac{1}{2}\,(2\,u[n] - 2\,u[-n]) = u[n-1] - u[1-n]$$

$$\{\ldots, 0, 0, \underline{2}, 2, 2, \ldots\} = \{\ldots, 1, 1, \underline{2}, 1, 1, \ldots\} + \{\ldots, -1, -1, \underline{0}, 1, 1, \ldots\}$$

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
     - Some elementary discrete-time signals
     - Signal notation
     - Signal support characteristics
     - Classification of discrete-time signals
     - Simple manipulations of discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Amplitude modifications

Amplitude modifications

- amplitude scaling $y[n] = ax[n]$
- amplitude shift $y[n] = x[n] + b$
- sum of two signals $y[n] = x[n]_1 + x[n]_2$
- product of two signals $y[n] = x[n]_1 x[n]_2$

Amplitude modifications

Amplitude modifications

- amplitude scaling $y[n] = ax[n]$
- amplitude shift $y[n] = x[n] + b$
- sum of two signals $y[n] = x[n]_1 + x[n]_2$
- product of two signals $y[n] = x[n]_1 x[n]_2$

Amplitude modifications

Amplitude modifications

- amplitude scaling $y[n] = ax[n]$
- amplitude shift $y[n] = x[n] + b$
- sum of two signals $y[n] = x[n]_1 + x[n]_2$
- product of two signals $y[n] = x[n]_1 x[n]_2$

Amplitude modifications

Amplitude modifications

- amplitude scaling $y[n] = ax[n]$
- amplitude shift $y[n] = x[n] + b$
- sum of two signals $y[n] = x[n]_1 + x[n]_2$
- product of two signals $y[n] = x[n]_1 x[n]_2$

Time modifications

Time modifications

- Time shifting $y[n] = x[n-k]$.
  k can be positive (delayed signal) or negative (advanced signal) if signal stored in a computer
- Folding or reflection or time-reversal $y[n] = x[-n]$
- Time-scaling or **down-sampling** $y[n] = x[2n]$. (discard every other sample) (*cf.* continuous $f(t) = g(2t)$)

Question

Why down-sampling?

Time modifications

Time modifications

- Time shifting $y[n] = x[n-k]$.
  k can be positive (delayed signal) or negative (advanced signal) if signal stored in a computer
- Folding or reflection or time-reversal $y[n] = x[-n]$
- Time-scaling or **down-sampling** $y[n] = x[2n]$. (discard every other sample) (*cf.* continuous $f(t) = g(2t)$)

Question

Why down-sampling?

Time modifications

Time modifications

- Time shifting $y[n] = x[n-k]$.
  k can be positive (delayed signal) or negative (advanced signal) if signal stored in a computer
- Folding or reflection or time-reversal $y[n] = x[-n]$
- Time-scaling or **down-sampling** $y[n] = x[2n]$. (discard every other sample) (*cf.* continuous $f(t) = g(2t)$)

Question

Why down-sampling?

Time modifications

Time modifications

- Time shifting $y[n] = x[n-k]$.
  k can be positive (delayed signal) or negative (advanced signal) if signal stored in a computer
- Folding or reflection or time-reversal $y[n] = x[-n]$
- Time-scaling or **down-sampling** $y[n] = x[2n]$. (discard every other sample) (*cf.* continuous $f(t) = g(2t)$)

Question

Why down-sampling?

Time modifications

Time modifications

- Time shifting $y[n] = x[n-k]$.
  k can be positive (delayed signal) or negative (advanced signal) if signal stored in a computer
- Folding or reflection or time-reversal $y[n] = x[-n]$
- Time-scaling or **down-sampling** $y[n] = x[2n]$. (discard every other sample) (*cf.* continuous $f(t) = g(2t)$)

Question

Why down-sampling?

to reduce CPU time in a preliminary data analysis, or to reduce memory.

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
     - Input-output description of systems
     - Block diagram representation of discrete-time systems
     - Classification of discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Discrete-time systems (1)

Definition

A **discrete-time system** is a device or algorithm that, according to some well-defined rule, operates on a discrete-time signal called the **input signal** or **excitation** to produce another discrete-time signal called the **output signal** or **response**.

Mathematically speaking, a system is also a **function**.

Discrete-time systems (1)

Definition

A **discrete-time system** is a device or algorithm that, according to some well-defined rule, operates on a discrete-time signal called the **input signal** or **excitation** to produce another discrete-time signal called the **output signal** or **response**.

Mathematically speaking, a system is also a **function**.

Discrete-time systems (2)

The input signal $x[n]$ is **transformed** by the system into a signal $y[n]$, which we express mathematically as

$$y[\cdot] = \mathcal{T}x[\cdot] \quad \text{or} \quad y[n] = \mathcal{T}x[\cdot][n] \quad \text{or} \quad x[\cdot] \xrightarrow{\mathcal{T}} y[\cdot]\,.$$

The notation $y[n] = \mathcal{T}[x[n]]$ is mathematically vague. In general $y[n]$ is a function of the entire sequence $\{x[n]\}$, not just the single time point $x[n]$.

$$\begin{array}{c} \text{Input Signal} \\ x[n] \end{array} \rightarrow \boxed{\text{Discrete-time system}} \rightarrow \begin{array}{c} \text{Output Signal} \\ y[n] \end{array}$$

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
     - Input-output description of systems
     - Block diagram representation of discrete-time systems
     - Classification of discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Input-output relationship

A discrete-time system can be described in many ways. One way is by its **input-output relationship**, which is a formula expressing the output signal in terms of the input signal.

Example

The accumulator system.

1. $y[n] = \sum_{k=-\infty}^{n} x[k] = x[n] + x[n-1] + x[n-2] + \cdots$

$$x[n] = u[n] - u[n-3] = \{\ldots, 0, 0, \underline{1}, 1, 1, 0, 0, \ldots\}$$

2. $y[n] = \{\ldots, 0, 0, \underline{1}, 2, 3, 3, \ldots\}$.
3. Alternative expression: $y[n] = \sum_{k=-\infty}^{n} x[k] = \sum_{k=-\infty}^{n-1} x[k] + x[n] = y[n-1] + x[n]$

Input-output relationship

A discrete-time system can be described in many ways. One way is by its **input-output relationship**, which is a formula expressing the output signal in terms of the input signal.

Example

The accumulator system.

1. $y[n] = \sum_{k=-\infty}^{n} x[k] = x[n] + x[n-1] + x[n-2] + \cdots$

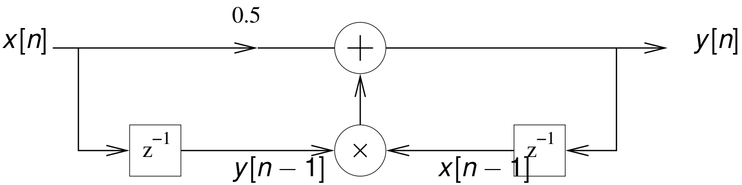
$$x[n] = u[n] - u[n-3] = \{\ldots, 0, 0, \underline{1}, 1, 1, 0, 0, \ldots\}$$

2. $y[n] = \{\ldots, 0, 0, \underline{1}, 2, 3, 3, \ldots\}$.
3. Alternative expression:
$y[n] = \sum_{k=-\infty}^{n} x[k] = \sum_{k=-\infty}^{n-1} x[k] + x[n] = y[n-1] + x[n]$

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
     - Input-output description of systems
     - Block diagram representation of discrete-time systems
     - Classification of discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Block diagram



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

- adder
- constant multiplier
- signal multiplier
- **unit delay** element (why z^{-1} clear later)

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
     - Input-output description of systems
     - Block diagram representation of discrete-time systems
     - Classification of discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Classification of discrete-time systems

Skill: *Determining classifications of a given DT system*

Two general aspects to categorize:

1. time properties
    1. causality
    2. memory
    3. time invariance
2. amplitude properties
    1. stability
    2. invertibility
    3. linearity

Causality

For a **causal** system, the output $y[n]$ at any time n depends only on the "present" and "past" inputs, *i.e.*,

$$y[n] = F\{x[n], x[n-1], x[n-2], \ldots\}$$

where $F\{\cdot\}$ is any function.
Otherwise **noncausal** system.

Causality is necessary for real-time implementation, but many DSP problems involved stored data, *e.g.*, image processing (OCR) or restoration of analog audio recordings.

Causality

For a **causal** system, the output $y[n]$ at any time n depends only on the "present" and "past" inputs, *i.e.*,

$$y[n] = F\{x[n], x[n-1], x[n-2], \ldots\}$$

where $F\{\cdot\}$ is any function.
Otherwise **noncausal** system.

Causality is necessary for real-time implementation, but many DSP problems involved stored data, *e.g.*, image processing (OCR) or restoration of analog audio recordings.

Causality

For a **causal** system, the output $y[n]$ at any time n depends only on the "present" and "past" inputs, *i.e.*,

$$y[n] = F\{x[n], x[n-1], x[n-2], \ldots\}$$

where $F\{\cdot\}$ is any function.
Otherwise **noncausal** system.

Causality is necessary for real-time implementation, but many DSP problems involved stored data, *e.g.*, image processing (OCR) or restoration of analog audio recordings.

Memory

- For a **static system** or **memoryless** system, the output $y[n]$ depends only on the current input $x[n]$, not on previous or future inputs.
  Example: $y[n] = e^{x[n]}/\sqrt{n-2}$.
- Otherwise it is a **dynamic system** and must have memory.

Dynamic systems are the interesting ones and will be our focus. (This time we take the more complicated choice!)

Memory

- For a **static system** or **memoryless** system, the output $y[n]$ depends only on the current input $x[n]$, not on previous or future inputs.
  Example: $y[n] = e^{x[n]}/\sqrt{n-2}$.
- Otherwise it is a **dynamic system** and must have memory.

Dynamic systems are the interesting ones and will be our focus. (This time we take the more complicated choice!)

Memory

- For a **static system** or **memoryless** system, the output $y[n]$ depends only on the current input $x[n]$, not on previous or future inputs.
  Example: $y[n] = e^{x[n]}/\sqrt{n-2}$.
- Otherwise it is a **dynamic system** and must have memory.

Dynamic systems are the interesting ones and will be our focus. (This time we take the more complicated choice!)

Memory & Causality

Question

1. *Is a memoryless system necessarily causal?*
2. *Is a dynamic system necessarily noncausal?*

Memory & Causality

Question

1. *Is a memoryless system necessarily causal?* Yes.
2. *Is a dynamic system necessarily noncausal?* No.
   Dynamic systems van be causal or noncausal.

Time invariance (1)

Systems whose input-output behavior does not change with time are called **time-invariant** and will be our focus.

- "Easier" to analyze.
- Time-invariance is a desired property of many systems.

Time invariance (1)

Definition

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

$$x[n] \xrightarrow{\mathcal{T}} y[n] \quad \text{implies that} \quad x[n-k] \xrightarrow{\mathcal{T}} y[n-k]$$

for every input signal $x[n]$ and integer time shift k.
Otherwise the system is called **time variant** or **shift variant**.

Graphically:

$$x[n] \rightarrow \boxed{\text{delay } z^{-k}} \rightarrow \boxed{\text{system } \mathcal{T}} \rightarrow y[n]$$
$$x[n] \rightarrow \boxed{\text{system } \mathcal{T}} \rightarrow \boxed{\text{delay } z^{-k}} \rightarrow y[n]$$

Time invariance (1)

Definition

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

$$x[n] \xrightarrow{\mathcal{T}} y[n] \quad \text{implies that} \quad x[n-k] \xrightarrow{\mathcal{T}} y[n-k]$$

for every input signal $x[n]$ and integer time shift k.
Otherwise the system is called **time variant** or **shift variant**.

Graphically:

$$x[n] \quad \rightarrow \quad \boxed{\text{delay } z^{-k}} \rightarrow \boxed{\text{system } \mathcal{T}} \rightarrow y[n]$$

$$x[n] \quad \rightarrow \quad \boxed{\text{system } \mathcal{T}} \rightarrow \boxed{\text{delay } z^{-k}} \rightarrow y[n]$$

Recipe for showing time-invariance

Recipe for showing time-invariance.

1. Determine output $y_1[n]$ due to a generic input $x[n]_1$.
2. Determine the delayed output signal $y_1[n-k]$, by replacing n with $n-k$ in $y[n]$ expression.
3. Determine output $y_2[n]$ due to a delayed input $x_2[n] = x_1[n-k]$.
4. If $y_2[n] = y_1[n-k]$, then system is time-invariant.

Example (1)

Example

3-point **moving average** $y[n] = \frac{1}{3}\left(x[n-1] + x[n] + x[n+1]\right)$.
Is this system time invariant?

Example (1)

Example

3-point **moving average** $y[n] = \frac{1}{3}\left(x[n-1] + x[n] + x[n+1]\right).$
Is this system time invariant?

Yes.

Example (2)

Example

down-sampler $y[n] = x[2n]$. Is this system time invariant?

Example (2)

Example

down-sampler $y[n] = x[2n]$. Is this system time invariant?

No. How do we show lack of a property? Find counter-example.

- *If* $x_1[n] = \delta[n]$ *then* $y_1[n] = \delta[2n] = \delta[n]$.
- *If* $x_2[n] = \delta[n-1]$ *then*

$$y_2[n] = x_2[2n] = \delta[2n-1] = 0 \neq \delta[n-1].$$

Simple counter-example is all that is needed.

Invertibility

A system $\mathcal{T}$ is **invertible** if every output signal corresponds to a unique input signal. If so, then there exists an **inverse system** $\mathcal{T}^{-1}$ that can recover the input signal:

$$x[n] \to \boxed{\mathcal{T}} \to y[n] \to \boxed{\mathcal{T}^{-1}} \to x[n]$$

Stability

A system is **bounded-input bounded-output (BIBO) stable** iff every bounded input produces a bounded output.

If $\exists M_x$ s.t. $|x[n]| \leq M_x < \infty \; \forall n$, then there must exist an M_y s.t.

$$|y[n]| \leq M_y < \infty \; \forall n.$$

Usually M_y will depend on M_x.

This property is important for digital systems because all such systems have a finite maximum signal value that can be represented with a finite number of bits, so one must worry about "over-ranging" or "over flow," *i.e.*, exceeding the maximum value that can be represented.

Stability

A system is **bounded-input bounded-output (BIBO) stable** iff every bounded input produces a bounded output.

If $\exists M_x$ s.t. $|x[n]| \le M_x < \infty \; \forall n$, then there must exist an M_y s.t.

$$|y[n]| \le M_y < \infty \; \forall n.$$

Usually M_y will depend on M_x.

This property is important for digital systems because all such systems have a finite maximum signal value that can be represented with a finite number of bits, so one must worry about "over-ranging" or "over flow," *i.e.*, exceeding the maximum value that can be represented.

Stability: example

Example

Is the accumulator $y[n] = y[n-1] + x[n]$ stable?

Stability: example

Example

Is the accumulator $y[n] = y[n-1] + x[n]$ stable?

Consider input signal $x[n] = u[n]$, which is bounded by $M_x = 1$. But

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

blows up, so the accumulator is an unstable system.

Linearity (1)

Definition

A system $\mathcal{T}$ is **linear** iff

$$\mathcal{T}[a_1 x[n]_1 + a_2 x[n]_2] = a_1 \mathcal{T}[x[n]_1] + a_2 \mathcal{T}[x[n]_2]$$

i.e.,

$$a_1 x[n]_1 + a_2 x[n]_2 \xrightarrow{\mathcal{T}} a_1\, y_1[n] + a_2\, y_2[n],$$

for any signals $x[n]_1, x[n]_2$, and constants a_1 and a_2.
Otherwise the system is called **nonlinear**.

Response to a weighted sum of input signals is the weighted sum of the individual responses.

Linearity (1)

Definition

A system $\mathcal{T}$ is **linear** iff

$$\mathcal{T}[a_1 x[n]_1 + a_2 x[n]_2] = a_1 \mathcal{T}[x[n]_1] + a_2 \mathcal{T}[x[n]_2]$$

i.e.,

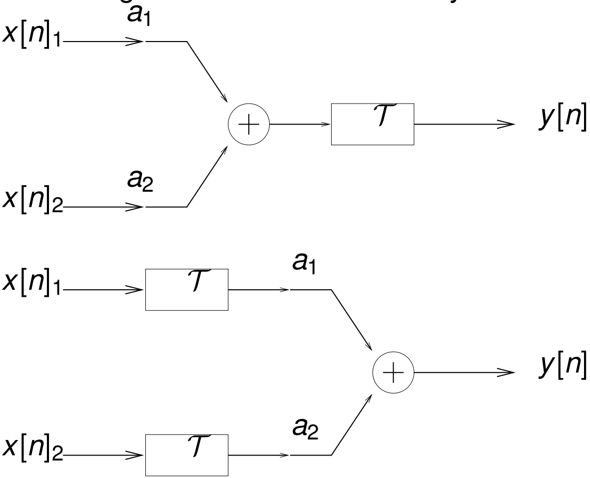
$$a_1 x[n]_1 + a_2 x[n]_2 \xrightarrow{\mathcal{T}} a_1\, y_1[n] + a_2\, y_2[n],$$

for any signals $x[n]_1, x[n]_2$, and constants a_1 and a_2.
Otherwise the system is called **nonlinear**.

Response to a weighted sum of input signals is the weighted sum of the individual responses.

Linearity (2)

Block diagram illustration of linearity test.



Linearity (3)

Question

We will focus on linear systems. Why?

Linearity (3)

Question

We will focus on linear systems. Why?

1. *The class of linear systems is easier to analyze.*
2. *Often linearity is desirable - avoids distortions (Example: amplifiers, audio mixers (superposition!)).*
3. *Many nonlinear systems are approximately linear.*

Two important special cases of linearity property (1)

Property

scaling property or *homogeneity property*:

$$\mathcal{T}[ax[n]] = aT[x[n]]$$

Note that from $a = 0$ we see that zero input signal implies zero output signal for a linear system.

Two important special cases of linearity property (1)

Property

scaling property or *homogeneity property*:

$$\mathcal{T}[ax[n]] = aT[x[n]]$$

Note that from $a = 0$ we see that zero input signal implies zero output signal for a linear system.

Two important special cases of linearity property (2)

Property

additivity property:

$$\mathcal{T}[x[n]_1 + x[n]_2] = \mathcal{T}[x[n]_1] + \mathcal{T}[x[n]_2]$$

Using proof-by-induction, one can easily extend this property to the general superposition property

Property

general superposition property

$$\mathcal{T}\left[\sum_{k=1}^{K} x[n]_k\right] = \sum_{k=1}^{K} \mathcal{T}[x[n]_k]$$

In words: the response of a linear system to the sum of several signals is the sum of the response to each of the signals.

Two important special cases of linearity property (2)

Property

additivity property:

$$\mathcal{T}[x[n]_1 + x[n]_2] = \mathcal{T}[x[n]_1] + \mathcal{T}[x[n]_2]$$

Using proof-by-induction, one can easily extend this property to the general superposition property

Property

general superposition property

$$\mathcal{T}\left[\sum_{k=1}^{K} x[n]_k\right] = \sum_{k=1}^{K} \mathcal{T}[x[n]_k]$$

In words: the response of a linear system to the sum of several signals is the sum of the response to each of the signals.

Two important special cases of linearity property (2)

Property

additivity property:

$$\mathcal{T}[x[n]_1 + x[n]_2] = \mathcal{T}[x[n]_1] + \mathcal{T}[x[n]_2]$$

Using proof-by-induction, one can easily extend this property to the general superposition property

Property

general superposition property

$$\mathcal{T}\left[\sum_{k=1}^{K} x[n]_k\right] = \sum_{k=1}^{K} \mathcal{T}[x[n]_k]$$

In words: the response of a linear system to the sum of several signals is the sum of the response to each of the signals.

General superposition property

- In general superposition need not hold for infinite sums; additional continuity assumptions are required.
- We assume the superposition summation holds even for infinite sums without further comment in this course.

General superposition property

- In general superposition need not hold for infinite sums; additional continuity assumptions are required.
- We assume the superposition summation holds even for infinite sums without further comment in this course.

Determining a system is linear or nonlinear

Skill: *Determining a system is linear or nonlinear.*

1. Find output signal $y_1[n]$ for a general input signal $x_1[n]$.
2. "Repeat" for input $x_2[n]$ and $y_2[n]$.
3. Find output signal $y[n]$ when input signal is $x[n] = a_1 x_1[n] + a_2 x_2[n]$.
4. If $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then the system is linear.
5. If it does not appear that $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then find a specific counter-example.

Determining a system is linear or nonlinear

Skill: *Determining a system is linear or nonlinear.*

1. Find output signal $y_1[n]$ for a general input signal $x_1[n]$.
2. "Repeat" for input $x_2[n]$ and $y_2[n]$.
3. Find output signal $y[n]$ when input signal is $x[n] = a_1 x_1[n] + a_2 x_2[n]$.
4. If $y[n] = a_1\, y_1[n] + a_2\, y_2[n]$ $\forall t$, then the system is linear.
5. If it does not appear that $y[n] = a_1\, y_1[n] + a_2\, y_2[n]$ $\forall t$, then find a specific counter-example.

Determining a system is linear or nonlinear

Skill: *Determining a system is linear or nonlinear.*

1. Find output signal $y_1[n]$ for a general input signal $x_1[n]$.
2. "Repeat" for input $x_2[n]$ and $y_2[n]$.
3. Find output signal $y[n]$ when input signal is $x[n] = a_1 x_1[n] + a_2 x_2[n]$.
4. If $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then the system is linear.
5. If it does not appear that $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then find a specific counter-example.

Determining a system is linear or nonlinear

Skill: *Determining a system is linear or nonlinear.*

1. Find output signal $y_1[n]$ for a general input signal $x_1[n]$.
2. "Repeat" for input $x_2[n]$ and $y_2[n]$.
3. Find output signal $y[n]$ when input signal is $x[n] = a_1 x_1[n] + a_2 x_2[n]$.
4. If $y[n] = a_1\, y_1[n] + a_2\, y_2[n]$ $\forall t$, then the system is linear.
5. If it does not appear that $y[n] = a_1\, y_1[n] + a_2\, y_2[n]$ $\forall t$, then find a specific counter-example.

Determining a system is linear or nonlinear

Skill: *Determining a system is linear or nonlinear.*

1. Find output signal $y_1[n]$ for a general input signal $x_1[n]$.
2. "Repeat" for input $x_2[n]$ and $y_2[n]$.
3. Find output signal $y[n]$ when input signal is $x[n] = a_1 x_1[n] + a_2 x_2[n]$.
4. If $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then the system is linear.
5. If it does not appear that $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then find a specific counter-example.

Determining a system is linear or nonlinear

Skill: *Determining a system is linear or nonlinear.*

1. Find output signal $y_1[n]$ for a general input signal $x_1[n]$.
2. "Repeat" for input $x_2[n]$ and $y_2[n]$.
3. Find output signal $y[n]$ when input signal is $x[n] = a_1 x_1[n] + a_2 x_2[n]$.
4. If $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then the system is linear.
5. If it does not appear that $y[n] = a_1\, y_1[n] + a_2\, y_2[n]\ \forall t$, then find a specific counter-example.

Linearity: example (1)

Example

Is the accumulator $y[n] = \sum_{k=-\infty}^{n} x[k]$ a linear system?

Linearity: solution (1)

For the accumulator,

1. $y_1[n] = \sum_{k=-\infty}^{n} x_1[n]$
2. $y_2[n] = \sum_{k=-\infty}^{n} x_2[n]$.
3. *If the input is* $x[n] = a_1 x_1[n] + a_2 x_2[n]$ *then the output is*

$$y[n] = \sum_{k=-\infty}^{n} x[n] = \sum_{k=-\infty}^{n} (a_1 x_1[n] + a_2 x_2[n])$$

$$= a_1 \sum_{k=-\infty}^{n} x_1[n] + a_2 \sum_{k=-\infty}^{n} x_2[n] = a_1\, y_1[n] + a_2\, y_2[n]\,.$$

4. *Since this holds for all* n, *for all input signals* $x_1[n]$ *and* $x_2[n]$, *and for any constants* a_1 *and* a_2, *the accumulator is linear.*

Linearity: example (2)

Example

Is the system $y[n] = \sqrt{x[n]}$ linear?

Linearity: example (2)

Example

Is the system $y[n] = \sqrt{x[n]}$ linear?

To show a system is nonlinear all that is needed is a counter-example to the above linearity properties. The scaling property will usually suffice.

- *Let $x_1[n] = 2$, a constant signal. Then $y_1[n] = \sqrt{2}$.*
- *Now suppose the input is $x[n] = 3x_1[n] = 6$, then the output is $y[n] = \sqrt{6} \neq 3\,y_1[n]$,*

so the system is nonlinear.

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
     - Techniques for the analysis of linear systems
     - Response of LTI systems to arbitrary inputs
     - Properties of convolution and the interconnection of LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Discrete-time LTI systems

Overview: $x[n] \rightarrow \boxed{\text{LTI } h[n]} \rightarrow y[n] = x[n] * h[n]$.

Linearity leads to the superposition property, which simplifies the analysis. Time-invariance then further simplifies.

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
     - Techniques for the analysis of linear systems
     - Response of LTI systems to arbitrary inputs
     - Properties of convolution and the interconnection of LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Techniques for the analysis of linear systems

1. Decompose input signal $x[n]$ into a weighted sum of elementary functions $x[n]_k$, *i.e.*,

$$x[n] = \sum_k c_k x_k[n]$$

2. Determine response of system to each elementary function (this should be easy from input-output relationship):

$$x_k[n] \xrightarrow{\mathcal{T}} y_k[n]$$

3. Apply superposition property:

$$x[n] = \sum_k c_k x_k[n] \xrightarrow{\mathcal{T}} y[n] = \sum_k c_k\, y_k[n]\,.$$

Techniques for the analysis of linear systems

1. Decompose input signal $x[n]$ into a weighted sum of elementary functions $x[n]_k$, *i.e.*,

$$x[n] = \sum_k c_k x_k[n]$$

2. Determine response of system to each elementary function (this should be easy from input-output relationship):

$$x_k[n] \xrightarrow{\mathcal{T}} y_k[n]$$

3. Apply superposition property:

$$x[n] = \sum_k c_k x_k[n] \xrightarrow{\mathcal{T}} y[n] = \sum_k c_k y_k[n] .$$

Techniques for the analysis of linear systems

1. Decompose input signal $x[n]$ into a weighted sum of elementary functions $x[n]_k$, *i.e.*,

$$x[n] = \sum_k c_k x_k[n]$$

2. Determine response of system to each elementary function (this should be easy from input-output relationship):

$$x_k[n] \xrightarrow{\mathcal{T}} y_k[n]$$

3. Apply superposition property:

$$x[n] = \sum_k c_k x_k[n] \xrightarrow{\mathcal{T}} y[n] = \sum_k c_k\, y_k[n]\,.$$

Elementary functions

Two particularly good choices for the elementary functions $x_k[n]$:

1. impulse functions $\delta[n-k]$
2. complex exponentials $e^{j\omega_k n}$.

Representation of discrete-time signal using impulses

It follows directly from the definition of $\delta[n]$ that

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

This is the sifting property of the unit impulse function.

Example

$$x[n] = \left\{\underline{8}, \pi, 0, \sqrt{7}\right\}$$

$$\Longrightarrow x[n] = 8\,\delta[n-0] + \pi\,\delta[n-1] + \sqrt{7}\,\delta[n-3]\,.$$

Representation of discrete-time signal using impulses

It follows directly from the definition of $\delta[n]$ that

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

This is the sifting property of the unit impulse function.

Example

$$x[n] = \left\{\underline{8}, \pi, 0, \sqrt{7}\right\}$$

$$\Longrightarrow x[n] = 8\,\delta[n-0] + \pi\,\delta[n-1] + \sqrt{7}\,\delta[n-3]\,.$$

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
     - Techniques for the analysis of linear systems
     - Response of LTI systems to arbitrary inputs
     - Properties of convolution and the interconnection of LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Response of LTI systems to arbitrary inputs

For an LTI system with impulse response $h[n]$, the response to an arbitrary input is given by the **convolution sum**:

$$\boxed{\underbrace{x[n] \xrightarrow{\mathcal{T}} y[n]}_{\text{LTI}} = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]\, h[k]}$$

$$\boxed{y[n] \overset{\triangle}{=} x[n] * h[n] = (x * h)[n].}$$

The convolution sum shows that with the impulse response $h[n]$, you can compute the output $y[n]$ for any input signal $x[n]$. Thus

An LTI system is characterized completely by its impulse response $h[n]$.

Response of LTI systems to arbitrary inputs

For an LTI system with impulse response $h[n]$, the response to an arbitrary input is given by the **convolution sum**:

$$\boxed{\underbrace{x[n] \xrightarrow{\mathcal{T}} y[n]}_{\text{LTI}} = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]\, h[k]}$$

$$\boxed{y[n] \triangleq x[n] * h[n] = (x * h)[n].}$$

The convolution sum shows that with the impulse response $h[n]$, you can compute the output $y[n]$ for any input signal $x[n]$. Thus

> An LTI system is characterized completely by its impulse response $h[n]$.

Computing convolution

Skill: *convolving*.

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

Recipe:

1. Fold: fold $h[k]$ about $k = 0$ to get $h[-k]$
2. Shift: shift $h[-k]$ by n to get $h[n-k]$
3. Multiply: $x[k]$ by $h[n-k]$ for every k
4. Sum: $y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

Repeat for all possible n; generally breaks in to a few intervals.

Mathematically, replace n with $n-k$ to complete step 1 and 2.

Computing convolution

Skill: *convolving*.

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

Recipe:

1. Fold: fold $h[k]$ about $k = 0$ to get $h[-k]$
2. Shift: shift $h[-k]$ by n to get $h[n-k]$
3. Multiply: $x[k]$ by $h[n-k]$ for every k
4. Sum: $y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

Repeat for all possible n; generally breaks in to a few intervals.

Mathematically, replace n with $n-k$ to complete step 1 and 2.

Computing convolution

Skill: *convolving*.

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

Recipe:

1. Fold: fold $h[k]$ about $k = 0$ to get $h[-k]$
2. Shift: shift $h[-k]$ by n to get $h[n-k]$
3. Multiply: $x[k]$ by $h[n-k]$ for every k
4. Sum: $y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

Repeat for all possible n; generally breaks in to a few intervals.

Mathematically, replace n with $n-k$ to complete step 1 and 2.

Computing convolution

Skill: *convolving*.

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

Recipe:

1. Fold: fold $h[k]$ about $k = 0$ to get $h[-k]$
2. Shift: shift $h[-k]$ by n to get $h[n-k]$
3. Multiply: $x[k]$ by $h[n-k]$ for every k
4. Sum: $y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

Repeat for all possible n; generally breaks in to a few intervals.

Mathematically, replace n with $n-k$ to complete step 1 and 2.

Computing convolution

Skill: *convolving*.

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

Recipe:

1. Fold: fold $h[k]$ about $k = 0$ to get $h[-k]$
2. Shift: shift $h[-k]$ by n to get $h[n-k]$
3. Multiply: $x[k]$ by $h[n-k]$ for every k
4. Sum: $y[n] = \sum_{k=-\infty}^{\infty} x[k]\,h[n-k]$

Repeat for all possible n; generally breaks in to a few intervals.

Mathematically, replace n with $n-k$ to complete step 1 and 2.

Computing convolution

Skill: *convolving*.

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

Recipe:

1. Fold: fold $h[k]$ about $k = 0$ to get $h[-k]$
2. Shift: shift $h[-k]$ by n to get $h[n-k]$
3. Multiply: $x[k]$ by $h[n-k]$ for every k
4. Sum: $y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

Repeat for all possible n; generally breaks in to a few intervals.

Mathematically, replace n with $n-k$ to complete step 1 and 2.

Computing convolution

Skill: *convolving*.

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

Recipe:

1. Fold: fold $h[k]$ about $k = 0$ to get $h[-k]$
2. Shift: shift $h[-k]$ by n to get $h[n-k]$
3. Multiply: $x[k]$ by $h[n-k]$ for every k
4. Sum: $y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$

Repeat for all possible n; generally breaks in to a few intervals.

Mathematically, replace n with $n-k$ to complete step 1 and 2.

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
     - Techniques for the analysis of linear systems
     - Response of LTI systems to arbitrary inputs
     - Properties of convolution and the interconnection of LTI systems
   - Properties of LTI systems in terms of the impulse response
   - DT systems described by difference equations

Properties of convolution (1)

Skill: *Use properties to simplify LTI systems.*

Property

1. *Time-shift*
   $x[n] * h[n] = y[n] \Longrightarrow x[n-n_1] * h[n-n_2] = y[n-n_1-n_2]$
2. *Commutative law* $x[n] * h[n] = h[n] * x[n]$
3. *Associative law* $(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$
4. *Distributive law*
   $x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n])$

Properties of convolution (1)

Skill: *Use properties to simplify LTI systems.*

Property

1. *Time-shift*
   $$x[n] * h[n] = y[n] \Longrightarrow x[n-n_1] * h[n-n_2] = y[n-n_1-n_2]$$
2. *Commutative law* $x[n] * h[n] = h[n] * x[n]$
3. *Associative law* $(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$
4. *Distributive law*
   $$x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n])$$

Properties of convolution (1)

Skill: *Use properties to simplify LTI systems.*

Property

1. *Time-shift*
   $x[n] * h[n] = y[n] \Longrightarrow x[n-n_1] * h[n-n_2] = y[n-n_1-n_2]$
2. *Commutative law* $x[n] * h[n] = h[n] * x[n]$
3. *Associative law* $(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$
4. *Distributive law*
   $x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n])$

Properties of convolution (1)

Skill: *Use properties to simplify LTI systems.*

Property

1. *Time-shift*
   $x[n] * h[n] = y[n] \Longrightarrow x[n-n_1] * h[n-n_2] = y[n-n_1-n_2]$
2. *Commutative law* $x[n] * h[n] = h[n] * x[n]$
3. *Associative law* $(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$
4. *Distributive law*
   $x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n])$

Properties of convolution (2)

Property

convolution with impulses

1. *Time shift / delay:*

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

2. *Identity:*

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

3. *Cascade of time shifts:*

$$\delta[n-n_1] * \delta[n-n_2] = \delta[n-n_1-n_2]$$

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
     - T-1 Causal LTI systems
     - A-3 Invertibility of LTI systems
     - A-2 Stability of LTI systems
     - T-2 Memory of LTI systems
   - DT systems described by difference equations

LTI system properties via impulse response

Since an LTI system is completely characterized by its impulse response, we should be able to express the remaining four properties in terms of $h[n]$.

- T-1 causality
- T-2 memory
- A-2 stability
- A-3 invertibility

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
     - T-1 Causal LTI systems
     - A-3 Invertibility of LTI systems
     - A-2 Stability of LTI systems
     - T-2 Memory of LTI systems
   - DT systems described by difference equations

T-1 Causal LTI systems (1)

Recall a system is **causal** iff output $y[n]$ depends only on present and past values of input.

For an LTI system with impulse response $h[n]$:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} h[k]\,x[n-k] \\ &= \sum_{k=0}^{\infty} h[k]\,x[n-k] + \sum_{k=-\infty}^{-1} h[k]\,x[n-k]. \end{aligned}$$

1. The first term depends on present and past input samples $x[n], x[n-1], \ldots$
2. The second term depends on future input samples $x[n+1], x[n+1], \ldots.$

T-1 Causal LTI systems (1)

Recall a system is **causal** iff output $y[n]$ depends only on present and past values of input.

For an LTI system with impulse response $h[n]$:

$$
\begin{aligned}
y[n] &= \sum_{k=-\infty}^{\infty} h[k]\,x[n-k] \\
&= \sum_{k=0}^{\infty} h[k]\,x[n-k] + \sum_{k=-\infty}^{-1} h[k]\,x[n-k].
\end{aligned}
$$

1. The first term depends on present and past input samples $x[n], x[n-1], \ldots$
2. The second term depends on future input samples $x[n+1], x[n+1], \ldots.$

T-1 Causal LTI systems (1)

Recall a system is **causal** iff output $y[n]$ depends only on present and past values of input.

For an LTI system with impulse response $h[n]$:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} h[k]\,x[n-k] \\ &= \sum_{k=0}^{\infty} h[k]\,x[n-k] + \sum_{k=-\infty}^{-1} h[k]\,x[n-k]. \end{aligned}$$

1. The first term depends on present and past input samples $x[n], x[n-1], \ldots$
2. The second term depends on future input samples $x[n+1], x[n+1], \ldots$.

T-1 Causal LTI systems (1)

Recall a system is **causal** iff output $y[n]$ depends only on present and past values of input.

For an LTI system with impulse response $h[n]$:

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} h[k]\,x[n-k] \\ &= \sum_{k=0}^{\infty} h[k]\,x[n-k] + \sum_{k=-\infty}^{-1} h[k]\,x[n-k]. \end{aligned}$$

1. The first term depends on present and past input samples $x[n], x[n-1], \ldots$
2. The second term depends on future input samples $x[n+1], x[n+1], \ldots$.

T-1 Causal LTI systems (2)

Thus the system is causal iff the impulse response terms corresponding to the second sum are zero. These terms are $h[-1], h[-2], \ldots$.

Definition

An LTI system is **causal** iff its impulse response $h[n] = 0$ for all $n < 0$.

In the causal case the convolution summation simplifies slightly since we can drop the right sum above:

$$y[n] = \sum_{k=0}^{\infty} h[k]\, x[n-k] = \sum_{k=-\infty}^{n} x[k']\, h\big[n-k'\big] \ \text{(using } k' = n-k\text{)}.$$

T-1 Causal LTI systems (2)

Thus the system is causal iff the impulse response terms corresponding to the second sum are zero. These terms are $h[-1], h[-2], \ldots$.

Definition

An LTI system is **causal** iff its impulse response $h[n] = 0$ for all $n < 0$.

In the causal case the convolution summation simplifies slightly since we can drop the right sum above:

$$y[n] = \sum_{k=0}^{\infty} h[k]\, x[n-k] = \sum_{k=-\infty}^{n} x[k']\, h\big[n-k'\big] \text{ (using } k' = n-k\text{).}$$

T-1 Causal LTI systems (2)

Thus the system is causal iff the impulse response terms corresponding to the second sum are zero. These terms are $h[-1], h[-2], \ldots$.

Definition

An LTI system is **causal** iff its impulse response $h[n] = 0$ for all $n < 0$.

In the causal case the convolution summation simplifies slightly since we can drop the right sum above:

$$y[n] = \sum_{k=0}^{\infty} h[k]\, x[n-k] = \sum_{k=-\infty}^{n} x[k']\, h\big[n-k'\big] \text{ (using } k' = n-k).$$

T-1 Causal LTI systems (3)

Example

Is the LTI system with $h[n] = u[n - n_0 - 5]$ causal?

T-1 Causal LTI systems (3)

Example

Is the LTI system with $h[n] = u[n - n_0 - 5]$ causal?

Solution

It is causal only if $h[n] = 0$ *for all* $n < 0$. $\Rightarrow n_0 + 5 \geq 0$.

T-1 Causal LTI systems (4)

Definition

A **causal sequence** is a sequence $x[n]$ which is zero for all $n < 0$.

If the input to a causal LTI system is a causal sequence, then the output is simply

$$y[n] = \begin{cases} 0, & n < 0 \\ \sum_{k=0}^{n} h[k]\, x[n-k] = \sum_{k=0}^{n} x[k]\, h[n-k], & n \geq 0. \end{cases}$$

The above sum is precisely what is computed by MATLAB's `conv` function, for finite-length $x[n]$ and $h[n]$.

T-1 Causal LTI systems (4)

Definition

A **causal sequence** is a sequence $x[n]$ which is zero for all $n < 0$.

If the input to a causal LTI system is a causal sequence, then the output is simply

$$y[n] = \begin{cases} 0, & n < 0 \\ \sum_{k=0}^{n} h[k]\, x[n-k] = \sum_{k=0}^{n} x[k]\, h[n-k], & n \geq 0. \end{cases}$$

The above sum is precisely what is computed by MATLAB's `conv` function, for finite-length $x[n]$ and $h[n]$.

T-1 Causal LTI systems (4)

Definition

A **causal sequence** is a sequence $x[n]$ which is zero for all $n < 0$.

If the input to a causal LTI system is a causal sequence, then the output is simply

$$y[n] = \begin{cases} 0, & n < 0 \\ \sum_{k=0}^{n} h[k]\,x[n-k] = \sum_{k=0}^{n} x[k]\,h[n-k], & n \geq 0. \end{cases}$$

The above sum is precisely what is computed by MATLAB's `conv` function, for finite-length $x[n]$ and $h[n]$.

Outline

Invertibility of LTI systems

Fact: if an LTI system $\mathcal{T}$ is invertible, then its corresponding inverse system is also LTI. Thus the inverse system has an impulse response, say, $h_{\text{inv}}[n]$ such that

$$x[n] \rightarrow \boxed{h[n]} \rightarrow \boxed{h_{\text{inv}}[n]} \rightarrow x[n].$$

In other words: $\boxed{h[n] * h_{\text{inv}}[n] = \delta[n]}$.

Invertibility of LTI systems

Fact: if an LTI system $\mathcal{T}$ is invertible, then its corresponding inverse system is also LTI. Thus the inverse system has an impulse response, say, $h_{\text{inv}}[n]$ such that

$$x[n] \to \boxed{h[n]} \to \boxed{h_{\text{inv}}[n]} \to x[n].$$

In other words: $\boxed{h[n] * h_{\text{inv}}[n] = \delta[n]\,.}$

Outline

Stability of LTI systems (1)

Recall $y[n] = \sum_{k=-\infty}^{\infty} h[k]\, x[n-k]$ so by the triangle inequality

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

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

if $|x[n]| \leq M_x \; \forall n$.

Stability of LTI systems (2)

Thus, for an LTI system to be BIBO stable, it is sufficient that its impulse response be absolutely summable, *i.e.*,

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

since in that case, if $x[n]$ is bounded, so will be $y[n]$.

Stability of LTI systems (2)

Definition

An LTI system is **BIBO stable** iff its impulse response is absolutely summable, *i.e.*

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

Example

Accumulator: $y[n] = y[n-1] + x[n]$.
What is impulse response? Is it stable?

Stability of LTI systems (2)

Definition

An LTI system is **BIBO stable** iff its impulse response is absolutely summable, *i.e.*

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

Example

Accumulator: $y[n] = y[n-1] + x[n]$.
What is impulse response? Is it stable?

Solution (1)

Solution

Let $x[n] = \delta[n]$, *then* $y[n] = u[n]$. *So* $h[n] = u[n]$.
No: $\sum_{n=-\infty}^{\infty} |h[n]| = \infty$, *so unstable.*

Outline

1. Discrete-time signals and systems
   - Discrete-time signals
   - Discrete-time systems
   - Discrete-time LTI systems
   - Properties of LTI systems in terms of the impulse response
     - T-1 Causal LTI systems
     - A-3 Invertibility of LTI systems
     - A-2 Stability of LTI systems
     - T-2 Memory of LTI systems
   - DT systems described by difference equations

T-2 Memory (1)

Recall a system is **static** or **memoryless** if the output $y[n]$ depends only on the current input $x[n]$, not on previous or future values of the input signal.

For an LTI system with impulse response $h[n]$:

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

the only way this can be true is if

$$h[n] = 0, \quad \text{for} \quad n \neq 0$$

T-2 Memory (1)

Recall a system is **static** or **memoryless** if the output $y[n]$ depends only on the current input $x[n]$, not on previous or future values of the input signal.

For an LTI system with impulse response $h[n]$:

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

the only way this can be true is if

$$h[n] = 0, \quad \text{for} \quad n \neq 0$$

T-2 Memory (2)

Definition

An LTI system is **memoryless** iff its impulse response is $h[n] = a\,\delta[n]$. Otherwise the system is **dynamic** (has memory).

In this case the response is
$y[n] = x[n] * h[n] = x[n] * a\,\delta[n] = ax[n]$.

T-2 Memory (2)

Definition

An LTI system is **memoryless** iff its impulse response is $h[n] = a\,\delta[n]$. Otherwise the system is **dynamic** (has memory).

In this case the response is
$y[n] = x[n] * h[n] = x[n] * a\,\delta[n] = ax[n]$.

T-2 Memory (3)

There are two classes of dynamic systems.

1. **finite impulse response** or **FIR**: only a finite number of $h[n]$ are nonzero.
2. **infinite impulse response** or **IIR**: an infinite number of $h[n]$ are nonzero.

Outline

1. Discrete-time signals and systems
    - Discrete-time signals
    - Discrete-time systems
    - Discrete-time LTI systems
    - Properties of LTI systems in terms of the impulse response
    - DT systems described by difference equations

Convolution summation

- The convolution summation

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

  looks fine on paper, but if the impulse response is IIR (infinite impulse response), it cannot be implemented directly with that formula since there would be an infinite number of adds and multiplies!
- For an arbitrary impulse response $h[n]$, there may not exist an implementation with a finite number of flops. Fortunately however, there is a broad class of interesting and useful IIR systems that one can implement using **difference equations**.

Convolution summation

- The convolution summation

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

  looks fine on paper, but if the impulse response is IIR (infinite impulse response), it cannot be implemented directly with that formula since there would be an infinite number of adds and multiplies!
- For an arbitrary impulse response $h[n]$, there may not exist an implementation with a finite number of flops. Fortunately however, there is a broad class of interesting and useful IIR systems that one can implement using **difference equations**.

Convolution summation: example(1)

Example

Consider an LTI system with impulse response $h[n] = a^n\, u[n]$. It is FIR or IIR?

Convolution summation: example(1)

Example

Consider an LTI system with impulse response $h[n] = a^n u[n]$. It is FIR or IIR?

IIR.

Convolution summation: example(1)

Example

Consider an LTI system with impulse response $h[n] = a^n\, u[n]$. It is FIR or IIR?

IIR.

Naive approach to implementation would just truncate the infinite sum to m terms (may need large m):

$$
\begin{aligned}
y[n] &= \sum_{k=-\infty}^{\infty} h[k]\, x[n-k] = \sum_{k=0}^{\infty} a^k x[n-k] \\
&\approx x[n] + ax[n-1] + a^2 x[n-2] + \cdots + a^m x[n-m].
\end{aligned}
$$

Convolution summation: example (2)

Efficient approach uses mathematical properties:

$$y[n] = \sum_{k=0}^{\infty} a^k x[n-k] = x[n] + \sum_{k=1}^{\infty} a^k x[n-k]$$

$$= x[n] + \sum_{l=0}^{\infty} a^{l+1} x[n-(l+1)] \quad (l = k-1)$$

$$= x[n] + a\sum_{l=0}^{\infty} a^l x[(n-1)-l]$$

$$= x[n] + ay[n-1].$$

This simple manipulation led to a **recursive** expression for $y[n]$, and one that is easily implemented using just one delay, one multiply, and one add. And, it is exact, unlike truncation!

Convolution summation: example (2)

Efficient approach uses mathematical properties:

$$y[n] = \sum_{k=0}^{\infty} a^k x[n-k] = x[n] + \sum_{k=1}^{\infty} a^k x[n-k]$$

$$= x[n] + \sum_{l=0}^{\infty} a^{l+1} x[n-(l+1)] \quad (l = k-1)$$

$$= x[n] + a \sum_{l=0}^{\infty} a^l x[(n-1)-l]$$

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

This simple manipulation led to a **recursive** expression for $y[n]$, and one that is easily implemented using just one delay, one multiply, and one add. And, it is exact, unlike truncation!

Convolution summation: example (2)

Efficient approach uses mathematical properties:

$$y[n] = \sum_{k=0}^{\infty} a^k x[n-k] = x[n] + \sum_{k=1}^{\infty} a^k x[n-k]$$

$$= x[n] + \sum_{l=0}^{\infty} a^{l+1} x[n-(l+1)] \quad (l = k-1)$$

$$= x[n] + a \sum_{l=0}^{\infty} a^l x[(n-1)-l]$$

$$= x[n] + ay[n-1].$$

This simple manipulation led to a **recursive** expression for $y[n]$, and one that is easily implemented using just one delay, one multiply, and one add. And, it is exact, unlike truncation!

Convolution summation: example (2)

Efficient approach uses mathematical properties:

$$y[n] = \sum_{k=0}^{\infty} a^k x[n-k] = x[n] + \sum_{k=1}^{\infty} a^k x[n-k]$$

$$= x[n] + \sum_{l=0}^{\infty} a^{l+1} x[n-(l+1)] \quad (l = k-1)$$

$$= x[n] + a \sum_{l=0}^{\infty} a^l x[(n-1)-l]$$

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

This simple manipulation led to a **recursive** expression for $y[n]$, and one that is easily implemented using just one delay, one multiply, and one add. And, it is exact, unlike truncation!

Convolution summation: example (2)

Efficient approach uses mathematical properties:

$$y[n] = \sum_{k=0}^{\infty} a^k x[n-k] = x[n] + \sum_{k=1}^{\infty} a^k x[n-k]$$

$$= x[n] + \sum_{l=0}^{\infty} a^{l+1} x[n-(l+1)] \quad (l = k-1)$$

$$= x[n] + a \sum_{l=0}^{\infty} a^l x[(n-1)-l]$$

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

This simple manipulation led to a **recursive** expression for $y[n]$, and one that is easily implemented using just one delay, one multiply, and one add. And, it is exact, unlike truncation!

Constant-coefficient difference equations

A realizable system must only require a finite number of operations. A very general form is those systems that can be described by **linear constant-coefficient difference equations**:

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

Such systems are the discrete-time analog of **linear constant-coefficient differential equations** for continuous-time systems.

$$\boxed{\sum_{k=0}^{N} a_k \frac{d^k}{dt^k} y(t) = \sum_{k=0}^{M} b_k \frac{d^k}{dt^k} x(t).}$$

Such systems are **linear** and **time-invariant** and **causal** with **initial rest**.

Constant-coefficient difference equations

A realizable system must only require a finite number of operations. A very general form is those systems that can be described by **linear constant-coefficient difference equations**:

$$\boxed{\sum_{k=0}^{N} a_k\, y[n-k] = \sum_{k=0}^{M} b_k x[n-k].}$$

Such systems are the discrete-time analog of **linear constant-coefficient differential equations** for continuous-time systems.

$$\boxed{\sum_{k=0}^{N} a_k \frac{d^k}{dt^k} y(t) = \sum_{k=0}^{M} b_k \frac{d^k}{dt^k} x(t).}$$

Such systems are **linear** and **time-invariant** and **causal** with **initial rest**.

Constant-coefficient difference equations

A realizable system must only require a finite number of operations. A very general form is those systems that can be described by **linear constant-coefficient difference equations**:

$$\boxed{\sum_{k=0}^{N} a_k\, y[n-k] = \sum_{k=0}^{M} b_k x[n-k].}$$

Such systems are the discrete-time analog of **linear constant-coefficient differential equations** for continuous-time systems.

$$\boxed{\sum_{k=0}^{N} a_k \frac{d^k}{dt^k} y(t) = \sum_{k=0}^{M} b_k \frac{d^k}{dt^k} x(t).}$$

Such systems are **linear** and **time-invariant** and **causal** with **initial rest**.

Categories

1. If $N \geq 1$ **recursive** system
   - the current output $y[n]$ depends on past outputs $y[n-1], \ldots, y[n-N]$,
   - the impulse response is IIR. (Usually we use z-transform to find $h[n]$.)
2. If $N = 0$ **nonrecursive** system
   - the output $y[n]$ depends only on the current input sample and on M past input samples
   - the impulse response is FIR and is given by

$$h[n] = \{\underline{b_0}, b_1, \ldots, b_M\} = \sum_{k=0}^{M} b_k\, \delta[n-k]\,.$$

Categories

1. If $N \geq 1$ **recursive** system
   - the current output $y[n]$ depends on past outputs $y[n-1], \ldots, y[n-N]$,
   - the impulse response is IIR. (Usually we use z-transform to find $h[n]$.)
2. If $N = 0$ **nonrecursive** system
   - the output $y[n]$ depends only on the current input sample and on M past input samples
   - the impulse response is FIR and is given by

$$h[n] = \{\underline{b_0}, b_1, \ldots, b_M\} = \sum_{k=0}^{M} b_k \, \delta[n-k] \, .$$

Solution of linear constant-coefficient difference equations

The **homogeneous solution** or **zero-input solution** is of the form of linear combinations of $\{\lambda^n,\ n\lambda^n,\ \ldots n^N\lambda^n\}$, analogous to $\{\mathrm{e}^{\lambda t},\ t\,\mathrm{e}^{\lambda t},\ \ldots t^N\,\mathrm{e}^{\lambda t}\}$ for differential equations, where the λ's are roots of the **characteristic polynomial** $\sum_{k=0}^{N} a_k\lambda^{n-k}$.

Impulse response of an LTI recursive system

Impulse response of an LTI recursive system

1. Brute force: let $x[n] = \delta[n]$ and execute recursion to find $y[n] = h[n]$.
2. Easier to study using z-transform. Later...