Digital Signal Processing
Course Introduction

D. Richard Brown III

Meeting 1

First Lecture: Major Topics

1. Administrative details:
   - Syllabus.
   - Textbook and companion website.
   - Course web page.
   - Piazza.
   - Academic honesty policy.
   - Students with disabilities statement.
   - Weekly quizzes.
   - Projects.
   - Comprehensive final exam.
2. Course introduction.
3. Basics (Chapter 2.1 – 2.5)
   - Discrete-time signals
   - Discrete-time systems
   - Impulse response
   - Linear constant coefficient difference equations

Why No Homework?

I have found almost no value in collecting and grading homework.

- Homework grades are a poor assessment of student understanding. In my experience, homework grades are almost entirely uncorrelated to final course grades.
- Academic honesty problems.
- Student focus is often on handing something in rather than really learning the material.

You still need to do problems to understand this material. I will provide several suggested problems and solutions each week for you to use for practice. You are encouraged to collaborate!

Why Weekly Quizzes?

The traditional midterm/final format encourages "cramming". There is a lot of research that shows cramming, while an efficient use of time for studying for exams, leads to very poor long-term retention of the material.

High-stakes midterm/final exams penalize students who have one bad day.

Weekly quizzes: Immediate feedback for instructor. Students can have a bad day and still do well in the course.

We will still have a low-stakes comprehensive final exam to encourage reflection and review of the full scope of the course.

Why No Lectures?

It is very difficult to pay attention to a three-hour evening lecture.

Recent trends "flip teaching" and "mobile learning":

- Watch lecture materials outside of the classroom on your own schedule
- Do assigned reading
- Work on suggested problems
- Collaborate with other students (in person or via Piazza) and reinforce your understanding of the material
- Classroom meeting time can be used for more hands-on discussion

Bottom line: I can spend class time interacting with you and helping you understand the material, instead of lecturing.

Why Piazza?

I will answer questions on Piazza. Those questions/answers will be visible to everyone.

You can answer questions (or contribute to the discussion) on Piazza too. Helping others understand the material is an excellent way to reinforce your own understanding.

I can mark questions/answers as "good questions" and "good answers". I can also mark duplicate questions. Try not to post duplicate questions.

If you have a question, chances are someone else has that question too. In fact, your question might already be answered on Piazza.

Course Introduction: What is Signal Processing?

"Signal processing deals with the representation, transformation, and manipulation of signals and the information the signals contain" (O&S p.2)

Lots of applications:

- Audio signals
- Communications
- Biological signals (EMG, ECG, ...)
- Radar/sonar
- Image/video processing
- Power metering
- Structural health monitoring
- ... (see textbook for even more)

Digital Signal Processing: Historical Context

The idea of processing signals digitally began in the 1950s with the availability of computers.

Early applications typically involved processing pre-recorded signals:

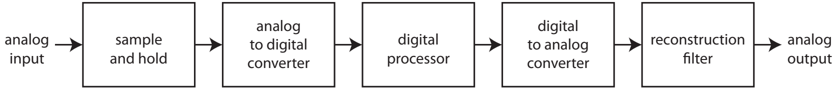
- Simulation of analog circuits
- Perceptual quality studies, e.g., vocoder circuits
- Geophysical exploration (low frequency seismic signals)

The idea of doing DSP in real time became realistic in the late in 1960s with the development of the Fast Fourier Transform (FFT).

Specialized DSP chips began appearing in the mid-1980s.

Digital Signal Processing: Advantages

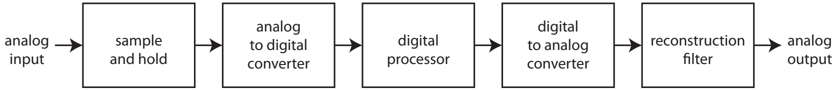
A typical DSP system:



Some advantages:

1. Digital storage for offline processing
2. Easy to reconfigure/reprogram (more flexible)
3. Less sensitive to component values, temperature, and aging.
4. Typically easier to manufacture (less need for calibration).
5. Typically more accurate (accuracy can be increased by increasing word length)
6. Typically higher dynamic range
7. Wider range of applications: signal multiplexing, adaptive filters, ...

Digital Signal Processing: Disadvantages



Some disadvantages:

1. Increased system complexity
2. Potential for software bugs (more testing required)
3. Typically higher power consumption than analog circuits
4. Limited frequency range
5. Decreasing ADC/DAC accuracy as sampling frequency is increased
6. ADC and DAC delay
7. Finite precision effects
8. Analog circuits must be used for some applications (like what?)

Some Notation

$$
\begin{aligned}
\mathbb{R} &= \text{the set of real numbers } (-\infty, \infty) \\
\mathbb{Z} &= \text{the set of integers } \{\dots, -1, 0, 1, \dots\} \\
j &= \text{unit imaginary number } \sqrt{-1} \\
\mathbb{C} &= \text{the set of complex numbers } (-\infty, \infty) \times j(-\infty, \infty) \\
t &= \text{continuous time parameter} \in \mathbb{R} \\
x(t) &= \text{continuous-time signal } \mathbb{R} \mapsto \mathbb{R} \text{ or } \mathbb{R} \mapsto \mathbb{C} \\
n &= \text{discrete time parameter} \in \mathbb{Z} \\
x[n] &= \text{discrete-time signal } \mathbb{Z} \mapsto \mathbb{R} \text{ or } \mathbb{Z} \mapsto \mathbb{C} \\
T &= \text{sampling period } \in \mathbb{R} \\
f_s &= \text{sampling frequency } \in \mathbb{R} \\
\Omega &= \text{frequency of continuous-time signal } \in \mathbb{R} \\
\omega &= \text{frequency of discrete-time signal} \in \mathbb{R}
\end{aligned}
$$

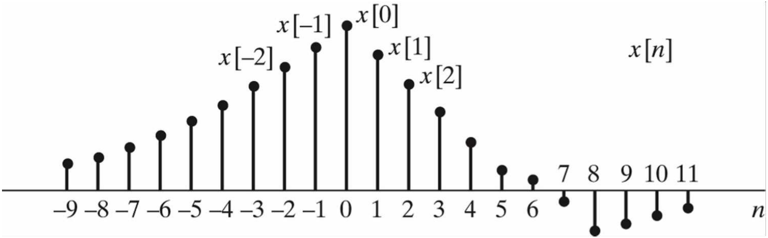
Representations of a Discrete-Time Signals

Discrete-time signals are represented mathematically as sequences of (usually real) numbers. The sequence x is formally written as

$$x = \{x[n]\} \qquad n \in \mathbb{Z}$$

Note that $x[n]$ is not defined for $n \notin \mathbb{Z}$.

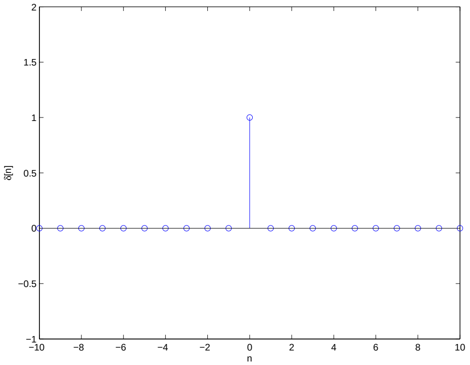
Example:



Unit Sample Sequence (Discrete-time Impulse Function)

An important discrete-time signal is the unit sample sequence (also called the discrete-time impulse function):

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$$



Expressing Sequences as Sums of Unit Sample Sequences

All discrete time signals can be written as the sum of weighted and delayed unit sample sequences. Example: Suppose

$$x[n] = \begin{cases} 1 & n = 0 \\ 2 & n = 1 \\ -1 & n = 2 \\ 0 & \text{otherwise} \end{cases}$$

This signal can be written as

$$x[n] = \delta[n] + 2\delta[n-1] - \delta[n-2]$$

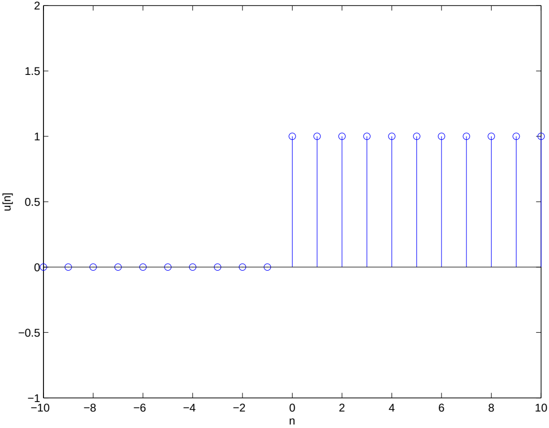
In general, any discrete-time sequence can be expressed as

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

for all $n \in \mathbb{Z}$.

Unit Step Sequence

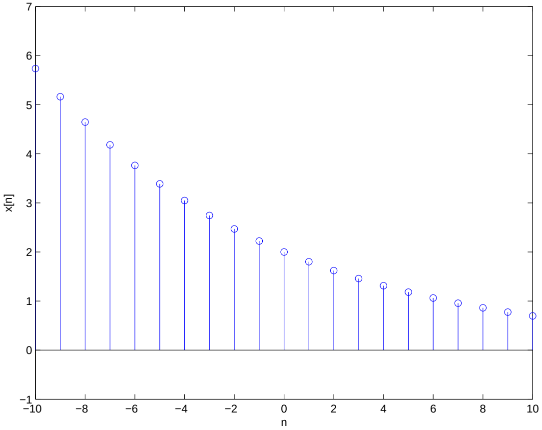
$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



Real Exponential Sequence

$$x[n] = A\alpha^n$$

Example with $A = 2$ and $\alpha = 0.9$:



Complex Exponential Sequence

Suppose

$$x[n] = A\alpha^n$$

with $A = |A|e^{j\phi}$ and $\alpha = |\alpha|e^{j\omega_0}$. Then

$$\begin{aligned} x[n] &= |A|e^{j\phi}\left(|\alpha|e^{j\omega_0}\right)^n \\ &= |A||\alpha|^n e^{j(n\omega_0+\phi)} \\ &= |A||\alpha|^n \cos(n\omega_0 + \phi) + j|A||\alpha|^n \sin(n\omega_0 + \phi) \end{aligned}$$

Non-Uniqueness of Discrete-Time Sinusoidal Signals

Suppose

$$x_1[n] = \cos(\omega_0 n + \phi)$$
$$x_2[n] = \cos((\omega_0 + r2\pi)n + \phi).$$

These two sequences are identical for any integer $r \in \mathbb{Z}$.

Consqeuence: When discussing complex exponential signals of the form

$$x[n] = Ae^{j\omega_0 n}$$

or sinusoidal signals of the form

$$x[n] = \cos(\omega_0 n + \phi)$$

we only need to consider frequencies over an interval of length 2π. Typically we choose $-\pi < \omega_0 \leq \pi$.

Discrete-Time Periodicity

A periodic discrete-time signal with integer period $N \in \{1, 2, \dots\}$ must satisfy

$$x[n] = x[n+N]$$

for all $n \in \mathbb{Z}$.

Consider the discrete-time sequence $x[n] = \cos(\omega_0 n + \phi)$. For this signal to be periodic, it must satisfy

$$\begin{aligned} \cos(\omega_0 n + \phi) &= \cos(\omega_0 (n+N) + \phi) \\ \Leftrightarrow \cos(\omega_0 n + \phi) &= \cos(\omega_0 n + \omega_0 N + \phi) \\ \Leftrightarrow \omega_0 N &= r 2\pi \end{aligned}$$

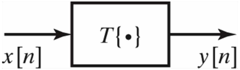
for some $r \in \mathbb{Z}$. Hence discrete-time sinusoidal signals are periodic only if

$$\omega_0 = \frac{r}{N} 2\pi$$

for some integer r and N.

Discrete-Time Systems

We typically denote the input of a discrete-time system as $x[n]$ and the output as $y[n]$.



The system T computes the outputs as a function of the inputs. In general, the output at time n may depend on several samples of the input sequence x.

Example: Moving average

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$$
$$= \frac{1}{N}\left(x[n] + x[n-1] + \cdots + x[n-N+1]\right)$$

Memoryless Discrete-Time Systems

Definition

A system T is memoryless if the output $y[n]$ at every value of n depends only on the output $x[n]$ at the same value of n.

Examples:

$$y[n] = 3x[n]$$
$$y[n] = x^2[n]$$

Is the moving average system memoryless?

Linear Discrete-Time Systems

Given $y_1[n] = T\{x_1[n]\}$ and $y_2[n] = T\{x_2[n]\}$.

Definition

A system T is linear if satisfies the additivity and homogeneity properties:

$$T\{x_1[n] + x_2[n]\} = T\{x_1[n]\} + T\{x_2[n]\} = y_1[n] + y_2[n] \text{ (additivity)}$$

for all sequences $x_1[n]$ and $x_2[n]$ and

$$T\{ax[n]\} = aT\{x[n]\} = ay[n] \text{ (homegeneity)}$$

for all sequences $x[n]$ and any constant a.

- Is the moving average $y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$ linear?
- Is $y[n] = x[n-1]$ linear?
- Is $y[n] = 3x[n]$ linear?
- Is $y[n] = x^2[n]$ linear?

Time-Invariant Discrete-Time Systems

Definition

A system T is time-invariant if a time shift of the input sequence causes a corresponding time shift in the output sequence. In other words, if $y[n] = T\{x[n]\}$ then $T\{x[n-n_0]\}$ must produce $y[n-n_0]$, where n_0 is an arbitrary integer.

Examples:

- Is the moving average $y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$ time-invariant?
- Is $y[n] = x[n-1]$ time-invariant?
- Is $y[n] = 3x[n]$ time-invariant?
- Is $y[n] = x^2[n]$ time-invariant?
- Is $y[n] = x[2n]$ time-invariant?

Causal Discrete-Time Systems

Definition

A system T is causal if, for every $n_0 \in \mathbb{Z}$, the output sequence value at the index $n = n_0$ depends only on the input sequence values for $n \leq n_0$.

Examples:

- Is the moving average $y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$ causal?
- Is $y[n] = x[n-1]$ causal? What about $y[n] = x[n+1]$?
- Is $y[n] = 3x[n]$ causal?
- Is $y[n] = x^2[n]$ causal?
- Is $y[n] = x[2n]$ causal?

Sequence Boundedness and Summability

Definition

A sequence $\{x[n]\}$ is said to be bounded if there exists some finite $B_x < \infty$ such that

$$|x[n]| \le B_x \text{ for all } n.$$

Definition

A sequence $\{x[n]\}$ is said to be absolutely summable if

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty.$$

Definition

A sequence $\{x[n]\}$ is said to be square-summable if

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

Such a sequence has finite energy and is an energy signal.

Stable Discrete-Time Systems

Definition

A system T is bounded-input bounded-output (BIBO) stable if every bounded input sequence produces a bounded output sequence.

Examples:

- Is the moving average $y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$ stable?
- Is $y[n] = x[n-1]$ stable?
- Is $y[n] = 3x[n]$ stable?
- Is $y[n] = x^2[n]$ stable?
- Is $y[n] = x[2n]$ stable?
- Is $y[n] = \exp(x[n])$ stable?
- Is $y[n] = \log(|x[n]|)$ stable?

Input-Output Description: Capabilities and Limitations

Example (causal discrete-time system):

$$y[k] \;=\; f(y[k-1], y[k-2], \ldots, x[k], x[k-1], \ldots)$$

+ Can describe memoryless or dynamic systems.
+ Can describe causal or non-causal systems.
+ Can describe linear or non-linear systems.
+ Can describe time-invariant or time-varying systems.
+ Can describe relaxed or non-relaxed systems (non-zero initial conditions).

– No explicit access to internal behavior of systems.
– Difficult to analyze directly.

Impulse Response Description: Capabilities and Limitations

Definition

The **impulse response** of a system is the output of the system given an input $x[n] = \delta[n]$ assuming relaxed initial conditions. We denote the impulse response of the system T as $h[n] : \mathbb{Z} \mapsto \mathbb{R}$.

Example: If $y[n] = x[n] + 0.5x[n-1]$ then $h[n] = \delta[n] + 0.5\delta[n-1]$.

+ Can describe memoryless and dynamic systems.
+ Can describe causal and non-causal systems.
– Nonlinear systems have an impulse response, but it isn't useful.
+ Can describe time-invariant and time-varying systems.
– No explicit access to internal behavior of system.
– Implicitly assumes that system is relaxed. Can't describe systems with non-zero initial conditions.

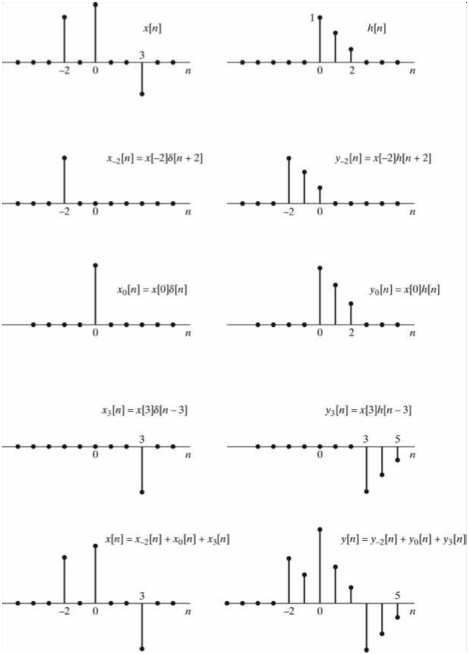
LTI Discrete-Time Systems

The focus of this course is on an important class of systems: **linear time-invariant** (LTI) systems.

What is the response of an LTI system T to a general input $x[n]$?

$$\begin{aligned}
y[n] &= T\{x[n]\} \\
&= T\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\right\} \\
&\overset{\text{linearity}}{=} \sum_{k=-\infty}^{\infty} x[k]T\left\{\delta[n-k]\right\} \\
&\overset{\text{time-invariance}}{=} \sum_{k=-\infty}^{\infty} x[k]h[n-k]
\end{aligned}$$

If a system T is LTI, its relaxed behavior is **fully characterized** by its impulse response $\{h[n]\}$.



x[n]
h[n]
x-2[n] = x[-2]δ[n + 2]
y-2[n] = x[-2]h[n + 2]
x0[n] = x[0]δ[n]
y0[n] = x[0]h[n]
x3[n] = x[3]δ[n - 3]
y3[n] = x[3]h[n - 3]
x[n] = x-2[n] + x0[n] + x3[n]
y[n] = y-2[n] + y0[n] + y3[n]

Discrete-Time Convolution

The output of a (relaxed) LTI system to the input sequence $x[n]$ can be computed by convolving $x[n]$ with the impulse response $h[n]$. To perform convolution, you must understand:

- Time shifting
- Time reversal
- Multiplication and addition

Notation:

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

See the Matlab command `conv` and examples in your textbook. Also see the companion website for some interactive examples.

Impulse Response of an LTV System

If a system T is LTV, it may have a different impulse response if the impulse is applied to the input at different times. Example:

$$y[n] = nx[n]$$

- Applying an input $x[n] = \delta[n]$ results in what output?
- Applying an input $x[n] = \delta[n-1]$ results in what output?

If we denote $h[n,k]$ as the response of the system T at time n to an impulse at time k, we can derive the convolution sum for an LTV system as

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

I don't know of any Matlab function that will compute this directly. Note $h[n,k] = h[n-k]$ if the system is LTI.

Convolution Matrix of an LTI System (1 of 2)

Suppose you want to convolve two finite-length sequences: $\{a[0], \dots, a[M-1]\}$ and $\{b[0], \dots, b[N-1]\}$. The result $\{c[n]\} = \{a[n]\} * \{b[n]\}$ will have $M+N-1$ elements and can be computed as

$$
\begin{aligned}
c[0] &= a[0]b[0] \\
c[1] &= a[1]b[0] + a[0]b[1] \\
c[2] &= a[2]b[0] + a[1]b[1] + a[0]b[2] \\
\vdots &= \vdots \\
c[M+N-3] &= a[M-1]b[N-2] + a[M-2]b[N-1] \\
c[M+N-2] &= a[M-1]b[N-1]
\end{aligned}
$$

If you know a little linear algebra, you can write this convolution as the product of a **convolution matrix** and a vector.

Convolution Matrix of an LTI System (2 of 2)

To illustrate the idea, suppose we have $\{a[0], a[1]\}$ and $\{b[0], b[1], b[2]\}$.

$$\begin{array}{rclll} c[0] & = & a[0]b[0] & & \\ c[1] & = & a[1]b[0] & +a[0]b[1] & \\ c[2] & = & & a[1]b[1] & +a[0]b[2] \\ c[3] & = & & & a[1]b[2] \end{array}$$

This is the same as

$$\begin{bmatrix} c[0] \\ c[1] \\ c[2] \\ c[3] \end{bmatrix} = \underbrace{\begin{bmatrix} a[0] & 0 & 0 \\ a[1] & a[0] & 0 \\ 0 & a[1] & a[0] \\ 0 & 0 & a[1] \end{bmatrix}}_{\text{convolution matrix}} \begin{bmatrix} b[0] \\ b[1] \\ b[2] \end{bmatrix} = \underbrace{\begin{bmatrix} b[0] & 0 \\ b[1] & b[0] \\ b[2] & b[1] \\ 0 & b[2] \end{bmatrix}}_{\text{convolution matrix}} \begin{bmatrix} a[0] \\ a[1] \end{bmatrix}$$

The convolution matrix has a Toeplitz structure and can be generated in Matlab with the `convmtx` command.

Properties of Convolution

- Commutative

$$x[n] * h[n] = h[n] * x[n]$$

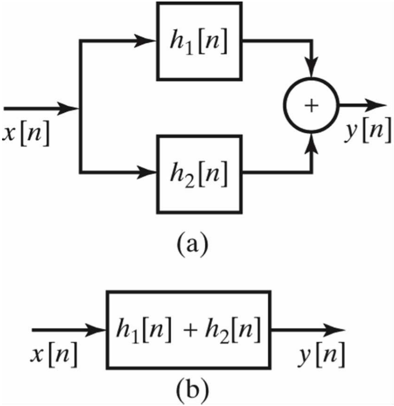
- Distributes over addition (direct result of linearity)

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

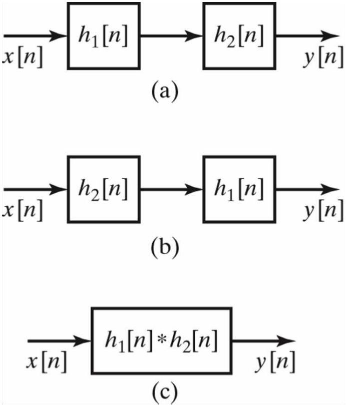
- Associative

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

Parallel LTI Systems (Distributive Property)



Series/Cascade LTI Systems (Commutative and Associative Properties)



When are LTI Systems Stable? (part 1 of 3)

Theorem

A discrete-time LTI system is BIBO stable if and only if its impulse response is absolutely summable.

Sufficiency: Assume an absolutely summable impulse response $\sum_{k=-\infty}^{\infty} |h[k]| = B_h < \infty$. Given a bounded input $|x[n]| \leq B_x < \infty$ for all n we can write

$$\begin{aligned}
|y[n]| &= \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right| \\
&\leq \sum_{k=-\infty}^{\infty} |h[k]| \cdot |x[n-k]| \\
&\leq B_x \sum_{k=-\infty}^{\infty} |h[k]| \\
&\leq B_x B_h < \infty
\end{aligned}$$

When are LTI Systems Stable? (part 2 of 3)

Theorem

A discrete-time LTI system is BIBO stable if and only if its impulse response is absolutely summable.

Necessity: Now assume the impulse response is not absolutely summable such that $\sum_{k=-\infty}^{\infty} |h[k]| = \infty$. To show the system is not BIBO stable, let

$$x[n] = \begin{cases} \frac{h^*[-n]}{|h[-n]|} & h[-n] \neq 0 \\ 0 & h[-n] = 0 \end{cases}$$

and note that $|x[n]| \leq 1$ for all n. Then

$$y[0] = \sum_{k=-\infty}^{\infty} x[0-k]h[k] = \sum_{k=-\infty}^{\infty} \frac{|h[k]|^2}{|h[k]|} = \infty.$$

Since there exists a bounded input which causes an unbounded output, the system is not BIBO stable.

When are LTI Systems Stable? (part 3 of 3)

In general, systems with finite-duration impulse response (called FIR systems) are always stable as long as $|h[n]| < \infty$ for all $n \in \mathbb{Z}$.

Examples:

- Is the moving average $y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$ stable?
- Is the accumulator $y[n] = \sum_{k=-\infty}^{n} x[k]$ stable?
- Is the "forgetting accumulator" $y[n] = \sum_{k=-\infty}^{n} a^{n-k} x[k]$ stable?

When are LTI Systems Causal?

Theorem

A discrete-time LTI system is causal if the impulse response $h[n] = 0$ for all integer $n < 0$.

This is easy to see from the convolution expression

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

$$\stackrel{\text{causal}}{=} \sum_{k=0}^{\infty} h[k]x[n-k]$$

since $y[n]$ only depends on $x[n], x[n-1], \dots$.

Linear Constant-Coefficient Difference Equations

Many useful LTI systems can be described by finite-dimensional constant-coefficient difference equations:

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m]$$

with M and N both finite.

Example 1: The moving average system $y[n] = \frac{1}{M+1} \sum_{k=0}^{M} x[n-k]$ is obviously a finite-dimensional constant-coefficient difference equation.

Example 2: The accumulator $y[n] = \sum_{k=-\infty}^{n} x[k]$ is not in this form. But

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

hence $y[n] = y[n-1] + x[n]$, or equivalently, $y[n] - y[n-1] = x[n]$, which is a finite-dimensional constant-coefficient difference equation.

Linear Constant-Coefficient Difference Equations

Note that there can be multiple difference equations that have the same impulse response.

System 1: $y[n] = \frac{1}{M+1}\sum_{k=0}^{M} x[n-k]$. The impulse response of this system is clearly

$$h[n] = \frac{1}{M+1}\sum_{k=0}^{M} \delta[n-k]$$
$$= \begin{cases} \frac{1}{M+1} & n \in \{0, 1, \dots, M\} \\ 0 & \text{otherwise} \end{cases}$$

continued...

Linear Constant-Coefficient Difference Equations

System 2: $y[n] - y[n-1] = \frac{1}{M+1}\left(x[n] - x[n-M-1]\right)$. To compute the impulse response, we assume a relaxed initial condition $y[n] = 0$ for all $n < 0$.

$$\begin{aligned}
y[0] &= \frac{1}{M+1}\left(\delta[0] - \delta[0-M-1]\right) + y[-1] = \frac{1}{M+1} \\
y[1] &= \frac{1}{M+1}\left(\delta[1] - \delta[1-M-1]\right) + y[0] = \frac{1}{M+1} \\
&\vdots \\
y[M] &= \frac{1}{M+1}\left(\delta[M] - \delta[M-M-1]\right) + y[M-1] = \frac{1}{M+1} \\
y[M+1] &= \frac{1}{M+1}\left(\delta[M+1] - \delta[M+1-M-1]\right) + y[M] = 0 \\
y[M+2] &= \frac{1}{M+1}\left(\delta[M+2] - \delta[M+2-M-1]\right) + y[M+1] = 0 \\
&\vdots
\end{aligned}$$

which is the same as system 1.

Solving LTI Systems Described by Difference Equations

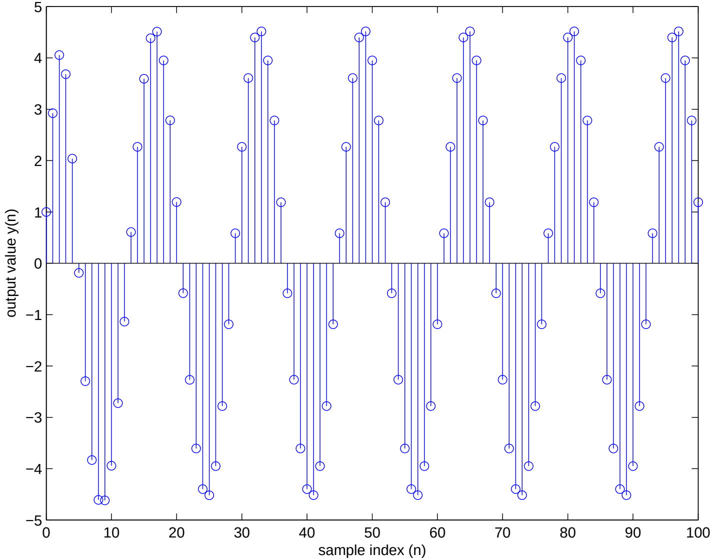
Matlab can also numerically solve LTI systems described by finite-dimensional constant-coefficient difference equations

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m]$$

Example

```
a = [1 -1 0.5];          % vector containing a0, a1, a2
b = [1 1];               % vector containing b0, b1
n = 0:100;               % sample indices
x = cos(pi/8*n);         % input function
zi = [0 0];              % initial conditions (relaxed here)
y = filter(b,a,x,zi);    % compute output
stem(0:length(y)-1,y); % plot
xlabel('sample index (n)');
ylabel('output value y(n)');
```

Also check out Matlab functions `impulse` and `step`.


5
4
3
2
1
0
−1
−2
−3
−4
−5
output value y(n)
0
10
20
30
40
50
60
70
80
90
100
sample index (n)

Summary

1. Discrete-time signals
2. Discrete-time systems and qualitative properties (linearity, time-invariance, causality, stability)
3. DT-LTI systems
4. Impulse response and convolution
5. Linear constant-coefficient difference equations

What's Next?

This is the only "lecture" in ECE503. The remaining class meetings will be structured as

- First half: Discussion/examples related to reading assignment, screencasts, suggested problems.
- Second half: 60-minute quiz.

Next week's meeting will be focused on topics in Chapter 2 of your textbook.

Any questions/discussion during the week should be posted to Piazza. Students are encouraged to collaborate on everything except the quizzes.