6.003: Signal Processing

Systems

- System Abstraction
- Linearity and Time Invariance

Homework 5 is posted
- shorter than usual
- no lab

Quiz 1 results are posted
- grades and grade definitions

October 7, 2021
Notes on Grading

We explicitly use the MIT definitions of grades.

A **Exceptionally good performance** demonstrating a superior understanding of the subject matter, a foundation of extensive knowledge, and a skillful use of concepts and/or materials.

B **Good performance** demonstrating capacity to use the appropriate concepts, a good understanding of the subject matter, and an ability to handle the problems and materials encountered in the subject.

C **Adequate performance** demonstrating an adequate understanding of the subject matter, an ability to handle relatively simple problems, and adequate preparation for moving on to more advanced work in the field.

D **Minimally acceptable performance** demonstrating at least partial familiarity with the subject matter and some capacity to deal with relatively simple problems, but also demonstrating deficiencies serious enough to make it inadvisable to proceed further in the field without additional work.

F **Failed.**
Points, GPA Scale, and Letter Grade

Grading Procedure

• We grade the exams on a point basis.
• We convert the 100 point score into a 5-point GPA scale using MIT’s definitions of letter grades.
• Your final score in 6.003 will be a weighted sum of your 5-point scores for homeworks, labs, quizzes, and final exam.

<table>
<thead>
<tr>
<th>Total Points</th>
<th>&quot;GPA&quot; Scale</th>
<th>Letter Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>A/B boundary</td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>B/C boundary</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>C/D boundary</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>D/F boundary</td>
<td>1</td>
<td>D</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td>F</td>
</tr>
</tbody>
</table>

The goal of this scheme is to be transparent about your grade status.
Represent a \textbf{system} (physical, mathematical, or computational) by the way it transforms an \textbf{input signal} into an \textbf{output signal}.
Example: Mass and Spring
Example: Mass and Spring

- Mass and Spring System
  - $x(t)$
  - $y(t)$
Example: Mass and Spring

\[ x(t) \quad y(t) \]

\[ x(t) \quad y(t) \]

mass & spring system

\[ x(t) \quad y(t) \]

mass & spring system

\[ x(t) \quad y(t) \]
Example: Tanks

\[ r_0(t) \rightarrow r_1(t) \rightarrow r_2(t) \]

\[ h_1(t) \rightarrow h_2(t) \]

Tank system

\[ r_0(t) \rightarrow \text{tank system} \rightarrow r_2(t) \]
Example: Tanks

\[ r_0(t) \]
\[ r_1(t) \]
\[ r_2(t) \]

\[ h_1(t) \]
\[ h_2(t) \]
Example: Cell Phone System
Example: Cell Phone System
The Signals and Systems approach has broad application: electrical, mechanical, optical, acoustic, biological, financial, ...
Signals and Systems: Modular

The representation does not depend upon the physical substrate.

focuses on the flow of **information**, abstracts away everything else
Signals and Systems: Hierarchical

Representations of component systems are easily combined.

Example: cascade of component systems

Component and composite systems have the same form, and are analyzed with same methods.
System Abstraction

The system abstraction builds on and extends our work with signals.

We will use this approach to process a variety of signals:

- **audio**: equalization, noise reduction, reverberation reduction, echo cancellation, pitch shift (auto-tune)
- **image**: smoothing, edge enhancement, unsharp masking, feature detection
- **video**: image stabilization, motion magnification
System Abstraction

The system abstraction builds on and extends our work with signals.

We will look at three different representations for systems:

- **Difference Equation**: algebraic constraint on samples
- **Convolution**: represent a system by its unit-sample response
- **Filter**: represent a system by its frequency response
Example: Three-Point Averaging

The output at time $n$ is average of inputs at times $n-1$, $n$, and $n+1$.

$$y[n] = \frac{1}{3} (x[n-1] + x[n] + x[n+1])$$
Example: Three-Point Averaging

The output at time $n$ is average of inputs at times $n-1$, $n$, and $n+1$.

$$y[n] = \frac{1}{3} \left( x[n-1] + x[n] + x[n+1] \right)$$
Example: Three-Point Averaging

The output at time \( n \) is average of inputs at times \( n-1, n, \) and \( n+1 \).

\[
y[n] = \frac{1}{3} \left( x[n-1] + x[n] + x[n+1] \right)
\]
Example: Three-Point Averaging

The output at time $n$ is average of inputs at times $n-1$, $n$, and $n+1$.

$$y[n] = \frac{1}{3}(x[n-1] + x[n] + x[n+1])$$
Example: Three-Point Averaging

The output at time $n$ is average of inputs at times $n-1$, $n$, and $n+1$.

$$y[n] = \frac{1}{3} \left( x[n-1] + x[n] + x[n+1] \right)$$
Example: Three-Point Averaging

The output at time $n$ is average of inputs at times $n-1$, $n$, and $n+1$.

$$y[n] = \frac{1}{3} \left( x[n-1] + x[n] + x[n+1] \right)$$
Example: Three-Point Averaging

The output at time $n$ is average of inputs at times $n-1$, $n$, and $n+1$.

$$y[n] = \frac{1}{3} \left( x[n-1] + x[n] + x[n+1] \right)$$

Think of this process as a system with input $x[n]$ and output $y[n]$.
Properties of Systems

We will focus primarily on systems that have two important properties:

- **linearity**
- **time invariance**

Such systems are both useful and mathematically tractable.
Additivity

A system is additive if its response to a **sum of signals** is equal to the **sum of the responses** to each signal taken one at a time.

Given

\[ x_1[n] \rightarrow \text{system} \rightarrow y_1[n] \]

and

\[ x_2[n] \rightarrow \text{system} \rightarrow y_2[n] \]

the **system is additive** if

\[ x_1[n] + x_2[n] \rightarrow \text{system} \rightarrow y_1[n] + y_2[n] \]

is true for all possible inputs and all times \( n \).

Example: the three-point averager is additive:

The three-point average of the sum of two signals is equal to the sum of the three-point averages of the individual signals.
**Homogeneity**

A system is homogeneous if multiplying its input signal by a constant multiplies the output signal by the same constant.

Given

\[ x_1[n] \xrightarrow{\text{system}} y_1[n] \]

the **system is homogeneous** if

\[ \alpha x_1[n] \xrightarrow{\text{system}} \alpha y_1[n] \]

is true for all \( \alpha \) and all possible inputs and all times \( n \).

Example: the three-point averager is homogeneous.
Doubling an input signal doubles its three-point average.
Linearity

A system is linear if its response to a **weighted sum of input signals** is equal to the **weighted sum of its responses** to each of the input signals.

Given

\[ x_1[n] \rightarrow \text{system} \rightarrow y_1[n] \]

and

\[ x_2[n] \rightarrow \text{system} \rightarrow y_2[n] \]

the system is linear if

\[ \alpha x_1[n] + \beta x_2[n] \rightarrow \text{system} \rightarrow \alpha y_1[n] + \beta y_2[n] \]

is true for all \( \alpha \) and \( \beta \) and all possible inputs and all times \( n \).

A system is linear if it is both additive and homogeneous.
Time-Invariance

A system is time-invariant if delaying the input signal simply delays the output signal by the same amount of time.

Given

\[ x[n] \rightarrow \text{system} \rightarrow y[n] \]

the \textbf{system is time invariant} if

\[ x[n-n_0] \rightarrow \text{system} \rightarrow y[n-n_0] \]

is true for all \( n_0 \) and for all possible inputs and all times \( n \).

Example: The three-point averager is time-invariant. Shifting the input to a 3-pt averager simply shifts the output by that same amount.
Consider a system represented by the following difference equation:

\[ y[n] = x[n] + x[n-1] \]

for all \( n \).

Is this system **linear**?
Consider a system represented by the following difference equation:

\[ y[n] = x[n] + x[n-1] \]

for all \( n \).

Is this system **linear**?

Assume that \( x_1[n] \to y_1[n] \). Then \( y_1[n] = x_1[n] + x_1[n-1] \).

Assume that \( x_2[n] \to y_2[n] \). Then \( y_2[n] = x_2[n] + x_2[n-1] \).

Multiply \( \alpha \) times equation 1 and add \( \beta \) times equation 2:

\[ \alpha y_1[n] + \beta y_2[n] = \alpha x_1[n] + \beta x_2[n] + \alpha x_1[n-1] + \beta x_2[n-1] \]

This equation shows that \( \alpha x_1[n] + \beta x_2[n] \to \alpha y_1[n] + \beta y_2[n] \).

Therefore the system is **linear**.
Representing Systems with Difference Equations

Determining linearity from a difference equation representation.

Example 2.

\[ y[n] = x[n] \times x[n-1] \]

for all \( n \).

Is this system \textit{linear}?
Representing Systems with Difference Equations

Determining linearity from a difference equation representation.

Example 2.

\[ y[n] = x[n] \times x[n-1] \]

for all \( n \).

Is this system \textit{linear}?

Assume that \( x_1[n] \rightarrow y_1[n] \). Then \( y_1[n] = x_1[n] \times x_1[n-1] \).

Find the response \( y_2[n] \) when \( x_2[n] = \alpha x_1[n] \):

\[
\begin{align*}
y_2[n] &= x_2[n] \times x_2[n-1] \\
      &= \alpha x_1[n] \times \alpha x_1[n-1] \\
      &= \alpha^2 x_1[n] \times x_1[n-1] \\
      &= \alpha^2 y_1[n]
\end{align*}
\]

Multiplying input \( x_1[n] \) by \( \alpha \) does \textbf{not} multiply the output \( y_1[n] \) by \( \alpha \). It multiplies \( y_1[n] \) by \( \alpha^2 \)!

Therefore the system is \textit{neither homogeneous nor linear}. 
Representing Systems with Difference Equations

Determining linearity from a difference equation representation.

Example 3:
\[ y[n] = nx[n] \]
for all \( n \).

Is the system \textbf{linear}?
Representing Systems with Difference Equations

Determining linearity from a difference equation representation.

Example 3:

\[ y[n] = nx[n] \]

for all \( n \).

Is the system **linear**?

Let \( x[n] = \alpha x_1[n] + \beta x_2[n] \).

Then

\[
\begin{align*}
y[n] &= n(\alpha x_1[n] + \beta x_2[n]) \\
    &= \alpha nx_1[n] + \beta nx_2[n] \\
    &= \alpha y_1[n] + \beta y_2[n]
\end{align*}
\]

Therefore the system is **linear**.
Representing Systems with Difference Equations

Determining time invariance from a difference equation.

Example 3.

\[ y[n] = nx[n] \]

for all \( n \).

Is the system **time-invariant**?
Representing Systems with Difference Equations

Determining time invariance from a difference equation.

Example 3.

\[ y[n] = nx[n] \]

for all \( n \).

Is the system **time-invariant**?

If time-invariant, delaying input by 1 should delay output by 1. Let \( x_1[n] \) represent a delayed version of the input.

\[ x_1[n] = x[n-1] \]

The corresponding output \( y_1[n] \) is given by

\[ y_1[n] = nx_1[n] = nx[n-1] \]

This is not the same as delaying the original output:

\[ y[n-1] = (n-1)x[n-1] \]

Since \( y_1[n] \neq y[n-1] \), the system is **not time-invariant**.
Linear Difference Equations with Constant Coefficients

If a discrete-time system can be described by a linear difference equation with constant coefficients, then the system is linear and time-invariant.

General form:
\[ \sum_{l} c_{l}y[n-l] = \sum_{m} d_{m}x[n-m] \]

**Additivity:** output of sum is sum of outputs
\[ \sum_{l} c_{l}(y_{1}[n-l] + y_{2}[n-l]) = \sum_{m} d_{m}(x_{1}[n-m] + x_{2}[n-m]) \quad \checkmark \]

**Homogeneity:** scaling an input scales its output
\[ \sum_{l} \alpha c_{l}y[n-l] = \sum_{m} \alpha d_{m}x[n-m] \quad \checkmark \]

**Time invariance:** delaying an input delays its output
\[ \sum_{l} c_{l}y[(n-n_0)-l] = \sum_{m} d_{m}x[(n-n_0)-m] \quad \checkmark \]

Notice that \( y[n] = x[n] + 1 \) does not represent a linear system. Why?
Summary: System Abstraction

The system abstraction builds on and extends our work with signals.

**Goal:** characterize a **system** to better understand the relation between two signals.

Next Time: Three representations for systems:
- **Difference Equation:** algebraic **constraint** on samples
- **Convolution:** represent a system by its **unit-sample response**
- **Filter:** represent a system by its **frequency response**