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

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>“GPA” 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>B</td>
</tr>
<tr>
<td>B/C boundary</td>
<td>3</td>
<td>C</td>
</tr>
<tr>
<td>C/D boundary</td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>D/F boundary</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>0%</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

The goal of this scheme is to be transparent about your grade status.

The System Abstraction

Represent a system (physical, mathematical, or computational) by the way it transforms an input signal into an output signal.

Example: Mass and Spring

Example: Tanks

Example: Cell Phone System
Signals and Systems: Widely Applicable
The Signals and Systems approach has broad application: electrical, mechanical, optical, acoustic, biological, financial, ...

- mass & spring system
- tank system
- cell phone tower tower cell
- E/M optic fiber
- E/M sound

Signals and Systems: Modular
The representation does not depend upon the physical substrate.

- sound in
- sound out

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

Composite system

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.

- signal in
- signal out

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.

- signal in
- signal out

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 \( n+1 \).

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

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
\[ x_1[n] \rightarrow \text{system} \rightarrow y_1[n] \]
the system is homogeneous if
\[ \alpha x_1[n] \rightarrow \text{system} \rightarrow \alpha y_1[n] \]
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 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.

Representing Systems with Difference Equations
Consider a system represented by the following difference equation:
\[ y[n] = x[n] + x[n-1] \]
for all \( n \).

Is this system 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 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?

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
\[ \sum_l a(y[n-l] + y'[n-l]) = \sum_m d_m (x_1[n-m] + x_2[n-m]) \]

Homogeneity: scaling an input scales its output
\[ \sum_l a y[n-l] = \sum_m ad_m x[n-m] \]

Time invariance: delaying an input delays its output
\[ \sum_l y[(n-n_0)-l] = \sum_m d_m x[(n-n_0)-m] \]

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