Major topics of 6.01

- Controlling complexity
  - abstraction and modularity
- Interacting with the real world
  - models
- Coping with error and incomplete information
  - reasoning about uncertainty

PCAP framework for Python

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primitives</td>
<td>+, *, ==</td>
</tr>
<tr>
<td>Combination</td>
<td>if, f(g(x))</td>
</tr>
<tr>
<td>Abstraction</td>
<td>def</td>
</tr>
<tr>
<td>Patterns</td>
<td>higher-order fns</td>
</tr>
<tr>
<td></td>
<td>polymorphism, inheritance</td>
</tr>
</tbody>
</table>

PCAP framework for signals and systems

<table>
<thead>
<tr>
<th>Primitives</th>
<th>signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combination</td>
<td>adder, gain, delay</td>
</tr>
<tr>
<td>Abstraction</td>
<td>system function</td>
</tr>
<tr>
<td>Patterns</td>
<td>feedback</td>
</tr>
</tbody>
</table>
Ideal linear resistor model

Series combination
\[ R = R_1 + R_2 \]

Parallel combination
\[ \frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} \]

Voltage divider
\[ V_v = V \cdot \frac{R_2}{R_1 + R_2} \]
Ideal independent voltage source

\[ V = V_s \]
Different styles of computational models

- Functional models
  - Systems are described as collections of mathematical functions with inputs and outputs

- Object models
  - Things are described as collections of objects that have behavior and internal state

- Constraint models
  - Systems are described as collections of the constrains that must hold among their components. There are no "inputs" and "outputs", just constraints that must be satisfied.

* For 6.01, we’ll assume \( f \) is a linear function.
Patterns

Abstraction

Combination

Wire parts together at the terminals

Resistors, sources, 2-terminal devices (assume all linear)

Primitives

PCAP framework for circuits

For the SPCVM, applying the cycle-averaging technique, we obtain a circuit with dependent voltage and current sources:

\[ V_c = CVP \]
\[ V_a = ABP \]
\[ V_v = VP \]

1-port

Apply a voltage and measure the current. The 1-port is completely described by the relation of the voltage and the current. It doesn’t matter what’s in the box, so long as the relation holds.

Analogy with software: An abstract data type is described by its operations.
Thévenin model

Any 1-port made up of linear resistors and sources, when viewed from the terminals, is completely electrically equivalent to a network composed of a single resistor and a single voltage source.

\[ v = V_{TH} + iR_{TH} \]

Open circuit voltage

Resistance seen from the terminals what all independent sources are suppressed

Example: From Wikipedia

The voltage at the terminals is 7.5V, so \( V_{TH} \) is 7.5V
The resistance seen from the terminals is 2kΩ, so $R_{TH}$ is 2kΩ.

$V = 7.5 \times \frac{3}{3+2} = 7.5 \times \frac{3}{5} = 4.5$

**PCAP framework for circuits**

<table>
<thead>
<tr>
<th>Primitives</th>
<th>Resistors, sources, 2-terminal devices (assume all linear)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combination</td>
<td>Wire parts together at the terminals</td>
</tr>
<tr>
<td>Abstraction</td>
<td>1-port (Thévenin model)</td>
</tr>
<tr>
<td>Patterns</td>
<td></td>
</tr>
</tbody>
</table>
... and in general, n-ports

Operational amplifier (op-amp)
5-terminal device

...
Ideal op-amp model

\[ V_0 - V_{\text{out}} = K (v_+ - v_-) \]
\[ i_1 = i_2 = 0 \]

\[ K \text{ might be around } 1,000 \]

Voltage follower (or buffer)

\[ V_{\text{out}} (1 + K) = KV_S \]
\[ V_{\text{out}} = K (v_+ - v_-) \]
\[ V_{\text{out}} = K (V_S - V_{\text{out}}) \]
\[ V_{\text{out}} = KV_S - K V_{\text{out}} \]
\[ V_{\text{out}} + K V_{\text{out}} = KV_S \]

\[ \frac{V_{\text{out}}}{V_S} = \frac{K}{1 + K} \approx 1 \]
Using a buffer to cascade voltage dividers in lab last week

An even simpler op-amp model (K infinite)

\[ v_{out} - v_{in} = K(v_+ - v_-) \]
- Draws no current, that is, \( i_1 = i_2 = 0 \)
- If \( K \) is very large, and \( v_{out} \) is finite, then \( v_+ = v_- \)

Inverting amplifier
Non-inverting amplifier

Harold S. Black (1898-1983)
Inventor of the negative feedback amplifier (1927)