6.01 Midterm Review

LTI

→ **Goal:** Framework for analyzing behavior
   - Model physical systems (e.g., robot, bunnies)

→ **Signal:** Function of time
   - In 6.01 we just consider DT signals, \( x[n] \) defined for \( n \in \mathbb{Z} \)
   - **Notation:** Signals represented by upper case, e.g., \( X \)
   - Samples represented by lowercase, e.g., \( x[n] \)
   - \( x[n] \) = the value of signal \( X \) at time \( n \)

→ **Useful Signals:**
   - Unit Sample Signal \((\delta)\)
     \[
     \delta[n] = \begin{cases} 
     1, & \text{if } n = 0 \\
     0, & \text{otherwise}
     \end{cases}
     \]
     → Any signal can be written as a sum of scaled and delayed versions of \( \delta \).
   - Unit Step Signal \((u)\)
     \[
     u[n] = \begin{cases} 
     1, & \text{if } n \geq 0 \\
     0, & \text{otherwise}
     \end{cases}
     \]
     \[
     u[n] = \sum_{i=0}^{n} \delta[n-i]
     \]

→ **System:** Transforms Signals

\[
\begin{array}{c}
X \quad \xrightarrow{\text{System}} \quad Y \\
\text{input signal} \quad \rightarrow \quad \text{output signal}
\end{array}
\]

→ In 6.01 we just consider DT-LTI systems.

→ **LTI = Linear Time-Invariant**

→ **Linear:**
   - \( X_1 \) \( \rightarrow \) System \( \rightarrow \) \( Y_1 \)
   - and \( X_2 \) \( \rightarrow \) System \( \rightarrow \) \( Y_2 \)
   - \( \alpha X_1 + \beta X_2 \rightarrow \text{System} \rightarrow \alpha Y_1 + \beta Y_2 \)
- We have many different representations for systems:
  1. State Machines
  2. Block Diagrams
  3. Difference Equations
  4. Operator Equations
  5. System Functionals
  6. Poles

- Good for: Seeing exact output of a system to a particular input.

- Infrastructure forStateMachine:
  
  - Start(C)
  - Step(inp)
  - transduce (rops)

  - change internal state of SM
  - already defined for you.

- Example:
  ```python
  class MysterySM(SM):
      def __init__ (self, P):
          Self. p = P
          self. start = 0
      def get_next_values (self, state, inp):
          out = self. p * state + inp
          return (out, out)

  MysterySM(0.5). transduce ([1, 0, 0, 0]) → [1.0, 0.5, 0.25, 0.125]
  MysterySM(2). transduce ([1, 0, 0, 0]) → [1, 2, 4, 8]
  ```

- Moving between reps:
  - Labeling individual signals usually helps.
Block Diagrams: Visualize Signal Flow Paths

→ Example: \[ \begin{array}{c}
\[ \begin{array}{c}
X \\
+ \\
\rightarrow P \\
\rightarrow R \\
\rightarrow Y
\end{array}
\end{array} \]

→ Easy to see how each sample of the output \( Y \) is computed.

Difference Equations: Let us compute output to specific inputs

→ Example: \[ y[n] = x[n] + p y[n-1] \]

→ Unit Sample Response:

\[
\begin{array}{c|cccccc}
 n & 0 & 1 & 2 & 3 & 4 & \ldots \\
---------------------
 s[n] & 0 & 1 & 0 & 0 & 0 & 0 \\
 h[n] & 0 & 1 & p & p^2 & p^3 & p^4
\end{array}
\]

→ Another description of LTI using difference equations:

\[ y[n] = c_0 y[n-1] + c_1 y[n-2] + \ldots + c_{k-1} y[n-k] +
\]
\[ d_0 x[n] + d_1 x[n-1] + \ldots + d_j x[n-j] \]

where \( c_0, \ldots, c_{k-1}, d_0, \ldots, d_j \) are fixed constants.

Operator Equations: Use the Right Shift Operator \( R \)

→ Very easy to manipulate using Polynomial algebra

→ Polynomial algebra "works" because we're dealing with LTI systems (you're not responsible for knowing the details of why polynomial algebra works).

→ Example: \( Y = X + PRY \)

→ Description of LTI systems:

\[
Y = c_0 R Y + c_1 R^2 Y + \ldots + c_k R^k Y +
\]
\[ d_0 X + d_1 R X + \ldots + d_j R^j X. \]
5. **System Functionals**: Ratio of output signal to input signal in $R$.

   $\frac{Y}{X} = \frac{1}{1-PR}$

   - Example: $X \rightarrow \frac{1}{1-PR} \rightarrow Y$

   - Very nice method of abstraction (problems 2, 3)

6. **Poles**: to understand long-term behavior of System

   - The first fire representations specify the exact system, but poles do not! That is, two different systems may have the same set of poles.

   - Basis for poles: $x \rightarrow \quad \begin{array}{c}
   \text{Pole: } P \\
   \end{array} \quad \rightarrow Y$

   - Unit Sample Response: $h[n] = P^n u[n]$

   - Idea: Any LTI system can be described as a sum of small components like the small system above.

   - The Unit Sample response to an arbitrary System can be written as:

   $$h[n] \sim \sum_{i=1}^{K} C_i P_i^n$$

   We Call $P_1, P_2, \ldots, P_K$ the poles of the system.

   - $C_1, C_2, \ldots, C_K$ will all be polynomials in $n$.

   - If $P_1, P_2, \ldots, P_K$ are distinct, $C_1, \ldots, C_K$ will be constants. The key is that the response is dominated by the exponential $P_i^n$ terms.

Don't worry too much about these details.
→ Often we look at the dominant poles, the poles of largest magnitude, to understand the long-term behavior of Systems.

→ Real dominant pole:

\[ \begin{array}{c|c|c|c}
\text{alternating} & \text{divergent} & \text{monotonic} & \text{divergent} \\
\text{Convergent} & & & \\
\hline
-1 & & \text{stabilizing, stable} & \\
O & & & \\
1 & & \text{monotonic, stable} & \\
\end{array} \]

→ Complex poles come in conjugate pairs

→ Complex dominant pole:

\[ \begin{array}{c}
\text{Convergent} \\
\text{Stable, not convergent} \\
\text{divergent} \\
\end{array} \]

→ Complex numbers: two representations

\[ \begin{array}{c}
\text{Rectangular} \\
\text{Polar} \\
\end{array} \]

\[ \begin{array}{c}
\text{Rectangular: } a + bi \\
\text{Polar: } r e^{i \omega} \\
\end{array} \]

\[ \begin{array}{c}
\text{Relationships: } r = \sqrt{a^2 + b^2}, \quad \omega = \tan^{-1} \left( \frac{b}{a} \right), \quad a = r \cos(\omega), \quad b = r \sin(\omega) \\
\end{array} \]

→ Computing poles from System Functional.

→ Trick: \[ R = \frac{1}{3} \] (more on this in 6.003 😜)

→ Example: \[ H(R) = \frac{R}{1 - \frac{3}{2} R + R^2} \quad \Rightarrow \quad H(z) = \frac{\frac{1}{3}}{1 - \frac{3}{2} z + z^2} \]

\[ = \frac{\frac{2}{3}}{z^2 - \frac{3}{2} z + 1} \]

→ Poles = roots of denominator of \( H(z) \)

\[ z^2 - \frac{3}{2} z + 1 = (z - 1)(z - \frac{1}{2}) \quad \Rightarrow \quad 1, \frac{1}{2} \]
- **Primitives and Combinations for LTI Systems**:

  - **Primitives**
    1. **Gain**: \( y[n] = k \cdot x[n] \)
      \[
      \begin{align*}
      Y &= kX \\
      \frac{Y}{X} &= k
      \end{align*}
      \]
    2. **Delay**: \( y[n] = x[n-1] \)
      \[
      \begin{align*}
      Y &= R X \\
      \frac{Y}{X} &= R
      \end{align*}
      \]

  - **Combinations**
    1. **Cascade**:
      \[
      H = H_1 H_2
      \]
      \[
      \begin{align*}
      H &= \frac{H_1}{1 - H_4 H_2} \\
      \end{align*}
      \]
    2. **Feedback**:

        (Problem 4)

- **Feedback Systems**

  Desired \( \rightarrow \) Error \( \rightarrow \) Controller \( \rightarrow \) Command \( \rightarrow \) Plant \( \rightarrow \) Actual

  \( \rightarrow \) **Controller**: Commands to Physical System

  \( \rightarrow \) **Plant**: Model of Physical System

  \( \rightarrow \) **Sensor**: Model of the information we have about the attribute we are trying to control.
- Feedback Systems from Design Labs 4, 5, 6

- Wall Finder:

\[ D_i \rightarrow E \rightarrow K \rightarrow Y \rightarrow -T \rightarrow V \rightarrow R \rightarrow D_o \]

\[ Y = KE \]
\[ D_o = RD_o - TRV \]
\[ D_s = RD_o \]

Root Locus:

\[ \text{Poles: } 1 \pm \frac{1}{2} \left( 1 + 4TK \right) \]

- Wall Follower

\[ \frac{S}{E} = K \]
\[ \frac{\Theta}{S} = \frac{TR}{1-R} \]
\[ \frac{D_o}{\Theta} = VTR \]
\[ \frac{D_o}{\Theta} = \frac{VTR}{1-R} \]
\[ D_o = \frac{VTR}{1-R} \]

\[ \frac{D_o}{D_i} = \frac{KVT^2R^2}{(1-R)^2} \]
\[ \frac{D_o}{D_i} = \frac{KVT^2R^2}{1-2R + (1+KVT^2)R^2} \]
Wall Follower: Proportional + angle

\[ D_c \rightarrow E \rightarrow K_a \rightarrow J_2 \rightarrow \text{Plant 1} \rightarrow \text{Plant 2} \rightarrow D_o \]

- Can choose \( K_a \) and \( K_p \) to produce convergent system!

Superposition
- If we know the unit sample response \( h[n] \), we can compute the response to any signal.

\[ x[n] \rightarrow \text{System with USR } h[n] \rightarrow y[n] = h[n] - 2h[n-1] + 1.5 h[n-3] \]

\[ x[n] = 5[n] - 25[n-2] + 1.5 5[n-3] \]