How does a computer experience the world?

\[ u(t) \xrightarrow{DAC} \text{System} \xrightarrow{A/D} y(t) \]

\[ u_c(t) = u(t) \iff t \in (\Delta, (n+1)\Delta) \]

Discrete model:
\[ x_d[i+1] = a \cdot x_d[i] + b \cdot u_c[i] + \omega[i] \]

**Vector case:**
\[ x[t+1] = A \cdot x[t] + B \cdot u[t] + w[t] \]

From last lecture:
- Look by rows of unknowns
- \( A \) and \( B \)

Stack unknowns vertically:
\[ \begin{bmatrix} x[0]^T, u[0]^T \end{bmatrix} \xrightarrow{\text{A}} \begin{bmatrix} \tilde{a}_1 \tilde{b}_1 \end{bmatrix} \]

Full matrix:
\[ T = \begin{bmatrix} \tilde{P} & \tilde{R} & \tilde{s} \end{bmatrix} \]
\[ D \hat{\rho}_r = \hat{s}_r \]

\[ \text{Least Squares: } \hat{\rho}_r = (D^T D)^{-1} D^T \hat{s}_r \]

\[ \hat{\rho}_1 = \left( \frac{D^T D}{m} \right)^{-1} D^T \hat{s}_1 \]
\[ \hat{\rho}_2 = \left( \frac{D^T D}{m} \right)^{-1} D^T \hat{s}_2 \]
\[ \vdots \]
\[ \hat{\rho}_n = \left( \frac{D^T D}{m} \right)^{-1} D^T \hat{s}_n \]

\[ \hat{\rho} = (D^T D)^{-1} D^T \hat{s} \]

We have \( \hat{\rho} \), how do we know if it is any good?

\textbf{Gold Standard}: Use the \( \hat{\rho} \) for control and see if it works?

\textbf{Silver Standard}: Test \( \hat{\rho} \) against some other data.

\textbf{Test Traces}: \( x_{\text{test}\{0\}}, \ldots, x_{\text{test}\{T\}} \) \( \{ \} \) Collected from the real world.

\textbf{Not Used to estimate} \( \hat{\rho} \).
Try predicting the next state:
\[ \hat{\mathbf{x}}_{+1} = \mathbf{A} \hat{\mathbf{x}}_{t-1} \mathbf{B} \mathbf{u}_{t-1} \]

Test data

Learned parameters

Computation Test Error = \[ \frac{1}{T} \sum_{t=1}^{T} [\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}]^2 \]

Used to compute \( P \)

Computation Training Error = \[ \frac{1}{T} \sum_{t=1}^{T} [\mathbf{x}_{t} - \hat{\mathbf{x}}_{t}]^2 \]

If close, and acceptable for the application, good!

Least squares minimized this

If test error is significantly worse, bad.

State: What's relevant about the past to predict the future.

Might not know state.

Guess: \( \mathbf{x}_{t+1} \)

Another guess: \( \mathbf{x}_{t-1} \mathbf{B} \mathbf{u}_{t-1} \)

Yes!

Another guess: \( \mathbf{x}_{t+2} \)

Another guess: \( \mathbf{x}_{t+4} \)

Try all of these.

And choose the best one.

We have a model. What now?

Divide our goal into 2 parts:

1. Planning: Choose inputs \( \mathbf{u}(t) \) to achieve our goal assuming the model is perfectly correct & noise inputs.

2. Feedback Control: Responsive execution. Achieve goal despite disturbances.
Key Concept: Stability

\[ x[t+1] = x[t] + w[t] \]

\[ \uparrow \text{known} \uparrow \text{going to be our choice} \]

Assume \( w[t] = 0 \) for now.

Are bad things going to happen because of \( w[t] \)?

\[ x[t+1] = 2 \times x[t] + w[t] \]

Basic question: Does this “blow up”?

Simple example: \( x[t+1] = 2 \times x[t] \) Does this blow up?

If \( x[0] = 0 \), then we're fine.

\[ x[0] = 10^{-9} \]

\[ x[32] = 2^{32} \times 10^{-9} = 4 \]

\[ x[40] \approx 1000 \]

We don't like growing exponentials.

\[ x[t+1] = \frac{1}{2} \times x[t] \]

Not blowing up.

First attempt to define stability.

“Stay in the box” stability. 

A system is stable if \( \exists K \text{ s.t. } \|x[t]\| \leq K \forall t \geq 0 \)

Clearly \( K \) needs to depend on initial condition.

\[ x[t+1] = 2 \times x[t] + w[t] \rightarrow \text{unstable} \]

\[ x[t+1] = \frac{1}{2} \times x[t] + w[t] \rightarrow \text{stable} \]
Stability seems to depend on $W_i$.

\[ x(t) \]

Need to bound $W_i$.

**New Definition:** Bounded-input Bounded-output stable (BIBO)

"$x(t)$ stays in a box as long as $|w(t)|$ is not too big."

The system is BIBO-stable if $\forall \epsilon > 0 \exists K > 0$ s.t.

\[
\|w(t)\| \leq \epsilon \Rightarrow \|x(t)\| \leq K \cdot \epsilon.
\]

\[ x(t+1) = x(t) + w(t) \]

**Case 1:** $|x| > 1$.

Not stable

Consider $x(0) = 0$, $w(t) = \epsilon$, $w(t = 0) = 0$

\[
x(t) = ?.
\]

$\begin{align*}
  x(\hat{t}) &= 0 \\
  x(\hat{1}) &= \epsilon \\
  x(\hat{2}) &= x \epsilon \\
  x(\hat{3}) &= \epsilon^{\hat{2}} \\
  x(\hat{t}) &= x^{n-1} \epsilon \\
  \|x(t)\| &= \|x^{n-1}\| \epsilon \\
  \lim_{t \to +\infty} &\to \infty \text{ since } |x| > 1
\end{align*}
\]
Case 2: $|x| < 1 \quad \in \text{Guess stable}$

Recall Discussion 2A or 6A:

$$x_k^+ = x^+ x_0 + \sum_{k=0}^{\infty} x^k \omega [t-1-k]$$

Want $|x_k^+| < \delta$

Consider $|x_k^+| = |x^+ x_0 + \sum_{k=0}^{\infty} x^k \omega [t-1-k]|$

I remember triangle inequality, $|A + B| \leq |A| + |B|$

$$|x_k^+| \leq |x^+ x_0| + \sum_{k=0}^{\infty} |x^k \omega [t-1-k]|$$

I know $|x_0^+| \leq \epsilon \left| \omega (-1) \right| \leq \epsilon$

$$|x_0^+| \leq \epsilon \cdot |x^+| + \epsilon \sum_{k=0}^{\infty} |x^k|$$

$$\leq \epsilon \sum_{k=0}^{\infty} |x^k| = \frac{\epsilon}{1 - |x^+|} < \delta$$

$\implies \text{stable!}$

One case left: $|x| = 1 \quad x = e^{i\theta}$ for some real $\theta$