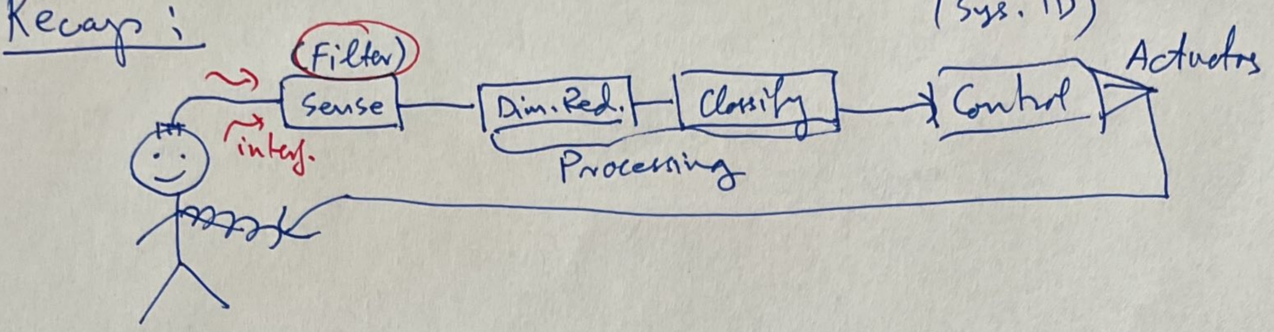
EECS16B Wrap-up

Recap:



Design perspective:

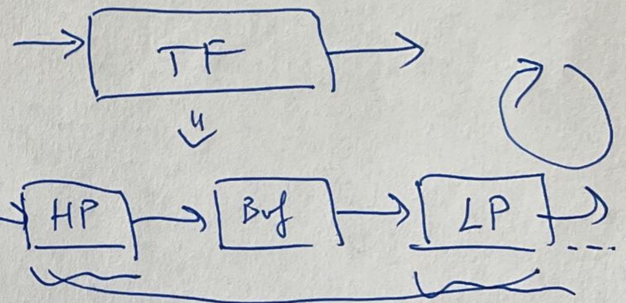
e.g. Filter

Step 1: Specifications

Attenuate interf. 100x while keeping the signal intact.

Step 2: Game-plan (strategy) System-level.

- helps derive the sub-block specs.



Step 3: Implementation (Circuits)

- may evaluate some candidates

RC
RL

Step 4: Validation (Si) Verification (pre-Si):

Does the design meet the specification over all manufacturing variations & operating conditions.

EECS16B material view (chronological) l2

Motivation

Why don't computers work ∞ fast?

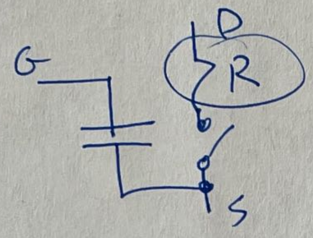
- why are they heating up?

- average AI training run ~1-10MW & lasts for days

Modeling

Logic: consist of transistors.

Transistor: NMOS/PMOS modeled as switches + R & C



RC-time constants & delays

Power spent on R

Math

Diff. equations.

$$\frac{d}{dt}x(t) = \lambda x(t) + u(t)$$

* Solve by guess & check

* Based on uniqueness proofs

Transformations:

1) Change of variables

2) Discretization & limits

L3

Motivation

How can we filter interference & keep the signal ?

Modeling

- * Analog information processing with filters
- * Inductor as a "dual" to capacitor
- * Impedance idea
  - reduces the R, L, C circuit to look like a resistive ckt
- * Transfer functions
  - * Low-pass
  - * high-pass
  - * bandpass
  - * Magnitude plots
  - * Phase plots
  - Composition of filters (TFs)
    - * Multiply TFs (series)
    - * sometimes add (parallel)

Bode-plot approximations

Math

Systems of diff-eqns.

Transformations:

- * Vector "change-of-variables"
  - coordinate changes
  - Eigenbasis
    - * complex evalues
    - * complex evectors

s-phasors
s-impedances
phasors &
impedances ($j\omega$)

$\Rightarrow$ Reduce the sys. of diff-eqs. to a sys of lin. eqs.

$\Downarrow$

* Transfer functions

Another perspective: By themes & patterns

(1) Reductions (the most important theme of 16B!)

Turn the problem you don't yet know how to solve into a problem(s) you know how to solve.

know

- Derivatives $\xRightarrow[\text{Guess \& check}]{}$ Solve homog. diff. eqn.
- $\xRightarrow[\text{change-of variables}]{}$ non-homog. with const. input
- $\xRightarrow[\text{piece-wise \& limit}]{}$ cont. input

$\vec{x} = V\vec{\tilde{x}}$

$$\begin{tikzcd}
\vec{x}: \arrow[r, "A"] \arrow[d, bend right, "V^{-1}"'] & \frac{d}{dt}\vec{x}(t) \arrow[d, bend right, "V^{-1}"'] \\
\vec{\tilde{x}}: \arrow[u, "V"'] \arrow[r, "\tilde{A}=V^{-1}AV"'] & \frac{d}{dt}\vec{\tilde{x}}(t) \arrow[u, bend right, "V"']
\end{tikzcd}$$

Eigenbasis (when it works): reduces to a set of parallel problem (scalar)

Orthonormal basis: Preserve length, angles easy inverses (conj. transpose) easy projections.

Phasors are what mix of transformations?

- coordinate change
- linearization (if needed)
- Eigenbases.
- (out-of-scope) orthonormal basis