

# 6.01 Intro to EECS



- Prereq: [8.02]  
U (Fall, Spring)  
2-4-6 1/2 Institute LAB
- **An integrated introduction to electrical engineering and computer science, taught using substantial laboratory experiments with mobile robots. Key issues in the design of engineered artifacts operating in the natural world: measuring and modeling system behaviors; assessing errors in sensors and effectors; specifying tasks; designing solutions based on analytical and computational models; planning, executing, and evaluating experimental tests of performance; refining models and designs. Issues addressed in the context of computer programs, control systems, probabilistic inference problems, circuits and transducers, which all play important roles in achieving robust operation of a large variety of engineered systems.**
- 6 Engineering Design Points
- H. Abelson, L. P. Kaelbling, J. K. White

# 6.02: Intro to EECS II

- Prereq: 6.01; 18.03 or 18.06  
U (Fall, Spring)  
2-3-7 1/2 Institute LAB
- **An integrated introduction to electrical engineering and computer science, taught using substantial laboratory experiments that explore communication signals, systems and networks.** Physical characterization and modeling of transmission systems in the time and frequency domains; analog and digital signaling; coding; detecting and correcting errors; relating information transmission rate to signal power, bandwidth and noise; engineering of packet-switched networks. These explorations are used to illustrate the role of abstraction and modularity in engineering design; building reliable systems using imperfect components; selecting appropriate design metrics; choosing effective representations for information; analyzing the performance and correctness of algorithms; and tradeoffs in complex systems.
- 6 Engineering Design Points
- C. G. Sodini, C. J. Terman, M. H. Perrott, H. Balakrishnan

# Introduction to EECS 1 - 6.01

Abstraction in Multiple Contexts for Complex System Design

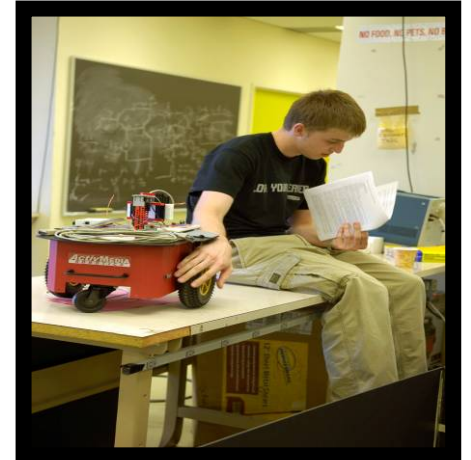
Concepts strengthened in heavily staffed Robotic Design Labs

Skills re-enforced through the opportunity to teach

**Recursion, Objects, Control, Circuits, Planning, Estimation**



- The Weekly Experience
  - 90 minute lecture
  - 90 minute post-lecture software lab
  - 3 hour open-ended design lab in pairs
  - 4 to 1 student-staff ratio for labs!
  - At least  $\frac{1}{4}$  EECS Undergrads get to Teach!



---

# Introduction to EECS I

## 6.01

---

Hal Abelson  
Leslie Kaelbling  
Jacob White

# Intellectual Themes

---

- Controlling software complexity
- Modeling and controlling physical systems
- Managing the interaction of computing artifacts with physical systems
- Building and using models in the face of perceptual and control uncertainty

# Organizing View

Domain	Primitives	Means of combination	Means of abstraction	Example application
Data	numbers, strings	lists, dictionaries, objects	abstract data types, classes	processing sonar readings
Software processes	*, +, =	if, while, $f(g(x))$	procedures, classes	utility functions, constraint resolver
Circuits	resistors, $V_{\text{sources}}$ , opamps	circuits; KVL, KCL	Thevenin equivalence	photoresistor divider; op-amp driving motor (robot head)
Linear Systems	discrete-time signals; difference eqns	composition, feedback	transfer functions, z-transforms	control for driving robot down hall
State estimation	observations and actions	filtering	probability distribution over states	robot position estimation
Dynamic programming	solutions to primitive problems	solution composition	subproblem solutions	robot path planning



# Software Engineering

---

- Concepts
  - lists, procedures, computational complexity
  - higher-order functions
  - object-oriented programming; state machines
- Labs
  - Modeling robot sensors and effectors
  - composing utility functions
  - virtual sensors (averaging, smoothing)
  - finite-state controller

# Linear Systems

---

- Concepts
  - difference equations and state
  - z-transforms
- Labs
  - solving D.E.'s; model of simple unstable robot controller
  - manipulating transfer functions; stabilize previous week's robot controller

# Circuits

---

- Concepts

- constraint model of systems; KVL and KCL
- Thevenin equivalents; 2-port models; op-amps
- designing control systems; analog vs digital

- Labs

- constraint resolution; identify unknown resistor net; use voltage divider to make sensor
- difference eqn model of op-amp; driving motor with op-amp circuit
- add rotating head to robot; combine analog and digital control systems

# Algorithms and Systems

---

- Concepts

- discrete probability and state estimation
- abstracting continuous probability models
- search, dynamic programming

- Labs

- state estimation in simulated discrete worlds
- robot localization based on noisy sonar
- robot path planning and execution

Grand finale: locate lights in a maze using head, light sensors, analog and digital control, state estimation, and planning

# Pedagogical Methods

---

- Practice - theory - practice
- Open ended problems; no right answer
- Supervised labs
- Small student/staff ratio
- Undergraduate Lab Assistants teach and learn
- Week's organization
  - Tuesday lecture 1.5 hours
  - Wednesday TA hours
  - Thursday: pre-lab due, 4-hour lab
  - Tuesday: post-lab due



---

# Introduction to EECS II

## 6.02

---

Charles G. Sodini  
Chris J. Terman  
Michael H. Perrott  
Hari Balakrishnan

# Intellectual Themes

---

- Role of abstractions and modularity in engineering design
- Building reliable systems using imperfect components
- Selecting appropriate design and evaluation metrics
- Limits on channel capacity imposed by energy and noise
- Choosing effective representations for information
- Analyzing the performance and correctness of algorithms
- Engineering tradeoffs in complex systems

# Analog/Continuous Time

---

## Volt-By-Volt

### Concepts

- Fourier Series, Frequency and time domain
- Modulation/Demodulation Filtering with difference equations, I-Q transmission
- Wireless propagation, dipole antennas, relation of energy transmission to data rate

### Labs

- Analyze and Synthesize music in time & frequency domains
- Receiving Frequency Division Multiplexed (FDM) signals
- 4 dipole antenna array demonstrating directional transmission



# Digital/Discrete Time

---

## Bit-By-Bit

### Concepts

- Digital Abstraction, Digital representation of signals, A/D, D/A
- Digital modulation, Pulse amplitude modulation, I-Q, QAM, eye diagrams
- Fixed length codes, Hamming distance, error detection and correction codes, Reed-Solomon codes
- Entropy, variable-length codes, lossless compression, Perceptual codes

### Labs

- Analyze sampled signals, anti-alias and reconstruction filtering
- Digital receiver, design 16 QAM receiver
- Formulate and test error correction code algorithms
- Implement and evaluate Huffman and LZW compression



# Network Architecture

---

## Packet-By-Packet

### Concepts

- Network topologies, channel sharing, performance metrics
- Multi-hop networks, addressing/routing in simple grid network, simulation to predict performance, shortest path routing in arbitrary network
- Distributed algorithms, distributed implementation of Dijkstra algorithm
- Abstraction in engineering, layering of network abstractions

### Labs

- Throughput of circuit- and packet-switched link, queue length as function of utilization
- Implement and evaluate centralized shortest-path algorithm for arbitrary network
- Implement & evaluate distributed shortest-path algorithm for arbitrary network



# Course Description

---

EECS 2 gives a broad exposure to many EECS topics by investigating how information (e.g., voice and data) is transmitted over a wireless or wired network. The course is divided into three major sections dealing with issues at three levels of abstraction: volt-by-volt (analog), bit-by-bit (digital), and packet-by-packet (network architecture).

The course is organized as week-long modules, each consisting of

- \* a lecture that lays the foundation for the week's topic
- \* a homework assignment preparing students for the week's lab
- \* a 3-hour lab involving both analysis and design
- \* a homework assignment involving the lab results
- \* a lecture that closes the loop on what happened in lab
- \* a short in-class concept "quiz"

