CENG 215

Circuits & Electronics

Introduction to Systems Engineering through Electrical Systems

Course Syllabus

Academic Year:	2026–2027
Semester:	Fall Semester
Course Level:	Undergraduate
Department:	Computer Engineering

Understanding, modeling, analyzing, and designing engineering systems through electrical abstractions

Course Description

This course introduces the fundamental principles of Systems Engineering through electrical systems. Rather than studying electrical circuits as an isolated discipline, students learn how engineering systems are modeled, analyzed, and designed using electrical circuits as an accessible and mathematically rigorous example.

The course develops a unified understanding of abstraction, system modeling, state, dynamics, feedback, modularity, and hierarchical design before transitioning to digital hardware implementation using MOSFETs and CMOS logic. The ultimate goal is to establish systems thinking as a foundation for later Computer Engineering courses such as Logic Design, Computer Architecture, Embedded Systems, Control, and Distributed Systems.

Course Philosophy

Throughout the semester, students repeatedly address one central question:

How can we understand, model, analyze, and design engineering systems?

Electrical systems provide an intuitive yet mathematically precise platform for developing these concepts. The course gradually builds from physical systems to dynamic behavior, then to system design principles, and finally to digital hardware abstraction.

Course Learning Outcomes

Upon successful completion of this course, students will be able to:

CLO1: Explain the fundamental concepts of systems engineering, including abstraction, interfaces, hierarchy, modularity, and decomposition.

CLO2: Model engineering systems using mathematical representations derived from physical laws.

CLO3: Analyze static and dynamic electrical systems using appropriate analytical techniques.

CLO4: Explain the concepts of state, energy storage, transient response, stability, and feedback in dynamic systems.

CLO5: Apply systems thinking principles to engineering problem solving.

CLO6: Explain nonlinear behavior and the motivation for digital abstraction.

CLO7: Describe MOSFET operation and CMOS logic implementation.

CLO8: Explain how digital hardware realizes higher-level computer systems.

CLO9: Connect electrical system concepts with later Computer Engineering topics, including Logic Design, Computer Architecture, Embedded Systems, and Control.

Textbook

Authors: Anant Agarwal and Jeffrey H. Lang

Title: *Foundations of Analog and Digital Electronic Circuits*

Publisher: Morgan Kaufmann

Year: 2005

Additional Resources

Supplementary lecture notes, simulation materials, laboratory instructions, and additional readings will be provided through the course learning management system.

Learning Journey

The course is organized around four progressive stages:

1. Understanding engineering systems.
2. Understanding system dynamics.
3. Designing robust engineering systems.
4. Transitioning from analog systems to digital computing systems.

Assessment and Grading

Student performance will be evaluated through one midterm examination, one final examination, and laboratory work. The final course grade will be calculated according to the following distribution:

Assessment Component	Weight
Midterm Examination	35%
Final Examination	45%
Laboratory Work	20%
Total	**100%**

$$G_{\text{course}} = 0.35\, G_{\text{midterm}} + 0.45\, G_{\text{final}} + 0.20\, G_{\text{laboratory}}.$$

Weekly Schedule

Week	Main Topic	Systems Engineering Concepts	Weekly Learning Outcomes
1	Introduction to Systems Engineering	Systems, abstraction, hierarchy, engineering systems	Explain the concept of a system; distinguish among system, subsystem, environment, and boundary; explain abstraction and hierarchy; interpret electrical circuits as engineering systems.
2	Modeling Engineering Systems	Variables, conservation laws, mathematical models	Model engineering systems using variables and physical laws; apply Kirchhoff's laws as manifestations of conservation principles.
3	Static Systems	Linear systems, decomposition, equivalent models	Analyze resistive networks; simplify engineering systems using equivalent representations and systematic decomposition.
4	Dynamic Systems	State variables, energy storage	Explain the concept of system state; identify energy-storage mechanisms; model dynamic behavior using capacitors and inductors.
5	First-Order Dynamic Systems	Transient response, time constants	Analyze first-order system behavior; explain transient, steady-state, and time-constant concepts.
6	Second-Order Dynamic Systems	Oscillation, damping, natural response	Analyze second-order systems; explain natural response, damping, and oscillatory behavior.

Continued on the next page

Weekly Schedule continued from the previous page

Week	Main Topic	Systems Engineering Concepts	Weekly Learning Outcomes
7	Feedback and Stability	Feedback, robustness, sensitivity	Explain positive and negative feedback; discuss stability, sensitivity, and robustness using electrical and engineering examples.
8	System Design Principles	Interfaces, modularity, decomposition	Apply modular design principles; explain interfaces, decomposition, and hierarchical organization.
9	Nonlinear Systems	Threshold behavior, switching	Explain nonlinear system behavior; analyze diode-based threshold and switching circuits.
10	Analog-to-Digital Abstraction	Abstraction, digital representation	Explain the motivation for digital abstraction; describe MOSFET operation using the switch abstraction.
11	Digital System Implementation	Hierarchy, hardware realization	Explain CMOS inverter operation; analyze the transistor-level implementation of basic CMOS logic gates.
12	Building Digital Systems	Composition, digital design	Construct larger digital systems from logic gates; explain the composition of combinational building blocks.
13	Mixed-Signal Systems	System interfaces	Explain analog-to-digital and digital-to-analog conversion; analyze interactions between analog and digital subsystems.
14	Integrated Engineering Systems	System integration, engineering perspective	Integrate concepts from the entire course; explain how systems engineering connects physical electrical systems to computer engineering.

Laboratory and Recitation Component

The course includes a two-hour weekly laboratory/recitation period in addition to the three hours of lectures. The first week is reserved for course introduction and software preparation, and one week is reserved for the midterm examination. The remaining twelve weeks alternate between laboratory work and problem-solving recitations.

Laboratory sessions emphasize modeling, numerical simulation, verification, and system-level interpretation. Recitation sessions reinforce analytical problem solving and connect mathematical models across electrical, mechanical, thermal, and other engineering domains.

Week	Session	Topic	Main Activity / Tools
1	Preparation	No formal laboratory	Course orientation and software setup: Python/Jupyter, NumPy, SciPy, Matplotlib, and KiCad/ngspice.
2	Recitation 1	Modeling Engineering Systems	Variables, physical laws, conservation principles, mechanical and thermal models, and Kirchhoff's laws.
3	Lab 1	Static System Modeling	Resistive-network modeling; nodal equations; Python/NumPy solution; parameter study; KiCad/ngspice verification.
4	Lab 2	Dynamic System Modeling	RLC dynamic model; Forward Euler; SciPy `solve_ivp`; time-step analysis; transient SPICE verification; mechanical analogy.
5	Recitation 2	First-Order Systems	RC/RL, thermal, and mechanical first-order problems; time constants, equilibrium, and transient response.
6	Lab 3	Second-Order Systems	Second-order dynamic response; effects of physical parameters on oscillation and damping; numerical and SPICE comparison.
7	Recitation 3	Feedback and Stability	Feedback, sensitivity, stability, and robustness problem solving.
8	Midterm	Midterm Examination	No laboratory or recitation activity.
9	Lab 4	Nonlinear Systems	Diode-based nonlinear and threshold models; comparison of linear and nonlinear behavior.
10	Lab 5	Analog-to-Digital Abstraction	MOSFET switching and transistor-level behavior; connection between continuous electrical signals and digital abstraction.
11	Recitation 4	Digital System Implementation	CMOS inverter and basic CMOS gates; transition from transistor-level models to logic-level abstraction.
12	Lab 6	Building Digital Systems	Gate-level composition; small digital subsystems; modularity, interfaces, and hierarchy.
13	Lab 7	Mixed-Signal Systems	Analog–digital interfaces, sampling and/or conversion, and mixed-signal system analysis.
14	Recitation 5	Integrated Engineering Systems	Integrated problem solving across static, dynamic, feedback, nonlinear, and digital systems.

Table 2: Laboratory and Recitation Schedule

Laboratory performance constitutes 20% of the final course grade. Graded laboratory work may include preparation, in-session modeling and simulation, computational notebooks, verification results, short analyses, and instructor checkpoints. Recitation sessions are primarily intended for guided practice and reinforcement of lecture material.

Course Narrative

The course intentionally follows a continuous engineering story.

Students first learn how engineering systems are represented and modeled. They then examine how systems evolve over time and how their dynamic behavior can be analyzed. This foundation leads to the study of feedback, stability, robustness, modularity, and system design. Finally, students discover how digital hardware emerges as a carefully constructed abstraction over physical electrical systems.

Rather than viewing electrical circuits as an isolated subject, students develop systems thinking that serves as a conceptual foundation for the remainder of the Computer Engineering curriculum.

Academic Integrity

Students are expected to comply with the academic integrity regulations of İzmir Institute of Technology. All submitted work must represent the student's own effort unless collaboration is explicitly permitted.

Unauthorized collaboration, plagiarism, fabrication of laboratory results, and the use of unapproved materials or tools during examinations are not permitted.

Attendance and Examination Policy

Attendance requirements, eligibility conditions for examinations, make-up examination procedures, and other administrative matters are governed by the applicable undergraduate education regulations of İzmir Institute of Technology.

Students are responsible for following all course announcements and scheduled assessment dates.

Generative Artificial Intelligence (GenAI) Policy

Generative Artificial Intelligence (GenAI) tools (e.g., ChatGPT, Claude, Gemini, Copilot, or similar systems) may be used as learning aids throughout this course. Students are encouraged to use these tools to explore concepts, brainstorm ideas, improve writing, or receive explanations of technical topics.

However, GenAI tools are intended to support learning—not replace it. Students remain fully responsible for the accuracy, originality, and technical correctness of every statement, calculation, figure, design decision, piece of code, and conclusion submitted as their own work.

Students should therefore adhere to the following principles:

- Every submitted solution must reflect the student's own understanding.
- Students must be able to explain and justify every part of any submitted work, regardless of whether GenAI was used during its preparation.
- The use of GenAI does not transfer responsibility for errors, misconceptions, or academic misconduct.
- Course staff may ask students to explain, reproduce, or extend any submitted work during examinations, laboratory sessions, or oral discussions.

The goal of this policy is not to discourage the use of modern AI tools, but to promote responsible engineering practice and independent technical judgment.

> *An engineer is ultimately accountable for the systems they design. Likewise, students are accountable for every word, equation, diagram, design decision, and line of code they submit, regardless of the tools used during their preparation.*