CAEDMI
Central Asia Engineering Dentistry Medicine Institute

Alpha Institute of Technology

Course Syllabus

Course Title:	MeRo 5010: Simulation Techniques for Dynamic Systems
Lead Instructor:	Boris Kuznetsov, PhD
Contacts:	boris.kuznetsov@caedmi.com
Office hours:	Wednesday 10.00 - 13.00

1. Course Description

This course introduces the use of software tools for dynamic system modeling, control system analysis, and design. The course focuses on practical applications in real-world dynamic systems, emphasizing model development, validation processes, parameter identification techniques, effective control algorithms, and result presentation. Students will gain hands-on experience with various software tools, such as Octave and Micro Cap 12, and learn to apply them to solve complex dynamic system problems.

2. Basic Information

Course Academic Level:	Master-level
Course Semester:	1st Semester
Number of ECTS credits:	5
Course Prerequisites:	Knowledge of electrical circuits and linear systems control

Type of Assessment: Graded

Mapping from grades to percentage:

Letter Grade	Numeric Value (GPA)	100-point Scale
A	4.0	93-100
A-	3.7	90-92
B+	3.3	87-89
B	3.0	83-86
B-	2.7	80-82
C+	2.3	77-79
C	2.0	73-76
C-	1.7	70-72
D+	1.3	67-69
D	1.0	63-66
D-	0.7	60-62
F	0.0	<60

3. Course Content

№ of week	Topic	Laboratory Sessions
1	Introduction to Dynamic System Modeling - Basic concepts and definitions - Classification of models, examples of modeling applications	Introduction to Micro Cap 12 - Installing and configuring Micro Cap 12 - Basic functions and capabilities
2	Modeling Tools - Overview of software tools for modeling - Introduction to Octave and its modeling capabilities	Micro Cap 12 - Analyzing simple DC circuits, transient processes in reactive circuits, frequency analysis

	- Introduction to Micro Cap 12 and its modeling capabilities	
3,4	Modeling Electrical and Electronic Circuits - Theoretical foundations of electrical system modeling, topological and component models - Basic analysis methods: DC analysis, transient analysis, AC analysis, frequency analysis, harmonic distortion analysis	**Comprehensive Analysis of Electrical Circuits Using** Micro Cap 12
5,6	Modeling and Analyzing Electrical Circuits in Micro Cap 12 - Setting up and performing DC, transient, AC, frequency, and harmonic distortion analysis - Interpreting results	Modeling MOSFET Output Stages in Micro Cap 12 - Creating and analyzing models of output stages
7	Modeling Embedded System Components - Modeling and analyzing microcontroller output stages - Calculating and analyzing signal lines - Analyzing the operation of PWM converters	Analyzing PWM Converters in Micro Cap 12 - Creating and analyzing models of PWM converters
8,9	System Modeling in Octave - Basic mathematical functions in Octave - Vectors and matrices in Octave - Matrix operations - Creating models of mechanical systems - Creating models of electrical systems - Analyzing time and frequency characteristics	Modeling Mechanical Systems in Octave - Creating and analyzing models of mechanical systems
10	Control Systems, Synthesis, and Modeling - Introduction to control systems - PID controller: principles, tuning, application - Real-world examples	Modeling Electrical Systems in Octave - Creating and analyzing models of electrical systems (complex DC circuits)
11	Stability Analysis of Dynamic Systems in Octave - Stability analysis methods - Applying methods to various models	Working with COM Ports in Octave - Connecting and configuring COM ports - Receiving data via COM port
12	Data Visualization - Data visualization methods in Octave	Software Development for AVR Microcontrollers

	- Plotting graphs and diagrams - Presenting modeling results	- Creating and testing programs for the Arduino platform
13	Interfacing Octave with External Systems - Connecting and configuring COM ports in Octave - Receiving data via COM port - Usage examples	Software Development for Raspberry Pi - Creating and testing programs for Raspberry Pi - Interfacing with peripheral devices and sensors
14	Software Development for AVR Microcontrollers and Raspberry Pi - Basics of programming AVR microcontrollers - Programming the Arduino platform - Developing software for Raspberry Pi - Interfacing with peripheral devices and sensors	Developing and Interfacing Embedded Systems with AVR Microcontrollers and Raspberry Pi
15,16	Control Algorithm Development - Creating and testing control algorithms in Octave - Applying algorithms to dynamic system models	Developing and Interfacing Embedded Systems with AVR Microcontrollers and Raspberry Pi

4. Learning Outcomes

- Understand the Fundamentals of Dynamic System Modeling:
  - Grasp the basic concepts, classifications, and applications of dynamic system models in engineering and science.
- Apply Modeling Tools and Software:
  - Proficiently use software tools such as Octave and Micro Cap 12 for the modeling and analysis of dynamic systems, including electrical, mechanical, and electronic circuits.
- Analyze and Simulate Electrical and Electronic Circuits:
  - Perform comprehensive analyses of electrical systems, including DC, transient, AC, frequency, and harmonic distortion analyses using Micro Cap 12.
  - Develop and interpret models of electrical and electronic circuits, including MOSFET output stages and PWM converters.
- Model and Analyze Embedded System Components:
  - Create and analyze models of embedded system components such as microcontroller output stages and signal lines, utilizing both theoretical knowledge and simulation tools.
- Develop Control Algorithms and Stability Analysis:

    - Synthesize and implement control algorithms, including PID controllers, for dynamic systems.
    - Conduct stability analyses of dynamic systems using Octave and apply these methods to various real-world models.
- Integrate System Modeling with Data Visualization and External Interfaces:
    - Develop models in Octave for mechanical and electrical systems, and effectively visualize data using graphs and diagrams.
    - Interface Octave with external systems, including configuring and using COM ports for data exchange.
- Design and Implement Embedded Systems:
    - Program AVR microcontrollers and Raspberry Pi for embedded system applications, including interfacing with peripheral devices and sensors.
    - Develop software for embedded systems that integrate control, sensing, and communication components.

5. Assignments and Grading

Assignment Type	% of Final Course Grade
Laboratory Work	40
Midterm Tests	20
Final Project	40

6. Assessment Criteria

<u>Laboratory Work</u>
Task Completion Accuracy: This includes evaluating the correctness of laboratory task implementation, adherence to technical requirements and specifications, and minimizing errors in code and circuit design.
Report Documentation: The quality and completeness of the laboratory report, including the description of methods used, results obtained, and conclusions drawn.
Teamwork: For group work, the contribution of each team member, interaction, and task distribution are evaluated.
Punctuality: Timely submission of laboratory work also affects the final grade.

<u>Midterm Tests:</u>
Competency in Theoretical Material: This will be assessed through test scores reflecting the understanding of key concepts covered in lectures and the ability to utilize them.
Problem-Solving: The capability to apply theoretical knowledge to solve practical problems in programming and microcontrollers.
Reasoning Quality and Justification: The quality of reasoning behind chosen solutions and the ability to explain the approach to problem-solving.

<u>Final Project:</u>
Complexity and Depth: Evaluation of the project's complexity, thoroughness in working out components, and how comprehensive the final solution is.

Functionality: Assessment of how accurately and reliably the developed system operates, and whether the project meets its objectives.
Innovation: The originality and novelty of the project idea and approach, along with the use of modern and innovative solutions.
Documentation Quality: The readiness and depth of project documentation, including technical implementation descriptions, diagrams, code, and testing results.
Project Presentation: The ability to clearly present the project, explain goals, approaches, and results, and effectively answer questions from the audience.

7. Textbooks and Internet Resources

Required Textbooks:

1) Micro-Cap 12 Electronic Circuit Analysis Program Reference Manual
2) Andreas Stahel Octave and MATLAB for Engineers, Bern University of Applied Sciences, Switzerland

Recommended Textbooks:

John W. Eaton David Bateman Søren Hauberg Rik Wehbring Free Your Numbers

8. Facilities

Required Course Materials (software or equipment):

- Micro-Cap 12
- GNU Octave

Optional:

- Raspbian (Raspberry Pi OS)
- Arduino IDE

9. Additional Notes

Project Examples:

Design and Simulation of a PID-Controlled Temperature System:

- Develop a model of a temperature control system using a PID controller. Simulate the system in Octave, and analyze its stability and performance under different conditions.

Modeling and Analysis of a DC Motor Control System:

- Create a dynamic model of a DC motor, including its electrical and mechanical components. Use Micro Cap 12 to simulate the motor's behavior under various load conditions and design a control system to optimize performance.

Simulation of a Multi-Stage Amplifier Circuit:

- Design a multi-stage amplifier circuit in Micro Cap 12. Perform DC, AC, and transient analysis to evaluate its performance, and optimize the design for specific gain and bandwidth requirements.

Development of a Smart Home Lighting System Using PWM Control:

- Model a smart lighting system using PWM control for brightness adjustment. Simulate the system in Micro Cap 12, and implement the control algorithm on an AVR microcontroller, interfacing it with sensors and user input.

Modeling and Simulation of an Autonomous Vehicle Steering System:

- Develop a model of an autonomous vehicle's steering system, incorporating sensor data and control algorithms. Simulate the system's response to various driving conditions using Octave, and analyze its stability and performance.

Design of a Wireless Sensor Network for Environmental Monitoring:

- Create a model of a wireless sensor network for monitoring environmental conditions such as temperature, humidity, and air quality. Use Raspberry Pi and AVR microcontrollers for data acquisition and processing, and simulate the network's performance.

Simulation and Control of a Solar Power System:

- Model a solar power system, including photovoltaic cells, inverters, and storage components. Use Octave to simulate the system's energy output under different environmental conditions and develop a control strategy for efficient energy management.

Design and Analysis of a Power Supply with Overcurrent Protection:

- Create a model of a power supply circuit with overcurrent protection using Micro Cap 12. Simulate the circuit's response to fault conditions and optimize the protection mechanism to prevent damage to the system.

Development of a Robotic Arm Control System:

- Model the dynamics of a robotic arm and design a control system for precise movement. Use Octave to simulate the arm's response to control inputs, and implement the control algorithms on an embedded system.

Simulation of a Communication System Using UART Protocol:

- Develop a model of a communication system using the UART protocol. Simulate data transmission between microcontrollers and analyze the system's performance under different noise and interference conditions.

Practical Sessions:

- Weekly laboratory sessions related to lecture topics.
- Continuous project work starting from week 5.