Master Course Syllabus for EE 347 (ABET sheet)

Title: Introduction to Robotics and Control Systems

Credits: 4 (changing it to 5?)

[link]

UW Course Catalog Description

This course introduces students to the fundamentals of robotics and control systems, focusing on the modeling, design, and control of robotic systems. Topics include forward and inverse kinematics, control theory, robot sensing, navigation, and path planning. Students start with Blockly for basic control concepts before progressing to the Python Control Library for more advanced applications. The course also incorporates simulations using MATLAB, alongside hands-on labs where students program and control a physical robot arm. Prerequisite: CSE 142 and CSE 143 (or equivalent) and Matrix Algebra. Offered: Autumn/(maybe)Spring

Coordinators: S. Makhsous, Electrical and Computer Engineering

(Team) Faculty who have or are willing to teach this core course:
Sam Burden, Blake Hannaford, Kim Ingraham, Sep Makhsous

Goals: To equip students with a comprehensive understanding and practical skills in robotics and control systems, focusing on the analysis, modeling, and control of robotic systems. Through detailed exploration of fundamental concepts such as kinematics, control theory, and path planning, students will gain the ability to design, simulate, and manipulate robotic systems for a variety of applications. This course emphasizes hands-on experience with physical robot arms, programming with Python, and simulations using Simulink in MATLAB, preparing students for more advanced topics in control systems and robotics.

Learning Objectives: At the end of this course, students will be able to:

1. *Describe* robotic systems using forward and inverse kinematics, including Denavit-Hartenberg (DH) parameters for systematic kinematic analysis.
2. *Understand and implement* Jacobian matrices for analyzing differential motion in robotic manipulators.
3. *Analyze and design* control strategies for robotic systems, understanding the dynamics of manipulators, including the effects of forces and torques.
4. *Plan and generate* robot trajectories in both joint and Cartesian spaces, applying principles of motion planning to avoid obstacles.
5. *Understand* the implications of robotic sensing and localization techniques and apply them to real-world navigation and path planning problems.

For the Hands-on Course Laboratories:

6. *Implement* kinematic models of robot arms using Python and MATLAB, including forward and inverse kinematics with DH parameters.

7. *Control* physical robotic manipulators and tune their performance using Python Control Library.
8. *Simulate and evaluate* robot dynamics and control strategies using MATLAB Simulink, incorporating dynamic and kinematic principles.
9. *Design and execute* motion planning algorithms, including trajectory generation and obstacle avoidance, on a physical robot arm.
10. *Apply* manipulator dynamics to simulate forces, torques, and system behavior under varying conditions.

Textbook:
Models of Robot Manipulation by Blake Hannaford, 2022

Referenced Textbooks:
Introduction to Robotics: Mechanics and Control by John J. Craig, 4th Edition, Pearson, 2018.
Control Systems Engineering by Norman S. Nise, 7th Edition, Wiley, 2015.
Modern Robotics: Mechanics, Planning, and Control by Kevin M. Lynch and Frank C. Park, Cambridge University Press, 2017.
Feedback Control of Dynamic Systems by Gene F. Franklin, J. Da Powell, and Abbas Emami-Naeini, 8th Edition, Pearson, 2021.

Prerequisite courses:
CSE 122, CSE 123 (or equivalent)
EE 241
Matrix Algebra (MATH 208)

Prerequisites by Topic:
1. Linear algebra
2. Basic programming in Python or MATLAB

Lecture Topics:
1. Introduction to Robotics and Control Systems (1 week)
2. Forward Kinematics and Denavit-Hartenberg (DH) Parameters (2 weeks)
3. Inverse Kinematics (2 weeks)
4. Jacobian Matrices and Differential Motion (1 week)
5. Manipulator Dynamics (1 week)
6. Control Strategies for Robotic Systems (1 week)
7. Trajectory Generation in Joint and Cartesian Spaces (1 week)
8. Motion Planning and Obstacle Avoidance (1 week)

Laboratory Topics:
As scheduling per quarter permits
- Lab 1: Setup the Python Environment and Foundations of Robotics Programming
- Lab 2: Forward Kinematics and Drag Teaching
- Lab 3: Inverse Kinematics and Pose Vectors
- Lab 4: Final Project - Industrial Application Simulation

Note: Labs are unlikely to always directly follow the content presented in class, but content for each lab is covered prior to students beginning the lab, even if the lab lags 1-2 weeks behind the lecture material.

Course Structure: The class meets four times a week, with each lecture lasting 50 minutes, or twice a week for 110-minute sessions. Students are assigned weekly homework and complete several laboratory exercises, all of which must be done using Python. A total of 12 hours per week is dedicated to lab sessions, which are facilitated by TAs. These lab sessions are open to all students, with no formal registration required, allowing students the flexibility to attend and work on their labs as needed.

The structure and delivery of EE347 lectures are at the instructor's discretion. Typically, lectures include a combination of theoretical explanations, practical examples that demonstrate the application of key concepts, and live demonstrations of these principles in real-world devices. Additionally, lectures often incorporate student-centered activities, such as small group problem-solving and think/pair/share exercises, to help reinforce understanding and address misconceptions.

Assignments in the course consist of homeworks, lab assignments (based on scaffolded laboratory reports), and exams/quizzes.

Computer Resources: The course uses Python for the laboratory exercises and optionally for checking homework problems. Students are expected to use their personal computers in the labs. Outside of the two-hour lab section, students spend an additional hour per week on average to complete the labs, including prelab assignments and lab reports.

Laboratory Resources: (see Computer Resources)

Grading: Formative assignments include homeworks, in-class exercises, and (in part) laboratory reports. Summative assignments include midterms and final exams. The course syllabus and grading rubrics will clearly identify the goal of each assignment (formative or summative).

- Homework+In-Class Exercises: 40%
- Laboratory Reports: 30%
- Exams: 30% (2 midterms)

It is highly recommended that part of the lab assignment grading include a direct, face-to-face laboratory demonstration that verifies that each student is learning the hands-on laboratory skills for which the course is designed and minimizes academic dishonesty. If students work in teams to complete the laboratories, a means for ensuring that each student contributes something meaningful to each laboratory report is strongly recommended. Homeworks may be completed in teams or solo (at the discretion of the instructor), while summative assignments (exams, quizzes, etc.) should be completed solo.

ABET Student Outcome Coverage: This course addresses the following outcomes:
H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.* (H)
The course introduces fundamental mathematical principles used for the analysis of continuous-time signals and systems. Students routinely solve problems in systems analysis using mathematical tools of convolution and transforms. They are introduced to computer analysis methods via Python-based computer lab assignments.

(3) *An ability to communicate effectively with a range of audiences (M)*

Students are expected to provide clear, concise answers to questions in exams that include only information relevant to the question. In addition, they answer questions about lab assignments orally during laboratory sections and provide written lab reports in electronic notebook format. Some instructors include a brief writing assignment where students are asked to pick an example of modern technology and explain how some aspect of signal processing plays a role in this technology.

(5) *An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.* (M)
The computer labs are conducted in teams. Labs constitute about 20% of their grade (depending on the instructor).

(7) *An ability to acquire and apply new knowledge as needed, using appropriate learning strategies*. (M)
Students are expected to use online documentation to learn the Python programming language for use in lab exercises, building on their knowledge of programming in other languages.

Prepared By: Sep Makhsous
Last Revised: October 26, 2024

Additional information and resources regarding teaching ECE courses (e.g., links to course repositories for materials from previous course offerings; guidelines for using AI tools in courses; syllabus language for course accommodations, etc.) can be found on the UW ECE Intranet:
https://peden.ece.uw.edu/academic-ops/

Justification:
The current ECE curriculum lacks junior-level courses in the control systems and robotics pathway. As a result, students do not gain exposure to control systems until their senior year in ECE 447, a theoretical course that lacks hands-on experience. Additionally, students selecting this pathway have limited opportunities to engage in practical, real-world robotics work until their senior capstone projects. This course fills the gap by providing earlier hands-on exposure to robotics and control systems through the use of physical robot arms and advanced simulation tools. By offering practical experience earlier in the curriculum, this course equips students with essential skills in kinematics, control, and programming, which they can build on in more advanced courses and projects. Furthermore, this course serves as a valuable prerequisite for higher-level robotics and control systems courses, helping students transition smoothly from foundational concepts to more complex real-world applications.