**Course description: Autonomous CyberPhysical Systems**

**Instructor: Jyo Deshmukh**

**Introduction**

Have you wondered how we will accomplish the dream of driverless cars that improve transportation safety and reduce manual effort, or unmanned autonomous aerial vehicles that perform a variety of tasks such as search and rescue, disaster relief, or even mundane package delivery? This course introduces you to the basics of computational models, control algorithms, architectures, and challenge problems for such *Autonomous CyberPhysical Systems*. A CyberPhysical System (CPS) consists of two main parts: (1) physical components (electrical, electronic, mechanical, hydraulic, *etc.*), and (2) software that is used to control the behavior of the physical components. An autonomous CPS is a system that is designed to operate *without human intervention.* Examples of autonomous CPSs include driverless cars, unmanned aerial or submarine vehicles, and humanoid and other autonomous robots.

In this course, we will first study the basics of CPS from a computer science and formal methods/reasoning perspective. This includes: formal models of computation for CyberPhysical Systems, formal languages for specification and testing of CPS systems, and introduction to some common coding platforms for CPS software development. In the second part of the course, we will study the architectures and algorithms used in autonomous CPSs through some important case studies. One of the homework assignments will teach you an end-to-end implementation of a simple self-driving vehicle subsystem. The course will position you to gain the skills required for industrial development of autonomous systems, and will also enable you to think about research problems in autonomy.

**Learning Objectives**

1. Gain basic familiarity with CyberPhysical Systems.
2. Learn how to develop software for a CPS using a model-based development approach.
3. Learn how to write formal requirements for CPS models and perform testing.
4. Learn basics of simulation-based testing and falsification.
5. Learn basics of autonomous systems, and the autonomous systems software stack.

**Course Structure**

In this course, bulk of the teaching will be accomplished through lectures and assigned readings. In addition, we will have the following elements:

1. *Homework assignments*: We will have 3 homework assignments through the semester. One assignment will be written, while two will be programming assignments.
2. *Project*: Students will have to do a project that can be done either in a team of 1 or 2. Each project will have to be approved by the instructor pending a formal project proposal. Informal proposals over email are encouraged prior to the formal project proposal. Projects can be of three types:
3. Research project: Here, the team is expected to choose a fundamental question related to the course material, and identify possible solutions to the problem. The final product should be a 10-page research paper in at most 12-point font, 1” overall margins and at most single spacing.
4. Survey project: Here, the team considers a large body of literature related to a topic and writes a paper summarizing the key research directions and results in that topic. The final product should be a 20-page paper in at most 12-point font, 1” overall margins and at most single spacing.
5. Programming Project: Here, the team can build a tool for analyzing an autonomous CPS, or construct the model of an autonomous CPS. Projects involving hardware implementations following a thorough model-based development process are encouraged; however, support for deployment to hardware will be limited.
6. *Examinations*: The course has no final exam, but we will have two written exams. The purpose of the exams will be to test basic understanding of the material.

**Pre-requisites**

* Automata theory (required), Experience in Simulink/Matlab (preferred but not required).
* This course is aimed at graduate students, but upper-level undergraduates can take this course if interested.

**Texbooks**

The course will use the following textbook:

1. Principles of Cyber-Physical Systems by Rajeev Alur, MIT Press.

Supplementary Reading:

1. Introduction to Embedded Systems - A Cyber-Physical Systems Approach, by Lee & Seshia, Second Edition, MIT Press. <http://leeseshia.org>
2. Cyber-Physical Systems, by Rajkumar, de Niz and Klein, Addison-Wesley.
3. Autonomous Ground Vehicles, by Ümit Özgüner, Tankut Acarman, Keith Redmill, Artech House.

**Readings**

**HO1**: Handout on “Markov Chains, Markov Decision Processes, Continuous-time Markov Chains”

**HO2**: Handout on “CTL, CTL\*, pCTL”

**HO3**: Handout on “Metric Temporal Logic, Signal Temporal Logic”

**HO4**: Handout on “Temporal-Logic Falsification”

**HO5**: Handout on “Introduction to Software for Autonomous Systems”

In addition, all course notes will be made available prior to the lectures in .pdf and .pptx formats.

**Grading and Evaluation**

Students will be graded based on the following breakdown.

|  |  |
| --- | --- |
|  | Weight |
| Homework Assignments | **35%** |
| HW1 (10%) |  |
| HW2 (10%) |  |
| HW3 (15%) |  |
| In-class Examinations | **30%** |
| Exam 1 (15%)  |  |
| Exam 2 (15%) |  |
| Final Project | **25%** |
| Project Proposal Presentation (5%) |  |
| Project/Final Paper (15%) |  |
| Final Presentation (5%) |  |
| Attendance and Class Participation | **5%** |

Project/Paper Grading: Projects will be graded based on the quality of prototype tools, demos, or the developed models. Survey papers will be graded based on their comprehensiveness. Technical papers will be graded based on their novelty and technical results.

**Course Schedule: Weekly Breakdown. (All deadlines indicate before class on that day)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| W | Date | **Topic** | **Readings** | **Milestones** |
| 1 | 1/9 | Introduction, Overview, Synchronous Models | 2.1 |  |
| 1/11 | Synchronous models of Computation | 2.2-3, 2.4.2 |  |
| 2 | 1/16 | Asynchronous models/Timed Models | 4.1.1-5, 4.2.1-3, 7.1 |  |
| 1/18 | Timed Models | 7.2,7.3.1 |  |
|  3 | 1/23 | ***Simulink/Matlab Review*** |  |  |
| 1/25 | Dynamical Systems: Modeling, Numerical Methods | 6.1, 6.4.1 |  |
|  4 | 1/30 | Dynamical Systems: Basics of Control | 6.2, 6.3 |  |
| 2/1 | Hybrid Systems, Stability | 9.1 | HW1 due |
|  5 | 2/6 | Hybrid Systems, Designing | 9.2-3 |  |
| 2/8 | Probabilistic Models | HO1 |  |
|  6 | 2/13 | Safety Requirements and Monitors, Barrier Certificates | 3.1, 6.4.2  |  |
| 2/15 | Temporal Logics (LTL, CTL, CTL\*, pCTL) | 5.1 + HO2 |  |
|  7 | 2/20 | Real-time Temporal Logics | HO3 |  |
| 2/22 | ***Review*** |  | HW2 due |
|  8 | 2/27 | **1st Exam** |  |  |
| 3/1 | Exam discussion, Intro to Verification & Testing | 3.2-4 |  |
|  9 | 3/6 | Model Checking  | 5.2, 5.3, 7.3.2-3 |  |
| 3/8 | Temporal Logic Falsification  | HO4 |  |
|  10 | 3/13 | **No Classes; Spring Break** |  |  |
| 3/15 |  |  |
|  11 | 3/20 | Project Proposal Presentation |  | **Proposals Due** |
| 3/22 | Autonomous Systems Software Stack | HO5 |  |
|  12 | 3/27 | Introduction to Perception |  |  |
| 3/29 | Introduction to Decision/Planning |  |  |
|  13 | 4/3 | Introduction to Low-level Control |  |  |
| 4/5 | Building a Prototype Autonomous Driving Sub-system |  |  |
|  14 | 4/10 | ***Review/Guest Lecture*** |  |  |
| 4/12 | **2nd Exam** |  |  |
|  15 | 4/17 | Unmanned Aerial Vehicles |  |  |
| 4/19 | Medical CyberPhysical Systems |  | HW3 due |
|  16 | 4/24 | Final Project Presentations |  | **Projects Due** |
| 4/26 | Final Project Presentations |  | **Projects Due** |

**Academic Conduct**

From <https://dornsife.usc.edu/engl/academic-conduct/>:

Plagiarism – presenting someone else’s ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Part B, Section 11, “Behavior Violating University Standards” <https://policy.usc.edu/scampus-part-b/>. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, <http://policy.usc.edu/scientific-misconduct>.

Discrimination, sexual assault, intimate partner violence, stalking, and harassment are prohibited by the

university. You are encouraged to report all incidents to the Office of Equity and Diversity/Title IX Office

<http://equity.usc.edu> and/or to the Department of Public Safety <http://dps.usc.edu>. This is important for the health and safety of the whole USC community. Faculty and staff must report any information regarding an incident to the Title IX Coordinator who will provide outreach and information to the affected party. The sexual assault resource center webpage <http://sarc.usc.edu> fully describes reporting options. Relationship and Sexual Violence Services <https://engemannshc.usc.edu/rsvp> provides 24/7 confidential support.

**Note on Collaborative Work**

For collaborative projects, students are expected to have equal distribution. If there is any perceived imbalance in the collaborative project, the student should bring this to the attention of the instructor or the teaching assistant.

**Support Systems for Writing, Learning Disabilities, etc.**

Several USC’s schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute <http://ali.usc.edu>, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs <http://dsp.usc.edu> provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information <http://emergency.usc.edu> will provide safety and other updates, including ways in which instruction will be continued by means of Blackboard, teleconferencing, and other technology.