Syllabus

Course Outline:

----------------------------------*Module 1: Model-Based Design Principles*-------------------------------------

Lecture 1: Model-Based Design Fundamentals

Topics:

- Definitions
  - Model Based Design
  - Realtime Control Systems
  - Cyber-Physical Systems (CPS)
- How this course fits in the CPS landscape
- MBD process in system concept phase
  - Creating a control system overview/synopsis
    - High level textual summary
    - Context diagram(s)
    - Capture system goals
  - Defining control system boundary
    - Control system interactions and interfaces
    - Identify environmental variables, monitored variables, potential controlled and manipulated variables
  - Defining operational concepts/use cases
    - Use case mapping to system goals
    - Exception/failure cases
- Course housekeeping
  - Software availability
  - Procedures

Lecture 2: Analysis in Model Based Design

Topics:

- Model, simulate, characterize physical system
  - Model requirements
  - Model abstraction
  - Model validation
- Capturing control system behavior and performance requirements
  - Dynamic requirements
  - Steady state requirements
  - Uncertainty bounds

  - Translating control system requirements to objective functions
- Developing control system architecture
  - Selecting controlled variables
  - Selecting manipulated variables
  - Cascaded control architectures
  - Identifying system modes
- Verify and validate control system design
  - Uncertainty analysis
  - Simulation
  - Test
- Peer review
- Revise control system requirements, model requirements and models
  - Refine models with test data
- Proceed to next phase/program increment

---------------------------*Module 2: Acausal Modeling of Physical Systems*---------------------------

Lecture 3: Modeling Dynamical Systems Using Modelica

Topics:

- Modelica modeling language
  - Declarative, equation-oriented language constructs
  - Acausal modeling
  - Object-oriented modeling
- Model development and simulation using Dymola
  - Building and editing a model
  - Compiling and simulating a model
  - Model analysis
- Modelica language syntax and semantics
  - First order, linear ODE example
  - Variables and variable types
  - Modelica language keywords and operators
  - A non-linear ODE example
  - Linear analysis
  - Automatic/symbolic differentiation

Lecture 4: Acausal Modeling and Differential Algebraic Equations

Topics:

- Differential Algebraic Equations (DAE)
    - Algebraic constraints arising in models of physical systems
    - Challenges of directly solving DAE systems
    - Classification of DAE systems by analysis of Jacobian
    - Symbolic reduction of DAE systems to ODE systems
    - Symbolic inversion of nonlinear plant models
    - Examples

Lecture 5: Physical Systems Modeling Principles

Topics:

- General modeling principles
    - Potential, Flow, Accumulation
    - Analogies between mechanical, electrical, hydraulic systems
    - Acausal model ports/connectors
    - Modelica modeling keywords and semantics
- Modeling Fundamentals for Thermo-fluid Systems
    - Robust reversing flow
    - Robust convective flow transport
    - Syntax, semantics and use of Modelica **stream** variables
    - Symbolic DAE reduction to reduce order of thermo-fluid models
    - Thermodynamic state selection implications
    - Examples

-----------*Module 3: Building Reusable Models At Different Levels of Abstraction*-----------------------

Lecture 6: Structuring Component Models for Standardization and Reuse

Topics:

- Object oriented principles
    - Partial models
    - Inheritance
    - Separation of device physics and fluid properties
        - Inner/outer declarations
        - Fluid property functions and function derivatives
    - Examples
- Model structure and architecture
    - Acausal connectors
    - Component models
    - System model architecture and replaceable components

  - Parameter propagation

Lecture 7: Model Libraries

Topics:

- Open Source Model Libraries
- Modelica Standard Library (MSL)
- Building and visualizing models using MSL
  - Building 2D and 3D mechanical system models
  - Simulating and animating multi-physics mechanical-electrical models
  - Creating multi-fidelity models using replaceable components
- Model library structure
  - Packages
  - Examples
- Library repositories and version control

Lecture 8: Cross-Platform Model Integration

Topics:

- Export of Modelica models
  - FMI standard and FMU export/import
  - FMU export to Simulink
  - Dymola model export to MATLAB
- Import of FMUs into Dymola
- Examples

--------------- *Module 4: Model Predictive Control and State Estimation* ---------------------

Lecture 9: Model Predictive Control Fundamentals

Topics:

- Motivation for MPC
- Derivation of linear MPC equations
- MPC design parameters impacting
  - Robust control stability
  - Reference tracking
  - Disturbance rejection
- Tuning guidelines
- Using nonlinear models for MPC prediction
- Examples

Lecture 10: Solving Model Predictive Control Equations

Topics:

- Constrained optimization fundamentals
- Quadratic programming
  - Active set method
  - Interior point method
- Problem scaling/normalization
- Approximations for fast, realtime applications
  - Suboptimal solutions
    - Fixed iterations
    - Constraint relaxation
    - Small control horizon
  - Model reduction
- Examples

Lecture 11: Controllability, Observability, and Stability

Topics:

- Controllability/observability gramian concepts and energy analogy
- Empirical/simulation-based gramian for nonlinear systems
- Detectability and Stabilizability
- MPC stability

Lecture 12: State Estimation Fundamentals

Topics:

- Role of estimation in feedback control, FDA, and redundancy management • Steady state uncertainty propagation
- The discrete-time Kalman Filter (KF) derivation
- Implications of model reduction to KF structure
- Simultaneous State/Parameter Estimation
- Nonlinear state estimation
  - Extended Kalman Filter
  - Moving horizon estimation (MHE) introduction
    - Relationship to KF under certain assumptions
    - MHE problem formulation including constraints
    - Approximations for fast, real-time applications
  - UKF

---------------*Module 5:* Analysis of Control System Robustness to Uncertainty------------------

Lecture 13: Model Validation

Topics:

- Uncertainty quantification
- Model refinement using data
  - Parameter bias estimation/smoothing
  - Model structure updates
- Model validation methods
  - Frequency domain methods
  - Time domain error analysis
- Model robustness validation
  - Quantifying domains of attraction
  - Invertibility
- Examples

Lecture 14: Control System Qualification

Topics:

- Establishing system test and qualification requirements
  - Quantitative requirements
  - Test case definition
  - Uncertainty quantification
- Analysis methods
  - Frequency domain dynamic analysis
  - Frequency domain control stability margin analysis
  - Time-domain analysis of objective functions
  - Sensitivity analysis and critical parameter identification
  - Assessment of robustness to variation of critical parameters using monte carlo methods
  - Simulation-based FMEA validation
    - failure injection simulations
  - Optimization-based worst-case/reachability analysis

--- *Module 6:* Real-time, Discrete-time Modeling of Nonlinear Dynamical Systems ------

Lecture 5: Modeling Discrete-Time, Multi-rate Control Systems

Topics:

- Realtime numerical methods for nonlinear ordinary differential equations (ODE) o Explicit methods
  - Implicit methods
  - Numerical stability analysis
  - Numerical integration error analysis
- Modeling precise timing and discretization in multi-rate systems
  - Sampling rates

  - Sampling time shifts
  - Computational delays
  - A/D quantization
  - Signal noise and aliasing
- Examples

USEFUL READING (TBD)

Course Delivery & Grading

The course will be offered online in small, recorded modules according to the course syllabus. Direct communication with the instructor will be available twice a week for discussion, questions and quizzes. Grading of the course will be based on the Course Project described below and homework. Optional homework and quizzes will be assigned to students during the semester for bonus grade.

Project, Presentations and Project Report

A project is to be developed by each student, which is expected to evolve during the entirety of the semester. The project refers mainly to design project identification, challenge quantification, significance and relevance to the MBD philosophy, plan of attack, system modeling, system optimization using cross-platform programming. The final deliverable (presentation) should identify all the aforementioned elements in a quantifiable manner and suggest a strategy for solution. A separate rubric with the details of the project will be provided to the students on HuskyCT. A mid-term and final report are the main deliverables of this project, on the basis of which student will be graded.

Other Policies

Student Conduct: http://www.dosa.uconn.edu/student_code.html. Students are responsible for adherence to the University of Connecticut student code of conduct. Perhaps the most important policy to pay attention to is the section on Student Academic Misconduct. "Academic misconduct is dishonest or unethical academic behavior that includes, but is not limited, to misrepresenting mastery in an academic area (e.g., cheating), intentionally or knowingly failing to properly credit information, research or ideas to their rightful originators or representing such information, research or ideas as your own (e.g., plagiarism)." Examples of academic misconduct in this class include, but are not limited to: copying solutions from the solutions manual, using solutions from students who have taken this course in previous years, copying your friends homework, looking at another student's paper during an exam, lying to the professor or TA and incorrectly filling out the student workbook.

Attendance: Attendance will not be taken; however, it is practically impossible to follow the class if classes are missed.

Absences: Make-up of missed exams requires permission from the Dean of Students, see "Academic Regulations." Midterm-exams are treated the same as Final Examinations. Students involved in official University activities that conflict with class time must inform the instructor in writing prior to the anticipated absence and take the initiative to make up missed work in a timely fashion. In addition, students who will miss class for a religious observance must "inform their instructor in writing within the first three weeks of the semester, and prior to the anticipated absence, and should take the initiative to work out with the instructor a schedule for making up missed work."

*Course Schedule**

Date[1]	Topic	Module No	Project Dates
Jan 18	*Lecture 1: Model-Based Design Fundamentals*	1	
Jan 25	*Lecture 2: Analysis in Model Based Design*	1	
Feb 1	*Lecture 3: Modeling Dynamical Systems Using Modelica*	2	
Feb 8	*Lecture 4: Acausal Modeling and Differential Algebraic Equations*	2	Project Proposal Report
Feb 15	*Lecture 5: Physical Systems Modeling Principles*	2	
Feb 22	*Lecture 6: Structuring Models for Standardization and Reuse*	3	
Mar 1	*Lecture 7: Model Libraries*	3	
Mar 8	*Lecture 8: Cross-platform Model Integration*	3	
Mar 15	*No Class – Spring Break*		
Mar 22	*Lecture 9: Model Predictive Control Fundamentals*	4	Project Mid-Term Report
Mar 29	*Lecture 10: Solving Model Predictive Control Equations*	4	
Apr 5	*Lecture 11: Controllability, Observability, and Stability*	4	
Apr 12	*Lecture 12: State Estimation Fundamentals*	4	
Apr 19	*Lecture 13: Model Validation*	5	
Apr 26	*Lecture 14: Control System Qualification*	5	
May 3	*Lecture 15: Modeling Discrete-Time, Multi-Rate Control Systems*	6	
May 7	*End of class, final project due*		Project Final Report

* Schedule is tentative and may change

[1] Date indicates release of lecture modules

Instructors' Contact Information:

• Tim Crowley: timothy.crowley@uconn.edu

Helpful links:

• UConn AnyWare is a virtual desktop service that you can use from any device with a network connection: https://software.uconn.edu/uconn-software-online/ • Course Material: https://huskyct.uconn.edu/

• Institute for Advanced Systems Engineering: http://www.utc-iase.uconn.edu/