The first goal is to learn how to formulate models for the purposes of control, in applications ranging from finance to power systems to medicine. Linear and Markov models are chosen to capture essential dynamics and uncertainty. The course will provide several approaches to design control laws based on these models, and methods to approximate the performance of the controlled system. In parallel with these algorithmic objectives, students will be provided with an introduction to basic dynamic programming theory, closely related stability theory for Markovian and linear systems, and simulation and stochastic approximation concepts underlying reinforcement learning.

It is intended for graduate students who have some background in control and stochastic processes. Experience with *Matlab* or *Python* is essential.

**Why do we need noisy models?** When you introduce the word “stochastic” to control, this just means that you are bringing in a larger range of tools for understanding how to control systems, and evaluate their performance. Name a tool from probability, and you have something useful for control synthesis. In particular, there is the question of *information*. This may mean the data available for control, or information about the system to be controlled. There may be variables of interest that are not directly observed, so we will want to estimate. Tools to be applied include nonlinear filtering and stochastic approximation (a foundation of reinforcement learning).

**Course Outline:** The typewriter font refers to handouts that I have been refining for the past 20 years.

### I. Control and Stability Theory

1. **Overview & examples.** Review of concepts from optimal control
   
   Section 3.8 and 5.6, and examples from Chapter 7 of [12] (see also handouts: 3bHJB.tex and 3ACOE+SpeedScaling.tex)
   
   Introduction of [12], including warnings concerning adaptive control disasters from the 1980s.

2. **Controlled Markov models and MDPs.** Chapter 7 of [12].

3. **Markov models and more examples.** First half of Chapter 6 of [12].

4. **Lyapunov theory for stability and performance.**
   
   Second half of Chapter 6 of [12].
   
   See also 2Representations.tex

5. **Numerical techniques:** Value iteration (without control), and Perron-Frobenius techniques for steady-state and value functions
   
   Appendix B of [12] and 2Representations.tex

6. **Monte-Carlo for performance estimation:** Emphasis here on how to estimate confidence bounds – a tutorial on “how to do simulation right”.
   
   Section 6.7 of [12] and 2Representations.tex.

---

1. A *Markov process* is nothing more than a nonlinear state space model subject to noise.
II. Optimal Control

1) Everything boils down to total cost (mainly essays and examples to unify what’s to come – theory applies to many different performance metrics).
2) Approximate dynamic programming.
3) Numerical techniques: Policy and value iteration; LP formulation.
   VAPIAandLP.tex (8 dense pages)
4) Partial information (belief state). Multi-armed bandits (UCB heuristic).
   NonlinearFilter_BeliefState.tex on the creation of the belief state.

III. Adaptation and Learning

1) Simulation and stochastic approximation: theory & applications
   The ODE Method: Chapter 8 of [12] (see also the big book chapter [6]).
2) TD- and Q-learning from Chapters 9 and 10 of [12].

References: The following are available free on-line (send your thanks to CUP):

⊙ S. P. Meyn, Control Systems and Reinforcement Learning.
   https://meyn.ece.ufl.edu/2021/08/01/control-systems-and-reinforcement-learning/
   www.meyn.ece.ufl.edu/archive/spm_pubs.html
⊙ S. P. Meyn, Control Techniques for Complex networks.
   www.meyn.ece.ufl.edu/archive/spm_pubs.html

The following are valuable background (send your thanks to Profs. Hajek and van Handel):

⊙ B. Hajek, Exploration of Random Processes for Engineers.
   www.ifp.illinois.edu/~hajek/Papers/randomprocesses.html Review: \((\Omega,\mathcal{F},P) \star \mathbb{P}(A) \star \mathbb{E}[X | Y]\)

The following textbooks are of value, but not needed to follow the course.

⊙ D. Bertsekas and J. Tsitsiklis, Neuro-Dynamic Programming (see also Sutton’s new book on reinforcement learning).
   web.mit.edu/dimitrib/www/soc.html
⊙ P. R Kumar and P. Varaiya, Stochastic systems: Estimation, identification, & adaptive control.

MoreReferences: The monograph [12] was based on the following papers, along with more ancient material from other masters of RL and control:

Feature Selection for Neuro-Dynamic Programming (book chapter used in the course for many years)

See also lots of great material from Adithya Devraj’s thesis: [2, 1, 3, 5, 4, 7, 8, 9].

References


