

ECE 302 Probability Models and Stochastic Processes

Syllabus Spring 2021

Course Description:

Topics in probability, random variables and stochastic processes applied to the fields of electrical and computer engineering. Probability, events, random variables, expectation, moments, characteristic functions, conditional probability and expectation. Functions of random variables, random vectors, Gaussian random vectors, Poisson points. Bounding and limit theorems. Relations among important distributions and probability models. Stochastic processes: stationarity, ergodicity, Brownian motion, Markov processes. Deterministic systems with stochastic inputs, correlation and power spectral density, ARMA models. Hilbert space and applications: orthogonality principle, discrete Wiener and Kalman filters, linear prediction, lattice filters.

Textbook:

Alberto Leon-Garcia., Probability, Statistics, and Random Processes For Electrical Engineering (3rd Edition) Prentice Hall, 2008.

Reserve texts:

Stark and Woods, Probability, Statistics, and Random Processes for Engineers (4th Edition), Prentice Hall; 4 edition (August 20, 2011).

Papoulis and Pillai, Probability, Random Variables and Stochastic Processes 4th edition. McGraw-Hill 2002

Other resources:

These course notes are very good, and some of them will be used as supplementary material:

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-011-introduction-to-communication-control-and-signal-processing-spring-2010/readings/>

Homework:

Homework will be assigned, we will go over it in class.

MATLAB Projects:

There will be 5 MATLAB programming exercises to complement the material. Small groups (2-3) are acceptable, but not required, in completing these exercises. There will be a code-review after both projects, which will count for 20% of the project grade. 1 person will be selected at random from your group, who will be orally quizzed on your project by Prof. Keene. The resulting grade will count for all team members.

Project 1 - Dungeons and Dragons

Project 2 - MMSE Estimation

Project 3 - ML Estimation

Project 3 - Detection Theory

Project 5 - MMSE FIR filtering

Grading:

6 quizzes, 10 points each

5 projects, 10 points each

90 - 100 points = A

80-89 points = B

70-79 points = C

Project grading scheme

8 points: Technical Correctness of the code and any accompanying derivations

1 point: Quality of figures, meaning they should be annotated fully (legends, axes labels). You should also make judicious use of overlaid plots, or subplots, as appropriate. In general you should be striving to use only a few, well-annotated plots.

1 point: Quality of comment in code. Your code should be well commented, with a few paragraphs describing your solution, referring to figures, as appropriate. The overall goal here is a single document that contains both your code and all necessary information so that someone with a general EE background could understand the document.

Policies:

Project deadlines will be 'soft deadlines' but please contact me if you need to turn a project in late. Don't just assume it's ok.

Collaboration between teams is allowed and encouraged; explicit copying of code is not. Any code that you did not write yourself that is not part of the basic MATLAB installation must be attributed. Failure to do so results in an F on the project.

Quiz Reviews: If you get less than 8 out of 10 on a quiz, you can contact me to do a review. There's 6 quizzes so there's ample opportunity to make up quiz points if you get a 8 or 9. Subtle change from last semester but I did wind up spending a lot of time on quiz reviews for just a single point and the time adds up. Also, please contact me and schedule the review within a week or so, there were too many end of semester quiz reviews that caused a big crunch.

Anti-Plagiarism method

Unfortunately, coding exercises are easily plagiarised. It is normal and productive to share snippets of code with colleagues, or to find code on the internet to use. It is not acceptable to turn in code you do not understand. In industry, a common practice is code review, in which before your code gets accepted into a larger code base, you sit with another coder and explain what your code does and why. We will employ this method to reduce plagiarism in this course. I will randomly select a few people to explain their code to me, in person, in my office. Failure to explain your code to my satisfaction will result in an F on the project, for yourself and your teammates. If during the course of grading the assignments, plagiarism is suspected, additional action will be taken.

The above anti-plagiarism method is in addition to the policies found here:

<https://cooper.edu/engineering/curriculum/academic-standards-regulations>