Outline

1. Transfer Function Analysis and Filter Design
   (a) Analytic functions. Regions of convergence and inverse Laplace and z-transforms.
   (b) Minimum-phase, zero-phase and all-pass analog and digital systems. Spectral factorization.
   (c) Amplitude and phase distortion.
   (d) Classical analog filters: Butterworth, Chebyshev, elliptic. Frequency transformation.
   (e) IIR filter design: impulse invariance and bilinear transform methods.
   (f) FIR filter design: linear-phase filters; window design; equiripple filters.

2. Sampling and DFT
   (a) DFT and properties. FFT.
   (b) Infinite impulse trains, derivation of sampling theorems.
   (c) D/A models.
   (d) Spectral resolution and leakage. Periodograms.

3. Quantization and Filter Structures
   (a) Number representations. Uniform and logarithmic quantizers. SNR and dynamic range.
   (b) Limit cycles. Roundoff noise gain. Sensitivity.
   (c) Cascade of second-order sections. Lattice filters.

4. Multidimensional Signals and Systems
   (a) M-D Fourier transform and properties. M-D FIR filters.
   (b) 2-D FT in polar form. Tomography.
   (c) FT approach to Maxwell’s equations, wave equation and plane waves.
5. Multirate Systems and Filter Banks


(b) Polyphase structures. Maximally decimated filter banks. QMF and PR filter banks. Lossless systems.

6. Statistical Signal Processing

(a) Review of stochastic processes.

(b) Discrete-time Wiener filter, linear prediction, lattice filters.

(c) Differential coding.

Texts and References

Main textbook:

Additional references:
I will be giving out extensive notes on multidimensional Fourier transforms. Standard alternate references on DSP are [4] and [6]. A good reference on MATLAB is [1]; a new edition, based on MATLAB 8, is coming out later this Fall, but Mastering MATLAB 7 should still be fine. Haykin [2] is a useful reference for stationary random signals, linear algebra and linear prediction. Papoulis [5] is a good references for probability and stochastic processes. I also recommend Hogg [3] for probability and random variables. Vaidyanathan [7] is the reference for multirate DSP.

References


Course Work and Schedule

There will be several homework assignments, including theoretical problems and MATLAB problems.
There will also be two exams, tentatively scheduled for Tuesday, October 18 and Friday, December 9. The exams will count as 70% of your grade, and homeworks 30%.
There will also be a core knowledge quiz on Friday, September 16, from 8:00-8:30AM. (Class lecture will follow, 8:45-11:00AM). The grade is Pass/Fail. (There will be make-up opportunities)

A passing grade requires: a satisfactory attendance record; passing the quiz; passing at least one out of two exams; and a passing homework grade.
We will be meeting Tues 10–12 and Fri 8-10 except for no class on the following dates: 9/30, 10/14, 10/21, 12/13, 12/16.
Note the schedule is subject to change.
Regular attendance and participation is expected. You MUST join the course googlegroup, cuece11411. The instructions for joining are as follows:

- if you are going to use a gmail address, go to http://www.googlegroups.com/cuece11411 and ask to join.

- if you are going to use another email address, send a message to fredf88cooper@gmail.com, and specify the email address you want to use, and I will send you an invite to the group.

I prefer if you use fredf88cooper@gmail.com to contact me, or to submit homework electronically. If you want to meet me outside of class time, the best procedure is to send me an email requesting a meeting. I can also do skype, gmail chat, etc.

Homework Guidelines

MATLAB code can be written in pairs or individually (not groups larger than two); all authors of the code must be clearly labeled. You MUST submit code, not just results.
Otherwise, although you can “study” together, including discussing homework problems with each other, in the end you are responsible for your own work and you must not blindly copy from someone else.
Please make electronic submissions to fredf88cooper@gmail.com
ALL MATLAB problems must be submitted electronically. Make sure the code is executable; I will not go through the effort of trying to debug your code. Make sure your code works—sometimes there may be variables in your workspace, and so forth, that makes it appear that your code works when it doesn’t. To be sure, clear your workspace and figures and test your code before you submit it. Be sure to include “wrapper code” for a sample run, generating graphs etc., or else include figures and so-forth. Provide instructions when needed. In other words, make life easier for me and I will be more merciful to you.
Other homework problems can be scanned in and emailed to me, or given to me in hardcopy form.