# DEPARTMENT OF ELECTRICAL ENGINEERING
## University of Washington
Winter Quarter 2017

**Course:** EE 518
**Title:** Principles of Discrete-Time Signal Processing **Credits:** 4 Units
**Prerequisite:** EE PMP Introductory class 590, offered every Fall quarter, or similar.
**Class website:** http://canvas.uw.edu/courses/1098557

**Why take this class?** This class is intended for engineers and scientists within and outside Electrical Engineering. What is it useful for? Some examples, from basic to advanced: How many engineers and scientists know that linear interpolation is a poor way to increase sample rate? And why are cubic splines also, under most conditions, a bad choice? How can you do a better job of interpolation? What is an optimal interpolator and what metric space is it optimal for? Why is that optimal interpolator impractical, yet how can you, in practice, get as close as possible to this optimal interpolator?

Why is a Fourier transform and frequency important? How does the concept of frequency have principled depth which goes way beyond simply decomposing arbitrary signals into oscillating components? Yet what are the limitations of Fourier transforms, and how do more general $z$-transforms and perhaps wavelets get around these limitations?

Knowing the theory of $z$-transforms can, among many other applications, allow you to potentially massively decrease the number of operations needed for changing the sample rate of signals, images, and video. This same transform can also be applied to equalizing communication channels with magnitude and phase distortion. Why is phase distortion so important? What is the difference between phase delay and group delay? Under what conditions are phase delay and group delay identical and why is that equality such an important thing to achieve, unusually by well-known computer algorithms which approximately equalize channels for high bit rate communication?

A fast Fourier transform is a well-known fast implementation of a discrete Fourier transform. How does the fast Fourier transform potentially massively speed up the discrete Fourier transform? But how is a discrete Fourier transform (and hence a fast Fourier transform) often a really poor approximation of a true discrete-time Fourier transform? Why is a fast Fourier transform usually unsuitable for signal filtering? What are some better and often much more efficient filters, which also won't ring close to their transition frequencies? What is a design criteria and metric space (hint: it's not Euclidean or $l_2$) which allows for, in practice, the more generally useful frequency filter designs?

Optionally, where the last week's topic is chosen by class vote: How can you best characterize signals from time-varying systems, like real-world examples of speech, patches of images, and video? Why do Fourier transforms not fit this problem and how can they be modified? How do adaptive, Wiener, and Kalman filters fit this problem? Why are concepts like wavelets suitable, and at what cost? How can homomorphic deconvolution, with a simplified version called cepstral processing, apply? And, if chosen by class vote, why is the more recent literature like sparseness, compressive sensing, Markov chain Monte Carlo sampling, and/or modulation spectra (e.g. Clark and Atlas) potentially useful and worth your future investigation? Why are more recent and popular concepts like convolutional[1] and recurrent[2] neural and deep networks potentially more than simply hype and perhaps useful for your future engineering work?

**EE 518 Prerequisite:** PMP 590 or similar such as *recent undergraduate signal processing or other quantitative, particularly Fourier transform, theory background is required. You need to be comfortable with complex numbers, know at least elementary matrix theory, and know what a Fourier transform is.* If you have an undergraduate background in DSP, this class will still be challenging. For example, it will cover graduate level DSP concepts such as signal processing for signals from time-varying systems, multirate signal processing, and non-Euclidean decomposition spaces, which are needed to understand much of the more advanced signal processing, control, and related literature.

This EE 518 course quickly reviews linear time-invariant systems, discrete-time signals, sampling, Fourier transforms and bilateral $z$-transforms. If you don't have this background, these two inexpensive books can be suitable for quick and intense self-study: Hayes, **Schaum's Outline of Theory and Problems of Digital Signal Processing**, and, more basic, Hsu, **Schaum's Outline of Signals and Systems**, 2$^{nd}$ Edition. Note that some on-line courses also provide fine background, but most don't provide the kind of in-depth problem solving that will be central to EE 518.

---

[1] For first use of convolution within artificial neural networks, see right side of figure 1 of Homma, Atlas, and Marks, "An Artificial Neural Network for Spatio-Temporal Bipolar Patterns," *Proc. Neural Information Processing Systems (NIPS)*, 1987.

[2] For an early paper on recurrent networks, which also overlaps statistical signal processing, atlas nips see: Connor, Martin, and Atlas. For a recent recurrent net paper, see this example just published in *Proc. Neural Information Processing Systems*, Nov. 2016. For a more general modern background see, for example, http://www.deeplearningbook.org/ .

**Course:** EE PMP 518
**Title:** Principles of Discrete-Time Signal Processing
**Credits:** 4

**Course Web Site:** https://canvas.uw.edu/courses/1021826

**Course Description:** This class addresses the representation, analysis, and design of discrete time signals and systems. The major concepts covered include: Discrete-time processing and modeling of continuous-time signals and systems; decimation, interpolation, and sampling rate conversion; time-and frequency-domain design techniques for non-recursive (FIR) filters; prediction; discrete Fourier transforms, fast Fourier transform (FFT) algorithms and turning block into stream processing; short-time Fourier analysis and filter banks; multirate techniques; and various applications of these techniques. Some of the class homework will make use of MATLAB™ programs on computers within the UW or on your work or home computer. The course grade will be based upon weekly homework, a midterm exam, and the final exam. Prerequisites: The Fall EE PMP class. A mathematical/quantitative undergraduate degree, with knowledge of Fourier transforms, and some discrete math and linear algebra.

**Lecture Time:** T 6:00–8:50 pm in EE 003, with a break at 7:20–7:35 pm.
**Instructor:** Prof. Les Atlas, EE 410, atlas@uw.edu
**Atlas Office Hours:** T 5:00–5:50 pm in EE 410, Sunday 12:00-1:00 PM, Sieg Hall 1st floor 128, or feel free to request other times by email.

**Optional Discussion/Problem Session:** Sunday 12:00-1:00 PM, Sieg Hall 1st floor 128
**Teaching Assistant:** Elyas Bayati, elyasb@uw.edu
**TA Office Hours:** Sieg Hall 1st floor 128, Sunday 1:00–3:50 PM, Sieg Hall 1st floor 128, or feel free to request other times by email.

**Required Textbooks:**
1. Oppenheim and Schafer, **Discrete-Time Signal Processing**, 3rd Edition, Pearson Prentice Hall, 2010.
2. **MATLAB Student Version (earlier)**, or **MATLAB & Simulink Student Version (current)**:
   http://www.mathworks.com/academia/student_version/ (Any version with Signal Processing Toolbox)

Homework: Weekly homework is due in class on Tuesday (no later than 6:00 pm on Tuesday at the start of discussion), starting with Homework #1, due Tuesday 1/10, 6:00 pm. Solutions will be posted on the class website. Solutions will be covered in the same Tuesday discussion section. Late Homework will not be accepted.

**Midterm Exam:** In classroom EE 003, 6:00-7:50 on Tuesday, February 14. Open books and notes. No turned-on electronic devices (calculators, phones, etc.) allowed.

**Final Exam:** In classroom EE 003, 6:00-8:50 on Tuesday, March 14. Open books and notes. No turned-on electronic devices (calculators, phones, etc.) allowed.

**Course Grading:**
- Attendance: nominally required, since some material is discussed in class which is not in the text or notes.
- Weekly Homework: 15%
- Midterm Exam (2/14 in EE 003): 35% (Open book and notes.)
- Final Exam (3/14 in EE 003): 50% (Open book and notes.)

| Date | Week | Topic (subject to change) | Oppenheim *et al* chapter sections |
|---|---|---|---|
| 01/03/16 | 1 | Introduction, discrete-time (DT) sequences, DT systems, properties, LTI systems, convolution sum, difference equations, eigenfunctions, frequency domain, frequency response, Fourier operator, Fourier transform symmetries, and Fourier transform theorems | 1, 2.0-2.9 |
| 01/10/16 | 2 | *z*-transforms, region of convergence, inverse *z*-transforms, properties and uses of the *z*-transform | 3 |
| 01/17/16 | 3 | Sampling, DT vs. CT processing, downsampling, upsampling, sample rate conversion | 4.0-4.6 |
| 01/24/16 | 4 | Multirate signal processing, A/D & D/A conversion, and polyphase structures | 4.7-4.8, 5.0-5.2 |
| 01/31/16 | 5 | Frequency response of LTI systems, phase and group delay, pole/zero diagrams, all pass and minimum phase systems | 5.3-5.6 |
| 02/07/16 | 6 | Generalized linear phase and FIR types, FIR filter design by windowing | 5.7, 7.1, 7.5-7.6 |
| 02/14/16 | 7.1 | **Midterm Exam: 6:00-7:50, EE EE 003** **Covers all lectures, homework, and discussion through Week 6** | |
| 02/14/16 | 7.2 | **Review Midterm Solutions 8:00-8:50, EE 003** | |
| 02/21/16 | 8 | Optimal (equiripple) approximations for FIR filters | 7.7-7.9, and Atlas' Notes |
| 02/28/16 | 9 | Discrete Fourier series, circularity, the discrete Fourier transform (DFT), spectral analysis with the DFT | 8.0-8.7, 10.1-10.2 |
| 03/07/16 | 10 | The fast Fourier transform (FFT) and fast convolution. How to get to the deeper literature: Reference for tutorials on linear prediction, Kalman filtering, and time-varying adaptive systems. | 9.3-9.4 Atlas' Notes |
| 03/14/16 | | **Final Exam: 6:00-8:50, EE 037** **Covers all material from Week 1 through Week 10, with less detail on Week 10** **Review Final Exam Solutions 9:00-9:50, EE 037** | |