ECE 5650/4650
Modern Digital Signal Processing

Fall Semester 2018: Osborne B211, 8:00–10:40 AM

Instructor: Dr. Mark Wickert **Office:** EB-292 **Phone:** 255-3500
mwickert@uccs.edu **Fax:** 255-3589
http://www.eas.uccs.edu/~mwickert/ece5650/

Office Hrs: Wed. 10:40–11:15 am & Mon/Wed 1:30–2:15 pm, others by appointment.

Required Texts: Alan V. Oppenhiem and Ronald V. Schafer, Discrete-Time Signal Processing, third edition, Prentice-Hall, Englewood Cliffs, New Jersey, 2009.

Notes: Course lecture notes will be posted on the course Web Site as password required PDF files. Students are encouraged to download and print them.

Open source Python 3.6 using the Jupyter Notebook & Lab. I suggest Anaconda then install the package sckikit-dsp-comm using pip or conda see SP-Comm-Tutorial-using-scikit-dsp-comm.

Grading:

1.) Graded homework assignments, including use of Python with the scipy-stack + Python project 1, assignments 25%
2.) Final Python computer project worth 20%/15%. Grade option with final.
3.) Two “Hour” exams at 15% each, 30% total.
4.) Final exam worth 25%/30%.

Topics	Text Sections
1. Introduction and course overview	1
2. Discrete-time signals and systems	2.0–2.9
3. The z-transform	3.0–3.4
4. Sampling of continuous-time signals and discrete-time random signals	4.0–4.6 2.10, App. A 4.7–4.9
5. Transform analysis of linear time-invariant systems	5.0–5.7
6. Structures for discrete-time systems and finite precision issues	6.0–6.9
7. The discrete Fourier transform	8.0–8.7, 8.9?
8. Computation of the discrete Fourier transform	9.0–9.6
9. Applications of the DFT	Portions of 10.0–10.6

Important Deadlines: Review the Fall 2018 deadlines: https://www.uccs.edu/registrar/course-deadlines/fall-2018. Performance histograms (HW, Quiz, & Exams) will be discussed in class prior to the last day to drop, Friday October 26. Use this to decide on continuing or dropping the course – the deadline for dropping without ECE Chair signature (**NOT the Dean as stated in the link above**) is October 26. Only under **extenuating circumstances** will a late drop be considered.

Learning Outcomes

The expected learning outcomes of this course are a more in-depth treatment of discrete-time signals and systems as first started in the UCCS course ECE2610 or similar from another university. As a discipline within electrical engineering this known as digital signal processing (DSP). Specifically the student will learn how to model discrete-time signals and systems in the time domain; extend the time domain modeling to the frequency domain using the discrete-time Fourier transform (DTFT); working signals and linear time invariant (LTI) systems using z-transform (ZT) techniques; sampling theory and multirate sampling theory as found in modern DSP; discrete-time random processes and modeling quantization/fixed-point arithmetic; DSP problem solving using time, frequency, and z-domains effectively; properties of LTI systems having minimum phase and linear phase; choice of various DSP implementation topologies; the value and power of the discrete Fourier transform (DFT) and its efficient implemetation via fast Fourier transform (FFT) algorithms; Simulation of DSP algorithms and subsystems using Python with the Scipy stack.

Installing Python & scikit-dsp-comm

Download and Install the Anaconda Python 3.6 Distribution: https://anaconda.org/anaconda/python. ***Do not choose the defaults!*** I recommend you have Anaconda configure paths to the tools. This is not the default. Jupyter Notebook, QT Console, and Jupyter Lab will be installed with the Anaconda distribution, but more formation can be found at: http://ipython.org/.

Optionally Download and Install a code development environment that also integrates git version control such as Visual Studio Code (VS Code): https://code.visualstudio.com/ or PyCharm Community Edition IDE for Python: https://www.jetbrains.com/pycharm/.

Download and Install the distributed version control application git: https://git-scm.com/ on Windows systems (included on macOS and Linux).

Clone and Install the package scikit-dsp-comm by following the GitHub README page at: https://github.com/mwickert/scikit-dsp-comm.

Maintain scikit-dsp-comm using git pull origin master (see the information in the GitHub link).

Optional Jupyter Notebook to PDF Conversion

To render a Jupyter notebook as a PDF document a few more open source software components are needed:

Install Pandoc for file conversion to LaTeX and other formats: https://pandoc.org/index.html. I think Anaconda now installs this by default.

Install MikTeX for converting LaTeX documents to PDF on Windows: https://miktex.org/. When installing MikTeX be sure to choose the option to automatically download needed LaTeX packages on-the-fly. Or **Install TeXLive** for converting LaTeX documents to PDF on macOS and Linux: https://www.tug.org/texlive/

Install Typora: As an alternative to installing LaTeX (MikTeX or TeXLive), install the markdown editor Typora: https://typora.io/. Now you can export as *.md and then open the file in Typora and save to PDF directly. You can also do some nice file editing, including LaTeX equations, if need be.

Install Inkscape for converting embedded SVG graphics in Jupyter notebooks via Pandoc to LaTeX and then PDF: https://inkscape.org/en/release/0.92.2/.