ECE 5650/4650
Modern Digital Signal Processing

Fall Semester 2022

Instructor: Dr. Mark Wickert **Office:** EB-292 **Phone:** 255-3500
mwickert@uccs.edu **Fax:** 262-3589
http://ece.uccs.edu/~mwickert/ece5650/

Office Hrs: Wed. after class including during/after the ECE 4680 DSP lab, others by appointment, including *Slack*, *Zoom*, or *Teams*.

Required Texts: Alan V. Oppenhiem and Ronald V. Schafer, *Discrete-Time Signal Processing*, third edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1999.

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

Modeling Software: Python and/or Julia Jupyter notebook. Consider Julia using Pluto *reactive* notebooks. A Web Site document will explain the install details.

Grading:

1.) Graded homework assignments, including Python/Julia assignments 25%
2.) Final computer project worth 20%/15%. Grade option with final.
3.) Two "Hour" exams at 15% each, 30% total.
4.) Final exam worth 25%/30%. Grade option with final computer project

reorder planned

Topics	Text Sections
1. Introduction/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 2022 deadlines: https://www.uccs.edu/registrar/course-deadlines/fall-2022. Performance histograms (HW, Quiz, & Exams) will be discussed in class prior to the last day to drop, Friday October 28. 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 28. 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 multi-rate 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 implementation via fast Fourier transform (FFT) algorithms; Simulation of DSP algorithms and subsystems using Python with the Scipy stack and Julia which can also leverage the Scipy stack.

Slack

We will be using Slack *Channels* for group discussions and Slack *Direct Messaging* for audio and screen sharing. You will receive an invite to join the *DSP Comm Courses* Slack workspace.

Installing Julia and Python, with Jupyter Lab together

Documents linked on the course Web Site provide details on how to install both Python and Julia along with various Julia and Python packages (Julia_with_Python_Setup.pdf). Under Jupyter Lab you create notebook documents (.ipynb) for either Python of Julia.

Note: Jupyter notebook is the perfect place to write code, document code, write text using markdown, import figures, and typeset math equations using LaTeX syntax. Julia makes a significant step in solving the *two language problem* (Python for rapid prototyping and C/C++ for execution speed). I want you to experience this. You will also see the *reactive* Julia Pluto notebook in action.

Optionally Install vs code

Microsoft Visual Studio Code (vs code): https://code.visualstudio.com/ is great for writing and debugging Python and Julia code source code files or Jupyter Lab notebooks. *Extensions* are available for Python, Julia, and Jupyter Lab.

Notebook to markdown, edit, PDF print

Install Typora: A ($14.99 & 3 machines) markdown editor is at: https://typora.io/. Now you can export Jupyter notebooks as *.md and then open the file in Typora and save to PDF directly. You can also do some nice file editing if need be. Custom Header/footer, page numbers, etc.