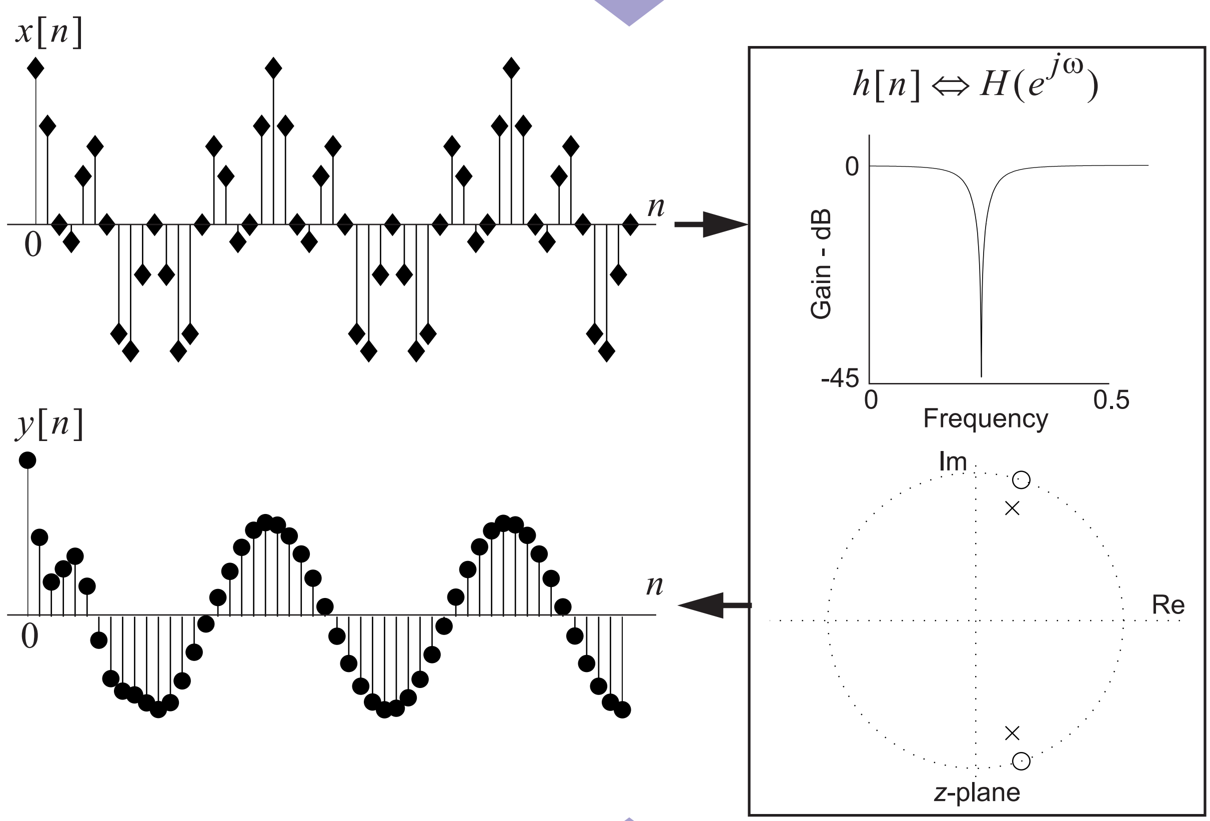
Modern Digital Signal Processing

ECE 5650/4650 Lecture Notes



Chapter 1

Course Introduction/Overview

Contents

.

1.1 Introduction to Modern Digital Signal Processing

- Contemporary DSP
  - Theory
  - Technology
  - Applications
- Course perspective
  - Expected background
  - Where to go from here
- What is this course about?
- The role of computer analysis/simulation tools in and outside this course
- The computer simulation project
- Instructor policies

1.2 Contemporary DSP

- The theoretical foundation for Modern signal processing got it's start back in the late 1960s and early 1970s
- Two key texts which started this era are *Digital Signal Processing* by Oppenheim and Schafer and *Theory and Application of Digital Signal Processing* by Rabiner and Gold, both published in 1975
- The Oppenheim and Schafer text *Discrete-Time Signal Processing* (editions 1989, 1999, 2010) is still one of my favorite
- A much newer text I also like is *Digital Signal Processing: Principles and Applications* by Thomas Holton [1]; Available on Amazon for $118.46

1.2.1 The Technology

- DSP chips/architectures
  - Integer/Fixed point implementations
  - Floating point implementations
  - FPGA based implementations
- DSP design and software and tools

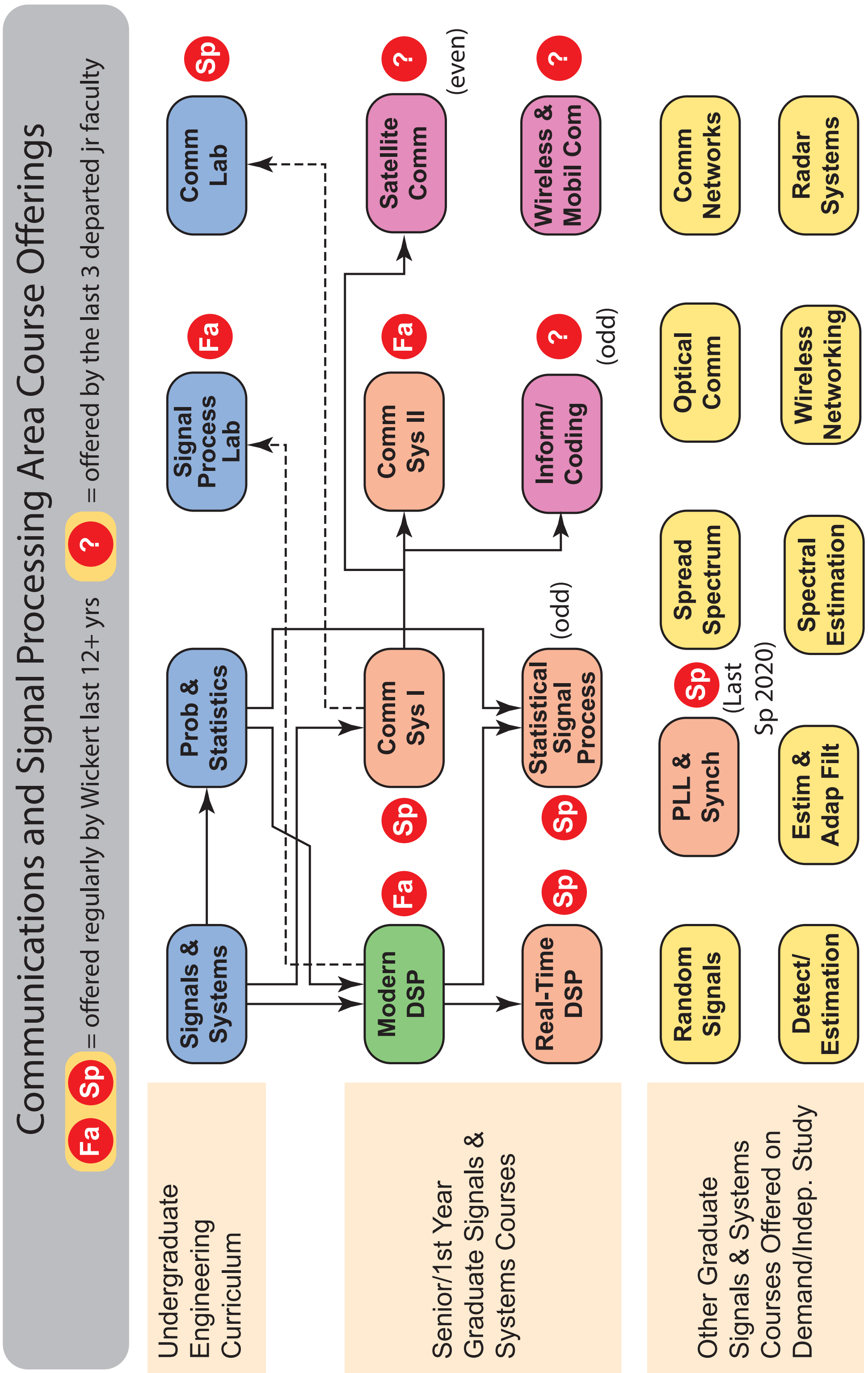
[1] https://www.cambridge.org/highereducation/books/digital-signal-processing/CFB036179B9CEDB3CB2E3B7D6E646D72

1.2.2 Key Applications Areas

- Computers/Internet
  - Broadband access, e.g., cable modem
  - Voice over IP
  - MP3 and related high quality audio formats
  - Storage
- Wireless Communications
  - Mobile communications (over 200 million phones/yr over the world)
  - xDSL
  - Real-time data compression for voice and video
  - Wireless and telecommunication infrastructure (5G, 6G and beyond)
  - Commercial aircraft WiFi
- Industrial
  - Real-time processing of industrial and medical signals
  - PC with DSP for virtual instruments for test and analysis
- Audio
  - Internet audio streaming
  - Home audio and theatre systems
  - Noise cancelling, e.g., quieting a car interior using adaptive noise cancellation or headphones
  - Sound stage and studio audio processing

- Video Imaging
  - HDTV in its various forms
  - Special purpose image processing in instrumentation and medical
- Biomedical
  - Many possibilities
  - Hearing aids and hearing assistive devices
  - Diagnostic imaging
- Automotive
  - Engine control
  - BMS (battery management systems)
  - Interior electronics audio/graphics
  - ADAS (Advanced driver-assistance systems, e.g. mm-wave radar, LIDAR, and cameras)
- Military/Aerospace
  - An active DSP area for over 60 years
  - Many consumer DSP applications had their start here
  - Sophisticated surveillance systems and smart weapons
  - Specialty technologies include:
    * Frequency domain processing
    * Parallel processing
    * Radar signal processing
    * Software defined radio (SDR)

1.3 Course Perspective in Comm/DSP Area ECE



1.4 What is this course about?

- This course has as its focus the *nuts and bolts* of one-dimensional discrete-time signals and systems analysis
  - Developing analytical skills is of primary importance
  - Introducing specialized applications is secondary
- Most all of the theoretical developments will parallel those of a traditional continuous-time signals and systems course
  - Here at UCCS, we now introduce discrete-time signal and systems in ECE 2610 (Intro to Sig & Sys), and more DSP topics are covered in ECE 3205 (Ckts & Sys II)
  - This course pushes the math well beyond the introduction in those courses
- Unique aspects of discrete-time signals and systems include:
  - Analog-to-digital and digital-to-analog interfaces
  - Multi-rate processing systems, i.e., more than one sampling rate in the signal input/output path
  - Software reconfigurability of systems
- Discrete random signals will be introduced
  - This will allow analysis of quantization affects
  - Key to follow-on courses involving statistical signal processing
  - Building simulation models that include *noise*

1.5 Software Analysis/Simulation Tools

- The experimental investigation of discrete-time signals and systems does not require a lab full of test equipment
- What is available from the academic perspective:
  - Basic mathematical analysis tools include
    - Python, Julia, MATLAB, Mathematica, wxMaxima (new DSP package), and others
  - DSP specific tools include
    - Python using `scikit-dsp-comm`[2] along with the SciPy module `scipy.signal`
    - Something simpler, e.g., wxMaxima with the new DSP package[3]
    - Julia which solves the two language problem (prototype and deploy fast models)
  - Starting from scratch
    - The C/C++ programming language
  - Real-time DSP Development
    - For ARM Cortex M – MDK *Keil*
    - Verilog/VHDL hardware description languages
- Inexpensive hardware evaluation is also a reality, e.g.,
  - ARM Cortex M4/M7 with CMSIS DSP library (NXP, ST-Micro, Infineon, TI, ...)

[2] http://scikit-dsp-comm.readthedocs.io/en/latest/?badge=latest

[3] https://github.com/mwickert/signal_processing_for_wxmaxima

- Currently exploring Bella Gem[4]
- ST X-CUBE-Audio-Kit 1.3.0

1.6 The Computer Projects

- Computer based exercises using Python will be assigned during the semester (Jupyter Notebook or Lab with Python 3.13)
  - These projects first familiarize you with Python, then work through the many details of DSP in the time and frequency domains
  - Python examples are also embedded in the notes and will be used for in-class examples and demos
  - The module `pyaudio_helper` of `scikit-dsp-comm` allows real-time DSP to be implemented inside the Jupyter notebook, including GUI widgets[5]
  - For Fall 2025 at least one project problem will introduce you to pyaudio_helper and your PCs audio subsystem
- A *capstone* project of sorts, will be assigned during the second-half of the semester
- In the Spring Semester ECE 5655, *Real-Time DSP*, has been offered as a complement to Modern DSP (in the future?)
  - In this course and ECE4680 ARM Cortex M processors and the Keil MDK IDE

[4] https://www.crowdsupply.com/bela/bela-gem-stereo-and-multi

[5] http://conference.scipy.org/proceedings/scipy2018/mark_wickert_250.html

- Another follow-on path is ECE 5615, *Statistical Signal Processing*, which builds upon the theory side of DSP and introduces random signals (offered next?)

1.7 Course Syllabus

ECE 5650/4650
Modern Digital Signal Processing

Fall Semester 2025

Instructor: Dr. Mark Wickert **Office:** EN-242 **Phone:** 719-332-6996
mwickert@uccs.edu **Fax:** 719-262-3589
https://faculty.uccs.edu/mwickert/

Office Hrs: Wed. after class including during/after the ECE 4680 DSP lab, others b appointment, including *Slack*, *Zoom*, or *Teams*. (my preference is Slack)

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. Studen are encouraged to download and print them.

Modeling Software: Python and Julia Jupyter notebook and wxMaxima. Consider Julia using Plut *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

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

reorder planned

Important: If you are a student with a disability and believe you will need accommodations for this class, it is your responsibility to register with Disability Services and provide them with documentation of your disability. They will work with you to determine what accommodations are appropriate for your situation. To avoid any delay, you should contact Disability Services as soon as possible. Please note that accommodations are not retroactive and disability accommodations cannot be provided until a Faculty Accommodation Letter has been given to me. Please contact Disability Services for more information at Main Hall, 3rd floor, Room 324, (719) 255-3354, or dservice@uccs.edu.

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. Download is under *Resources*.

Setting up Python via miniconda

To set up Python alone but with the advantage of virtual environments, miniconda works very well (on Windows install for Powershell). The helper document new_python_virtual_environment.pdf is provided. To run real-time DSP code in Jupyter Lab or vs code using pyaudio_helper. pyaudio_helper runs under Python 3.13.

vs code

Microsoft Visual Studio Code (vs code) 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, and many others. See the Julia_with_Python_Setup doc linked below for vs code setup info

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 Lab 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). You will also see the *reactive* Julia Pluto notebook in action in class.

wxMaxima

Similar to Mathematica, as it is a computer algebra system (CAS), but open-source. Simpler to use than Python, but not as fast at numerical DSP type operations. I will be showing off the new DSP package in class.

Typst

Typst is a new writing tool, similar to LaTeX, but it has a very nice cloud-based interface with real-time preview and one click PDF export. Why? To allow you to create documents that are a hybrid of scanned hand written solutions, typed out problems with text and math, and code with plots. I will supply a homework template and examples and you can decide if you want to try it.

1.8 Instructor Policies

- Homework papers are due via Slack by noon on due date
- If business travel or similar activities prevent you from attending class and turning in your homework, please inform me beforehand
- Grading is done on a straight 90, 80, 70, ... scale with curving below these thresholds if needed
- Screencasts of the lectures will be made available as soon as possible after each lecture; this may be of help to those of you that travel and to others for review purposes
- Homework solutions will be posted on the course Web site as PDF documents with password protection
- Old exams will be posted on the Web site prior to the hour exams