ECE 161A: Introduction to Digital Signal Processing

Florian Meyer
University of California, San Diego
Email: flmeyer@ucsd.edu

Course Adminstration

- Textbook: Discrete-Time Signal Processing, Oppenheim, Schafer and Buck, 3rd edition
- Additional References:
  - Digital Signal Processing, John Proakis and Dimitris Manolakis, 4th Edition
  - Digital Signal Processing using Matlab, Vinay Ingle and John Proakis (Good complement to our textbook)

Staff

- Instructor: Prof. Florian Meyer (flmeyer@ucsd.edu)
  - Office: EBU1 4402
  - Office Hours: Fri. 2:00 PM – 3:30 PM via Zoom (Class Material)
- TA: Luisa Watkins (l1watkins@ucsd.edu)
  - Office Hours: Fri. 3:30 PM – 4:30 PM in 5101E (Homework)
  - Discussion Hour: Wed. 2:00 PM – 2:50 PM in CENTR 113
- Website: Canvas
- Both instructor and TA will answer questions on the Canvas discussion board

Course Overview

- Prerequisite: ECE 101A. Should be familiar with the following topics and concepts in the following sections in the book.
  - Chapter 2: 2.1 — 2.9 (LTI systems, Fourier transform of discrete time signals)
  - Chapter 4: 4.1 — 4.5 (Sampling theorem)
  - Chapter 5: 5.2 — 5.3 (Difference equations, transfer functions, poles and zeros)
- Course Content: Filter Design and DFT/FFT
  - Chapter 3: 3.1 — 3.5 (Z-transforms)
  - Chapter 5: 5.1, 5.5, 5.7 (Group delay, All-Pass filters, linear phase filters)
  - Chapter 6: 6.1 - 6.5, 6.7, 6.8 (Filter implementation and finite precision effects)
  - Chapter 7: 7.1 - 7.7 (Filter design: IIR and FIR)
  - Chapter 8: 8.1 - 8.7 (DFS, DFT, linear convolutions using DFT)
  - Depending on time available: Chapter 9: 9.2, 9.3 (FFT)

Grading

- Midterm – 25%
- Final – 50%
- Homework – 25% (Weekly assignments (20%), and online Quizzes (5%))
  - Usually assigned on Thursday and due the following Friday at noon. No late submissions.
  - Must be original work
  - No studying of old solutions
  - Do not share notes and solutions
  - Discussion among classmates is ok, but no copying
  - Please review the academic integrity document on class website and also academicintegrity.ucsd.edu

Some Relevant Questions

- What are Discrete Time Signals?
- How do they arise?
- What do we want to do with them?
- What are the potential applications?
- Why discrete time signals and processing as opposed to continuous (analog) time signal processing?

Discrete Time Signals

Discrete Time signal: Sequence of numbers denoted by $\{x[n]\}$. If the numbers (values) are real, we have a real sequence. If complex, we have a complex sequence.

Examples:

1. $x[n] = 3n$. A real sequence. $x[0] = 0$, $x[1] = 3, ...$
2. $x[n] = e^{j\frac{\pi}{4}n}$. A complex sequence, $(j = \sqrt{-1}.)$

Origin of these Sequences

Usually obtained by sampling a continuous time sequence.
$x[n] = x_c(nT)$, where T is the sampling interval.

How should we choose T?

- T large preferred. Have to store smaller number of samples. For real-time applications, provides for more time to carry out the computations needed.
- Problem: could loose information. T small allows for a better approximation of the continuous time signals by its samples.

Goal: Choose T as large as possible or the sampling frequency $F_s = \frac{1}{T}$ as small as possible subject to fidelity constraints.

Sampling Theorem: If the highest frequency in the real continuous time signal $x_c(t)$ is $F_{max} = B \;\; Hz$, i.e. $|X_c(j\Omega)| = 0, \;\; |\Omega| > 2\pi B$ rads/sec (band limited),and the signal is sampled at a rate $F_s = \frac{1}{T} \geq 2F_{max} = 2B \;\; Hz$, then $x_c(t)$ can be exactly recovered from the sample values $x[n] = x_c(nT)$.

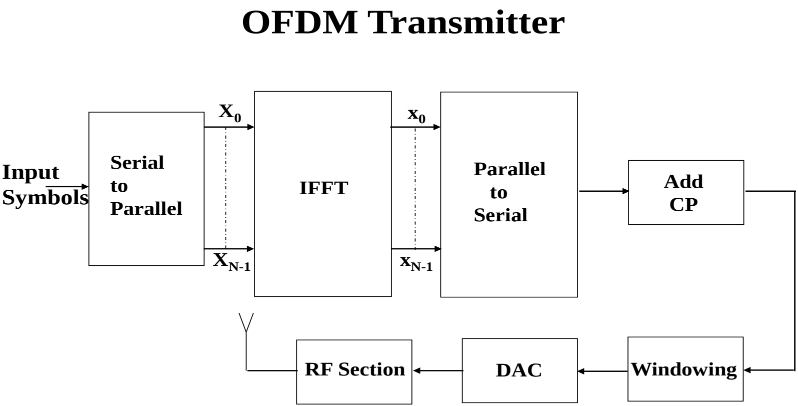
Goals of discrete time signal processing

- Analysis: Analyze signals for structure. Concepts of DFT, FFT, become handy
- Processing: Process signals to achieve an objective. Filtering. This leads to filter design (FIR, IIR).

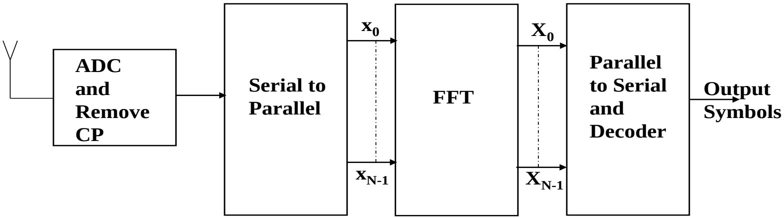
Applications

- Speech and Audio
- Images and Video
- Digital Communications
- Biomedical applications
- Radars and Sonars
- ...

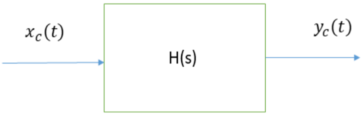
LTE Wireless Transmitter(Orthogonal Frequency Division Multiplex (OFDM))



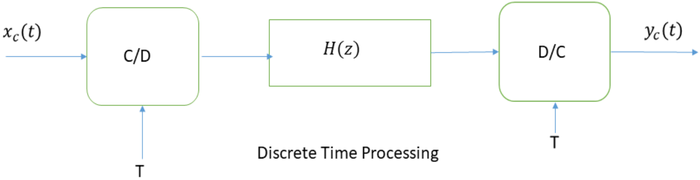
OFDM Receiver



Digital versus Analog



Continuous Time Processing



Discrete Time Processing

Benefits of Discrete Time Processing

- Analog components tend to drift and have to be more accurately chosen. Digital circuits are more robust.
- Programmable - Changing filters is easy
- Processor can be shared
- Cascade filters and there is no loading problem
- Low-frequency systems: large inductors and capacitors

Analog systems: Suitable for large bandwidth systems, low power systems

Discrete Time Processing versus Digital Signal Processing

- Discrete Time Processing: Signal sampled (Time discrete) and samples are represented with infinite precision
- Digital Signal Processing: Signal sampled (Time discrete) and samples are represented with finite precision, e.g. 8 or 12 bits.