ECE 420

Lecture 1

Jan 14 2019

Course Overview

Course Administrative Info

- Webpage: https://courses.engr.illinois.edu/ece420/sp2019/
- Lecture: Mondays 2:00-2:50PM, 2013 ECEB
  - Lecturer: Dr. Jeffrey Brokish
- Labs: 2:00-3:50PM, 5027 ECEB
  - ABA: Tue, ABC: Wed, ABD: Thu, ABE: Fri
  - TAs
    - Grant Greenberg
    - Yu-Jeh Liu

Course Structure

t = 0

Lecture	Labs
Background for Labs	Structured Labs
	Assigned Project
Special Topics/ Guest Lectures	Final Project
Final Quiz	Final Project Demo

t = End of Semester

ECE 420 Overview

- First half: 7 Structured Labs
  - Embedded DSP development framework
    - High-level (Python) → Embedded (Android with Java/C)
  - Different signal modalities and interfaces: IMU, audio, visual
  - Basic DSP algorithms
    - Digital filtering
    - Spectral analysis
    - Auto-correlation analysis: pitch detection/correction
    - Image and multidimensional signal processing

ECE 420 Overview

- First half: 7 Structured Labs - Format
  - Prelab [Individual]
    - Complete individually prior to lab, submit to TA
  - Quiz [Individual]
    - Overview of concepts from previous lab
  - Demo [Group]
    - Demonstrate work from previous week to TA, answer questions
  - Lab work [Group]
- No prelab, quiz or demo this week

ECE 420 Overview

- Second half: ‘Student Choice’ Group Projects (subj. to approval)
  - Start with an Assigned Project Lab
    - Explore implementation of a DSP algorithm from the literature
    - In Python, 2 week duration
    - Jumping off point for the Final Project
  - Final project
    - Proposal (the pitch) and Design Review
    - Deliverables and 3 weekly milestones
    - Final Project Demo and Presentation
    - Final Report (and optional Video)
  - Recommended not to wait until week 7 to start exploring options

ECE 420 Overview

- Grading
  - Structured Labs – 40%
  - Assigned Project Lab – 10%
  - Final Project – 45%
  - Final Quiz – 5%

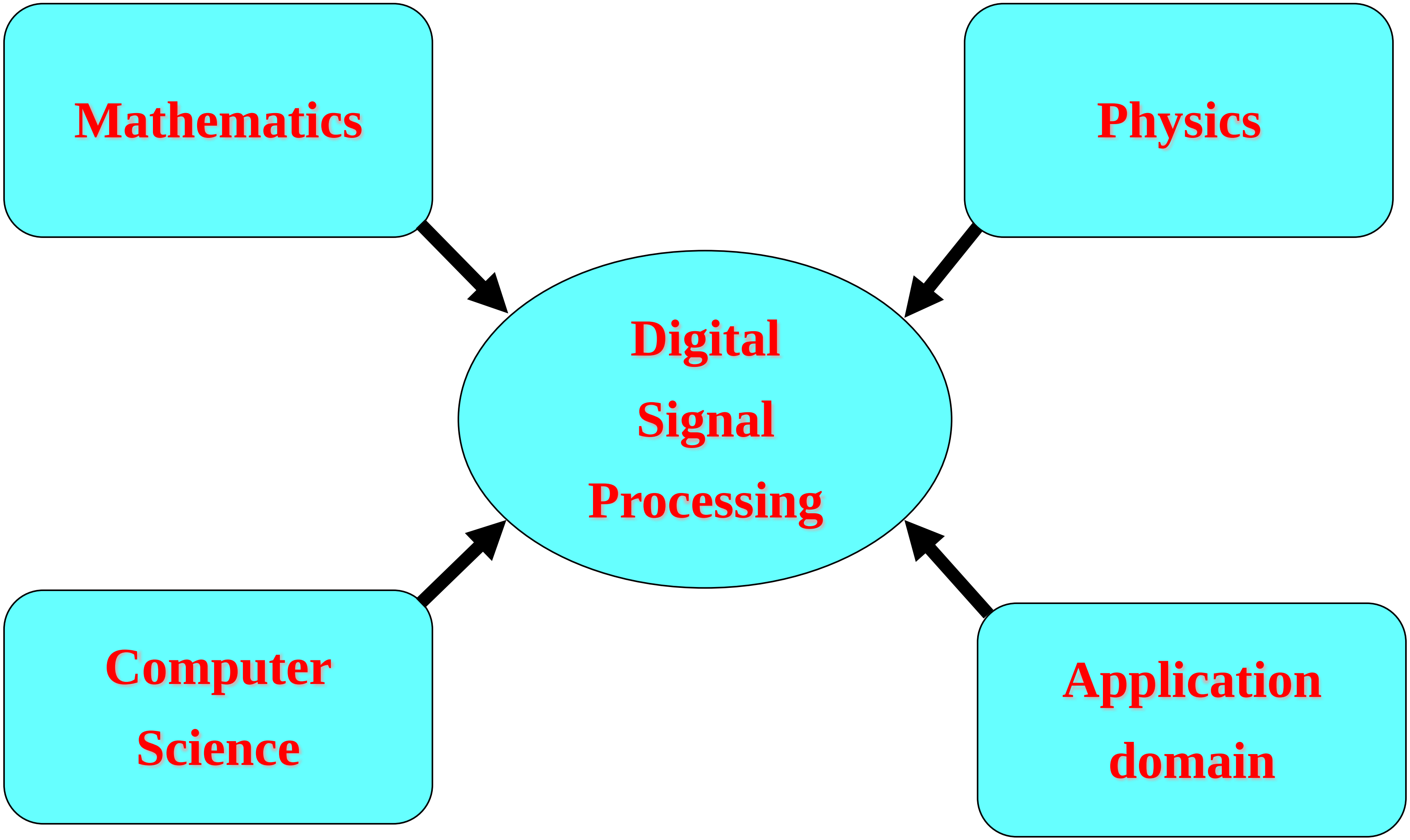
- For more detailed breakdown consult the course website

Embedded Digital Signal Processing

Embedded Digital Signal Processing (DSP)

- **"Signal"**: physical quantity that carries information
- **"Processing"**: series of steps to achieve a particular end
- **"Digital"**: done by computers, microprocessors, or logic circuits
- "**Embedded**": part of a complete device (hardware), often with real-time constraints

Background for DSP



Example: Speech Recognition using DSP

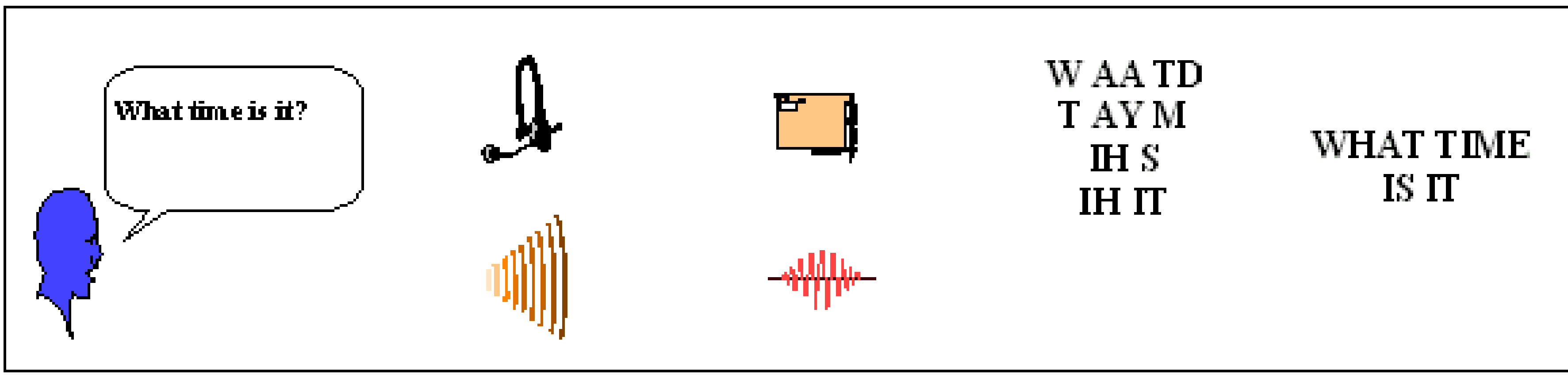
USER

MICROPHONE

SOUND CARD

SPEECH RECOGNITION ENGINE

SPEECH-AWARE APPLICATION



User speaks into the microphone.

Microphone captures sound waves and generates electrical impulses.

Sound card converts acoustical signal to digital signal.

Speech recognition engine coverts digital signal to phonemes, then words.

Application processes words as text input.

Digital Cameras

Original

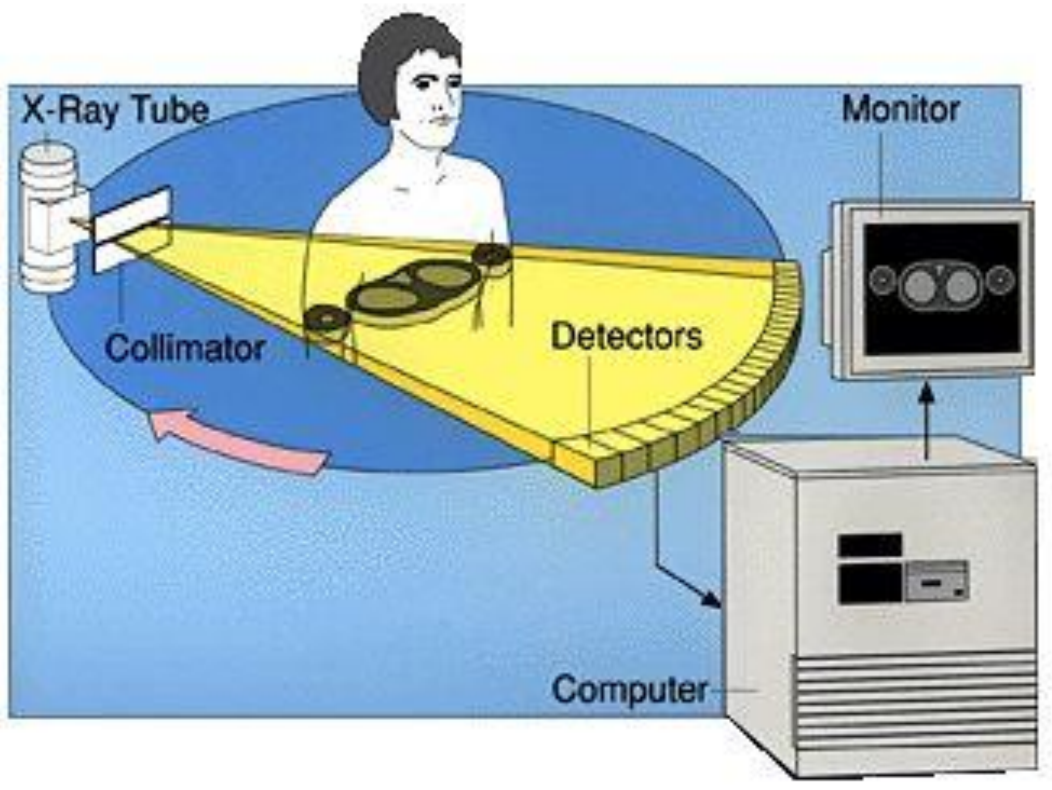
After DSP

www.dxo.com

Multimedia Compression

- Provide the crucial technology for:
  - WWW with multimedia content (e.g. audio, image, and video)
  - DVD
  - Digital cameras, camera phones
  - MP3, iPod

Medical Imaging: Ultrasound (US), Computer Tomography (CT), Magnetic Resonance Imaging (MRI), …

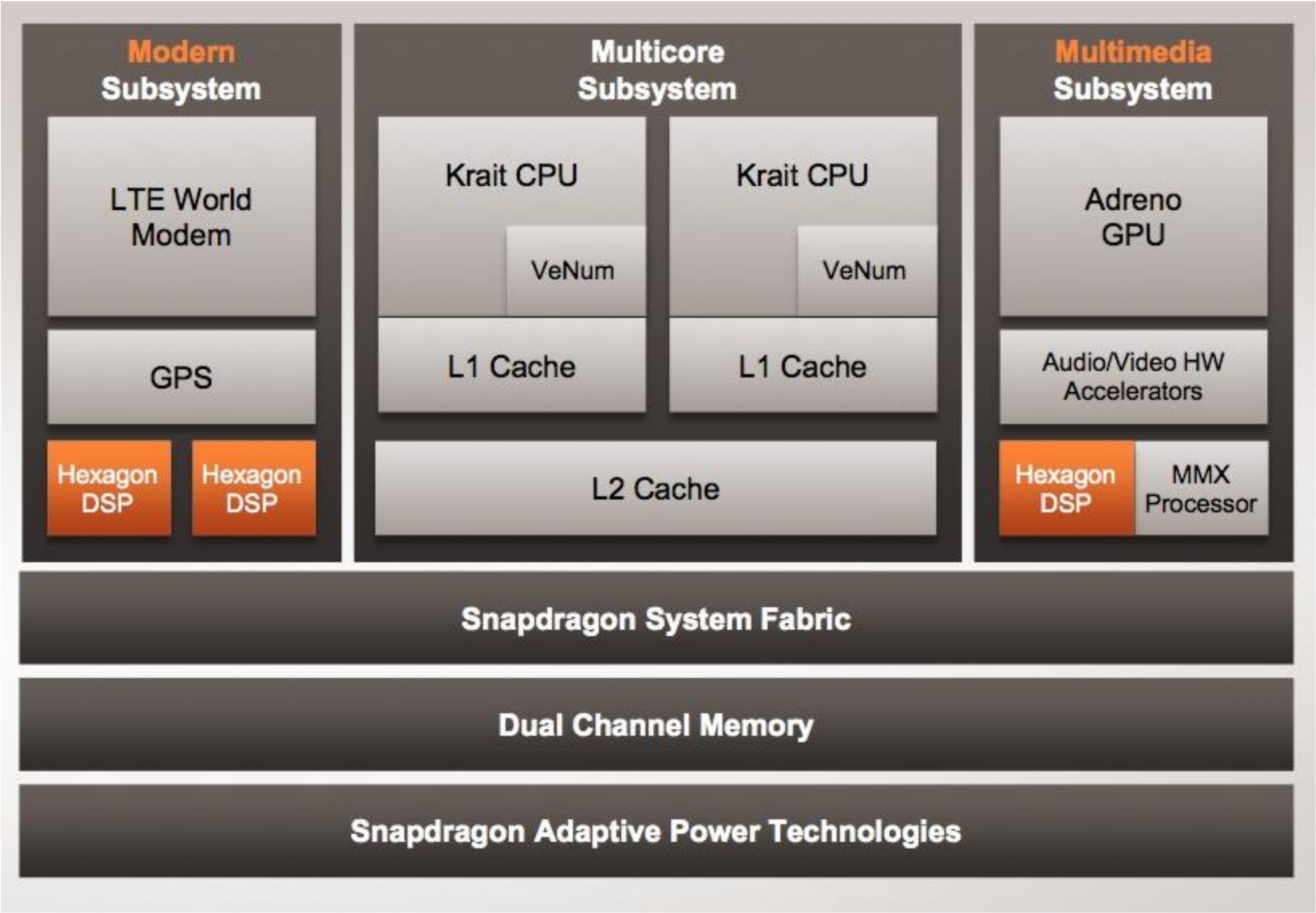


www.imaginis.com/ct-scan

DSP Appliances

Smart Phones

Example Smartphone Chip



Best Practices in Developing DSP Software: Systematic Debugging

- First, develop and test DSP algorithms in high-level languages (Python, MATLAB)
  - More rapid development, more extensive tools available
  - Use of test/training signals
    - Tuning of algorithmic parameters
    - Quantification of algorithm performance (SNR, detection accuracy)
    - Examination of intermediate / final signal outputs in various domains
  - Provides a reference for comparison against embedded implementation

- Then, port tested algorithms into embedded platform (Android)

- Sometimes, may need to go back and refine algorithms

Practical Considerations

- Reducing power is *critical* for mobile real-time devices
  - Battery drain is #1 reason for users to turn off an app
- Ways to save power
  - 16-bit fixed point, not floating point
  - Low clock speed/voltage through parallelism
  - Simple, low-power microprocessor architecture
  - Program in low-level languages
  - Use hardware accelerators, or dedicated computing units

Next Labs

Upcoming Labs

- Lab 1 – Pedometer
  - Not as signal processing heavy
  - Introduction to development tools and Android platform
- Lab 2 – Digital Filtering
  - Signal processing at its finest!
  - Work with audio signals, filter design, and the OpenSL framework

Filter Design: Mapping Analog to Digital Frequencies

If we sample an analog signal $x_a(t)$ to obtain a digital signal $x_d[n] = x_a(nT)$ using the sampling frequency $f_s = 1/T$, then their Fourier transforms are related by:

$$X_d(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a\left(\frac{\omega - 2k\pi}{T}\right).$$

Hence, assuming no aliasing (i.e. $X_a(\Omega) = 0$ for $|\Omega| \leq \pi/T$) then an analog frequency $\Omega = 2\pi f$ (where $|\Omega| \leq \pi/T$) is mapped to a digital frequency

$$\omega = \Omega T = \frac{2\pi f}{f_s}.$$

In particular, the Nyquist frequency $f = f_s/2$ is mapped to $\omega = \pi$.

Digital Filter Implementation

Given a digital filter

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_K z^{-K}}{1 + a_1 z^{-1} + \ldots + a_L z^{-L}},$$

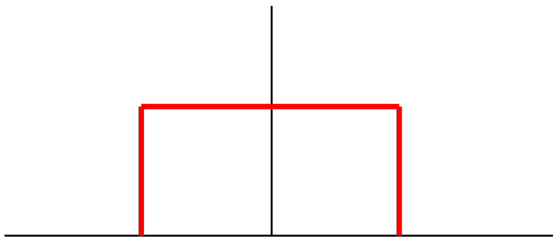
then the filtering by $H(z)$:

$$x[n] \longrightarrow \boxed{H(z)} \longrightarrow y[n]$$

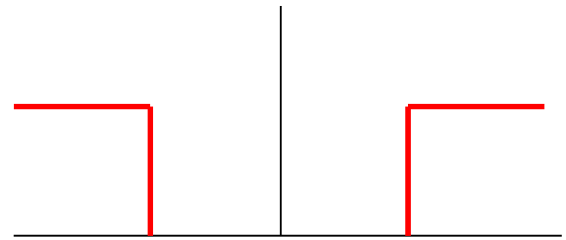
can be implemented for each n as:

$$y[n] = (b_0\, x[n] + b_1\, x[n-1] + \ldots + b_K\, x[n-K]) - (a_1\, y[n-1] + \ldots + a_L\, y[n-L]).$$

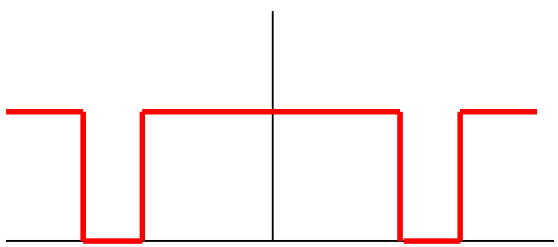
Examples of Filters



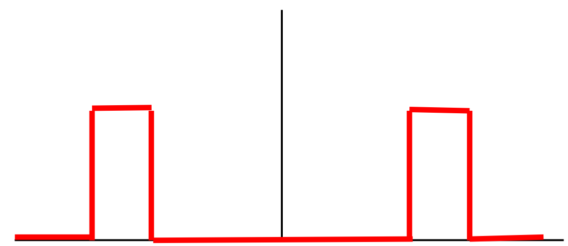
Low-pass



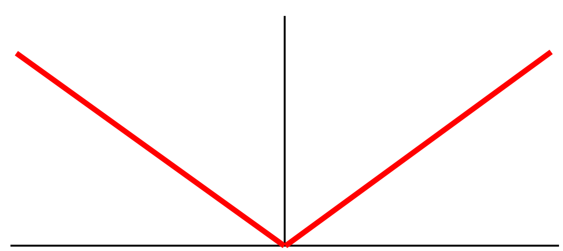
High-pass



Band-stop



Band-pass



Ramp

Digital Filter Design

- Typically filters have a target frequency response expressed in the Fourier domain
- We will be filtering in the spatial domain, and for most applications achieving a ‘perfect’ frequency response is impossible
- One of the biggest constraints will be computational limits
- How do you get the most out of a filter?
  - Design criteria – ‘optimal’ in some sense
  - FIR vs. IIR – complexity vs. fidelity

Summary

- Lab 1 this week (no prelab)
- No lecture next Monday (Jan 21)
- Lab 2 next week (with prelab)
- Let’s go do some amazing work!