ECE 420

Lecture 2

Jan 28 2019

The Compromise of Embedded DSP

- Embedded DSP is an inherent compromise of many, many constraints
  - Beyond just developing an algorithm so that it performs the desired action / produces the desired result

- Platform itself may be fixed a priori, or if a part of the design then cost considerations will limit what can be included

- Limits on
  - Computational capability
  - Memory speed and capacity
  - Word size (floating vs. fixed point)
  - Battery/power

The Compromise of Embedded DSP

- Real-time constraints can pose the biggest design challenges
  - Even “off-line” processing algorithms still need to complete in moderate short order for user satisfaction

- Real-time processing of data requires a certain level of throughput or perceived (and actual) performance will suffer!
  - Once the firehose of data is turned on, the algorithm must be able to process the data quickly enough or bad things will happen
  - Bad things = choppy/missing data in output streams, or malfunctioning algorithms = Unhappy users

- Data rates allow to perform envelope calculations about the target performance of your algorithm

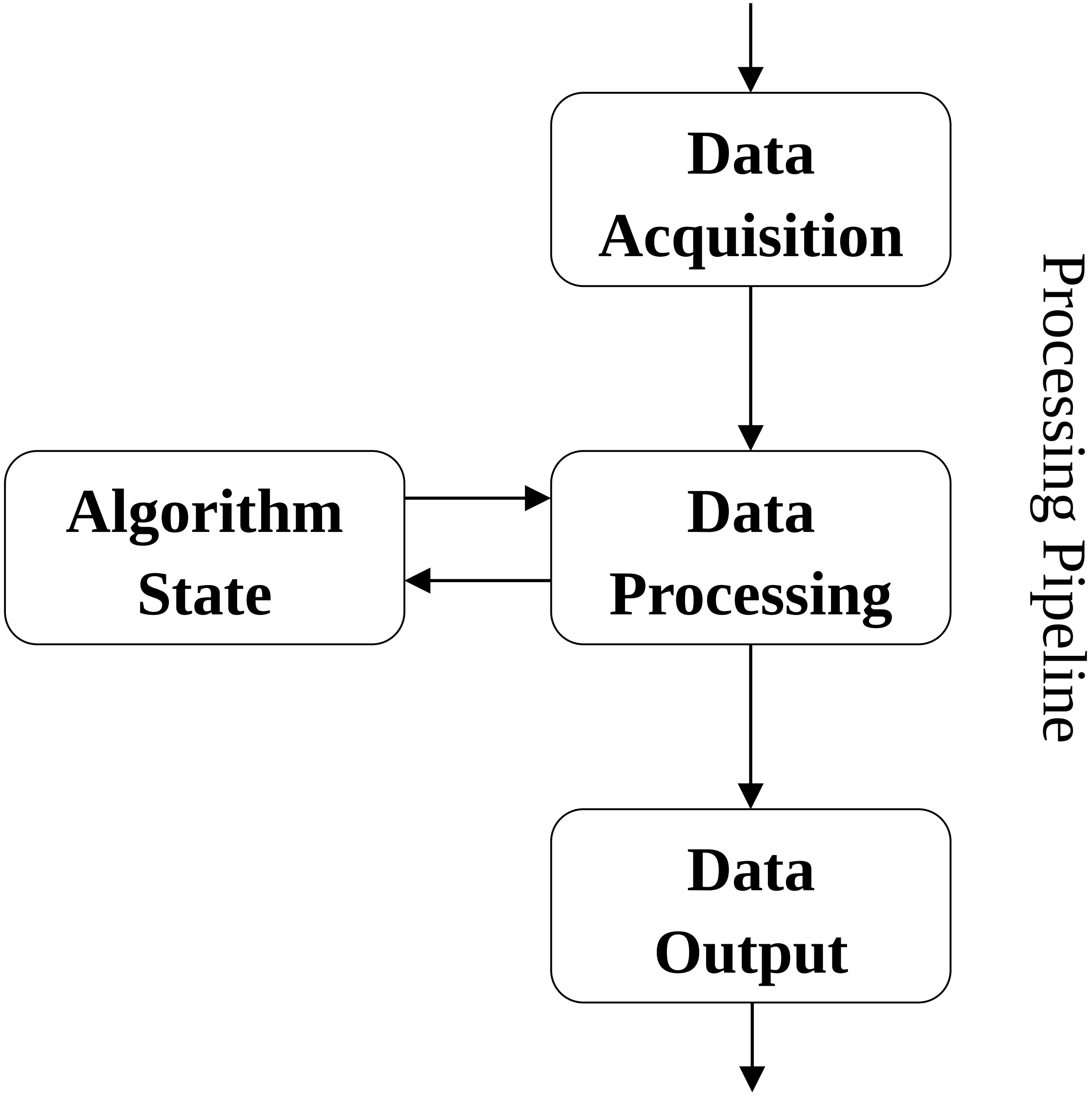
Block Data Processing

- Blocking up the data into batches helps address some of these constraints

- Both practically and theoretically motivated practice
  - We can't hold all the data in memory at once
  - We don't want to wait for all the data to produce output
  - Relative to the processing task at hand, the data tends to become uncorrelated given enough distance in time

- The information we want is available locally in the data
  - How do we get it out? What algorithm to use?

Block Processing

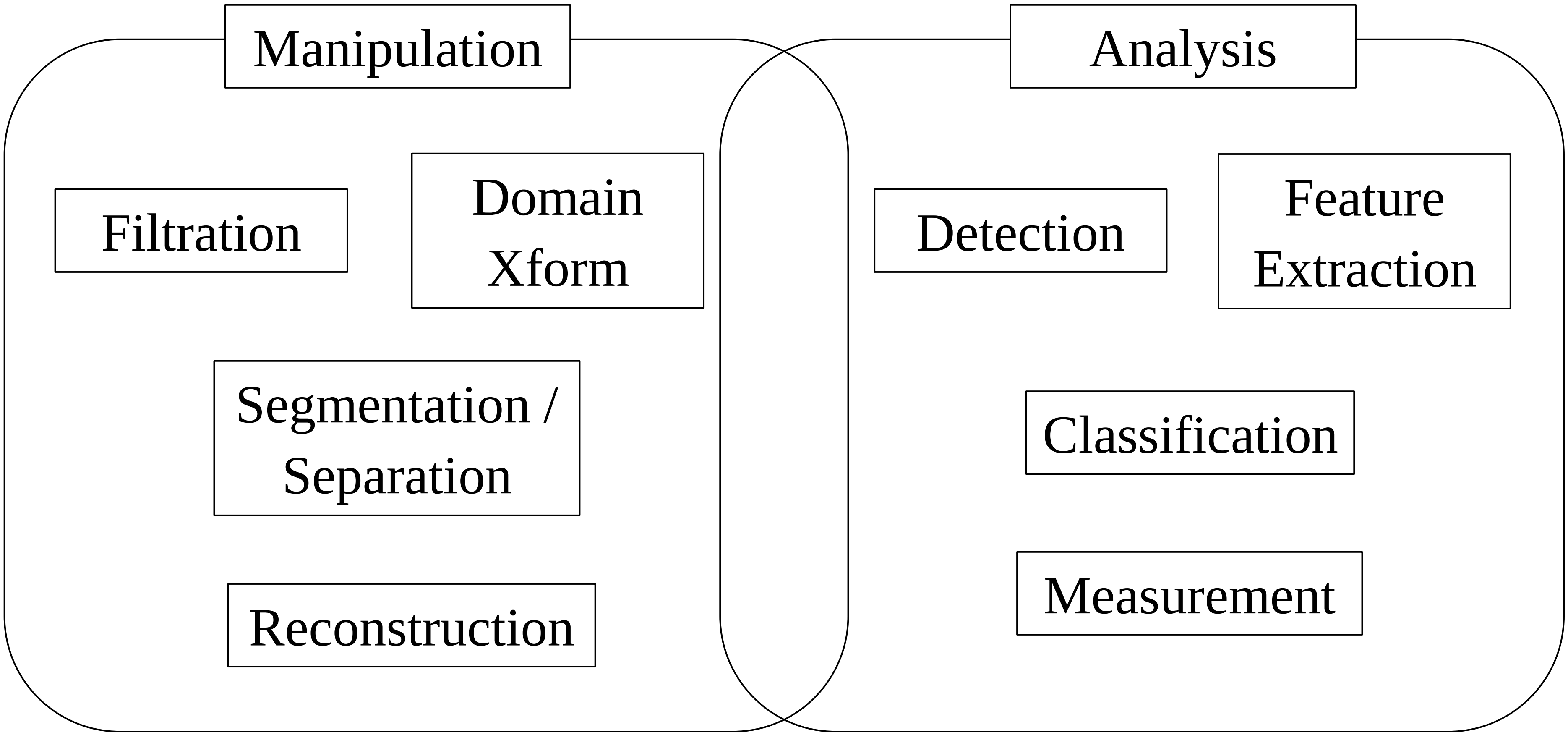


Block Data Processing

- Calculate properties of your algorithm
  - What is a practical/required block size of data?
  - How much computation is required?
  - How much memory is required?
    - Don’t forget working buffers and internal state of the algorithm!
  - How much time is allotted?

- Doing these high level calculations will help determine how well an algorithm will map to a particular embedded system
  - An extension of the ‘measure twice, cut once’ mentality

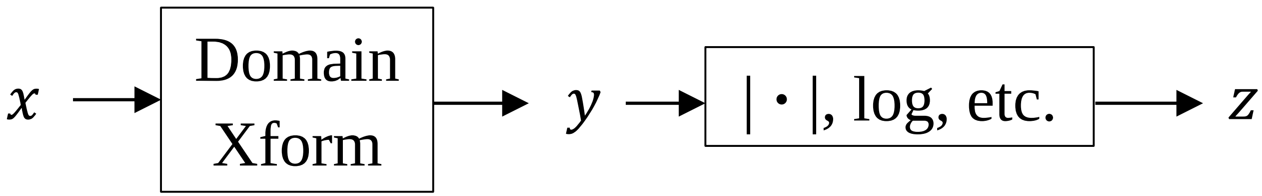
Signal Processing Atlas



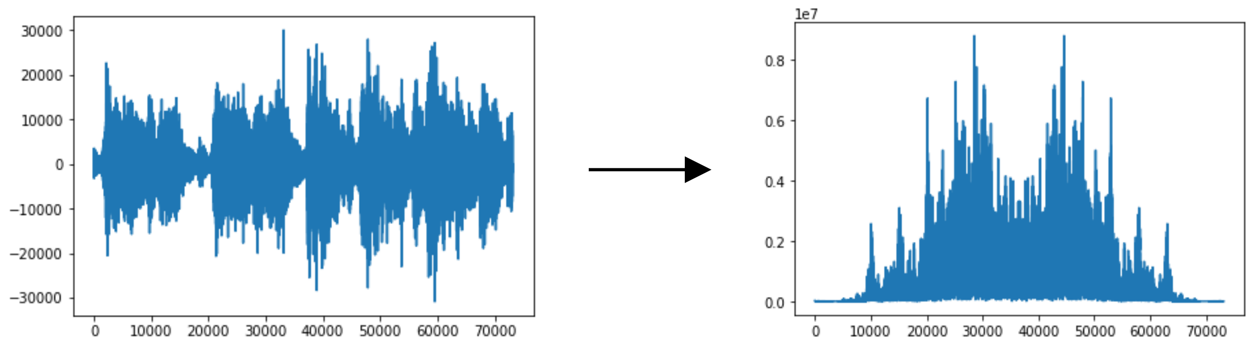
Lab Summary Thus Far

- Lab 1 – Pedometer
  - Detection Algorithm
- Lab 2 – Digital Filtering
  - Filtration
- Lab 3 – Spectrogram [This week]
  - Domain transformation (with a little analysis)

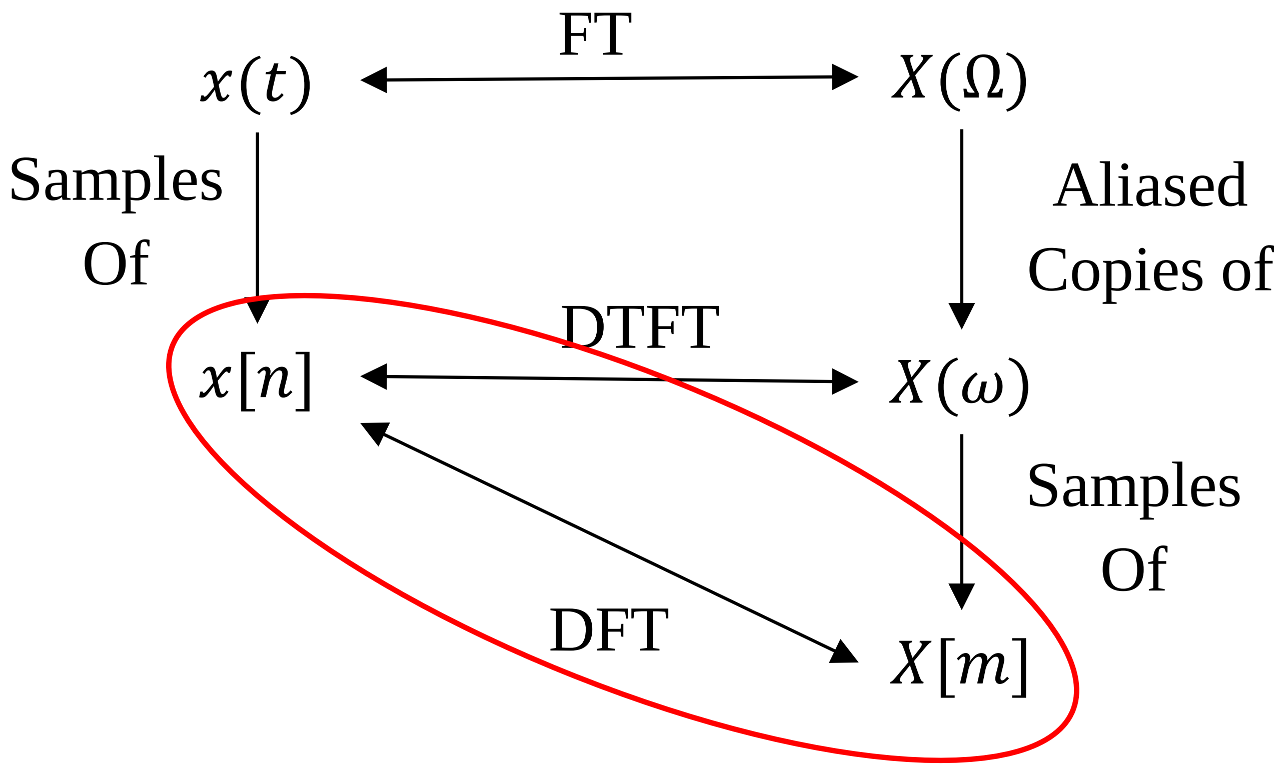
Spectral Analysis



- Relative distribution of signal ‘energy’ in a different basis
- Common choice is Fourier basis
- Magnitude/log or other post-processing if necessary or for human consumption



Fourier Transforms (An aside)

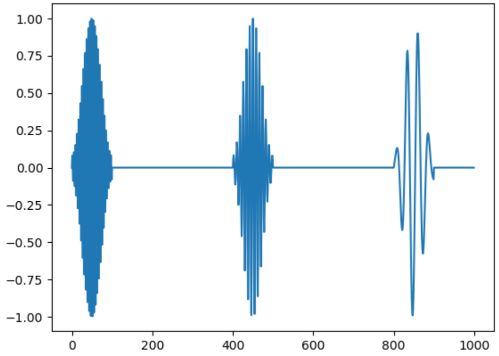


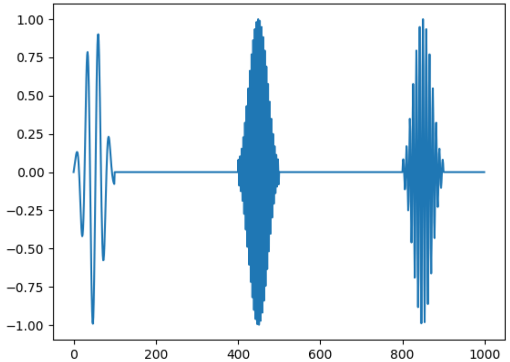
Our good friend the Fast Fourier Transform is which of these?

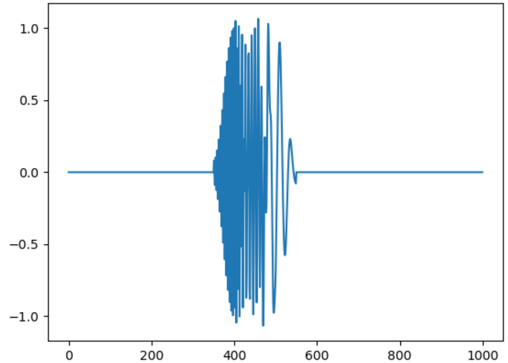
Digital Fourier Transforms (An aside)

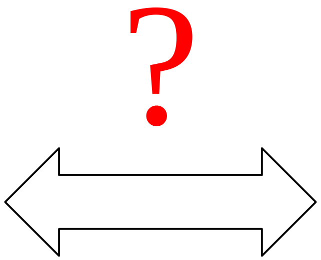
- Two useful properties of DFTs
- Element-wise multiplication in one domain equivalent to circular convolution in the other domain
- Zero padding in one domain equivalent to increased sampling density in the other domain

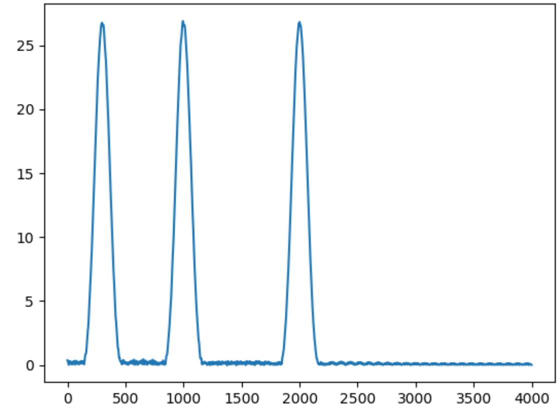
Fourier Transform Localization





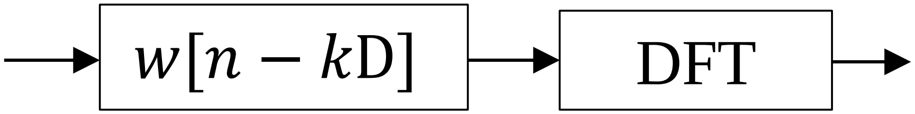






Short Time Fourier Transform

- Divide the signal into (windowed) blocks of data
- Apply DFT to each block
  - Take magnitude if desired
- Solves* our local spectral analysis problem

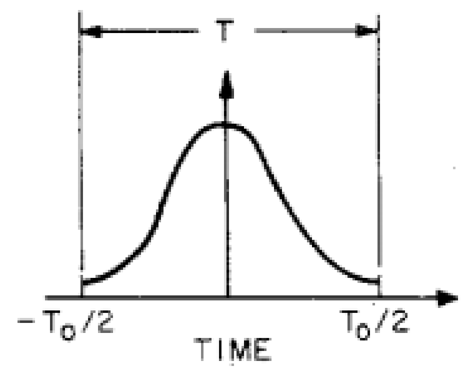


A Brief Word from Allen77

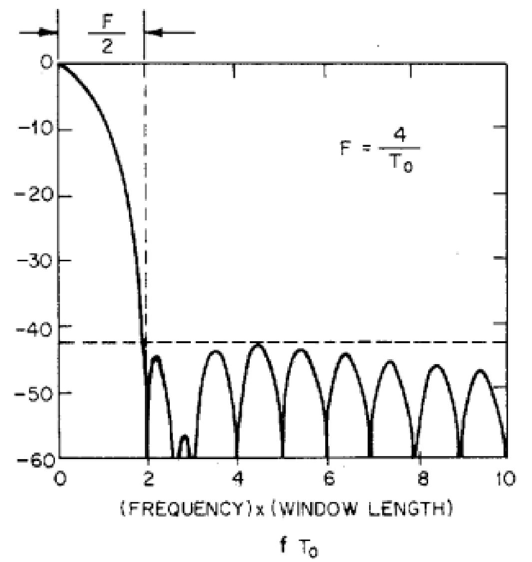
- Continuous STFT
  - $X(f,t) = \int_{-\infty}^{\infty} w(t-\tau)x(\tau)e^{j2\pi f\tau}d\tau$
- Assume window $w(t)$ is time limited and essentially bandlimited function
- Questions to be answered
  - What sampling is required in f, t?
  - How does the choice of $w(t)$ influence this?
  - How can this STFT be used as an analysis/synthesis filterbank?

Discretization of STFT

- Window function has effective time/frequency limits of T, F respectively
  - For Hamming window, $F = 4/T$



- Allen argues sampling of
  - $t_n = nD = n/F$
  - $f_m = m/T$



- Discrete STFT
  - $X_{nm} = DFT\{w[nD - k]x[k]\}$

Synthesis Properties of STFT

- Up to this point, discussed the *Analysis* properties of STFT
- What about *Synthesis* properties?
- Exploiting the property that $w[k]$ forms a partition of unity

$$\sum_{n=-\infty}^{\infty} w[nD-k] = 1$$

- The signal is directly recoverable

$$x[k] = \sum_{n} DFT^{-1}\{X_{nm}\}$$

Using Synthesis for Signal Manipulation

- Spectral-based filtering operations can be carried out in STFT domain
- Leverage Fourier multiplication == spatial convolution
  - Requires zero padding to prevent circular aliasing!
- Can implement Fourier based filtering routines using STFT when manipulation is a shift invariant filter
- Can also implement spatially varying filtration at your own risk

Limitations of STFT

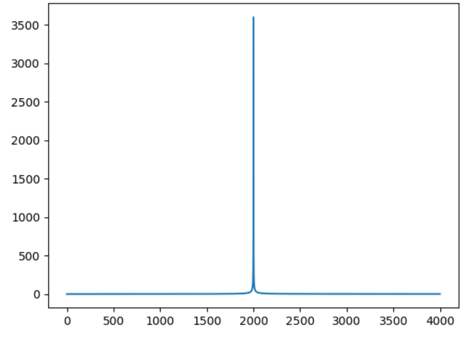
- When using overlapping windows, straightforward synthesis is possible
- This comes at the expense of extra data
  - Expansion of data = D/T = TF
  - Allen claims this allows for more stable synthesis when spatially varying filters are considered
- Time/frequency localization tradeoff not eliminated
  - Short $T \rightarrow$ Wide F and vice-versa

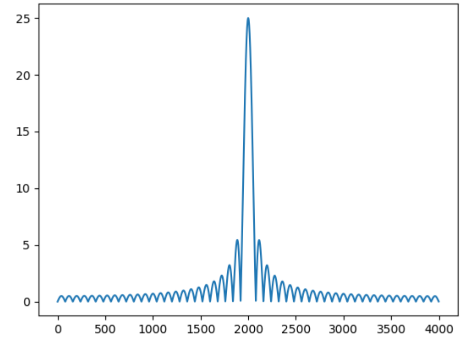
Signal Detection / Separation

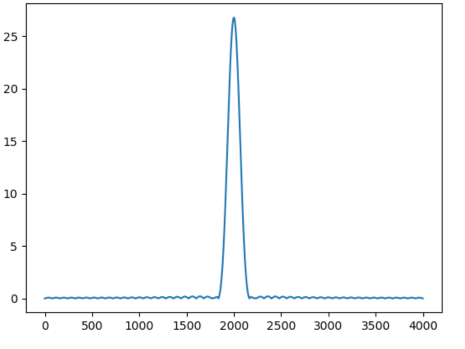
- Problem statement: we want to identify key components of a signal
- Example: Finding peaks in our spectrogram
- Question: Where are the peak locations and intensities?
- Challenging Scenarios:
  - Peaks closely spaced to one another
  - Peaks with relatively small amplitude
  - Peaks with relatively small amplitude spaced closely to one another

Effect of Windowing on Spectral Analysis

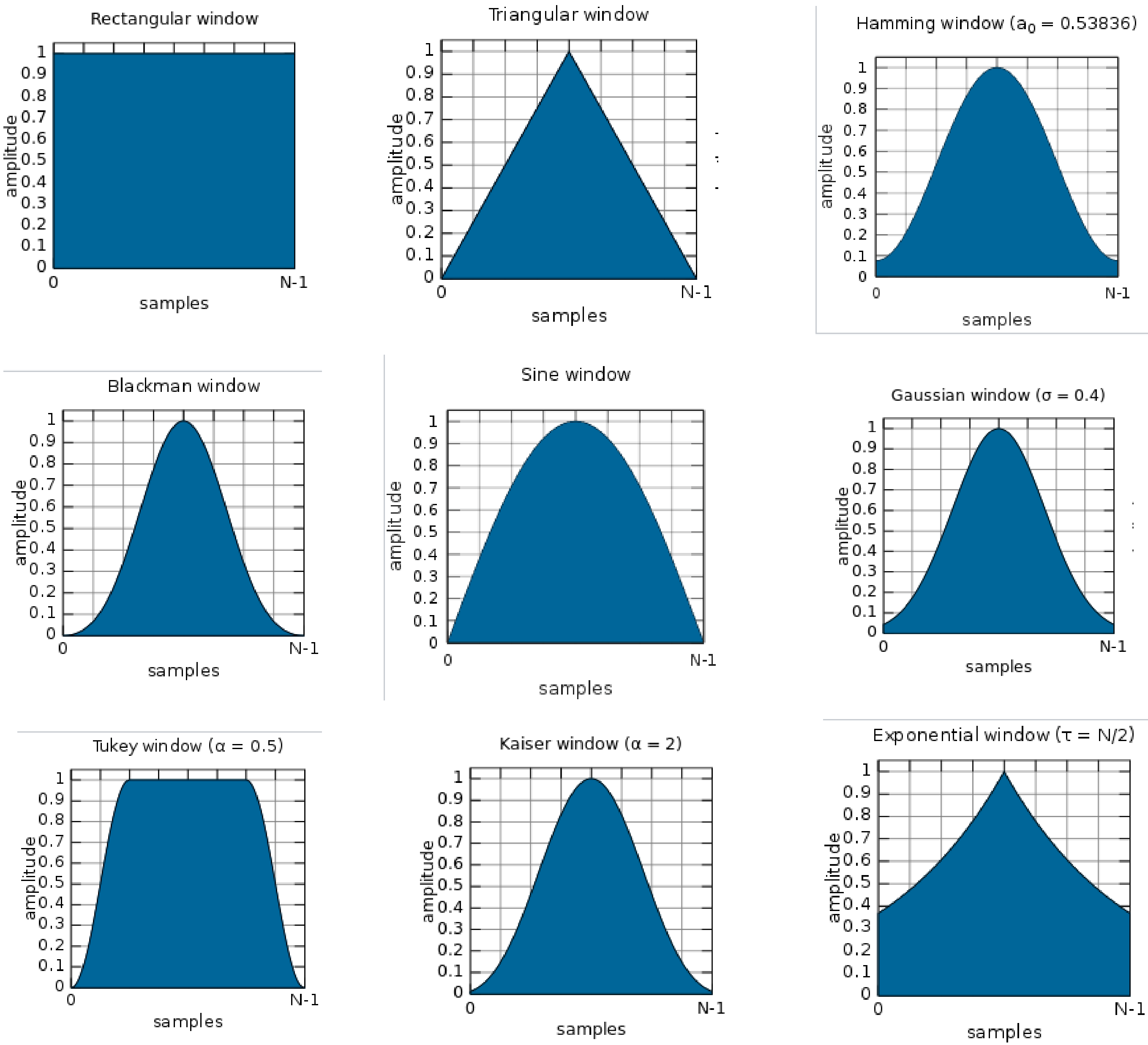
- Just by blocking the data we are applying a window
- DFT tells us windowing results in convolution in frequency domain
  - Signals that might once had good frequency localization become muddled
  - This convolution can be viewed as the response of our analysis system and characterizes the performance of the ultimate algorithm
- Why even care about windowing?







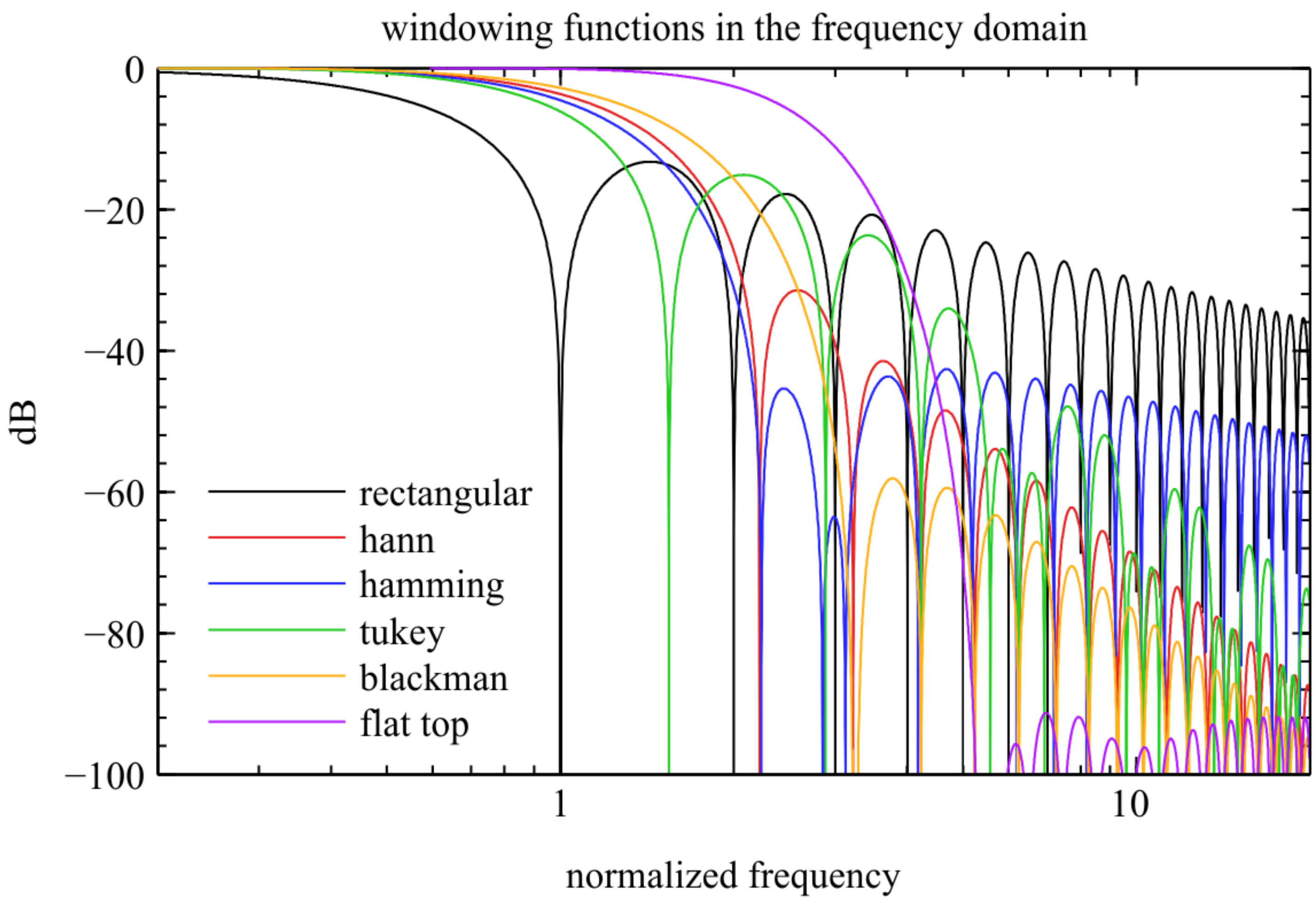
Window Selection



Window Selection

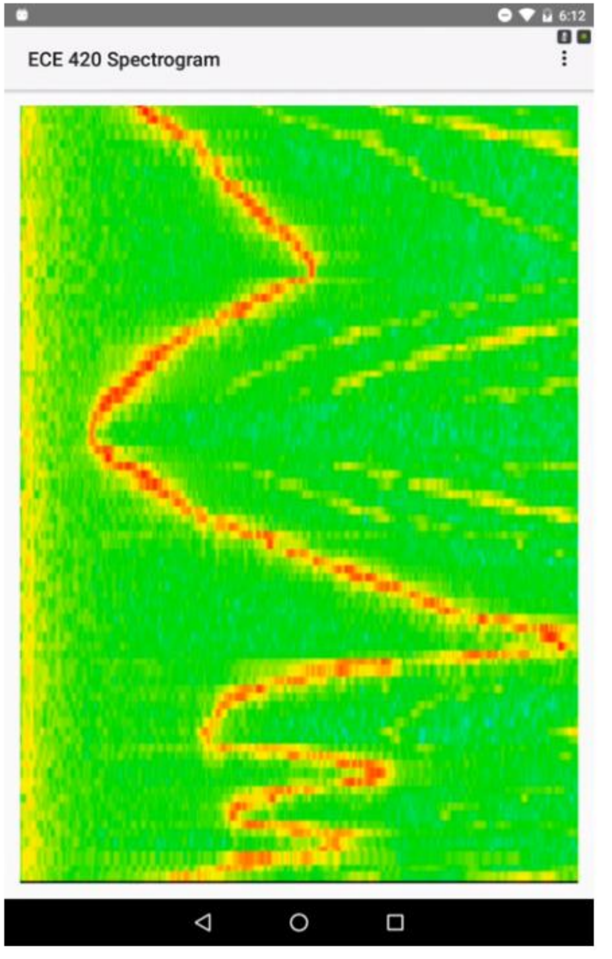
- Windows generally provide a tradeoff between mainlobe width and sidelobe height (or sidelobe decay)
  - Determines how well we can resolve neighboring signals
- Generally a good idea to select a window with an effective bandlimit
  - This is why not explicitly applying a window is not a good strategy
- From a detection performance perspective, we want the transform to be as close to a delta as possible
  - In practice we have to decide on the tradeoff

Window Comparison



Real Time Spectral Analyzer

- Block processing and analysis on the tablet using STFT
- Move to floating point data processing
- Prize for fastest implementation of the FFT algorithm
- (KIDDING)
- ALWAYS leverage existing libraries unless explicitly told not to
  - Correct
  - Well tuned/optimized



This week

- Lab 2: Digital Filtering Quiz/Demo
- Lab 3: Real time spectral analyzer
- Be thinking about Assigned Project Labs
    - More info at lecture next week