ECE 420

Lecture 3

Feb 4 2019

Or

Everything you wanted to know about Glottal Waveforms and Laryngeal Excitation Cycles but were afraid to ask

Converting Frequency Domains

- A common question: A particular digital frequency corresponds to what analog frequency?
- Option 1: Work through the equations relating the various transforms to find your answer
  - FT: $X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}\,dt$
  - DTFT: $X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$
  - DFT: $X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi mn/N}$
  - Replacing x(t) with samples at x[nT]
    - $\sum_{n=-\infty}^{\infty} x(nT)e^{-j2\pi fnT}$
    - So $\omega = 2\pi fT$
  - DFT/DTFT easily related, and then also DFT/FT
    - $\omega = 2\pi m/N$
    - $f = m/NT$

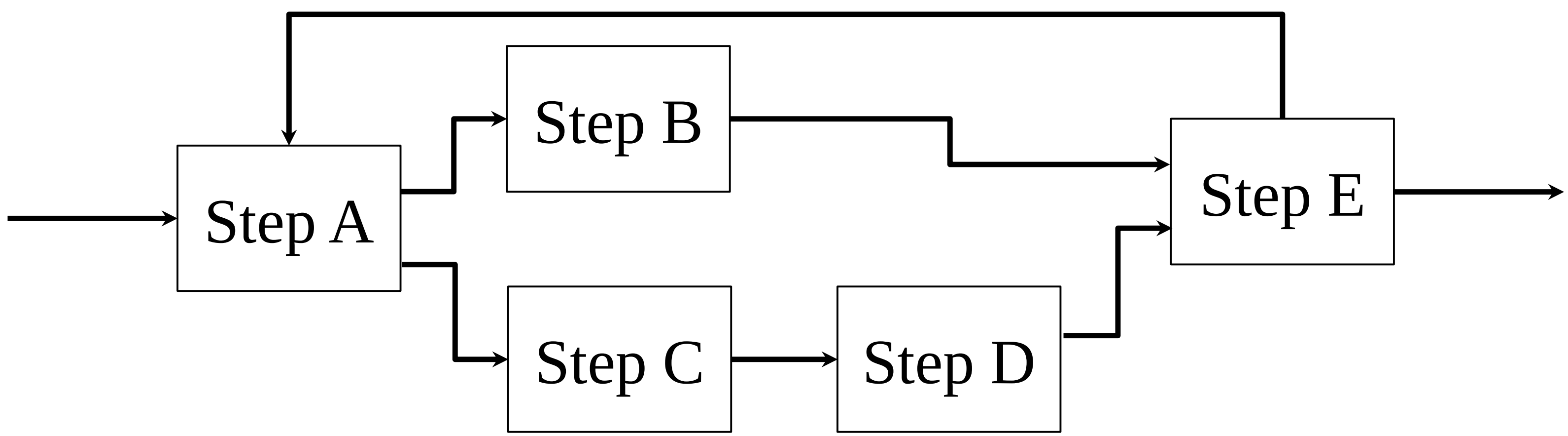
Converting Frequency Domains

- Option 2: Recall that these are all linear relationships, and use ratios based on maximum frequency
- 'Max frequency' of analog signal is $1/T \triangleq F_s$
- DTFT, DFT have periods of $2\pi, N$ respectively
- Yields the relationships $\frac{m}{N} = \frac{\omega}{2\pi} = \frac{f}{F_s}$
- Allows to fluidly convert in either direction
- Always 'gets the units right' (Hz, rads, normalized freq., etc.)

Always think: Ratios!

Composite Algorithms

- Algorithms composed of one or more steps/pieces
- Might involve decision points ("control flow") or combinations from different paths
- Allows for sophisticated algorithms but increases system complexity substantially
- Flow graph helps manage system and convey operation to others



Implementing Composite Algorithms

- Always implement components as separate functions/classes
  - Sometimes these will be in different libraries anyway
- Separation of concerns allows for better focus to the code, generally resulting in better coding style and code that is much easier to debug
- Creates code segments that can be portable among projects and not tied to a specific implementation

A Note on Testing

- Good practice to validate components of your code first
  - Writing each step as separate functions facilitates for good unit testing
  - Increases probability of success when putting everything together
- Types of testing
  - Unit
  - Integration
  - System/End-to-End
- These testing categories get cover progressively larger aspects of the system, but at higher cost

Unit Testing is your friend!

A Note on Testing

- How do you know if something is working?
  - Test signals!
- What test signals do you use?
  - They might be provided for you
  - You can make them up
  - You can use a repository of test data
  - You can acquire them
- Classes of test data
  - 'Easy' vs. 'Hard'
  - 'Noisy' vs. 'Clean'
    - Different noise characteristics
  - Corner cases

A Note on Testing

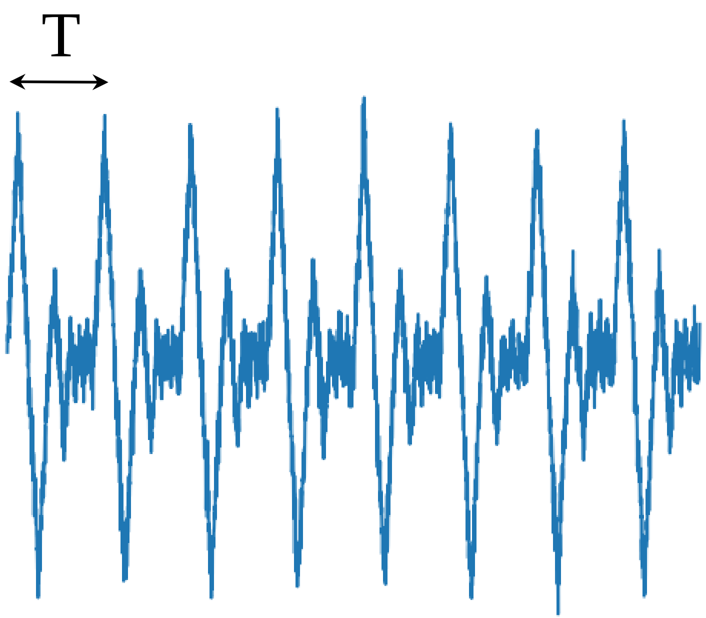
- A key is to have ‘high quality’ data to establish ‘ground truth’
  - Nothing is worse than getting misled about algorithm performance due to incorrect data
  - Allows for quantitative measures of performance if sufficiently large set available
- Simple and clean data can facilitate establishing basic algorithm functionality
- Always a good idea to at least try with a non-ideal test signal as a sanity check
  - Some algorithms can catastrophically break down when certain assumptions are not met or noise is introduced!

Test Database / Benchmarks

- Huge value!
- Provides a suite of test cases
  - Ground truth for evaluation of algorithm efficacy
  - Common set to allows for apples-to-apples comparison among algorithms
  - ‘Real world’ conditions
- Facilitates automatic tuning and evaluation of algorithms
- Prevalent in many areas of engineering

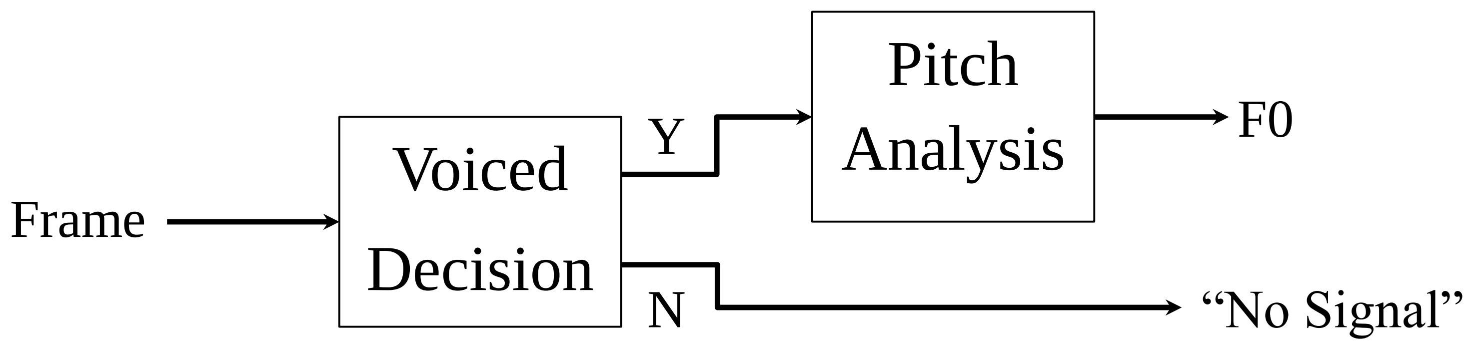
Pitch Detection

- This lab is a pitch detection implementation
- What exactly is pitch? Not actually strictly defined.
  - Period of signal in time domain
  - Fundamental frequency in spectral domain (caution!)
  - Perceived frequency of signal by human ear
  - Anatomically as distance between pulses/cycles of larynx



Basic Pitch Detection Algorithm

- Operate on a single frame of audio input
- Initial decision: voiced or unvoiced?
- If voiced, perform pitch analysis and report pitch frequency



Challenges of Pitch Detection

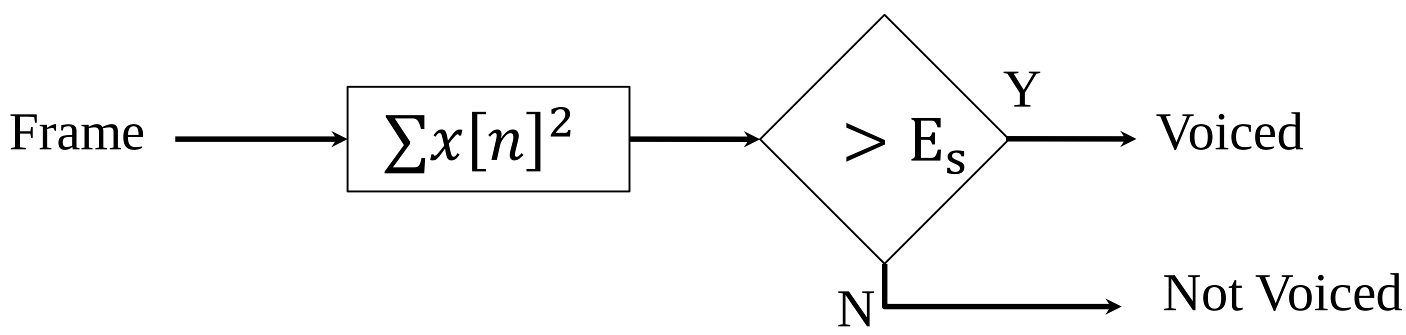
- Frame size selection
  - Likely not an integer multiple of period length
  - Probably need at least one period, maybe more
  - Larger frame sizes = more latency
- Pitch values may not be integral values of sampling rate / frequency bins
  - May distort analysis sufficiently to pick improper bin
- Speech vs. Music
- Non-ideal environmental situation
  - Multiple concurrent signals
  - Noise
  - Bad microphone quality

Categorization of Frames

- Three coarse categorization of frames
- Voiced sounds – generated by vibration of the vocal cords
- Unvoiced sounds – speech generated without vibration of the vocal cords
- Silence / noise – no active speech
- We will only be concerned with classifying and measuring pitch for Voiced frames

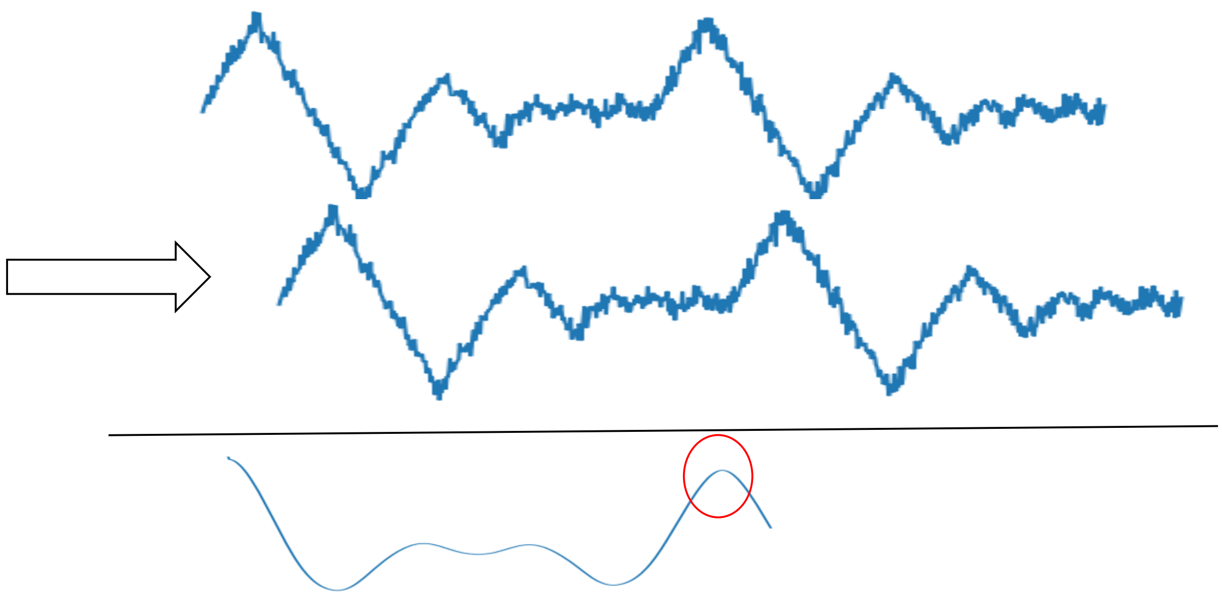
Voiced Decision

- Voiced signals tend to be louder and more sustained
- Unvoiced signals have more abrupt characteristics
- Over the span of a frame, we would therefore expect voiced signals to have a higher energy than unvoiced/silent frames
- Calculate the energy of the frame, compare to a threshold for voiced decision



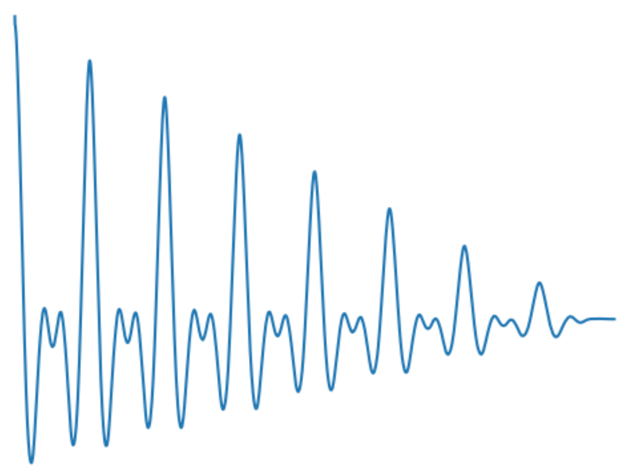
Pitch Calculation - Autocorrelation

- $R_{xx}[l] = \frac{\sum_{n=0}^{N-1} x[n]x[n-l]}{\sum_{n=0}^{N-1} x[n]^2}$
- Provides an estimate of how self-similar a signal is given a particular delay (lag) l
- When offset corresponds to period of signal, values coherently combine to yield a large correlation value

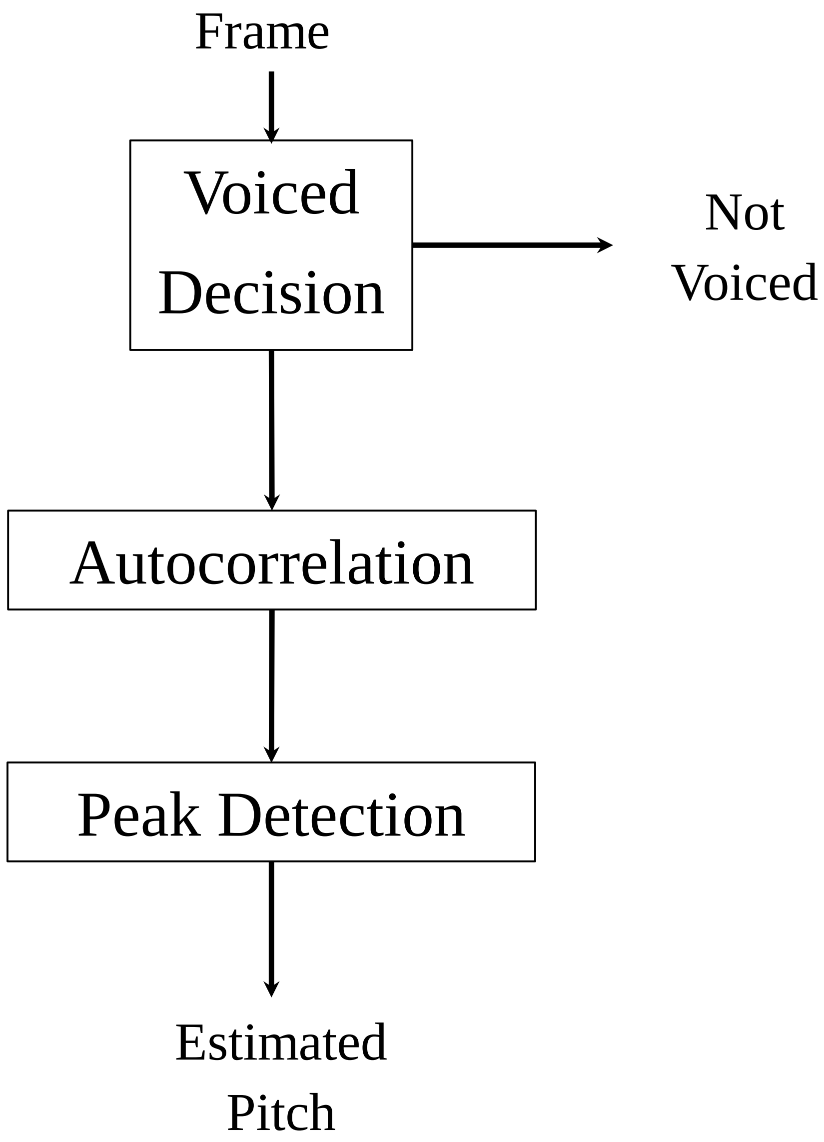


Challenges of Harmonics

- Pitch estimation searches for a peak value in the autocorrelation function
- Two main challenges
  - Lag of 0 will generally have the largest overall value, but is certainly not the value we are looking for
  - Multiples of the pitch (higher order harmonics) will also yield local peaks but must be rejected
- Solving these challenges is part of the Lab design objectives



Basic Pitch Detection Algorithm



Algorithmic Complexity

- "Big O" notation
  - $O(N), O(N^2), O(\log N), O(1), O(2^N)$
- Key property of complexity analysis is that the higher order term dominates
  - $O(N + N^2) = O(N^2)$
- Always a good idea to know the complexity of the algorithms you are using
  - Know what you are getting yourself into!
- For a complicated composite algorithm, each step might have different complexity
  - Knowing complexity tells you where the dominant computation step
  - Optimize there!

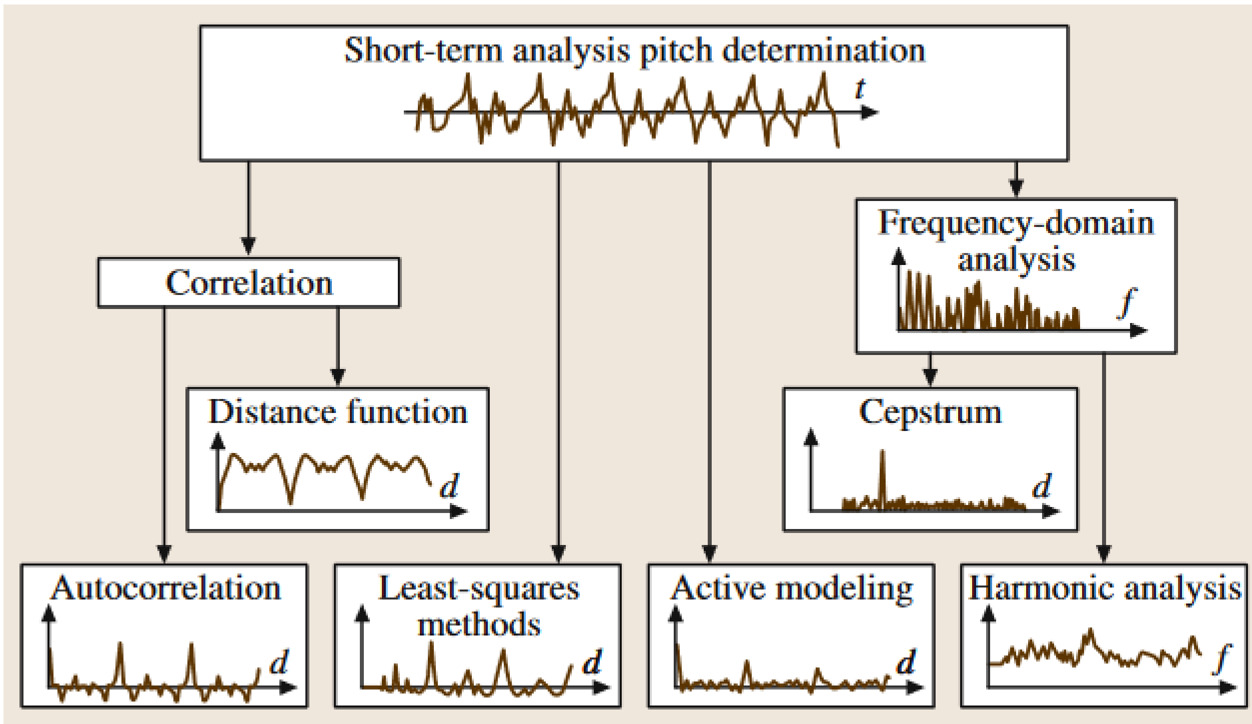
Efficient Cross-Correlation

- $R_{xx}[l] = x[n] * \tilde{x}[n]$ is an $O(N^2)$ operation
  - What is complexity of energy calculation step?
  - What is complexity of peak detection step?
- Calculate correlation using $R_{xx}[l] = F^{-1}\{F\{x\}F^*\{x\}\}$
  - $F^*\{x\}$ denotes complex conjugate
- Replace with an $O(NlogN)$ algorithm
  - What is the dominant component of our system now?
- Can actually perform joint voiced/unvoiced and pitch analysis with just R_{xx}

Postprocessing

- Improve overall pitch values by post-processing of the estimates
- Smoothing of pitch values
  - Linear
  - Non-linear (e.g. median)
  - Can also apply as a pre-processing if we wish (with caution!)
- Consider evolution of pitch value over successive frames
  - Attribute cost associated with changes in pitch and transition voiced to/from unvoiced
  - Can work in concert with multiple pitch estimates to pick ‘the best’

Categories of Pitch Analysis (subset)



Algorithm Selection

- Dozens of pitch detection algorithms out there
  - How do you pick one? (Or decide to develop your own?)
- What is your problem domain?
  - Are the algorithms targeted at this domain?
- Is it possible to quantify performance and compare?
  - Training/test data
  - Different aspects to compare: measurement accuracy, computational speed, output latency
- What are your requirements?
  - What is most important? What compromises can be made?
  - Can an algorithm be modified to meet those requirements?
- The more specific you are, the more discriminating you can be regarding algorithm selection

YIN Algorithm

- Distance Metric
  - $d(\tau) = \sum(x[n] - x[n+\tau])^2$
- Cumulative normalized distance
  - $d'(\tau) = d(\tau)/[1/\tau \sum_{j=1}^{\tau} d(j)]$
- Absolute threshold
  - $argmin_\tau \{d'(\tau) | d'(\tau) < E\}$
- Parabolic Interpolation
  - Expand $d'(\tau)$ analytically and find minimum
- Best Local Estimate
  - Examine small frame location offsets for better $d'(\tau)$

Algorithm	Gross Error %
Baseline Autocorrelation	10.0
Distance Metric	1.95
+ Cumulative Norm.	1.69
+ Absolute Thresh	0.78
+ Parabolic Interpolation	0.77
+ Best Local Estimate	0.50

‘Ad-hoc’ Algorithm Development

- Identify weak points / scenarios in the algorithm
  - Large set of training data is helpful
- Augment the algorithm
  - Change existing parameters / steps
  - Add extra steps
- Validate the changes
  - Did performance improve in the desired areas?
  - Did good performance persist in other areas (no regressions)?
- Emergent complex system
  - Potentially very powerful
  - Potentially very hard to analyze
  - Potentially very hard to maintain

Assigned Lab

- Lead-in to the Final Project
  - Forming up groups for both Assigned Lab and Final Project
- Explore a DSP algorithm from the literature
  - Implementation in Python for this stage, NOT on tablet yet
- Proposal for Assigned Lab
  - Overview of proposed algorithm, cite source(s)
  - Plan for testing and validation
  - Rough idea(s) for Final Project application
  - Due March 1
- Assigned Lab Report upon completion, demo incorporated into Final Proposal design review

This week

- Lab 3: Real time spectral analyzer Quiz/Demo
- Lab 4: Pitch Analyzer
- Be thinking about Assigned Project Labs / Groups
  - Early vetting of ideas is fine!