Yandex

SCHOOL OF DATA ANALYSIS

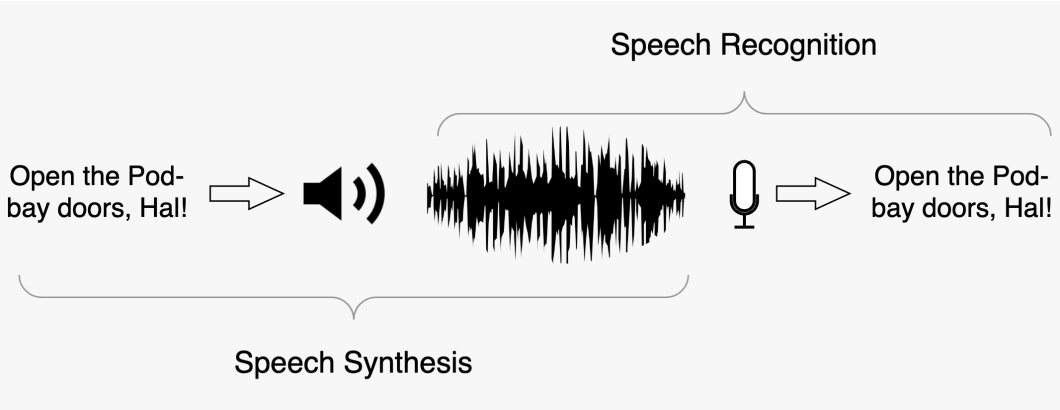
Digital Signal Processing for Speech

Andrey Malinin, Vladimir Kirichenko, Sergey Dukanov

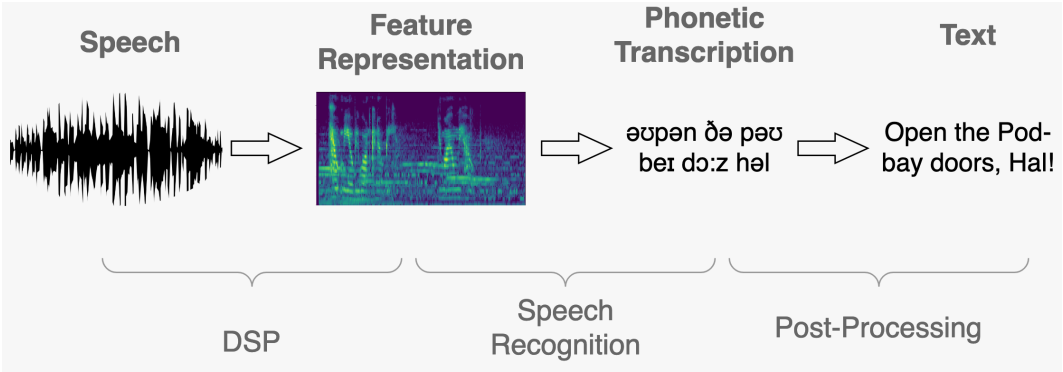
14th February 2022

In this previous episode...

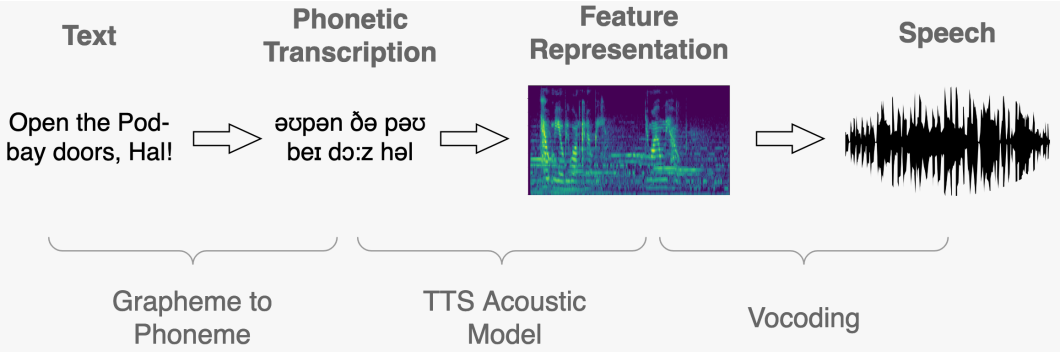
Story so far


Speech Recognition
Open the Pod-
bay doors, Hal!
Open the Pod-
bay doors, Hal!
Speech Synthesis

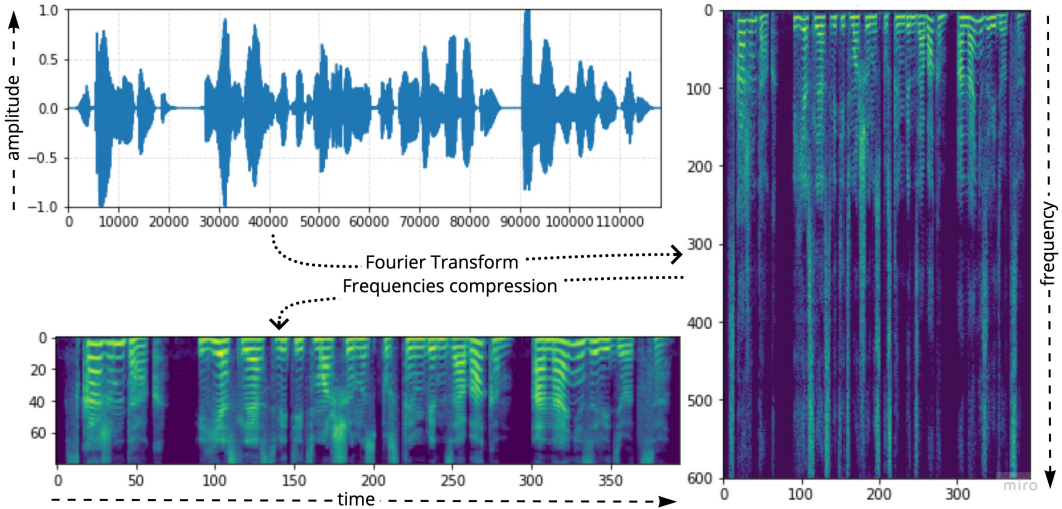
Recap - High-level Speech Recognition Pipeline



Recap - High-Level Speech Synthesis Pipeline



DSP basics - Time and Frequency Compression



How can we extract useful representations of speech?

How can we reconstruct speech from these representations?

Lecture Structure

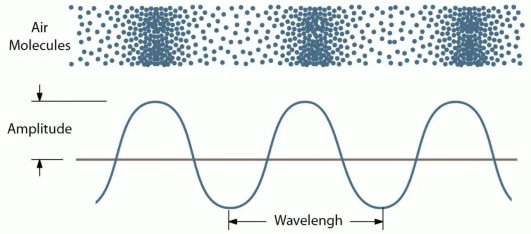
- Representing sound as discrete digital signal
- Mathematical Foundations of DSP
- Spectral representations of speech
- Reconstructing speech from spectral representations

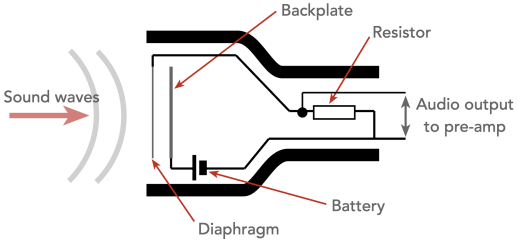
Lecture Structure

- **Representing sound as discrete digital signal**
  - Discretization of time and amplitude
- Mathematical Foundations of DSP
- Spectral representations of speech
- Reconstructing speech from spectral representations

What is sound?

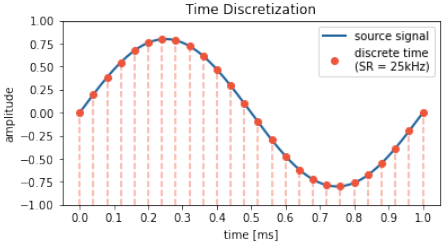
- Sound waves – longitudinal waves travelling through the air
  - Defined by **Amplitude** (pressure) and **Frequency** (inverse of Wavelength)
- A microphone captures these variations converting them into analog signal
  - Analog signal amplified, filtered and passed to ADC (analog-digital converter)

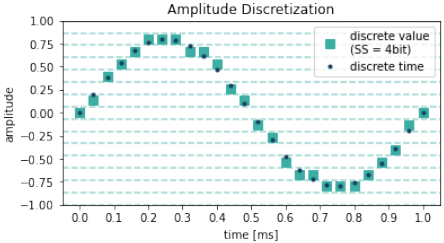




Waveform as Pulse-Code Modulation

- Digital signals are discrete in time and amplitude
- Time discretization $\rightarrow$ sample analog signal at constant sample rate
  - Sample rate – number of audio samples per second (8kHz, 22.05kHz, 44.1kHz)
- Amplitude discretization - round continuous amplitude to nearest discrete value
  - Precision depends on bit rate – number of bits per sample (eg. 8, 16, 24, 32 bits)
- Number of channels – number of signals recorded in parallel (ex: mono vs. stereo)



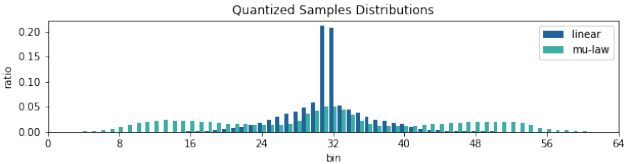


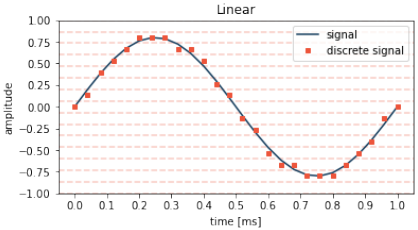
Non-linear Quantizations

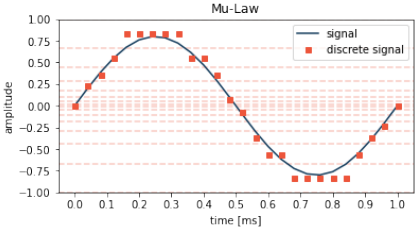
- Linear quantization encodes values uniformly across dynamic range, but most of information is contained near 0.
- μ-law encoding tries to solve the problem by re-scaling quantization bins:

$$f_{encode}(x) := sign(x)\frac{\log(1+\mu|x|)}{\log(1+\mu)}$$

$$\mu = 2^{\text{bits}} - 1, x \in (-1, 1)$$







Audio Formats

- **Non-compressed** formats: `WAV`, `raw` (header-less PCM), etc.
- **Lossless** compression formats: `FLAC`, `ALAC`, etc. (a compression ratio is about 2:1)
- **Lossy** compression formats: `MP3`, `Opus`, etc. (10:1)
  - Lossy compression formats are exploiting **psychoacoustics**
- **Bit rate** and **Sample rate** measure degrees of compression

Properties of Signals - Power

- Consider a discrete-time signal $\{f_n\}_{1:T}$ of length T.
  - The energy and power of the signal are defined as:

$$E = \sum_{n=1}^{T} f_n^2, \quad P = \frac{1}{T} \sum_{n=1}^{T} f_n^2$$

- The signal energy and power is typically computed for a fixed window τ:

$$E_t = \sum_{n=t}^{t+\tau} f_n^2, \quad P_t = \frac{1}{\tau} \sum_{n=t}^{t+\tau} f_n^2$$

- The *signal mean* is negligible, so the variance of amplitude is equal to power

$$P_t = \frac{1}{\tau} \sum_{n=t}^{t+\tau} f_n^2 = \mathrm{Var}\left[\{f_n\}_{n=t}^{t+\tau}\right] = \sigma_t^2$$

Properties of Signals - Decibels and Signal-to-Noise Ratio (SBR)

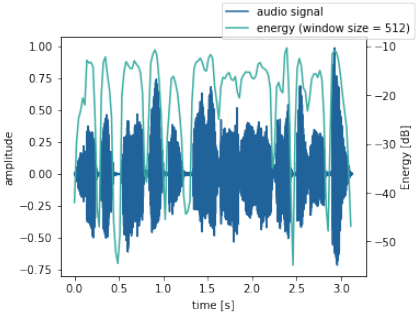
- Signal energies have a large dynamic range
  - A logarithmic scale is more informative
- **Decibels** are a logarithmic unit of energy
  - Closer to human perception of acoustic energy.
- Decibels defined in terms of energy (power):

$$P_t(dB) = 10 \log_{10} \sigma_t^2 = 20 \log_{10} \sigma_t$$

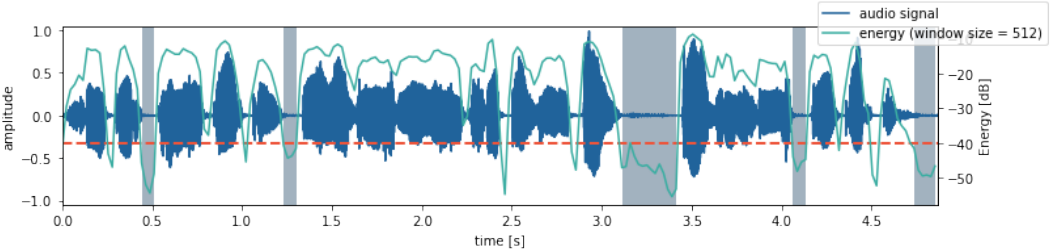
- Decibels often used as a relative power:

$$SNR_{dB} = 20 \log_{10} \frac{\sigma_t(f_{signal})}{\sigma_t(f_{noise})}$$

- Here, noise power is used as a reference



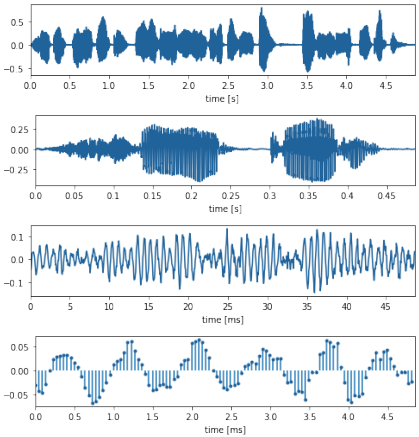
Energy-based Voice Activity Detection (Naïve approach)



- Find intervals (ignore the short ones) with energy lower than the threshold
  - Simple (+)
  - Not robust when SNR is low and fails at low-energy (unvoiced) sounds (-)
- ▸ Audio – audio with beeped silence regions

Are waveforms enough for sound representation?

- Advantages of waveforms:
  - East to obtain
  - Easy to synthesize via DAC
- Disadvantages of waveforms:
  - Extreme dimensionality (44K samples / sec)!
  - Periodic structure are different time scales
  - Inaudible variations (time shift, amplitude scaling) produce different audio signals.
- Need a more compact and robust representation!



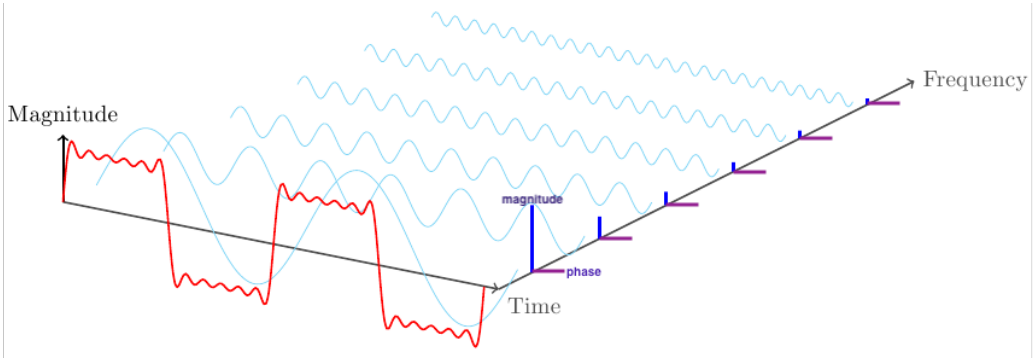
Lecture Structure

- Representing sound as discrete digital signal
- **Mathematical Foundations of DSP**
  - Continuous and Discrete Fourier Transforms
  - Sampling and Aliasing
  - Windowing and Short-Time Fourier Transform
- Spectral representations of speech
- Reconstructing speech from spectral representations

Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP
  - **Continuous and Discrete Fourier Transforms**
  - Sampling and Aliasing
  - Windowing and Short-Time Fourier Transform (STFT)
- Spectral representations of speech
- Reconstructing speech from spectral representations

Periodic functions basis



- We can try to represent functions (from time argument) in basis of periodic functions (from frequency argument) parameterized by **phase** and **magnitude**.

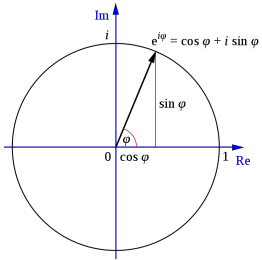
▸ Cool Demo!

Euler's Formula and the Complex function basis

- Complex numbers of the form $x + i \cdot y$ can be represented in magnitude-phase space via Euler's formula

$$\rho e^{i\phi} = \rho \cos\phi + i \cdot \rho \sin\phi$$

- Here magnitude $\rho = \sqrt{x^2 + y^2}$ and phase ϕ is $\arctan(\frac{y}{x})$



Fourier Transform

- Fourier Transform maps a **time domain** signal to the **frequency domain**.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i t \omega} \, dt \qquad f(t) = \int_{-\infty}^{\infty} F(\omega) e^{2\pi i t \omega} \, d\omega$$

time $\rightarrow$ **frequency** **frequency** $\rightarrow$ **time**

- Signals encountered in practice are usually real $f(t) \in \mathbb{R}$ (e.g. audio)
  - *Fourier Transform* maps a real signal into complex space $F(\omega) \in \mathbb{C}$.
- **Convolution theorem**: convolution in one domain (e.g., time domain) equals point-wise multiplication in the other domain (e.g., frequency domain).

Discrete Fourier Transform (DFT)

- Discrete FT maps a discrete-time signal to the discrete-frequency domain
- Consider a function of period N:
  - Given a sequence $\{x_n\}_{n=0}^{N-1}$ the **Discrete Fourier Transform** $\mathcal{F}$ is defined by:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn}$$

- The inverse transform is defined by:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{i\frac{2\pi}{N}kn}$$

- **In practice** - given a vector `x[0:N-1]` we can take DFT via `np.fft.rfft(x)`.

Discrete Fourier Transform (DFT) - Power Spectrum

- The DFT of a sequence $\{x_n\}$ produces the spectrum $\{X_k\}_{k=1}^{K}$ of the signal
  - Each X_k is a complex number is defined by magnitude $|X_k|$ and phase ϕ_k

$$X_k = |X_k| e^{i\phi_k}$$

- In speech processing we only look at the spectral magnitudes and ignore phase.
  - It is more information to consider the *power spectrum* $|X_k|^2$
- **Power Spectrum** - power of each frequency component $|X_k|^2$ of our signal.

Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP
  - Continuous and Discrete Fourier Transforms
  - **Sampling and Aliasing**
  - Windowing and Short-Time Fourier Transform (STFT)
- Spectral representations of speech
- Reconstructing speech from spectral representations

Theorem

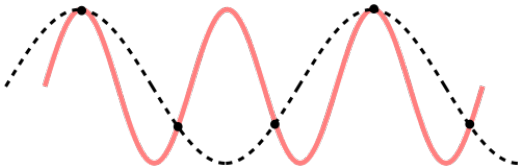
If a function $x(t)$ contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2B}$ seconds apart.

Example: If signal contains frequency 100Hz, the sampling rate for this signal needs to be 200Hz at least.

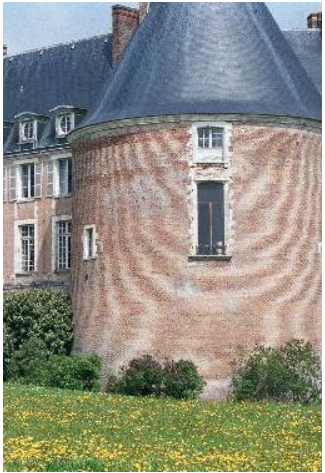
But what happens if we take sampling rate equal to 150Hz?

Aliasing

Aliasing – effect that causes different signals to become indistinguishable (or aliases of one another) when sampled.



Solution: low-pass filtering or sampling rate increase.



Sampling Rate

Recall - when digitising speech, we sample signal at a *sample rate*

- Choice of sample rate depends on which frequencies you want to retain!

Human speech contains 20Hz - 20kHz frequencies

- Must sample *at least* at 40kHz in order to retain all informationn
- In practice, we need a transition band where low-pass filter works
- Common sample rate - 44.1kHz (CD, retro hi-fi) or 48kHz (modern hi-fi)

Other common sampling rates:

- 8kHz - telephone narrowband - adequate for human speech but without sibilance
- 16kHz - Wideband extension over standard telephone narrowband.
- 22.05kHz - MPEG audio, AM radio, digitization of old records (pre-vinyl).
- 96kHz - DVD-Audio, Blu-ray Disc audio tracks, professional audio editing

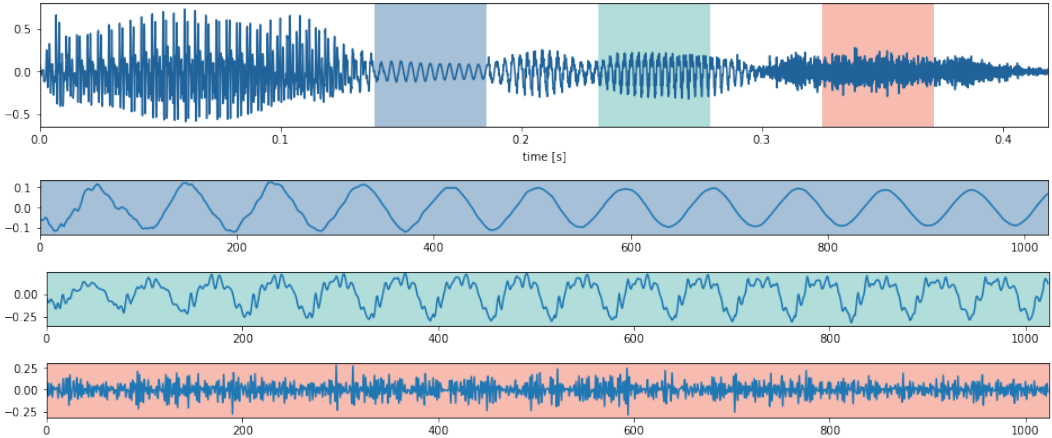
Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP
  - Continuous and Discrete Fourier Transforms
  - Sampling and Aliasing
  - **Windowing and Short-Time Fourier Transform (STFT)**
- Spectral representations of speech
- Reconstructing speech from spectral representations

Audio signals

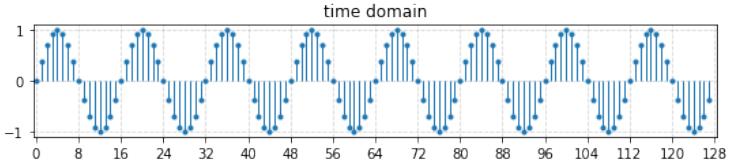
- We could take our digitized audio signal and convert into the frequency domain
  - Produces spectral decomposition of out entire speech signal.
- However, speech varies and is non-stationary
  - Whole-signal DFT does not helps us to retrieve information about local structures.
- Need to examine local spectrum to see how it changes!

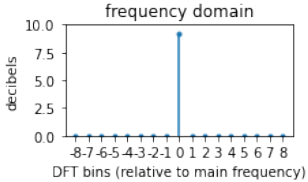
Audio signals

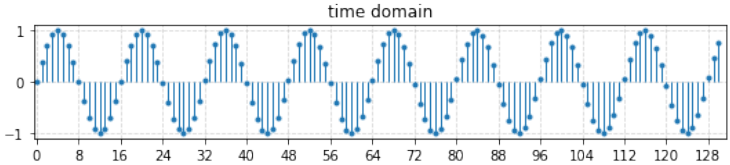


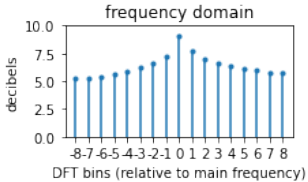
Partition and than DFT? Is it really that simple? **Not really.**

DFT from $\sin(x)$









- DFT from part of the signal may cause **spectral leakage**.
- It comes out in form of spreading out values from one bin (representing frequency of signal) to neighboring.
- Periodic extension of the function becomes the other function with another

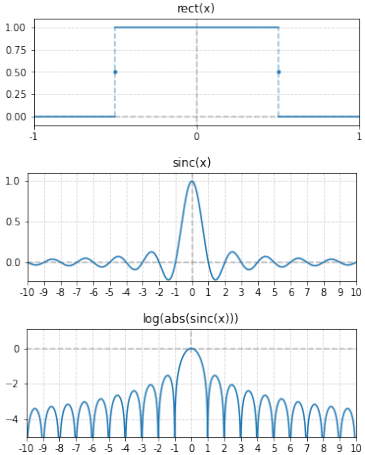
Let's take a look on a simple **rectangular function**:

$$\Pi(x) = \begin{cases} 0, & \text{if} |t| > \frac{1}{2} \\ \frac{1}{2}, & \text{if } |t| = \frac{1}{2} \\ 1, & \text{if} |t| < \frac{1}{2} \end{cases}$$

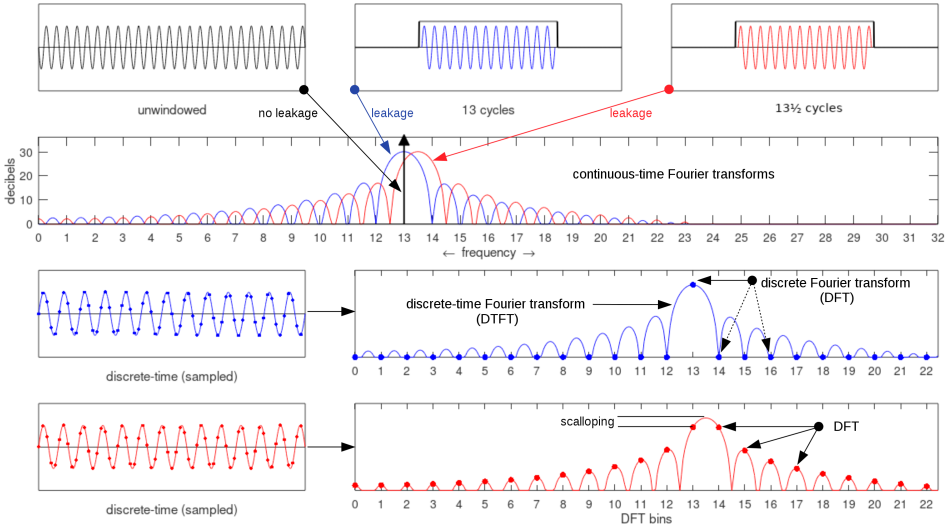
FT of this function is equal **sine cardinal** function:

$$\int_{-\infty}^{\infty} \Pi(x) e^{-j2\pi fx}\, dx = \frac{\sin(\pi f)}{\pi f} = sinc(\pi f)$$

The sinc function is widely used in DSP, and here is why:



Spectral leakage



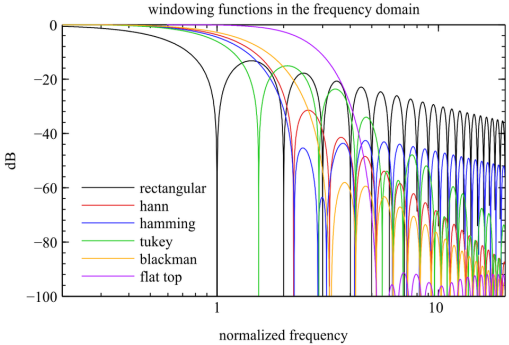
Windowing Functions (Hann Window)

Mostly used window function is **Hann** window function:

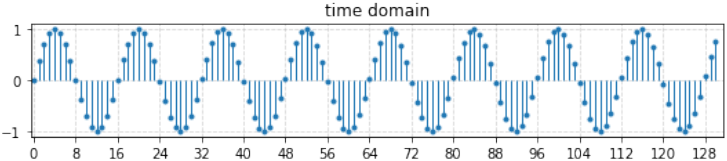
$$w_{\text{hann}}(x) = \begin{cases} \frac{1}{2}(1+\cos(\frac{2\pi n}{M})), & \text{if} |x| < \frac{M}{2} \\ 0, & \text{if} |x| > \frac{M}{2} \end{cases}$$

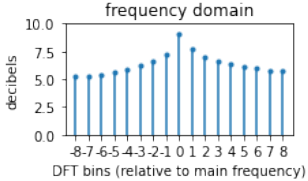
$$\frac{1}{2}(1+\cos(\frac{2\pi n}{M})),\ n \in [-\frac{M}{2}, \frac{M}{2}]$$

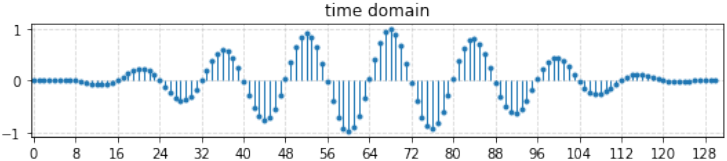
General criteria of selection: minimization of spectral leakage.

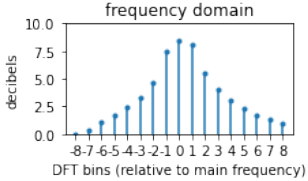


Spectral leakage with Hann



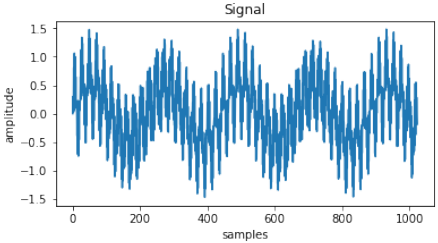


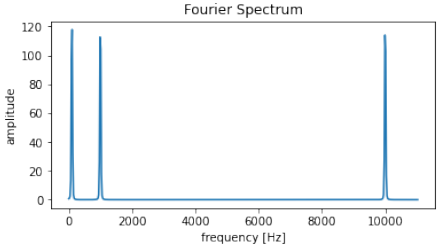


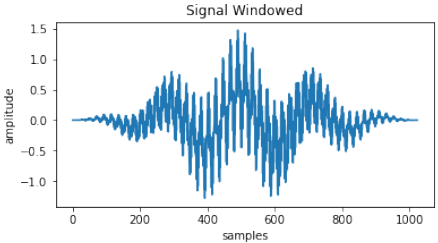


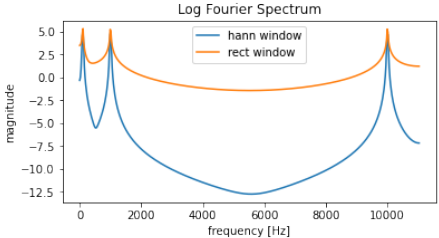
Spectrum Example

Three Sinusoidal Signals (100Hz, 1kHz, 10kHz)



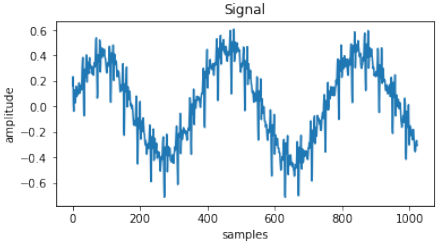


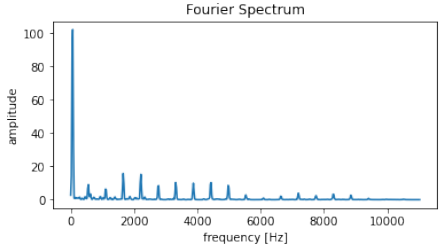


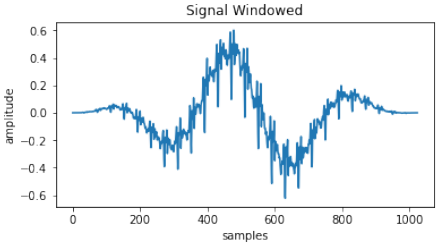


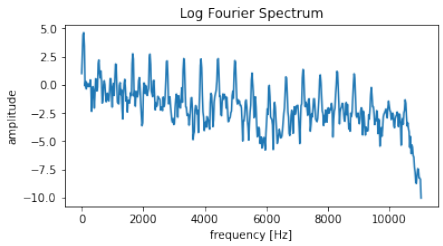
Spectrum Example

Saxophone



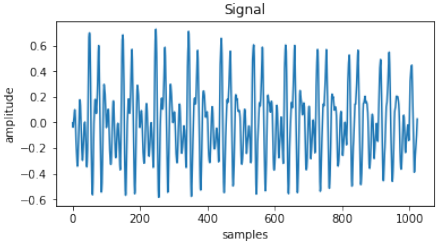


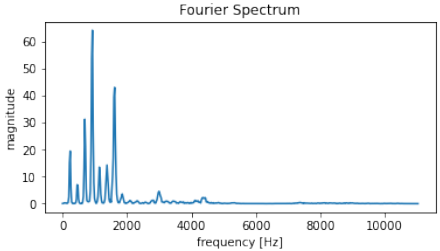


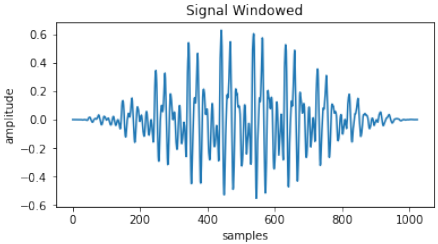


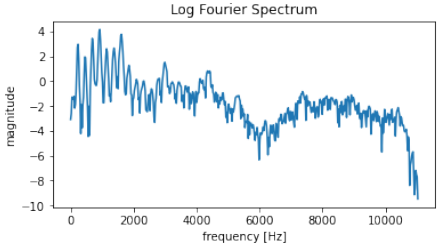
Spectrum Example

Human Voice









Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP:
- **Spectral representations of speech**
  - Spectrograms
  - The Mel-Scale and Mel-Spectrograms
  - Cepstrum and MFCCs
- Reconstructing speech from spectral representations

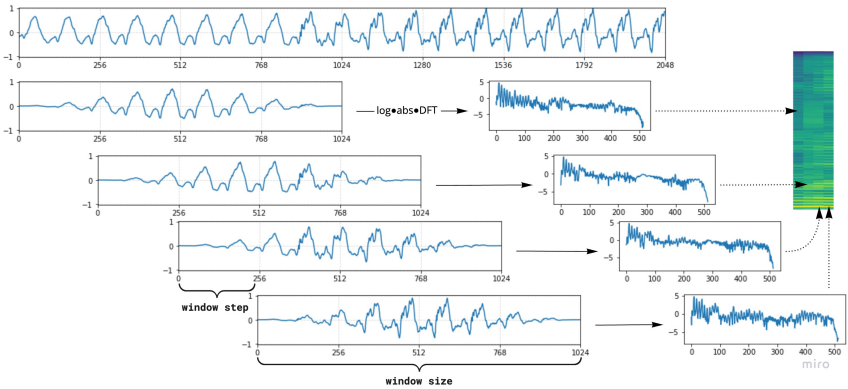
Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP:
- Spectral representations of speech
  - **Spectrograms**
  - The Mel-Scale and Mel-Spectrograms
  - Cepstrum and MFCCs
- Reconstructing speech from spectral representations

Limitations of waveforms

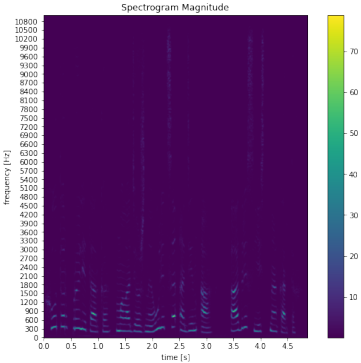
- Recall limitations of waveforms
  - Extreme dimensionality
  - Periodic structure are different time scales
  - Inaudible variations (time shift, amplitude scaling) produce different audio signals.
- We want a representations which is:
  - Compact in time
  - Disentangles structure at different time scales (frequencies)
  - Robust to inaudible variables
- Use the power spectrogram!

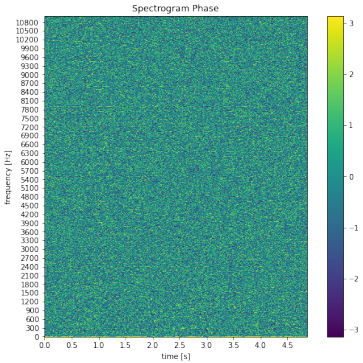
Spectrograms



- Consider stacking STFTs of a window sliding over the signal
  - Each window is called an acoustic frame

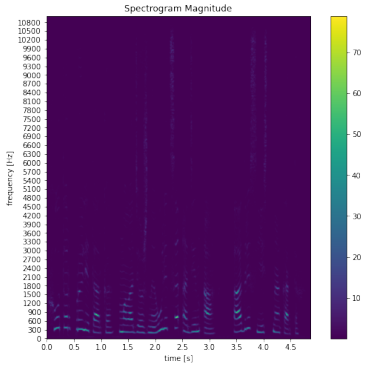
Spectrogram magnitude and phase

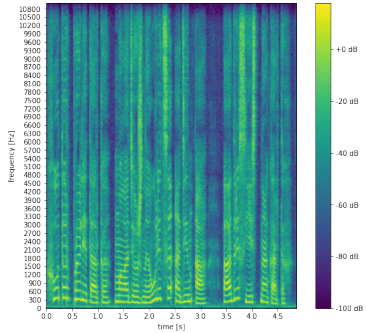




- Retain magnitude, discard phase information.

Power (dB) Spectrogram of human voice

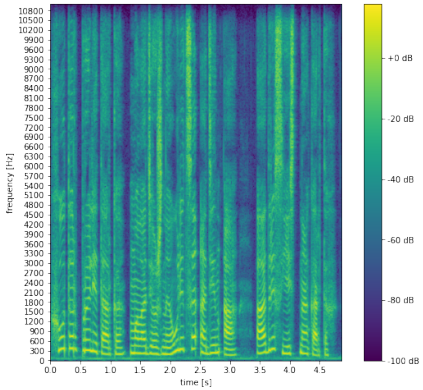




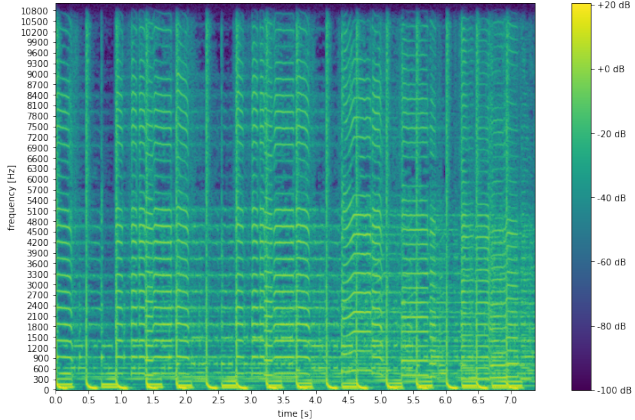
- Spectrograms usually represented as energies of corresponding frequencies
  Logarithmic scale (decibels) emphasize high frequency structure

Power (dB) Spectrogram of human voice

- $0 - 4kHz$ frequencies covers most of the speech signal
  - $8kHz$ (telephone codec) sampling rate is enough for intelligible speech
  - ... but sounds bad.
- Voiced sounds have non-constant fundamental frequency (intonation) (▸ Audio)

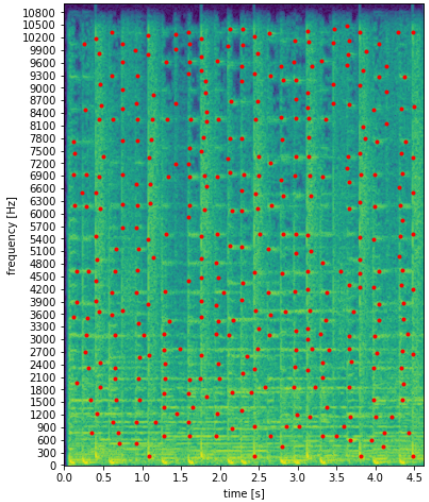


STFT Example (Saxophone)



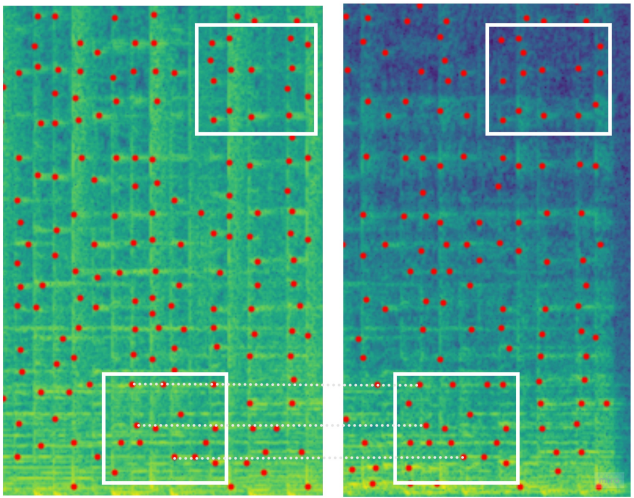
- High harmonics are more visible (comparing to speech signal)
  - Harmonics are mostly constant in time (notes) (Audio)

Songs Fingerprinting



- In "play $\rightarrow$ record" transform frequency corresponding amplitudes are changed (smoothly) and some background noise is added.
- $\Rightarrow$ most of local maximums will be preserved and their relative location is constant.
- ▸ Song played from speaker
- ▸ Recording from smart-phone's microphone

Song Fingerprinting



Fingerprinting algorithm:

- obtain local maximums,
- for each point: we iterate over all pairs with this point in given quadrant,
- for each pair: frequency and time differences are hashed,
- $\Rightarrow$ each song is represented as a set of hashes;

How to trick this algorithm?

Spectrograms vs Waveforms

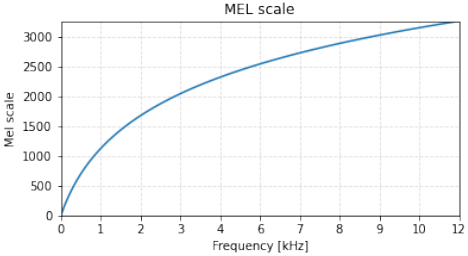
- Spectrograms overcome some limitations of waveforms
  - Compresses time axis by the size of window-shift (x256 or x512 time-steps)
  - Expresses structure at different frequences as an additional axis
  - More robust to minor time shift and amplitude scaling
- However, we still are very large in frequency dimension
  - Human hearing 20Hz - 20kHz + transition band.
- Make use of Psychoacoustics to compress frequency axis!

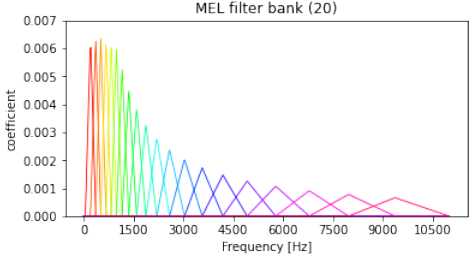
Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP:
- Spectral representations of speech
  - Spectrograms
  - **The Mel-Scale and Mel-Spectrograms**
  - Cepstrum and MFCCs
- Reconstructing speech from spectral representations

Mel Scale and Filter Bank

- Human auditory perception which is logarithmic
  - We are better at discriminating between low frequencies
- The Mel scale is an approximate scale of human frequency perception
  - Pays more attention to lower part of spectrum.
- Frequency-to-mel transform is defined as $m = 2595 \log_{10}(1 + \frac{f}{700})$

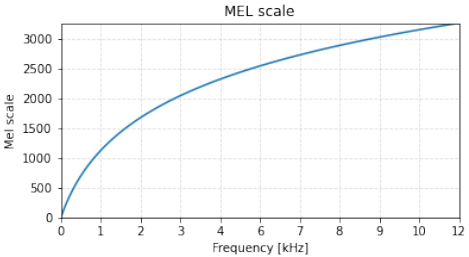


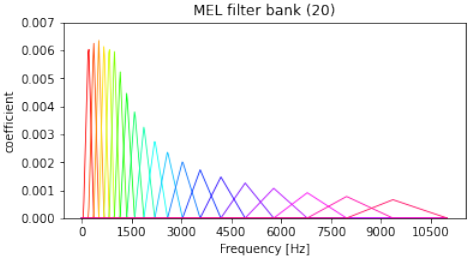


Mel Scale and Filter Bank

- Mel spectrum is produced by linear projection with Mel filter bank.
  - For a filter bank $\boldsymbol{F}$ and spectrum $\boldsymbol{s}$, Mel spectrum $\boldsymbol{m}$ is produced as $\boldsymbol{m} = \boldsymbol{F}\boldsymbol{s}$
  - Given a spectrogram $\boldsymbol{S}$, the Mel-Spectrogram $\boldsymbol{M}$ is produced as $\boldsymbol{M} = \boldsymbol{F}\boldsymbol{S}$
- The filter bank is defined by frequency range and number of filters (usually 80).
- The spectrogram can be approximately reconstructed via Pseudo-Inverse:

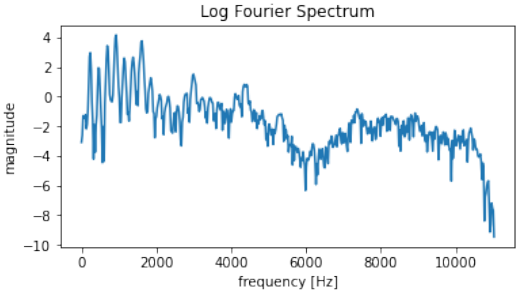
$$\hat{\boldsymbol{S}} = (\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F})^{-1}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{M}$$

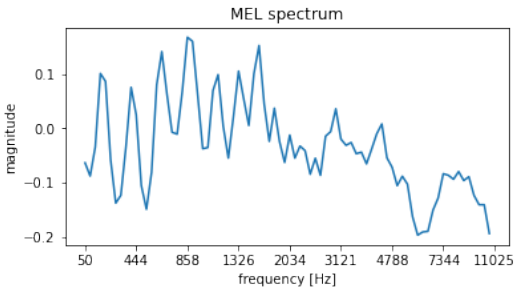




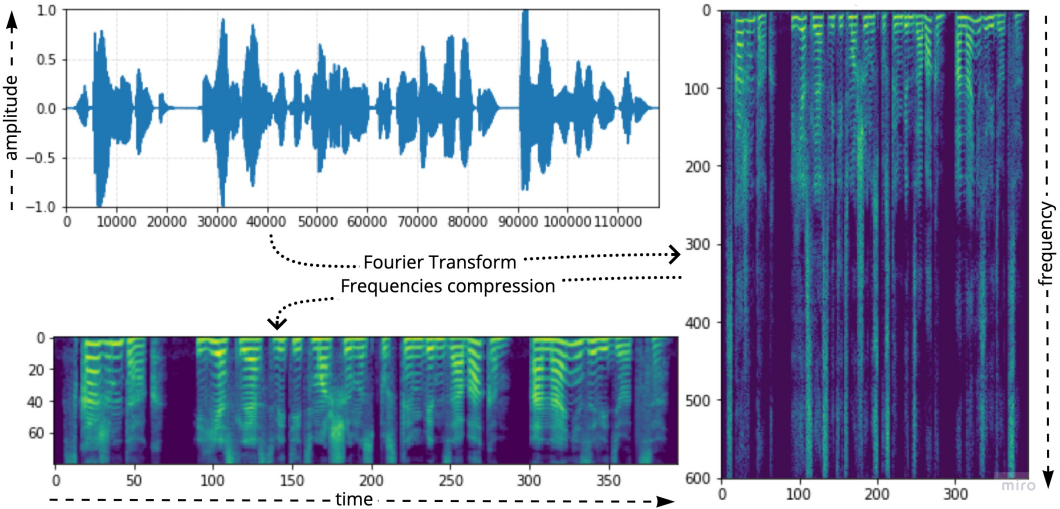
Mel Spectrum Example

- We have now greatly compressed the Spectrogram along the frequency axis
  - Overall structure is retained, but high-frequency fidelity is lower

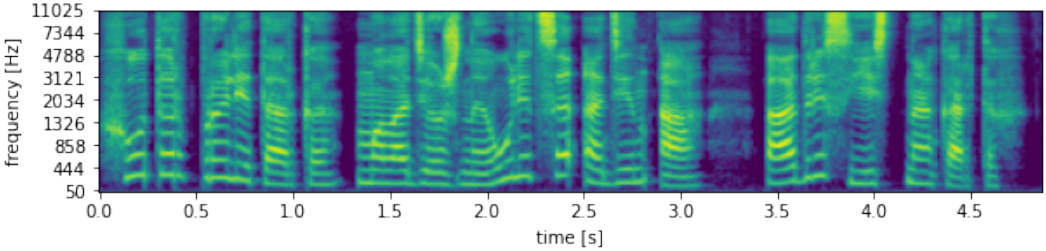




Time and Frequency Compression



Mel Spectrogram

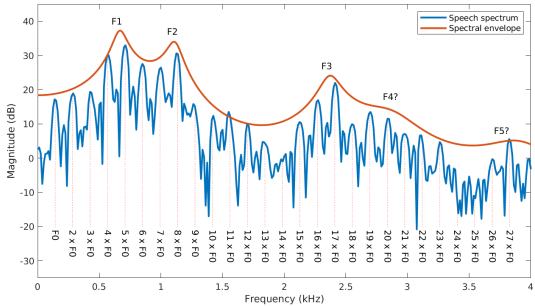


- Mel-Spectrogram → compact, efficient, but lossy representation of speech
  - Compression in time via STFT, compression in Frequency via Mel-Scale
  - Lossy in time (phase) and frequency (linear projection)
- Can be reconstructed via Griffin-Lim Algorithm - used in speech synthesis! ▸ Audio
- We can easily detect vowels sounds on Mel-Spectrogram
  - Phoneme classification needs deep expertise in phonetics (formants recognition).

Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP:
- Spectral representations of speech
  - Spectrograms
  - The Mel-Scale and Mel-Spectrograms
  - **Cepstrum and MFCCs**
- Reconstructing speech from spectral representations

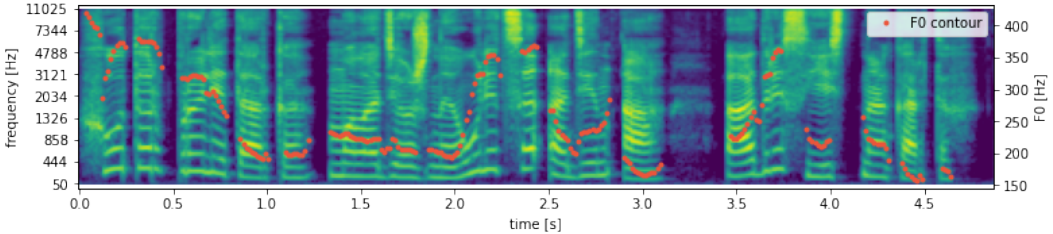
Fundamental frequency (F_0) and formants



- F_0 – *fundamental frequency*, $k \times F_0$ – *harmonics*
- Peaks on envelope curve (**red**) – formants
- *Pitch* is perceptual value, F_0 is physical (correlated)
- Changing (in time) pitch forms voice intonation
- How can we find F_0?

F0 over time

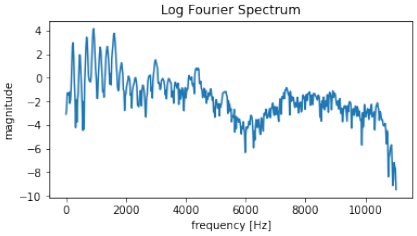
Demonstration of PRAAT algorithm for F0 extraction (one more beautiful image):



- cepstrum analysis algorithms
- signal auto-correlation algorithms
- hybrid algorithms
- neural networks

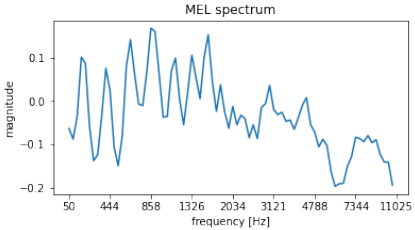
Cepstrum

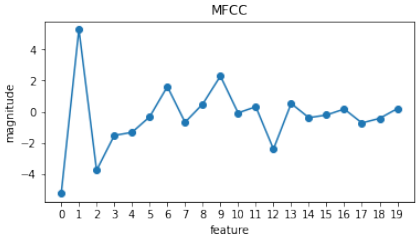
- Fourier Spectrum of voice has periodic structure (period is equal to F_0)
- $\Rightarrow$ let's get spectrum from spectrum (Discrete Cosine Transform) – brings us to the **Cepstrum**
- Two time-frequency transforms leads us back to time-domain (not signal)
- Low quefrencies encode information about formants (non-trivial to extract)
- Peak in the cepstrum should be located at $\frac{1}{F_0}$



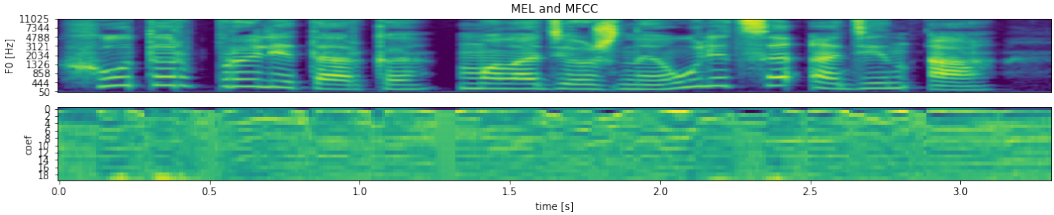
Mel-Frequency Cepstral Coefficients (MFCCs)

- Taking the **Discrete Cosine Transform (DCT)** of mel spectrum, we obtain the representation known as **mel-frequency cepstral coefficients (MFCCs)**.
- Mel-scale transform decorrelates F_0 harmonics.
- Phone definition is based on macro-shapes in the spectrum (preserves that information)
- MFCC separates the impact of **source** and **filter** in a speech signal.





MFCC over time



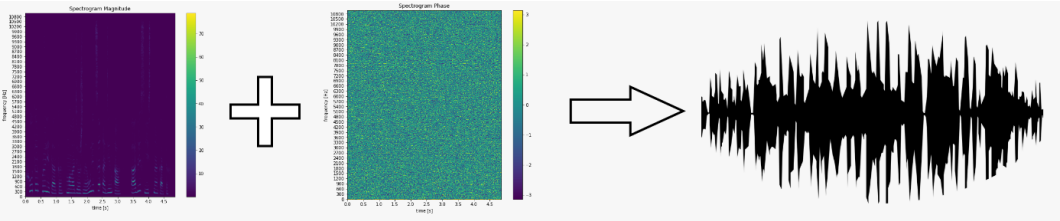
Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP
- Spectral representations of speech
- **Reconstructing speech from spectral representations**
  - Inverse Short-time Fourier Transform
  - Griffin-Lim Algorithm for phase reconstruction

Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP
- Spectral representations of speech
- Reconstructing speech from spectral representations
  - **Inverse Short-time Fourier Transform**
  - Griffin-Lim Algorithm for phase reconstruction

inverse Short-time Fourier Transform

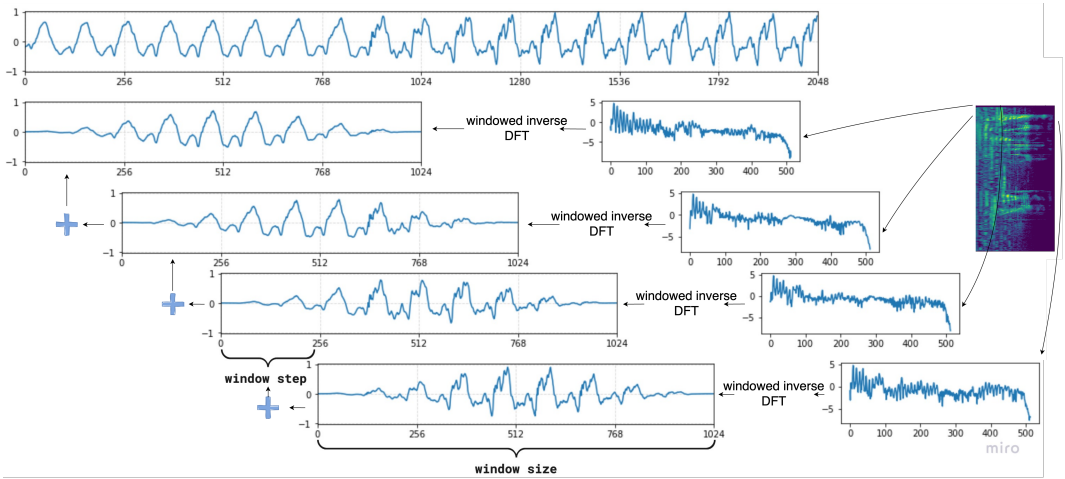


- Let's examine reconstructing audio from **magnitude** and **phase** of the spectrum
  - Remember to convert from decibels to linear power and from power to magnitude!

Overlap and Add inverse STFT

- Recall - each frame is an STFT of a hanh-windowed signal shifted by a hop-length
  - We can individually take the inverse DFT of each frame using `np.fft.irfft()`
- How do we combine all the restored overlapping signal windows together?
  - Overlap, window and add!

Overlap and Add inverse STFT



Lecture Structure

- Representing sound as discrete digital signal
- Mathematical Foundations of DSP
- Spectral representations of speech
- Reconstructing speech from spectral representations
  - Inverse Short-time Fourier Transform
  - **Griffin-Lim Algorithm for phase reconstruction**

Phase reconstruction

- Ok, but how do we get the phase information? We threw it away
  - Use the **Griffin-Lim Phase reconstruction algorithm**!
- STFT produced by **overlapping** windows – **redundant representation** of signal.
  - Iteratively take STFT and iSTFT, enforcing magnitude consistency
  - Eventually will recover a consistent phase.

Input: $\rho = |STFT(f)|$ – absolute of STFT f
Result: $\tilde{f} \approx f$ – reconstructed signal f
begin
  $\phi \sim \mathbb{U}(0, 2\pi)$ – random initialization
  $\tilde{f} = ISTFT(\rho * e^{i\phi})$ – first approximation
  for *number of iterations* **do**
    $\phi = \arg STFT(\tilde{f})$ – calculate angle
    $\tilde{f} = ISTFT(\rho * e^{i\phi})$ – new approximation
  end
end

Algorithm 1: Griffin-Lim

Griffin-Lim Algorithm

- GLA is a simple and very useful procedure – used as a simple vocoder in TTS
  - Simple, converges to decent audio in a few interation.
- However, GLA has limitations:
  - GLA is not robust even to float-arithmetic (metallic sound effect): Audio
  - GLA is not robust to noise – Audio reconstruction of slightly blurred spectrogram
- If our **synthesized** spectrogram is not perfect, GLA will generate weird artifacts.
  - GLA good starting point, but need better vocoders (TTS lectures).

Lecture Summary

- Covered how waveforms are recorded:
  - Sample at discrete time intervals and record quantized amplitude value
- Had a brief recap of DSP
  - Fourier Transform $\rightarrow$ Discrete Fourier Transform $\rightarrow$ Short-Time Fourier Transform
  - Aliasing and Sample Rate
- Used the STFT to obtain a compact spectral representation of sounds
  - STFT $\rightarrow$ Spectrogram dB $\rightarrow$ Mel Spectrogram $\rightarrow$ MFCC
- Learned how to reconstruct audio from spectrograms
  - Griffin-Lim Phase reconstruction algorithm