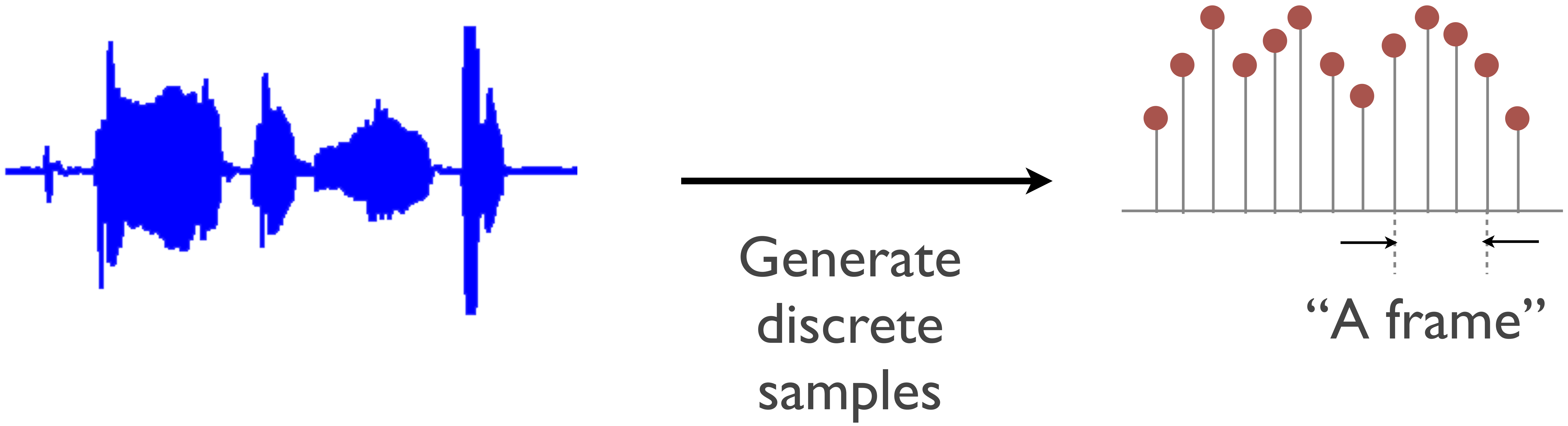
Acoustic Feature Analysis

Lecture 9c

CS 753, 2021

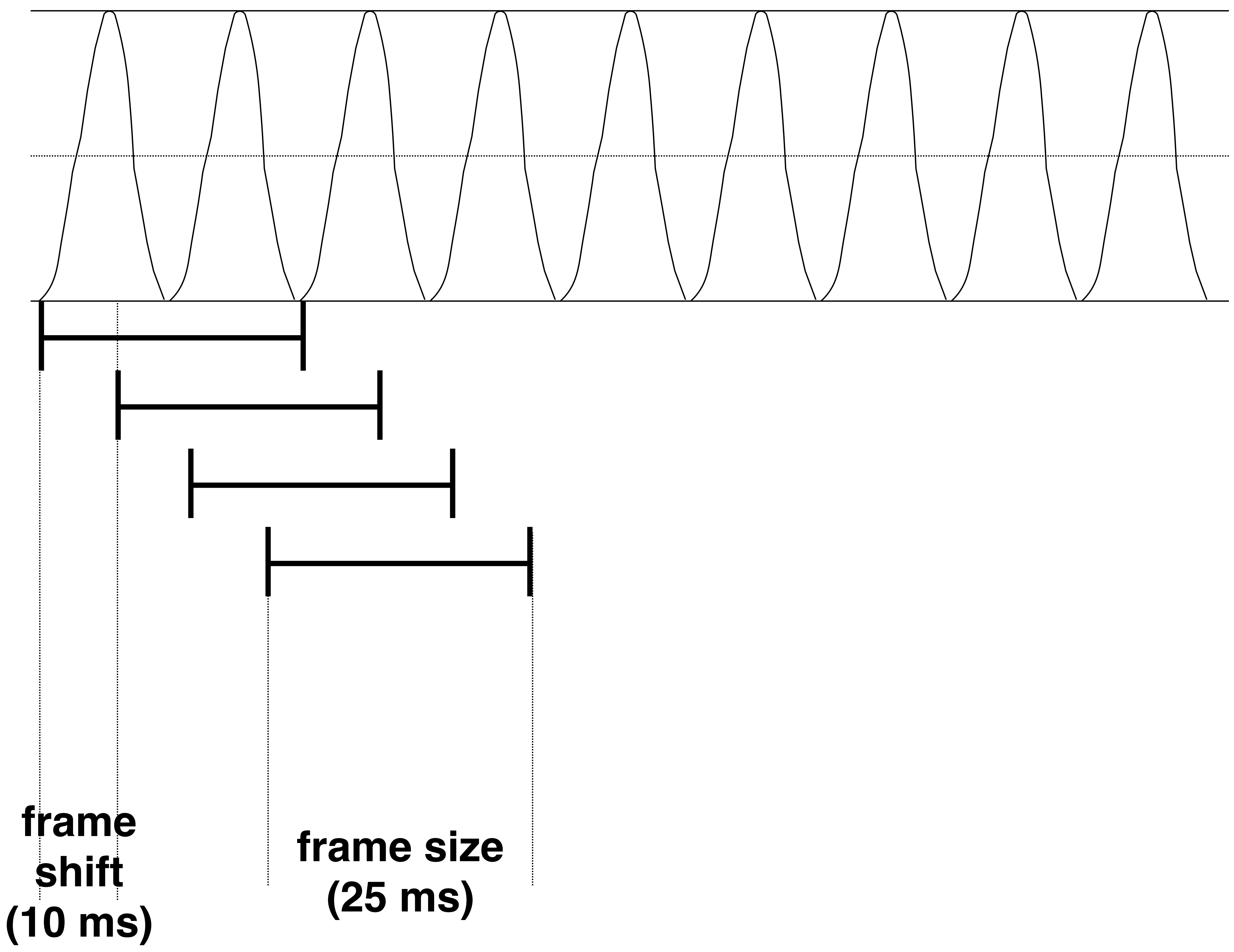
Instructor: Preethi Jyothi, IITB

Speech Signal Analysis

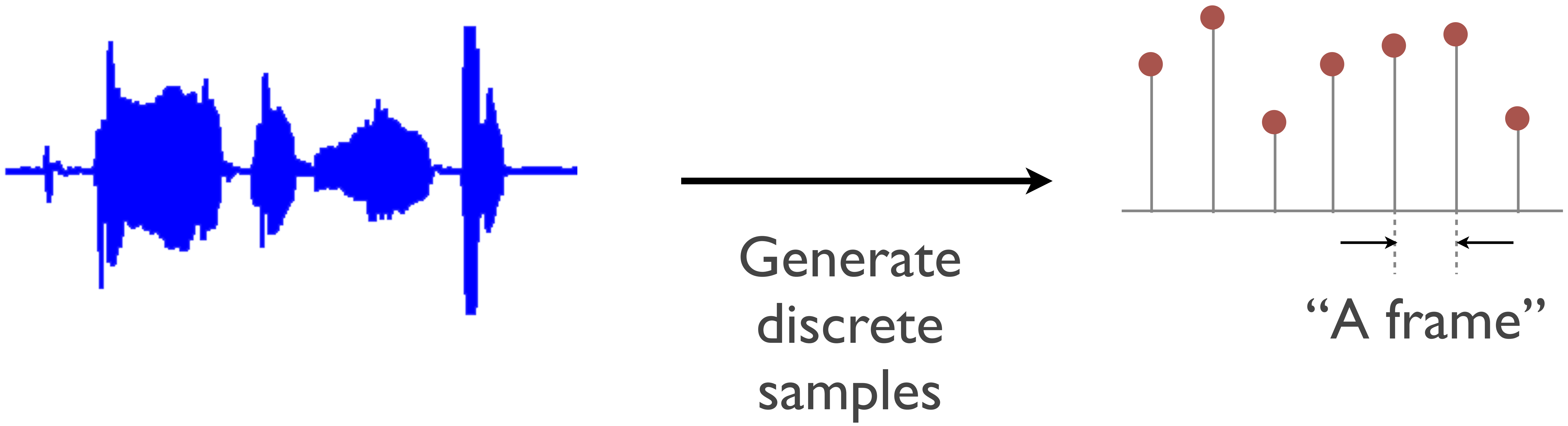


- Need to focus on short segments of speech (*speech frames*) that more or less correspond to a discrete speech unit and are stationary
- Each speech frame is typically 20-50 ms long
- Use overlapping frames with frame shift of around 10 ms

Frame-wise processing



Speech Signal Analysis



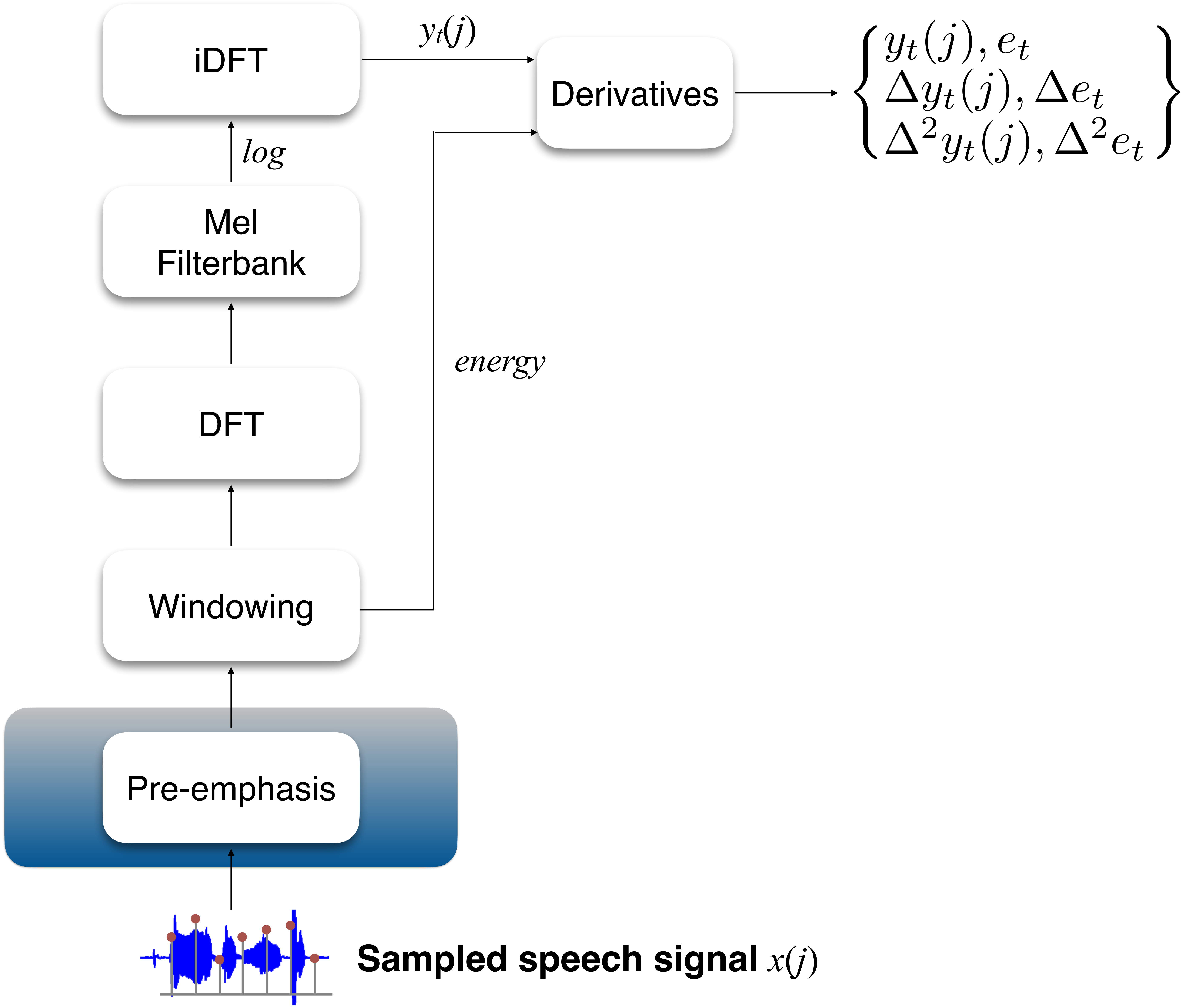
- Need to focus on short segments of speech (*speech frames*) that more or less correspond to a phoneme and are stationary
- Each speech frame is typically 20-50 ms long
- Use overlapping frames with frame shift of around 10 ms
- Generate acoustic features corresponding to each speech frame

Acoustic feature extraction for ASR

Desirable feature characteristics:

- Capture essential information about underlying phones
- Compress information into compact form
- Factor out information that's not relevant to recognition e.g. speaker-specific information such as vocal-tract length, channel characteristics, etc.
- Would be desirable to find features that can be well-modelled by known distributions (Gaussian models, for example)
- Feature widely used in ASR: Mel-frequency Cepstral Coefficients (**MFCCs**)

MFCC Extraction



Pre-emphasis

- Pre-emphasis increases the amount of energy in the high frequencies compared with lower frequencies
- Why? Because of *spectral tilt*
  - In voiced speech, signal has more energy at low frequencies
  - Attributed to the glottal source
- Boosting high frequency energy improves phone detection accuracy

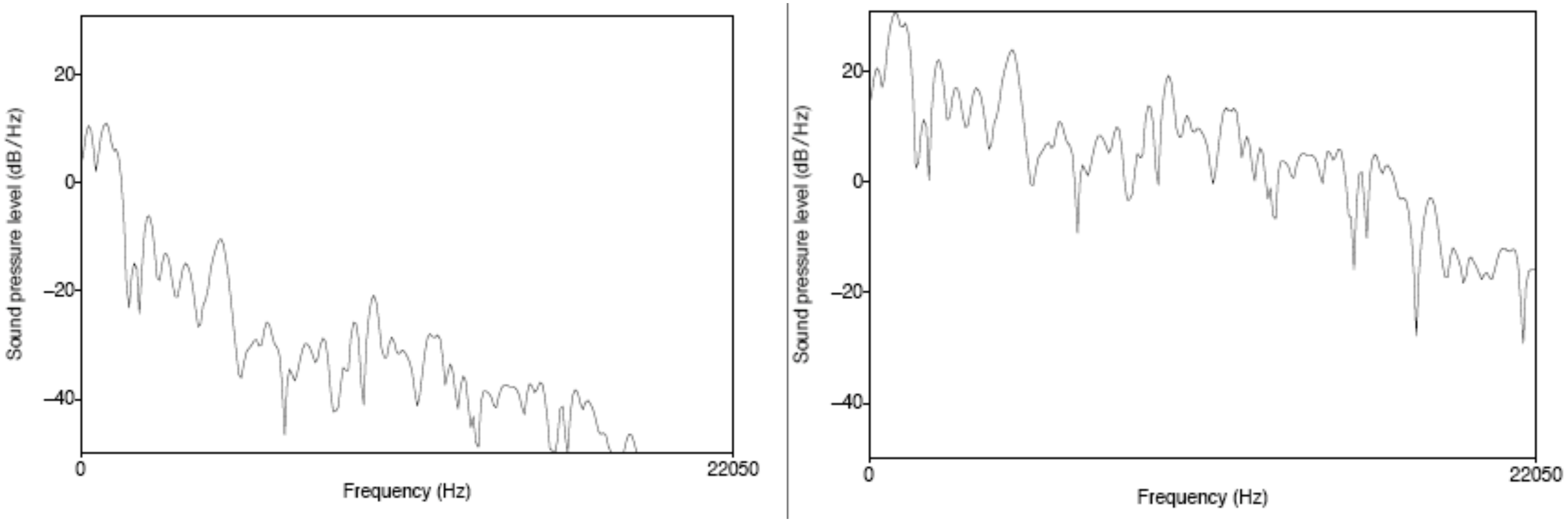
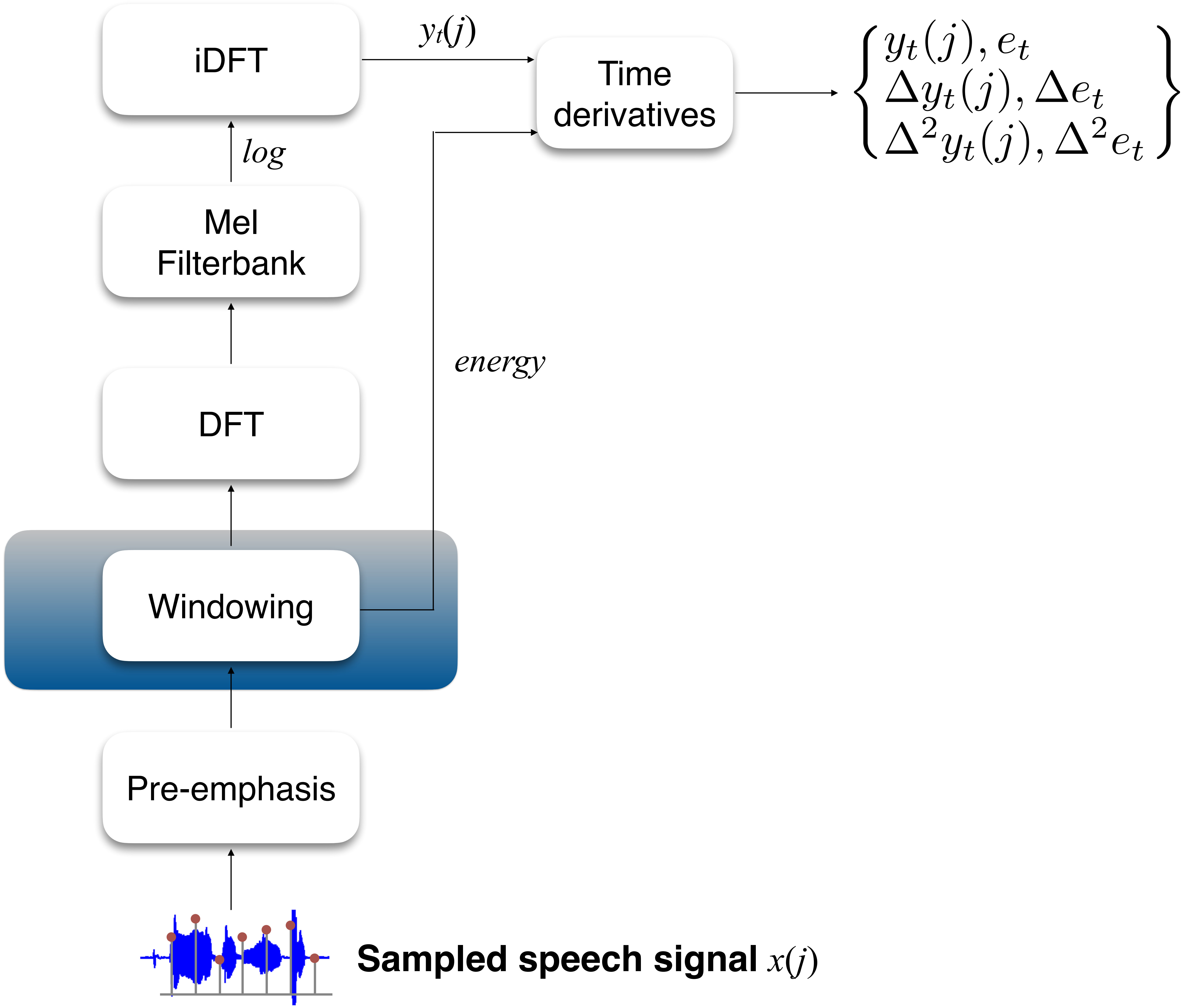


Image credit: Jurafsky & Martin, Figure 9.9

MFCC Extraction



Windowing

- Speech signal is modelled as a sequence of frames (assumption: stationary across each frame)

- Windowing: multiply the value of the signal at time n, $s[n]$ by the value of the window at time n, $w[n]$: $y[n] = w[n]s[n]$

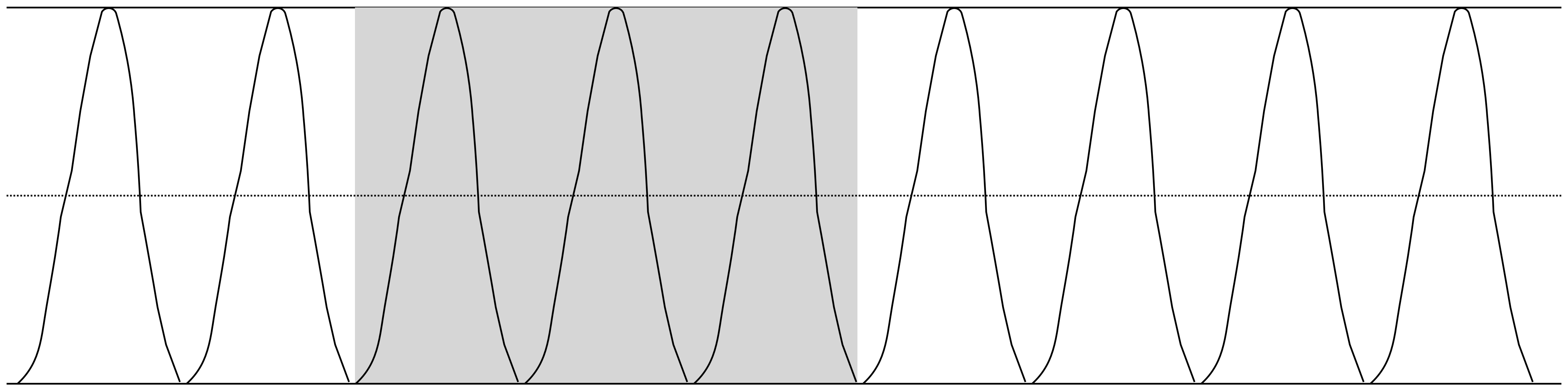
Rectangular:

$$w[n] = \begin{cases} 1 & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

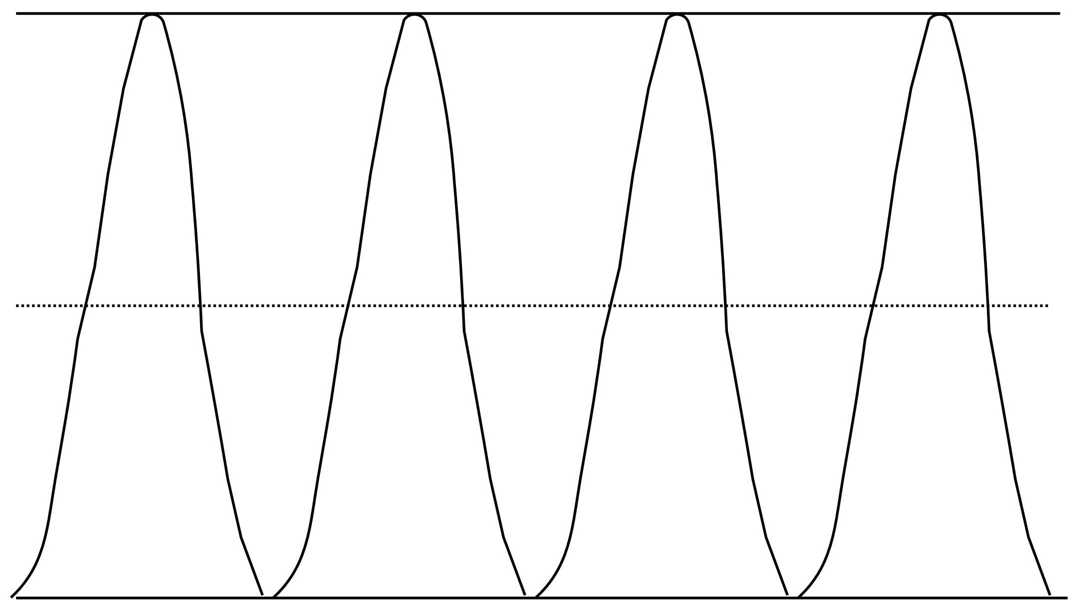
Hamming:

$$w[n] = \begin{cases} 0.54 - 0.46\cos\frac{2\pi n}{L} & 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

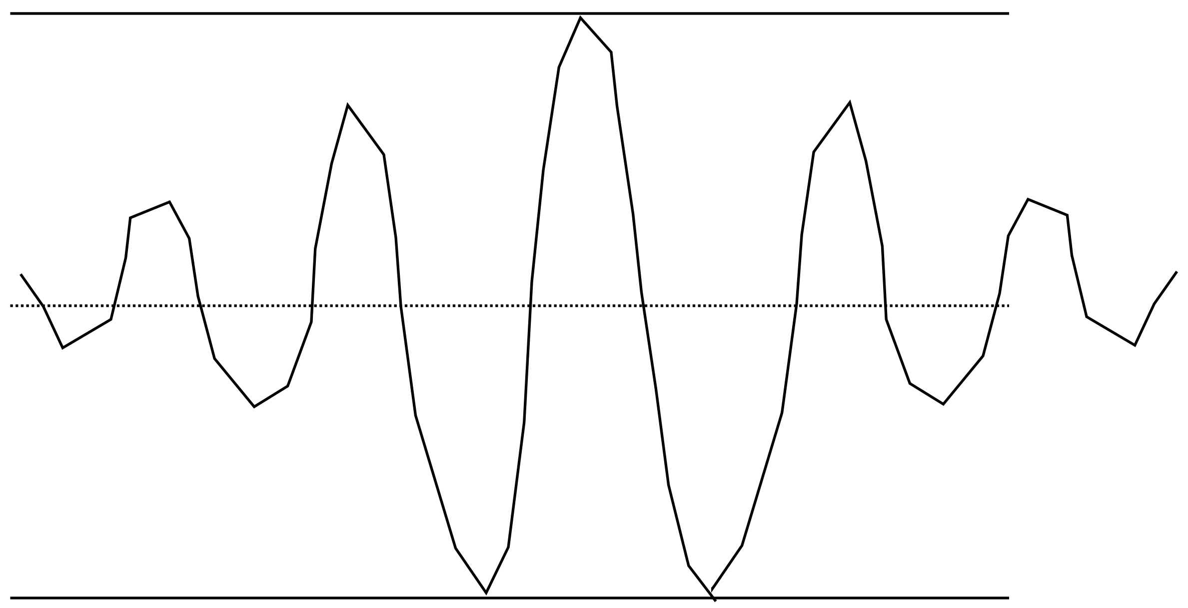
Windowing: Illustration



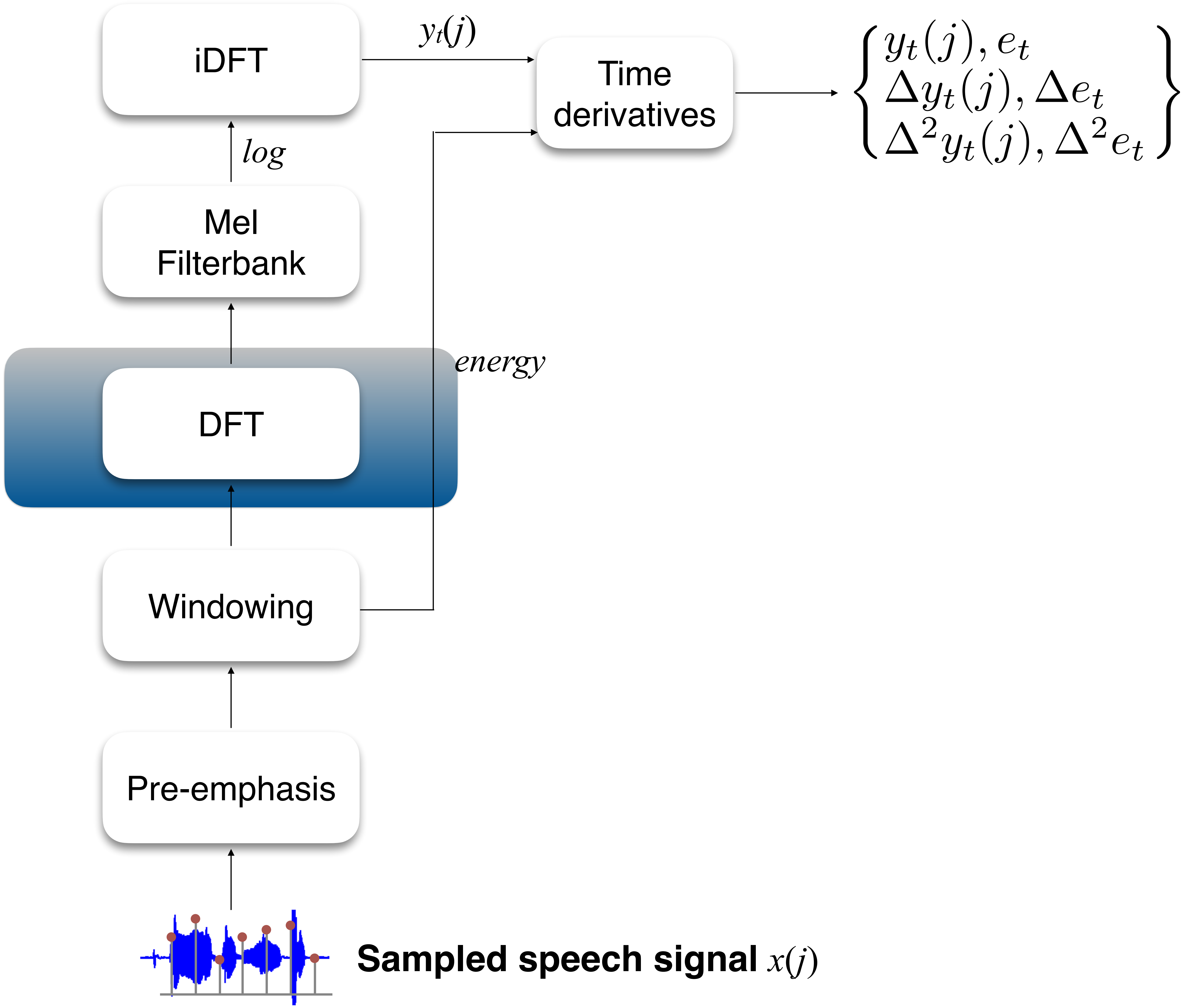
Rectangular window



Hamming window



MFCC Extraction



Discrete Fourier Transform (DFT)

Extract spectral information from the windowed signal:
Compute the DFT of the sampled signal

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Input: windowed signal $x[1],\ldots,x[n]$
Output: complex number $X[k]$ giving magnitude/phase for the kth frequency component

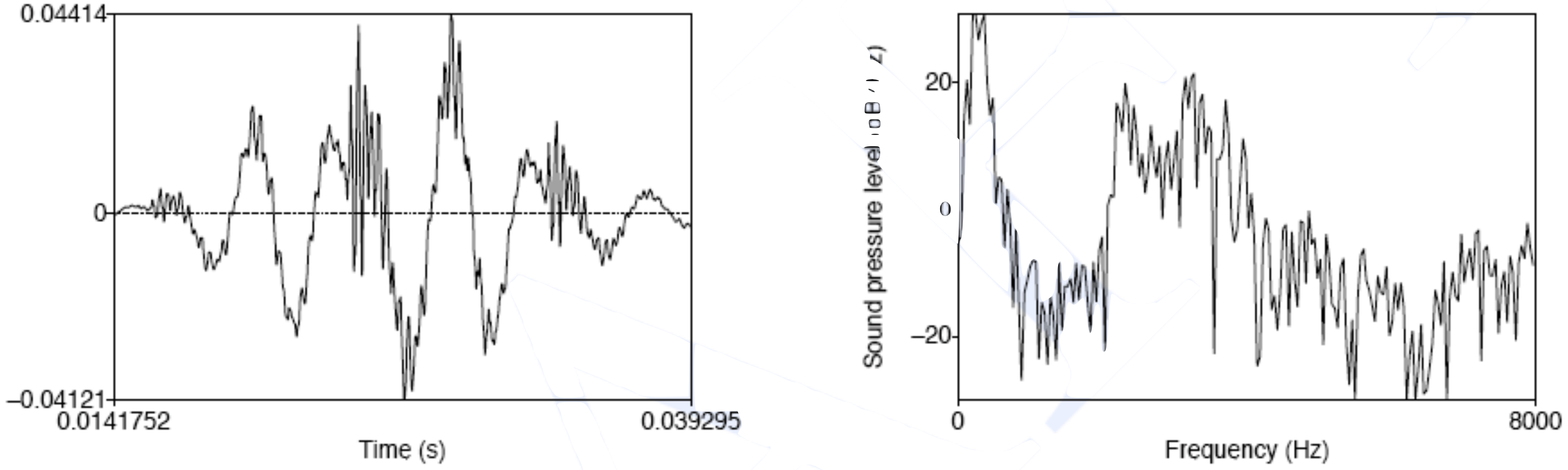
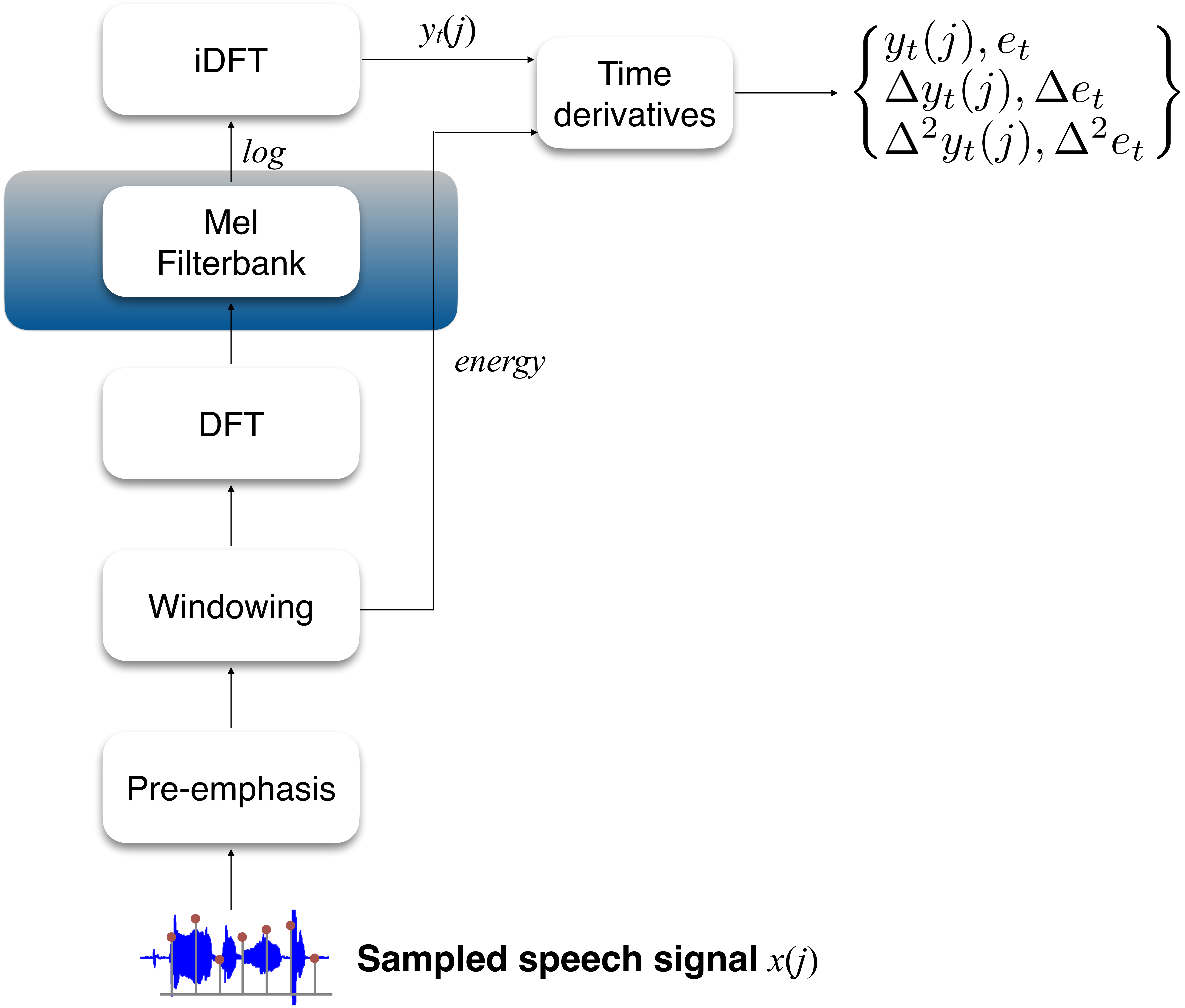


Image credit: Jurafsky & Martin, Figure 9.12

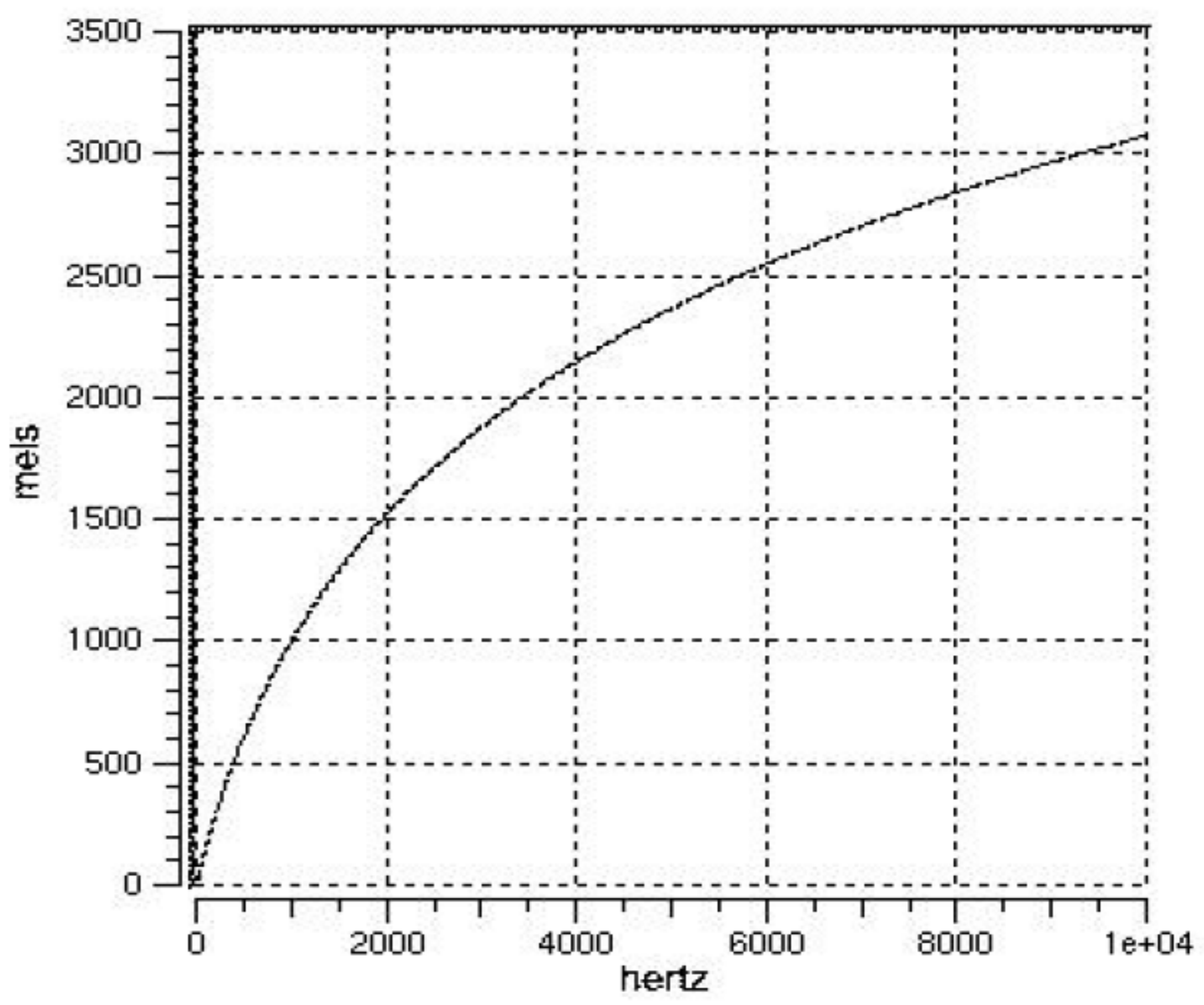
MFCC Extraction



Mel Filter Bank

- DFT gives energy at each frequency band
- However, human hearing is not sensitive at all frequencies: less sensitive at higher frequencies
- Warp the DFT output to the *mel* scale: *mel* is a unit of pitch such that sounds which are perceptually equidistant in pitch are separated by the same number of mels

Mels vs Hertz



Mel filterbank

- Mel frequency can be computed from the raw frequency f as:

$$\mathrm{mel}(f) = 1127\ln(1 + \frac{f}{700})$$

- 10 filters spaced linearly below 1kHz and remaining filters spread logarithmically above 1kHz

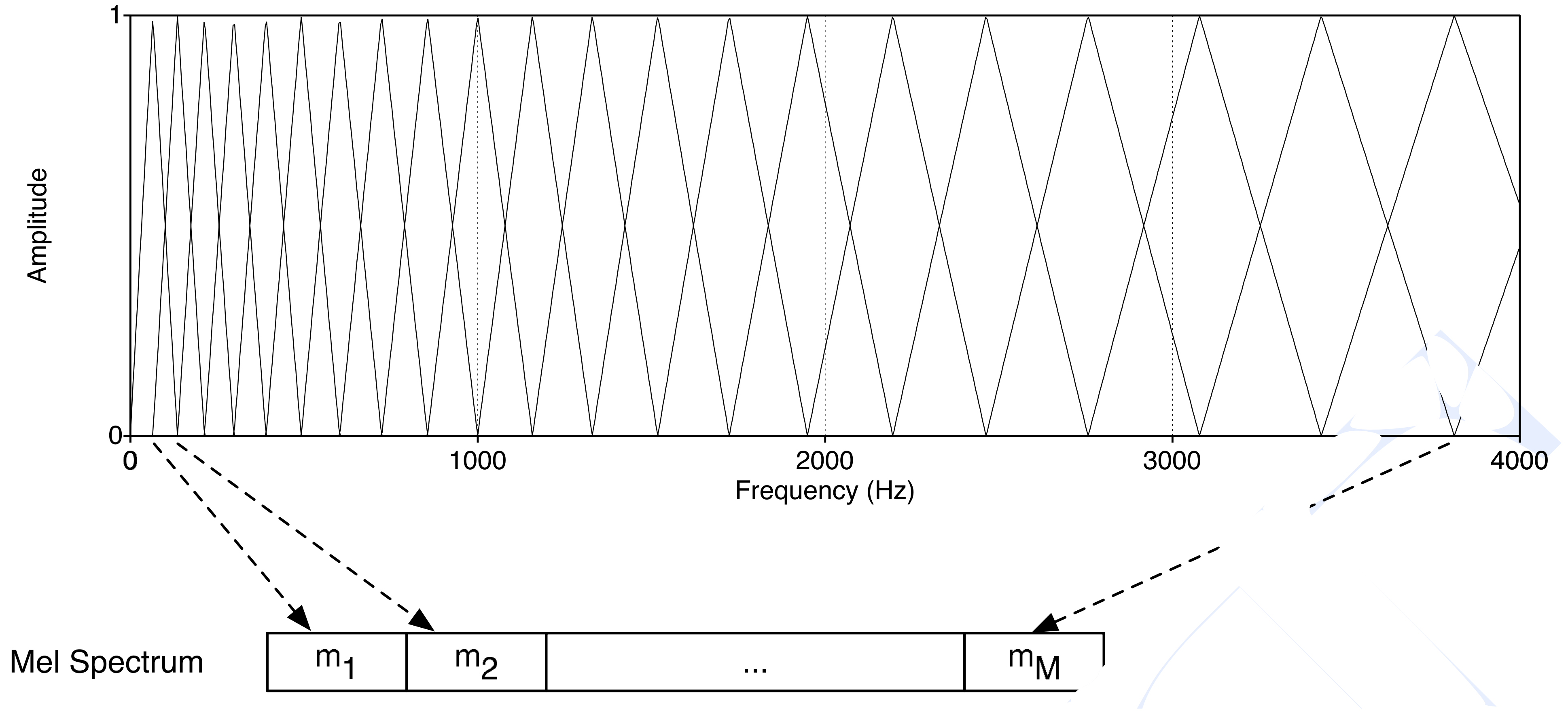


Image credit: Jurafsky & Martin, Figure 9.13

Mel filterbank inspired by speech perception

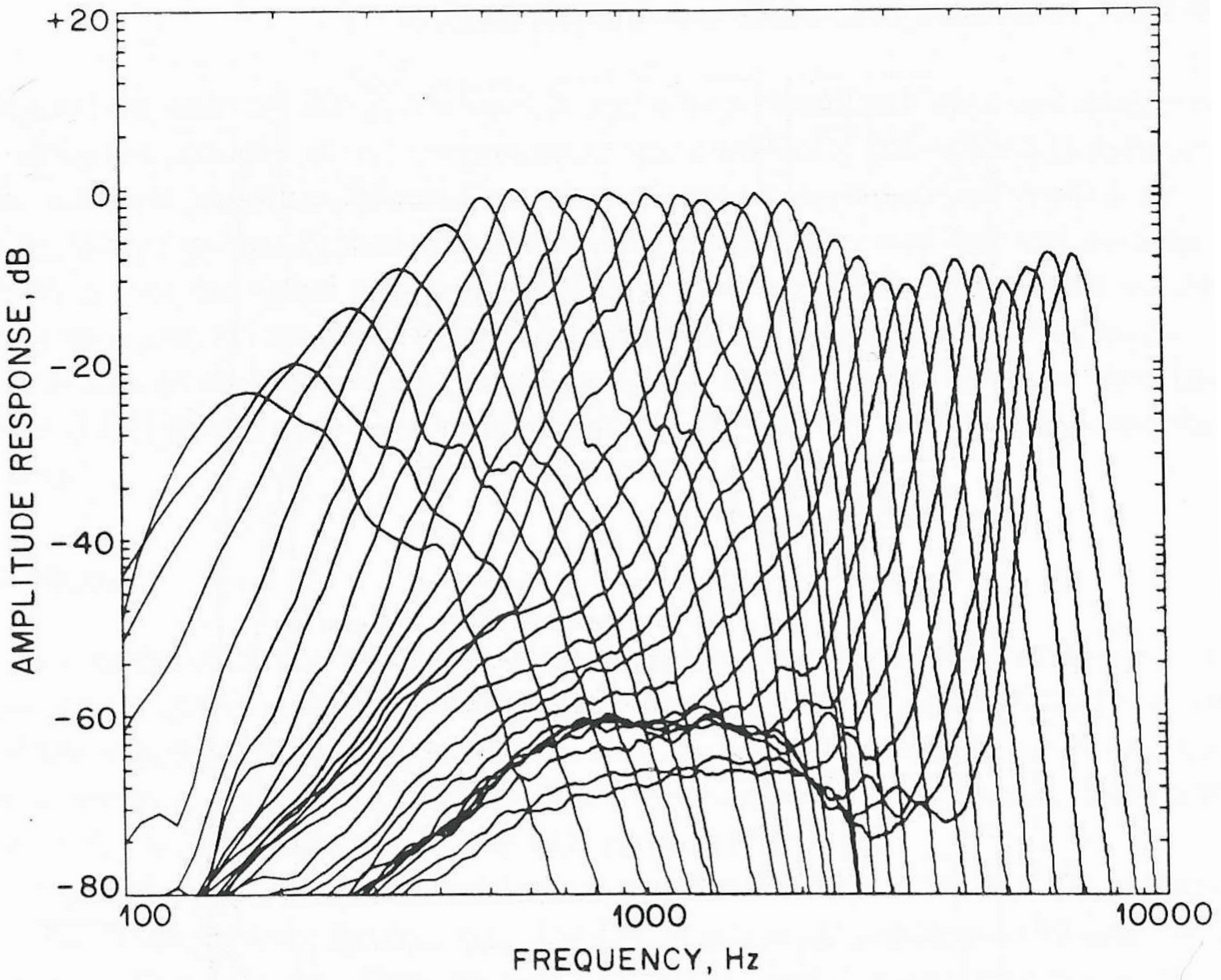


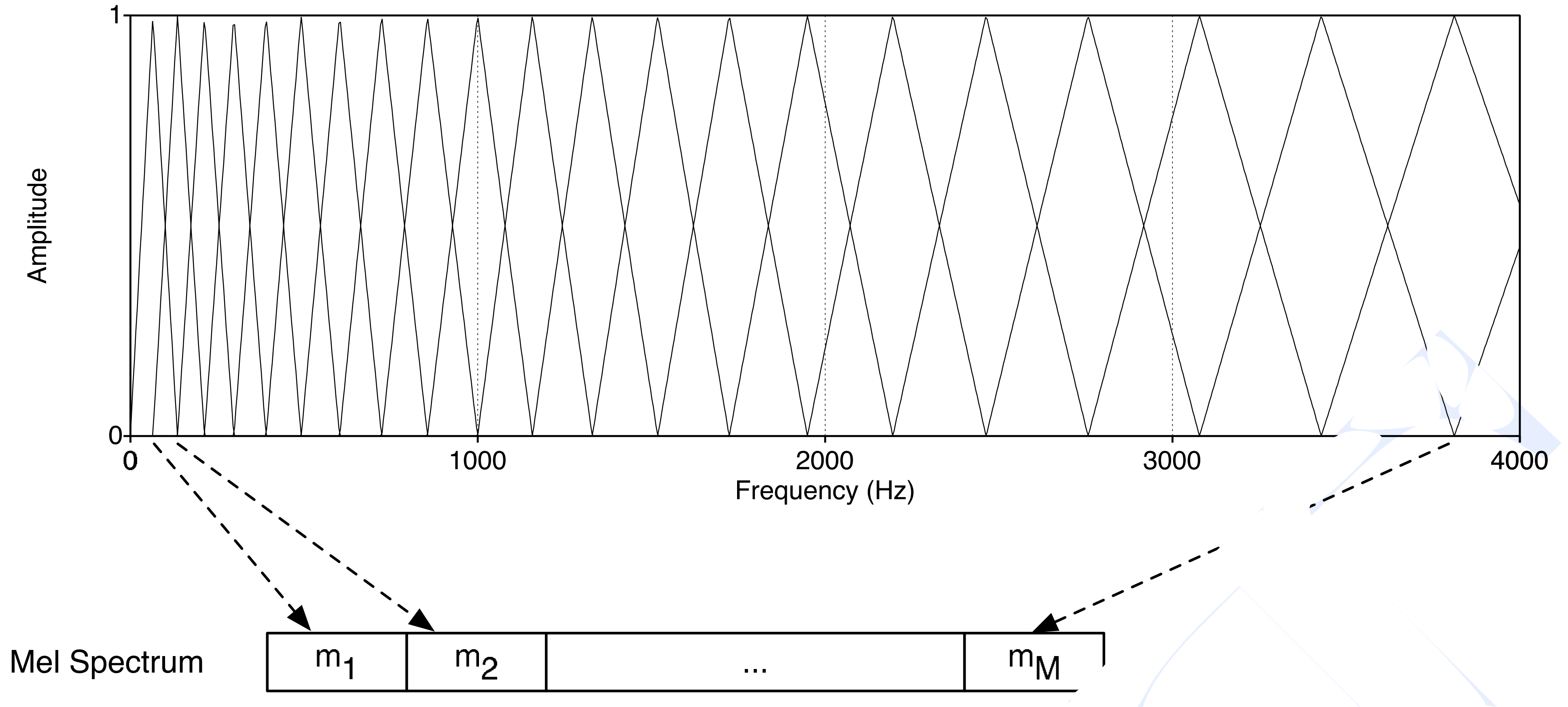
Figure 3.50 Frequency response curves of a cat's basilar membrane (after Ghitza [13]).

Mel filterbank

- Mel frequency can be computed from the raw frequency f as:

$$\mathrm{mel}(f) = 1127 \ln(1 + \frac{f}{700})$$

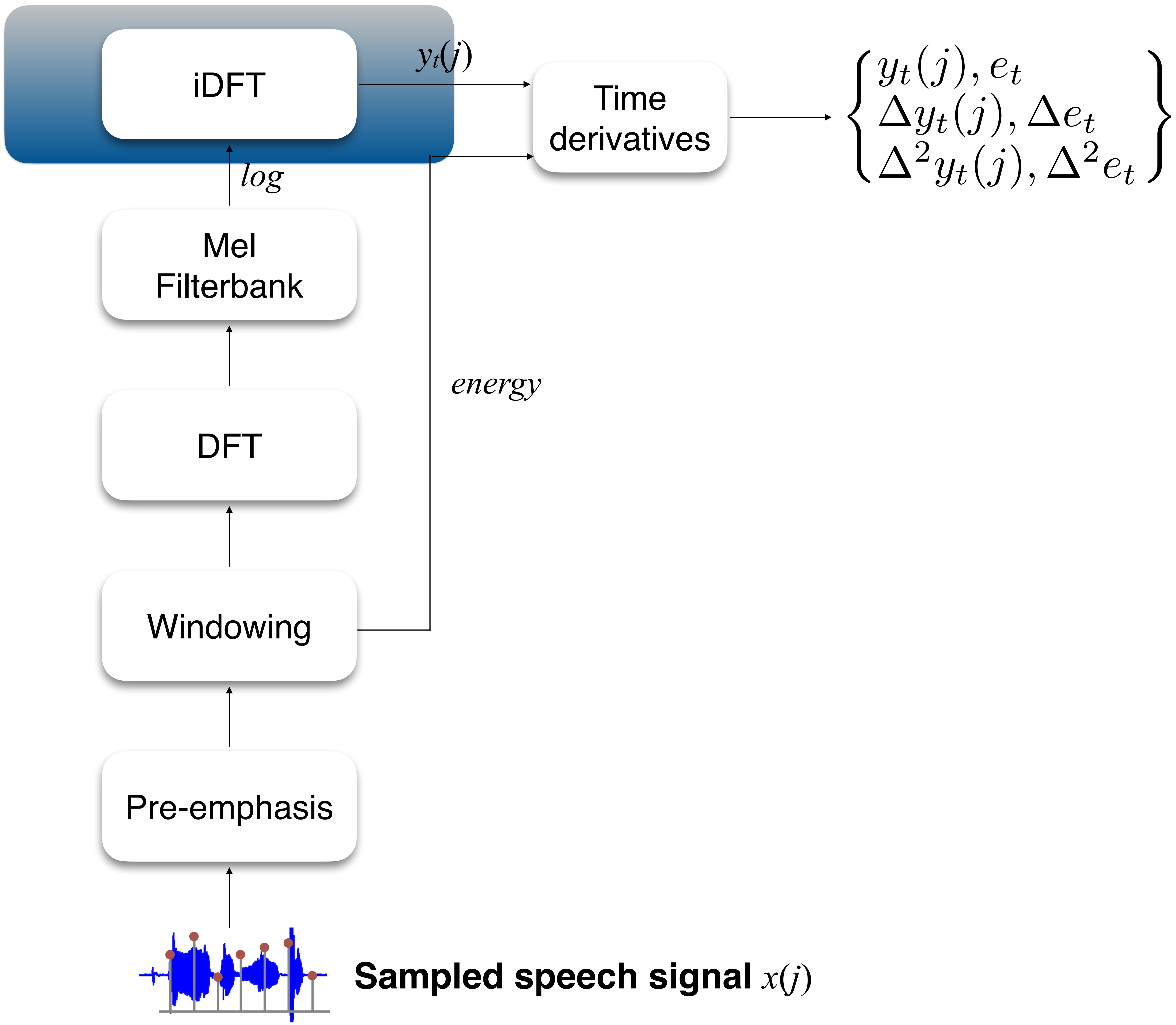
- 10 filters spaced linearly below 1kHz and remaining filters spread logarithmically above 1kHz



- Take log of each mel spectrum value 1) human sensitivity to signal energy is logarithmic 2) log makes features robust to input variations

Image credit: Jurafsky & Martin, Figure 9.13

MFCC Extraction

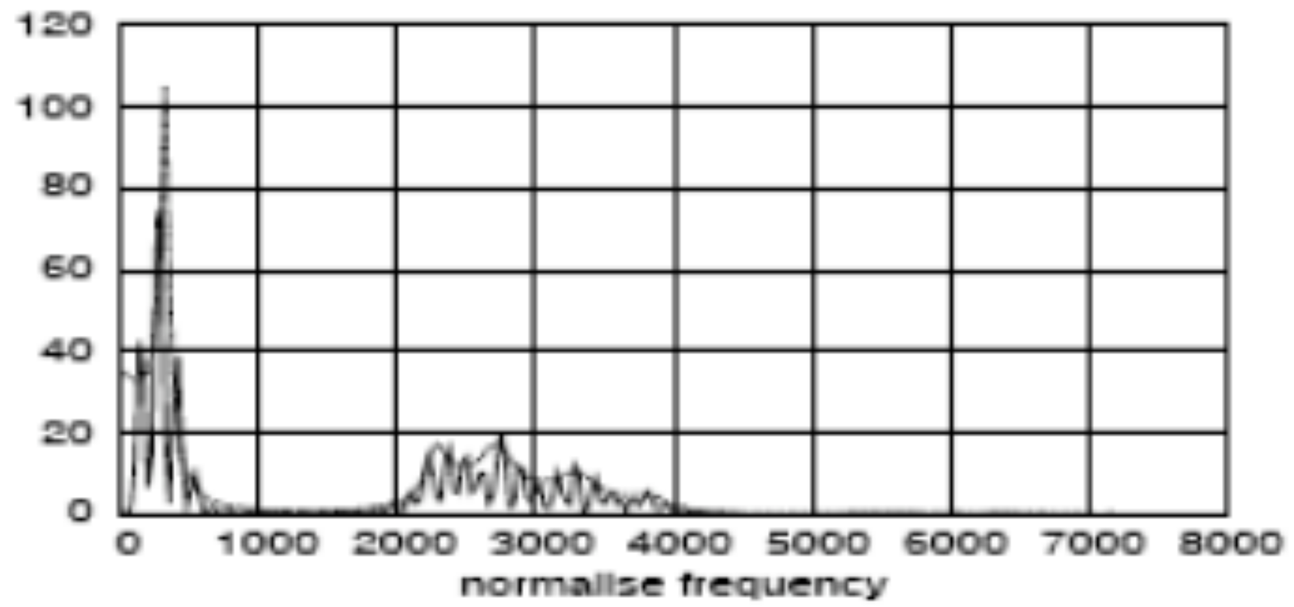


Cepstrum: Inverse DFT

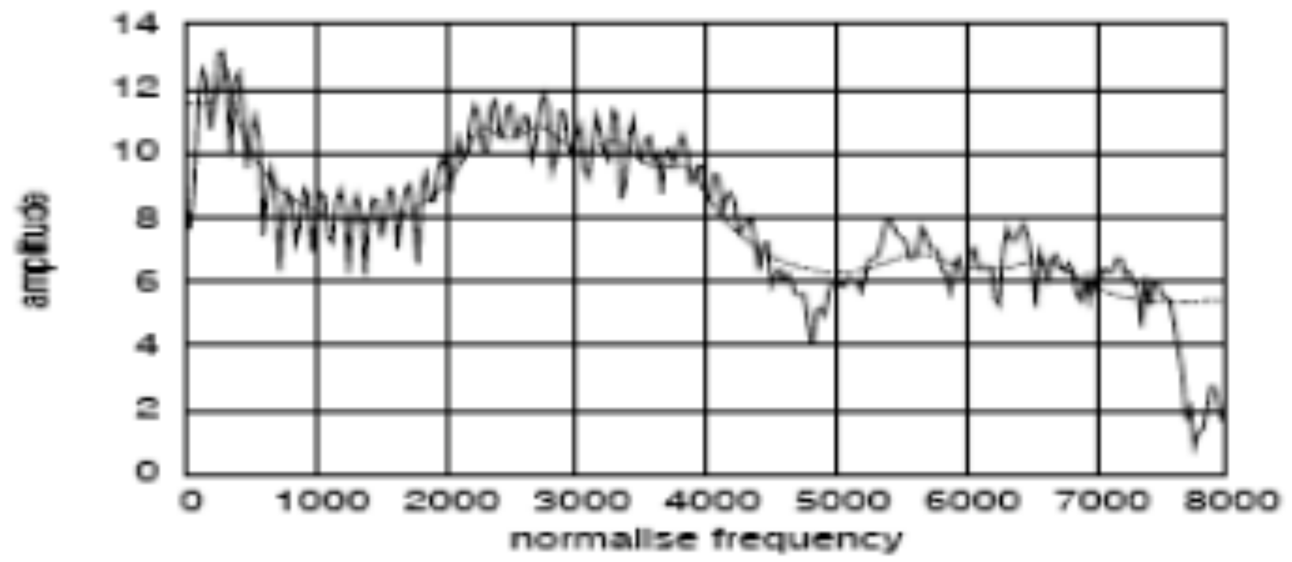
- Recall speech signals are created when a glottal source of a particular fundamental frequency passes through the vocal tract
- Most useful information for phone detection is the vocal tract filter (and not the glottal source)
- How do we deconvolve the source and filter to retrieve information about the vocal tract filter? Cepstrum

Cepstrum

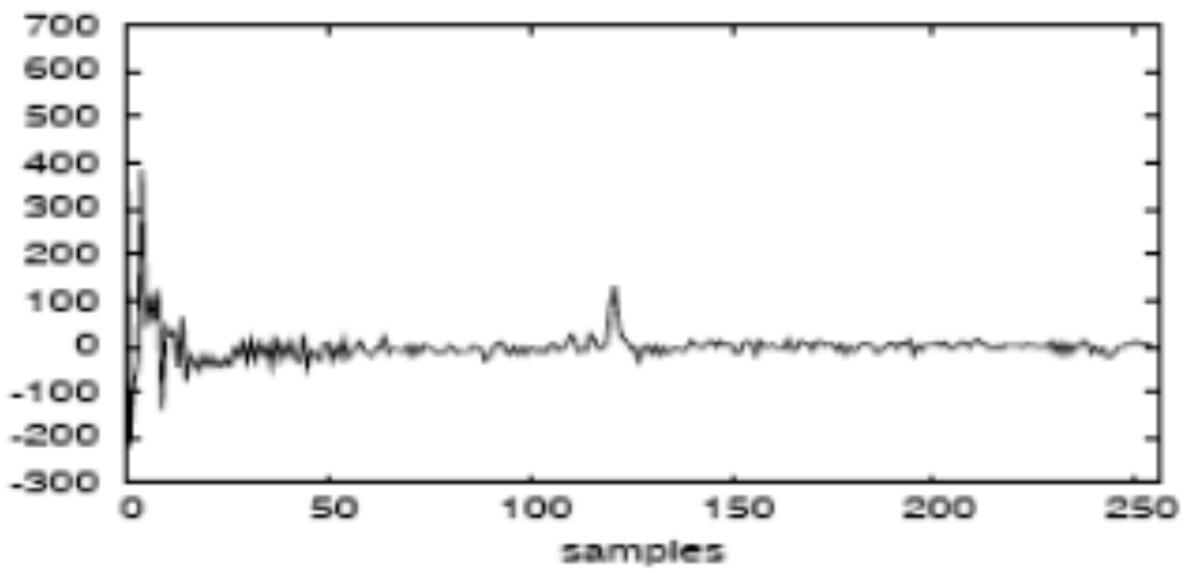
- Cepstrum: spectrum of the log of the spectrum



magnitude spectrum



log magnitude spectrum



cepstrum

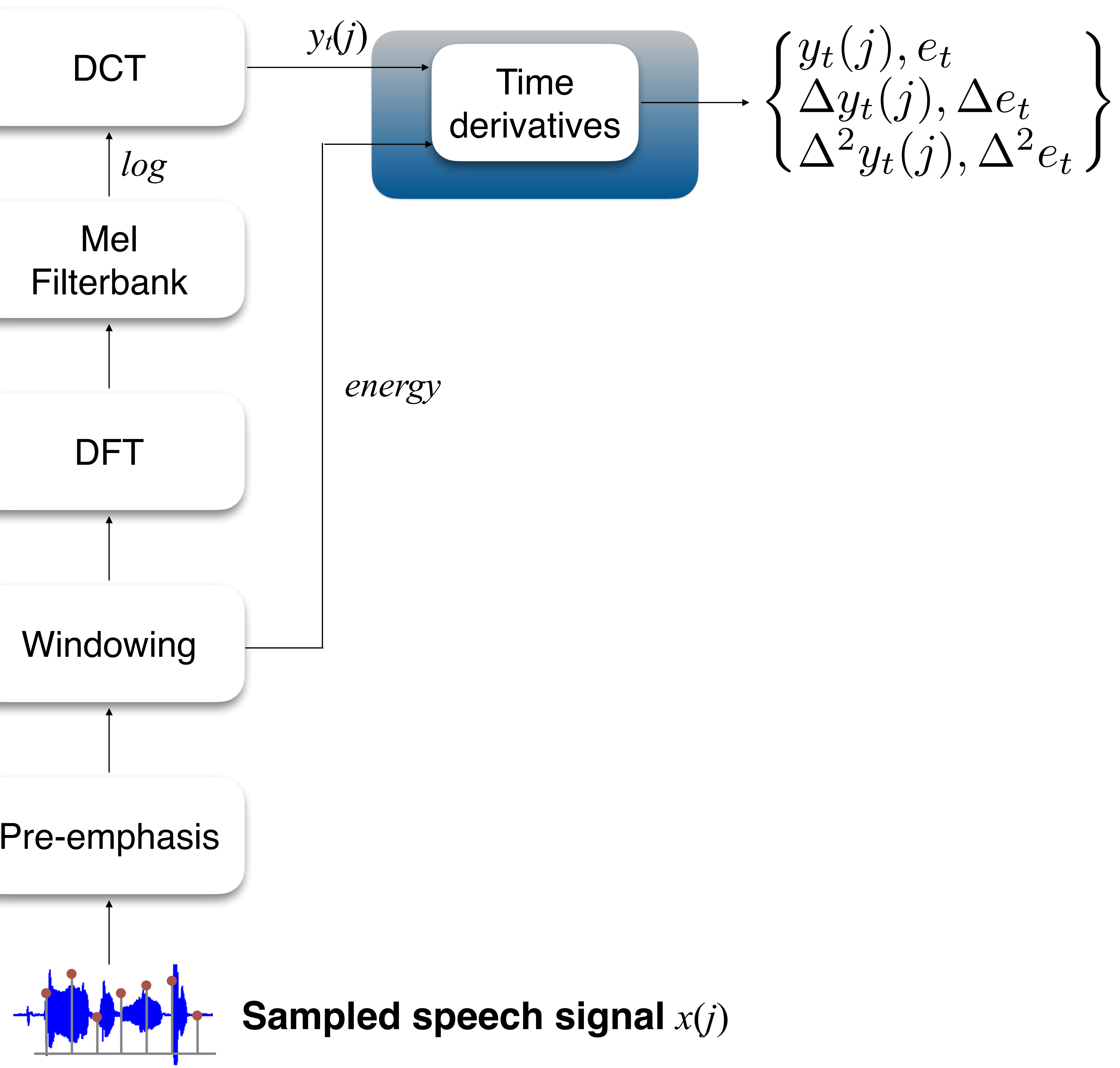
Image credit: Jurafsky & Martin, Figure 9.14

Cepstrum

- For MFCC extraction, we use the first 12 cepstral values
- Variance of the different cepstral coefficients tend to be uncorrelated
  - Useful property when modelling using GMMs in the acoustic model — diagonal covariance matrices will suffice
- Cepstrum is formally defined as the inverse DFT of the log magnitude of the DFT of a signal

$$c[n] = \sum_{n=0}^{N-1} \log\left(\left|\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}\right|\right) e^{j\frac{2\pi}{N}kn}$$

MFCC Extraction



Deltas and double-deltas

- From the cepstrum, use 12 cepstral coefficients for each frame
  - 13th feature represents energy from the frame — computed as sum of the power of the samples in the frame
- Also add features related to change in cepstral features over time to capture speech dynamics:

$$\Delta x_t = x_{t+\tau} - x_{t-\tau}$$ (if x_t is feature vector at time t)

- Typical value for τ is 1 or 2.
- Add 13 delta features (Δx_t) and 13 double-delta features ($\Delta^2 x_t$)

Recap: MFCCs

- Motivated by human speech perception and speech production
- For each speech frame
  - Compute frequency spectrum and apply Mel binning
  - Compute cepstrum using inverse DFT on the log of the mel-warped spectrum
  - 39-dimensional MFCC feature vector: First 12 cepstral coefficients + energy + 13 delta + 13 double-delta coefficients

Other features

- Perceptual Linear Prediction (PLP) features
- Mel filter-bank features (used with DNNs)
- Neural network-based “bottleneck features”
  - Train deep NN using conventional acoustic features
  - Introduce a narrow hidden layer (e.g. 40 hidden units) referred to as the bottleneck layer, forcing the neural network to encode relevant information in this layer
  - Use hidden unit activations in the bottleneck layer as features

Features used for speaker recognition

- E.g. from a recent speaker identification (VoxCeleb) task.
- Input features, F: Spectrograms generated in a sliding window fashion using a Hamming window of width 25ms and step 10ms
- F used as input to a CNN architecture
- Mean and variance normalisation performed on every frequency bin of the spectrum (crucial for performance!)

Accuracy	Top-1 (%)	Top-5 (%)
I-vectors + SVM	49.0	56.6
I-vectors + PLDA + SVM	60.8	75.6
CNN-fc-3s no var. norm.	63.5	80.3
CNN-fc-3s	72.4	87.4

Nagrani et al.,"VoxCeleb: a large-scale speaker identification dataset", Interspeech 2017