Most slides come from http://www.inf.ed.ac.uk/teaching/courses/asr/2019-20/asr04-signal.pdf

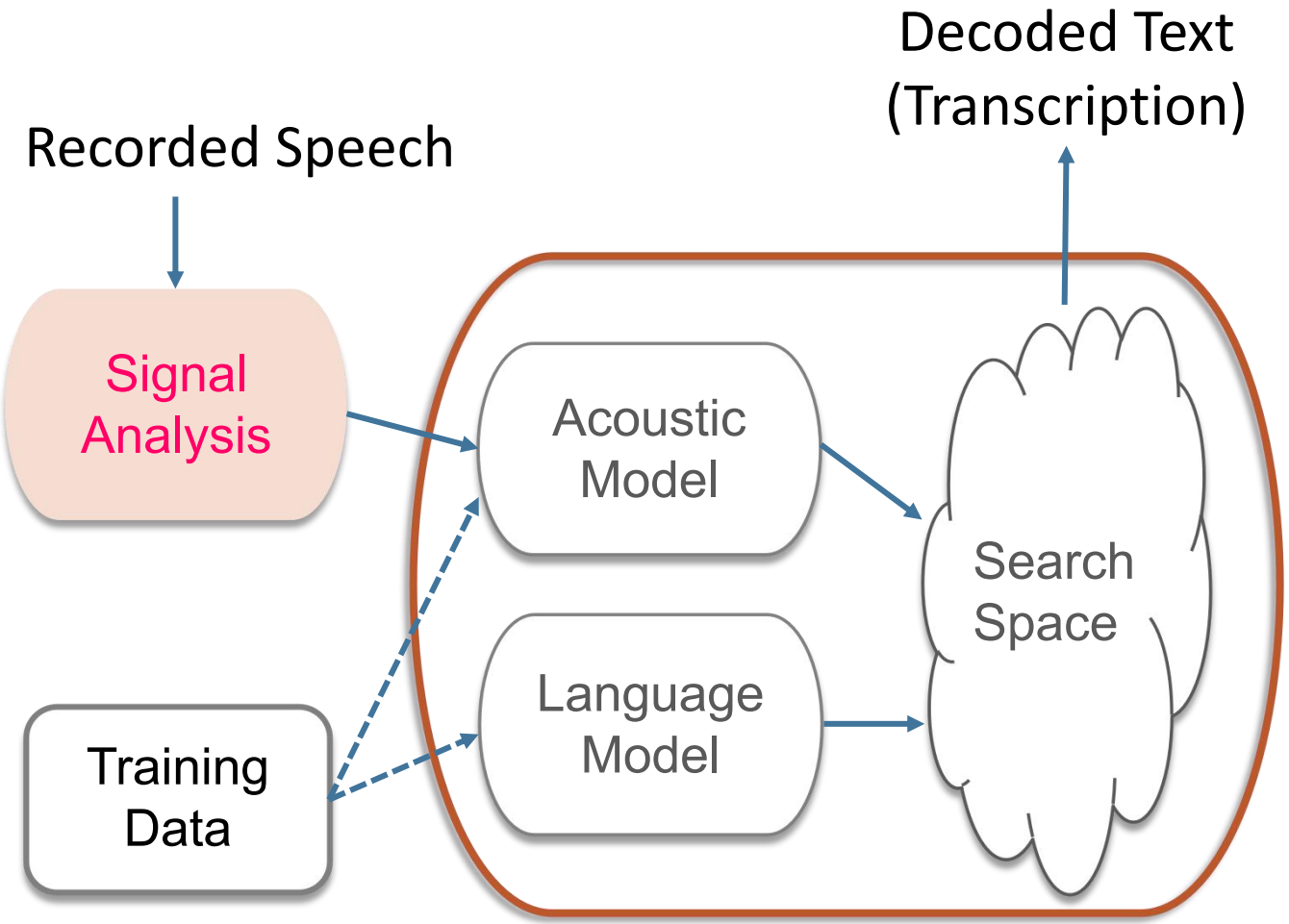
Speech Signal Analysis

YING SHEN

SCHOOL OF SOFTWARE ENGINEERING

TONGJI UNIVERSITY

Speech signal analysis for ASR



Content

Features for ASR

Spectral analysis

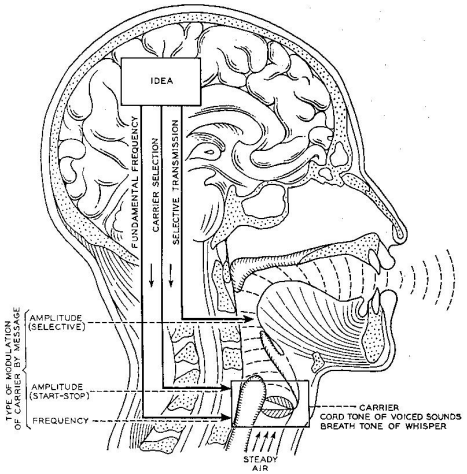
Cepstral analysis

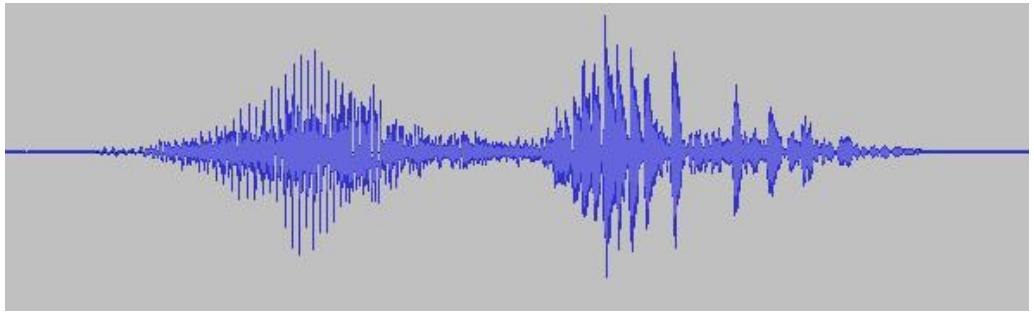
Standard features for ASR: FBANK, MFCCs and PLP analysis

Dynamic features

A/D conversion

Convert analogue signals in digital form



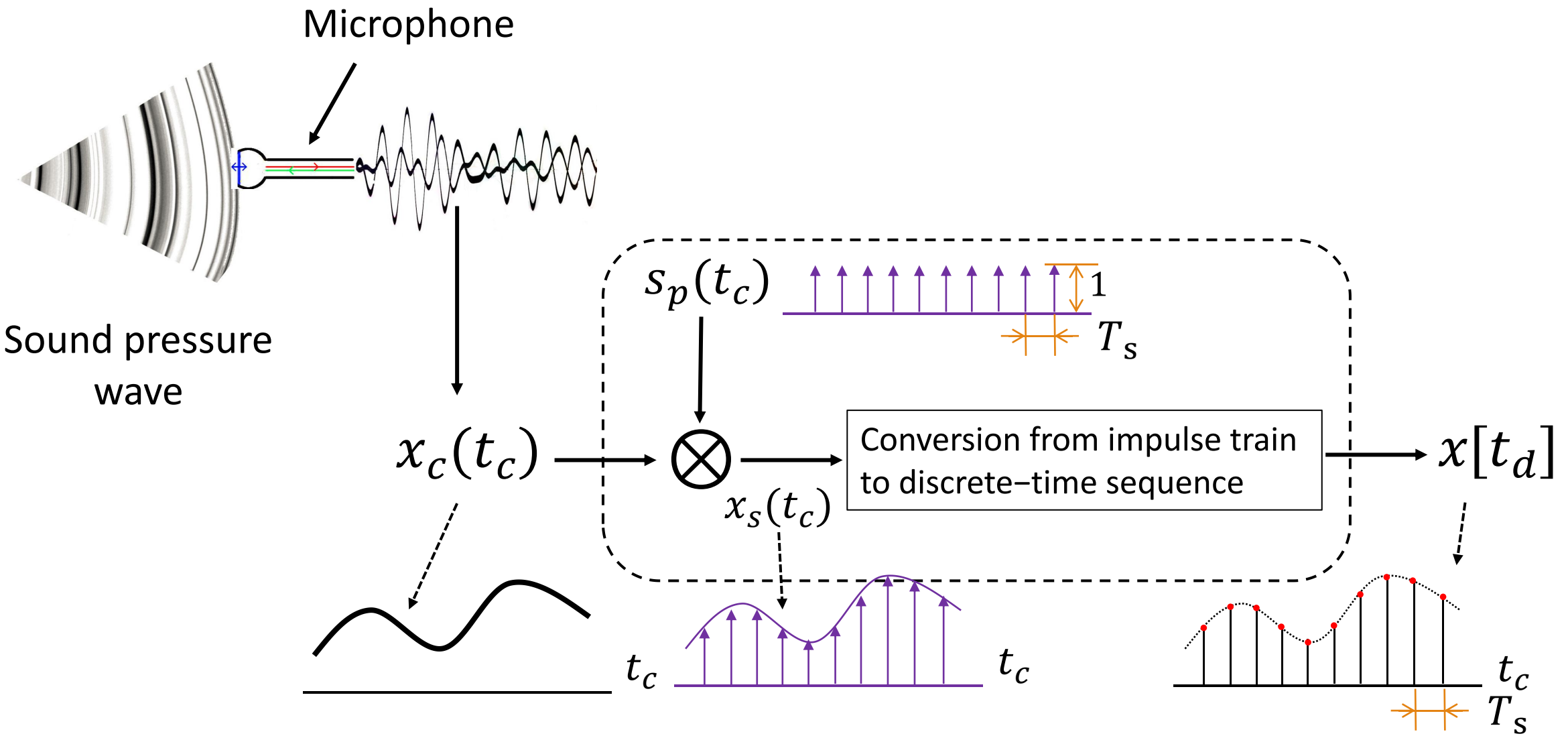


Sound pressure wave

“你好”

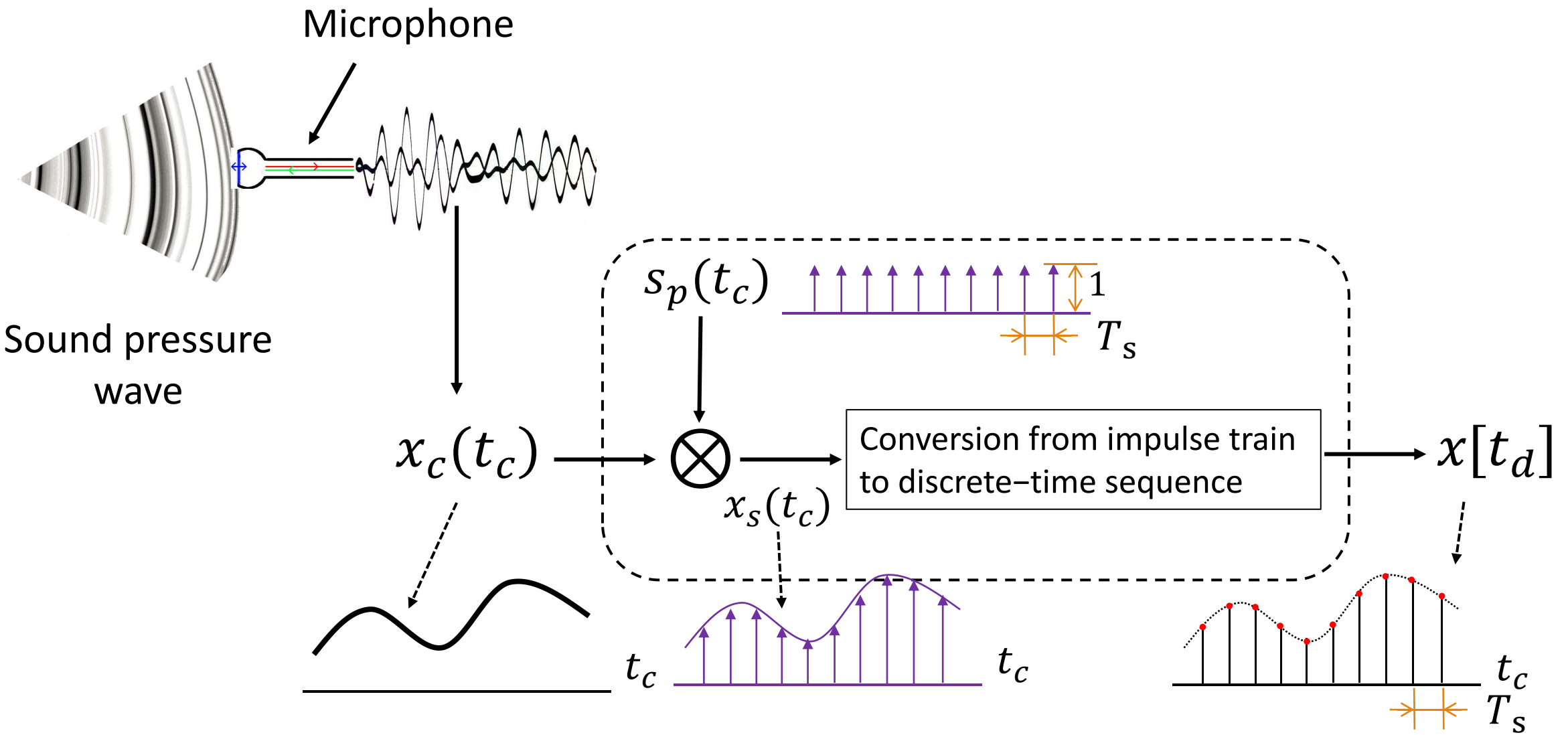
A/D conversion

Convert analogue signals in digital form



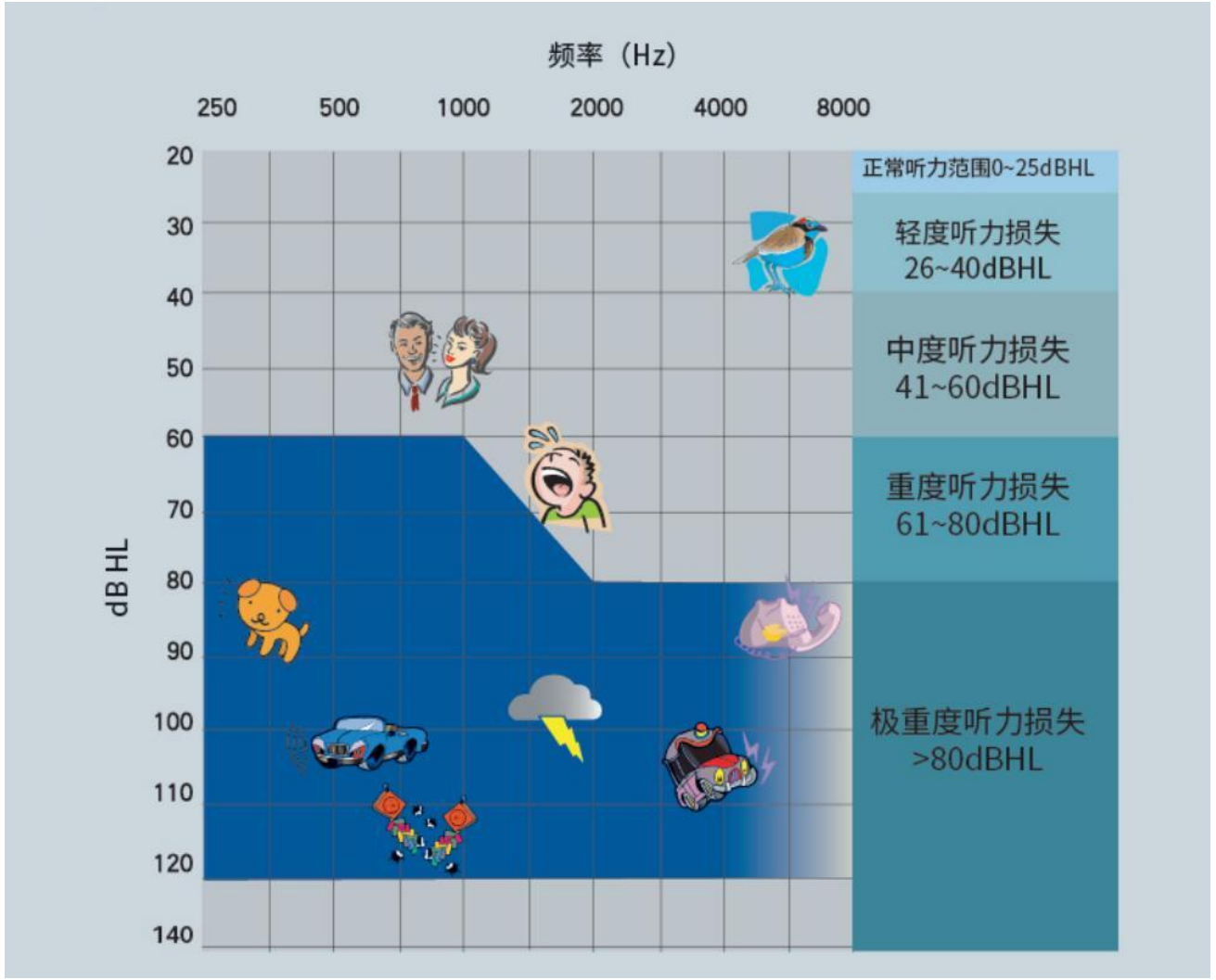
A/D conversion - Sampling

Convert analogue signals in digital form



A/D conversion - Sampling

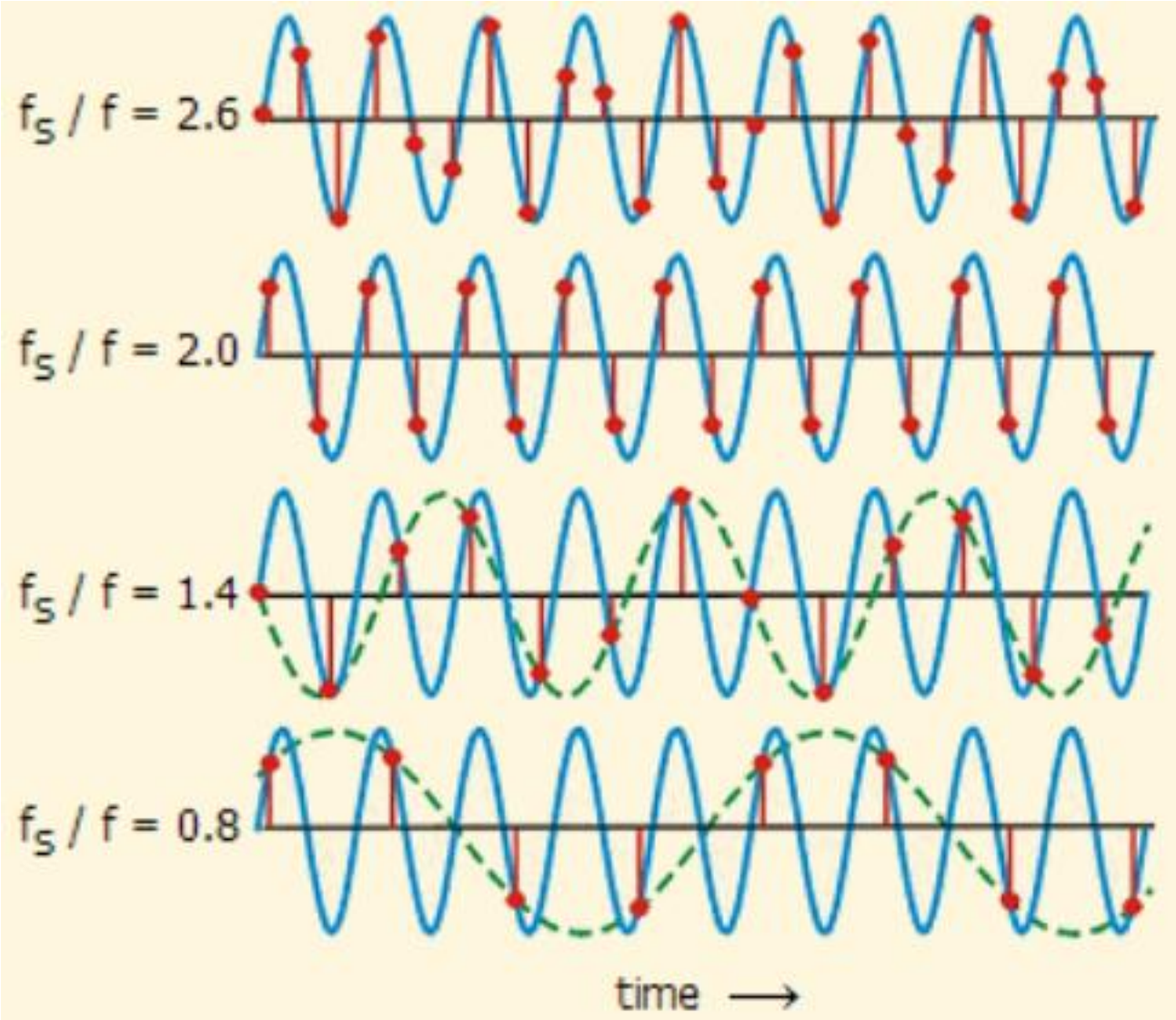
$$SPL = 20\log\left(\frac{p}{p_0}\right) dB$$



Nyquist-Shannon sampling theorem

The minimum sampling frequency of a signal that it will not distort its underlying information, should be double the frequency of its highest frequency component.

If f_s is the **sampling frequency**, then the **critical frequency** (or **Nyquist limit**) f_N is defined as equal to $\frac{f_2}{2}$.



Nyquist-Shannon sampling theorem

The minimum sampling frequency of a signal that it will not distort its underlying information, should be double the frequency of its highest frequency component.

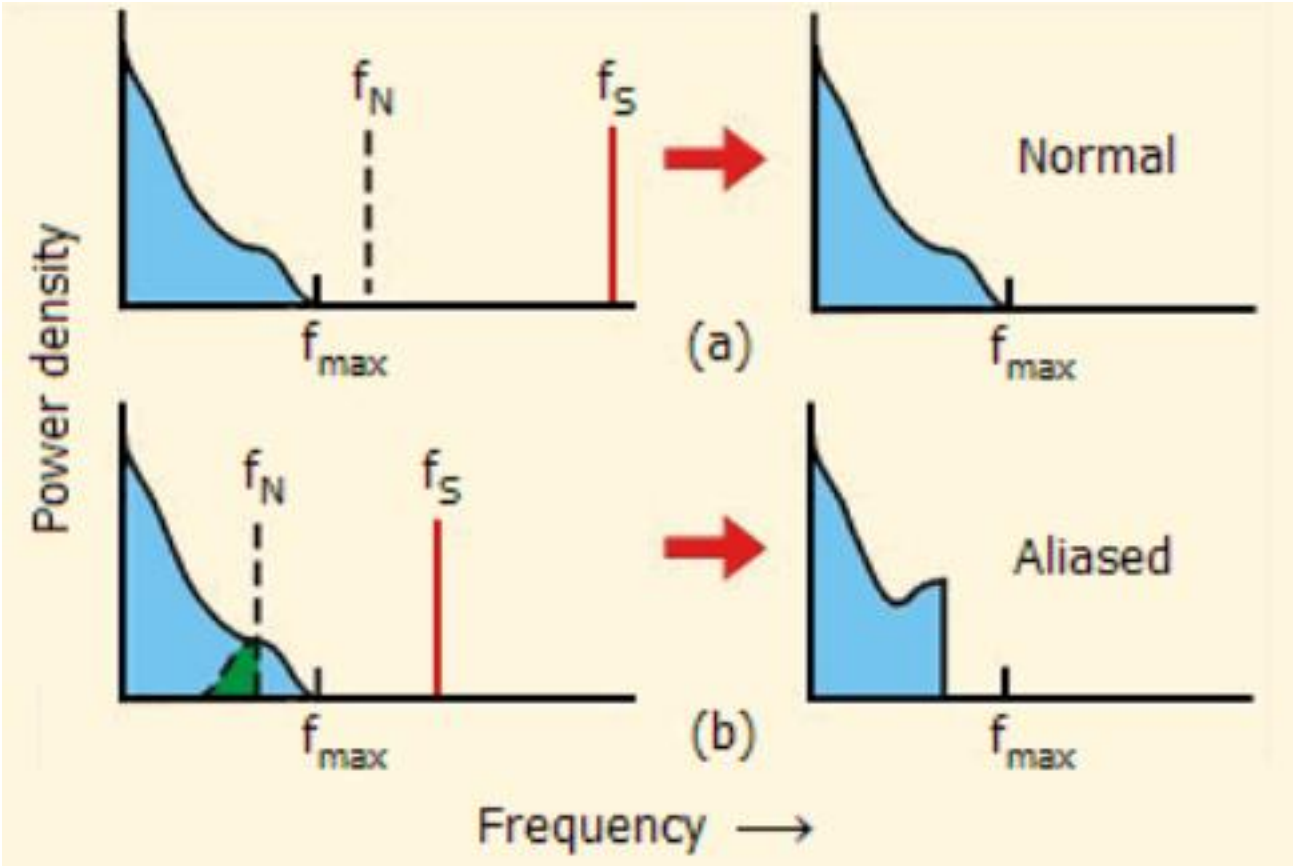
If f_s is the **sampling frequency**, then the **critical frequency** (or **Nyquist limit**) f_N is defined as equal to $\frac{f_2}{2}$.

Nyquist-Shannon sampling theorem

Notes:

(1) In practice, the sampling rate f_s is commonly selected in the range 2.5 ~$4f_{max}$. For digital recording of music in CD a sampling rate of 44.1 kHz is commonly used.

(2) Prior to sampling, the signal must pass through a low-pass filter which will remove all unnecessary components (e.g. noise) higher than f_{max}, preventing thus the "contamination" of the stored signal by their aliased frequencies.



A/D conversion – Sampling

Things to know:

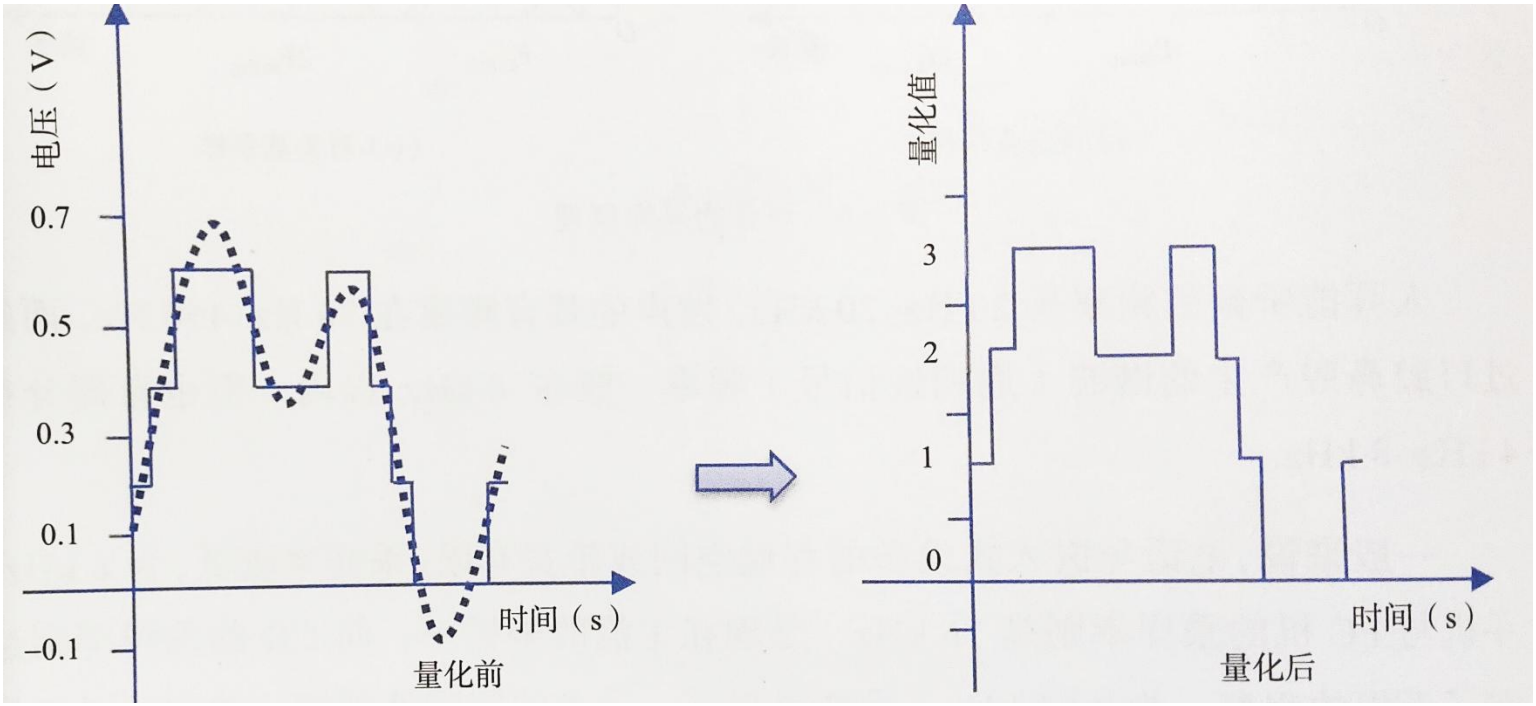
Sampling Frequency ($F_s = 1/T_s$)

Speech	Sufficient F_s
Microphone voice (< 10kHz)	20 kHz
Telephone voice (< 4kHz)	8 kHz

Analogue low-pass filtering to avoid ‘aliasing’

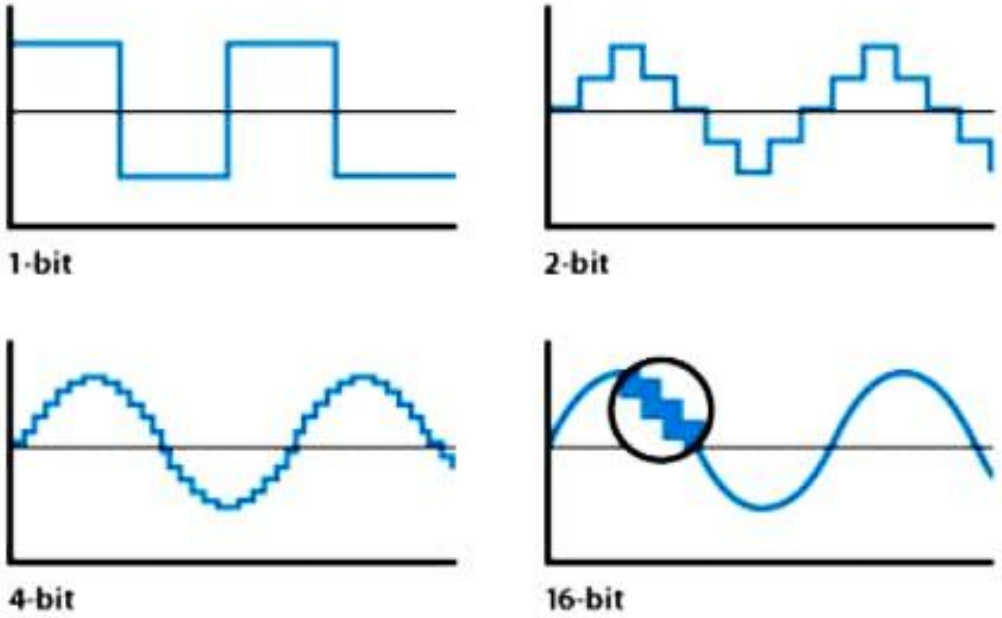
NB: the cut-o frequency should be less than the Nyquist frequency (=$F_s/2$)

A/D conversion – Quantization



A/D conversion – Code scheme

The resolution of the ADC can be determined by its bit length



Bit Length	Levels	Step Size (5V Range)
8-bits	256	19.53 mV
10-bits	1024	4.88 mV
12-bits	4096	1.22 mV
16-bits	65536	76.29 µV
18-bits	262144	19.07 µV
20-bits	1048576	4.76 µV
24-bits	16777216	0.298 µV

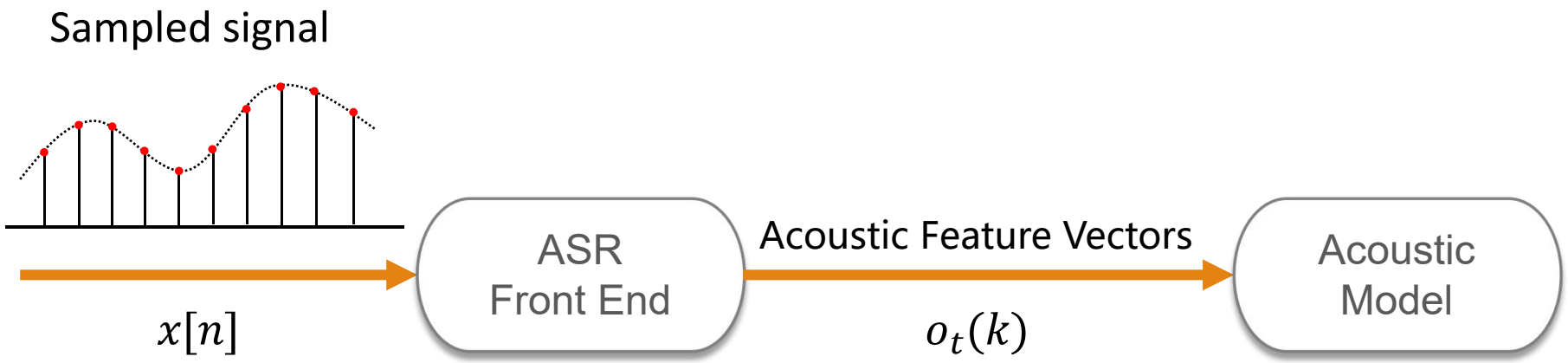
A/D conversion – Code scheme

NBC (Natural Binary Code), FBC (Folded Binary Code), RBC (Grayor Reflected Binary Code)

量化值序号	自然码 NBC	折叠码 FBC	格雷码 RBC
15	1111	1111	1000
14	1110	1110	1001
13	1101	1101	1011
12	1100	1100	1010
11	1011	1011	1110
10	1010	1010	1111
9	1001	1001	1101
8	1000	1000	1100
7	0111	0000	0100
6	0110	0001	0101
5	0101	0010	0111
4	0100	0011	0110
3	0011	0100	0010
2	0010	0101	0011
1	0001	0110	0001
0	0000	0111	0000

Acoustic Features for ASR

Speech signal analysis to produce a sequence of acoustic feature vectors



Acoustic Features for ASR

Features should contain sufficient information to distinguish between phones

- Good time resolution (10ms)
- Good frequency resolution (20 ~ 40 channels)

Be separated from F_0 and its harmonics

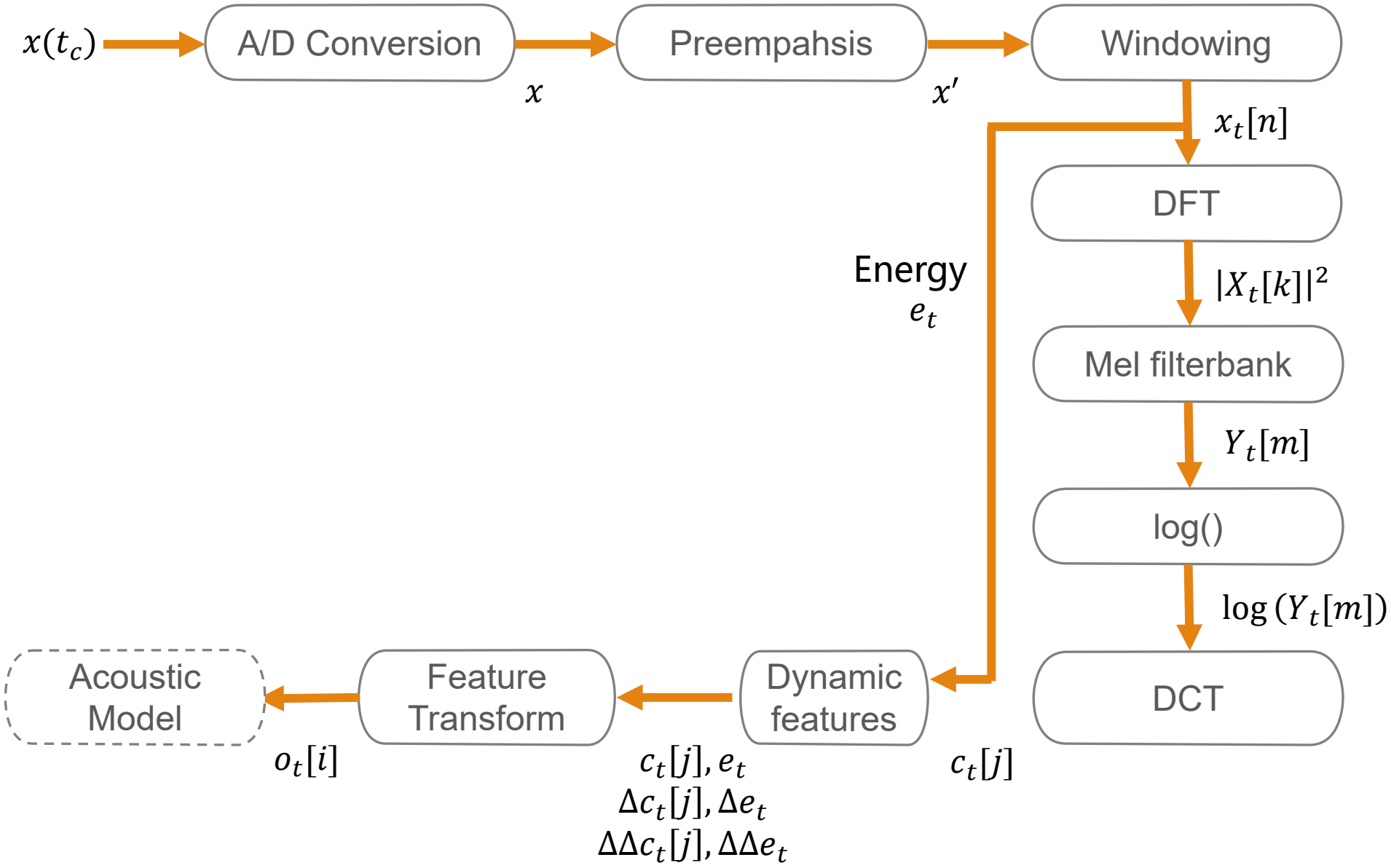
Be robust against speaker variation

Be robust against noise or channel distortions

Have good “pattern recognition characteristics”

- Low feature dimension
- Features are independent of each other (NB: this applies to GMMs, but not required for NN-based systems)

MFCC-based front end for ASR



Pre-emphasis and spectral tilt

Pre-emphasis increases the magnitude of higher frequencies in the speech signal compared with lower frequencies

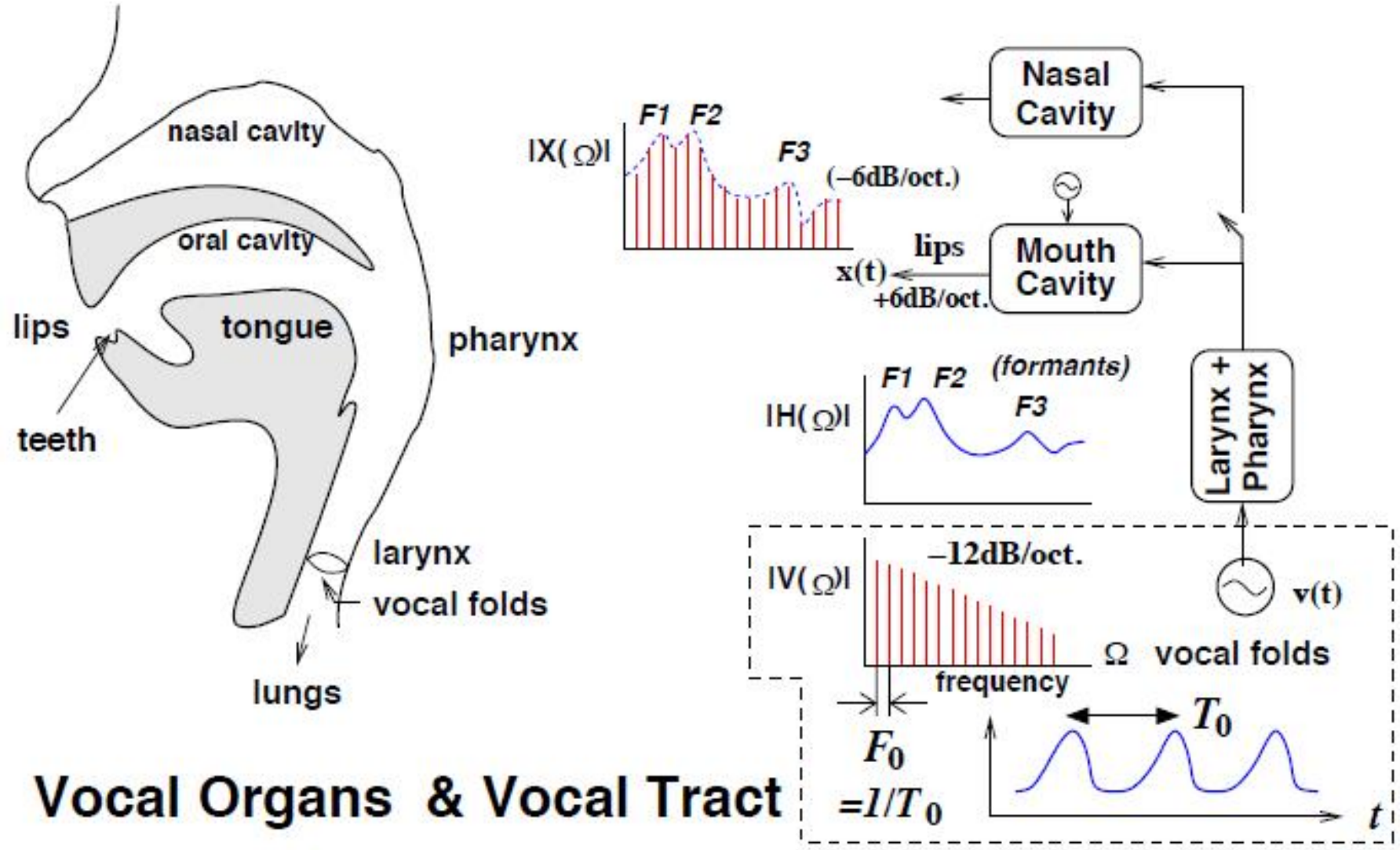
Spectral Tilt

- The speech signal has more energy at low frequencies (for voiced speech)
- This is due to the glottal source (see the figure)

Pre-emphasis (first-order) filter boosts higher frequencies:

$$x'[n] = x[n] - \alpha x[n-1], \;\; 0.95 < \alpha < 0.99, \; 1 \leq n \leq \#\, of \;\; samples$$

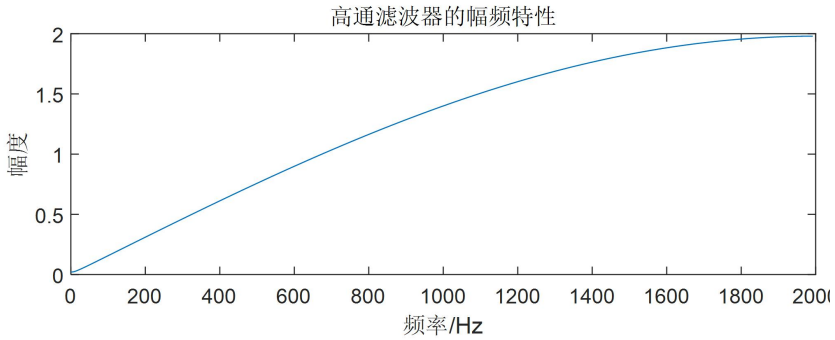
Speech production model

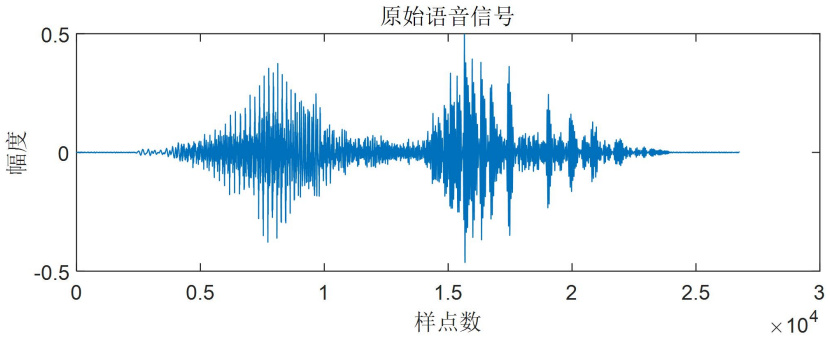


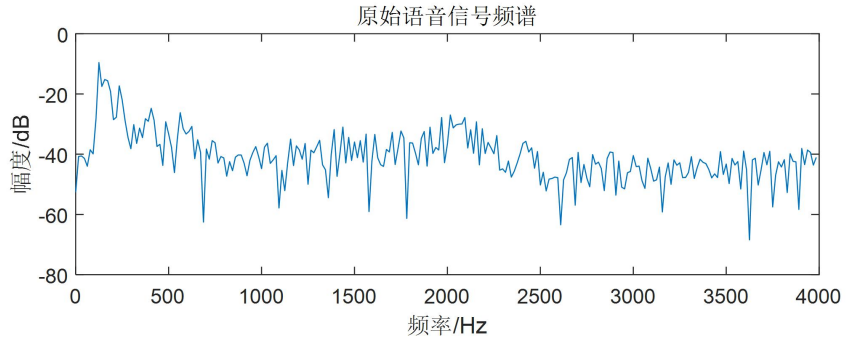
Vocal Organs & Vocal Tract

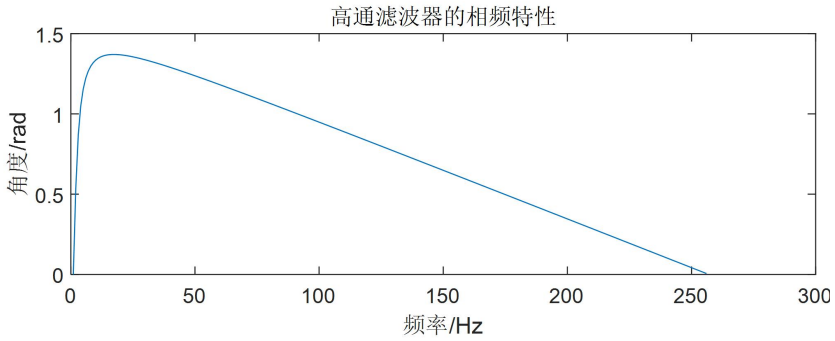
(F_0 : fundamental frequency)

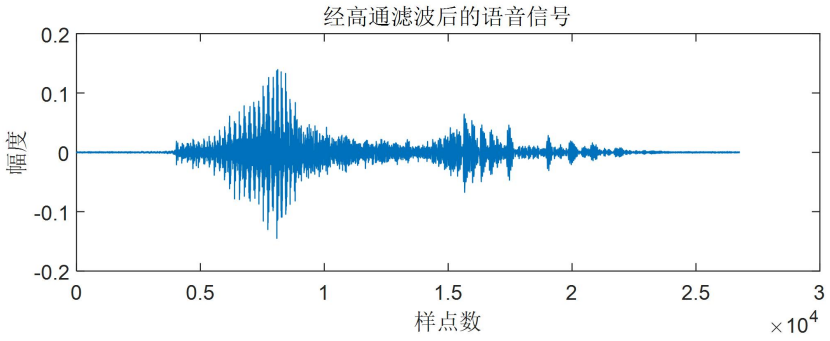
Pre-emphasis and spectral tilt

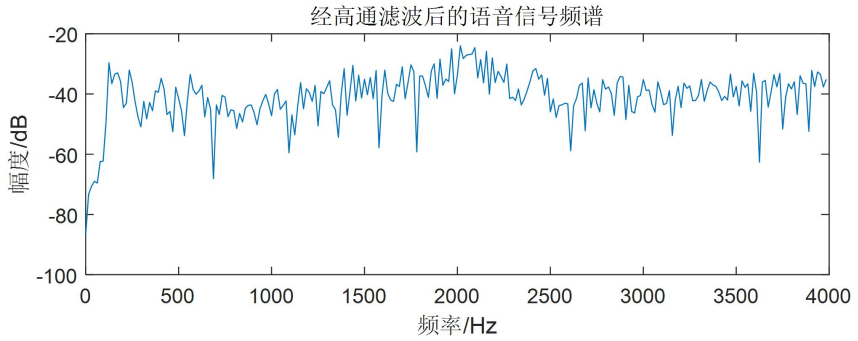












Windowing

The speech signal is constantly changing (non-stationary)

Signal processing algorithms usually assume that the signal is stationary

Piecewise stationarity: model speech signal as a sequence of frames (each assumed to be stationary)

Windowing

Windowing: multiply the waveform $x'[n]$ of time frame t by a window $w_t[n]$ (in time domain):

$$x_t[n] = w_t[n]x'[n]$$

Simply cutting out a short segment (frame) from $x'[n]$ is a rectangular window – causes discontinuities at the edges of the segment

$$w_t[n] = \begin{cases} 1, & (t-1) * L \leq n \leq t * L - 1 \\ 0, & n \leq (t-1) * L, n \geq t * L - 1 \end{cases}, \quad 1 \leq t \leq \#\ of\ frames$$

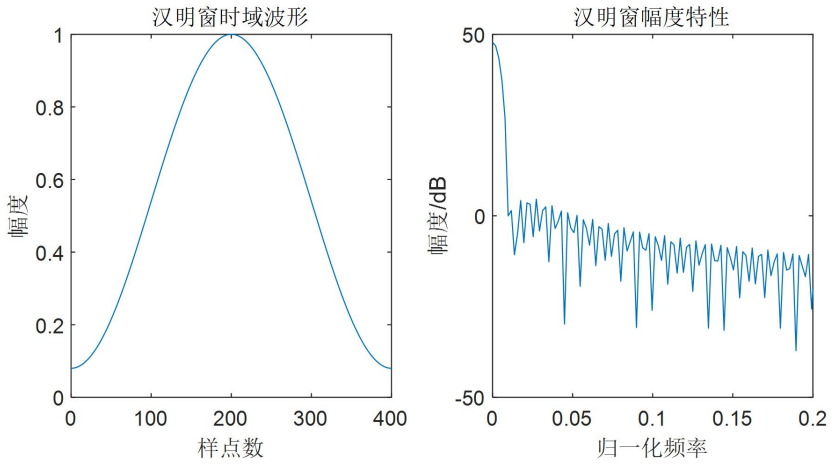
L: window width

Windowing

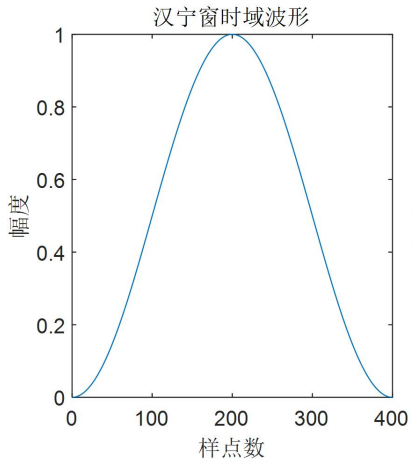
Instead, a tapered window is usually used e.g. Hamming ($\alpha = 0.46164$) or Hanning ($\alpha = 0.5$) window

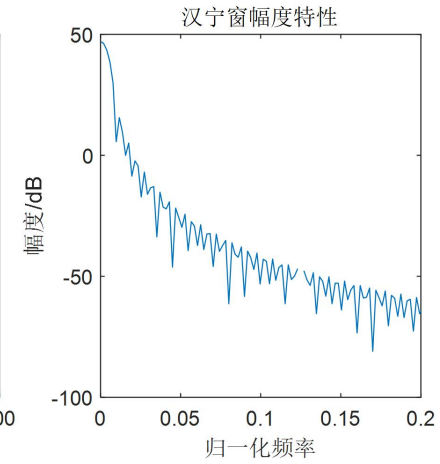
$$w_t[n] = \begin{cases} (1-\alpha) - \alpha\cos\left(\frac{2\pi n}{L-1}\right), & (t-1) * L \leq n \leq t * L - 1 \\ 0, & n \leq (t-1) * L, n \geq t * L - 1 \end{cases}$$

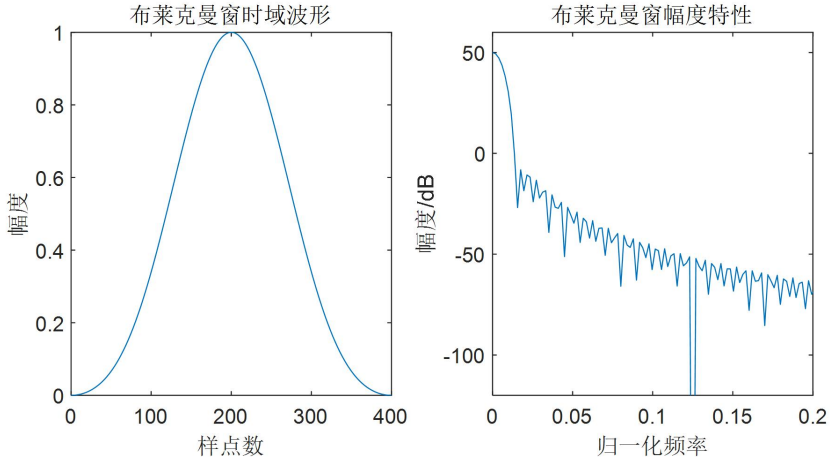
Windowing


汉明窗时域波形
幅度
样点数
0
0.2
0.4
0.6
0.8
1
0
100
200
300
400

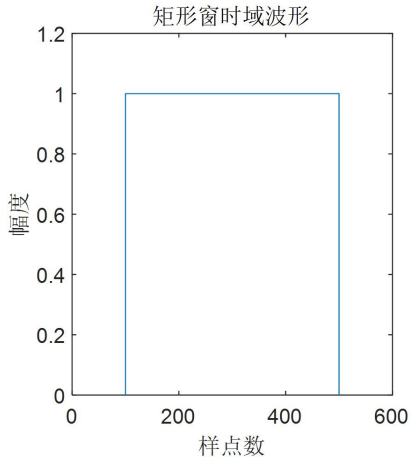
汉明窗幅度特性
幅度/dB
归一化频率
50
0
-50
0
0.05
0.1
0.15
0.2

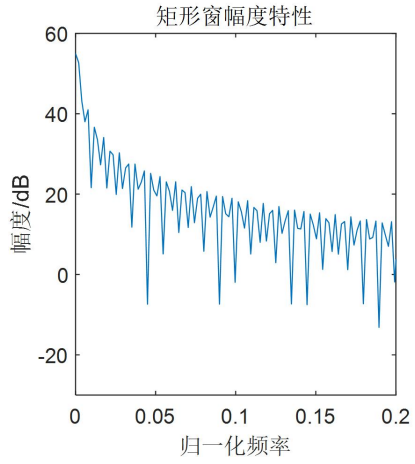

汉宁窗时域波形
幅度
样点数
0
0.2
0.4
0.6
0.8
1
0
100
200
300
400


汉宁窗幅度特性
幅度/dB
归一化频率
50
0
-50
-100
0
0.05
0.1
0.15
0.2


布莱克曼窗时域波形
幅度
样点数
0
0.2
0.4
0.6
0.8
1
0
100
200
300
400

布莱克曼窗幅度特性
幅度/dB
归一化频率
50
0
-50
-100
0
0.05
0.1
0.15
0.2

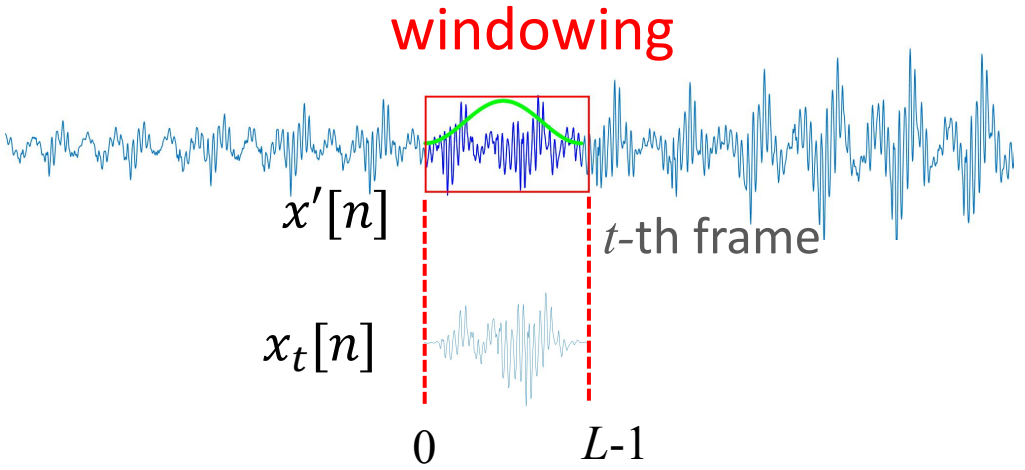

矩形窗时域波形
幅度
样点数
0
0.2
0.4
0.6
0.8
1
1.2
0
200
400
600


矩形窗幅度特性
幅度/dB
归一化频率
60
40
20
0
-20
0
0.05
0.1
0.15
0.2

Windowing

Window the signal $x'[n]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: wide-band, high time resolution, low frequency resolution
- Long frame width: narrow-band, low time resolution, high frequency resolution



For ASR:

- frame width ~ 25ms
- frame shift ~ 10ms

Windowed signal

$$x_t[n] = w_t[n]x'[n]$$

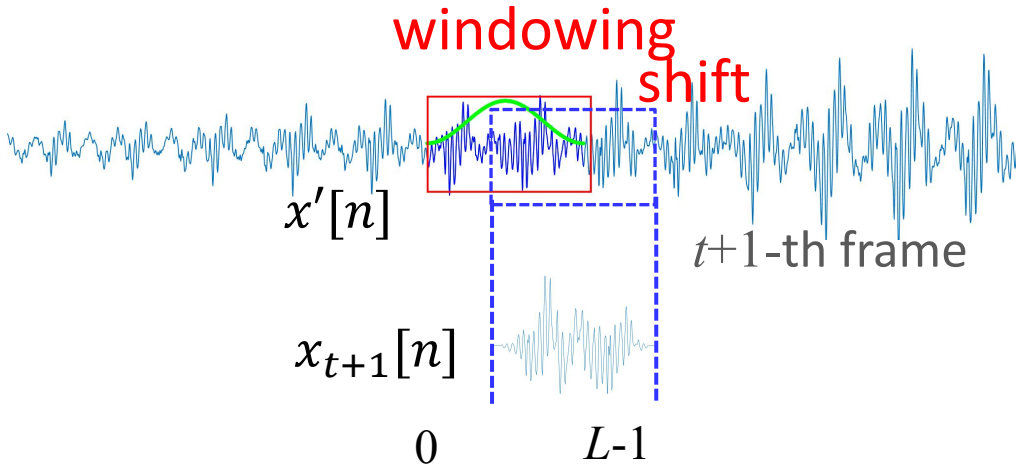
Windowing

Window the signal $x'[n]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: wide-band, high time resolution, low frequency resolution
- Long frame width: narrow-band, low time resolution, high frequency resolution

For ASR:

- frame width ~ 25ms
- frame shift ~ 10ms



Windowed signal

$$x_t[n] = w_t[n]x'[n]$$

Discrete Fourier Transform (DFT)

Purpose: extracts spectral information from a windowed signal (i.e. how much energy at each frequency band)

Input: windowed signal $x[0], \ldots, x[L-1]$ (time domain)

Output: a complex number $X[k]$ for each of N frequency bands representing magnitude and phase for the k-th frequency component (frequency domain)

Discrete Fourier Transform (DFT):

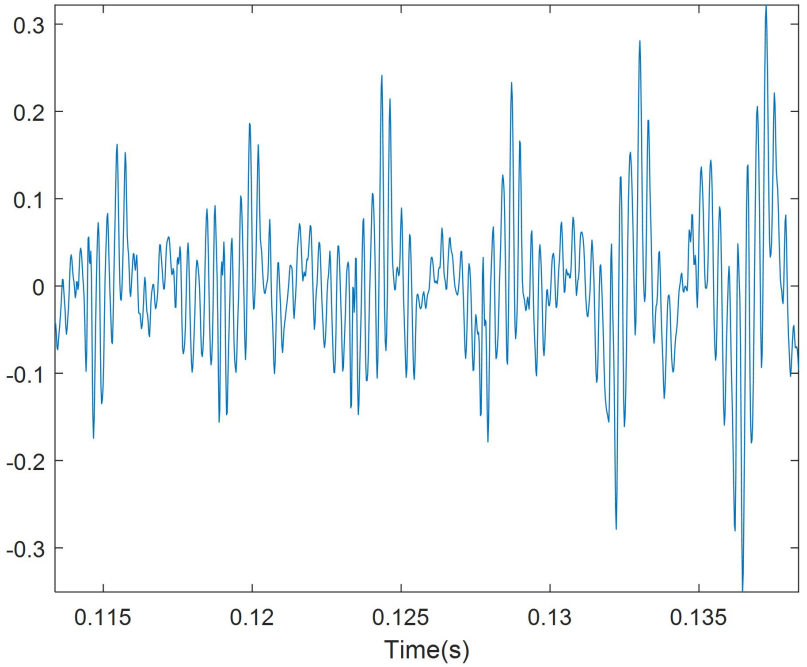
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

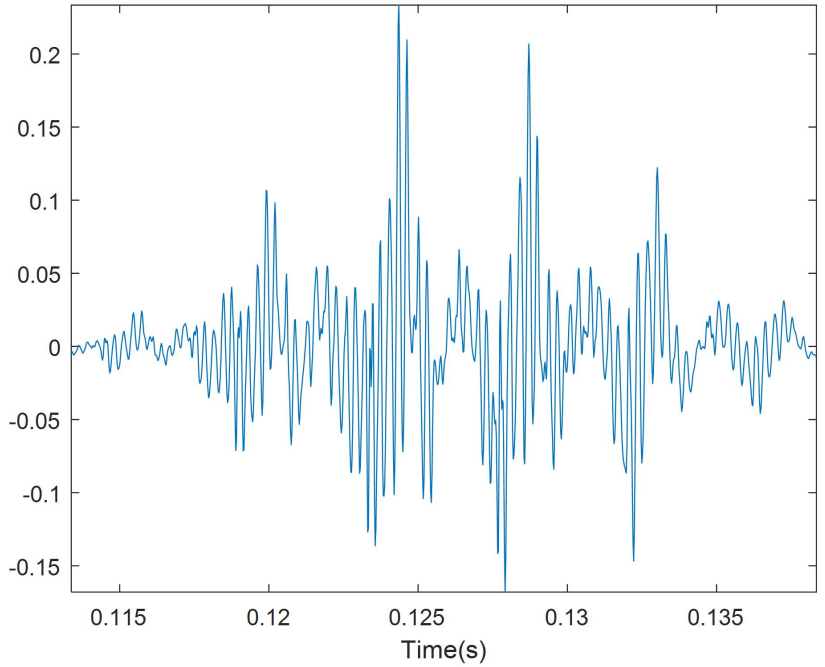
$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$

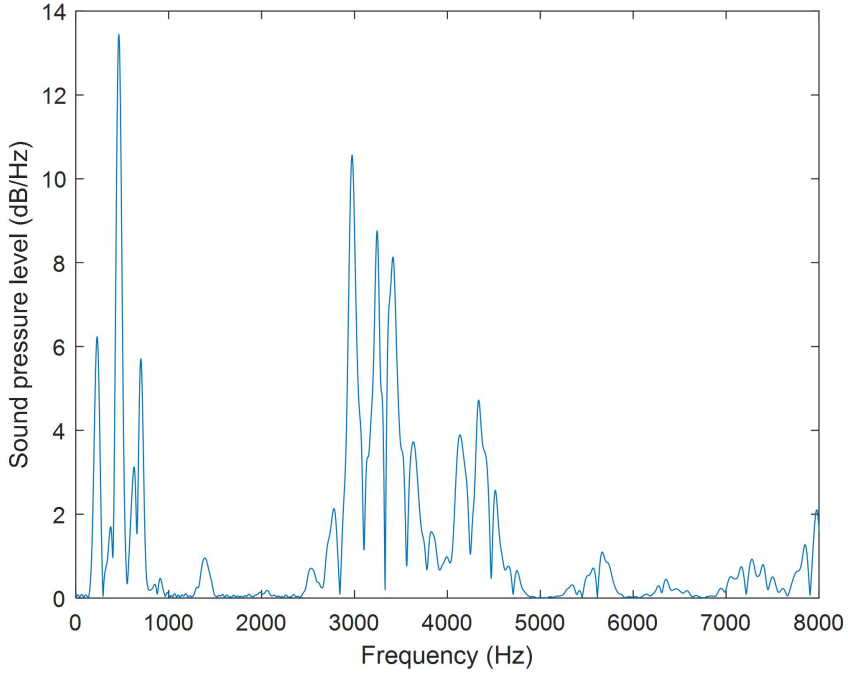
Fast Fourier Transform (FFT) — efficient algorithm for computing DFT when N is a power of 2, and $N \geq L$.

DFT Spectrum

25ms Hamming window of “你好” speech (sample rate = 44100HZ) and its spectrum computed by DFT







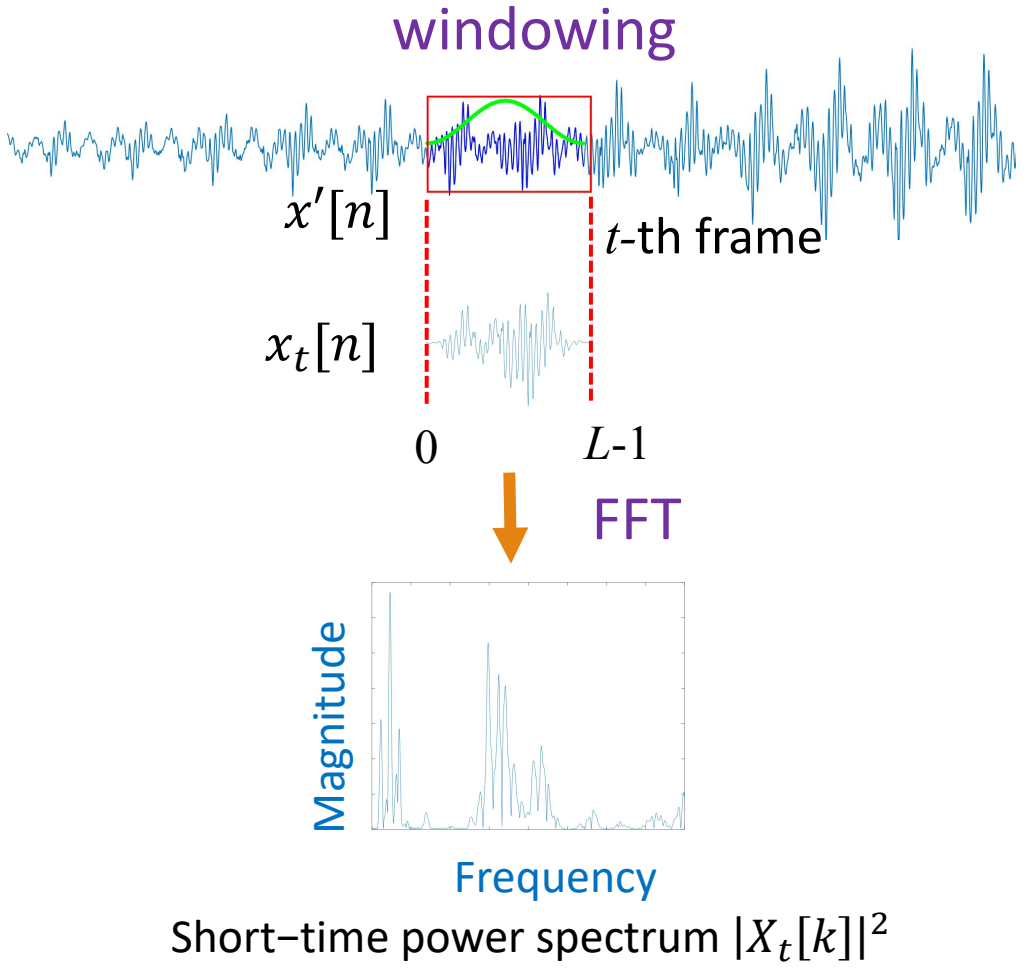
Windowing and spectral analysis

Window the signal $x'[n]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: wide-band, high time resolution, low frequency resolution
- Long frame width: narrow-band, low time resolution, high frequency resolution

For ASR:

- frame width ~ 25ms
- frame shift ~ 10ms



Short−time power spectrum $|X_t[k]|^2$

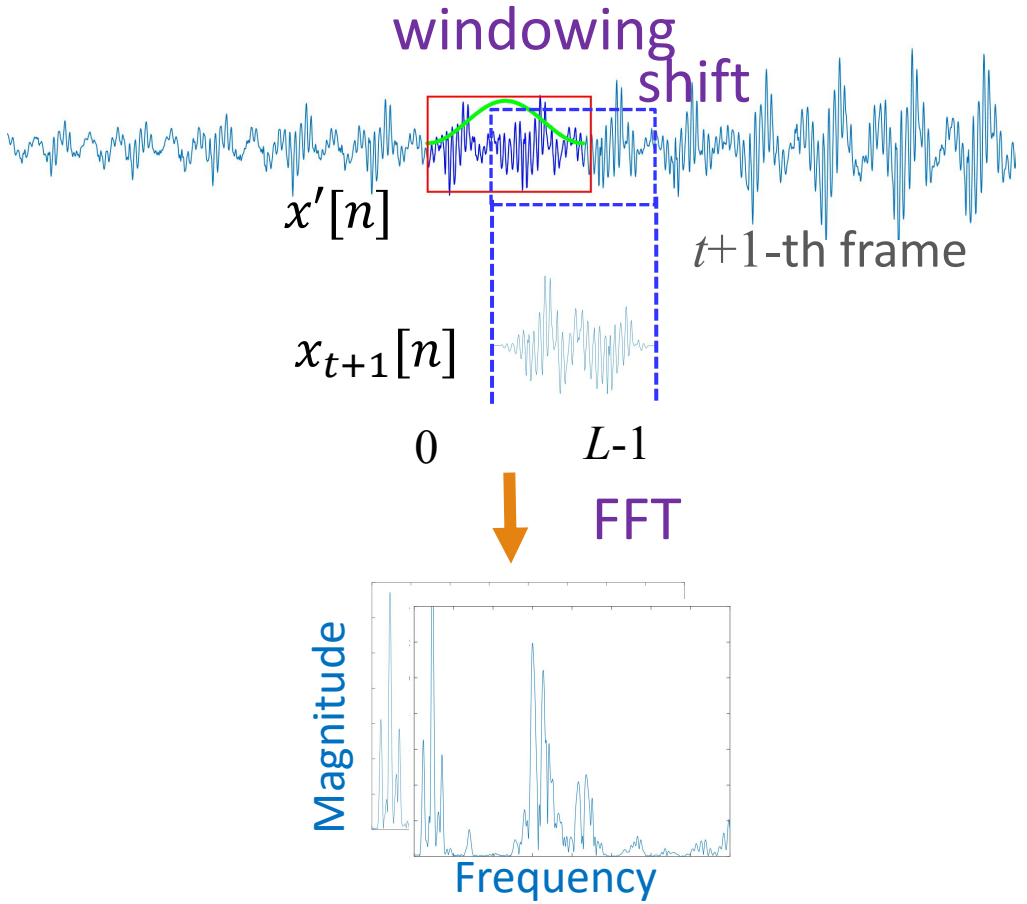
Windowing and spectral analysis

Window the signal $x'[n]$ into frames $x_t[n]$ and apply Fourier Transform to each segment.

- Short frame width: wide-band, high time resolution, low frequency resolution
- Long frame width: narrow-band, low time resolution, high frequency resolution

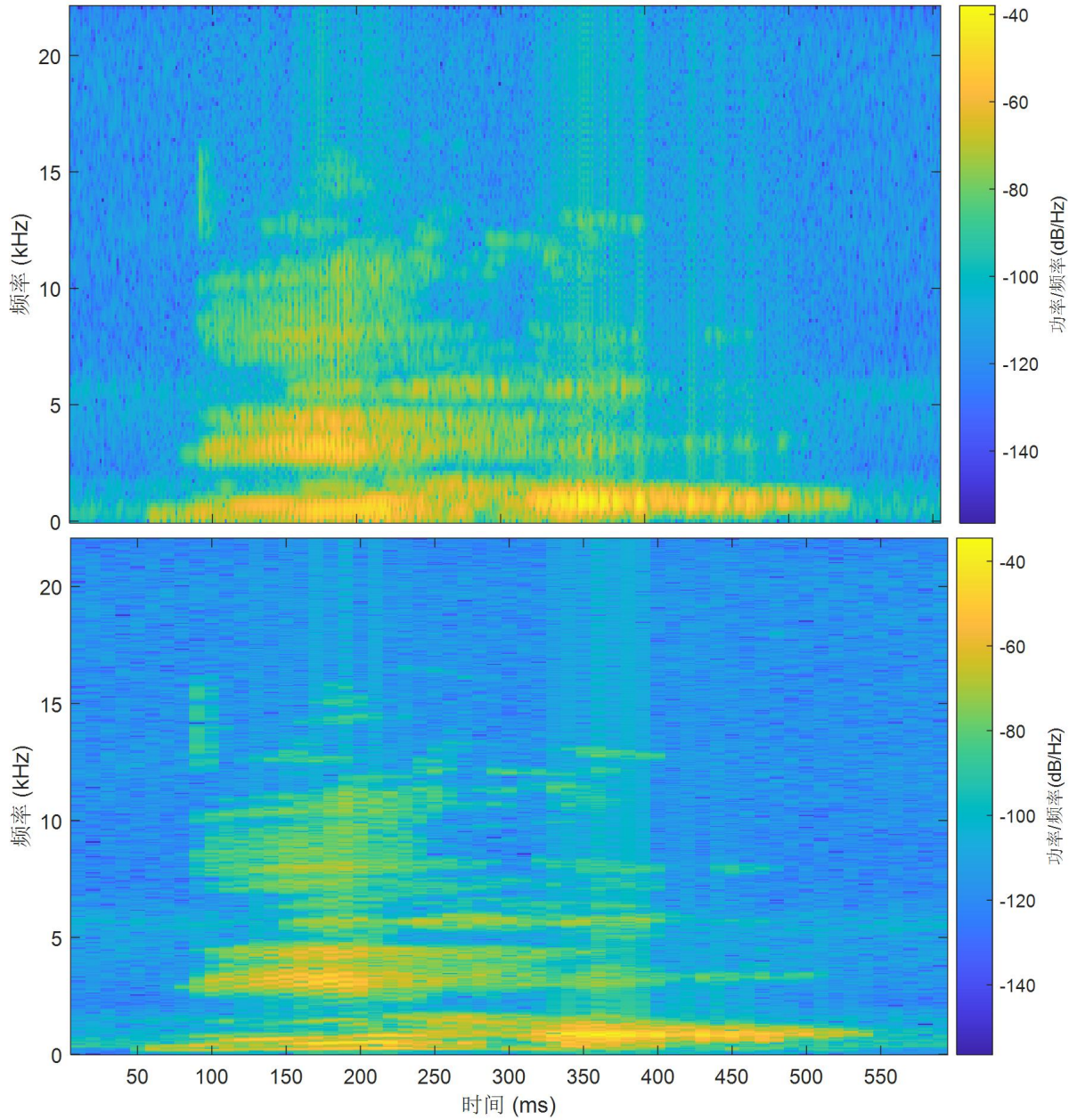
For ASR:

- frame width ~ 25ms
- frame shift ~ 10ms



Short−time power spectrum $|X_{t+1}[k]|^2$

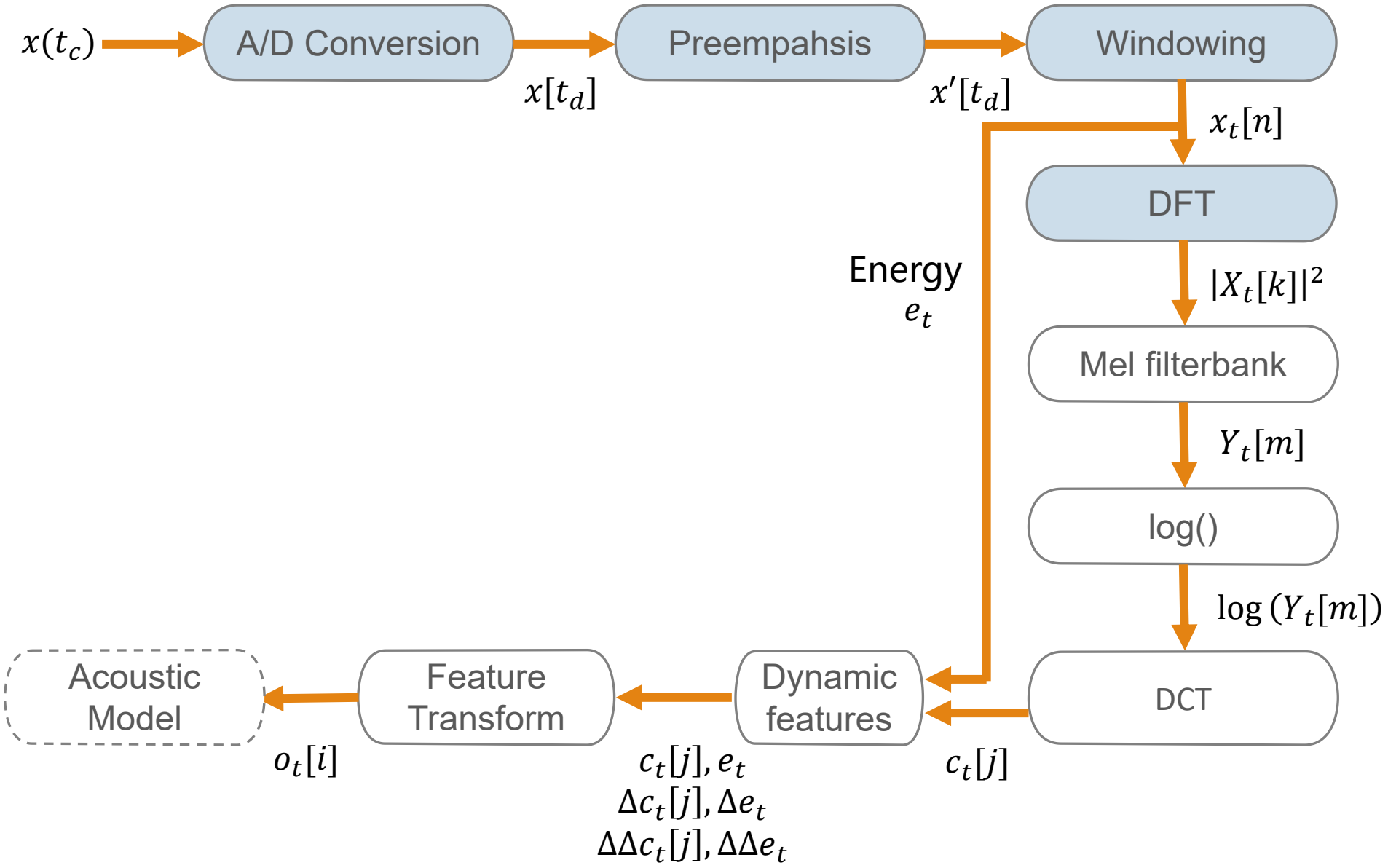
Wide-band and narrow-band spectrograms



Wide band spectrogram
Window width = 2ms

Narrow band spectrogram
Window width = 20ms

MFCC-based front end for ASR



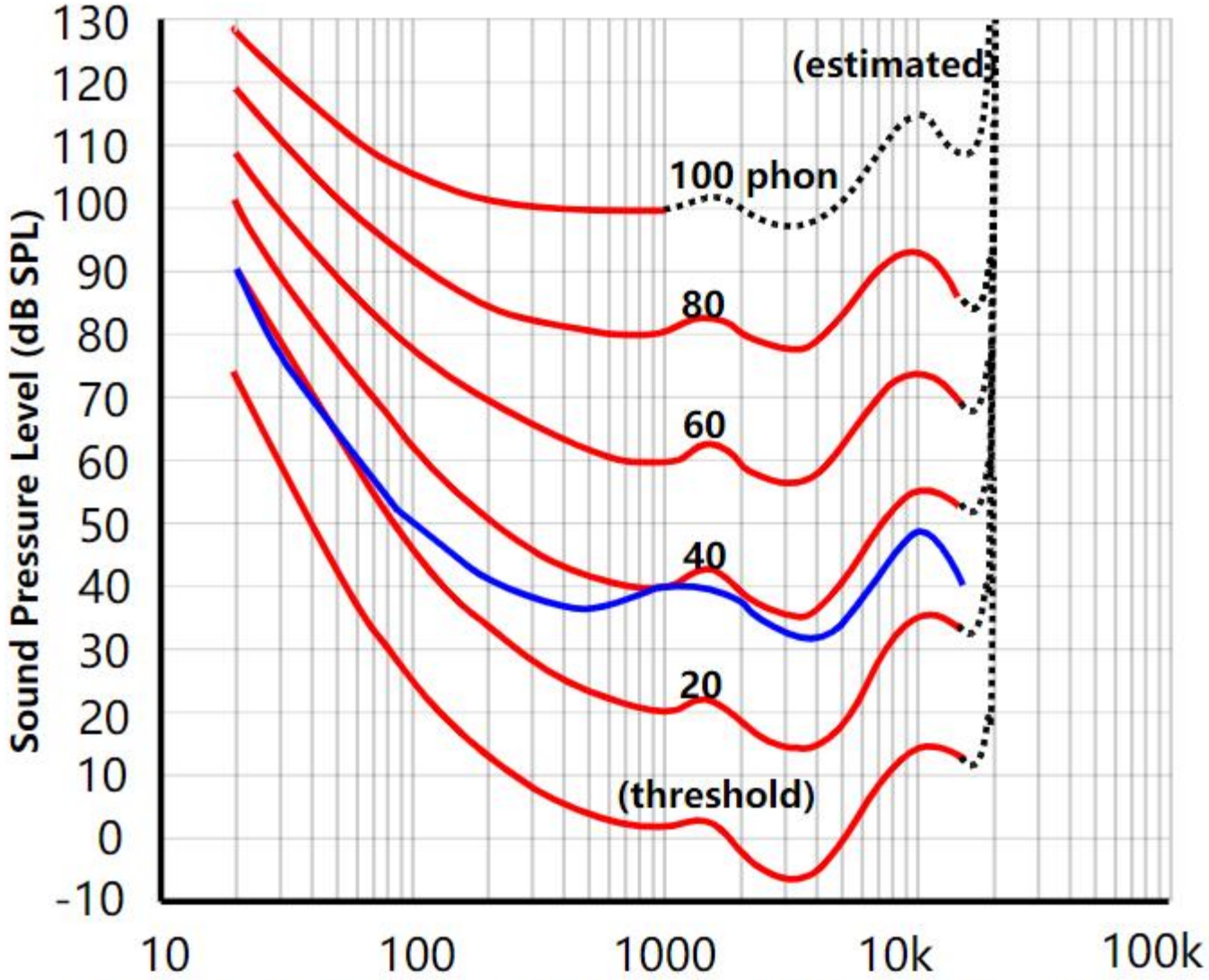
Human hearing

Technical terms

- equal-loudness contours
- Masking
- auditory filters (critical-band filters)
- critical bandwidth

Physical quality	**Perceptual quality**
Intensity	Loudness
Fundamental frequency	Pitch
Spectral shape	Timbre
Onset/offset time	Timing
Phase difference in binaural hearing	Location

Equal loudness contour



Nonlinear frequency scaling

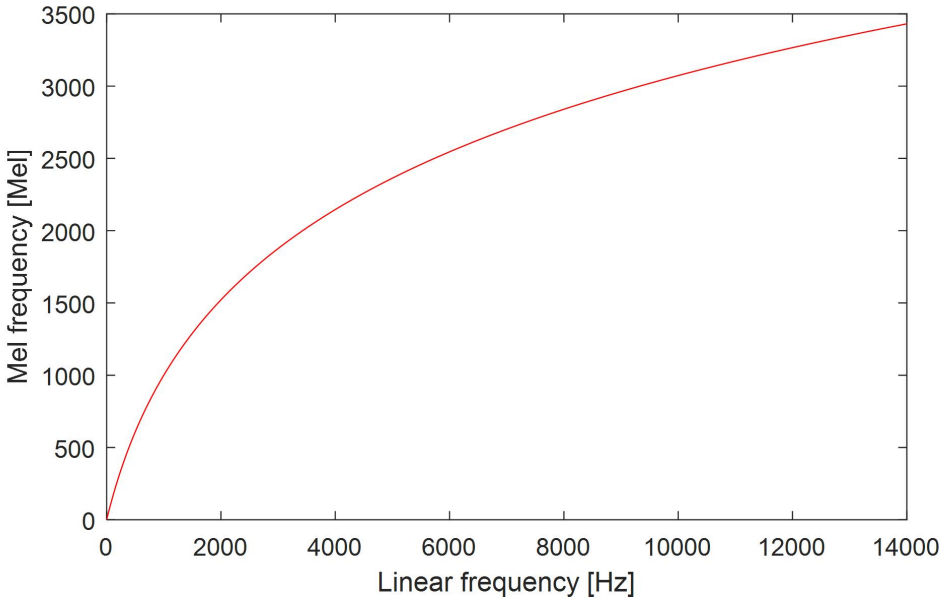
Human hearing is less sensitive to higher frequencies -- thus human perception of frequency is nonlinear

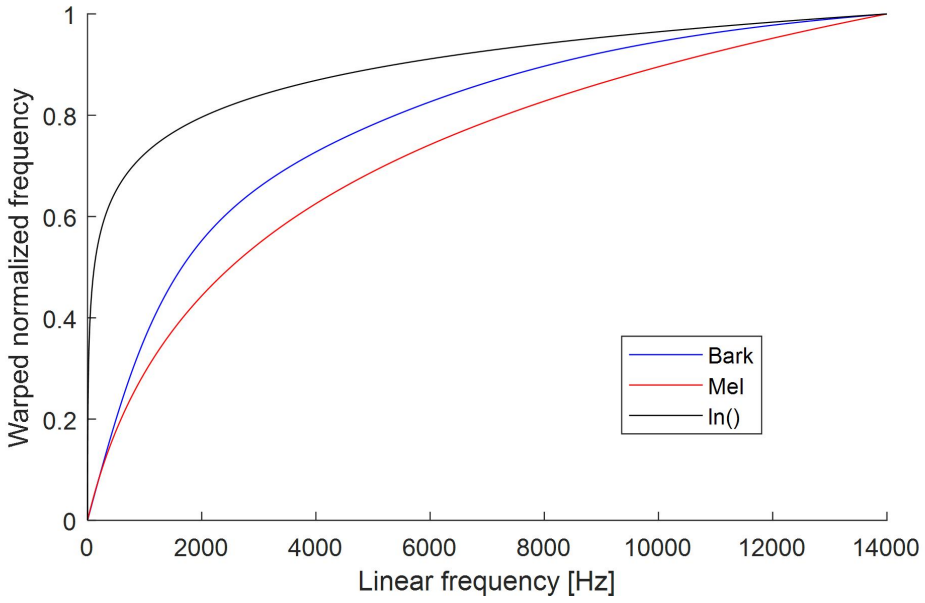
Mel scale

$$f_{mel}(f) = 1127 \ln\left(1 + \frac{f}{700}\right)$$

Bark scale

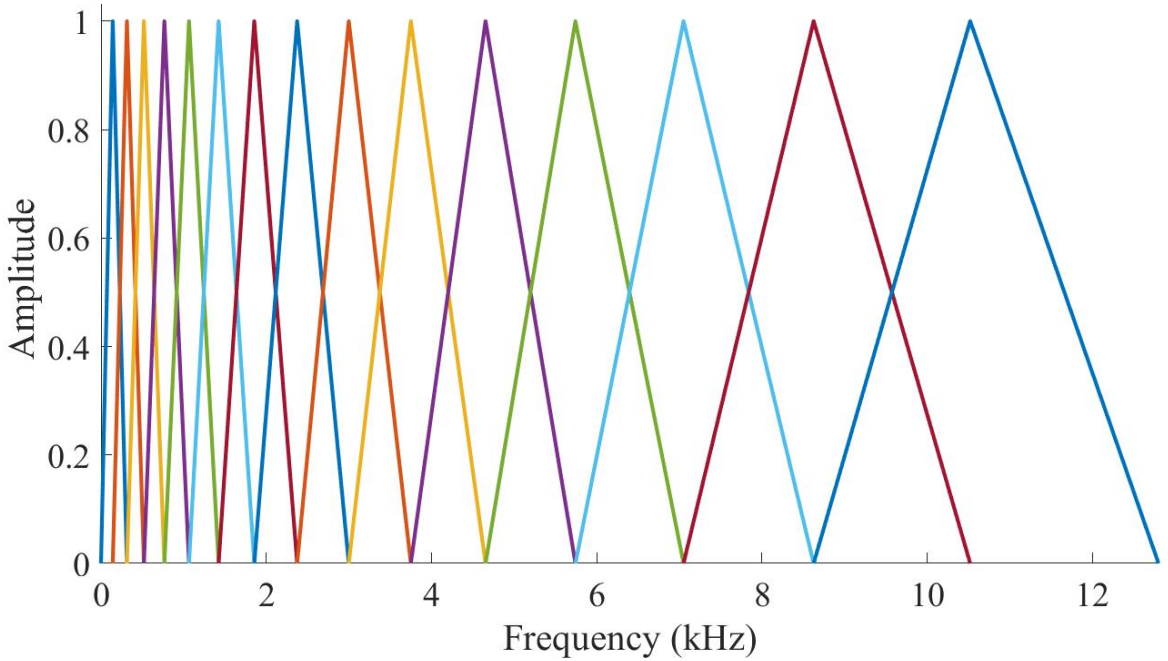
$$f_{bark}(f) = 13 \arctan(0.00076 f) + 3.5 \arctan\left(\left(\frac{f}{7500}\right)^2\right)$$





Mel-Filter Bank

Each filter in the filter bank is triangular having a response of 1 at the center frequency and decrease linearly towards 0 till it reaches the center frequencies of the two adjacent filters where the response is 0



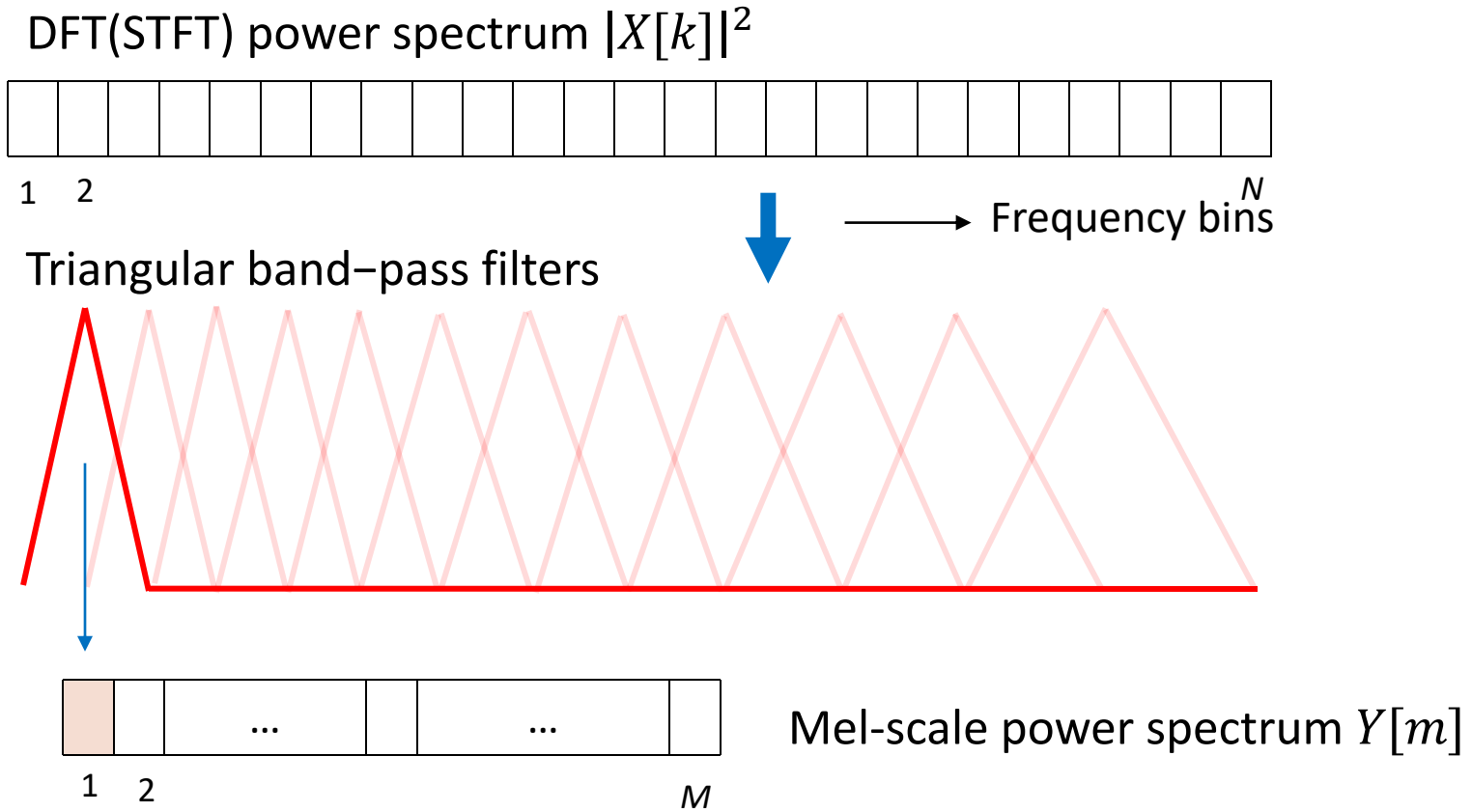
$$H_m[k] = \begin{cases} 0, & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) < k < f(m) \\ 1, & k = f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, & f(m) < k < f(m+1) \\ 0, & k > f(m+1) \end{cases}$$

Mel-Filter Bank

Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum

Each filter collects energy from a number of frequency bands in the DFT

Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz

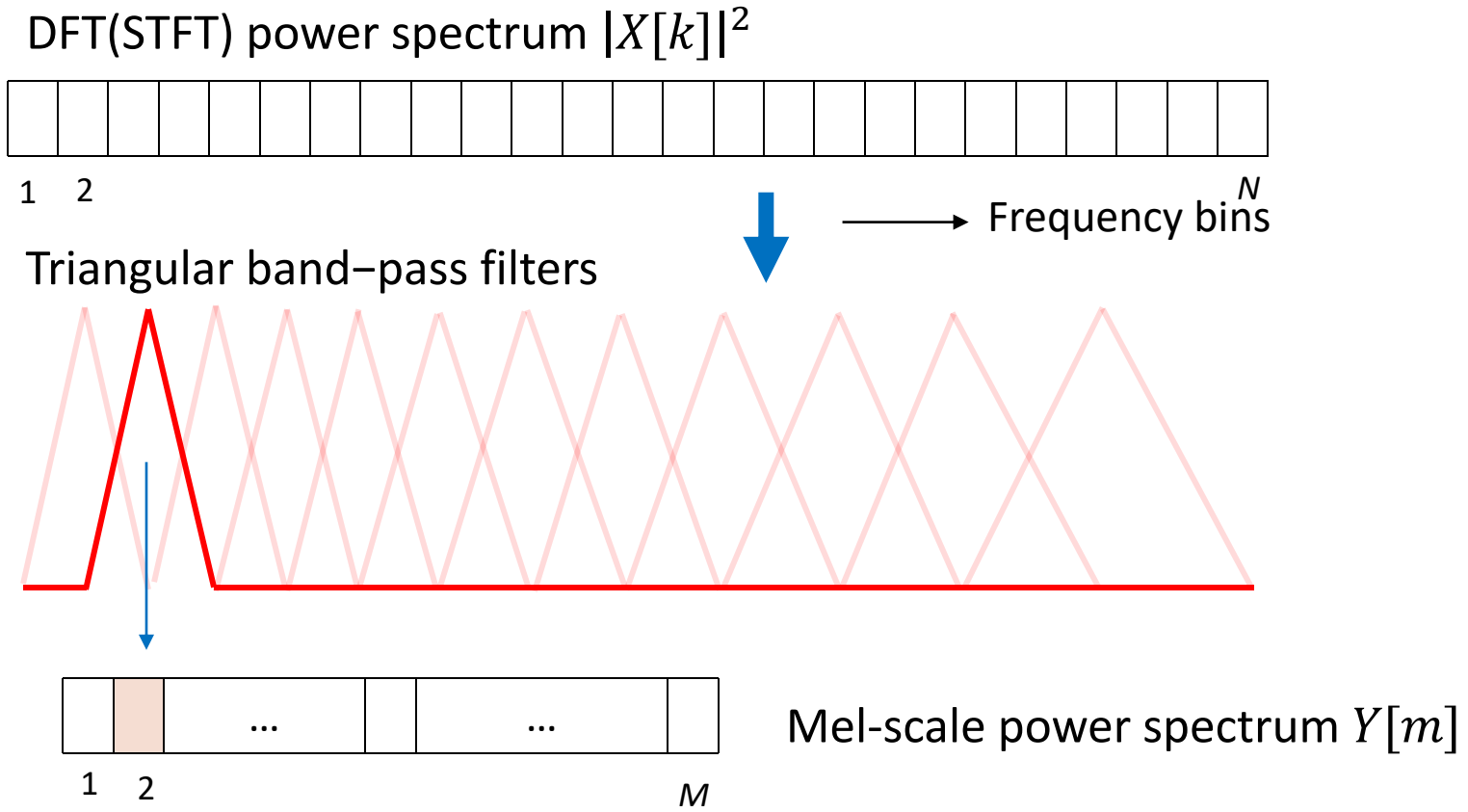


Mel-Filter Bank

Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum

Each filter collects energy from a number of frequency bands in the DFT

Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz

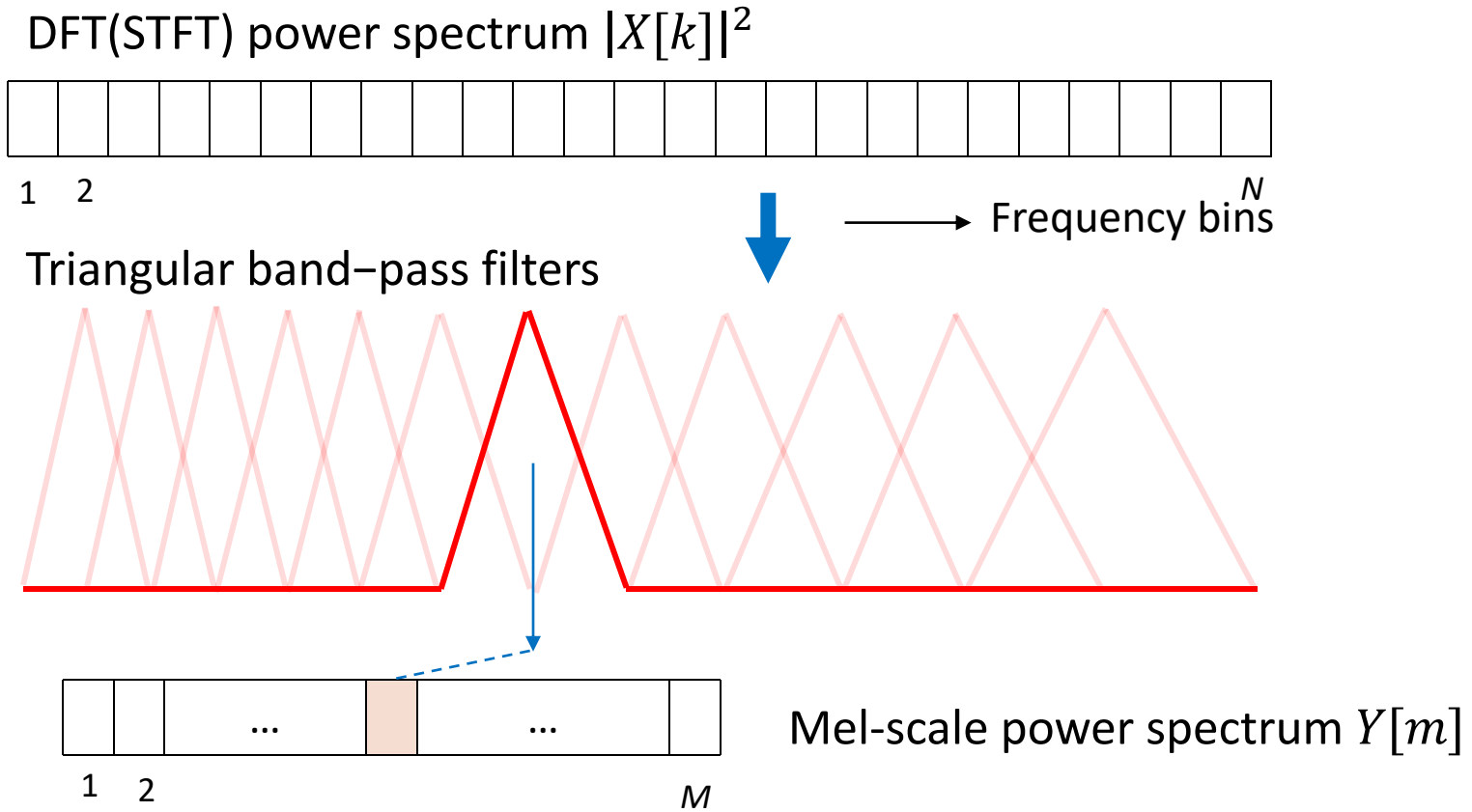


Mel-Filter Bank

Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum

Each filter collects energy from a number of frequency bands in the DFT

Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz

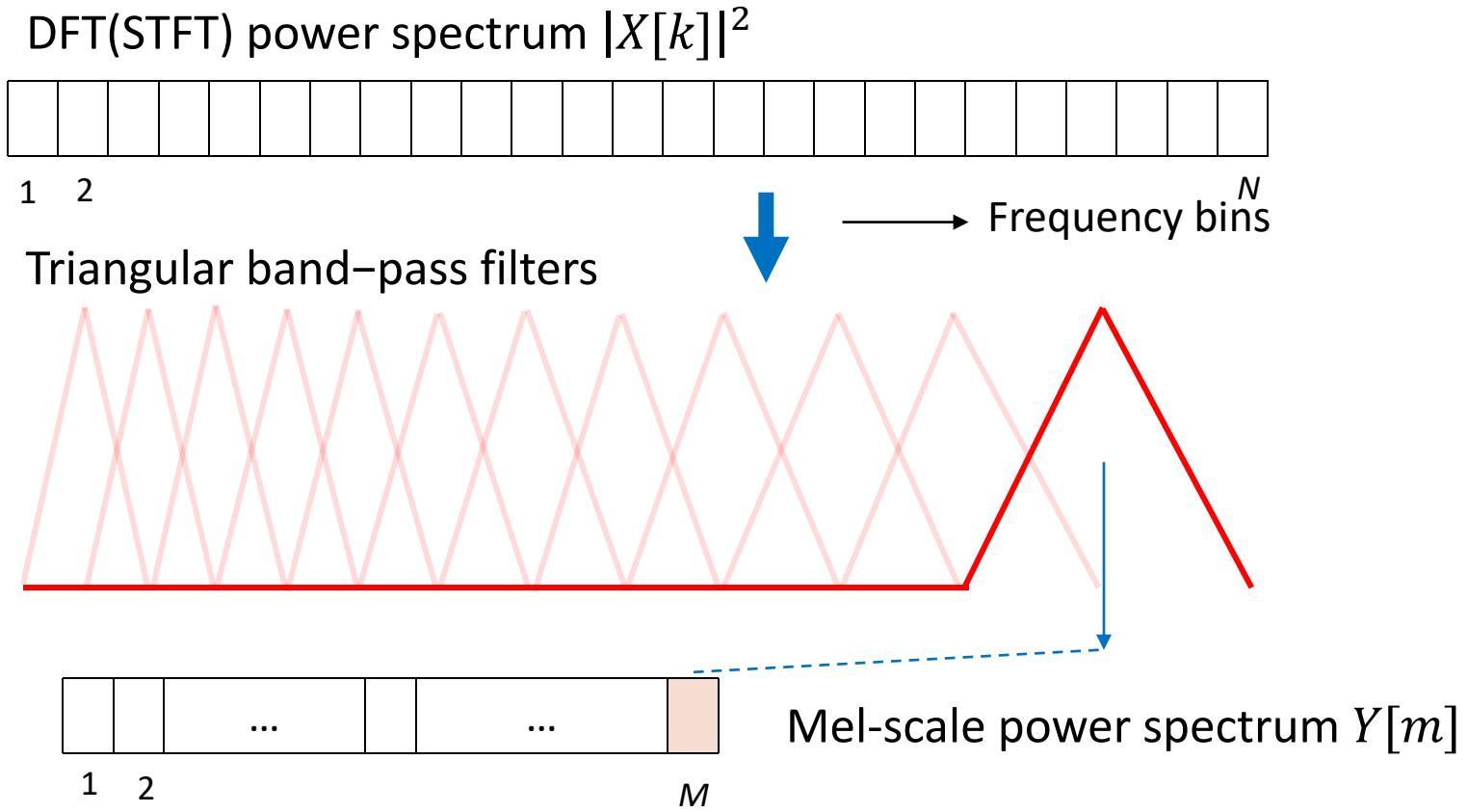


Mel-Filter Bank

Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum

Each filter collects energy from a number of frequency bands in the DFT

Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz

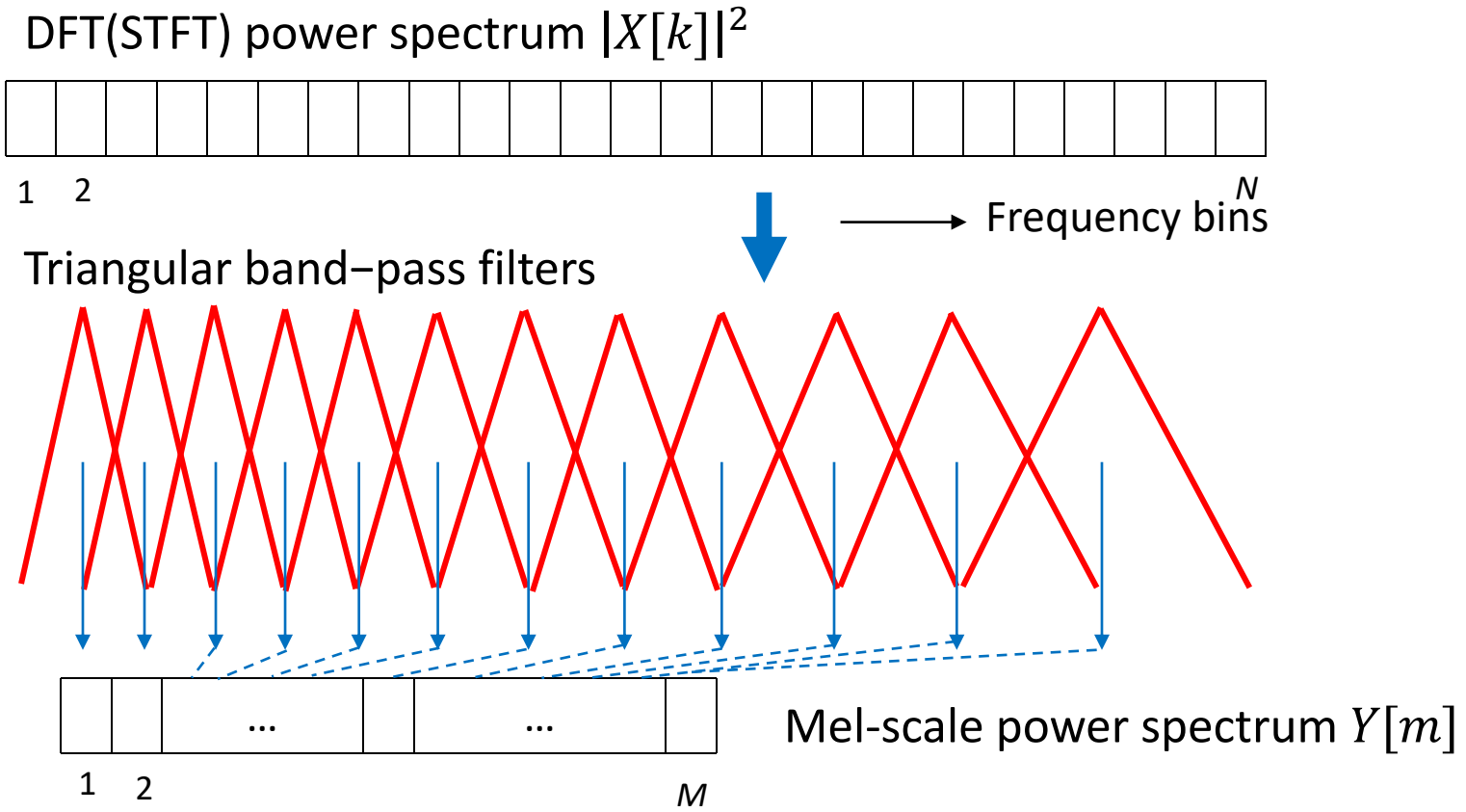


Mel-Filter Bank

Apply a mel-scale filter bank to DFT power spectrum to obtain mel-scale power spectrum

Each filter collects energy from a number of frequency bands in the DFT

Linearly spaced < 1000 Hz, logarithmically spaced > 1000 Hz



An example of Mel-Filter Bank computation

The example will use 10 filterbanks because it is easier to display, in reality you would use 26-40 filterbanks.

- Choose a lower and upper frequency: 300Hz~8000Hz (Suppose sample rate is 16KHz)
- Using Eq. 1, convert the upper and lower frequencies to Mels. In our case 300Hz is 401.25 Mels and 8000Hz is 2834.99 Mels.

$$m = f_{mel}(f) = 1127 \ln\left(1 + \frac{f}{700}\right) \qquad (1)$$

- Because we will do 10 filterbanks, for which we need 12 points. This means we need 10 additional points spaced linearly between 401.25 and 2834.99. This comes out to:

$m(i)$ = 401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74, 1949.99, 2171.24, 2392.49, 2613.74, 2834.99

- Now use Eq. 2 to convert these back to Hertz:

$$f = f_{mel}^{-1}(m) = 700(e^{\frac{m}{1127}} - 1) \qquad (2)$$

$h(i)$ = 300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33,3261.62, 4122.63, 5170.76, 6446.70, 8000

An example of Mel-Filter Bank computation

The example will use 10 filterbanks because it is easier to display, in reality you would use 26-40 filterbanks.

- Suppose DFT has 512 bins. Round those frequencies to the nearest FFT bin.

$$f(i) = floor((nfft + 1)\frac{h(i)}{F_s}) \quad (3)$$

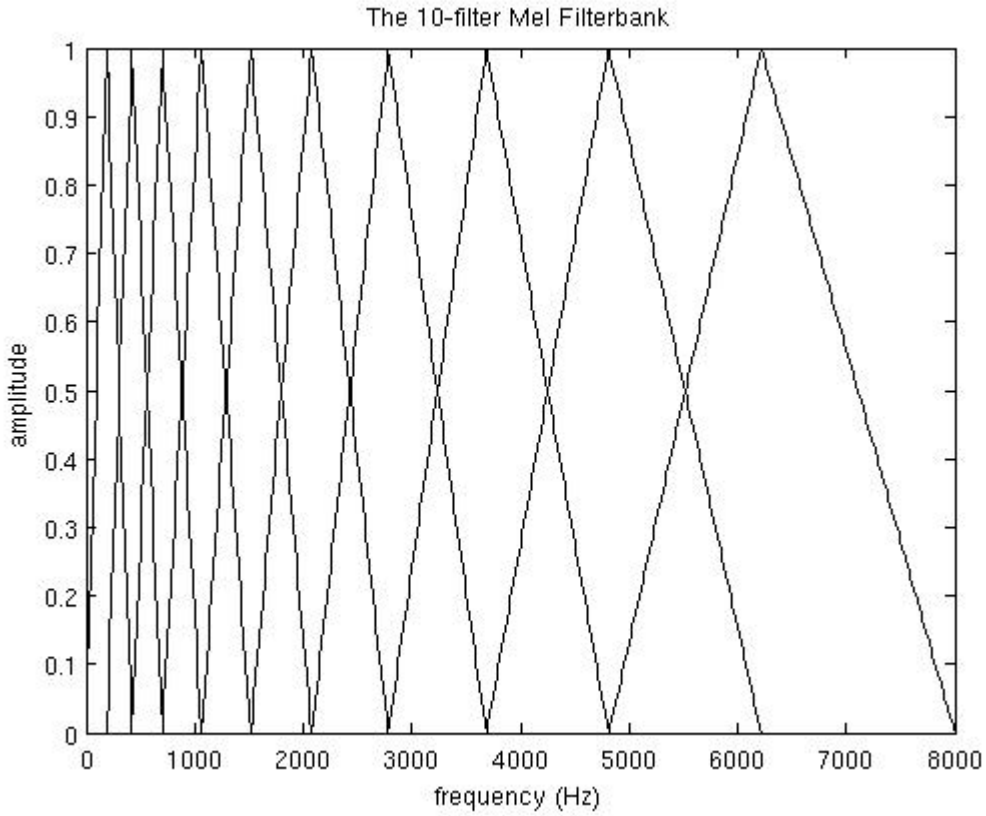
$f(i) = 9, 16, 25, 35, 47, 63, 81, 104, 132, 165, 206, 256$

- Now we create our filterbanks.

$$H_m[k] = \begin{cases} 0, & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & f(m-1) \le k < f(m) \\ 1, & k = f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, & f(m) < k \le f(m+1) \\ 0, & k > f(m+1) \end{cases}$$

An example of Mel-Filter Bank computation

A Mel-filterbank containing 10 filters. This filterbank starts at 0Hz and ends at 8000Hz. This is a guide only, the worked example above starts at 300Hz.



Mel-Filter Bank (Cont.)

$$Y_t[m] = \sum_{k=1}^{N} H_m[k]|X_t[k]|^2$$

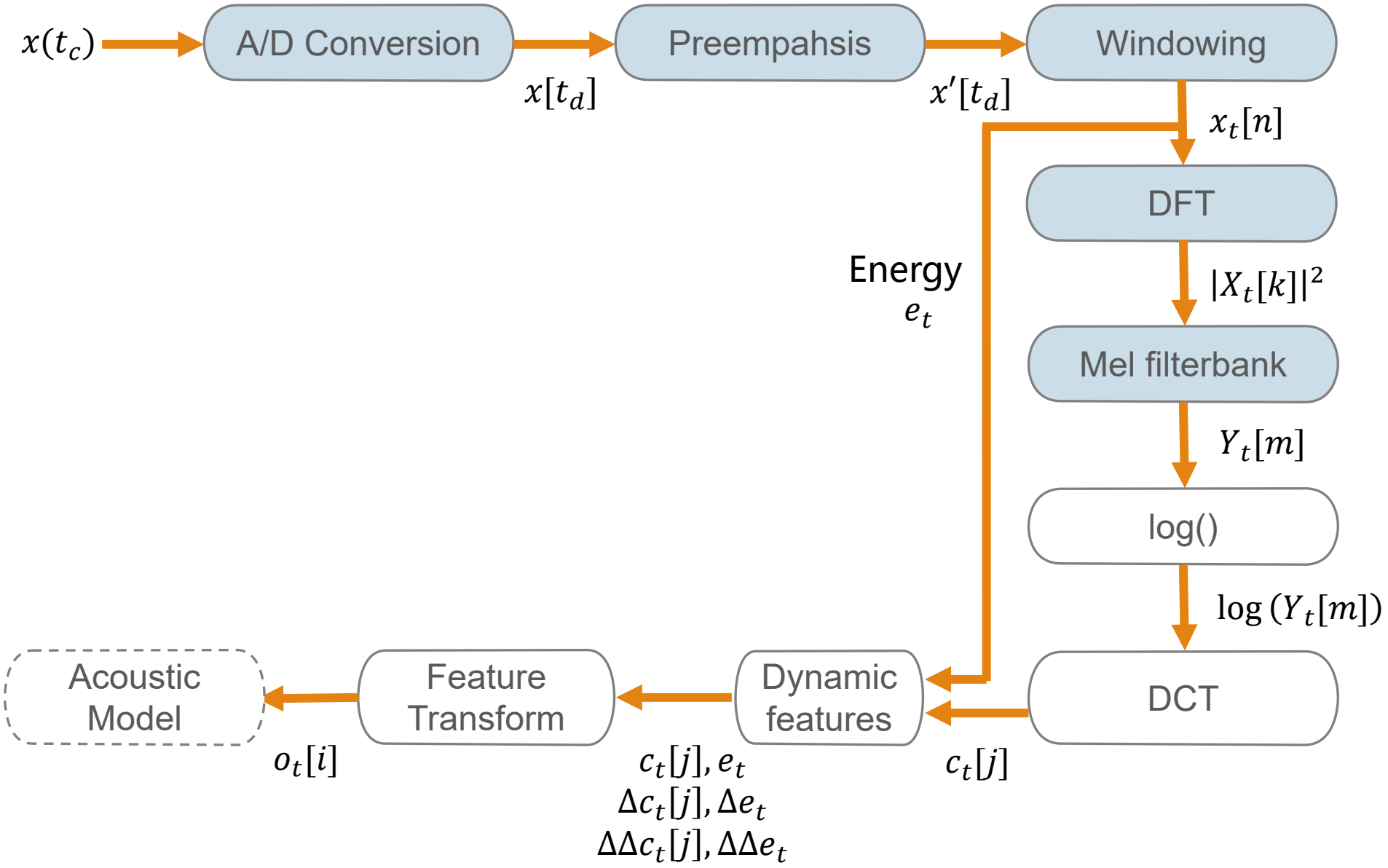
where k: DFT bin number $(1,\ldots,N)$
m: mel-filter bank number $(1,\ldots,M)$

How many number of mel-filter channels?

≈ 20 for GMM-HMM based ASR

$20\sim40$ for DNN-HMM based ASR

MFCC-based front end for ASR



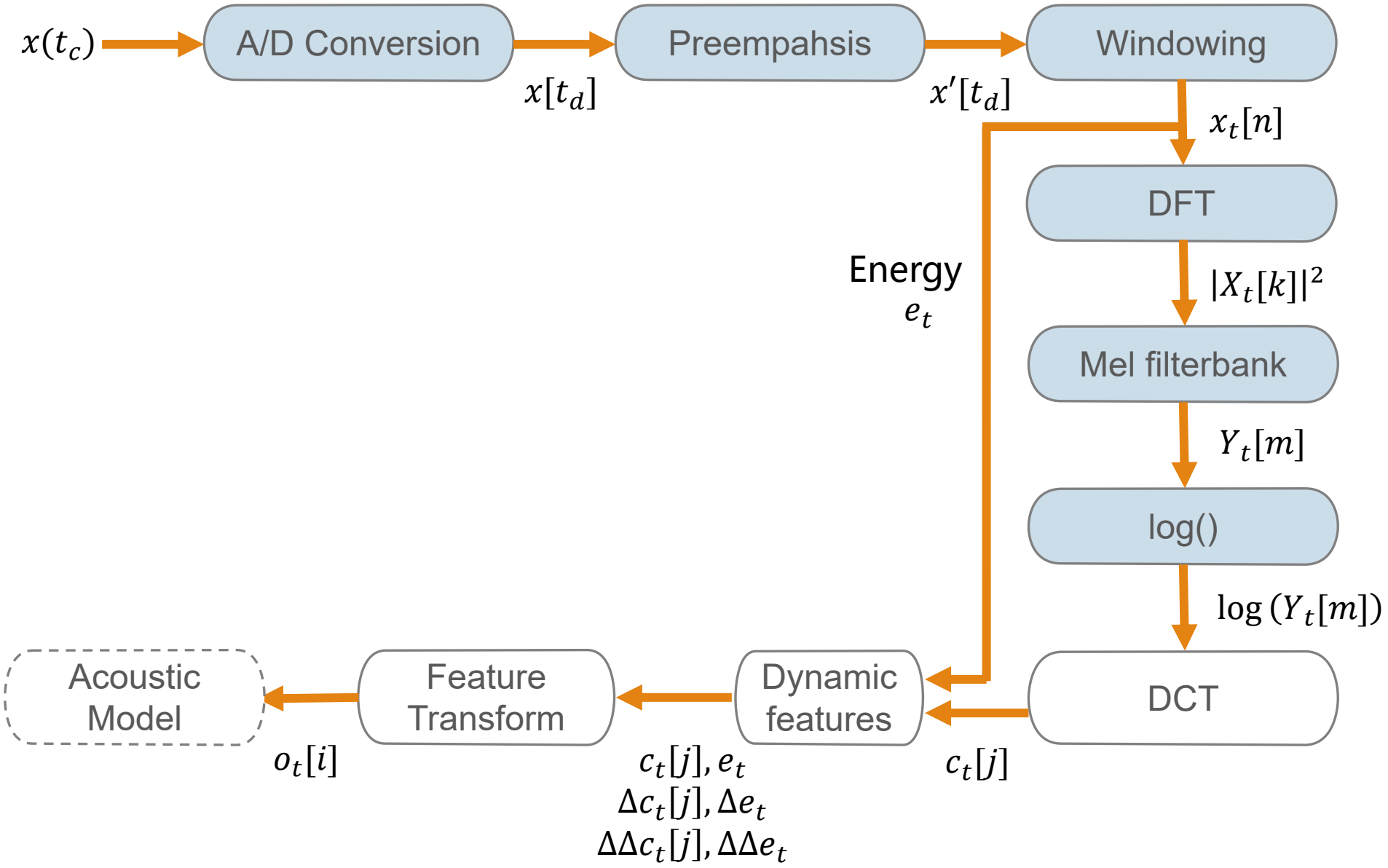
Log Mel Power Spectrum

Compute the log magnitude squared of each mel-filter bank output: $\log Y_t[m]$

- Taking the log compresses the dynamic range
- Human sensitivity to signal energy is logarithmic -- i.e. humans are less sensitive to small changes in energy at high energy than small changes at low energy
- Log makes features less variable to acoustic coupling variations
- Removes phase information -- not important for speech recognition (not everyone agrees with this)

Aka “log mel-filter bank outputs” or “FBANK features”, which are widely used in recent DNN-HMM based ASR systems

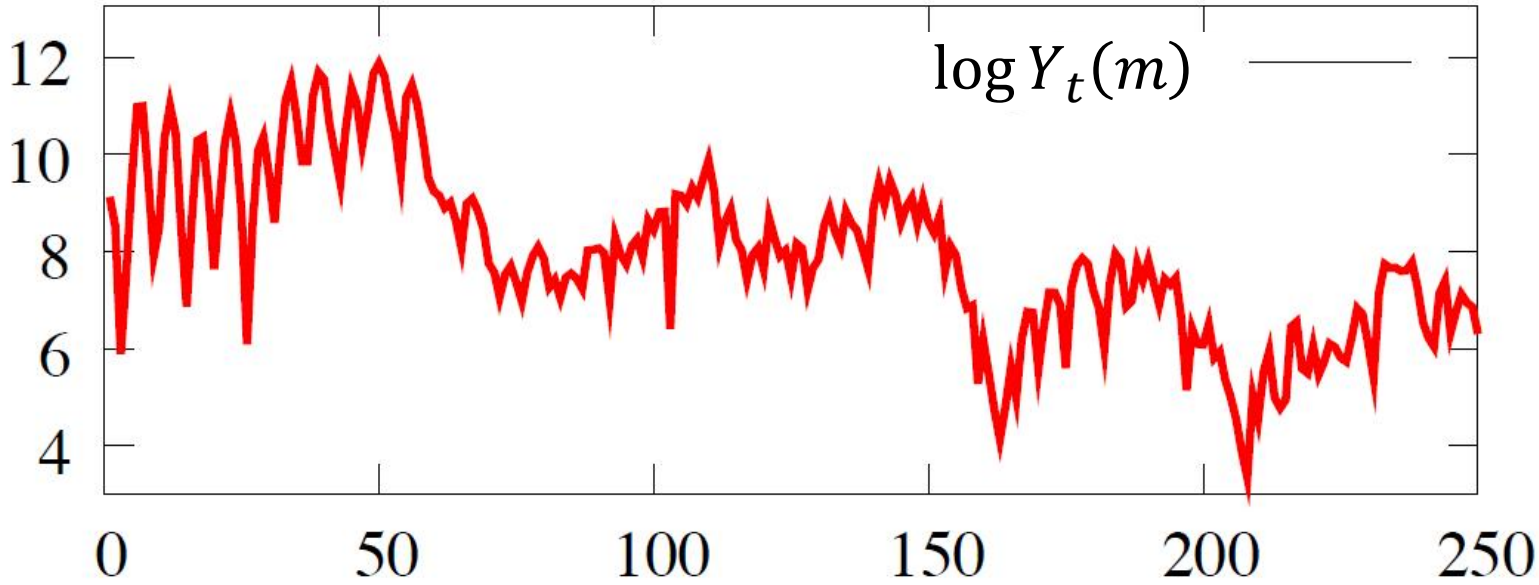
MFCC-based front end for ASR



DFT Spectrum Features for ASR

Equally-spaced frequency bands -- but human hearing less sensitive at higher frequencies (above ~ 1000Hz)

The estimated power spectrum contains harmonics of F_0, which makes it difficult to estimate the envelope of the spectrum



Frequency bins of STFT are highly correlated each other, i.e. power spectrum representation is highly redundant

Cepstral Analysis

Source-Filter model of speech production

- **Source**: Vocal cord vibrations create a glottal source waveform
- **Filter**: Source waveform is passed through the vocal tract: position of tongue, jaw, etc. give it a particular shape and hence a particular filtering characteristic

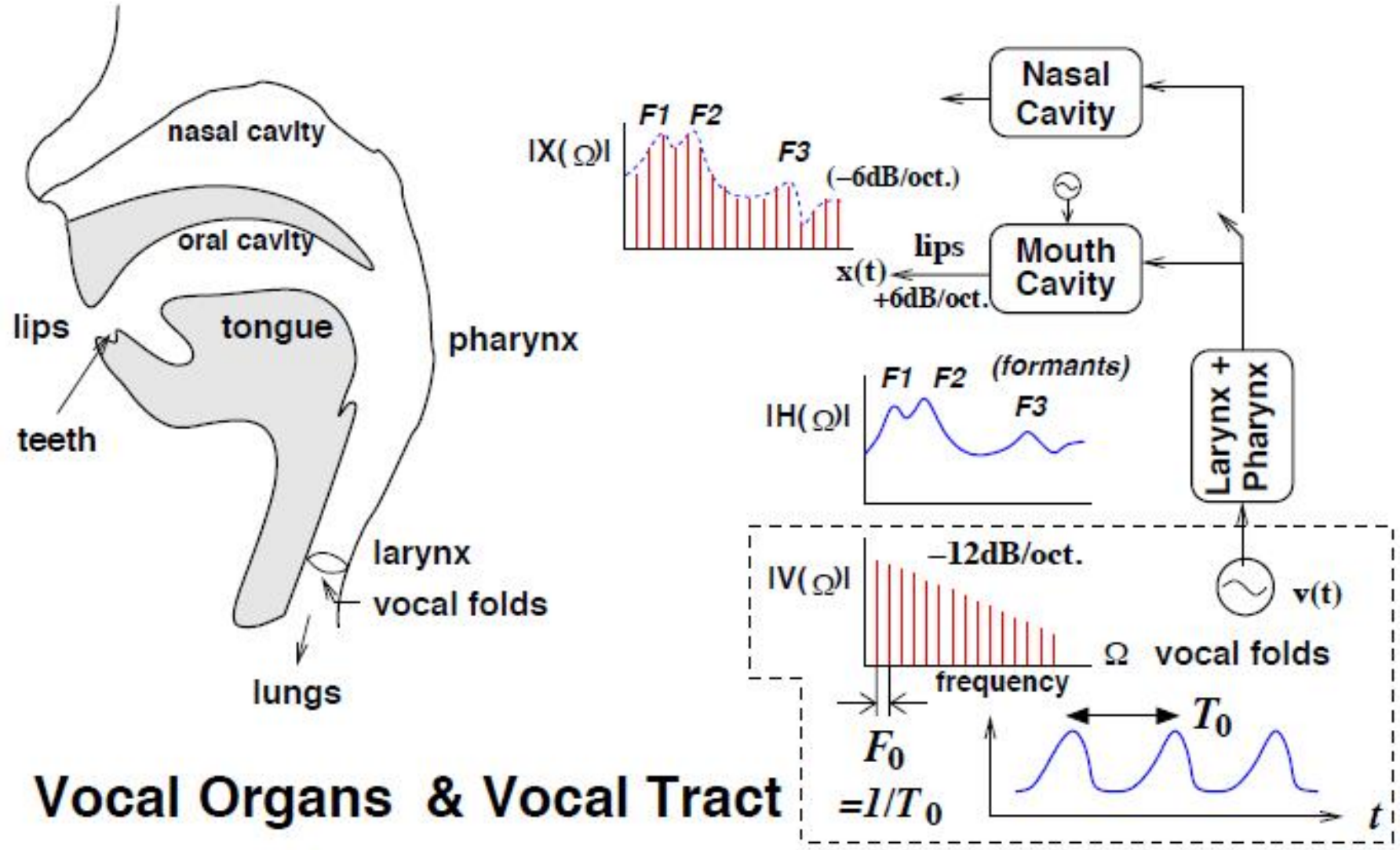
Source characteristics (F_0, dynamics of glottal pulse) do not help to discriminate between phones

The filter specifies the position of the articulators

... and hence is directly related to phone discrimination

Cepstral analysis enables us to separate source and filter

Speech production model

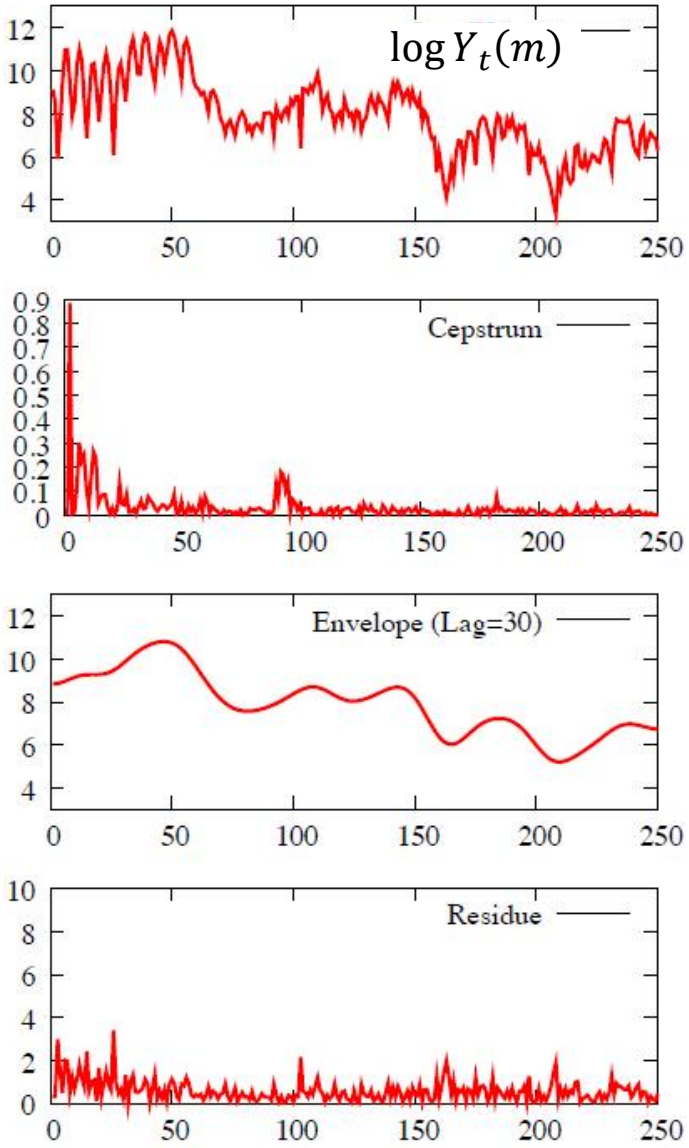


Vocal Organs & Vocal Tract

(F_0 : fundamental frequency)

Cepstral Analysis

Split power spectrum into spectral envelope and F_0 harmonics.



Log spectrum (freq domain)

↓ IDFT

Cepstrum (time domain) (quefrency)

↓ Liftering to get low/high part

(lifter: filter used in cepstral domain)

↓ Fourier Transform

Smoothed log spectrum (freq domain)
[low-part of cepstrum]

+

Fine structure
[high-part of cepstrum]

The Cepstrum

Cepstrum obtained by applying inverse DFT to log magnitude spectrum (may be mel-scaled)

Cepstrum is time-domain (we talk about quefrency)

IDFT:

$$|\mathcal{F}^{-1}\{\log(Y_t[m])\}|^2$$

$$|\mathcal{F}\{\log(Y_t[m])\}|^2$$

The cepstrum is also sometimes called the *spectrum of a spectrum*

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

The Cepstrum

Cepstrum obtained by applying inverse DFT to log magnitude spectrum (may be mel-scaled)

Cepstrum is time-domain (we talk about quefrency)

IDFT:

$$|\mathcal{F}^{-1}\{\log(Y_t[m])\}|^2$$

$$\downarrow$$

$$|\mathcal{F}\{\log(Y_t[m])\}|^2 \quad \rightarrow \quad X[k] = \sum_{m=0}^{M-1} \log Y_t[m]\, e^{-j\frac{2\pi}{N}km}$$

DCT:

$$c_t[j] = \sum_{m=0}^{M-1} \log(Y_t[m]) \cos\left(\left(m + \frac{1}{2}\right)\frac{j\pi}{M}\right), \quad j = 0, \ldots, C-1$$

C is the number of MFCCs

The Cepstrum

Cepstrum obtained by applying inverse DFT to log magnitude spectrum (may be mel-scaled)

Cepstrum is time-domain (we talk about quefrency)

IDFT:

$$|\mathcal{F}\{\log(Y_t[m])\}|^2 \longrightarrow X[k] = \sum_{m=0}^{M-1} \log Y_t[m]\, e^{-j\frac{2\pi}{N}km}$$

Since log power spectrum is real and symmetric the inverse DFT is equivalent to a discrete cosine transform (DCT)

$$c_t[j] = \sum_{m=0}^{M-1} \log(Y_t[m]) \cos\left(\left(m+\frac{1}{2}\right)\frac{j\pi}{M}\right), \quad j = 0, \ldots, C-1$$

C is the number of MFCCs

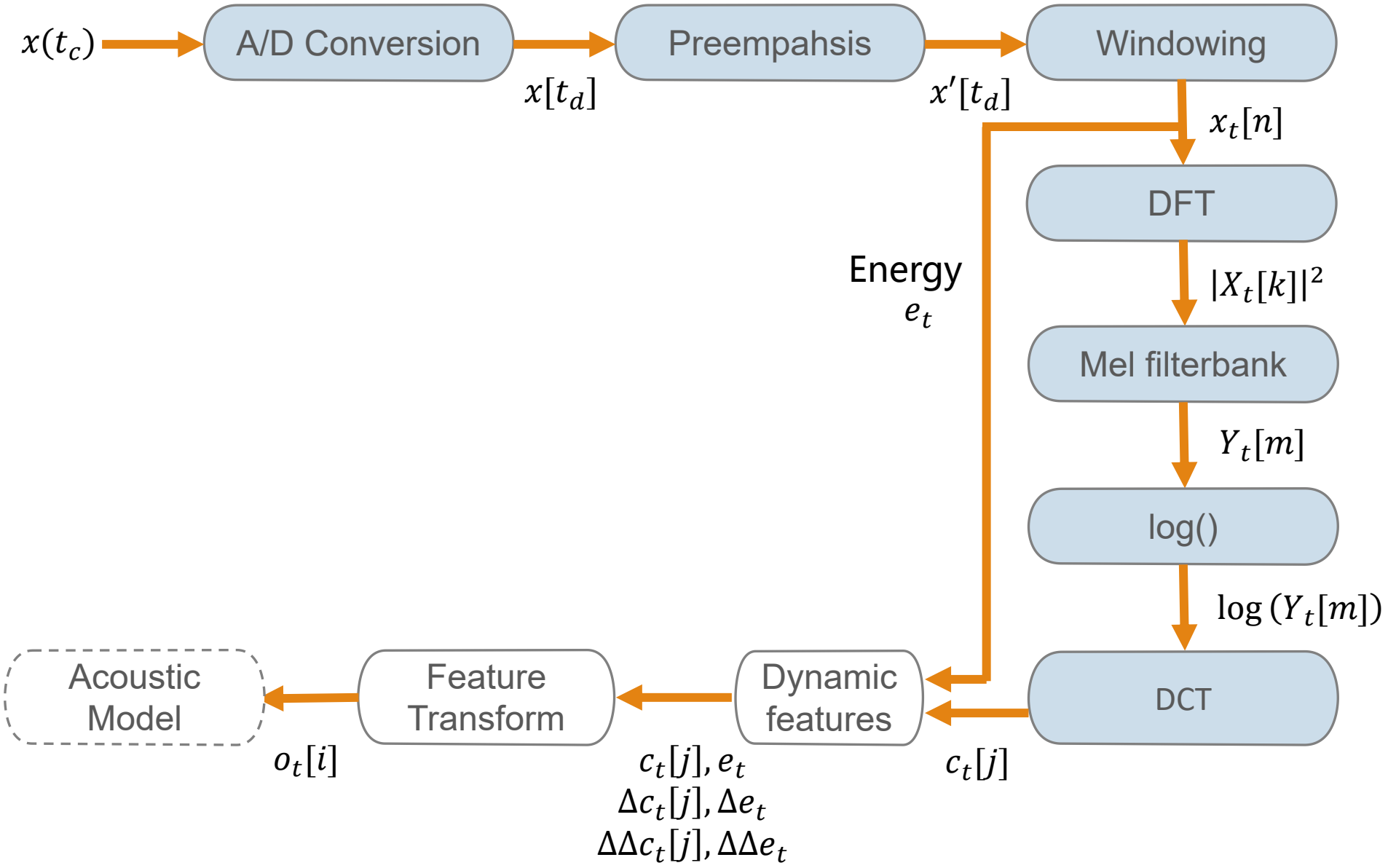
MFCCs

Smoothed spectrum: transform to cepstral domain, truncate, transform back to spectral domain

Mel-frequency cepstral coecients (MFCCs): use the cepstral coecients directly

- Widely used as acoustic features in HMM-based ASR
- First 12 MFCCs are often used as the feature vector (removes F0 information)
- Less correlated than spectral features – easier to model than spectral features
- Very compact representation – 12 features describe a 20ms frame of data
- For standard HMM-based systems, MFCCs result in better ASR performance than filter bank or spectrogram features
- MFCCs are not robust against noise

MFCC-based front end for ASR



Dynamic features

Speech is not constant frame-to-frame, so we can add features to do with how the cepstral coefficients change over time

Δ, Δ^2 are delta features (dynamic features / time derivatives)

Simple calculation of delta features Δ_t at time t for cepstral feature $\boldsymbol{c}_t$:

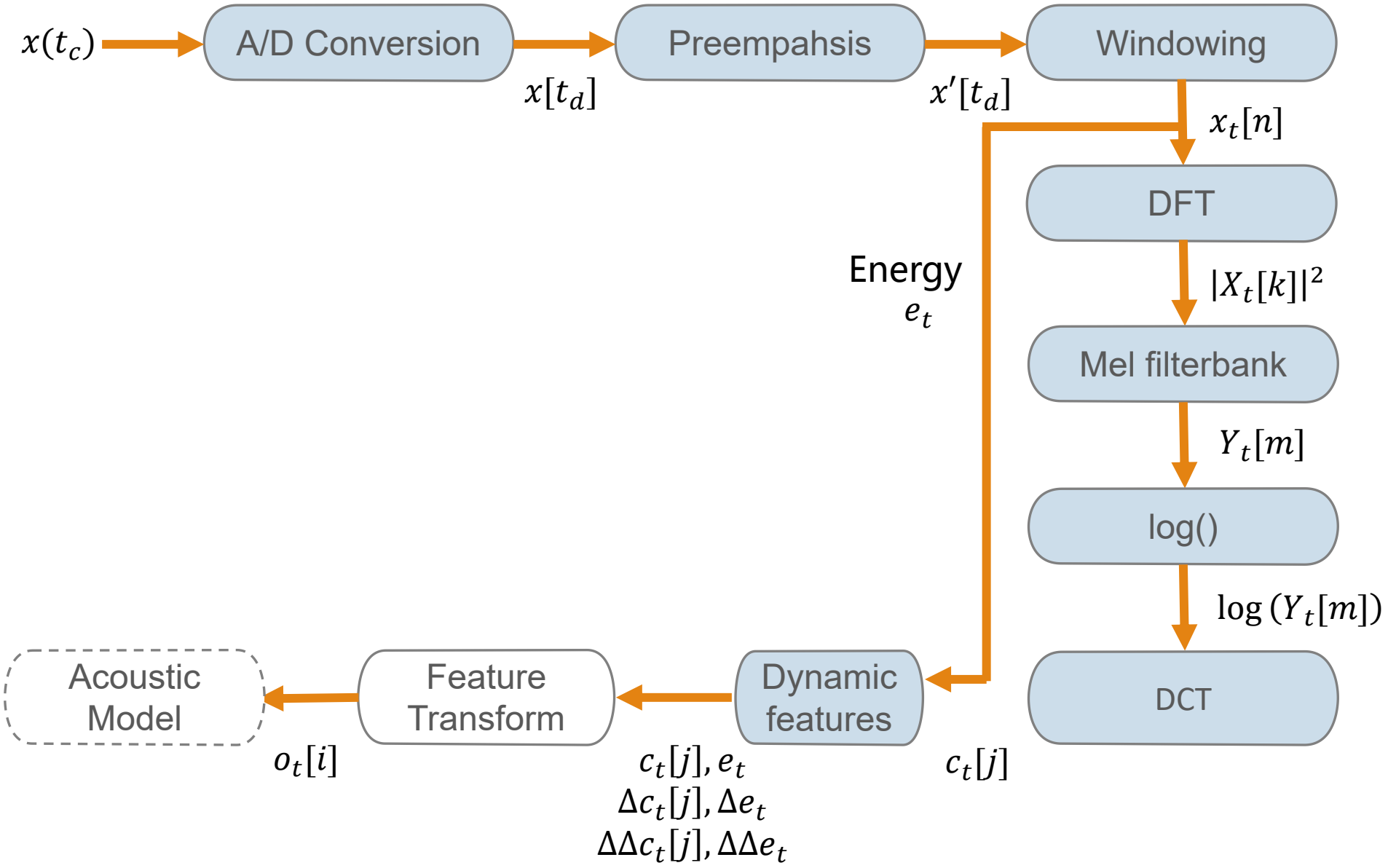
$$\Delta_t = \frac{\boldsymbol{c}_{t+1} - \boldsymbol{c}_{t-1}}{2}$$

More sophisticated approach estimates the temporal derivative by using regression to estimate the slope (typically using 4 frames each side)

"Standard" ASR features (for GMM-based systems) are 39 dimensions:

- 12 MFCCs, and energy
- 12 ΔMFCCs, Δenergy
- 12 Δ^2MFCCs, Δ^2energy

MFCC-based front end for ASR



Feature normalization

Basic Idea: Transform the features to reduce mismatch between training and test

Cepstral Mean Normalization (CMN): subtract the average feature value from each feature, so each feature has a mean value of 0. makes features robust to some linear filtering of the signal (channel variation)

Cepstral Variance Normalisation (CVN): Divide feature vector by standard deviation of feature vectors, so each feature vector element has a variance of 1

Cepstral mean and variance normalization, CMN/CVN:

$$\hat{y}_t[j] = \frac{y_t[j] - \mu(\boldsymbol{y}[j])}{\sigma(\boldsymbol{y}[j])}$$

Compute mean and variance statistics over longest available segments with the same speaker/channel

Real time normalization: compute a moving average

An example of MFCC computation

We start with a speech signal $s(n)$, we'll assume sampled at 16kHz.

1. Frame the signal into 20-40 ms frames. 25ms is standard. Frame step is usually 10ms.

- 0.025*16000 = 400 samples
- Denote the i-th frame $s_i(n)$, where n ranges over 1-400

2. Take the Discrete Fourier Transform of the frame $s_i(n)$

- $X_i[k] = \mathrm{DFT}\big(s_i(n)h(n)\big), 1 \leq k \leq K$, $h(n)$ is a hamming window with 400 length, K is the length of DFT (e.g. 512)
- The power spectrum $P_i(k) = |X_i(k)|^2$
- We would generally perform a 512 point FFT and keep only the first 257 coefficents

An example of MFCC computation

3. Compute the Mel-Filter Bank

- This is a set of 20-40 (26 is standard) triangular filters that are applied to $P_i(k)$
- Multiply each filterbank with $P_i(k)$, then add up the coefficents
- 26 numbers are left indicating of how much energy was in each filterbank.

4. Take the log of each of the 26 energies

5. Take the Discrete Cosine Transform (DCT) of the 26 log filterbank energies to give 26 cepstral coefficents $\boldsymbol{y}_t$. For ASR, only the lower 12 of the 26 coefficients are kept.

6. Compute energy e_t for $\boldsymbol{y}_t$

7. Compute Δ and $\Delta\Delta$ for $\boldsymbol{y}_t$ and energies Δe_t, $\Delta\Delta e_t$

Acoustic features in state-of-the-art ASR systems

See Tables 1, 2, and 3 in

Jinyu Li, Dong Yu, Jui-Ting Huang, and Yifan Gong, “Improving Wideband Speech Recognition Using Mixed-Bandwidth Training Data In CD-DNN-HMM”,

2012 IEEE Workshop in Spoken Language Technology (SLT2012).

https://doi.org/10.1109/SLT.2012.6424210

Table 1: Comparison of different input features for DNN. All the input features are mean-normalized and with dynamic features. Relative WER reduction in parentheses.

Setup	**WER (%)**
CD-GMM-HMM (MFCC, fMPE+BMMI)	34.66 (**baseline**)
CD-DNN-HMM (MFCC)	31.63 (-8.7%)
CD-DNN-HMM (24 log filter-banks)	30.11 (-13.1%)
CD-DNN-HMM (29 log filter-banks)	30.11 (-13.1%)
CD-DNN-HMM (40 log filter-banks)	29.86 (-13.8%)
CD-DNN-HMM (256 log FFT bins)	32.26 (-6.9%)

Table 2: Comparison of DNNs with and without dynamic features. All the input features are mean normalized.

CD-DNN-HMM (40 log filter-banks)	WER (%)
static+Δ+ΔΔ (11-frame)	29.86
static only (11-frame)	31.11
static only (19-frame)	30.48

Table 3: Comparison of features with and without mean normalization. Dynamic features are used.

CD-DNN-HMM (29 log filter banks)	WER (%)
With mean normalization	30.11
Without mean normalization	29.96

Summary: Speech Signal Analysis for ASR

Good characteristics of ASR features

FBANK features

- Short-time DFT analysis
- Mel-filter bank
- Log magnitude squared
- Widely used for DNN ASR ($M \approx 40$)

MFCCs-mel frequency cepstral coefficients

- FBANK features
- Inverse DFT (DCT)
- Use first few (12) coecients
- Widely used for GMM-HMM ASR

Delta features (dynamic features)

39-dimension feature vector (for GMM-HMM ASR): MFCC-12 + energy; + Deltas; +

References

J&M: Daniel Jurafsky and James H. Martin (2008). Speech and Language Processing, Pearson Education (2nd edition).

Taylor: Paul Taylor (2009). Text-to-Speech Synthesis, Cambridge University Press.

Hynek Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” The Journal of the Acoustical Society of America, Vol.87, No.4, pp.1737-1752, 1980.

Assignment 1

Extract acoustic features (MFCC) for a segment of speech. Comment your codes properly.

Processing steps include:

- Pre-emphasis
- Windowing
- STFT
- Mel-filter bank
- Log()
- DCT
- Dynamic feature extraction
- Feature transformation