SH Dr. Soufiane Hourri

drsoufiane.com

MASTER'S PROGRAMME · SECOND YEAR

Speech Processing

Session 01 · Speech as a signal: from sound to numbers

Dr. Soufiane Hourri, PhD, HDR

Associate Professor & AI Consultant

École Supérieure de Technologie de Safi · Cadi Ayyad University

2026-2027

Today's session

01 What speech processing is

02 How speech is produced

03 From sound to numbers

04 The signal in time

05 A first look at frequency

06 Lab: your first audio analysis in Python

TODAY'S FOCUS

To a computer, a voice is a long list of numbers.

Today we learn where those numbers come from and what they already tell us.

No signal processing background is needed.

Learning objectives

By the end of this session, you will be able to

1. Name the main speech processing tasks
2. Describe how voiced and unvoiced sounds are produced
3. Define frequency, period, amplitude and the decibel
4. Explain sampling, quantization and aliasing; compute a file size
5. Compute frame energy and ZCR in Python; read a spectrogram

PREREQUISITES

- Python basics: lists, functions, loops
- School maths: sine, powers of 2, log10 (reminded today)
- No signal processing needed

Planned progression of the module

01

Foundations

Speech as a signal: sound, sampling, frames (today)

02

Frequency analysis

Fourier transform and the spectrogram

03

Speech features

Mel scale, filter banks, MFCCs

04

Recognition

From classical models to neural ASR

05

Speakers and modern models

Speaker recognition, self-supervised models, synthesis

01

CHAPTER

What speech processing is

What can a computer do with a voice, and what must it know about sound first?

Speech processing: computing with voices

DEFINITION

A computer captures, analyses, transforms or generates human speech, starting from a recording stored as numbers: a signal.

EXAMPLE

You say “Set an alarm for 7.” The phone records your voice, converts it to text (recognition), interprets the request, then answers with a synthetic voice (synthesis).

SPEECH → TEXT recognition (ASR)
TEXT → SPEECH synthesis (TTS)
SPEECH → WHO speaker recognition
SPEECH → SPEECH enhancement, translation

The main speech processing tasks

Task	Input → output	Everyday example
Speech recognition (ASR)	speech → text	Subtitles, dictation, voice search
Speech synthesis (TTS)	text → speech	GPS directions, screen readers
Speaker recognition	speech → identity	Voice login, forensic comparison
Speech enhancement	noisy speech → clean speech	Noise cancellation in video calls
Speech translation	speech → speech or text in another language	Live interpretation, Darija ↔ Arabic
Classification	speech → a label	Language, emotion or keyword detection

Table 1.1. Every task starts from the same raw material: a sampled speech signal.

The common pipeline of a speech system

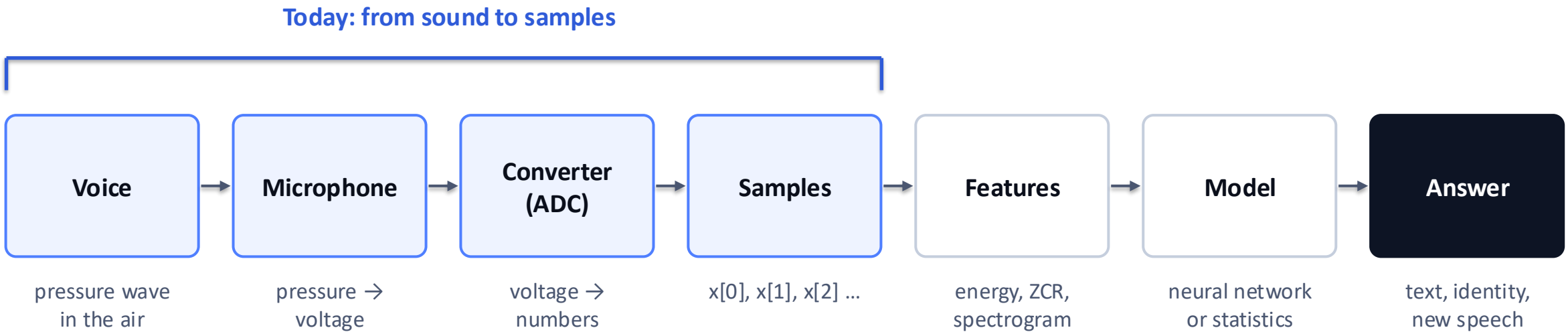


Figure 1.1. From the speaker's voice to the system's answer. Today: the first four boxes and the simplest features.

Speech systems in 2026: large models, same signal

What has changed

- Large neural networks learn from huge amounts of audio
- Whisper (2022): 680 000 hours of transcribed audio
- wav2vec 2.0 and WavLM learn from untranscribed speech
- Open ASR Leaderboard (2026): 86 models, 12 datasets

What has not changed

- Models still take sampled audio, usually at 16 kHz
- They first cut it into frames and build a spectrogram
- A wrong sample rate degrades results silently
- Low-resource languages such as Darija lack data

Knowledge check

QUESTION

A bank wants to check, from the voice alone, that the caller really is the account holder. Which task is this?

A

Speech recognition (ASR)

B

Speaker recognition (verification)

C

Speech synthesis (TTS)

D

Speech enhancement

02

CHAPTER

How speech is produced

Before we measure speech, we need to know what makes it: air, vibration and the shape of the mouth.

Sound is a pressure wave

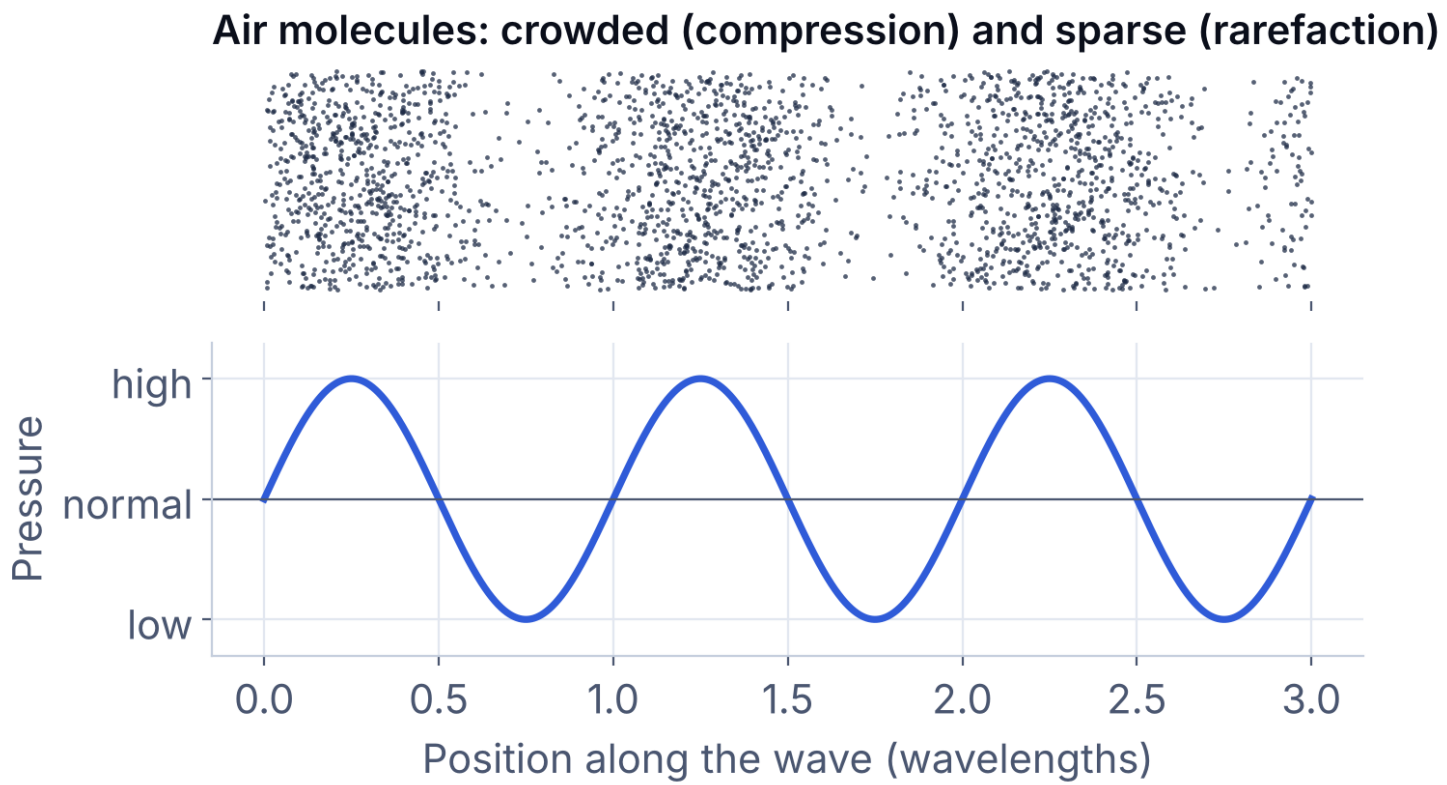
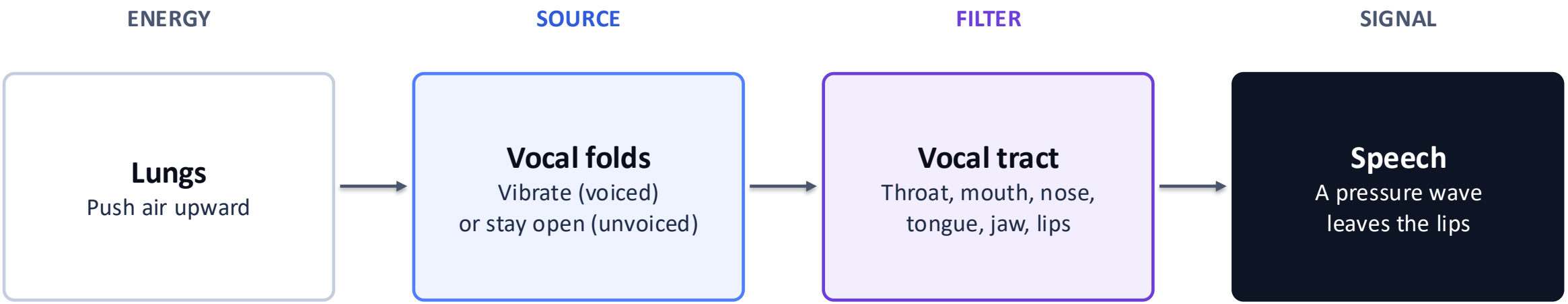


Figure 2.1. Molecule density (top) and the matching pressure (bottom). Illustration generated for this slide.

WHAT TO NOTICE

- A vibrating object pushes and pulls the air around it
- Zones of high and low pressure travel at about 343 m/s (air, 20 °C)
- A microphone, like the eardrum, follows the pressure at one point
- A recording is pressure measured over time

Speech production: a source and a filter



Source: pitch and voicing, **filter:** which sound is said

Figure 2.2. The source-filter model of speech production (after Fant, 1960).

Voiced and unvoiced sounds

Voiced

- The vocal folds vibrate
- The signal repeats regularly: it is periodic and has a pitch
- All vowels: /a/, /i/, /u/
- Many consonants: /m/, /n/, /z/, /v/, /b/
- Hand on the throat: you feel a buzz

Unvoiced

- The vocal folds stay open
- Air rushes through a narrow gap and makes noise
- The signal is noise-like: no regular repetition, no pitch
- Examples: /s/, /f/, /ʃ/ (“sh”), /p/, /t/, /k/
- Hand on the throat: no buzz

Pitch: how fast the vocal folds vibrate

DEFINITION

The fundamental frequency f_0 is the number of vocal fold vibrations per second, measured in hertz (Hz). We hear it as pitch: a high or low voice.

EXAMPLE

The vocal folds open and close 120 times in one second, so $f_0 = 120$ Hz. One vibration lasts $1/120$ s ≈ 8.3 ms.

TYPICAL SPEAKING f_0 (ROUGH RANGES)

Adult men	about 85 to 180 Hz
Adult women	about 165 to 255 Hz
Children	about 250 to 400 Hz

Individual voices vary widely.

Vowels differ by the shape of the vocal tract

- The vocal folds produce the same kind of buzz for every vowel
- Moving the tongue, jaw and lips changes which frequencies the vocal tract reinforces
- These reinforced frequency regions are called formants: F1, F2, F3 …
  - /i/ as in “see”: F1 ≈ 270 Hz, F2 ≈ 2290 Hz
  - /ɑ/ as in “father”: F1 ≈ 730 Hz, F2 ≈ 1090 Hz (adult men, Peterson & Barney, 1952)

KEY TAKEAWAY

Source (f_0): how it is said. Filter (formants): which sound is said.

Knowledge check

QUESTION

You whisper the word “see”. Which statement is true?

A

The /s/ becomes a voiced sound

B

The vowel has no fundamental frequency f_0

C

The formants of the vowel disappear

D

Whispering raises f_0

03

CHAPTER

From sound to numbers

A computer cannot store a continuous wave. It stores measurements taken at regular times, each rounded to a fixed precision.

From a voice to an audio file

01

Pressure wave

The air vibrates in front of the speaker

02

Microphone

Pressure becomes a continuous voltage

03

Sampling

Measure the voltage fs times per second

04

Rounding

Quantization: store each value with b bits

05

File

Store the list of numbers, for example as WAV

Sampling: measuring at regular times

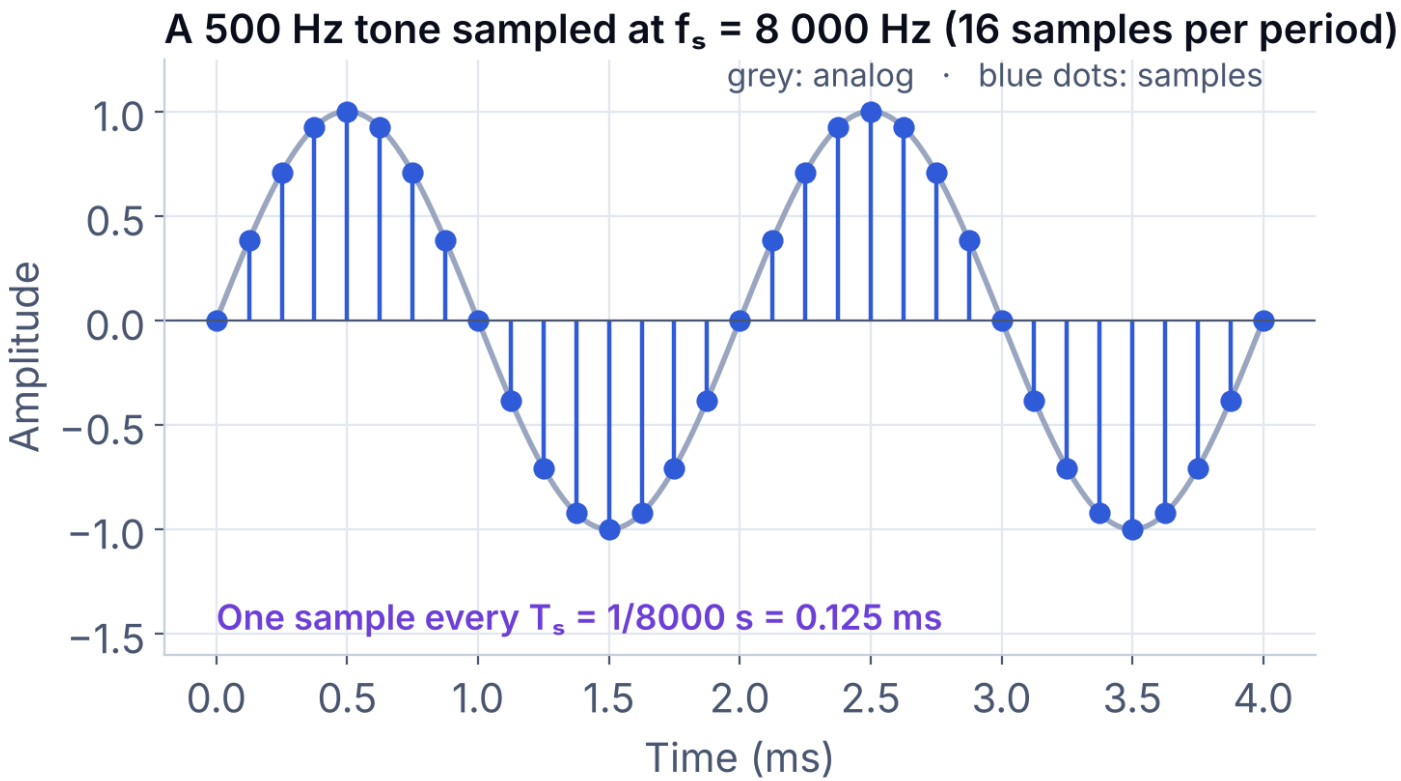


Figure 3.1. A 500 Hz tone sampled at 8 kHz. Generated with NumPy and matplotlib.

WHAT TO NOTICE

- Sample rate fs: samples per second (Hz)
- Sampling period Ts = 1/fs
- At 16 kHz: one sample every 62.5 µs, 16 000 numbers per second
- Between two samples, nothing is stored

The sampling theorem

$$f_s > 2 \cdot f_{max}$$

SYMBOLS

fs : sample rate, in Hz

fmax : highest frequency present in the signal

fs / 2 : the Nyquist frequency, the highest frequency fs can represent

INTUITION

You need more than two samples per period to see a wave go up and down.

Telephone speech stops near 3.4 kHz: 8 kHz is enough. Speech recognition keeps up to 8 kHz: 16 kHz.

Aliasing: too few samples invent a false frequency

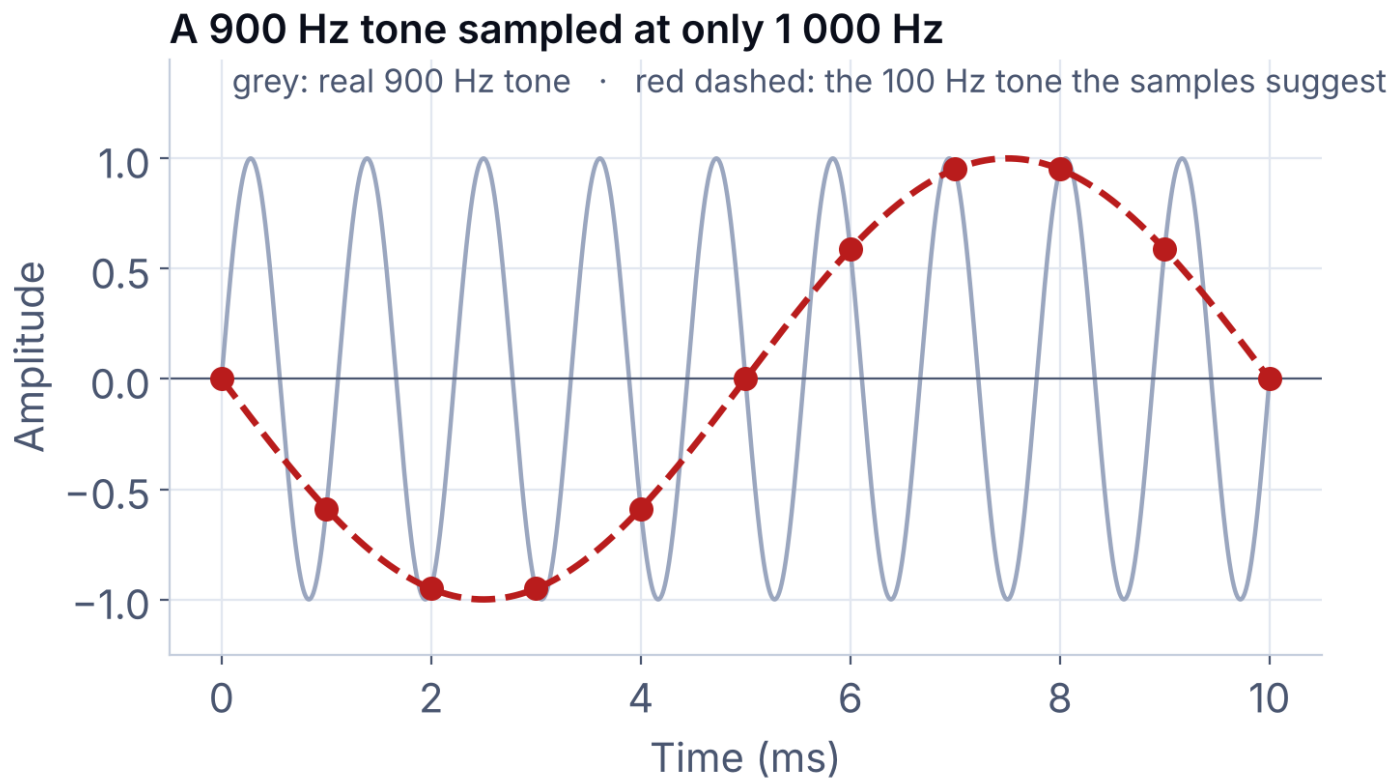


Figure 3.2. A 900 Hz tone sampled at 1 kHz looks like a 100 Hz tone. Generated with NumPy.

WHAT TO NOTICE

- 900 Hz needs fs > 1 800 Hz
- At fs = 1 000 Hz the samples match a 100 Hz tone exactly
- After sampling, the error cannot be undone
- Converters filter out frequencies above fs/2 before sampling

Common sample rates

Use	Sample rate fs	Highest frequency kept (fs / 2)
Telephone (narrowband)	8 kHz	4 kHz
Speech recognition, wideband calls	16 kHz	8 kHz
Audio CD	44.1 kHz	22.05 kHz
Video, professional audio	48 kHz	24 kHz

Table 3.1. Human hearing covers roughly 20 Hz to 20 kHz. Speech models usually expect 16 kHz.

Quantization: rounding each sample

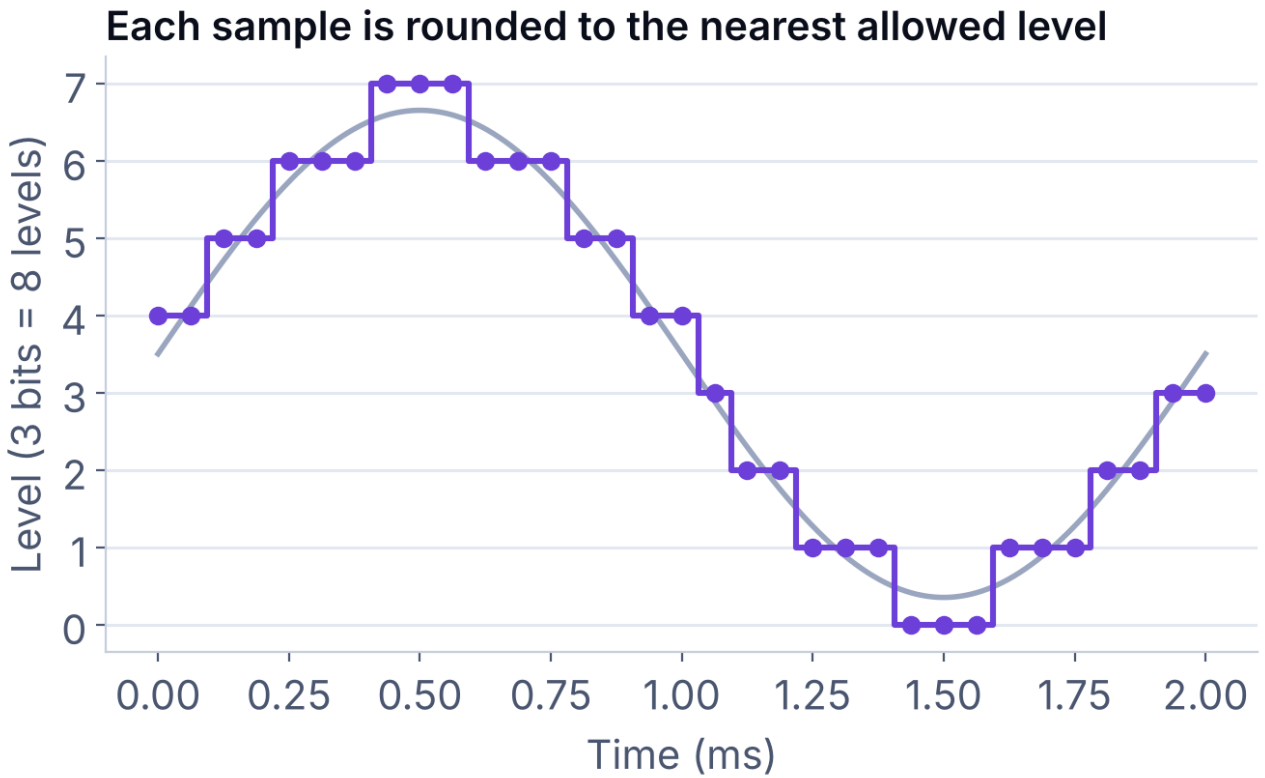


Figure 3.3. Samples of a sine wave rounded to 3 bits (8 levels). Generated with NumPy.

WHAT TO NOTICE

- b bits give 2^b levels
- 3 bits: 8 levels (figure)
- 16 bits: 65 536 levels, the usual choice for speech
- The rounding error is heard as noise
- Each extra bit: about 6 dB less noise

How big is one minute of speech?

PROBLEM

One minute of mono speech is recorded at 16 kHz with 16 bits per sample. How many bytes does it take, without compression?

REASONING

1. Samples per second: 16 000
2. Bytes per sample: 16 bits ÷ 8 = 2 bytes
3. Bytes per second: 16 000 × 2 × 1 channel = 32 000
4. One minute: 32 000 × 60 = 1 920 000 bytes

RESULT

1 920 000 bytes

≈ 1.92 MB

32 kB per second

+ 44-byte WAV header

Exercise · Compute file sizes

GOAL

Use the four factors (samples per second, bytes per sample, channels, seconds) on realistic cases.

TIME

10 min, in pairs

INSTRUCTIONS

1. 10 s of telephone speech: 8 kHz, 8 bits, mono
2. A 3-minute song on CD: 44.1 kHz, 16 bits, stereo
3. A dataset of 1 000 hours: 16 kHz, 16 bits, mono. How many gigabytes?
4. Which of the three settings cannot record a 5 kHz whistle? Why?

DELIVERABLE

Four answers on paper, with units.

Corrected together on the board.

Feeding a model the wrong sample rate

✗ INCORRECT APPROACH

```
y, sr = librosa.load(f, sr=None)
# sr = 44100, model expects 16000
text = model(y)
```

✓ CORRECTION

```
y, sr = librosa.load(f, sr=16000)
print(sr)       # check: 16000
text = model(y)
```

WHY IT FAILS

The model reads numbers, not seconds. At the wrong rate, every frequency it sees is shifted and its accuracy drops, without any error message.

Break · 15 minutes

After the break: what a speech signal looks like, frame by frame.

04

CHAPTER

The signal in time

Frequency, amplitude and loudness, then the key idea of speech analysis: cut the signal into short frames.

The sine wave: the simplest signal

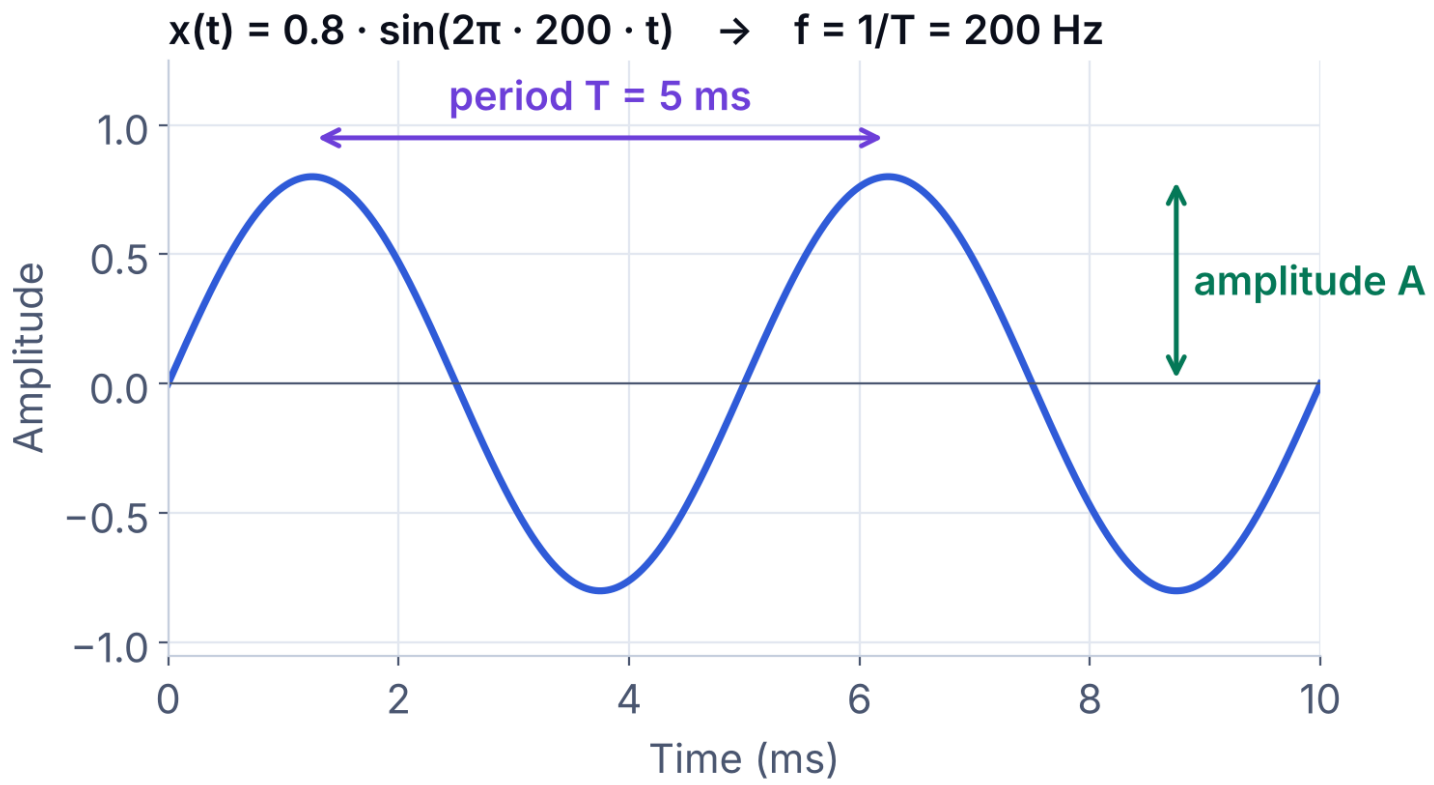


Figure 4.1. A 200 Hz sine wave: the pattern repeats every 5 ms. Generated with NumPy.

WHAT TO NOTICE

- Amplitude A: the height of the wave, heard as loudness
- Period T: the duration of one cycle, in seconds
- Frequency f = 1/T: cycles per second, in Hz
- Here T = 5 ms, so f = 200 Hz

Writing a sine wave

$$x(t) = A \cdot \sin(2\pi\, f\, t + \varphi)$$
$$x[n] = A \cdot \sin(2\pi\, f\, n \,/\, fs) \qquad n = 0, 1, 2, \ldots$$

SYMBOLS

A : amplitude

f: frequency (Hz), t : time (s)

φ : phase, where the cycle starts

n : sample number, fs : sample rate

INTUITION

sin repeats each time its angle grows by 2π. After t = 1/f, 2πft has grown by 2π: one cycle.

The second line is the sampled version: t is replaced by n / fs.

A real speech waveform

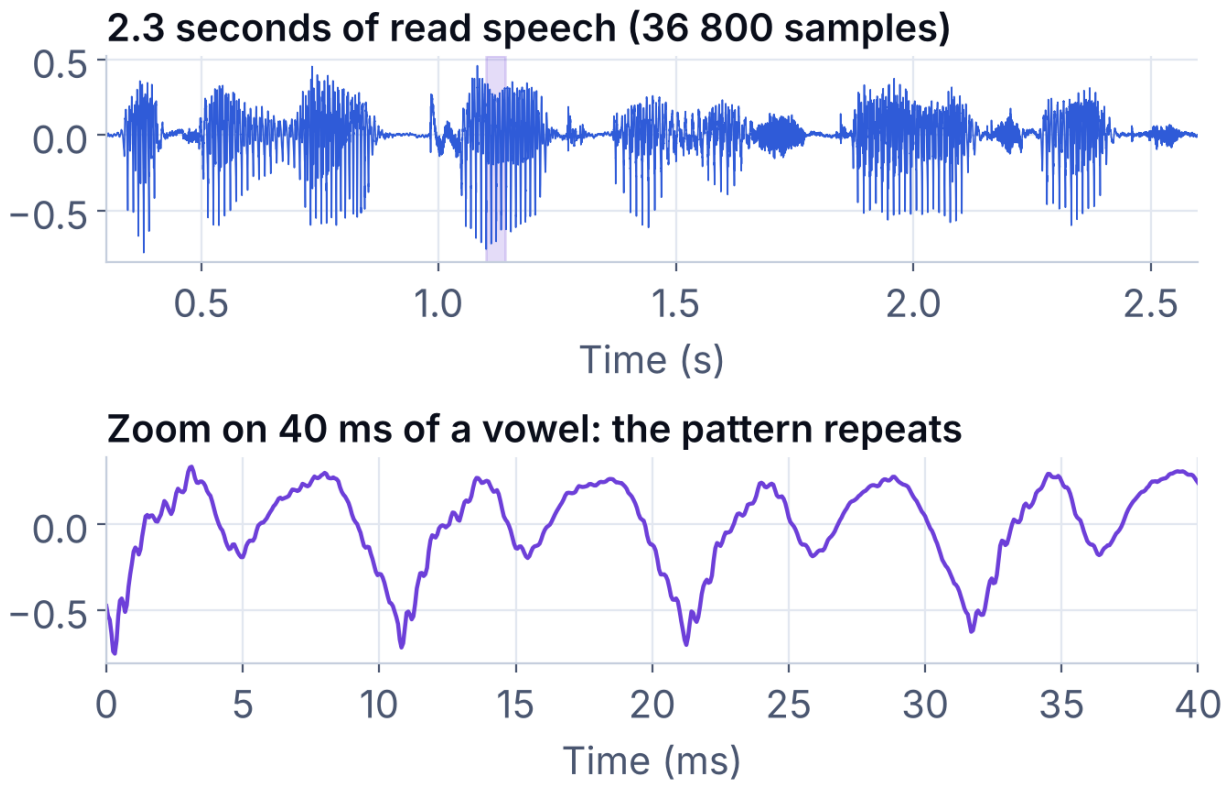


Figure 4.2. LibriSpeech 5703-47212-0000 (reader Garth Comira, CC BY 4.0) at 16 kHz. Plotted with librosa.

WHAT TO NOTICE

- Bursts are vowels, flat parts are pauses
- Inside a vowel, a shape repeats about every 10.5 ms
- $f_0 \approx 1 / 0.0105\ s \approx 95$ Hz: a low male voice
- Each period is not a sine: its shape carries the vowel

Loudness in decibels

$$L = 20 \cdot \log_{10}(A \,/\, Aref) \quad \text{dB}$$

SYMBOLS

A : measured amplitude

$Aref$: reference amplitude

L : level in decibels (dB)

INTUITION

The ear hears ratios, so we use a logarithm.

Amplitude × 2 → about +6 dB. Amplitude × 10 → +20 dB.

With the loudest sample as reference, levels are 0 dB or negative.

Short-time analysis: cutting speech into frames

DEFINITION

Speech changes constantly, but over 20 to 30 ms it is almost stable. We analyse it in short, overlapping frames and compute a few numbers for each frame.

EXAMPLE

Usual settings: frame length 25 ms, hop 10 ms. At 16 kHz: 400 samples per frame, a new frame every 160 samples, so 100 frames per second.

FRAMES AT 16 kHz

```
frame 0:  samples    0 to 399
frame 1:  samples  160 to 559
frame 2:  samples  320 to 719
…  one frame every 10 ms
```

Frame energy: the RMS value

$$RMS = \sqrt{\left((1/N) \cdot \Sigma n \, x[n]^2 \right)}$$

SYMBOLS

N: number of samples in the frame (400)

$x[n]$: sample n of the frame

Σ : sum over the frame

INTUITION

Square, average, square root: the typical size of the samples.

Squaring stops positive and negative values from cancelling.

High RMS: loud vowel. RMS near 0: silence.

Zero-crossing rate (ZCR)

$$ZCR = (1/N) \cdot \Sigma n \, 1[\, \text{sign}(x[n]) \neq \text{sign}(x[n-1]) \,]$$

SYMBOLS

N: number of samples in the frame

$1[\ldots]$: 1 if the sign changes, otherwise 0

INTUITION

Counts how often the signal crosses zero.

Vowels move slowly: low ZCR.

Noisy sounds such as /s/ and /f/ wiggle fast: high ZCR.

RMS and ZCR by hand

PROBLEM

A frame contains N = 8 samples: 0.2, 0.4, 0.2, −0.2, −0.4, −0.2, 0.2, 0.4. Compute its RMS and its ZCR.

REASONING

1. Sum of the squares = 0.68
2. Mean: 0.68 / 8 = 0.085, so RMS = √0.085
3. Sign changes: 0.2 → −0.2 and −0.2 → 0.2
4. 2 crossings in 8 samples: ZCR = 2 / 8

RESULT

RMS ≈ 0.29

ZCR = 0.25

Slow wave: low ZCR

Energy and ZCR on real speech

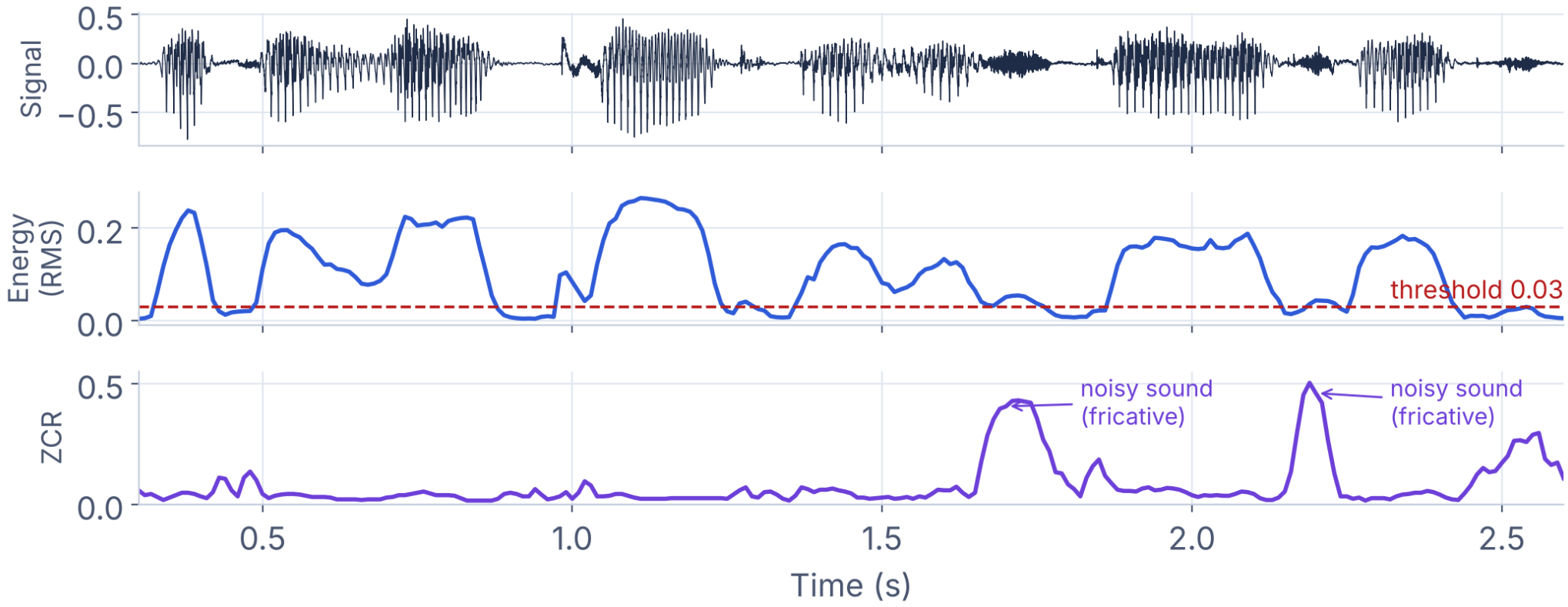


Figure 4.3. Same recording, 25 ms frames, 10 ms hop. Peaks of ZCR with low energy are fricatives. Computed with librosa 1.0.

Energy alone misses quiet consonants

✗ INCORRECT APPROACH

```
speech = rms > 0.03
```

Quiet fricatives such as /s/ and /f/ fall below the threshold

✓ CORRECTION

```
loud = rms > 0.03
noisy = zcr > 0.25
speech = loud | noisy
```

In noisy rooms: a trained voice activity detector

WHY IT FAILS

Fricatives have low energy but carry meaning. Cutting them turns “sees” into “ee”.

05

CHAPTER

A first look at frequency

Any sound can be described as a mix of pure tones. Listing those tones gives a second view of the same signal.

A complex sound is a sum of sine waves

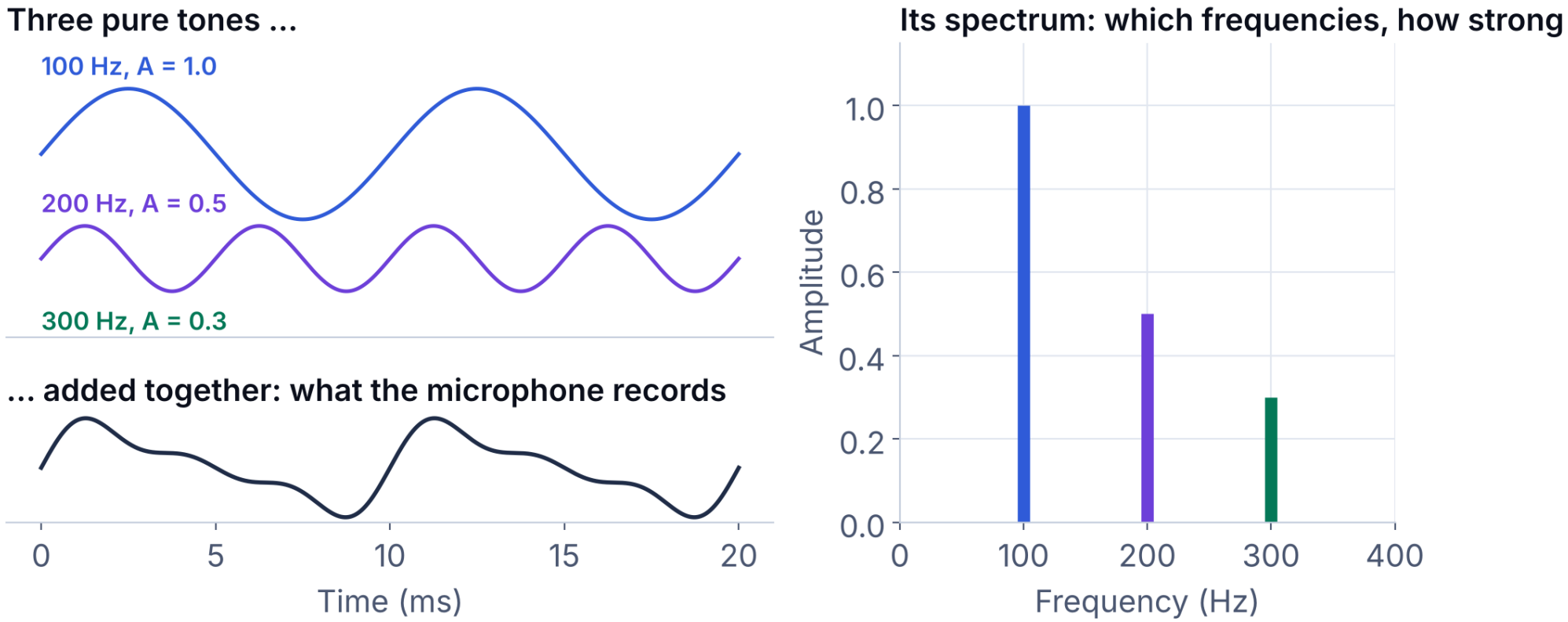


Figure 5.1. The sum of three tones looks complicated in time, but its spectrum has only three lines. Generated with NumPy.

Two views of the same signal

Time domain: the waveform

- Amplitude against time
- Shows when things happen: pauses, loud syllables
- Easy to compute energy and ZCR
- Hard to see which frequencies are present

Frequency domain: the spectrum

- Amplitude against frequency
- Shows which frequencies are present, and how strong
- Computed with the Fourier transform (Session 02)
- For a whole sentence, the timing is lost

Reading a spectrogram

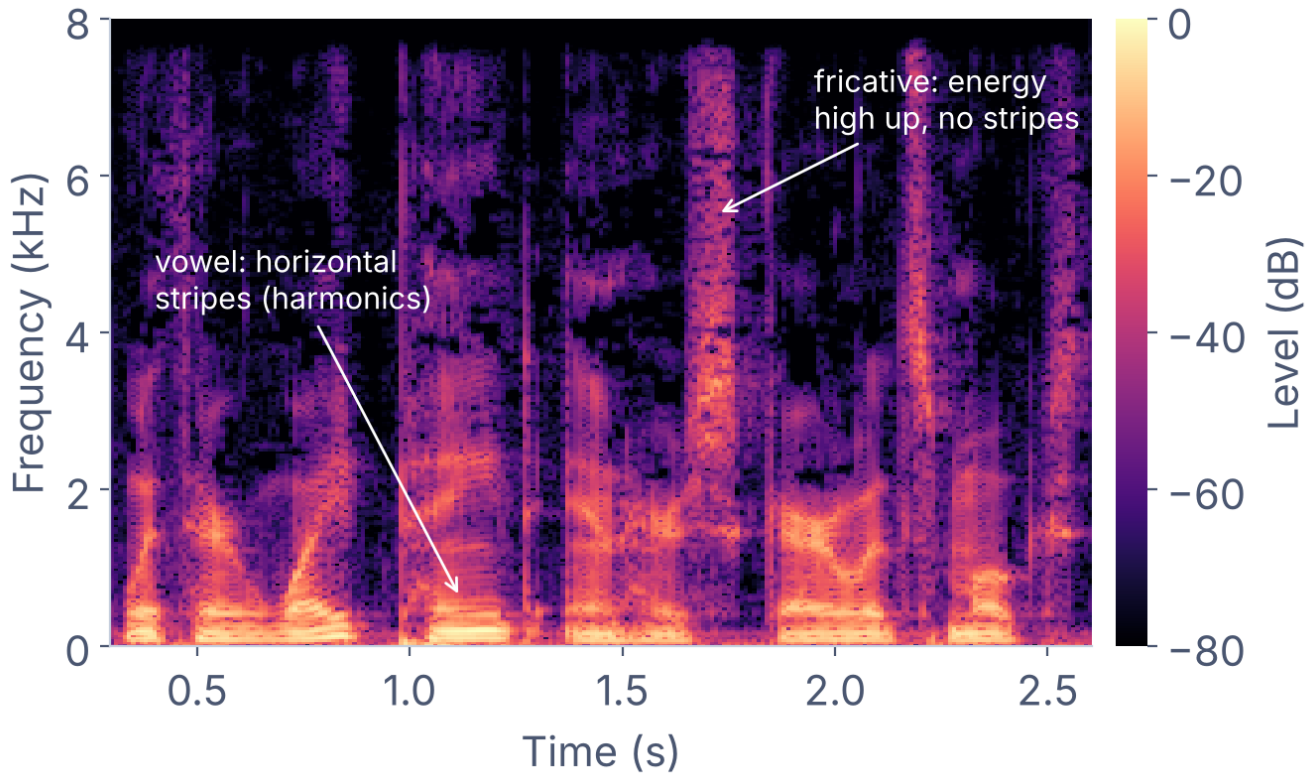


Figure 5.2. Spectrogram of the same recording (16 kHz, 25 ms window, 10 ms hop), computed with librosa.

WHAT TO NOTICE

- Horizontal: time. Vertical: frequency. Colour: energy (dB)
- Vowels: horizontal stripes (harmonics of f_0) at low frequency
- Fricatives: energy high up, no stripes
- Pauses: dark columns

What the sample rate removes

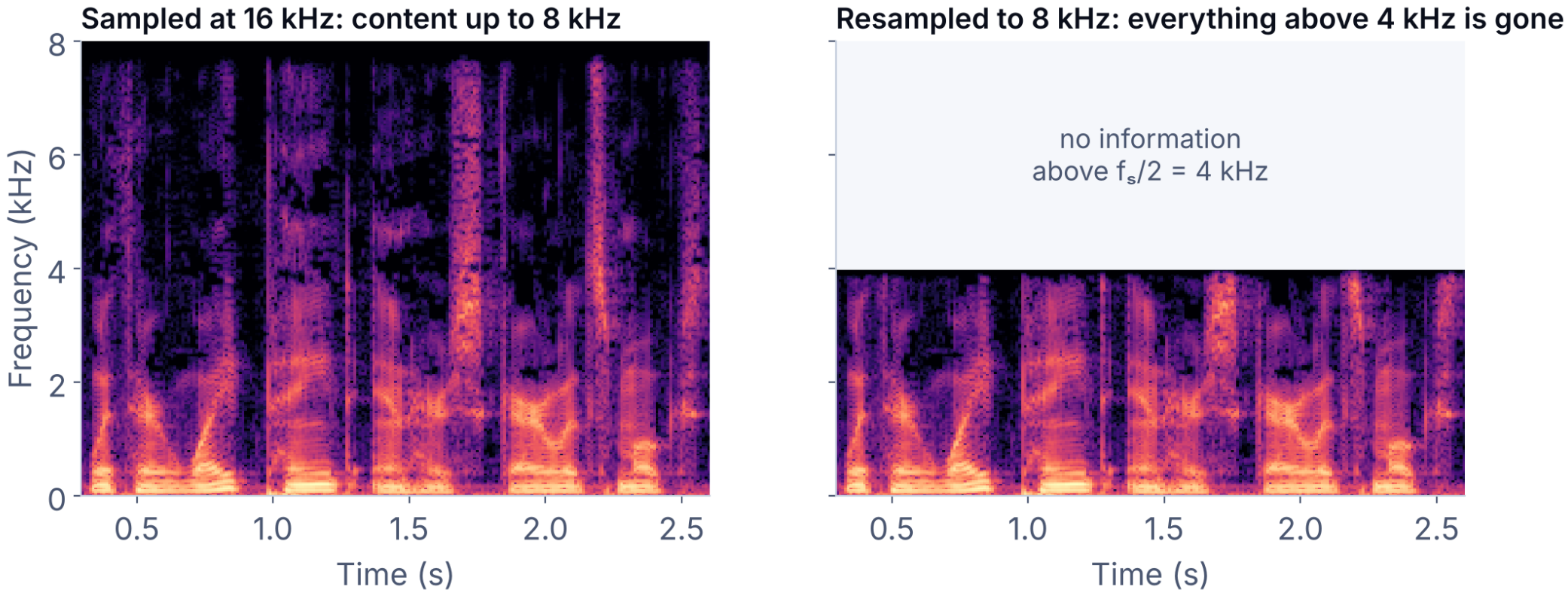


Figure 5.3. The same 2.3 s at 16 kHz and resampled to 8 kHz. Fricative energy above 4 kHz is lost. Computed with librosa.

Knowledge check

QUESTION

On a spectrogram you see 0.1 s of energy spread between 4 and 8 kHz, with no horizontal stripes. What is it most likely?

A

The vowel /a/

B

A pause

C

A fricative such as /s/

D

A vowel spoken by a man with a low voice

06

LAB

Lab: your first audio analysis in Python

Generate a tone, load real speech, change its sample rate and bit depth, then compute energy, ZCR and a spectrogram.

LAB · PYTHON

Setting up Google Colab

- Open colab.research.google.com and sign in with a Google account
- File ▸ Upload notebook: Session_01_Speech_as_a_Signal_Lab.ipynb
- Run the cells in order with Shift + Enter; the first prints the librosa version
- Use headphones for the listening cells
- Answer Q1 to Q6 in the text cell under each question

KEY TAKEAWAY

Colab runs Python 3.12 in the browser: nothing to install. The notebook works with librosa 0.11 and 1.0.

LAB · PYTHON

A pure tone in a few lines

CODE · PYTHON 3

```python
import numpy as np
from IPython.display import Audio
sr = 16000                  # per second
f, A = 440, 0.5
t = np.arange(sr) / sr  # 1 s
x = A*np.sin(2*np.pi*f*t)
print("Samples:", len(x))
print(x[:5].round(4))
Audio(x, rate=sr)
```

OUTPUT

```
Samples: 16000
[0.     0.086  0.1694
 0.2477 0.3187]
```

Wrapped here. Then an audio player appears.

LAB · PYTHON

Loading real speech

CODE · PYTHON 3

```python
import librosa

path = librosa.ex("libri1")
y, sr = librosa.load(path, sr=16000)

print("Sample rate:", sr)
print("Samples:", len(y))
print("Duration:",
      round(len(y)/sr, 2), "s")
```

OUTPUT

```
Sample rate: 16000
Samples: 237440
Duration: 14.84 s
```

237 440 / 16 000 = 14.84 s

LAB · PYTHON

Frames, energy and ZCR

CODE · PYTHON 3

```python
fl, hop = 400, 160  # samples
F = librosa.feature
rms = F.rms(y=y, frame_length=fl,
            hop_length=hop)[0]
zcr = F.zero_crossing_rate(y,
      frame_length=fl,
      hop_length=hop)[0]
print("Frames:", len(rms))
print("RMS:", rms.mean().round(4))
print("ZCR:", zcr.mean().round(4))
```

OUTPUT

```
Frames: 1485
RMS: 0.0878
ZCR: 0.0788
```

About 100 frames per second

LAB · PYTHON

A first speech detector

CODE · PYTHON 3

```
1 threshold = 0.03
2 is_speech = rms > threshold
3
4 n = is_speech.sum()
5 print("Speech frames:", n,
6       "of", len(rms))
7 print("Share:",
8       round(100*n/len(rms), 1), "%")
```

OUTPUT

```
Speech frames: 1030
of 1485
Share: 69.4 %
```

One line in the notebook; wrapped here.

LAB · PYTHON

Lab · Explore speech as a signal

GOAL

Hear and measure what sampling, quantization and framing do to a real voice.

TIME

30 min, individual

INSTRUCTIONS

1. Part A: change the tone frequency and count periods (Q1)
2. Part B: measure f_0 on a 40 ms zoom (Q2)
3. Parts C and D: listen at 8 and 4 kHz, and at 8, 4 and 2 bits (Q3, Q4)
4. Part E: change the energy threshold (Q5)
5. Part F: find a vowel and an /s/ on the spectrogram (Q6)

DELIVERABLE

Notebook (.ipynb) with Q1 to Q6 answered, before Session 02.

Optional: Part G.

Exit question

QUESTION

A model expects 16 kHz audio. You take a call recorded at 8 kHz and correctly resample it to 16 kHz. Which statement is true?

A

The file now contains frequencies up to 8 kHz

B

It has 16 000 samples per second, but still nothing above 4 kHz

C

The file is now twice as loud

D

Resampling up creates aliasing at 4 kHz

Recap

KEY TAKEAWAYS

- Speech = source (f_o, voicing) + filter (formants)
- Sampling: only frequencies below fs/2 survive
- Quantization: b bits give 2^b levels; speech uses 16 bits
- Analyse speech in frames: 25 ms every 10 ms
- Energy and ZCR separate vowels, fricatives and silence

NEXT

- Session 02: the Fourier transform and the spectrogram
- Finish lab Q1 to Q6
- Read: Jurafsky & Martin, SLP3, “Phonetics”

References

(1) Jurafsky, D., & Martin, J. H. Speech and Language Processing (3rd ed. draft). web.stanford.edu/~jurafsky/slp3

(2) Rabiner, L. R., & Schafer, R. W. (2011). Theory and Applications of Digital Speech Processing. Pearson.

(3) Fant, G. (1960). Acoustic Theory of Speech Production. Mouton.

(4) Peterson, G. E., & Barney, H. L. (1952). Control methods used in a study of the vowels. JASA, 24(2), 175–184.

(5) Panayotov, V., et al. (2015). LibriSpeech: an ASR corpus based on public domain audio books. ICASSP.

(6) Radford, A., et al. (2022). Robust speech recognition via large-scale weak supervision (Whisper). arXiv:2212.04356

(7) Srivastav, V., et al. (2026). Open ASR Leaderboard. arXiv:2510.06961

(8) librosa development team (2026). librosa 1.0 documentation. librosa.org/doc

SH

Thank you · Questions?

Next session: frequency analysis and the Fourier transform. Bring your lab notebook.

Dr. Soufiane Hourri, PhD, HDR

drsoufiane.com