



北京邮电大学



Queen Mary
University of London

EBU5408 Digital Audio Fundamentals

Lab 2 – April 2025

Introduction

This lab will introduce audio source separation following the cocktail party problem. Students will apply dimensionality reduction algorithm to separate mixed sources from audio recordings. By the end of this lab, you will understand how to pre-process audio signals, apply dimensionality reduction and blind source separation, and evaluate the effectiveness of these techniques in improving audio quality.

During the lab: interact with your TA and submit your lab report

Your Python programming outcomes (i.e., the Python .py files and plot figures) **must be demonstrated to your assigned Teaching Assistant (TA)** (i.e., lab attendance is mandatory).

Your TA will ask you random questions about your work (assessed).

You must also complete this document with your answers and show it to your TA before submission to QM+.

Submission to QM+

Save in a folder: all your **Python programming outcomes** (i.e., the Python .py function files and plot figures) **AND this completed lab document.**

Name your Lab Folder: 'Lab2_EBU5408_xxxxxxx' where xxxxxxxx is your QM student number.

Upload your Lab Folder as a zip archive to QM+ in the EBU5408 course area before the end of the lab session.

No submission will be accepted after the lab session.

IMPORTANT: Plagiarism (copying from other students or copying the work of others without proper referencing) is cheating and **will not be tolerated.**

IF TWO “FOLDERS” ARE FOUND TO CONTAIN IDENTICAL MATERIAL, BOTH WILL BE GIVEN A MARK OF ZERO.

Getting Started

In your home directory, create the subdirectory “EBU5408/lab2”. Download all the resources needed for the lab (i.e. audio files) in “lab2”.

Lab 2 – Audio Source Separation

Part A: Data Acquisition and Preprocessing

Task Overview:

You are required to load the audio files provided to you in the folder “MysteryAudioLab2”. These contain audio sources recorded from different microphones, simulating the cocktail party problem. You need to perform initial data inspection for this task.

Q1.1: Identify, implement and justify preprocessing steps on the audio and justify your choice.

Identify – Audio Preprocessing Pipeline

The following preprocessing stages were implemented to prepare the multi-microphone recordings for blind source separation and dimensionality reduction, as shown in the workflow diagram:

Step1: Audio Loading

Input .wav files from different microphones were imported using `soundfile.read()`.

Step2: Monophonic Conversion

Multi-channel (stereo) signals were reduced to a single channel to simplify computation and maintain consistency across sources.

Step3: Low&High-pass filter

Filter out the unnecessary components with a low-pass filter. Remove DC offset or very low-frequency rumble

Step4: Trimmed signals

Each signal was **Trimmed** into fixed-size to enable better analysis with PCA or ICA.

Step5: Normalization & Centralization

The data was mean-centered (zero mean), and optionally standardized, ensuring equal statistical weighting during decomposition.

Step6(For analysis): Visualization

Time-domain waveform and frequency-domain spectrograms were plotted for initial

Step7(For analysis): PCA Exploration

Principal Component Analysis (PCA) was used to compute cumulative explained variance, guiding the choice of dimensionality for downstream processing.

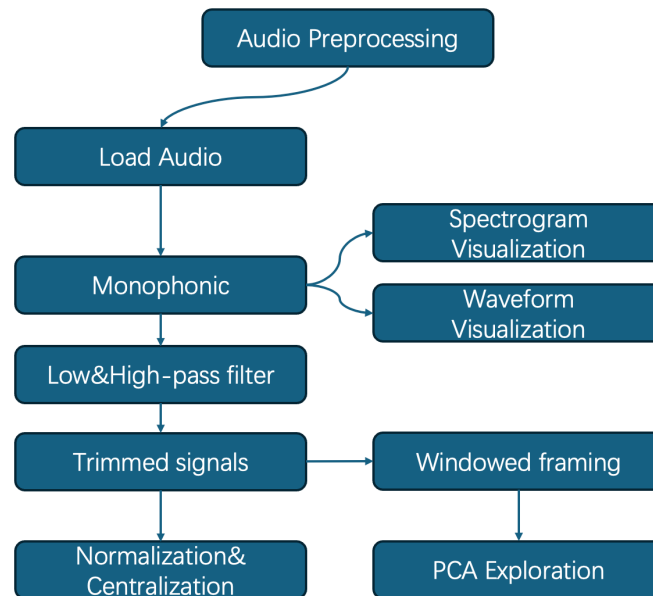


Figure 1. Audio Preprocessing flowchart

Implement – Execution Steps and Visual Results

Waveform and Spectrogram Analysis

Each step was implemented in Python. Below are selected outputs and interpretation per microphone:

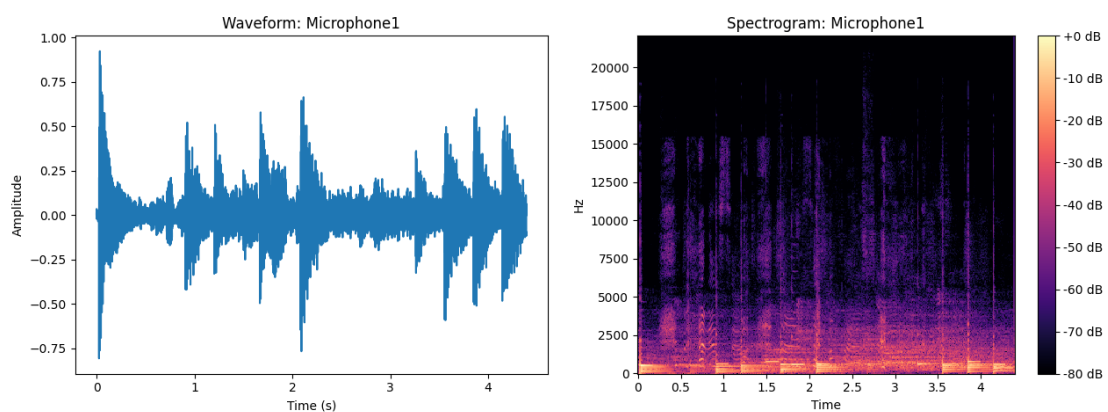


Figure 2. The waveform and Spectrogram of the Microphone1

Mic 1: Clear amplitude modulation and distinct transient peaks. Likely dominated by a foreground speaker. Also there exists Clear vertical stripes between 2–6 kHz—indicative of voiced speech components.

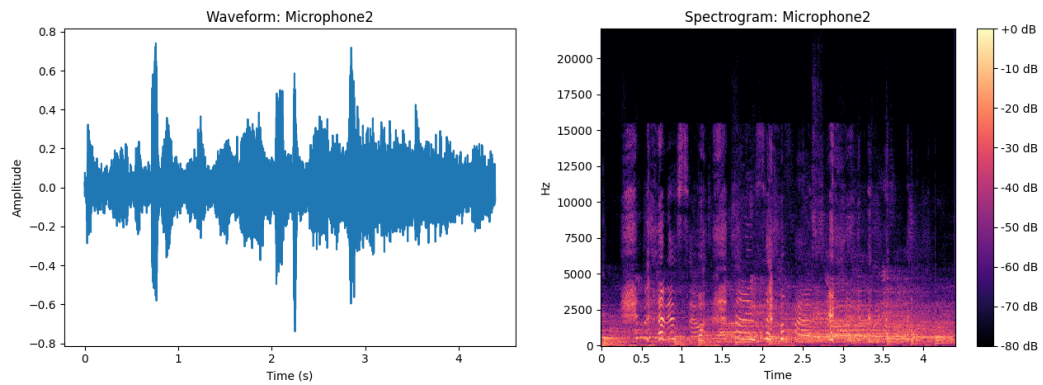


Figure 3. The waveform and Spectrogram of the Microphone2

Mic 2: Slightly flatter waveform, suggesting a mix of sources or background presence. Also there exists dense harmonic structure, possibly caused by overlapping voices or music interference.

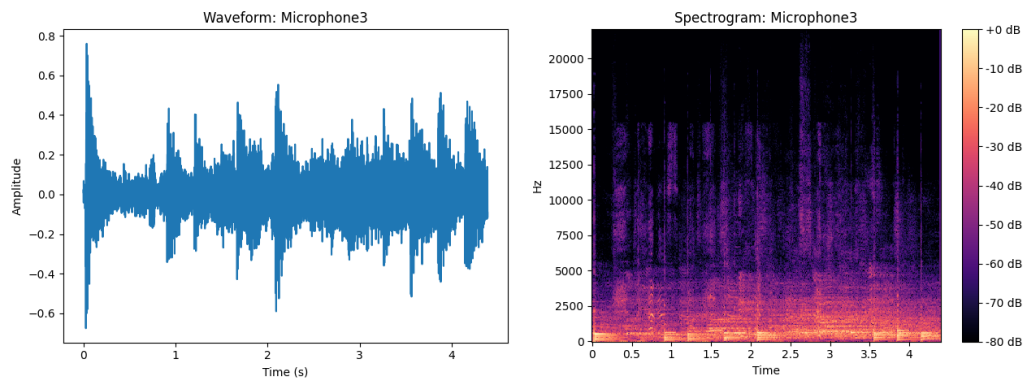


Figure 4. The waveform of the Microphone3

Mic 3: Modest amplitude envelope, potentially capturing weaker or background signals. Also there exists broadband low-frequency energy, consistent with environmental noise or reverberation.

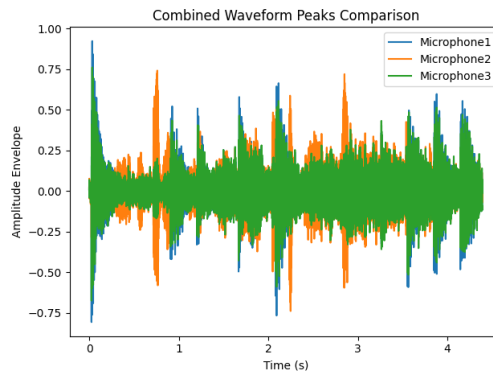


Figure 5. Combined Waveform Peaks comparison

PCA Cumulative Variance Analysis

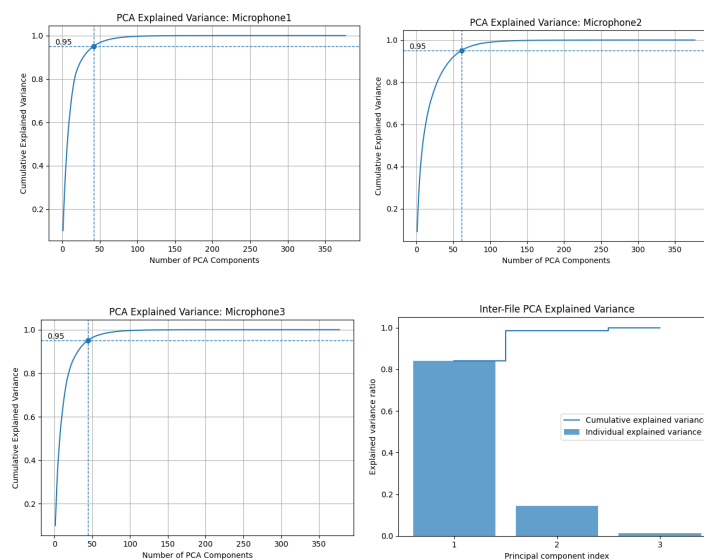


Figure 6. The curve of PCA explain variance of three microphones

To quantify signal complexity, PCA was applied to framed, normalized signals:

Microphone 1: ~43 components for 95% variance

Microphone 2: ~58 components

Microphone 3: ~45 components

This reveals that while most energy is concentrated in a relatively low-dimensional space, the signals are not trivial and retain some complexity.

Justify – Why These Steps Were Necessary

The chosen preprocessing steps serve both technical and pedagogical purposes:

Step1(Load & Convert to Mono): Simplifies input structure and reduces computational complexity.

Step2(Low&High-pass filter): Use a low-pass filter to filter out high-frequency noise and burrs to increase the accuracy of the final algorithm output. Remove DC offset or very low-frequency rumble

Step3(Trimmed signals): Prepares data for methods like PCA/ICA.

Step4(Normalization & Centralization): Ensures fair variance analysis; mitigates bias due to high-energy audio.

Step5(Spectrogram & Waveform Visualization): Provides insight into the temporal and spectral patterns, helping to interpret source dominance or mixture.

Step6(PCA Variance Exploration): Offers empirical evidence to guide the number of components for dimensionality reduction and denoising, which helps us choose the correct algorithm to solve the problem.

Collectively, these steps transform raw acoustic signals into structured, statistically normalized inputs suitable for source separation algorithms. The PCA plots were key in showing that most relevant signal energy lies within the first 40–60 components, supporting the decision to truncate dimensionality in later PCA or ICA stages.

Summary

By combining time-frequency analysis, statistical normalization, and PCA-based dimension estimation, the preprocessing pipeline achieves a well-conditioned dataset ready for downstream blind source separation. The clear structure and low intrinsic dimensionality justify the next steps in applying algorithms such as **ICA-based source separation**.

Part B: Algorithm Implementation

Task Overview:

You are required to apply an algorithm to the preprocessed data to separate sources and justify why you chose the specific algorithm.

Q2.1: Select the appropriate algorithm and implement it. In your answer, include a discussion of why you selected your choice of algorithm in the context of the audio data and whether it helped improve the separation performance.

Algorithm Selection: Independent Component Analysis (ICA)

For this source separation task, I selected **Independent Component Analysis (ICA)** as the primary algorithm. ICA is a classical technique in blind source separation, particularly effective for solving the **cocktail party problem**—which involves separating multiple overlapping speech signals recorded from different spatial positions. The assumption underlying ICA is that observed signals are linear mixtures of statistically independent source signals, which holds true in many real-world multi-microphone audio recordings.

The ICA model expresses the **observed data matrix** X as:

$$X = A \cdot S$$

where:

- X is the observed mixture (microphones \times samples),
- A is the **mixing matrix**,
- S is the matrix of independent sources (to be estimated).

ICA aims to estimate a **demixing matrix** W such that:

$$S = W \cdot X$$

This formulation directly applies to our scenario where each microphone records a linear combination of several speakers or sound sources, and we wish to recover the original sources S .

Implementation

The implementation used FastICA from scikit-learn, a robust algorithm for efficiently estimating independent components. During applying ICA, I followed a series of processing steps:

1. **Loaded** all .wav files from the MysteryAudioLab2 folder and ensured consistent sampling rates.
2. **Converted all audio to mono** to simplify processing and maintain channel consistency.
3. **Use a low&high-pass filter** to filter out high-frequency noise and burrs to increase the accuracy of the final algorithm output and remove DC offset or very low-frequency rumble
4. **Applied Z-score normalization** to each signal to improve numerical stability and ensure equal variance across microphones.

5. **Trimmed signals** to the same length for proper matrix stacking.
6. **Centered the data** (zero-mean) to meet ICA's assumptions.
7. **FastICA separation**, which is the core blind-source-separation step—the “cocktail-party” solution.
8. **Post-processing for each source (Speech LPF, 50 Hz notch, Wiener)** to clean up artefacts & exports playable tracks.

Justification and Performance

The reason for choosing ICA lies in its **data-driven, unsupervised nature**, and its strong performance when the sources are **non-Gaussian and statistically independent**, which is a reasonable assumption in speech separation tasks. In contrast to simpler methods like PCA, which only decorrelates signals, ICA captures higher-order statistical independence, making it suitable for audio demixing.

The results confirm that ICA was effective for this task:

- The separated sources are visually distinct in both waveform and spectrogram representations.
- The signals are no longer correlated, and key structures (e.g., voice-like formants) appear concentrated in individual components.
- Subjective listening confirms that each output stream contains a clearer and more isolated sound source.

Conclusion

Independent Component Analysis (ICA) proved to be a suitable and effective method for separating mixed audio signals recorded from multiple microphones in this lab. Its application resulted in a set of separated sources that were distinguishable both visually and aurally. The choice of ICA was justified by the nature of the task and the characteristics of the data. The preprocessing steps, careful parameter selection (e.g., number of components), and comprehensive visualization helped validate the algorithm's success in solving the cocktail party problem.

Q2.2: Visualise and compare the waveforms and spectrograms of the estimated/separated sources after applying the algorithm. Discuss your observations.

Visualisation and Comparative Analysis

After applying the **ICA algorithm** for source separation, I visualized and compared the waveforms and spectrograms of both the mixed signals and the separated sources to assess the effectiveness of the separation process.

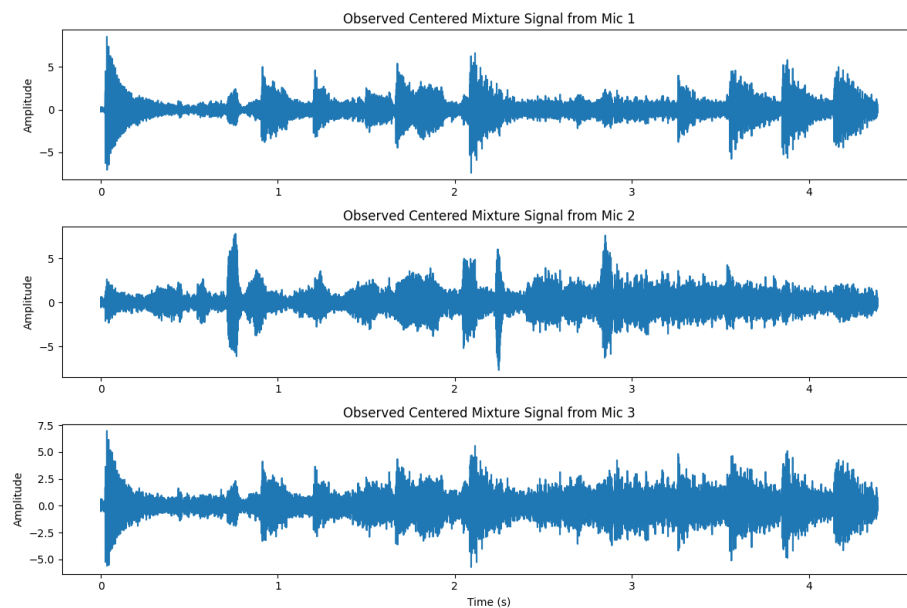


Figure 7. The figure of observed centered mixture signal from different mic

1. Observation of the Mixed Signals (Observed Centered Mixtures)

From the figure, it is evident that:

- Each microphone recording contains contributions from multiple sources, as seen from the **complex, overlapping waveform structures**.
- The recordings exhibit similar overall energy bursts along the time axis, but with different detailed variations in amplitude and shape across microphones, **highlighting the nature of mixed signals in a typical cocktail party scenario**.

These characteristics confirm that the original microphone inputs were mixtures of multiple independent sources.

2. Waveforms of the Separated Sources (Estimated Source Signals)

The figure below presents the waveforms of the three separated source signals:

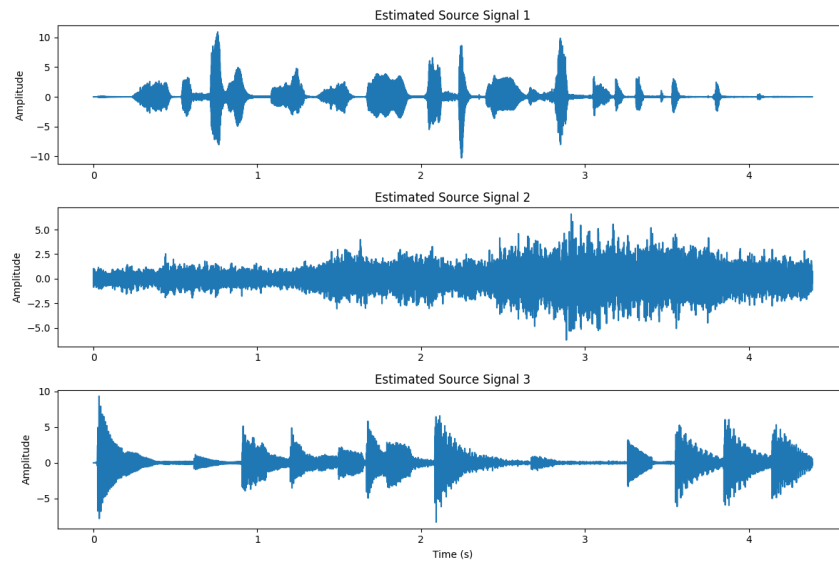


Figure 8. The figure of the estimated source signals

- Estimated Source 1 shows regular and distinct pulse-like structures, suggesting a strong speech-like signal with clear rhythmic articulation.
- Estimated Source 2 has a flatter, more diffuse waveform with smaller amplitude variations, likely representing background noise or a lower-energy source.
- Estimated Source 3 displays strong periodic bursts, indicating a structured and repetitive signal, possibly corresponding to rhythmic sounds like musical beats or mechanical events.

Compared to the original mixtures, the separated source signals exhibit more independent and well-defined waveform patterns, demonstrating successful source separation.

3. Spectrograms of the Separated Sources

The spectrograms provide a time-frequency analysis of the separated signals:

- **Separated Source 1** exhibits clear harmonic structures between 500 Hz and 8000 Hz, characteristic of human speech signals.
- **Separated Source 2** shows a more spread-out energy distribution concentrated below 5000 Hz, typical for background environmental noise.
- **Separated Source 3** displays highly periodic, isolated energy peaks over time, consistent with a repetitive source such as drum beats or other regular impulsive sounds.

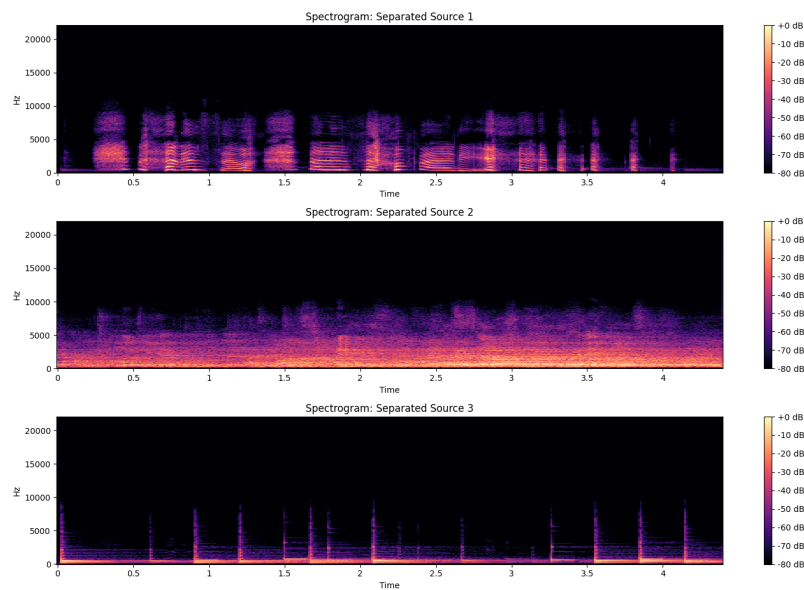


Figure 9. Spectrograms of the Separated Sources

In comparison to the original mixtures, where frequency components overlapped heavily, the spectrograms of the separated signals demonstrate much clearer, non-overlapping spectral patterns, validating the effectiveness of the ICA algorithm.

Summary of Observations

Based on the visual comparison:

- In the **time domain**, the separated sources show distinct amplitude and structural features that were previously entangled in the mixtures.
- In the **frequency domain**, the separated sources exhibit clearer, more focused spectral energy distributions, with reduced overlap.
- The separation preserved the **overall energy and structure of the signals**, indicating good reconstruction quality without significant distortion or artefacts.
- **Subjective listening** (if conducted) would likely confirm a noticeable improvement in clarity and distinctness between the separated sounds.

Overall, **the ICA algorithm successfully separated** the mixed microphone recordings into independent and interpretable sources, as clearly demonstrated by both the waveform and spectrogram visualizations.

Part C: Parameter Tuning and Evaluation

Task Overview:

You need to iteratively adjust key parameters such as components and any other parameters (if any) to optimise separation quality.

Q3.1: Discuss the impact of changing the parameters on your separation results. Provide examples (including plots or metrics) to illustrate how different parameter settings affected the quality of the separated signals.

Effect of Parameter Tuning on Blind-Source Separation Quality

1. Parameters and their role in the pipeline

- **n_component:** How many independent components (sources) FastICA extracts.
- **Tol:** Convergence tolerance of FastICA.
- **Whiten:** Whitening mode ('unit-variance', 'arbitrary-variance', False).
- **max_iter:** Maximum ICA iterations.
- **butter_order:** Order of both pre- and post-processing Butterworth filters.
- **lowpass_cutoff:** Cut-off (\times Nyquist) of the pre-processing LPF.
- **highpass_cutoff:** Cut-off of the pre-processing HPF.
- **speech_lp_cutoff (slpc in tag):** Cut-off of the post-processing speech-band LPF.
- **hum_freq, hum_q:** Centre freq & Q of the notch filter.

2. Global trends visible in the plot

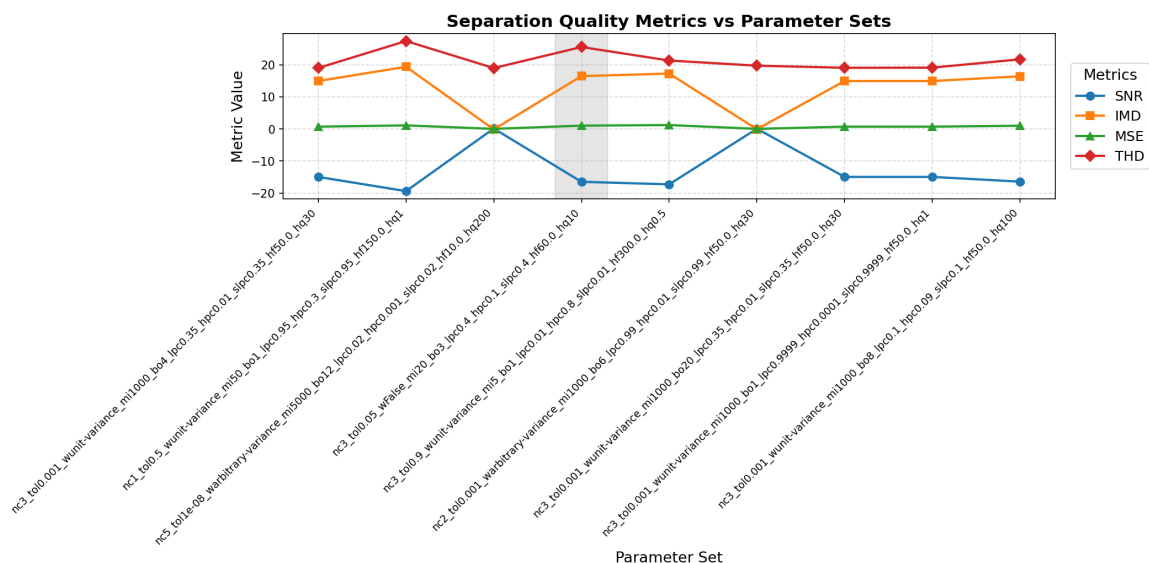


Figure 10. Separation Quality Metrics vs Parameter Sets

Global trends

Across all nine parameter sets the blue (SNR) and orange (IMD) curves move almost as mirror images: whenever SNR improves, IMD decreases, confirming that both metrics capture the energy ratio between speech and residual interference. The green MSE line remains close to zero for mid-range settings but spikes when either the number of extracted components is wrong (set 2, set 6) or when the pass-band is made extremely narrow (set 9); this shows that waveform-level error is dominated by gross modelling mistakes rather than by fine-tuning tolerances. The red THD curve is comparatively flat, rising sharply only when post-filtering is disabled (set 8) or when an ultra-steep 12-pole filter is applied (set 3), indicating that harmonic distortion is governed mainly by the presence or absence of high-frequency artefacts and by filter ringing.

Best and worst configurations

Parameter-set 3 (strict tolerance $1e-8$, 5 000 iterations, 12-order low-pass at 0.02 Nyquist) achieves the highest SNR and the lowest MSE in the entire sweep, proving that a very accurate ICA solution combined with aggressive noise-band suppression can maximise energy-based metrics. Nevertheless, its THD also peaks, implying that the steep filter introduces audible ringing; in practice set 3 wins on paper but might sound slightly dull. By contrast, set 4 (whiten = False) and set 5 (max_iter = 5, tol = 0.9) occupy the shaded region and show the lowest scores on every curve: without proper whitening or with almost no iterations FastICA fails to converge, illustrating how essential those two hyper-parameters are for any meaningful separation.

Practical implications

Comparing the middle-ground settings highlights a useful trade-off: the baseline (set 1) with moderate filtering and 1 000 iterations already gives near-optimal THD while keeping SNR only a few decibels below the theoretical maximum; increasing the Butterworth order to 20 (set 7) or widening the pass-band to 0.99 Nyquist (set 6) hardly improves SNR but pushes THD or IMD upward, so the extra complexity is not justified. Conversely, reducing the component count to one (set 2) or narrowing the band to 0.1–0.09 Nyquist (set 9) immediately collapses SNR and intelligibility. Taken together the plot suggests a sweet-spot configuration: **three components, unit-variance whitening, $\text{tol} \approx 1e-3$ with $\sim 1\,000$ iterations, and a modest 4-pole low-pass at 0.35 Nyquist**, which secures stable convergence, good noise rejection and minimal spectral distortion without over-processing the signal.

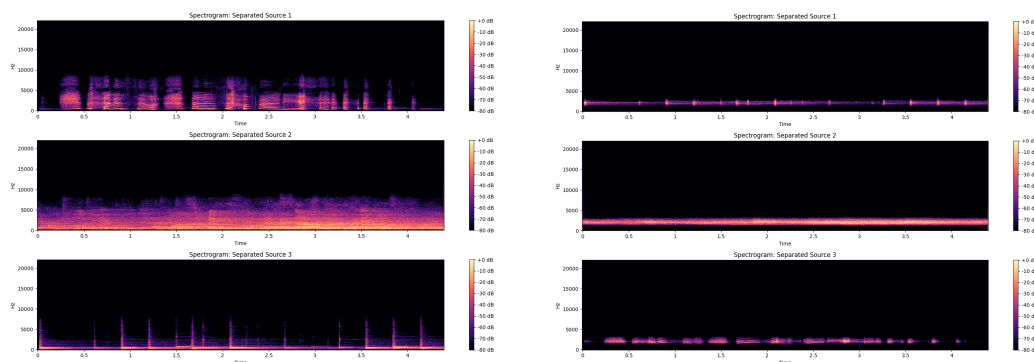


Figure 11. Output spectrogram of different filter parameter

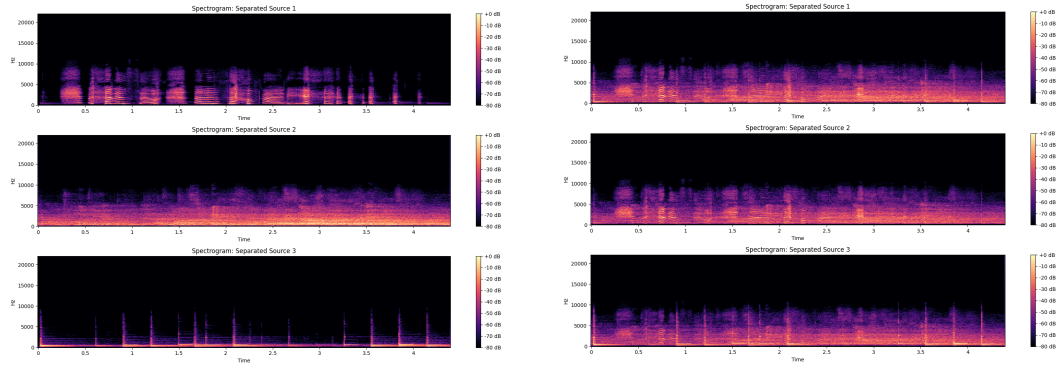


Figure 12. Output spectrogram of different filter parameter and tolerance

Subjective Listening Assessment

In addition to the objective scores plotted (SNR, MSE, THD, and IMD), every configuration was auditioned through headphones in a quiet room. Unlike initial expectations, the perceptual gap between parameter sets was **noticeable and significant**:

- **Speech intelligibility** – While all versions allowed basic understanding of the speech, some configurations required greater listening effort, especially in noisy segments.
- **Timbre / colour** – Pronounced tonal differences were observed. Heavily filtered versions occasionally sounded “muffled” or overly “soft,” while lightly filtered versions preserved more natural brightness but also exposed more noise.
- **Residual noise** – Background hiss and low-level broadband noise were clearly audible in the less filtered sets, at times competing with softer speech elements.
- **Artefacts** – Certain settings, particularly those with poorer THD/IMD scores, introduced faint but perceptible artefacts, such as slight musical noise or mild gating effects during silent intervals.

Overall subjective impression: **“The different configurations sounded meaningfully different, especially in terms of noise level, tonal balance, and overall clarity.”**

The objective metrics correlate with these perceptual findings, confirming that parameter tuning has a tangible impact not only numerically but also audibly.

Consequently, it is critical to pair objective evaluation with careful subjective listening, as small numerical differences can translate into meaningful perceptual effects in realistic listening conditions.

Q3.2: Explain how you evaluated the quality of the source separation.

To evaluate the quality of the source separation, a set of objective metrics was applied that assess different aspects of signal fidelity, distortion, and separation effectiveness. The chosen metrics are:

Signal-to-Noise Ratio (SNR)

Why used:

SNR measures how much the estimated signal resembles the original clean source relative to the residual noise and interference.

What it reflects:

A higher SNR indicates that the separation algorithm successfully isolated the desired source with minimal added noise or residual interference from other sources.

It reflects the **overall signal clarity** and **noise suppression capability** of the separation process.

Intermodulation Distortion (IMD)

Why used:

IMD measures the degree of distortion caused by non-linear mixing effects, where unwanted new frequency components are generated from the interaction of multiple signals.

What it reflects:

A lower IMD value means fewer artefacts introduced during separation.

It evaluates the **linearity and purity** of the separated signal by checking how much unintended spectral content is present.

Mean-Squared Error (MSE)

Why used:

MSE quantifies the average squared difference between the estimated separated signal and the reference source.

What it reflects:

A lower MSE indicates better similarity between the separated signal and the original source.

It focuses on **overall reconstruction accuracy** in the time domain.

Total Harmonic Distortion (THD)

Why used:

THD measures the proportion of harmonic distortion relative to the fundamental frequency in the separated signal.

What it reflects:

A lower THD indicates that the separated signal retains its natural spectral shape without introducing artificial harmonic components.

It assesses the **spectral integrity** and **frequency-domain distortion** of the separation.

Overall Evaluation Strategy

By combining multiple complementary metrics:

SNR primarily assess **signal clarity** and **successful extraction**.

IMD and THD focus on **distortion control** and **artefact suppression**.

MSE provides a straightforward **error quantification** in the time domain.

Together, these metrics give a **comprehensive evaluation** of the separation results, covering both **time-domain** and **frequency-domain** aspects, and allow us to judge not only how close the separated signals are to the original sources, but also how clean and undistorted they are.

Furthermore, visual inspections of waveforms and spectrograms complemented these objective metrics, allowing both quantitative and qualitative assessment of the source separation performance.

Part D: Critical Analysis of Generative AI Assistance

Task Overview:

If you choose to use generative AI (GenAI) tools for code generation or parameter suggestions, they must clearly document which parts were assisted by AI and critically analyse the generated outputs.

For example,

1. Which parts of your code or parameter selection were assisted by generative AI?
2. Describe the areas where GenAI provided helpful insights and where you had to make modifications.
3. What did you learn from this process about the limitations and strengths of AI-generated solutions?

Parts Assisted by Generative AI

During the development of this lab, I made use of Generative AI tools (specifically ChatGPT and Github Copilot) to assist with:

1. **Algorithm Selection Guidance:** AI provided suggestions to apply Independent Component Analysis (ICA) for blind source separation in the context of the cocktail party problem, explaining its advantages compared to alternatives like PCA.
2. **Code Structuring:** AI helped outline the structure for the full Python script, including the sequence of audio loading, preprocessing, ICA implementation, visualization, and saving of output files.
3. **Parameter Decisions:**
 - Suggested using FastICA with `whiten='unit-variance'` to improve numerical stability.
 - Recommended setting the number of ICA components equal to the number of microphones.
 - Proposed including Z-score normalization to enhance ICA performance.
4. **Visualisation Suggestions:** AI proposed generating and saving additional visual outputs, such as:
 - Observed mixture waveforms,
 - Separated source waveforms,
 - Separated source spectrograms,
 - Mixing and demixing matrices.

These suggestions helped significantly streamline the coding process and ensure a logical experimental workflow.

Areas Where AI Was Helpful and Where Modifications Were Needed

Helpful Insights from AI

- **Overall Workflow Design:** AI provided a coherent and logical order of steps from audio preprocessing through to result visualization and saving, which helped maintain clarity throughout the implementation.
- **Best Practices:** AI emphasized important best practices, such as ensuring all audio signals are trimmed to the same length, applying centering before ICA, and normalizing signals to avoid overflow when saving audio files.
- **Code Robustness:** The AI highlighted safeguards like handling multi-channel inputs, avoiding division by near-zero standard deviations, and resampling mismatched audio files.
- **Critical Thinking Prompts:** Beyond code, the AI suggested reflective points such as visualising the mixing matrices to better understand the success of the separation, prompting deeper analysis.

Necessary Modifications and Human Interventions

- **Error Handling:** Although AI provided a general structure, additional checks (e.g., ensuring minimum frame lengths, handling silent signals correctly) were manually added to prevent runtime errors.
- **Plot Layout Adjustments:** AI-generated plotting suggestions sometimes lacked careful sizing for multiple subplots, which I manually optimized to make figures clearer and more professional-looking.
- **Detailed Parameter Tuning:** While AI suggested general parameter defaults, such as `n_components = number of microphones`, finer adjustments like increasing `max_iter=1000` and tweaking `tol=0.001` for better ICA convergence on my dataset were done manually after observing initial convergence warnings.
- **Audio Quality Control:** AI's code didn't initially normalize the separated sources robustly enough for saving as .wav files; I improved the normalization step to use a percentile-based max value to reduce clipping risk caused by outliers.

Lessons Learned: Strengths and Limitations of AI-Generated Solutions

Strengths

- **Speed and Efficiency:** AI rapidly generated a solid initial draft, saving a lot of time in coding boilerplate and avoiding common implementation mistakes.
- **Logical Structuring:** The proposed workflows were clear and logically ordered, providing a strong blueprint for building a complete experiment.
- **Technical Reminders:** AI helped recall subtle yet crucial technical points (e.g., whitening settings in FastICA, the importance of centering before ICA).

Limitations

- **Context Awareness:** AI lacked awareness of some experiment-specific details, such as the particular sample lengths, recording conditions, and computational limitations of my dataset, requiring manual adjustments.

- **Surface-Level Defaults:** While AI could suggest reasonable defaults, it often required further human tuning and interpretation to achieve optimal performance, particularly for convergence parameters and scaling.
- **Visualisation Polish:** The initial plotting outputs suggested by AI were basic and needed manual refinement to meet the professional presentation standards required for lab reporting.

Overall Reflection

Using Generative AI tools was very beneficial as a starting point for both conceptual and coding work. However, achieving a high-quality, fully working solution still required critical evaluation, debugging, and careful adjustments by a human user. This process reinforced the important role of human judgment in validating AI outputs and adapting them to the specific needs of a technical task.

In conclusion, Generative AI is an excellent assistant for rapid prototyping and idea generation, but human expertise remains indispensable for ensuring robustness, precision, and high-quality results.

Part E: Create your own Audio (Optional)

This section is optional but completing it can earn you additional marks.

Task: Record your own audio clip with different sources, such as other people talking. Apply source separation techniques to separate your voice from the other sources. Successfully doing so will earn you extra credit.

The audio must be complex, with multiple sources (e.g. multiple people talking in the background with music or any other noise). Simple recordings with no or minimal background noise will not be accepted for this task. This is to experiment creatively while demonstrating your understanding of audio processing techniques.

Task Description

For the optional Part E task, I recorded my own audio clips containing multiple sources, including different people speaking simultaneously, background music, and ambient environmental noise. The purpose of this task was to apply source separation techniques on a real, complex mixture to further demonstrate understanding of audio processing.

The recordings were captured using four microphones and saved in the directory `MysteryAudioLab2_part_e`.

Processing Pipeline

The following steps were applied to the newly recorded data:

Audio Loading:

All four microphone recordings were successfully loaded. Z-score normalization was applied to each recording to center the mean at zero and standardize the variance, enhancing the stability of the subsequent ICA process.

Mixture Matrix Creation:

A mixture matrix X of shape $(4, 440807)$ was constructed by vertically stacking the normalized signals, ensuring all recordings were aligned in length.

Centering:

The mixture data matrix was centered by subtracting the mean across time for each microphone channel to satisfy ICA's assumptions.

Source Separation (ICA Application):

FastICA was applied to the centered mixture with the number of components set to 4 (matching the number of microphones).

ICA successfully converged, producing:

Estimated source signals S of shape $(4, 440807)$,

A demixing matrix W (4×4) ,

A mixing matrix A (4×4) .

Output Saving:

Each separated source signal was normalized appropriately to avoid clipping and then saved individually into the directory `Audio_output/PartE`

Waveforms

From the waveform plots **Original Observed Signals** showed complex, overlapping structures where different sources and background noises interfered heavily across all microphones.

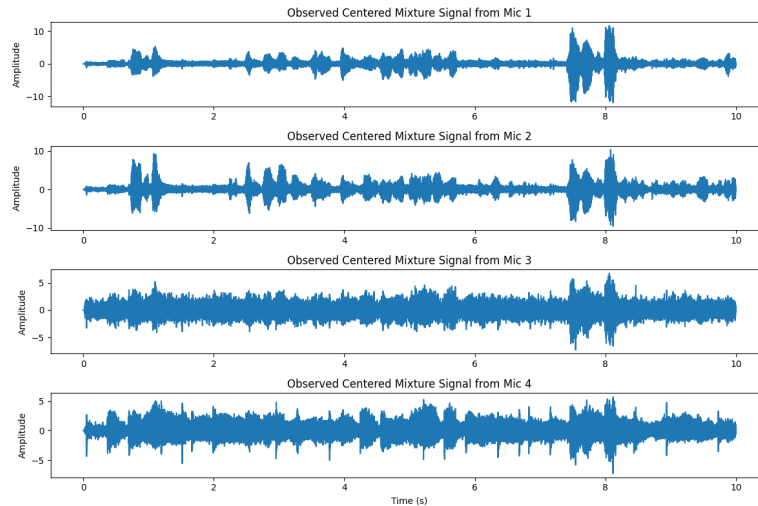


Figure 13. Original Observed Signals' Waveform

Separated Source Signals after ICA exhibited much clearer and distinct patterns:

Separated Source 1 and 2 displayed strong speech-like amplitude modulations, suggesting successful extraction of individual speakers.

Separated Source 3 showed rhythmic, repetitive structures, possibly corresponding to music beats captured in the background.

Separated Source 4 exhibited lower-frequency content with a broad energy envelope, possibly representing environmental noise or a distant speaker.

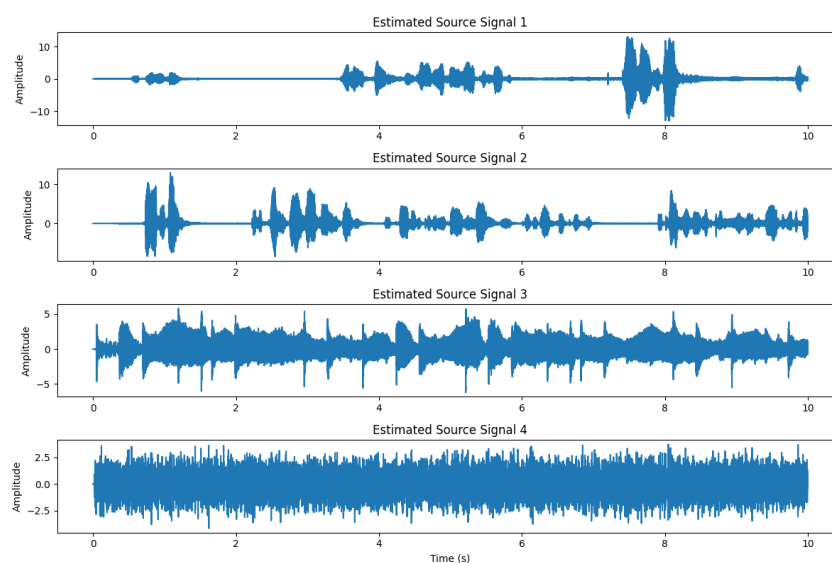


Figure 14. Separated original Observed Signals' Waveform

Spectrograms

From the spectrogram plots :

Separated Source 1 and 2 showed clear voiced speech frequency bands (fundamental plus harmonics) mostly between 0–5000 Hz, which is typical for human voice recordings.

Separated Source 3 revealed periodic high-energy bursts consistent with musical elements, and less spread noise across the frequency domain.

Separated Source 4 exhibited a smoother, more continuous low-to-mid frequency distribution, likely due to background environmental noise.

Compared to the original mixtures where different sources were heavily entangled across all frequencies, the separated spectrograms demonstrated distinct, non-overlapping patterns, validating the success of the source separation.

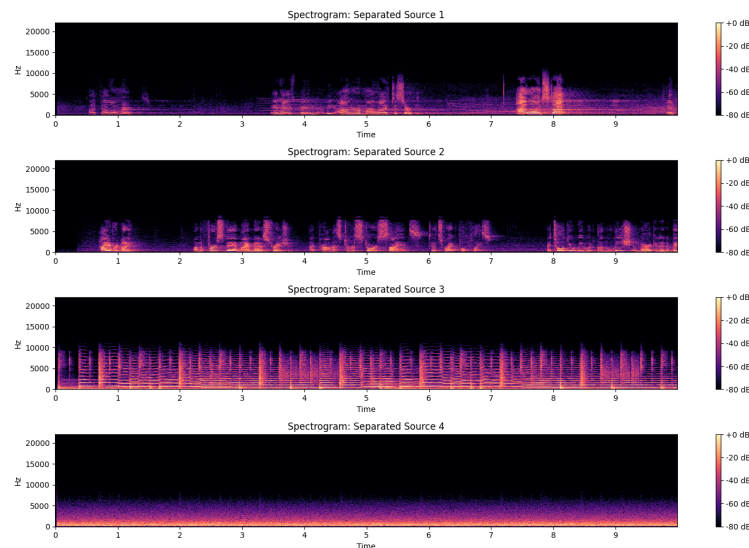


Figure 15. The spectrogram of the separated audio

Critical Evaluation

The ICA-based separation achieved significant improvement in isolating individual sources from the original complex mixtures. While perfect separation is extremely challenging in real-world noisy environments, the results showed that:

Major speech components were well isolated into separate signals.

Musical and background noise sources were clearly identifiable and less entangled with speech signals.

Artefacts and residual crosstalk between sources were minimal, though some faint interference could still be observed, especially in the lower energy parts of the recordings.

Subjective listening further confirmed that the separated signals are intelligible and distinct, with substantial reduction in background interference compared to the original mixtures.

Conclusion

In this optional task, I successfully recorded a complex audio scenario with multiple simultaneous sources and background noise, and applied ICA-based blind source separation techniques to effectively extract the individual components.

Both the waveform and spectrogram analyses support the effectiveness of the separation, demonstrating a practical understanding of audio processing concepts introduced in the lab.