RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

End of Studies Project Report

In partial fulfillment of the requirements for the State Engineer Degree in Electronics

---

Acoustic Source Separation and Localization using UMA-16 Microphone Array

---

Meskali Mohamed Ali

Presented and defended publicly on: 27/06/2026

Jury Members:

President:	Dr. Atidel Lahoulou	ENP
Examiner:	Dr. Nesrine Bouadjenek	ENP
Supervisor:	Pr. Adel Belouchrani	ENP
Co-Supervisor:	Dr. Soufiane Tebache	LDCCP/ENP

ENP 2026

Département d'Électronique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Algiers, Algeria.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

End of Studies Project Report

In partial fulfillment of the requirements for the State Engineer Degree in Electronics

---

Acoustic Source Separation and Localization using UMA-16 Microphone Array

---

Meskali Mohamed Ali

Presented and defended publicly on: 27/06/2026

Jury Members:

President:	Dr. Atidel Lahoulou	ENP
Examiner:	Dr. Nesrine Bouadjenek	ENP
Supervisor:	Pr. Adel Belouchrani	ENP
Co-Supervisor:	Dr. Soufiane Tebache	LDCCP/ENP

ENP 2026

Département d'Électronique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Algiers, Algeria.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

Mémoire de Projet de Fin d'Études

Pour l'obtention du diplôme d'Ingénieur d'État en Électronique

Séparation et localisation de sources acoustiques à l'aide du réseau de microphones UMA-16

Meskali Mohamed Ali

Présenté et soutenu publiquement le : 27/06/2026

Composition du Jury :

Président :	Dr. Atidel Lahoulou	ENP
Examinateur :	Dr. Nesrine Bouadjenek	ENP
Promoteur :	Pr. Adel Belouchrani	ENP
Co-Promoteur :	Dr. Soufiane Tebache	LDCCP/ENP

ENP 2026

Département d'Électronique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Alger, Algérie.

الملخص

تناول هذا العمل مشكلة الفصل الأعمى للمصادر الصوتية وتحديد اتجاهات وصولها في بيئات صوتية متعددة المتكلمين. اعتماداً على إطار عمل تحليل المتجهات المستقلة في المجال الترددي، صُمِّم محرك معالجة آني بلغة C++ يدعم المعالجة الدفعية غير المتصلة بالإنترنت والبث الحي الحقيقي، مع واجهة للربط العتادي لمصفوفة الميكروفونات MiniDSP UMA-16، وحساب STFT، وكشف النشاط الصوتي، والتصفية اللاحقة. تُستخلص تقديرات اتجاه الوصول هندسياً من مصفوفة الفصل المتقاربة عبر شبه العكس. دُمجت هذه المكونات في واجهة PyQt تفاعلية تعرض اتجاهات المصادر المُقدَّرة فوق تدفق فيديو حي. كما جرى تقييم المتغيرات الأسرع تقارباً AuxIVA و OverIVA مقارنةً بخوارزمية NG-IVA المرجعية.

الكلمات المفتاحية: الفصل الأعمى للمصادر - تحليل المتجهات المستقلة - تقدير اتجاه الوصول - تحديد موقع المصدر الصوتي - مصفوفات الميكروفونات - المعالجة الآنية.

Résumé

Ce travail traite de la séparation aveugle de sources et de l'estimation de la direction d'arrivée dans des environnements acoustiques multi-locuteurs. En s'appuyant sur l'Analyse Vectorielle Indépendante dans le domaine fréquentiel, un moteur C++ temps réel a été implémenté, supportant les modes batch hors ligne et de diffusion en continu, avec interfaçage matériel du réseau MiniDSP UMA-16, calcul de la STFT, détection d'activité vocale et post-filtrage. Les estimations de la direction d'arrivée sont dérivées géométriquement par pseudo-inversion de la matrice de démélange convergée. Ces composants sont intégrés dans une interface PyQt interactive superposant les directions des sources estimées sur un flux vidéo en direct. Les variantes à convergence plus rapide AuxIVA et OverIVA ont également été évaluées face au NG-IVA de référence.

Mots clés : Séparation aveugle de sources – Analyse Vectorielle Indépendante – Estimation de la direction d'arrivée – Localisation de sources sonores – Réseaux de microphones – Traitement temps réel

Abstract

This work addresses blind source separation and direction of arrival estimation in multi-speaker acoustic environment. Building on a frequency-domain Independent Vector Analysis framework, a real-time C++ engine was implemented supporting offline batch and true online streaming modes, with hardware interfacing for the MiniDSP UMA-16 array, STFT computation, voice activity detection, and post-filtering. Direction of arrival estimates are derived geometrically from the converged demixing matrix via pseudo-inversion. These components are integrated into an interactive PyQt interface overlaying estimated source directions onto a live video stream. Faster-converging variants AuxIVA and OverIVA were additionally evaluated against the NG-IVA baseline.

Keywords: Blind Source Separation – Independent Vector Analysis – Direction of Arrival Estimation – Sound Source Localization – Microphone Arrays – Real-Time Processing

Dedication

To my mother, who nurtured in me a passion for science and mathematics, and devoted herself to helping me become the best version of myself.

To my father, who did the impossible to provide me with every opportunity to succeed.

To both of them, whose greatest wish was always to see me excel.

To my grandmother, whose memory remains forever in my heart.

To all of my family, who believed in my potential and never ceased to encourage and guide me.

To my siblings, cousins, and friends, with whom I shared unforgettable moments, some wonderfully strange, some beautifully chaotic.

And finally, to the younger version of myself, who could have never imagined becoming the person I am today.

This is for you.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, **Pr. Adel Belouchrani**, for his guidance, trust, and support throughout this project. I also thank my co-supervisor, **Dr. Soufiane Tebache**, for his valuable advice and insightful feedback.

I am also grateful to the members of the jury, **Dr. Atidel Lahoulou** and **Dr. Nesrine Bouadjenek**, for evaluating my work, as well as to all the professors and teachers who carried out their mission with dedication and integrity, and who guided and inspired me throughout my academic journey at ENPC and ENP.

I owe a special debt of gratitude to my family. Growing up in an environment where scientific ambition was far from the norm, where distractions and destructive paths were often easier to follow than meaningful ones, they made sure I never lost sight of what truly mattered.

My academic journey brought together two complementary passions. During my preparatory studies at ENPC, I developed a deep appreciation for theory, particularly mathematics and physics. Later, at ENP, I embraced the practical side of engineering through hands-on projects and implementation. This combination naturally led me to signal processing, where I saw theory being implemented in reality. To me, it feels almost like magic, though perhaps science and magic are not so different after all.

One of the greatest discoveries for me at ENP was its community, the people I met here. I would particularly like to thank the ELN community, especially those senior students who served as role models in their own unique ways. **Belkhiri Ibrahim**, for his inspiring leadership and professionalism; **Amine Labbas**, for his diplomacy and wisdom; **Adel Benarab**, for his resilience and mental strength; and **Tarek Bouzand**, whom I once saw as a rival, then as a friend sharing common passions, and whose work I now continue through this project.

I would also like to thank my **IEEE family**, with whom I stepped out of my shell and shared two unforgettable years of activity, learning, teamwork, and growth.

My gratitude also goes to my Eurobot teammates. To last year's team, **Polybot**, with whom I shared a year of intense work, challenges, and unforgettable memories; and to this year's team, through whom I learned even more about myself, my strengths, and my limits. And to thank **Tarek Blg** and **Dahmen**, with whom I connected the most.

Finally, I would like to thank my closest classmates and friends, **Rayan** the social hacker, **Chicha** the IT guy, **Rached** the overly organized, **Aymen** the aura farmer, and **Khaled** el-binome, from whom I learned and with whom we grew together.

Meskali Mohamed Ali

Contents

List of Tables

List of Figures

List of Acronyms

General Introduction **16**

1 Background and State of the Art **18**
1.1 Problem Formulation: The Cocktail Party . . . 18
1.2 Acoustic Signal Models . . . 20
1.2.1 The Instantaneous Mixing Model . . . 20
1.2.2 The Convolutive Mixing Model . . . 20
1.2.3 The STFT Domain and the Narrowband Approximation . . . 21
1.2.4 The BSS Mixing Scenarios and Ambiguities . . . 24
1.3 Acoustic Environment Characterisation . . . 26
1.3.1 The Room Impulse Response . . . 26
1.3.2 Reverberation Time RT60 . . . 27
1.3.3 Impact of Reverberation on BSS and Localization . . . 27
1.4 Pre-Processing . . . 28
1.4.1 Voice Activity Detection . . . 28
1.4.2 Signal Conditioning . . . 29
1.4.3 Data Whitening: Motivation . . . 29
1.5 Blind Source Separation: Survey . . . 30
1.5.1 Independent Component Analysis . . . 30
1.5.2 Frequency-Domain ICA and the Permutation Problem . . . 30
1.5.3 Independent Vector Analysis (IVA) . . . 31
1.5.4 Adaptive Online IVA . . . 31
1.5.5 Advanced IVA Variants . . . 32
1.6 Sound Source Localization: Survey . . . 32
1.6.1 Time-Delay Based Methods . . . 33
1.6.2 Subspace Methods . . . 34
1.6.3 BSS-Based Localization . . . 34
1.6.4 Audio-Visual Fusion and Camera Overlay . . . 35
1.7 Post-Processing . . . 35
1.7.1 Why Post-Processing Is Needed . . . 36
1.7.2 Postfiltering Approaches . . . 36
1.7.3 MDP Rescaling as Implicit Post-Processing . . . 37
1.8 Conclusion and Positioning of This Work . . . 37

2 Blind Source Separation: From ICA to Adaptive IVA **39**
2.1 Independent Component Analysis . . . 39
2.1.1 The ICA Model and Its Assumptions . . . 39

2.1.2 Contrast Function, Mutual Information, and the Natural Gradient 40
2.1.3 Data Whitening . . . 40
2.2 Independent Vector Analysis . . . 41
2.2.1 The Permutation Problem in FDICA . . . 41
2.2.2 The IVA Principle: Multivariate Source Prior . . . 41
2.2.3 The IVA Cost Function and Batch Update . . . 42
2.2.4 The Minimal Distortion Principle (MDP) . . . 43
2.2.5 Batch IVA Algorithm . . . 43
2.3 Adaptive IVA with Whitening . . . 43
2.3.1 Adaptive Whitening . . . 43
2.3.2 Adaptive Separation Filter Update . . . 45
2.3.3 The Nonholonomic Constraint . . . 46
2.3.4 MDP Rescaling in the Online Setting . . . 46
2.3.5 Full Adaptive Algorithm and Complexity . . . 46
2.4 Advanced IVA Variants . . . 47
2.4.1 AuxIVA: Auxiliary-Function IVA . . . 47
2.4.2 Online AuxIVA . . . 48
2.4.3 OverIVA: Overdetermined IVA . . . 49
2.4.4 Algorithm Comparison and Theoretical Trade-offs . . . 52
2.5 Simulation Validation . . . 53
2.5.1 Simulation Setup . . . 53
2.5.2 Evaluation Metrics (BSS Eval) . . . 53
2.5.3 Single-Scenario Comparison . . . 55
2.5.4 Comprehensive Grid Evaluation . . . 56
2.5.5 Discussion . . . 57
2.5.6 Algorithm Selection for Implementation . . . 59
2.6 Pre-Processing Integration . . . 59
2.6.1 Voice Activity Detection in the Adaptive Loop . . . 59
2.6.2 Whitening as the Pre-Processing Bridge . . . 63
2.7 Post-Processing Integration . . . 64
2.7.1 Residual Artefacts After IVA Convergence . . . 64
2.7.2 Adaptive Leakage Suppression Postfilter . . . 65
2.8 Conclusion . . . 67

3 BSS-Based Sound Source Localization 68
3.1 Foundations of Array-Based Localization . . . 68
3.1.1 The TDOA-Angle Relationship . . . 68
3.1.2 Common Array Geometries and the Array Manifold . . . 69
3.1.3 The URA Array Manifold and Steering Vector . . . 71
3.2 From the BSS Demixing Matrix to ATF Vectors . . . 72
3.2.1 The MDP Relation and Its Spatial Interpretation . . . 72
3.2.2 Normalization to the Relative Transfer Function . . . 73
3.2.3 The Residual Permutation Ambiguity . . . 73
3.2.4 Frequency Averaging for Robustness . . . 74
3.3 Direction-of-Arrival Estimation Methods . . . 74
3.3.1 Phase-Slope Estimation . . . 74
3.3.2 Precomputed Dictionary Matching . . . 76
3.3.3 SRP-PHAT: Classical Reference Method . . . 76
3.3.4 Method Comparison: Theoretical Analysis . . . 77
3.4 Camera Overlay and 3D-to-2D Projection . . . 77
3.4.1 From DoA Vector to Camera Pixel . . . 77

3.4.2 Camera Calibration . . . 79
3.4.3 Depth Invariance of the Projection . . . 80
3.5 Simulation Validation . . . 81
3.5.1 Simulation Setup . . . 81
3.5.2 Stationary-Source Accuracy . . . 82
3.5.3 Elevation Accuracy . . . 84
3.5.4 Angular Resolution: Close Source Separation . . . 84
3.5.5 Discussion . . . 85
3.6 Conclusion . . . 86

4 Real-Time Implementation and Experimental Evaluation 87
Part A — System Architecture . . . 87
4.1 System Overview and Design Principles . . . 87
4.1.1 Design Principles . . . 87
4.1.2 Relation to the Generic Enhancement Pipeline . . . 88
4.1.3 End-to-End Signal Flow . . . 89
4.2 Codebase Organization . . . 90
4.3 The Processing Modules . . . 90
4.3.1 Audio Acquisition . . . 90
4.3.2 STFT Engine . . . 92
4.3.3 Voice Activity Detection . . . 92
4.3.4 Separation (IVA) Engine . . . 93
4.3.5 Postfilter . . . 93
4.3.6 Steering-Vector DoA . . . 94
4.3.7 Video Projection and Overlay . . . 94
4.3.8 Room Calibration (RT60) . . . 94
4.4 The Two Execution Pipelines . . . 95
4.4.1 Offline (Batch) Pipeline . . . 95
4.4.2 Online (Real-Time) Pipeline . . . 95
4.4.3 Thread Model and Synchronization . . . 96
4.5 Graphical User Interface and Messaging . . . 97
4.6 Implementation Status and Toward Deployment . . . 97
Part B — Experimental Evaluation . . . 99
4.7 Experimental Setup . . . 99
4.7.1 Hardware . . . 99
4.7.2 Room, Acoustics, and Test Scenarios . . . 100
4.8 Real-Time Performance . . . 101
4.9 Live Platform Demonstration . . . 101
4.9.1 Localization Accuracy (Two Sources) . . . 102
4.10 Real-World Separation Comparison . . . 103
4.10.1 Why a Reference-Free Metric Is Needed . . . 103
4.10.2 Inter-Output Correlation Metric . . . 103
4.10.3 Results . . . 104
4.11 Discussion . . . 105
4.11.1 The Reverberation–Stationarity Trade-off . . . 105
4.11.2 The Critical Distance . . . 105
4.11.3 What Remains? . . . 106

Conclusion and Perspectives 107

Bibliography 109

List of Tables

2.1 Per-frame, per-frequency-bin complexity of the adaptive NG-IVA + whitening. . . . 46
2.2 Comparison of IVA algorithm variants. "RT" = real-time capable. "$M>L$" = supports more microphones than sources. "Step size" = requires tuning. "Cvg. guar." = guaranteed monotone convergence. Complexity is the dominant cost *per frequency bin*, counted per iteration for the batch methods and per frame for the online methods, as implemented in the present MATLAB prototype (see the note below the table). M microphones, L sources, N frames. . . . 52
2.3 Common MATLAB simulation setup for the IVA experiments. . . . 53
2.4 Single-scenario final separation quality (dB), mean over the two sources, for batch (300 iterations) and online (1708 frames). Best per column in bold. 56
2.5 VAD Performance for Concatenated Isolated Words (Test 1). . . . 62
2.6 VAD Performance for Continuous Speech Segmentation (Test 2). . . . 63

3.1 Theoretical comparison of the DoA estimation methods. "BSS-based." indicates whether the method requires BSS-extracted, source-separated input. "RT" = real-time capable on the target hardware. . . . 77
3.2 MATLAB simulation setup for the DoA estimation experiments. . . . 82
3.3 Stationary-source accuracy: azimuth and elevation MAE per method, averaged over the seven stationary scenes ($L = 2$, RT60 = 200 ms, SNR = 30 dB). The azimuth standard deviation is across scenes. . . . 83
3.4 Azimuth and elevation MAE (°) on the non-zero-elevation scenes, per averaging variant and for SRP-PHAT. . . . 84
3.5 Azimuth MAE (°) versus angular separation between the two sources, per averaging variant (RT60 = 200 ms, SNR = 30 dB). SRP-PHAT is not evaluated on this sweep. . . . 86

4.1 Core library modules and their role. . . . 91
4.2 Online-engine threads: rate, data exchanged, and synchronization primitive. 97
4.3 Implementation maturity by component. . . . 98
4.4 Hardware used for all real-world experiments. . . . 99
4.5 Real-world room and acoustic conditions. . . . 100
4.6 Test scenarios. "live" = exercised on the online engine; "offline" = via the batch / MATLAB track. . . . 100
4.7 Measured per-module processing time, per hop, against the 16 ms real-time deadline. . . . 101
4.8 Real-world azimuth accuracy for the two-source live session (median A2 variant; elevation near 0°). . . . 103

List of Figures

1.1 Illustration of the cocktail party effect: multiple speakers conversing simultaneously in a noisy environment. . . . 18
1.2 Block diagram of the blind source separation model: L source signals mixed by an array of M microphones, plus additive noise. . . . 19
1.3 Multi-path propagation in a reverberant room. . . . 21
1.4 Spectrogram of a speech signal computed via the STFT in Eq. (1.7). Brighter regions indicate higher energy concentration. Harmonic structures and formant trajectories of voiced speech are clearly visible as time-varying horizontal bands. . . . 22
1.5 The Hann window function in the discrete-time domain. . . . 22
1.6 STFT frame-based analysis with the Hann window of length T. Successive frames are spaced by the hop size R and overlap by $T-R$ samples; each row shows one windowed frame $\mathbf{x}[nR+t]\gamma[t]$ prior to the DFT of Equation (1.7). 23
1.7 Illustration of BSS ambiguities for a simple two-source, two-microphone scenario. . . . 25
1.8 Schematic illustration of a room impulse response. Based on [6]. . . . 26
1.9 Schroeder decay curve of a room impulse response. The backward cumulative energy $E[n]$ is plotted on a logarithmic scale. The RT60 is determined by fitting a line to the linear decay region and extrapolating to the $-60\,\mathrm{dB}$ level (dashed line). Adapted from [14]. . . . 27
1.10 GCC-PHAT algorithm pipeline. For each microphone pair, the cross-correlation function $R_{ij}^{\mathrm{PHAT}}(\tau)$ of Equation (1.24) is computed and stored in a TDoA lookup table (left panel); the per-pair TDoA estimates are then aggregated over candidate directions $\mathbf{u}$ to form the SRP-PHAT spatial power map of Equation (1.25) (right panel), whose peak gives the DoA estimate. . . . 33
1.11 Residual cross-talk introduced by reverberation. . . . 36

2.1 The frequency-wise permutation problem in FDICA. ICA applied independently at each frequency bin f produces a valid demixing $\mathbf{W}(f)$, but the assignment of output channels to sources is arbitrary and inconsistent across bins. The ISTFT reconstruction of each output therefore mixes spectral components from different sources. . . . 42
2.2 Block diagram of the adaptive NG-IVA pipeline with whitening. Processing operates per STFT frame n. The dashed grey box groups the modules inherited from Kim [28]; the orange adaptive whitening block is the contribution of Merah & Ghecham [3] that lifts the $M = L$ restriction. The red branch shows the VAD freeze path (Section 2.6.1). . . . 44
2.3 Simulated shoebox room generated with Pyroomacoustics [41]: the 4×4 UMA-16 vertical array (black dots) is centred at $(1.5, 1.5, 1.25)\,\mathrm{m}$ and the two sources (coloured) lie 0.6 m from the array. Shown for the $\mathrm{RT60} = 0.2\,\mathrm{s}$ configuration. . . . 54

2.4 Batch convergence on the single scenario: SDR, SIR and SAR (dB) versus iteration for NG-IVA, AuxIVA and OverIVA. OverIVA converges fastest and highest; NG-IVA is the slowest and its SAR even declines as it suppresses interference. . . . 55
2.5 Online convergence on the single scenario: SDR, SIR and SAR (dB) versus frame for NG-IVA, AuxIVA and OverIVA. After a noisy cold start, OverIVA's SIR overtakes AuxIVA and keeps climbing as more frames are processed. . . . 56
2.6 Final batch SIR (dB) of NG-IVA, AuxIVA and OverIVA across the evaluation grid: reverberation (RT60), input SNR, diffuse noise, angular separation, source count, and speaker content. . . . 57
2.7 Representative online progression (RT60 0.1 s): SDR, SIR and SAR (dB) versus frame. OverIVA's SIR overtakes AuxIVA as the recursive statistics fill in. . . . 58
2.8 Robustness of the Sohn LRT VAD compared to the Energy+ZCR baseline across varying gap distances at 20 dB, 10 dB, and 0 dB SNR. The evaluation utilizes a diverse vocabulary corpus consisting of short words (*Yes*, *No*), medium-length words (*Hello*, *Hey*, *Beautiful*), and an extremely long technical word (*Pneumonoultramicroscopicsilicovolcanoconiosis*) to thoroughly validate phoneme tracking stability and boundary estimation under severe degradation. . . . 63
2.9 Sohn LRT vs Energy+ZCR Continuous speech and pause testing at different SNR levels . . . 64
2.10 Performance evaluation (SDR, SIR, and SAR) before and after the application of the adaptive leakage suppression postfilter. A marked improvement in SIR and SDR is observed. . . . 66

3.1 UMA-16 miniDSP Microphones Array . . . 69
3.2 Plane-wave propagation toward a uniform linear array (ULA), illustrating the direction of arrival θ and the inter-sensor time delay induced by the wavefront. . . . 70
3.3 Conceptual geometries for Uniform Linear (ULA), Uniform Circular (UCA), and Uniform Rectangular (URA) sensor arrays. . . . 71
3.4 Geometry of the UMA-16 uniform rectangular array (URA) and the azimuth-elevation parametrization of a source direction. The unit vector $\mathbf{u}(\theta, \phi)$ points from the array centre toward the source. . . . 72
3.5 From the BSS demixing matrix to direction-of-arrival estimates. The pseudo-inverse of the converged, MDP-rescaled demixing matrix $\mathbf{W}_s(f, n)$ yields per-source ATF columns $\hat{\mathbf{a}}_l(f, n)$. After RTF normalization, these spatial footprints feed parallel DoA estimation methods (detailed in Section 3.3). . . . 73
3.6 The pinhole camera model of Eq. (3.20): a source direction $\hat{\mathbf{d}}_l$ (orange ray) pierces the image plane, placed one focal length f in front of the camera centre, at the projected pixel $\mathbf{p}_l$. The principal point (c_x, c_y) is where the optical axis meets the image plane. . . . 78
3.7 Calibration targets: a plain checkerboard (a) versus the ChArUco board (b) adopted here, whose embedded ArUco markers allow corner identification even under partial occlusion. . . . 79
3.8 Sample calibration captures used to estimate the camera intrinsics of Eq. (3.22), with detected ChArUco corners overlaid. Took from [47] . . . 80

3.9 Depth invariance of the perspective projection. Three candidate source positions along the same DoA ray, at different depths $\lambda_1 < \lambda_2 < \lambda_3$, project to the same pixel $\mathbf{p}_l$ on the camera image plane, since the camera origin is approximately coincident with the array reference point ($\mathbf{t} \approx \mathbf{0}$). . 81
3.10 Stationary-source azimuth MAE by method (mean ± scene-to-scene standard deviation over the seven stationary scenes); the orange bar is SRP-PHAT on the separated-source reconstructions. The median (A2) is the most accurate; the energy- and presence-weighted means (A3, A4) are the least accurate. 83
3.11 Azimuth MAE as a function of the angular separation between the two simultaneous sources, for the five averaging variants. The error stays bounded (roughly 1–5° for A1/A2) even at 5° separation: because the sources are separated by BSS before localization, proximity does not collapse the estimate. 85

4.1 Staged promotion path followed by every algorithm before it enters the real-time platform. 88
4.2 Two-process separation between the C++ engine and the GUI, communicating exclusively over message passing. 88
4.3 Generic schema of multichannel speech enhancement and source separation, after Gannot et al. [6]. Source signals pass through room acoustics into a multichannel mixture, processed by spatial filtering and parameter estimation (with a feedback loop between them), followed by optional postfiltering. 89
4.4 End-to-end signal flow of the platform. Top/Middle: the audio separation path (capture → STFT → VAD → whitening → IVA → postfilter → reconstruction). Note how VAD dictates the matrix update logic. Bottom: the localization (orange) and visualization (purple) branches driven by the demixing matrix $\mathbf{W}(f)$ and the camera. Threading is omitted here and treated in Section 4.4. 89
4.5 Layered architecture of the whole project. Arrows point in the direction of allowed dependency. The core library (`src/`) holds every algorithm; the two applications and the GUI contain only wiring; the experiment area sits outside the library and feeds validated algorithms into it through the promotion pipeline of Figure 4.1. 90
4.6 Directory layout of the codebase. The four areas of Figure 4.5 (`src/`, `apps/`, `experiments/`, GUI) sit alongside the build and dependency scaffolding (`bin/`, `build/`, `third_party/`, `data/`, `docs/`). 91
4.7 Ring buffer used for live capture: the UMA-16's single serial, channel-interleaved stream is written in, de-interleaved, and read out by the STFT engine 92
4.8 STFT engine: analysis (top) windows and transforms each hop to $X(f, n)$; synthesis (bottom) inverts and overlap-adds successive frames to reconstruct $\hat{s}[n]$. 92
4.9 Streaming Sohn LRT VAD block (Algorithm 7): per-bin power feeds the decision-directed SNRs and the log-likelihood ratio; the aggregated statistic Λ is thresholded with hangover, while the noise PSD is updated at a speech/silence-dependent rate. 93

4.10 Separation engine: a common per-frame interface (centre / optional whitening → core update → MDP rescaling) with three interchangeable cores. NG-IVA uses the nonholonomic natural gradient; AuxIVA/OverIVA use iterative projection, OverIVA adding the background constraint $\mathbf{J}_f$. . . 93
4.11 Leakage-suppression postfilter (Algorithm 8): the interference proxy from the other outputs is scaled by γ_{lk} and subtracted from the smoothed target power; the floored Wiener gain $G \geq G_{\min}$ then multiplies Y_l. 94
4.12 Phase-slope DoA block (Algorithm 9): the pseudo-inverse of the demixing matrix yields per-source ATF columns, whose inter-element phase slopes across the 4×4 URA give the spatial frequencies and, after frequency aggregation, the angles. 94
4.13 Video projection and overlay block: the DoA angles become a 3D ray, projected to a pixel through the calibrated camera model (Eq. (3.20)) and drawn on the undistorted webcam frame, then JPEG-encoded for the interface. 94
4.14 Offline pipeline. Each GUI command launches one worker thread. The demixing matrices are persisted to disk between separation and localization, so memory cost is independent of recording length; the projection worker merges the overlaid video and audio with FFmpeg. 95
4.15 Online real-time pipeline. Each horizontal lane is one thread, annotated with its period; time flows left to right. Coloured arrows mark cross-thread handoffs and the synchronization primitive used for each: a lock-free ring buffer for audio, atomic variables for the localization results, and a single mutex for the camera frame. The localization stage runs only every eighth DSP hop. 96
4.16 The two ZeroMQ socket pairs: synchronous REQ/REP for commands on port 5555, asynchronous PUB/SUB for events on port 5556. 97
4.17 Overview of the graphical interface panels. 98
4.18 The MiniDSP UMA-16 microphone array, with the webcam seated in its central aperture. 99
4.19 Top-view schematic of the real-world setup: the UMA-16 with co-located webcam faces the speakers, who are seated at known azimuths relative to boresight; the array connects to the laptop over USB. 100
4.20 Per-module latency breakdown against the 16 ms real-time deadline. 101
4.21 Physical setup used for the live demonstration. 102
4.22 Graphical interface during a live recording session: the DoA sphere (left) and the camera overlay tracking each speaker (right). 102
4.23 Two-source live overlay used for the azimuth-accuracy measurement of Table 4.8. 103
4.24 Inter-output correlation matrices (Eq. 4.1) for the three algorithms; off-diagonal cells are the residual cross-talk between separated outputs. 104
4.25 Mean off-diagonal correlation $\bar{\rho}$ per algorithm and number of sources (lower is better). 105

List of Acronyms

- **SSL**: Sound Source Localization
- **DoA**: Direction of Arrival
- **TDoA**: Time Difference of Arrival
- **AoA**: Angle of Arrival
- **FDoA**: Frequency Difference of Arrival
- **ToA**: Time of Arrival
- **ToF**: Time of Flight
- **RSS**: Received Signal Strength
- **FFT**: Fast Fourier Transform
- **DFT**: Discrete Fourier Transform
- **STFT**: Short-Time Fourier Transform
- **COLA**: Constant Overlap-Add
- **OLA**: Overlap-Add
- **SNR**: Signal-to-Noise Ratio
- **SRP**: Steered Response Power
- **PHAT**: Phase Transform
- **SPBF**: Separated-Signal Beamforming
- **GCC**: Generalized Cross-Correlation
- **MVDR**: Minimum Variance Distortionless Response
- **LCMV**: Linearly Constrained Minimum Variance
- **MWF**: Multichannel Wiener Filter
- **MSDW-MWF**: Multiple Speech Distortion Weighted Multichannel Wiener Filter
- **GSC**: Generalized Sidelobe Canceller
- **RTF**: Relative Transfer Function

- **ATF**: Acoustic Transfer Function
- **ICA**: Independent Component Analysis
- **IVA**: Independent Vector Analysis
- **FDICA**: Frequency-Domain Independent Component Analysis
- **NG-IVA**: Natural Gradient Independent Vector Analysis
- **AuxIVA**: Auxiliary-function-based Independent Vector Analysis
- **OverIVA**: Overdetermined Independent Vector Analysis
- **MDP**: Minimal Distortion Principle
- **VAD**: Voice Activity Detection
- **URA**: Uniform Rectangular Array
- **ULA**: Uniform Linear Array
- **RIR**: Room Impulse Response
- **DRR**: Direct-to-Reverberant Ratio
- **RMSE**: Root Mean Square Error
- **MAE**: Mean Absolute Error

Introduction

The human auditory system can isolate a specific voice within an acoustic environment containing multiple overlapping sources, ambient noise, and reverberation. This capability is referred to as the cocktail party effect. In practical scenarios, audio signals captured by microphone arrays are mixed observations of target speech and environmental interference. Replicating this source separation performance computationally is a major focus in acoustic signal processing.

Conventional audio processing systems degrade when operating in reverberant rooms with multiple simultaneous speakers. Wall reflections introduce multi-path propagation, resulting in a convolutive mixing process. To process such environments, an engineered system must resolve two interdependent problems: Blind Source Separation (BSS) and Sound Source Localization (SSL). BSS unmixes the overlapping audio streams without prior knowledge of the source signals or the room geometry. SSL estimates the spatial coordinates or Directions of Arrival (DoA) of the active sources.

Project Context

This project builds upon two complementary research lines developed within the **LDCCP Laboratory** at the École Nationale Polytechnique (ENP):

- **The Algorithmic Track:** Berrah and Mendjel [1, 2] (2022) introduced an IVA-based framework for MIMO source separation exploiting spatial diversity through back projection. This was later extended by Merah and Ghecham [3] (2023) into an adaptive MATLAB implementation of natural gradient IVA, incorporating adaptive whitening to handle overdetermined configurations ($M > L$), where M is the number of microphones and L is the number of sources. However, the framework remained limited to offline or simulated execution.
- **The Spatial-Visual Track:** Bouznad [4] (2025) developed a soft real-time sound source localization system using a 16-channel UMA-16 microphone array and hierarchical SRP-PHAT, projecting speaker positions onto a live camera stream. However, this system lacked source-separation capabilities, so its localization performance degraded under overlapping speech and it was limited to tracking a single speaker at a time.

This project combines these two approaches into a unified audiovisual platform by migrating the underlying algorithms to a real-time C++ implementation, enabling concurrent source separation and spatial localization.

Motivation and Contributions

This project aims to design and implement a professional, end-to-end audiovisual platform for blind source separation and localization. The goal is to build a complete product that integrates all essential pipeline stages: calibration, preprocessing, real-time algorithmic execution, and post-processing. The system is designed with a modular architecture to ensure ease of integration and future development. The technical contributions are:

1. **Real-Time C++ Processing Engine:** A modular, multithreaded C++ backend was developed for low-latency audio processing using a MiniDSP UMA-16 array. It handles memory allocation, cache-efficient STFT layouts, online Voice Activity Detection (VAD), and post-filtering under strict real-time constraints.
2. **Implementation of Advanced IVA Variants:** The standard Natural Gradient IVA (NG-IVA) was implemented and evaluated alongside advanced, fast-converging variants: Auxiliary-function-based IVA (AuxIVA) and Overdetermined IVA (OverIVA).
3. **BSS-Based DoA Estimation:** A geometric Direction of Arrival (DoA) extraction method was formulated using the pseudo-inverse of the converged demixing matrix $\mathbf{W}(f)$. This extracts spatial steering vectors directly, bypassing computationally expensive standard spatial grid searches.
4. **Interactive GUI and Audiovisual Integration:** A PyQt graphical interface was designed to map estimated 3D acoustic source directions onto a 2D live camera feed. This real-time video overlay allows users to visually identify, manually select, and dynamically route specific separated audio sources; the underlying projection geometry and camera calibration are detailed in Chapter 3.

Project Roadmap

- Chapter 1 reviews acoustic propagation, convolutive mixing models, room acoustics, the evolution of BSS from ICA to online IVA, and classical source localization frameworks.
- Chapter 2 details the mathematics of ICA and Natural Gradient IVA (NG-IVA), incorporates pre- and post-processing stages, and evaluates two advanced variants: AuxIVA and OverIVA.
- Chapter 3 presents a methodology to extract steering vectors directly from the demixing matrix $\mathbf{W}(f)$, detailing frequency-averaging, dictionary-matching, and spatial likelihood mapping for DoA estimation.
- Chapter 4 documents the software architecture and deployment, covering the multithreaded [1] engine, cache-efficient STFT, PyQt GUI, camera projection, real-world benchmarking, latency, and execution time analysis.

Finally, the conclusion summarizes the technical achievements, environmental limitations, and future developments.

[1] A multithreaded engine executes multiple processing tasks concurrently across separate threads to improve throughput and responsiveness.

Chapter 1

Background and State of the Art

This chapter details the theoretical and algorithmic background for the developed system. The problem of separating and localizing multiple simultaneous speakers in a reverberant environment requires techniques from acoustic signal processing, blind source separation (BSS), sound source localization (SSL), and real-time system design. Understanding the contributions of this project requires familiarity with the mathematical foundations and the current state of the art in each of these areas.

1.1 Problem Formulation: The Cocktail Party

Humans possess a remarkable ability to focus on a single voice in a crowded, noisy environment, a phenomenon known in psychoacoustics as the *selective auditory attention* effect. In a room full of simultaneous conversations, a human listener can effortlessly track one speaker, suppress all others, and switch attention at will. This ability, referred to as the *cocktail party effect*, was first formally studied by Cherry [5] and has since motivated decades of research in acoustic signal processing, as illustrated in Figure 1.1.



Figure 1.1: Illustration of the cocktail party effect: multiple speakers conversing simultaneously in a noisy environment.

Replicating this behaviour in a machine is far from trivial. Consider L speakers talking simultaneously in a reverberant room, with their voices captured by an array of M microphones. Each microphone records a different mixture of all L sources, altered by the room acoustics and corrupted by background noise. The goal of *Blind Source Separation* (BSS) is to recover each individual speech signal from these observed mixtures, without any prior knowledge of the mixing process, hence the term "blind".

Three fundamental challenges make this problem difficult in practice:

- **Reverberation.** Sound reflects off walls, ceiling, and furniture before reaching the microphones, creating multiple delayed and attenuated copies of each source signal.
- **Overlapping speech.** When speakers are simultaneously active, their signals overlap in both time and frequency, making source-specific features difficult to extract.
- **Unknown mixing conditions.** The "blind" constraint means no prior information about room geometry, speaker positions, or acoustic transfer functions is assumed.

Figure 1.2 depicts the resulting block diagram of the mixing and separation process described above.

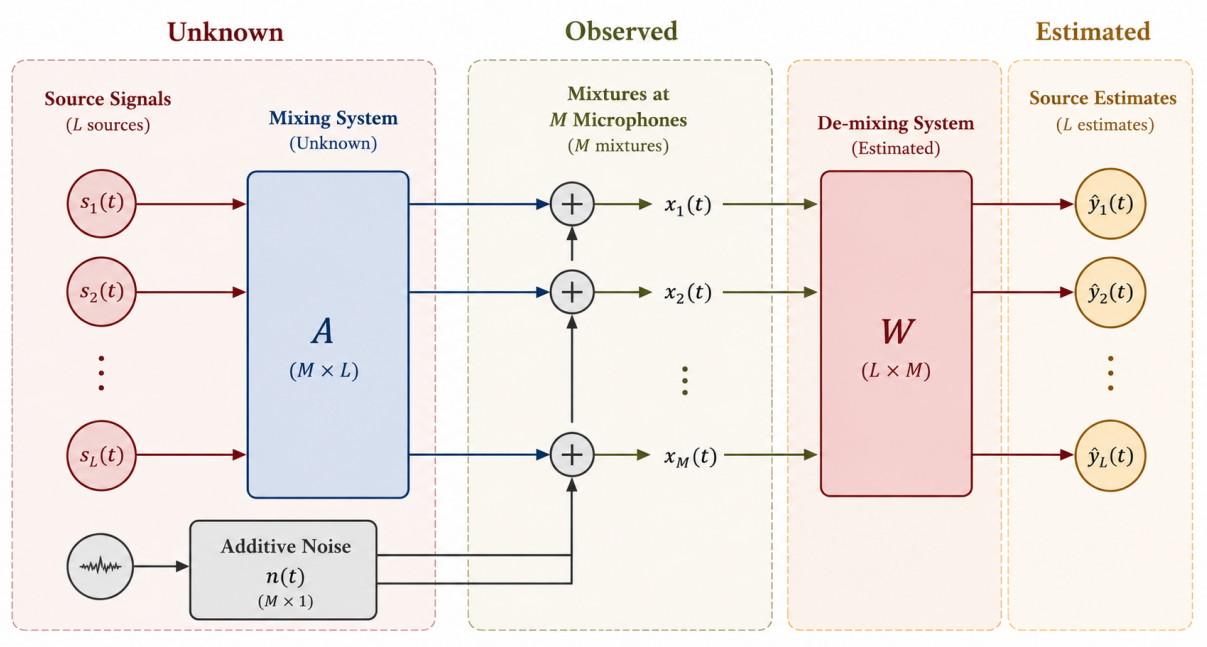


Figure 1.2: Block diagram of the blind source separation model: L source signals mixed by an array of M microphones, plus additive noise.

Beyond separation, a complete audiovisual system must also answer the question: *where is each speaker?* Localizing each separated source and projecting its direction onto a live camera feed adds a perceptual and interactive dimension to the separation task. The system developed in this project integrates both capabilities into a single platform: it separates the voices of multiple speakers captured by a 16-channel microphone array, estimates the direction of arrival (DoA) of each speaker extracted directly from the separation process, and overlays the estimated positions onto a live video stream. The user can interactively select which separated source to listen to by pressing a button in the graphical interface.

1.2 Acoustic Signal Models

The mathematical model chosen to describe how sound propagates from sources to microphones determines every design decision that follows: the choice of algorithm, the domain of processing, and the assumptions under which the system is valid. This section introduces the signal models used throughout this report, from the simplest instantaneous model to the frequency-domain narrowband approximation that underlies modern BSS and localization algorithms.

1.2.1 The Instantaneous Mixing Model

The simplest model assumes that the propagation delay from each source to each microphone is negligible. Under this *instantaneous mixing* assumption, the signal observed at microphone m at discrete time n is a linear combination of all L source signals:

$$x_m[n] = \sum_{l=1}^{L} a_{ml}\, s_l[n] + v_m[n], \quad m = 1, \ldots, M \tag{1.1}$$

where $s_l[n]$ is the l-th source signal, $a_{ml} \in \mathbb{R}$ is the mixing coefficient from source l to microphone m, and $v_m[n]$ is additive noise. In matrix form:

$$\mathbf{x}[n] = \mathbf{A}\,\mathbf{s}[n] + \mathbf{v}[n] \tag{1.2}$$

where $\mathbf{x}[n] = [x_1[n], \ldots, x_M[n]]^T \in \mathbb{R}^M$ is the mixture vector, $\mathbf{s}[n] = [s_1[n], \ldots, s_L[n]]^T \in \mathbb{R}^L$ is the source vector, $\mathbf{A} \in \mathbb{R}^{M \times L}$ is the *mixing matrix*, and $\mathbf{v}[n] = [v_1[n], \ldots, v_M[n]]^T \in \mathbb{R}^M$ is the noise vector.

BSS under this model aims to find a *demixing matrix* $\mathbf{W} \in \mathbb{R}^{L \times M}$ such that:

$$\mathbf{y}[n] = \mathbf{W}\,\mathbf{x}[n] \approx \mathbf{s}[n] \tag{1.3}$$

where $\mathbf{y}[n] = [y_1[n], \ldots, y_M[n]]^T \in \mathbb{R}^L$ is the estimated source vector. However, the instantaneous model is not realistic for speech in rooms. It assumes that sound reaches all microphones at the same time and with no reflections, which is never the case in practice.

1.2.2 The Convolutive Mixing Model

In a real room, sound propagates along multiple paths. A direct path from source to microphone, and many reflected paths off walls, furniture, and other surfaces. Each path has a different length, and therefore a different propagation delay and attenuation. The combined effect is modelled by a *room impulse response* (RIR), a finite impulse response (FIR) filter $\mathbf{a}_{ml}[p]$ of length P samples. The signal observed at microphone m becomes:

$$x_m[n] = \sum_{l=1}^{L} \sum_{p=0}^{P-1} a_{ml}[p]\, s_l[n-p] + v_m[n] \tag{1.4}$$

where $p \in \{0, \ldots, P-1\}$ is the tap index running over the samples of the room impulse response. In matrix form, using the multichannel convolution notation:

$$\mathbf{x}[n] = \sum_{p=0}^{P-1} \mathbf{A}[p]\,\mathbf{s}[n-p] + \mathbf{v}[n] \tag{1.5}$$

where $\mathbf{A}[p] \in \mathbb{R}^{M \times L}$ contains the contribution of the RIR at sample delay p from each source to each microphone. This model is called the *convolutive mixing model* [6, 7].

Under this model, the goal of BSS is to find a demixing filter $\mathbf{W}[p]$ of length Q samples such that:

$$\mathbf{y}[n] = \sum_{p=0}^{Q-1} \mathbf{W}[p]\, \mathbf{x}[n-p] \approx \mathbf{s}[n] \tag{1.6}$$

where the demixing filter length Q must be chosen sufficiently long to invert the mixing convolution, typically at least as long as the RIR itself. This time-domain approach is powerful but computationally expensive for long RIRs.

Figure 1.3 illustrates this multi-path propagation scenario in a reverberant room.

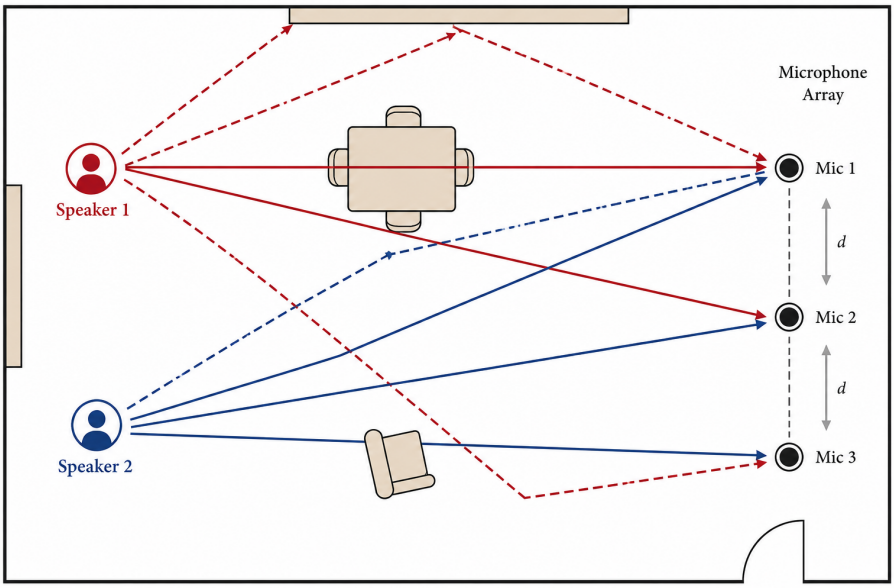


Figure 1.3: Multi-path propagation in a reverberant room.

Speech signals are inherently convolutive for two reasons. First, the room itself acts as a complex multi-path channel: every reflection adds a delayed, attenuated copy of each source to the mixture [8]. Second, the human vocal tract and head act as acoustic filters that shape each speaker's emission pattern, adding convolutive structure to the signal [9]. The RIR length P can span hundreds to thousands of samples depending on the room's reverberation time, making direct time-domain BSS extremely expensive.

1.2.3 The STFT Domain and the Narrowband Approximation

To make the convolutive problem tractable, modern BSS and localization systems work in the *Short-Time Fourier Transform* (STFT) domain. The STFT of a discrete-time signal $x[n]$ is defined as:

$$X(f, n) = \sum_{t=0}^{T-1} \mathbf{x}[nR + t]\, \gamma[t]\, e^{-j2\pi ft/F} \tag{1.7}$$

where T is the window length in samples, $\gamma[t]$ is an analysis window of length T (typically a Hann window), R is the hop size, F is the length of the Discrete Fourier Transform (DFT), $f \in \{0, \ldots, F-1\}$ is the frequency bin index, and n is the frame index [10].

Figure 1.4 shows an example time-frequency representation obtained from this transform.

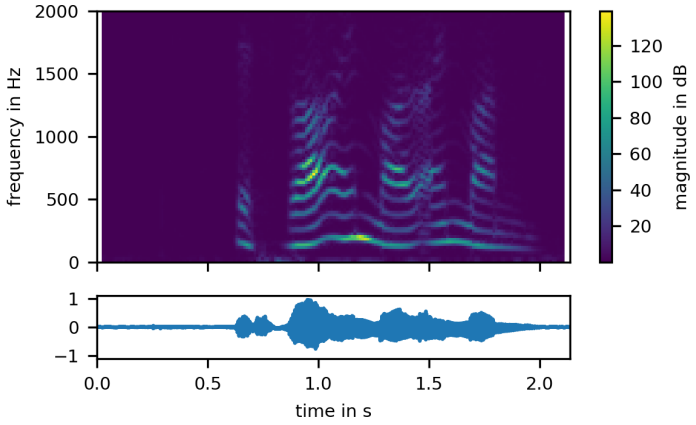


Figure 1.4: Spectrogram of a speech signal computed via the STFT in Eq. (1.7). Brighter regions indicate higher energy concentration. Harmonic structures and formant trajectories of voiced speech are clearly visible as time-varying horizontal bands.

The signal is reconstructed from its STFT coefficients by the *Inverse Short-Time Fourier Transform* (iSTFT), which synthesises and overlap-adds successive frames:

$$\mathbf{x}[n] = \frac{\displaystyle\sum_{n'} \left(\frac{1}{F} \sum_{f=0}^{F-1} X(f, n')\, e^{j2\pi f n/F} \right) \gamma[n - n'R]}{\displaystyle\sum_{n'} \gamma^2[n - n'R]} \tag{1.8}$$

where n' is the frame index over which the overlapping windowed segments are summed, mirroring the frame index n used in the forward STFT of (1.7).

Here, $\gamma[n]$ denotes the window function, commonly chosen as a Hann or Hamming window. For a window of length L, the symmetric Hann window is defined as:

$$\gamma[n] = 0.5 \left(1 - \cos\left(\frac{2\pi n}{L - 1} \right) \right), \quad 0 \le n \le L - 1 \tag{1.9}$$

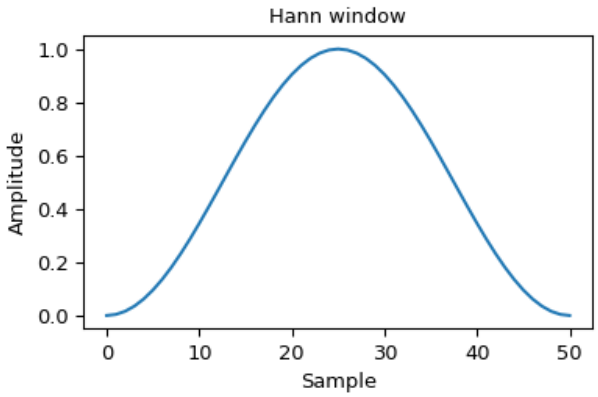


Figure 1.5: The Hann window function in the discrete-time domain.

There is a fundamental trade-off in the choice of window length, governed by the time-frequency uncertainty principle:

$$\Delta t\, \Delta f \ge \frac{1}{4\pi}, \tag{1.10}$$

where Δt and Δf denote, respectively, the achievable time and frequency resolutions for the chosen analysis window. A longer window improves frequency resolution at the expense of time resolution, and vice versa.

For perfect reconstruction, the analysis window $\gamma[t]$ and hop size R must satisfy the *Constant Overlap-Add* (COLA) condition[1]:

$$\sum_{l=-\infty}^{\infty} \gamma[n - kR] = C, \quad \forall\, n \in \mathbb{Z} \tag{1.11}$$

where C is a positive constant. When COLA holds, the windowed frames overlap and sum to a flat envelope, so no sample of $\mathbf{x}[n]$ is disproportionately amplified or suppressed during reconstruction. The Hann window satisfies the COLA condition for any hop size $R \leq T/2$; a 50 % overlap ($R = T/2$) is the standard choice in audio BSS [11, 10]. Figure 1.6 illustrates this frame-based windowing and overlap-add process.

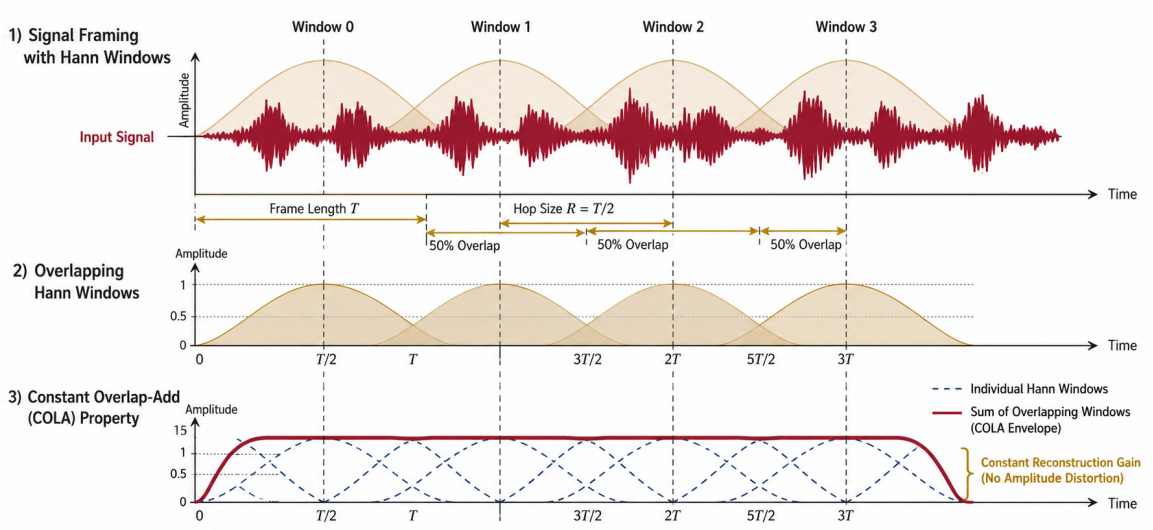


Figure 1.6: STFT frame-based analysis with the Hann window of length T. Successive frames are spaced by the hop size R and overlap by $T - R$ samples; each row shows one windowed frame $\mathbf{x}[nR + t]\gamma[t]$ prior to the DFT of Equation (1.7).

The key insight that makes STFT-domain processing powerful for convolutive BSS is the *narrowband approximation* [6, 11]: when the STFT frame length T is larger than the RIR length P, the time-domain convolution in (1.4) is well approximated by a *pointwise multiplication* in each frequency bin:

$$\mathbf{X}(f, n) \approx \mathbf{A}(f)\, \mathbf{S}(f, n) + \mathbf{N}(f, n) \tag{1.12}$$

where $\mathbf{X}(f, n) \in \mathbb{C}^M$, $\mathbf{S}(f, n) \in \mathbb{C}^L$, $\mathbf{A}(f) \in \mathbb{C}^{M \times L}$, and $\mathbf{N}(f, n) \in \mathbb{C}^M$ is the STFT of the additive noise $\mathbf{v}[n]$ at bin f and frame n. Each column $\mathbf{a}_l(f) \in \mathbb{C}^M$ is the *acoustic transfer function* (ATF) vector of source l at frequency bin f, representing the frequency-domain equivalent of the RIR vector.

Under the far-field assumption, when the source-to-array distance is much larger than the inter-microphone spacing, and in the absence of reverberation, the ATF simplifies to the

[1] COLA is closely related to the Nyquist zero-ISI criterion used in pulse-shaped digital communications: both impose a constant-sum condition on shifted basis functions to avoid reconstruction distortion.

steering vector:

$$\mathbf{d}_l(f) = \begin{bmatrix} e^{-j2\pi f \tau_{1l}/F} \\ \vdots \\ e^{-j2\pi f \tau_{Ml}/F} \end{bmatrix} \tag{1.13}$$

where τ_{ml} is the propagation delay from source l to microphone m in samples. The steering vector encodes the direction of source l through the phase pattern across the array.

A related quantity used extensively in this work is the *relative transfer function* (RTF), defined by normalising the ATF vector with respect to a reference microphone (indexed 1):

$$\mathbf{r}_l(f) = \frac{\mathbf{a}_l(f)}{a_{l,1}(f)} \tag{1.14}$$

where the subscript 1 denotes the chosen reference microphone and $a_{1l}(f)$ is the corresponding scalar entry of $\mathbf{a}_l(f)$. The RTF removes the dependence on the unknown source spectrum and retains only the spatial information encoded in the inter-microphone phase and level differences [6]. It is the object that the localization method of this work (Chapter 3) extracts from the BSS demixing matrix.

The narrowband approximation transforms a single hard convolutive BSS problem into $F/2$ independent instantaneous BSS problems, one per frequency bin, each of much lower dimension. This is the foundation of all frequency-domain BSS algorithms discussed in Section 1.5.

1.2.4 The BSS Mixing Scenarios and Ambiguities

Three mixing scenarios are distinguished depending on the relative numbers of sources L and sensors M:

- **Over-determined mixing** ($L < M$): more microphones than sources. The extra spatial diversity can improve separation quality, but standard $M = L$ algorithms must be extended, e.g. through adaptive data whitening.
- **Determined mixing** ($L = M$): equal numbers of sources and microphones. This is the standard scenario addressed by Independent Component Analysis (ICA) and Independent Vector Analysis (IVA).
- **Under-determined mixing** ($L > M$): more sources than microphones. The problem is ill-posed and requires additional assumptions such as source sparsity; it is outside the scope of this work.

Even when BSS succeeds in recovering the sources, two fundamental ambiguities remain that cannot be resolved from the observations alone [7, 3].

To see why, consider the instantaneous model. The observed mixture can be rewritten for any non-zero scalars $\alpha_1, \ldots, \alpha_L \in \mathbb{R} \setminus \{0\}$ as:

$$\mathbf{x}[n] = \sum_{l=1}^{L} \mathbf{a}_l \, s_l[n] = \sum_{l=1}^{L} \frac{\mathbf{a}_l}{\alpha_l} \Big(\alpha_l \, s_l[n]\Big) \tag{1.15}$$

Replacing each source $s_l[n]$ by $\alpha_l s_l[n]$ and the corresponding mixing column $\mathbf{a}_l$ by $\mathbf{a}_l/\alpha_l$ leaves $\mathbf{x}[n]$ identical. Since $\alpha_l \in \mathbb{R} \setminus \{0\}$, this scaling leaves the observations unchanged, so they carry no information about the absolute amplitude of each source. Figure 1.7 illustrates both the scaling and permutation ambiguities for a simple two-source, two-microphone scenario.

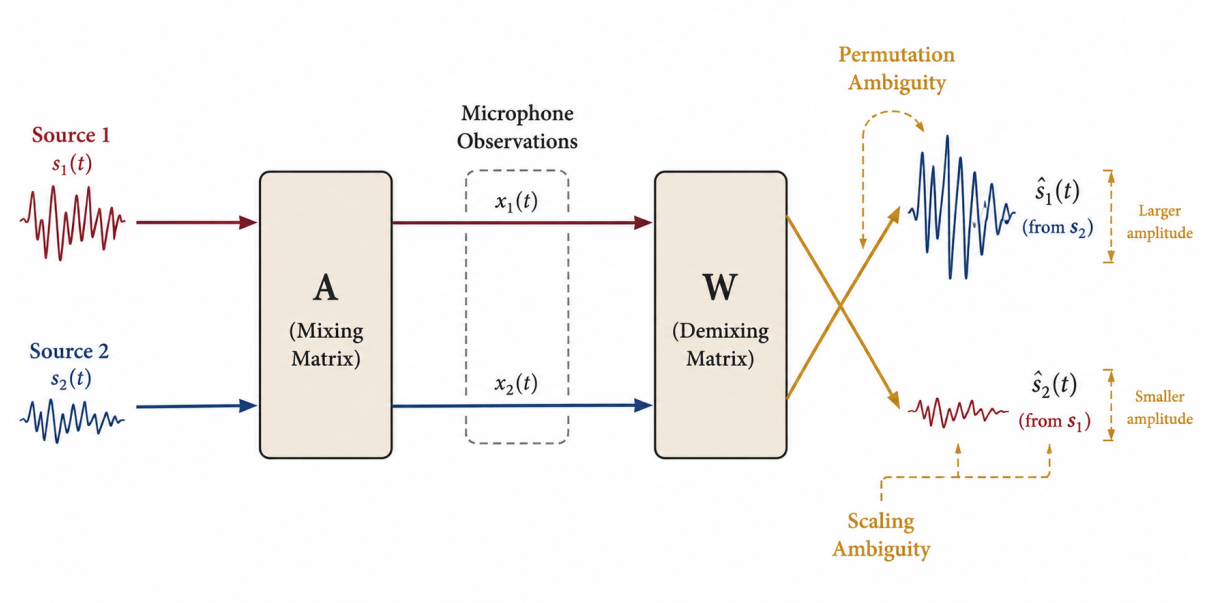


Figure 1.7: Illustration of BSS ambiguities for a simple two-source, two-microphone scenario.

Scaling Ambiguity: One can always multiply a source estimate by an arbitrary scalar and compensate by dividing the corresponding column of $\mathbf{A}$ by the same scalar, leaving the observations $\mathbf{x}$ unchanged. Therefore, BSS can only recover sources up to an unknown amplitude:

$$\hat{\mathbf{s}}[n] \approx \mathbf{D}\,\mathbf{s}[n] \tag{1.16}$$

where $\mathbf{D} = \mathrm{diag}(\alpha_1, \ldots, \alpha_L)$ is an unknown diagonal matrix. In practice, the *Minimal Distortion Principle* (MDP) is used to fix a canonical scaling that minimizes the distortion of each output relative to the original mixture [12]: the demixing matrix is rescaled as $\mathbf{W}_s(f) = \mathrm{diag}(\hat{\mathbf{A}}(f))\,\mathbf{W}(f)$, where $\hat{\mathbf{A}}(f) = \mathbf{W}(f)^{\#}$ is the pseudo-inverse estimate of the mixing matrix. A full derivation is given in Section 2.2.4 (Chapter 2).

Permutation Ambiguity: The order in which separated sources appear at the BSS output is arbitrary. The demixing matrix $\mathbf{W}$ can be pre-multiplied by any permutation matrix $\mathbf{P}$ without changing the separation criterion:

$$\hat{\mathbf{s}}[n] \approx \mathbf{P}\,\mathbf{s}[n] \tag{1.17}$$

In frequency-domain BSS, the permutation ambiguity is particularly severe: it can occur independently at each frequency bin, meaning that the signal reconstructed at one bin may correspond to source 1 while the same spectral position at an adjacent bin corresponds to source 2. Independent Vector Analysis (IVA), described in Section 1.5.3, addresses this by coupling the frequency bins through a multivariate source prior. Resolving the residual permutation ambiguity is also the central challenge of the BSS-based localization method presented in Chapter 3.

1.3 Acoustic Environment Characterisation

The acoustic environment in which the system operates directly determines its performance. A large reverberant room is far more challenging than an anechoic chamber for both BSS and localization. This section describes the physical and statistical properties of the room acoustic channel and introduces the key parameter used to characterise it.

1.3.1 The Room Impulse Response

The RIR $a_{ml}[p]$ introduced in Section 1.2.2 captures the complete acoustic path from source l to microphone m. Its shape can be decomposed into three successive parts, as illustrated in Figure 1.8 [6, 8]:

- **Direct path.** The first and strongest arrival, corresponding to the straight-line propagation from source to microphone. It carries the most reliable spatial information and arrives first with an amplitude inversely proportional to the source-microphone distance.
- **Early echoes.** The first few reflections off the room boundaries and furniture, typically arriving within the first 50 ms after the direct path. They are deterministic, directionally informative, and can be modelled explicitly.
- **Late reverberation.** Thousands of overlapping reflections that arrive after the early echoes and form an exponentially decaying diffuse tail. Because so many reflections superimpose, the late reverberant tail can be modelled statistically as a zero-mean Gaussian random process whose energy envelope decays exponentially [6].

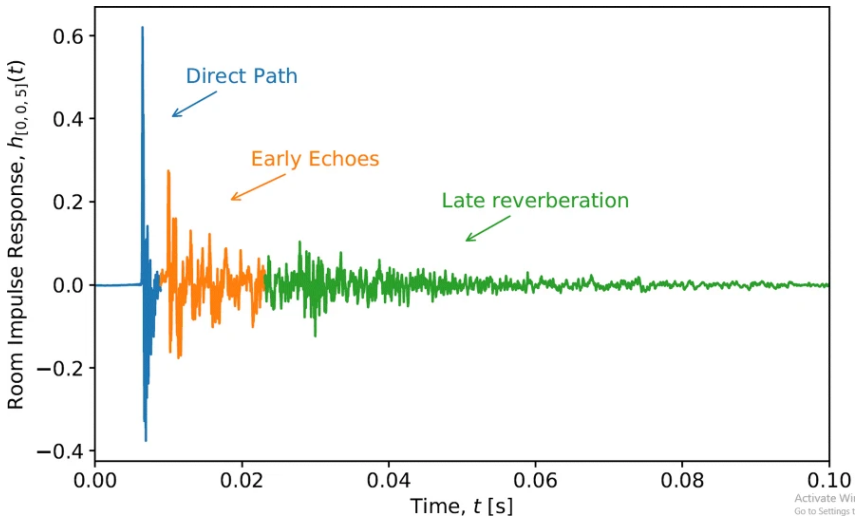


Figure 1.8: Schematic illustration of a room impulse response. Based on [6].

Two scalar quantities summarise the RIR and are used throughout this work:

- The **direct-to-reverberant ratio** (DRR): the ratio of the energy in the direct path to the energy in the rest of the RIR. High DRR means direct sound dominates, easier for both BSS and localization.
- The **reverberation time** RT60: defined in the next section.

1.3.2 Reverberation Time RT60

The reverberation time RT60 is defined as the time required for the reverberant energy to decay by 60 dB after the source stops emitting [13]. It depends solely on the room geometry and surface absorption properties. Typical values range from nearly 0 ms in outdoor environments, to 50 ms in a car, to 200–800 ms in office rooms, and above 1 s in large auditoriums [6].

RT60 can be estimated from a recorded RIR using the *Schroeder integral method* [13]. Let $a[n]$ denote the discrete-time room impulse response. The backward cumulative energy integral is computed as:

$$E[n] = \sum_{n'=n}^{\infty} a^2[n'] \tag{1.18}$$

This decay curve is plotted in dB and a linear regression is fitted to the portion between -5 dB and -35 dB (or -65 dB if dynamic range allows). The RT60 is then extrapolated from the fitted slope, as shown in Figure 1.9.

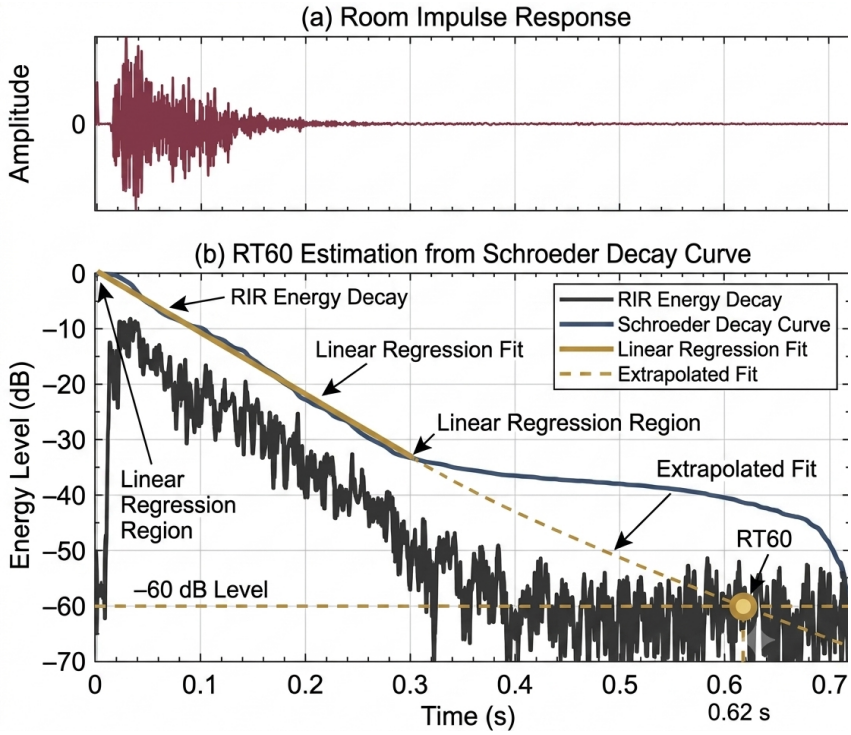


Figure 1.9: Schroeder decay curve of a room impulse response. The backward cumulative energy $E[n]$ is plotted on a logarithmic scale. The RT60 is determined by fitting a line to the linear decay region and extrapolating to the -60 dB level (dashed line). Adapted from [14].

In this project, RT60 estimation is implemented directly in the C++ application (described in Chapter 4) and is computed from the UMA-16 recordings before the BSS pipeline is run. It provides a live acoustic diagnostic: the user can see whether the current recording environment is favourable for separation and localization.

1.3.3 Impact of Reverberation on BSS and Localization

Reverberation is the primary acoustic challenge degrading both separation and localization performance. Its impact can be understood from two perspectives:

Impact on BSS. The narrowband approximation (1.12) holds only when the STFT frame length T exceeds the RIR length P. In rooms with RT60 above 300–400 ms, the RIR may span thousands of samples, making it impractical to use a frame length long enough for the approximation to hold exactly. It results in a modelling error in the estimated mixing matrix $\mathbf{A}(f)$, which degrades separation. Additionally, high reverberation violates the local stationarity assumption of the STFT, further reducing algorithm reliability.

Impact on Localization. In a reverberant environment, the ATF vector $\mathbf{a}_l(f)$ extracted from the BSS output is no longer equal to the anechoic steering vector $\mathbf{d}_l(f)$. It carries energy from all reflected paths, which smears the spatial information across directions. The estimated DoA is therefore biased toward the dominant reflections and away from the true direct-path direction. Frequency-averaging over multiple bins, described in Chapter 3, mitigates this effect by averaging out the random reverberant perturbations.

This direct link between RT60 and expected algorithm performance motivates the RT60 estimation module in the C++ platform: it provides a per-session prediction of how reliable the separation and localization outputs will be.

1.4 Pre-Processing

Before the BSS algorithm can be applied, the raw multichannel microphone signals must be conditioned to meet the algorithmic assumptions: signals should be zero-mean, reasonably stationary within processing frames, and free of silent segments that could destabilise the adaptive filter. This section surveys the pre-processing stages of the pipeline.

1.4.1 Voice Activity Detection

Voice activity detection (VAD) is a binary classification applied per time frame: *speech present* or *speech absent* [15]. It serves two purposes in an adaptive BSS system. First, it prevents the separation filter from being updated during silence, where no source information is available and the nonholonomic constraint (described in Chapter 2) can cause filter divergence. Second, it reduces unnecessary computation when no speech is present.

Three families of VAD methods are used in practice:

- **Energy-based VAD.** The short-time energy of the multichannel frame is compared to a fixed or adaptive threshold. Simple and computationally cheap, but sensitive to background noise and requires careful threshold calibration [15].
- **Statistical/Spectral feature-based VAD.** Features such as spectral entropy, or linear prediction residual energy provide better discrimination between speech and noise, at the cost of slightly higher computational load [16, 17].
- **Learning-based VAD.** GMMs, DNNs and RNNs can learn complex decision boundaries from large annotated datasets, achieving robust detection even at low SNR. However, they introduce significant latency and computational cost that may be prohibitive in real-time embedded deployments[18].

While simple energy-based methods are commonly adopted for their low latency (such as the approach used in the prior work by Bouznad [4]), they degrade severely in non-stationary noise and low Signal-to-Noise Ratio (SNR) conditions.

To achieve robust tracking in realistic acoustic environments, this project instead adopts a *statistical model-based VAD*, specifically, the Likelihood Ratio Test (LRT) formulated by Sohn et al. [17]. Rather than relying on a global energy threshold, this approach leverages a per-frequency-bin hypothesis test to reliably gate the BSS filter updates. The full mathematical formulation of this chosen method, alongside a comparative evaluation against an energy-based baseline, is detailed later in Section 2.6.1.

1.4.2 Signal Conditioning

Prior to STFT analysis, each microphone channel undergoes basic conditioning:

- **DC offset removal**: the time-averaged mean of each channel is subtracted to ensure zero-mean signals, a requirement for the whitening step.
- **Amplitude normalisation**: channels are normalised so that no single microphone dominates due to gain differences or proximity effects.
- **Bandpass filtering** (optional): limiting the signal to the speech band reduces out-of-band noise and is consistent with the speech intelligibility range [9].

1.4.3 Data Whitening: Motivation

Data whitening (or *sphering*) is a linear transformation $\mathbf{Q}$ applied to the multichannel STFT frames such that the resulting signal has an identity spatial covariance matrix. Let $\mathbf{R}_x = \mathbb{E}[\mathbf{x}[n]\mathbf{x}[n]^H]$ denotes the spatial covariance matrix of the input signal $\mathbf{x}[\mathbf{n}]$. Whitening seeks a transform satisfying:

$$\mathbf{Q}\,\mathbb{E}\Big[\mathbf{x}[n]\mathbf{x}[n]^H\Big]\,\mathbf{Q}^H = \mathbf{I}_L \tag{1.19}$$

Whitening serves three purposes in the BSS pipeline. First, it simplifies the optimization landscape. Once the data is white, the demixing matrix needs only to be unitary, reducing the search space significantly and accelerating convergence. Second, it performs dimensionality reduction from M microphones to L sources, discarding the noise subspace and retaining only the source subspace. Third, and most importantly for this project, *adaptive whitening* allows the system to exploit all $M = 16$ channels of the UMA-16 microphone array even when $L < M$ sources are present, capturing the full spatial diversity of the array.

This last point is the key contribution of Merah & Ghecham [3], who introduced adaptive whitening into the online natural gradient IVA framework and demonstrated its benefits over the standard $M = L$ restriction. The full derivation of the adaptive whitening procedure, including the recursive covariance estimation and eigenvalue decomposition at each frame, is given in Chapter 2.

1.5 Blind Source Separation: Survey

This section traces the evolution of BSS algorithms from the foundational work on Independent Component Analysis (ICA) to the adaptive Independent Vector Analysis (IVA) framework that this project implements and extends. The goal is to situate each contribution in the broader landscape and to explain why each step forward was necessary.

Scope of coverage: Most algorithms in this survey, particularly ICA and standard IVA, are designed for the *determined mixing* scenario where the number of microphones M equals the number of sources L. In the *over-determined* case ($M > L$, as in the present project with 16 microphones and 2–3 active sources), adaptive data whitening (Section 1.5.4) projects the observations onto the source subspace and reduces the problem to the determined case. The *under-determined* scenario ($M < L$) is fundamentally more difficult and requires different algorithmic approaches outside the scope of this work.

1.5.1 Independent Component Analysis

Independent Component Analysis (ICA) is the statistical framework at the origin of modern BSS. It seeks a linear transformation $\mathbf{W}$ of the observed mixture $\mathbf{x}$ such that the outputs $\mathbf{y} = \mathbf{W}\mathbf{x}$ are as *statistically independent* as possible. If the sources are indeed independent and at most one is Gaussian, this principle is sufficient to recover them up to scaling and permutation [19, 20].

The concept of ICA for BSS was first formalised by Comon [19] using mutual information minimisation as the contrast function. Bell & Sejnowski [21] independently proposed the *Infomax* approach: maximising the joint entropy of a nonlinearly transformed output is equivalent to minimising mutual information among its components, yielding the same demixing criterion. Amari et al. [22] introduced the *natural gradient* as a more efficient update direction than the ordinary gradient, greatly accelerating convergence. Hyvärinen & Oja [23, 24] proposed the *FastICA* fixed-point algorithm, which converges in far fewer iterations than gradient descent by exploiting the kurtosis of the source distributions.

Despite these advances, ICA under the instantaneous model is not directly applicable to real room recordings, where the convolutive mixing structure of (1.4) invalidates the instantaneous assumption.

1.5.2 Frequency-Domain ICA and the Permutation Problem

The natural extension to convolutive mixtures is to apply ICA independently in each frequency bin of the STFT. Under the narrowband approximation (1.12), each bin f presents an instantaneous complex-valued ICA problem of dimension $M \times L$, solvable independently. This *Frequency-Domain ICA* (FDICA) approach was studied by Sawada et al. [25] among others.

However, FDICA introduces a critical practical problem: the permutation ambiguity (Section 1.2.4) applies *independently* to each frequency bin. The demixing matrix $\mathbf{W}(f)$ found at bin f may assign output channel 1 to speaker A and channel 2 to speaker B, while at bin $f + 1$ the assignment is reversed. When the STFT outputs are summed to reconstruct the time-domain signals, the result is a mixture rather than a separated source.

Post-processing methods, inter-frequency correlation, direction-of-arrival consistency, or clustering, were proposed to realign the permutations across bins [25], but they add fragility and complexity.

1.5.3 Independent Vector Analysis (IVA)

Independent Vector Analysis (IVA), proposed by Kim et al. [26], provides an elegant solution to the permutation problem. Instead of modelling each source as a scalar random variable at each frequency bin, IVA models each source as a *random vector* across all frequency bins: $\mathbf{y}_l = [Y_l(1), Y_l(2), \ldots, Y_l(F)]^T$. By specifying a *multivariate* probability density function for each source vector that encourages statistical dependence *within* a source across frequencies, IVA implicitly couples the frequency bins and prevents permutation misalignment.

The standard choice for the multivariate source prior is the spherically symmetric Laplace distribution:

$$p(\mathbf{y}_l) \propto \exp\left(-\sqrt{\sum_{f=1}^{F} |Y_l(f)|^2}\right) \tag{1.20}$$

where the sum runs over all F frequency bins of the STFT, and $|Y_l(f)|$ denotes the magnitude of the l-th source's STFT coefficient at bin f. The gradient of the corresponding log-likelihood with respect to $\mathbf{W}(f)$ yields a natural gradient update rule applied simultaneously across all bins, ensuring frequency-consistent permutation.

The IVA cost function is the mutual information between the source vectors:

$$\mathcal{J}_{\text{IVA}} = -\sum_{f} \log|\det \mathbf{W}(f)| - \sum_{l} \log p(\mathbf{y}_l) + \text{const} \tag{1.21}$$

Minimising this over $\mathbf{W}(f)$ for all f jointly recovers source-consistent demixing matrices. The scaling ambiguity is resolved using the *Minimal Distortion Principle* (MDP) [12]. Lee et al. [27] proposed FastIVA, a fixed-point acceleration of the IVA update that converges significantly faster than the natural gradient version.

1.5.4 Adaptive Online IVA

Batch IVA requires access to all N frames of the mixture simultaneously, making it unsuitable for real-time operation. The adaptive (online) IVA framework processes the mixture frame by frame, updating the demixing matrix with each new observation.

Kim [28] proposed the first *adaptive natural gradient IVA*, using an instantaneous stochastic gradient approximation of the batch update and introducing a normalisation factor ξ to make the learning rate robust against sudden changes in input energy:

$$\mathbf{W}(f,n) = \mathbf{W}(f,n-1) + \frac{\eta}{\sqrt{\zeta(f,n)+\varepsilon}} \Delta\mathbf{W}(f,n) \tag{1.22}$$

where $\Delta\mathbf{W}(f,n)$ is the natural gradient, η is the learning rate, $\zeta(f,n)$ is a recursively estimated, frame-wise power-normalisation term that tracks the local signal energy, and ε is a small positive constant that prevents division by zero when $\zeta(f,n)$ is close to zero. A limitation of Kim's formulation is that it requires $M = L$: the number of

microphones must equal the number of sources. This restriction limits the framework to the *determined mixing* scenario. In the *over-determined* case where $M > L$ (more microphones than sources), a more sophisticated approach is required to exploit the extra spatial diversity without destabilising the adaptive filter. Taniguchi et al. [29] proposed an auxiliary-function-based adaptive IVA (AuxIVA) that achieves faster and more stable convergence by using majorisation-minimisation, at the cost of higher computational load per frame. Scheibler and Ono [30] extended AuxIVA to the overdetermined case, extending its applicability to practical multichannel recording scenarios. Nakashima & Ono [31] subsequently proposed an inverse-free rank-1 update variant that reduces the complexity significantly.

Merah & Ghecham [3] introduced *adaptive data whitening* as a pre-processing stage to the online NG-IVA, which is the direct predecessor of the present work. Their adaptive whitening, computed recursively at each frame using eigenvalue decomposition, lifts the $M = L$ constraint and allows the full 16 microphone channels of the UMA-16 to be exploited even when only 2 or 3 sources are active.

1.5.5 Advanced IVA Variants

Two more recent variants of IVA improve upon the natural gradient approach in convergence speed and numerical stability, and are studied in this project as potential substitute for or complements to the NG-IVA core:

Auxiliary-function-based adaptive IVA (AuxIVA). AuxIVA replaces the natural-gradient update with an auxiliary-function optimisation framework based on majorisation–minimisation. The resulting update rules require no step-size tuning, guarantee a monotonic decrease of the objective function, and generally achieve faster and more stable convergence than conventional NG-IVA [29].

Overdetermined IVA (OverIVA). OverIVA extends the IVA framework to overdetermined scenarios, where the number of microphones exceeds the number of sources. By modelling the observations as a combination of non-Gaussian sources and a Gaussian background, and by exploiting a reduced-parameter demixing matrix formulation, it achieves separation performance comparable to conventional IVA while significantly reducing computational complexity in multichannel configurations [30].

1.6 Sound Source Localization: Survey

Sound source localization (SSL) aims to estimate the direction or position of one or more acoustic sources from recordings acquired by a microphone array. Over the past decades, a wide range of localization techniques has been developed, ranging from time-delay estimation methods and high-resolution subspace algorithms to more recent approaches that exploit information extracted from blind source separation (BSS) systems.

The objective is to highlight their underlying assumptions, strengths, and limitations, and to motivate the BSS-based localization framework adopted in the proposed system.

1.6.1 Time-Delay Based Methods

The most widely used localization paradigm exploits the difference in arrival time of the source signal at different microphones. The *Time Difference of Arrival* (TDoA) between microphones i and j for a source at position $\mathbf{p}_s$ is:

$$\Delta\tau_{ij} = \frac{1}{c}\left(\|\mathbf{p}_s - \mathbf{r}_j\| - \|\mathbf{p}_s - \mathbf{r}_i\|\right) \tag{1.23}$$

where c is the speed of sound, and $\mathbf{r}_i, \mathbf{r}_j \in \mathbb{R}^3$ are the known positions of microphones i and j in the array. Under the far-field approximation, this simplifies to a linear function of the unit direction vector $\mathbf{u}(\theta, \phi)$ [32].

Generalised Cross-Correlation (GCC). The TDoA between a pair of microphones is estimated by finding the peak of the cross-correlation function in the frequency domain:

$$R_{ij}(\tau) = \int_{-\infty}^{\infty} \Psi(f)\, G_{ij}(f)\, e^{j2\pi f\tau}\, df \tag{1.24}$$

where $G_{ij}(f) = X_i(f)X_j^*(f)$ is the cross-power spectral density and $\Psi(f)$ is a frequency-domain weighting function [32]. The *PHAT* (Phase Transform) weighting $\Psi_{\text{PHAT}}(f) = 1/|G_{ij}(f)|$ whitens the cross-spectrum, sharpening the correlation peak and improving robustness to reverberation [33]. Figure 1.10 summarises this GCC-PHAT pipeline.

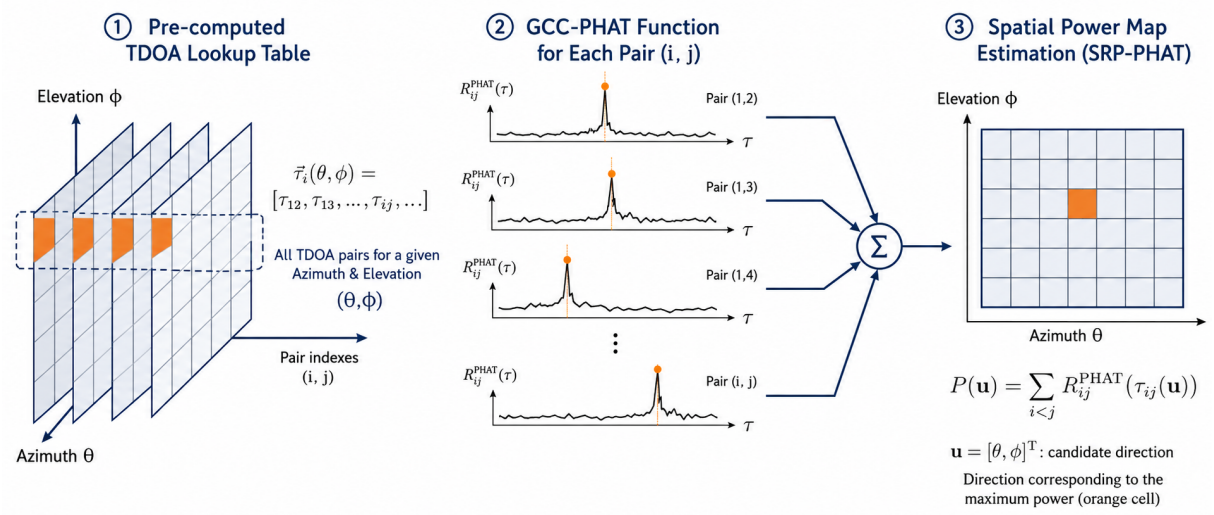


Figure 1.10: GCC-PHAT algorithm pipeline. For each microphone pair, the cross-correlation function $R_{ij}^{\text{PHAT}}(\tau)$ of Equation (1.24) is computed and stored in a TDoA lookup table (left panel); the per-pair TDoA estimates are then aggregated over candidate directions $\mathbf{u}$ to form the SRP-PHAT spatial power map of Equation (1.25) (right panel), whose peak gives the DoA estimate.

Steered Response Power with PHAT (SRP-PHAT). SRP-PHAT [34] aggregates GCC-PHAT values across all microphone pairs for each candidate direction $\mathbf{u}$:

$$P(\mathbf{u}) = \sum_{i<j} R_{ij}^{\text{PHAT}}(\tau_{ij}(\mathbf{u})) \tag{1.25}$$

The direction with the maximum accumulated response is taken as the DoA estimate. SRP-PHAT is robust and widely used, and was implemented by Bouznad [4].

A key limitation of SRP-PHAT in multi-speaker scenarios is that the spatial map $P(\mathbf{u})$ accumulates contributions from all active sources simultaneously, creating cross-talk between speakers. Peaks corresponding to different speakers can merge or mask each other,

1.6.2 Subspace Methods

Subspace methods exploit the eigenstructure of the spatial covariance matrix $\mathbf{R}_{xx}(f) = \mathbb{E}[\mathbf{x}(f)\mathbf{x}^H(f)]$ to separate the signal and noise subspaces.

MUSIC. The MUltiple SIgnal Classification algorithm [35] decomposes $\mathbf{R}_{xx}$ into signal and noise eigenvectors. Since the steering vector of any source is orthogonal to the noise subspace, the DoA is found by searching for the direction that minimizes the projection of the steering vector onto the noise subspace. MUSIC offers high angular resolution but requires knowledge of the number of sources and is sensitive to array calibration errors.

ESPRIT. ESPRIT [36] exploits the shift invariance of the array to estimate the DoA without an exhaustive grid search, at lower computational cost than MUSIC. It is limited to regular array geometries.

Both methods are mentioned for completeness; they are not the primary localization approach of this project.

1.6.3 BSS-Based Localization

The localization method developed in this project takes a fundamentally different approach from classical TDoA and subspace methods: it extracts spatial information *directly from the BSS demixing matrix*.

After IVA converges with MDP rescaling, the relationship between the demixing matrix and the mixing matrix is [6, 12]:

$$\mathbf{W}_s(f) \cdot \mathbf{A}(f) = \mathrm{diag}(\mathbf{A}(f)) \tag{1.26}$$

This follows directly from the Minimal Distortion Principle introduced in Section 1.2.4 and derived in full in Section 2.2.4: rescaling $\mathbf{W}(f)$ by the diagonal of the pseudo-inverse mixing-matrix estimate forces the diagonal entries of $\mathbf{W}_s(f)\mathbf{A}(f)$ to equal those of $\mathbf{A}(f)$ itself, while the off-diagonal entries vanish because, after IVA convergence, $\mathbf{W}(f)\mathbf{A}(f)$ is a permuted diagonal matrix up to residual estimation error. Taking the pseudo-inverse of $\mathbf{W}_s(f)$ gives an estimate of the mixing matrix:

$$\hat{\mathbf{A}}(f) = \mathbf{W}_s(f)^{\#} \tag{1.27}$$

Each column $\hat{\mathbf{a}}_l(f)$ of $\hat{\mathbf{A}}(f)$ is an estimate of the ATF vector of source l, the spatial fingerprint of that source across the 16 microphones at frequency f. Crucially, each column corresponds to exactly one source, unlike SRP-PHAT's mixed spatial map.

Given $\hat{\mathbf{a}}_l(f)$, three complementary DoA estimation strategies are applied in this project:

- **Steering-Vector Dictionary Matching.** Each estimated ATF column $\hat{\mathbf{a}}_l(f)$ is compared against a precomputed dictionary of theoretical steering vectors covering the azimuth-elevation hemisphere for the UMA-16 geometry. The candidate direction that minimizes a coherence-based distance to the observed ATF vector is taken as the DoA estimate for source l.

- **Spatial Likelihood Map.** The extracted RTF vectors $\tilde{\mathbf{a}}_l(f)$ are used to build a per-source spatial likelihood map over the angular grid. Averaging this map coherently across frequency bins concentrates the peak at the true DoA and suppresses reverberant side-lobes.

- **Inter-Element Phase-Gradient (Phase-Slope).** For a far-field source, the phase of $\hat{\mathbf{a}}_l(f)$ varies approximately linearly across the microphone elements, with a slope proportional to the inter-element propagation delay and hence to the direction of arrival. The DoA can therefore be read off directly from the inter-element phase differences, aggregated robustly across frequency bins using strategies ranging from a simple mean or energy-weighted mean to a least-squares sinusoidal regression. This approach avoids dictionary storage and is particularly suited to online operation.

1.6.4 Audio-Visual Fusion and Camera Overlay

Once a DoA estimate is available as a 3D unit direction vector $\mathbf{d}_l$, it can be projected onto the 2D image plane of a co-located camera using the pinhole camera model:

$$\mathbf{p}_l = \pi\left(\mathbf{K}\,[\mathbf{R}|\mathbf{t}]\,\begin{bmatrix}\mathbf{d}_l\\1\end{bmatrix}\right) \tag{1.28}$$

where $\mathbf{K}$ is the camera intrinsic matrix, $(\mathbf{R}, \mathbf{t})$ is the extrinsic microphone-to-camera transform, and $\pi(\cdot)$ is the perspective division operator. The resulting pixel coordinate $\mathbf{p}_l$ marks the estimated position of speaker l on the live video stream.

Bouznad [4] implemented this projection pipeline in MATLAB, driven by SRP-PHAT DoA estimates. The present project extends it to C++, integrates it with the BSS-based DoA extraction, and adds interactive speaker selection through a graphical interface. The full camera calibration procedure and projection implementation are described in Chapter 4.

1.7 Post-Processing

In practical acoustic environments, the outputs of a BSS system are rarely free of artefacts. Reverberation, background noise, finite filter lengths, and estimation errors can all degrade separation quality.

As illustrated in Figure 1.11, reflected propagation paths may allow part of an interfering speaker to leak into the output channel associated with the desired source. Such residual cross-talk, together with other artefacts introduced during the separation process, motivates the use of a dedicated post-processing stage.

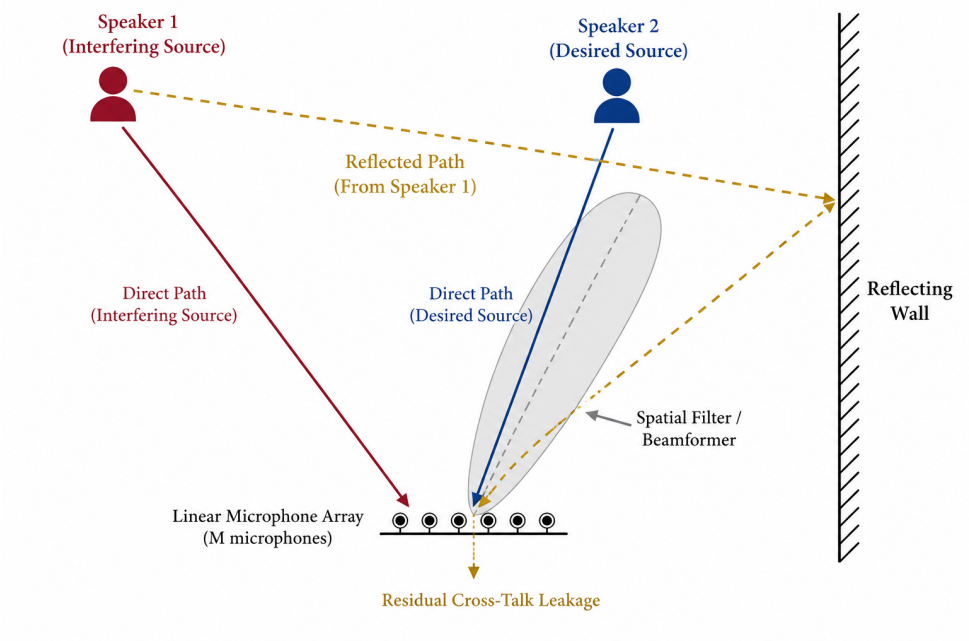


Figure 1.11: Residual cross-talk introduced by reverberation.

1.7.1 Why Post-Processing Is Needed

Three types of artefact commonly survive the BSS stage [6]:

- **Residual cross-talk.** In reverberant conditions, the demixing filter $\mathbf{W}$ cannot completely cancel the contribution of interfering speakers, especially at low SNR or high RT60. Traces of the other speakers remain audible in each output channel.
- **Musical noise.** The STFT-domain processing creates isolated non-zero time-frequency bins in otherwise quiet regions, which are perceived as a metallic, tonal background noise.
- **Scaling and phase errors.** Even after MDP rescaling, small amplitude and phase inconsistencies can remain due to model mismatch or numerical precision.

Gannot et al. [6] treat postfiltering as a fourth, optional stage of the multichannel processing pipeline, applied after spatial filtering to reduce these residuals further.

1.7.2 Postfiltering Approaches

Two main families of postfilter can be distinguished. *Single-channel postfilters* operate independently on each BSS output channel, applying a frequency-dependent gain or mask derived from the local signal statistics. They adapt quickly to changes in interference and impose minimal computational overhead. *Multichannel postfilters* jointly exploit residual inter-channel spatial information to further suppress cross-talk, at the cost of higher complexity. In this project, the focus is on single-channel postfilters, which offer a practical trade-off between residual noise suppression and speech distortion [6]. The main approaches in this family are:

Wiener Postfilter. A frequency-dependent real-valued gain $G(f, n) \in [0, 1]$ is applied to the beamformer or BSS output spectrum:

$$\hat{S}_l(f, n) = G_l(f, n)\, Y_l(f, n) \tag{1.29}$$

where $G_l(f, n)$ is estimated from the ratio of the target source power to the total output power at each bin. The Wiener postfilter is the most widely used and offers a good trade-off between noise suppression and speech distortion [6].

Spectral Subtraction. An estimate of the residual noise spectrum is subtracted from the output spectrum. Simpler to implement than the Wiener filter but more prone to musical noise artefacts.

Binary and Soft Time-Frequency Masking. A binary mask sets each TF bin to 1 if the target source is dominant at that bin and 0 otherwise. Soft masking uses an estimated source probability as a continuous gain. These approaches can achieve very high interference suppression but risk introducing distortion when the mask is inaccurate.

1.7.3 MDP Rescaling as Implicit Post-Processing

It is worth noting that the MDP rescaling applied inside the IVA algorithm is itself a form of post-processing: it corrects the scaling ambiguity so that the separated output preserves the correct amplitude relative to the original mixture. Formally, the rescaled separation matrix is defined as $\mathbf{W}_{\text{MDP}}(f) = \mathbf{D}(f)\mathbf{W}(f)$, where $\mathbf{D}(f) = \text{diag}(\mathbf{W}^{\#}(f))$ (or its conjugate transpose variant depending on the implementation) acts as the scaling correction matrix. This mechanism is already integrated into the adaptive IVA engine described in Chapter 2 and requires no additional processing stage.

1.8 Conclusion and Positioning of This Work

This chapter establishes the theoretical and algorithmic foundations required to understand and position the contributions of this project. The acoustic signal models (Section 1.2) defined the mathematical framework, from the convolutive mixing model to the narrowband STFT approximation, the ATF and RTF vectors, and the two fundamental BSS ambiguities. The room acoustics section (Section 1.3) characterised the reverberant environment and introduced RT60 as the key diagnostic parameter. The pre-processing, BSS, localization, and post-processing surveys (Sections 1.4–1.7) puts the methods of this project in the context of the existing literature.

The position of this work in the LDCCP/ENP research line can be summarised as follows:

- **BSS contribution.** This project extends Merah & Ghecham [3] by porting the adaptive NG-IVA with whitening from MATLAB to a real-time C++ platform, and by exploring advanced variants (AuxIVA, OverIVA) for improved convergence and noise robustness.

- **Localization contribution.** This project goes beyond Bouznad [4] by replacing the SRP-PHAT DoA estimator with a BSS-based approach that extracts source-specific ATF vectors directly from the demixing matrix $\mathbf{W}$, providing a speaker-separated spatial map free of inter-source cross-talk.

- **System contribution.** Both contributions are integrated into a single real-time C++ application with a graphical interface, camera overlay, interactive speaker selection, and a built-in RT60 estimator, bringing the system from a MATLAB research prototype to a deployable platform.

The following two chapters derive and validate the BSS and localization methods in detail. Chapter 2 presents the full mathematical derivation of adaptive IVA with whitening and its advanced variants, followed by simulation-based validation. Chapter 3 develops the BSS-based DoA extraction pipeline and validates it against classical SRP-PHAT. Chapter 4 then presents the complete C++ system and its experimental evaluation.

Chapter 2

Blind Source Separation: From ICA to Adaptive IVA

This chapter details the algorithmic framework used to solve the convolutive speech mixture problem. Resolving the fundamental challenges of frequency-domain separation, specifically the permutation and scaling ambiguities, requires techniques ranging from foundational Independent Component Analysis (ICA) to advanced, multivariate Independent Vector Analysis (IVA). Understanding the core separation engine implemented in this project requires familiarity with these statistical models, their adaptive online derivations, and the necessary pre- and post-processing stages that ensure robust performance.

2.1 Independent Component Analysis

Independent Component Analysis (ICA) is the statistical framework from which all modern BSS algorithms descend. Understanding its core principle, and its fundamental limitation for convolutive speech, is necessary to appreciate why IVA was developed.

2.1.1 The ICA Model and Its Assumptions

In its simplest form, ICA assumes that the M observed signals are *instantaneous* linear mixtures of L statistically independent source signals:

$$\mathbf{x}[n] = \mathbf{A}\,\mathbf{s}[n] \tag{2.1}$$

where $\mathbf{x}[n] \in \mathbb{C}^M$ is the observation vector, $\mathbf{s}[n] \in \mathbb{C}^L$ is the source vector, and $\mathbf{A} \in \mathbb{C}^{M\times L}$ is the unknown mixing matrix. The goal is to find a demixing matrix $\mathbf{W} \in \mathbb{C}^{L\times M}$ such that:

$$\mathbf{y}[n] = \mathbf{W}\,\mathbf{x}[n] \approx \mathbf{s}[n] \tag{2.2}$$

ICA is well-posed under four assumptions [20, 19]:

(i) **Mutual independence:** The sources are mutually statistically independent.

(ii) **Non-Gaussianity:** At most one source has a Gaussian distribution (this is the condition that makes ICA identifiable).

(iii) **Overdetermination:** The mixing is square or overdetermined ($M \geq L$).

(iv) **Zero-mean:** The signals are zero-mean.

When these conditions hold, the demixing matrix can be identified up to the permutation and scaling ambiguities described in Chapter 1: BSS succeeds when $\mathbf{WA} = \mathbf{PD}$, where $\mathbf{P}$ is an arbitrary permutation matrix and $\mathbf{D}$ is an arbitrary invertible diagonal matrix.

2.1.2 Contrast Function, Mutual Information, and the Natural Gradient

The separation criterion is based on a *contrast function* $C(\mathbf{y})$: a functional that measures statistical dependence among the outputs and whose extrema correspond to successful separation. Among all possible choices, *mutual information* provides the most rigorous foundation [19]:

$$I(\mathbf{y}) = \sum_{l=1}^{L} H(y_l) - H(\mathbf{y}) \tag{2.3}$$

where $H(\cdot)$ denotes differential entropy. $I(\mathbf{y}) \geq 0$ always, and equals zero if and only if the outputs are mutually independent, precisely the separation condition. Minimising $I(\mathbf{y})$ over $\mathbf{W}$ therefore drives the outputs toward independence.

This is equivalent, in the maximum-likelihood sense, to maximising $\log|\det \mathbf{W}| + \sum_l \log p(y_l)$ under a known source prior $p(\cdot)$ [21, 6]. The gradient of the mutual information cost with respect to $\mathbf{W}$ is:

$$\nabla_{\mathbf{W}} I = \left(\mathbf{W}^{-H} - \boldsymbol{\phi}(\mathbf{y})\mathbf{x}^H\right) \tag{2.4}$$

where $\boldsymbol{\phi}(\mathbf{y}) = -\partial \log p(\mathbf{y})/\partial \mathbf{y}^*$ is the *score function* of the assumed source distribution. Applying the *natural gradient* [22], pre-multiplying by $\mathbf{W}^H\mathbf{W}$ to account for the Riemannian geometry of the space of invertible matrices, yields the elegant update rule:

$$\mathbf{W} \leftarrow \mathbf{W} + \mu \left(\mathbf{I} - \boldsymbol{\phi}(\mathbf{y})\mathbf{y}^H\right)\mathbf{W} \tag{2.5}$$

where $\mu > 0$ is the step size. This is the celebrated *Infomax* learning rule of Bell & Sejnowski [21]. Its convergence depends critically on the choice of μ: too large causes divergence, too small causes impractically slow convergence, a tension that motivates the auxiliary-function variants developed in Section 2.4.

2.1.3 Data Whitening

Before applying ICA, the observed data is *whitened*: linearly transformed so that its spatial covariance equals the identity:

$$\mathbb{E}\left[\tilde{\mathbf{x}}[n]\,\tilde{\mathbf{x}}[n]^H\right] = \mathbf{I}_L \tag{2.6}$$

The whitening matrix is obtained via eigenvalue decomposition (EVD) of the sample covariance matrix $\hat{\mathbf{R}} = \mathbb{E}[\mathbf{x}\mathbf{x}^H] = \tilde{\mathbf{E}}\tilde{\mathbf{D}}\tilde{\mathbf{E}}^H$, retaining only the L dominant eigenpairs:

$$\mathbf{Q} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{E}}^H, \qquad \tilde{\mathbf{x}}[n] = \mathbf{Q}\mathbf{x}[n] \tag{2.7}$$

Whitening has two effects that simplify the ICA optimisation [20, 6]:

(i) It forces the effective mixing matrix $\tilde{\mathbf{A}} = \mathbf{QA}$ to be unitary, so the natural gradient needs only to search over rotations rather than all invertible matrices

(ii) When $M > L$, it simultaneously reduces the data dimension from M channels to L source dimensions, discarding the noise subspace. The second property is the key that Merah & Ghecham [3] exploited to extend the adaptive IVA framework to the overdetermined case $M = 16 > L$, as developed fully in Section 2.3.

2.2 Independent Vector Analysis

ICA in the form of Section 2.1 operates on instantaneous mixtures. For convolutive speech mixtures, the natural extension is to apply ICA independently at each frequency bin of the STFT, an approach known as Frequency-Domain ICA (FDICA). However, this introduces a critical failure mode that ICA alone cannot resolve: the frequency-wise permutation problem. Independent Vector Analysis (IVA) [26, 37] was designed precisely to eliminate this problem.

2.2.1 The Permutation Problem in FDICA

Under the narrowband approximation (Section 1.2.3), the convolutive mixture at each frequency bin f becomes an instantaneous ICA problem:

$$\mathbf{X}(f, n) = \mathbf{A}(f)\, \mathbf{S}(f, n) \tag{2.8}$$

Applying ICA independently to each bin f yields a demixing matrix $\mathbf{W}(f)$ that is valid for that bin. However, the permutation ambiguity is resolved independently at each f: the ordering of the L outputs at bin f may differ arbitrarily from the ordering at bin $f + 1$. When these per-bin solutions are recombined by the inverse STFT, each output channel receives spectral components from different sources at different frequencies, the reconstructed signal is a mixture rather than a separated source. This is the *frequency-wise permutation problem*, illustrated in Figure 2.1.

Post-processing methods that align permutations across bins using inter-frequency correlation or direction-of-arrival consistency [25] were proposed, but they are fragile and add significant complexity. IVA takes a fundamentally different route: it prevents permutation misalignment from occurring in the first place.

2.2.2 The IVA Principle: Multivariate Source Prior

IVA [26] resolves the permutation problem by changing the statistical model. Instead of treating each source as a scalar random variable at each frequency bin, IVA models each source as a *random vector* grouping all frequency components together:

$$\mathbf{y}_l = \Big[Y_l(1),\ Y_l(2),\ \ldots,\ Y_l(F)\Big]^T \in \mathbb{C}^F \tag{2.9}$$

A *multivariate* probability density function is then assigned to each such vector. This inter-frequency coupling has a crucial consequence: if the demixing matrices $\{\mathbf{W}(f)\}_{f=1}^{F}$

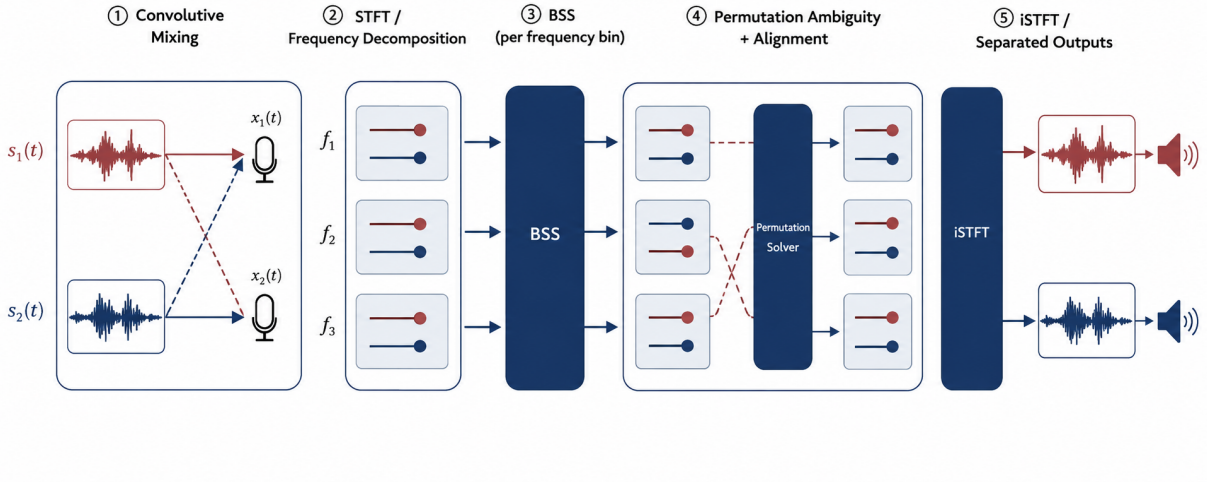


Figure 2.1: The frequency-wise permutation problem in FDICA. ICA applied independently at each frequency bin f produces a valid demixing $\mathbf{W}(f)$, but the assignment of output channels to sources is arbitrary and inconsistent across bins. The ISTFT reconstruction of each output therefore mixes spectral components from different sources.

are estimated jointly to maximise the independence of all $\mathbf{y}_l$ simultaneously, the permutation at each bin is implicitly constrained to be consistent with every other bin, because inconsistent permutations would violate the assumed multivariate structure.

The standard choice of multivariate prior for speech is the *spherically symmetric Laplace distribution* [26, 6]:

$$p(\mathbf{y}_l) \propto \exp(-\|\mathbf{y}_l\|_2) = \exp\left(-\sqrt{\sum_{f=1}^{F} |Y_l(f)|^2}\right) \tag{2.10}$$

This prior belongs to the family of generalised Gaussian distributions with shape parameter $q = 1$, whose distribution in the STFT magnitude domain closely matches empirical speech statistics [6]. The score function (the gradient of the negative log-prior) is:

$$\phi_l^{(f)} = \frac{\partial(-\log p(\mathbf{y}_l))}{\partial Y_l^*(f)} = \frac{Y_l(f)}{\|\mathbf{y}_l\|_2} \tag{2.11}$$

Note that $\phi_l^{(f)}$ depends on *all* frequency components of source l through the norm $\|\mathbf{y}_l\|_2$. It is precisely this cross-frequency coupling that ties the demixing matrices across bins and prevents permutation misalignment.

2.2.3 The IVA Cost Function and Batch Update

The IVA cost function is the mutual information between the source vectors $\{\mathbf{y}_l\}$, which, using the Laplace prior, takes the form [38]:

$$\mathcal{J}_{\text{IVA}}(\mathbf{W}) = \sum_{l=1}^{L} \mathbb{E}[\|\mathbf{y}_l\|_2] - \sum_{f=1}^{F} \log|\det \mathbf{W}(f)| \tag{2.12}$$

where $\mathbf{W}$ denotes the collection of all per-bin demixing matrices $\{\mathbf{W}(f)\}_{f=1}^{F}$. Minimising $\mathcal{J}_{\text{IVA}}$ over all $\mathbf{W}(f)$ jointly recovers source-consistent demixing matrices.

Applying the natural gradient to (2.12) at each frequency bin f yields the IVA update rule [26]:

$$\mathbf{W}(f) \leftarrow \mathbf{W}(f) + \mu \left(\mathbf{I} - \mathbb{E}\left[\boldsymbol{\phi}^{(f)}\, \mathbf{y}^{H}(f)\right]\right) \mathbf{W}(f) \tag{2.13}$$

where $\boldsymbol{\phi}^{(f)} = [\phi_1^{(f)}, \ldots, \phi_L^{(f)}]^T$ is the score vector at bin f. In the batch setting, the expectation is estimated from N frames: $\mathbb{E}[\cdot] \approx \frac{1}{N}\sum_n(\cdot)$. The step size μ must be chosen carefully to balance convergence speed against stability, a problem that AuxIVA eliminates entirely, as discussed in Section 2.4.

2.2.4 The Minimal Distortion Principle (MDP)

After convergence, the scaling ambiguity must be resolved. The *Minimal Distortion Principle* (MDP) [12] selects, from all demixing matrices that achieve separation, the one that minimises the distortion of each output relative to the original mixture. This leads to the closed-form rescaling:

$$\mathbf{W}_s(f) = \operatorname{diag}\left(\hat{\mathbf{A}}(f)\right) \mathbf{W}(f), \qquad \hat{\mathbf{A}}(f) = \mathbf{W}(f)^{\#} \tag{2.14}$$

where $\mathbf{W}(f)^{\#}$ denotes the Moore–Penrose pseudo-inverse. After MDP rescaling, the demixing matrix satisfies:

$$\mathbf{W}_s(f)\, \mathbf{A}(f) = \operatorname{diag}(\mathbf{A}(f)) \tag{2.15}$$

This relationship is the key bridge to Chapter 3: the pseudo-inverse $\hat{\mathbf{A}}(f)$ provides an estimate of the acoustic transfer function (ATF) matrix, whose columns are the spatial fingerprints of each separated source.

2.2.5 Batch IVA Algorithm

Algorithm 1 summarizes the batch IVA procedure.

2.3 Adaptive IVA with Whitening

Batch IVA (Section 2.2) requires access to all N frames simultaneously before updating $\mathbf{W}$. This makes it unsuitable for real-time operation, where audio arrives as a continuous stream and the system must respond to new frames as they arrive. The *adaptive* (online) IVA framework processes the mixture one STFT frame at a time, updating $\mathbf{W}(f, n)$ with each new observation.

The adaptive version described here follows Kim [28] for the core update and Merah & Ghecham [3] for the adaptive whitening extension that enables $M > L$.

2.3.1 Adaptive Whitening

The batch whitening of Section 2.1.3 requires the full data matrix and cannot be applied incrementally. Following Merah & Ghecham [3], whitening is made adaptive through a recursive estimation of the covariance matrix at each frame.

Algorithm 1 Batch Natural Gradient IVA

Require: Multichannel mixtures $\mathbf{x}_m[t]$, $m = 1, \ldots, M$; number of sources L; number of iterations N_{iter}; step size μ
Ensure: Separated outputs $\mathbf{y}_l[t]$, $l = 1, \ldots, L$

Compute STFT: $\mathbf{X}(f, n)$ for all f, n
for $f \leftarrow 1$ **to** F **do**
    Centre and whiten: $\mathbf{X}_p(f) \leftarrow \mathbf{Q}(f)\mathbf{X}(f)$
    Initialise: $\mathbf{W}_p(f) \leftarrow \mathbf{I}_L$
end for
for iter $\leftarrow 1$ **to** N_{iter} **do**
    for $f \leftarrow 1$ **to** F **do**
        $\mathbf{Y}(f) \leftarrow \mathbf{W}_p(f)\mathbf{X}_p(f)$
        Compute score: $\phi_l^{(f)}(n) = Y_l(f, n)/\|\mathbf{y}_l(n)\|_2$ for all l, n
        $\mathbf{R}(f) \leftarrow \frac{1}{N}\mathbf{\Phi}(f)\mathbf{Y}^H(f)$
        $\mathbf{W}_p(f) \leftarrow \mathbf{W}_p(f) + \mu\,(\mathbf{I} - \mathbf{R}(f))\,\mathbf{W}_p(f)$
    end for
end for
for $f \leftarrow 1$ **to** F **do** ▷ Rescaling
    $\mathbf{W}(f) \leftarrow \mathbf{W}_p(f)\mathbf{Q}(f)$
    $\mathbf{W}_s(f) \leftarrow \operatorname{diag}(\mathbf{W}(f)^{\#})\,\mathbf{W}(f)$ ▷ MDP
    $\mathbf{Y}(f) \leftarrow \mathbf{W}_s(f)\mathbf{X}(f)$
end for
Compute ISTFT: $\hat{s}_l[t] \leftarrow \text{ISTFT}(\{Y_l(f, n)\})$ for all l

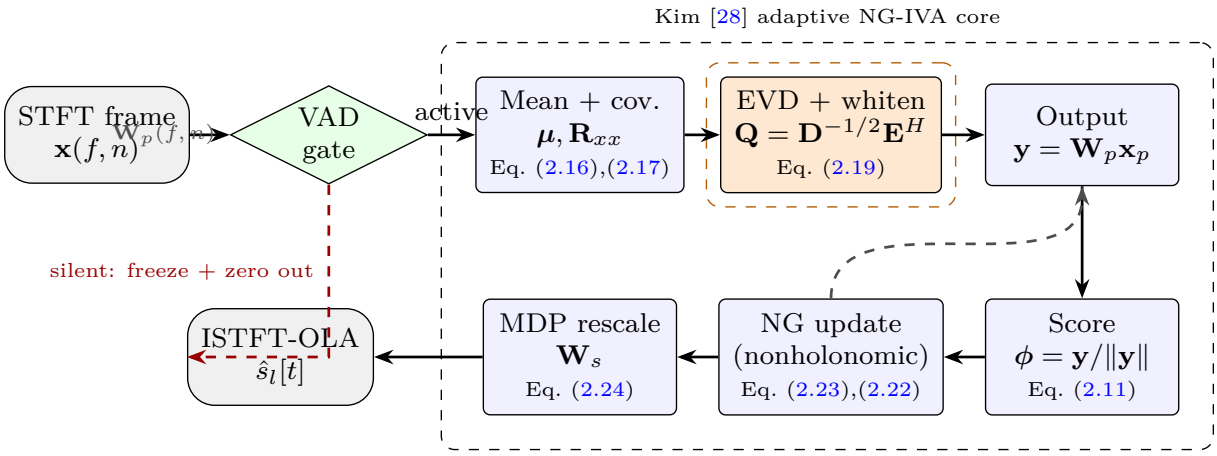


Figure 2.2: Block diagram of the adaptive NG-IVA pipeline with whitening. Processing operates per STFT frame n. The dashed grey box groups the modules inherited from Kim [28]; the orange adaptive whitening block is the contribution of Merah & Ghecham [3] that lifts the $M = L$ restriction. The red branch shows the VAD freeze path (Section 2.6.1).

Adaptive mean removal: The mean of the STFT coefficients at frequency bin f is tracked recursively using an exponential forgetting factor $\alpha \in (0, 1)$:

$$\boldsymbol{\mu}(f, n) = \alpha\, \boldsymbol{\mu}(f, n-1) + (1-\alpha)\, \mathbf{x}(f, n) \tag{2.16}$$

The centred frame used in all subsequent processing is $\tilde{\mathbf{x}}(f, n) = \mathbf{x}(f, n) - \boldsymbol{\mu}(f, n)$. The exponential window is preferred over a uniform recursive mean because it gives higher weight to recent observations and can track slow non-stationarities in the acoustic environment.

Adaptive covariance estimation: The spatial covariance matrix is updated recursively:

$$\mathbf{R}_{xx}(f, n) = \alpha\, \mathbf{R}_{xx}(f, n-1) + (1-\alpha)\, \tilde{\mathbf{x}}(f, n)\tilde{\mathbf{x}}^H(f, n) \tag{2.17}$$

initialised as $\mathbf{R}_{xx}(f, 0) = \mathbf{I}_M$.

Eigenvalue decomposition and whitening: At each frame, the EVD of the current covariance estimate is computed:

$$\mathbf{R}_{xx}(f, n) = \mathbf{E}(f, n)\, \mathbf{D}(f, n)\, \mathbf{E}^H(f, n) \tag{2.18}$$

Only the L largest eigenvalue-eigenvector pairs are retained, performing dimensionality reduction from M channels to L source dimensions. The whitening matrix and the whitened frame are:

$$\mathbf{Q}(f, n) = \mathbf{D}^{-1/2}(f, n)\mathbf{E}^H(f, n), \qquad \mathbf{x}_p(f, n) = \mathbf{Q}(f, n)\tilde{\mathbf{x}}(f, n) \tag{2.19}$$

This step is the computational bottleneck: the EVD of an $M \times M$ matrix costs $\mathcal{O}(M^3)$ per bin per frame. Its benefit is the ability to exploit all $M = 16$ UMA-16 microphones even when only $L < M$ sources are present, capturing the full spatial diversity of the array.

2.3.2 Adaptive Separation Filter Update

In the online setting, the batch expectation $\mathbb{E}[\boldsymbol{\phi}^{(f)}\mathbf{y}^H(f)]$ cannot be computed. It is replaced by a single-frame instantaneous estimate:

$$\mathbf{R}(f, n) \approx \boldsymbol{\phi}^{(f)}(n)\, \mathbf{y}^H(f, n) \tag{2.20}$$

To robustify the adaptive update against sudden changes in input energy, which cause the natural gradient to diverge if the effective step size becomes too large, a power-normalization factor ζ (distinct from the VAD a-priori SNR ξ introduced in Section 2.6.1) is maintained per frequency bin [28]:

$$\zeta(f, n) = \beta\, \zeta(f, n-1) + (1-\beta)\, \frac{1}{L} \sum_{l=1}^{L} |x_l(f, n)|^2 \tag{2.21}$$

where $\beta \in (0, 1)$ is a second forgetting factor controlling the energy smoothing time constant. The filter update is then:

$$\mathbf{W}_p(f, n) = \mathbf{W}_p(f, n-1) + \frac{\eta}{\sqrt{\zeta(f, n) + \varepsilon}}\, \Delta\mathbf{W}_p(f, n) \tag{2.22}$$

where $\varepsilon > 0$ prevents division by zero and η is the nominal learning rate.

2.3.3 The Nonholonomic Constraint

A practical failure mode of the stochastic natural gradient is filter divergence during silence. When source l is inactive, $\mathbf{y}_l \approx \mathbf{0}$ and the score function $\phi_l^{(f)} = Y_l(f)/\|\mathbf{y}_l\|$ becomes numerically ill-conditioned. The update rule then drives the diagonal entries of $\mathbf{W}_p$ to grow unboundedly as the filter tries to amplify what it interprets as a silent but present source.

The *nonholonomic constraint* introduced by Amari et al. [39] addresses this by setting the diagonal elements of the gradient matrix to zero before applying the update. Defining $\mathbf{R}(f,n) = \boldsymbol{\phi}^{(f)}\mathbf{y}^H(f,n)$, the constrained gradient is:

$$\Delta\mathbf{W}_p(f,n) = \Big(\mathbf{\Lambda}(f,n) - \mathbf{R}(f,n)\Big)\mathbf{W}_p(f,n) \tag{2.23}$$

where $\mathbf{\Lambda}(f,n) = \mathrm{diag}(\mathbf{R}(f,n))$ replaces the identity matrix $\mathbf{I}$ from the unconstrained update (2.13). This ensures that self-amplification terms (the diagonal entries) are cancelled exactly, preventing divergence during silence periods while leaving the cross-channel suppression terms (the off-diagonal entries) intact.

2.3.4 MDP Rescaling in the Online Setting

MDP rescaling applies at each frame identically to the batch case (Section 2.2.4), after projecting the adaptive whitened demixing matrix back to the original M-dimensional space:

$$\mathbf{W}'(f,n) = \mathbf{W}_p(f,n)\,\mathbf{Q}(f,n), \qquad \mathbf{W}_s(f,n) = \mathrm{diag}\Big(\mathbf{W}'(f,n)^{\#}\Big)\,\mathbf{W}'(f,n) \tag{2.24}$$

The separated STFT frame is: $\mathbf{y}(f,n) = \mathbf{W}_s(f,n)\mathbf{x}(f,n)$.

2.3.5 Full Adaptive Algorithm and Complexity

Algorithm 2 gives the complete frame-by-frame procedure.

Per-step complexity:

Table 2.1 summarizes the computational cost of each step per frame per frequency bin. The EVD step dominates, scaling as $\mathcal{O}(M^3)$.All other steps scale as $\mathcal{O}(ML)$ or $\mathcal{O}(L^2)$.

Table 2.1: Per-frame, per-frequency-bin complexity of the adaptive NG-IVA + whitening.

Step	**Operation**	**Cost**
Covariance update	Outer product + scaling	$\mathcal{O}(M^2)$
EVD (rank-L)	Truncated eigendecomposition	$\mathcal{O}(M^3)$
Whitening	Matrix-vector product	$\mathcal{O}(LM)$
Score function	Norm + division	$\mathcal{O}(LF)$
Total per bin		$\mathcal{O}(M^3)$
Total per frame	(Aggregated over F bins)	$\mathcal{O}(FM^3)$

Algorithm 2 Adaptive NG-IVA with Whitening [3]

Require: Audio stream; L, η, α, β, ε, θ_{VAD}
Ensure: Separated audio streams $\hat{s}_l$, $l = 1, \ldots, L$

Initialise: $\mathbf{R}_{xx}(f,0) \leftarrow \mathbf{I}_M$, $\mathbf{W}_p(f,0) \leftarrow \mathbf{I}_L$, $\boldsymbol{\mu}(f,0) \leftarrow \mathbf{0}$, $\zeta(f,0) \leftarrow 1$ for all f
for each incoming audio frame **do**
  Compute STFT frame: $\mathbf{x}(f,n)$ for all f
  if $E_{\text{frame}} \leq \theta_{\text{VAD}}$ **then**
    skip update (freeze filter); output zeros
    continue
  end if
  for $f \leftarrow 1$ **to** F **do**
    Update mean: $\boldsymbol{\mu}(f,n) \leftarrow \alpha\boldsymbol{\mu}(f,n-1) + (1-\alpha)\mathbf{x}(f,n)$
    Centre: $\tilde{\mathbf{x}}(f,n) \leftarrow \mathbf{x}(f,n) - \boldsymbol{\mu}(f,n)$
    Update covariance: $\mathbf{R}_{xx}(f,n) \leftarrow \alpha\mathbf{R}_{xx}(f,n-1) + (1-\alpha)\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H$
    EVD: $[\mathbf{E}, \mathbf{D}] = \text{eig}_L(\mathbf{R}_{xx}(f,n))$
    $\mathbf{Q}(f,n) \leftarrow \mathbf{D}^{-1/2}\mathbf{E}^H \quad \mathbf{x}_p(f,n) \leftarrow \mathbf{Q}(f,n)\tilde{\mathbf{x}}(f,n)$
    Compute output: $\mathbf{y}(f,n) \leftarrow \mathbf{W}_p(f,n-1)\mathbf{x}_p(f,n)$
    Compute score: $\phi_l^{(f)}(n) \leftarrow Y_l(f,n)/\|\mathbf{y}_l(n)\|_2$ for all l
    $\mathbf{R}(f,n) \leftarrow \boldsymbol{\phi}^{(f)}(n)\mathbf{y}^H(f,n)$
    $\boldsymbol{\Lambda}(f,n) \leftarrow \text{diag}(\mathbf{R}(f,n))$ ▷ Nonholonomic constraint
    Update normalization: $\zeta(f,n) \leftarrow \beta\zeta(f,n-1) + (1-\beta)\frac{1}{L}\sum_l |x_l(f,n)|^2$
    Update filter: $\mathbf{W}_p(f,n) \leftarrow \mathbf{W}_p(f,n-1) + \frac{\eta}{\sqrt{\zeta(f,n)+\varepsilon}}(\boldsymbol{\Lambda} - \mathbf{R})\mathbf{W}_p(f,n-1)$
    $\mathbf{W}'(f,n) \leftarrow \mathbf{W}_p(f,n)\mathbf{Q}(f,n) \quad \mathbf{W}_s(f,n) \leftarrow \text{diag}(\mathbf{W}'^{\#})\mathbf{W}'$ ▷ MDP
    Output: $\mathbf{y}(f,n) \leftarrow \mathbf{W}_s(f,n)\mathbf{x}(f,n)$
  end for
  Reconstruct: $\hat{s}_l[t] \leftarrow \text{ISTFT} - \text{OLA}(\{Y_l(f,n)\})$ for all l
end for

2.4 Advanced IVA Variants

The adaptive NG-IVA of Section 2.3 carries two practical limitations. First, the step size η requires careful tuning: too large causes divergence, too small causes impractically slow convergence, and the optimal value depends on the acoustic environment and is not known in advance. Second, the stochastic natural gradient converges slowly when sources are dynamic, when speakers move or start and stop speaking, the demixing matrix $\mathbf{W}$ must track the changing mixing matrix $\mathbf{A}(f)$, but the slow, noisy stochastic gradient updates lag behind. In dynamic scenes, teleconferences with participants moving freely, or meeting room scenarios with speakers turning their heads, this tracking delay directly translates to audible cross-talk spikes.

Two families of algorithms address these limitations without introducing new hyperparameters.

2.4.1 AuxIVA: Auxiliary-Function IVA

AuxIVA [38] replaces the step-size-dependent natural gradient with an auxiliary-function optimisation that guarantees monotone decrease of $\mathcal{J}_{\text{IVA}}$ at every iteration.

The auxiliary function technique:

The core idea [38] is to construct an auxiliary function $Q(\mathbf{W}, \mathbf{V})$ that satisfies:

$$\mathcal{J}_{\text{IVA}}(\mathbf{W}) \leq Q(\mathbf{W}, \mathbf{V}), \quad \forall\, \mathbf{V}, \quad \text{with equality when } r_k = \|\mathbf{y}_k\|_2 \tag{2.25}$$

Alternating minimisation of Q over $\mathbf{V}$ (in closed form) and over $\mathbf{W}$ (also in closed form) guarantees that $\mathcal{J}_{\text{IVA}}$ decreases at every step, with no step size involved. The auxiliary variables are the per-source *weighted covariance matrices*:

$$\mathbf{V}_k(f) = \mathbb{E}\left[\frac{G'(r_k)}{r_k}\,\mathbf{x}(f)\mathbf{x}^H(f)\right] \tag{2.26}$$

where $G(r_k) = r_k$ for the Laplace contrast function (so $G'(r_k)/r_k = 1/r_k$) and $r_k = \|\mathbf{y}_k\|_2$.

Iterative projection update:

Given current $\mathbf{V}_k(f)$, minimising Q over the l-th demixing row $\mathbf{w}_k(f)$ while keeping all others fixed reduces to a two-step update [38]:

$$\mathbf{w}_k(f) \leftarrow \Big(\mathbf{W}(f)\mathbf{V}_k(f)\Big)^{-1}\mathbf{e}_k \tag{2.27}$$

$$\mathbf{w}_k(f) \leftarrow \frac{\mathbf{w}_k(f)}{\sqrt{\mathbf{w}_k^H(f)\mathbf{V}_k(f)\mathbf{w}_k(f)}} \tag{2.28}$$

where $\mathbf{e}_k$ is the l-th standard basis vector. Applied sequentially for $l = 1, \ldots, L$, this is the *iterative projection* (IP) method. Each application of (2.27)–(2.28) is guaranteed to decrease Q and hence $\mathcal{J}_{\text{IVA}}$. The full batch AuxIVA algorithm is given in Algorithm 3.

Algorithm 3 Batch AuxIVA [38]

Require: STFT frames $\mathbf{X}(f, n)$; L; max iterations N_{iter}
Ensure: Separated STFT $\mathbf{Y}(f, n)$
Initialise: $\mathbf{W}(f) \leftarrow \mathbf{I}_L$ for all f
for iter $\leftarrow 1$ **to** N_{iter} **do**
  for $l \leftarrow 1$ **to** L **do**
    $r_{l,n} \leftarrow \sqrt{\sum_f |w_k^H(f)x(f,n)|^2}$ for all n
    for $f \leftarrow 1$ **to** F **do**
      $\mathbf{V}_k(f) \leftarrow \frac{1}{N}\sum_n \frac{1}{r_{l,n}}\mathbf{x}(f,n)\mathbf{x}^H(f,n)$
      $\mathbf{w}_k(f) \leftarrow (\mathbf{W}(f)\mathbf{V}_k(f))^{-1}\mathbf{e}_k$
      Normalise: $\mathbf{w}_k(f) \leftarrow \mathbf{w}_k(f)/\sqrt{\mathbf{w}_k^H\mathbf{V}_k\mathbf{w}_k}$
    end for
  end for
end for
Apply MDP rescaling (eq. (2.14))
$\mathbf{Y}(f, n) \leftarrow \mathbf{W}(f)\mathbf{X}(f, n)$

2.4.2 Online AuxIVA

Taniguchi et al. [29] extend the batch AuxIVA to the online setting by replacing the full-data expectation in (2.26) with an autoregressive approximation using forgetting factor α:

$$\mathbf{V}_k(f; n) = \alpha\,\mathbf{V}_k(f; n-1) + (1-\alpha)\,\frac{G'(r_{l,n})}{r_{l,n}}\,\mathbf{x}(f,n)\mathbf{x}^H(f,n) \tag{2.29}$$

The demixing update at each frame applies the same iterative projection (2.27)–(2.28) using the current $\mathbf{V}_k(f;n)$. To reduce the per-frame cost of the matrix inverse in eq. (2.27), the matrix inversion lemma (Woodbury identity) is applied: the inverse update reduces to a rank-one correction, avoiding any explicit matrix inversion and reducing the cost from $\mathcal{O}(L^3)$ to $\mathcal{O}(L^2)$ per step [29].

Online AuxIVA inherits the convergence guarantee of its batch counterpart asymptotically [40]: as $n \to \infty$ and the forgetting factor $\alpha \to 1$, the recursive estimate of $\mathbf{V}_k$ converges to the full time-average. In dynamic scenarios, a smaller α gives faster tracking of moving sources at the cost of noisier estimates; unlike the natural gradient step size, α has an intuitive interpretation as a time constant (approximately $1/(1-\alpha)$ frames) and is straightforward to tune.

The complete, frame-by-frame procedure for the data-free Online AuxIVA (strictly for the determined case $M = L$) is summarized in Algorithm 4. To prevent rank-deficient updates during speech pauses, the Laplace weights $w_{l,n}$ are clipped to a maximum threshold $w_{\max}$.

Algorithm 4 Online AuxIVA [29]

Require: Audio stream $\mathbf{x}[t]$ (L channels); Forgetting factors $\alpha_{\text{mean}}, \alpha_V$; Maximum weight $w_{\max}$
Ensure: Separated STFT $\mathbf{Y}(f,n)$
Initialise: $\mathbf{W}(f,0) \leftarrow \mathbf{I}_L$, $\mathbf{V}_k(f,0) \leftarrow \mathbf{0}$, $\boldsymbol{\mu}(f,0) \leftarrow \mathbf{0}$
for each incoming audio frame n **do**
  Compute STFT frame: $\mathbf{x}(f,n)$
  $\boldsymbol{\mu}(f,n) \leftarrow \alpha_{\text{mean}}\boldsymbol{\mu}(f,n-1) + (1-\alpha_{\text{mean}})\mathbf{x}(f,n)$
  $\tilde{\mathbf{x}}(f,n) \leftarrow \mathbf{x}(f,n) - \boldsymbol{\mu}(f,n)$
  for $l \leftarrow 1$ **to** L **do**
    $y_k(f,n) \leftarrow \mathbf{w}_k^H(f,n-1)\tilde{\mathbf{x}}(f,n)$ for all f
    $r_{l,n} \leftarrow \sqrt{\sum_f |y_k(f,n)|^2 + \varepsilon}$
    $w_{l,n} \leftarrow \min(1/r_{l,n}, w_{\max})$ ▷ Safeguard against silent frames
  end for
  for $f \leftarrow 1$ **to** F **do**
    $\mathbf{W}(f,n) \leftarrow \mathbf{W}(f,n-1)$
    for $l \leftarrow 1$ **to** L **do**
      $\mathbf{V}_k(f,n) \leftarrow \alpha_V\mathbf{V}_k(f,n-1) + (1-\alpha_V)w_{l,n}\tilde{\mathbf{x}}(f,n)\tilde{\mathbf{x}}^H(f,n)$
      $\mathbf{w}_k(f,n) \leftarrow (\mathbf{W}(f,n)\mathbf{V}_k(f,n))^{-1}\mathbf{e}_k$
      $\mathbf{w}_k(f,n) \leftarrow \mathbf{w}_k(f,n)/\sqrt{\mathbf{w}_k^H(f,n)\mathbf{V}_k(f,n)\mathbf{w}_k(f,n)}$
    end for
  end for
  Apply MDP rescaling and output separated frame $\mathbf{y}(f,n)$
end for

2.4.3 OverIVA: Overdetermined IVA

Both NG-IVA with whitening and AuxIVA (in the standard determined case) require either a dimensionality reduction step (whitening) or a square demixing problem ($M = L$). OverIVA [30] addresses the overdetermined case directly, without whitening, by modelling the extra microphone channels explicitly as a Gaussian noise background.

Signal model: The full microphone signal is modelled as:

$$\mathbf{x}(f,n) = \mathbf{A}_f\,\mathbf{s}(f,n) + \mathbf{\Psi}_f\,\mathbf{z}(f,n) \tag{2.30}$$

where $\mathbf{s}(f,n) \in \mathbb{C}^L$ are the L non-Gaussian target sources and $\mathbf{z}(f,n) \in \mathbb{C}^{M-L}$ is the Gaussian noise/background component. A full $M \times M$ demixing matrix $\hat{\mathbf{W}}_f$ is parametrised as a block structure:

$$\hat{\mathbf{W}}_f = \begin{pmatrix} \mathbf{W}_f \\ \mathbf{U}_f \end{pmatrix}, \quad \mathbf{W}_f \in \mathbb{C}^{L\times M}, \quad \mathbf{U}_f = [\mathbf{J}_f \mid -\mathbf{I}_{M-L}] \tag{2.31}$$

The signal rows $\mathbf{W}_f$ contain the L demixing vectors for the target sources; the noise rows $\mathbf{U}_f$ span the orthogonal complement and are updated in closed form.

Alternating updates: The noise covariance $\mathbf{R}_f = \mathbf{U}_f\mathbf{C}_f\mathbf{U}_f^H$ (where $\mathbf{C}_f = \frac{1}{N}\mathbf{X}_f\mathbf{X}_f^H$ is the input sample covariance) can be solved in closed form once $\mathbf{W}_f$ is fixed, and its contribution to the likelihood is constant. The signal rows are then updated by the same AuxIVA iterative projection (2.27)–(2.28) applied to $\mathbf{W}_f$ alone. After each signal update, the noise rows are restored to maintain the orthogonality constraint between signal and noise subspaces [30]:

$$\mathbf{J}_f = \left(\mathbf{E}_2\,\mathbf{C}_f\,\mathbf{W}_f^H\right)\left(\mathbf{E}_1\,\mathbf{C}_f\,\mathbf{W}_f^H\right)^{-1} \tag{2.32}$$

where $\mathbf{E}_1 = [\mathbf{I}_K\ \mathbf{0}]$ and $\mathbf{E}_2 = [\mathbf{0}\ \mathbf{I}_{M-L}]$ are selection matrices. The full batch OverIVA algorithm is given in Algorithm 5.

Algorithm 5 Batch OverIVA [30]

Require: STFT frames $\mathbf{X}(f,n)$; $L < M$; max iterations N_{iter}
Ensure: Separated STFT $\mathbf{Y}(f,n)$, L channels

Precompute: $\mathbf{C}_f \leftarrow \frac{1}{N}\mathbf{X}_f\mathbf{X}_f^H$ for all f
Initialise: $\mathbf{W}_f \leftarrow [\mathbf{I}_K\ \mathbf{0}]$, $\mathbf{J}_f \leftarrow \mathbf{0}$ for all f
for iter $\leftarrow 1$ **to** N_{iter} **do**
  for $l \leftarrow 1$ **to** L **do**
    $r_{l,n} \leftarrow \sqrt{\sum_f |w_{kf}^H x(f,n)|^2}$ for all n
    for $f \leftarrow 1$ **to** F **do**
      $\mathbf{V}_{l,f} \leftarrow \frac{1}{N}\sum_n \frac{1}{r_{l,n}}\mathbf{x}(f,n)\mathbf{x}^H(f,n)$
      $\mathbf{w}_{l,f} \leftarrow (\hat{\mathbf{W}}_f\mathbf{V}_{l,f})^{-1}\mathbf{e}_k$
      Normalise $\mathbf{w}_{l,f}$ (eq. (2.28))
      $\mathbf{J}_f \leftarrow (\mathbf{E}_2\mathbf{C}_f\mathbf{W}_f^H)(\mathbf{E}_1\mathbf{C}_f\mathbf{W}_f^H)^{-1}$
      $\hat{\mathbf{W}}_f \leftarrow \begin{bmatrix} \mathbf{W}_f \\ \mathbf{J}_f \quad -\mathbf{I}_{M-L} \end{bmatrix}$
    end for
  end for
end for
Apply MDP rescaling; output $\mathbf{Y} = \mathbf{W}_f\mathbf{X}$

The online variant of OverIVA follows by replacing the batch covariance $\mathbf{C}_f$ with a recursive estimate $\mathbf{C}_f(n)$ updated via a forgetting factor α_C. The iterative projection is then applied frame-by-frame. To prevent early-frame permutation swaps caused by poor initial statistics, the covariance matrices are seeded with small diagonal priors and the input-covariance forgetting factor is *ramped*: a cumulative average ($\alpha_C^{(n)} = (n-1)/n$) is

used during the first frames and capped at its steady-state value α_C once enough data has accumulated, which activates the background constraint $\mathbf{J}_f$ within a few frames instead of leaving it prior-dominated. The complete online procedure is detailed in Algorithm 6.

Algorithm 6 Online OverIVA [30]

Require: Audio stream $\mathbf{x}[t]$ (M channels); L sources ($L < M$); Forgetting factors $\alpha_{\text{mean}}, \alpha_V, \alpha_C$
Ensure: Separated STFT $\mathbf{Y}(f, n)$
Initialise: $\mathbf{W}_f(0) \leftarrow [\mathbf{I}_K\ \mathbf{0}]$, $\mathbf{J}_f(0) \leftarrow \mathbf{0}$
Initialise priors: $\mathbf{V}_{l,f}(0) \leftarrow \varepsilon_V \mathbf{I}$, $\mathbf{C}_f(0) \leftarrow \varepsilon_C \mathbf{I}$
for each incoming audio frame n **do**
    Compute STFT frame: $\mathbf{x}(f, n)$
    $\boldsymbol{\mu}(f, n) \leftarrow \alpha_{\text{mean}} \boldsymbol{\mu}(f, n-1) + (1 - \alpha_{\text{mean}}) \mathbf{x}(f, n)$
    $\tilde{\mathbf{x}}(f, n) \leftarrow \mathbf{x}(f, n) - \boldsymbol{\mu}(f, n)$
    $\alpha_C^{(n)} \leftarrow \min\big(\alpha_C,\ (n-1)/n\big)$ ▷ ramped forgetting: cumulative average until α_C is reached
    for $f \leftarrow 1$ **to** F **do**
        $\mathbf{C}_f(n) \leftarrow \alpha_C^{(n)} \mathbf{C}_f(n-1) + (1 - \alpha_C^{(n)}) \tilde{\mathbf{x}}(f, n) \tilde{\mathbf{x}}^H(f, n)$
    end for
    for $l \leftarrow 1$ **to** L **do**
        $y_k(f, n) \leftarrow \mathbf{w}_{l,f}^H(n-1) \tilde{\mathbf{x}}(f, n)$ for all f
        $r_{l,n} \leftarrow \sqrt{\sum_f |y_k(f, n)|^2} + \varepsilon$
    end for
    for $f \leftarrow 1$ **to** F **do**
        $\hat{\mathbf{W}}_f(n) \leftarrow \begin{bmatrix} \mathbf{W}_f(n-1) \\ \mathbf{J}_f(n-1) \quad -\mathbf{I}_{M-L} \end{bmatrix}$
        for $l \leftarrow 1$ **to** L **do**
            $\mathbf{V}_{l,f}(n) \leftarrow \alpha_V \mathbf{V}_{l,f}(n-1) + (1 - \alpha_V) \frac{1}{r_{l,n}} \tilde{\mathbf{x}}(f, n) \tilde{\mathbf{x}}^H(f, n)$
            $\mathbf{w}_{l,f}(n) \leftarrow (\hat{\mathbf{W}}_f(n) \mathbf{V}_{l,f}(n))^{-1} \mathbf{e}_k$
            $\mathbf{w}_{l,f}(n) \leftarrow \mathbf{w}_{l,f}(n) / \sqrt{\mathbf{w}_{l,f}^H(n) \mathbf{V}_{l,f}(n) \mathbf{w}_{l,f}(n)}$
            Update l-th row of $\mathbf{W}_f(n)$ and $\hat{\mathbf{W}}_f(n)$ with $\mathbf{w}_{l,f}(n)$
        end for
        $\mathbf{J}_f(n) \leftarrow (\mathbf{E}_2 \mathbf{C}_f(n) \mathbf{W}_f^H(n)) (\mathbf{E}_1 \mathbf{C}_f(n) \mathbf{W}_f^H(n) + \text{reg})^{-1}$
    end for
    Apply MDP rescaling using diagonal of $\hat{\mathbf{W}}_f(n)^{-1}$ and output $\mathbf{y}(f, n)$
end for

Computational advantage: OverIVA updates only L of the M demixing rows, whereas a full M-channel overdetermined AuxIVA would update all M. With the per-row iterative-projection solve costing $\mathcal{O}(M^3)$, this gives a per-bin per-iteration cost of $\mathcal{O}(KM^3)$ for OverIVA versus $\mathcal{O}(M^4)$ for the full M-row update, a factor of M/L reduction. For the UMA-16 array with $M = 16$ and $L = 2$ sources, this is an 8 times reduction in the dominant cost. Scheibler & Ono [30] report that this speedup is achieved without any loss in separation quality compared to full AuxIVA.

2.4.4 Algorithm Comparison and Theoretical Trade-offs

The theoretical and practical trade-offs between the discussed IVA variants can be summarized as follows:

- **AuxIVA Advantages over NG-IVA:** By relying on the auxiliary function technique, AuxIVA guarantees monotonic convergence and eliminates the need for empirical step-size tuning. This avoids the instability and slow convergence often associated with NG-IVA, resulting in considerably faster overall convergence rates.
- **The Online Whitening Flaw:** Extending AuxIVA to online overdetermined arrays ($M > L$) introduces a fundamental issue. While batch PCA whitening is stable, computing the eigenvalue decomposition frame-by-frame online causes the phase and ordering of eigenvectors to fluctuate arbitrarily.
- **Spatial Model Collapse:** These arbitrary rotations severely corrupt the recursive time-averaging of the weighted covariance matrices $\mathbf{V}_k(f, n)$ in online AuxIVA. Adaptive NG-IVA only survives this phase ambiguity because its adaptive step size heavily dampens the gradient updates, effectively smoothing over the rotations.
- **Loss of Spatial Diversity:** To avoid this collapse, standard online AuxIVA must abandon adaptive whitening and operate strictly in the determined case ($M = L$). This is sub-optimal, as discarding redundant microphones wastes valuable spatial information essential for robust separation and acoustic localization.
- **The OverIVA Solution:** OverIVA elegantly resolves this dilemma by structurally modeling the $M-L$ redundant channels as a background noise subspace rather than projecting them away. This parametric block-update bypasses unstable adaptive whitening entirely, allowing the system to exploit the full M-microphone array and marry AuxIVA's stability with overdetermined spatial richness.

We summarize the key properties of all IVA variants discussed in this chapter in Table 2.2.

Table 2.2: Comparison of IVA algorithm variants. "RT" = real-time capable. "$M>L$" = supports more microphones than sources. "Step size" = requires tuning. "Cvg. guar." = guaranteed monotone convergence. Complexity is the dominant cost *per frequency bin*, counted per iteration for the batch methods and per frame for the online methods, as implemented in the present MATLAB prototype (see the note below the table). M microphones, L sources, N frames.

Algorithm	**RT**	$M>L$	**Step size**	**Cvg. guar.**	**Complexity**
Batch NG-IVA	No	w/ whit.	Yes	No	$\mathcal{O}(NL^2)$
Adaptive NG-IVA+Whit.	Yes	Yes	Yes	No	$\mathcal{O}(M^3)$
Batch AuxIVA	No	No	No	Yes	$\mathcal{O}(NL^3)$
Online AuxIVA	Yes	No	No	Asymp.	$\mathcal{O}(L^4)$
Batch OverIVA	No	Yes	No	Yes	$\mathcal{O}(NKM^2)$
Online OverIVA	Yes	Yes	No	Asymp.	$\mathcal{O}(KM^2)$

2.5 Simulation Validation

This section validates the IVA algorithms developed in Sections 2.2–2.4 through controlled MATLAB simulations on synthetic convolutive mixtures. The evaluation proceeds in two stages: a *single-scenario* deep comparison of the three IVA variants (NG-IVA, Aux-IVA, OverIVA) in both batch and online form (Section 2.5.3), and a *comprehensive grid* that sweeps reverberation, noise, angular separation, source count, and speaker content (Section 2.5.4). Together they characterise each variant's convergence and steady-state separation quality and motivate the implementation choice made in Chapter 4.

2.5.1 Simulation Setup

All mixtures are generated with the Pyroomacoustics image-source method [41] in a shoebox room, using the UMA-16 geometry; the common parameters are listed in Table 2.3. Two datasets are used. The *single scenario* of Section 2.5.3 is a lightly reverberant, near-anechoic mixture of two speakers (RT60 = 0.1 s, two sources at azimuth/elevation $(-10°, +5°)$ and $(+20°, +10°)$, 0.6 m from the array, inside the critical distance). The *grid* of Section 2.5.4 reuses the same room and array but sweeps one factor at a time: reverberation (RT60 0.1–0.8 s), input SNR (0 dB to noiseless), diffuse noise, angular separation (5°–80°), source count (2–4), and speaker content (gender combinations, speech/music, speech/babble).

Table 2.3: Common MATLAB simulation setup for the IVA experiments.

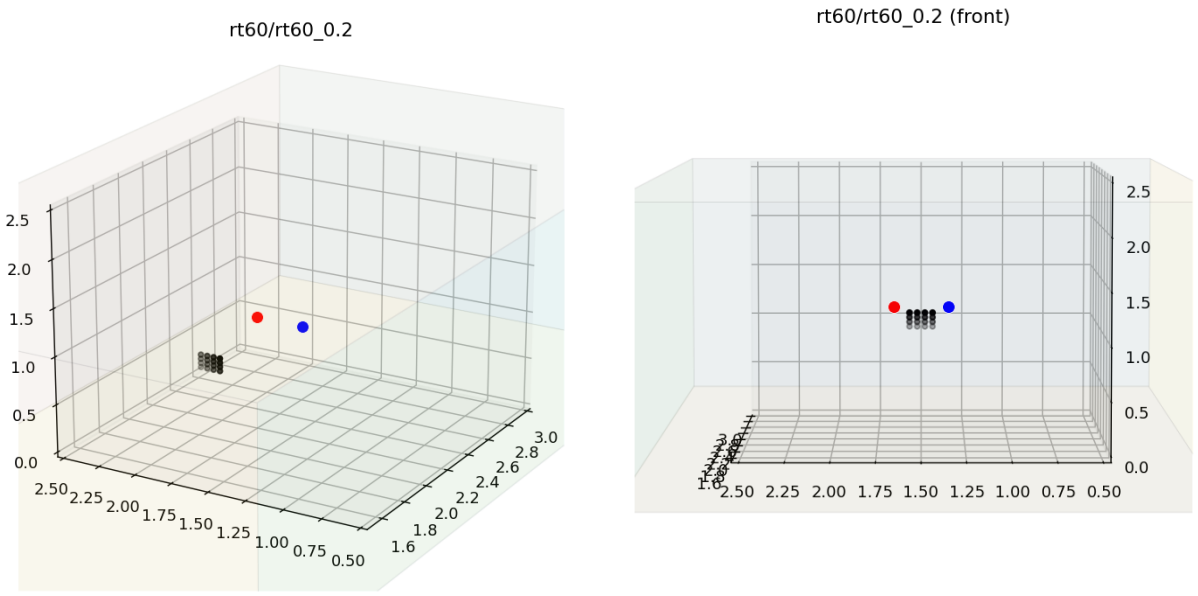
Parameter	**Value**
Room dimensions	$4 \times 3 \times 2.5$ m ($V = 30\,\mathrm{m}^3$)
Microphone array	UMA-16, 4×4 vertical URA, $d = 42$ mm
Number of microphones M	16
Number of sources L	2 (grid: up to 4)
Sampling rate f_s	16 kHz
STFT	NFFT = 512, hop = 256 (50% overlap)
Analysis window	$2 \cdot \mathrm{hann}(512)/512$
Signal duration	≈ 27 s (1708 frames)
Realizations	one per scenario (no Monte Carlo)

A representative rendering of the simulated room is shown in Figure 2.3: the UMA-16 vertical array sits at the centre of the $4 \times 3 \times 2.5$ m shoebox, with the two speech sources placed 0.6 m away at the prescribed azimuth/elevation angles.

2.5.2 Evaluation Metrics (BSS Eval)

Separation quality is measured using the standard BSS-Eval framework [42]. The evaluation procedure decomposes each estimated source $\hat{s}_j$ into four orthogonal components by projecting it onto the true sources and noise subspaces:

$$\hat{s}_j = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}} \tag{2.33}$$



(a) Perspective view.

(b) Front view.

Figure 2.3: Simulated shoebox room generated with Pyroomacoustics [41]: the 4×4 UMA-16 vertical array (black dots) is centred at $(1.5, 1.5, 1.25)$ m and the two sources (coloured) lie 0.6 m from the array. Shown for the RT60 $= 0.2$ s configuration.

where s_{target} is the allowable modified version of the true source (e.g., via a time-invariant filter to account for convolutive mixing), e_{interf} represents residual cross-talk from interfering sources, e_{noise} is the remaining sensor noise, and e_{artif} captures algorithmic artefacts such as musical noise or forbidden filter distortions.

Based on the energies of these components, three primary logarithmic metrics are calculated:

- **Source to Interference Ratio (SIR):** The primary indicator of separation performance, measuring the degree of interference rejection.

$$\text{SIR} = 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2} \tag{2.34}$$

- **Sources to Artifacts Ratio (SAR):** Isolates the algorithmic artefacts (such as musical noise) introduced by the separation filters, computed independently of the residual interference.

$$\text{SAR} = 10 \log_{10} \frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2} \tag{2.35}$$

- **Signal to Distortion Ratio (SDR):** A global performance measure that quantifies the overall signal quality by penalising all forms of degradation (interference, noise, and artefacts).

$$\text{SDR} = 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2} \tag{2.36}$$

2.5.3 Single-Scenario Comparison

The three IVA variants are first compared on the single near-anechoic two-speaker mixture, in both batch and online form, stepped under identical STFT and projection-back settings. Figure 2.4 (batch, versus iteration) and Figure 2.5 (online, versus frame) show the SDR/SIR/SAR trajectories, and Table 2.4 reports the final values.

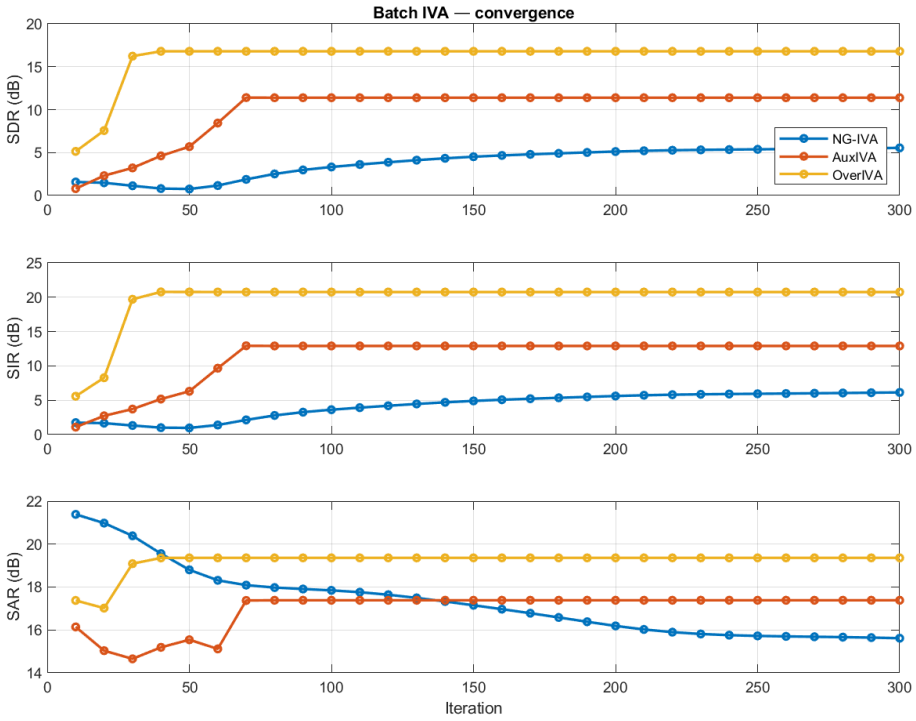


Figure 2.4: Batch convergence on the single scenario: SDR, SIR and SAR (dB) versus iteration for NG-IVA, AuxIVA and OverIVA. OverIVA converges fastest and highest; NG-IVA is the slowest and its SAR even declines as it suppresses interference.

Batch (300 iterations):

OverIVA converges fastest, reaching its plateau in about 30 iterations, and is best on all three metrics (SIR 20.7 dB). AuxIVA follows (plateau near 70 iterations, SIR 12.9 dB). NG-IVA is by far the slowest, climbing monotonically over the full budget to only SIR 6.1 dB, and its SAR even *decreases* as it trades artefact-freedom for interference suppression. The two iterative-projection methods thus dominate the older gradient method in both speed and quality, with no step-size tuning.

Online (1708 frames):

After a noisy cold start, OverIVA attains the best SIR (14.1 dB), its overdetermined advantage growing steadily with the number of frames seen; AuxIVA gives the best SDR/SAR balance (5.8 / 8.9 dB) from its determined two-microphone form; NG-IVA is weakest throughout. Every online figure sits below its batch counterpart, the expected causal-versus-non-causal gap discussed in Section 2.5.5.

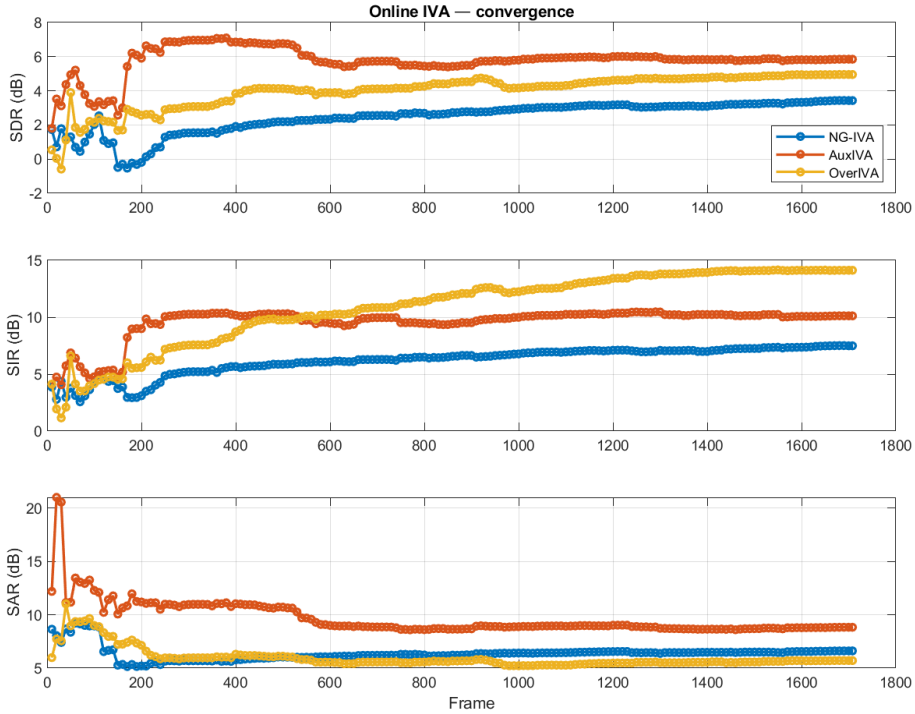


Figure 2.5: Online convergence on the single scenario: SDR, SIR and SAR (dB) versus frame for NG-IVA, AuxIVA and OverIVA. After a noisy cold start, OverIVA's SIR overtakes AuxIVA and keeps climbing as more frames are processed.

Table 2.4: Single-scenario final separation quality (dB), mean over the two sources, for batch (300 iterations) and online (1708 frames). Best per column in bold.

	Batch			**Online**		
Algorithm	SDR	SIR	SAR	SDR	SIR	SAR
NG-IVA	5.5	6.1	15.6	3.4	7.5	6.6
AuxIVA	11.4	12.9	17.4	**5.8**	10.1	**8.9**
OverIVA	**16.8**	**20.7**	**19.3**	5.0	**14.1**	5.5

2.5.4 Comprehensive Grid Evaluation

The single-scenario ranking is confirmed across a grid that varies one acoustic factor at a time, holding all others at the reference configuration of Table 2.3. Each study isolates a distinct stress factor and the hypothesis it probes:

- **Reverberation** (RT60 $\in \{0.1, 0.2, 0.3, 0.4, 0.6, 0.8\}$ s): tests how quickly the narrowband STFT approximation (Section 1.2.3) breaks down as the room impulse response outgrows the analysis frame.
- **Input SNR** (white noise, $\{0, 5, 10, 20, 30, \infty\}$ dB) and **diffuse noise** ($\{0, 10\}$ dB): probe robustness to sensor and field noise, where OverIVA's explicit background model is expected to help most.
- **Angular separation** ($\{5, 10, 20, 40, 80\}^\circ$): measures how close two sources can be before the array can no longer resolve them spatially.

- **Source count** ($L \in \{2, 3, 4\}$): stresses the spatial degrees of freedom of the 16-microphone array as the mixture becomes denser.
- **Speaker content** (male/female, male/male, female/female, speech/music, speech/babble): checks sensitivity to the source statistics underlying the Laplace independence prior.

Figure 2.6 reports the final batch SIR of the three variants for every condition.

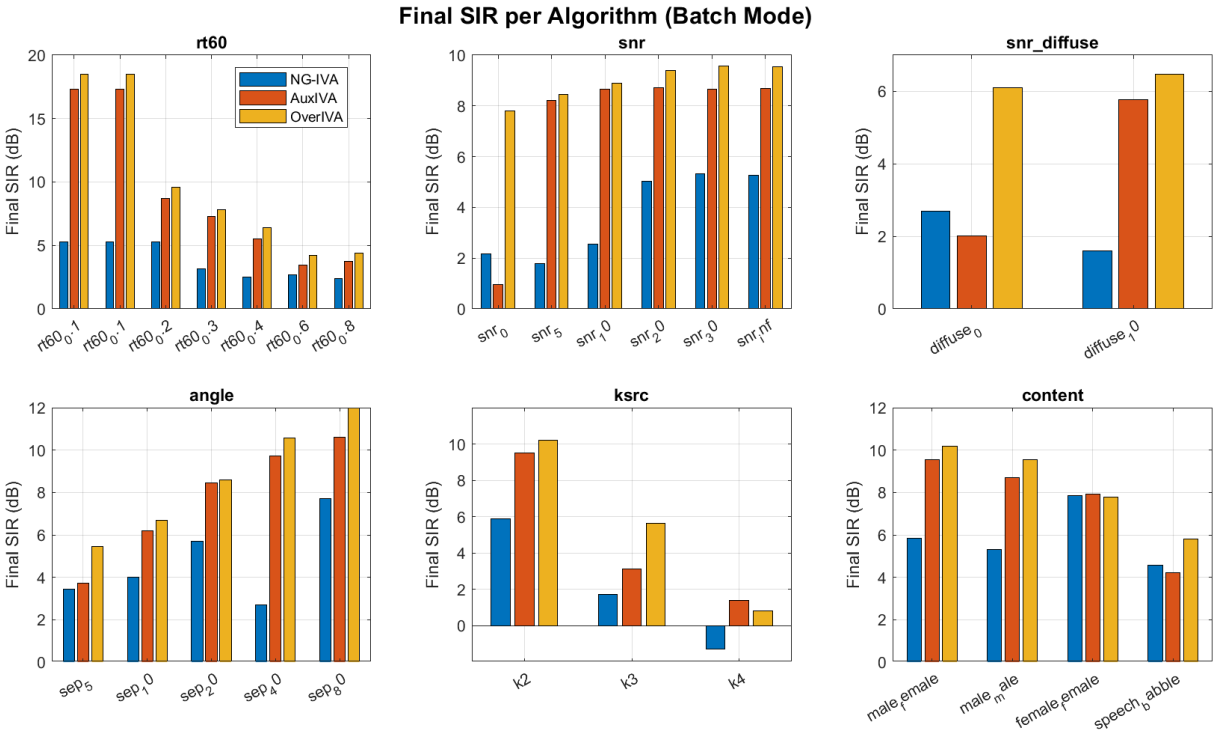


Figure 2.6: Final batch SIR (dB) of NG-IVA, AuxIVA and OverIVA across the evaluation grid: reverberation (RT60), input SNR, diffuse noise, angular separation, source count, and speaker content.

Across all conditions the ordering is stable, **OverIVA** $\geq$ **AuxIVA** $\gg$ **NG-IVA**. SIR falls monotonically with reverberation (from ≈ 18 dB at RT60 0.1 s to ≈ 4 dB at 0.8 s, as the direct path is progressively buried under reverberation), rises with input SNR and with angular separation, and collapses toward 0 dB at four simultaneous sources, where the two extra speakers exhaust the array's spatial degrees of freedom. OverIVA also leads under diffuse noise, where its explicit background model is most useful. The two iterative-projection methods track each other closely, with OverIVA edging ahead wherever the extra microphones help (low reverberation, wide separation, diffuse noise).

The online behaviour over the grid mirrors the single scenario. Figure 2.7 shows a representative frame-by-frame progression: after the cold-start transient all three settle, and OverIVA's SIR climbs steadily past AuxIVA as more frames accumulate, while NG-IVA stays lowest.

2.5.5 Discussion

The empirical results across the grid test reveal distinct behavioral differences between the three IVA variants, largely driven by how each algorithm handles excess microphone channels and spatial covariance.

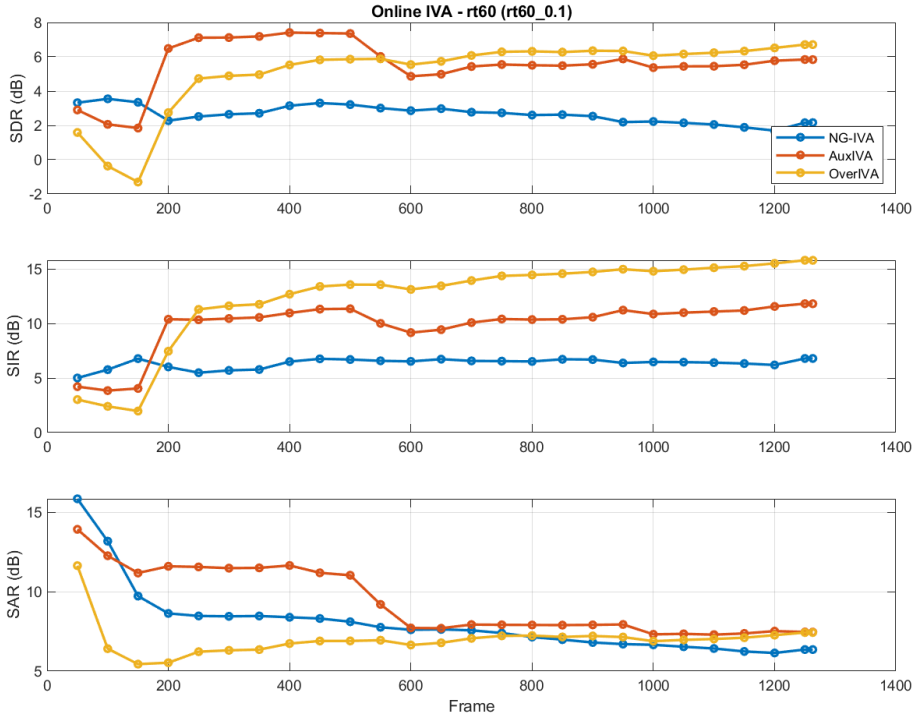


Figure 2.7: Representative online progression (RT60 0.1 s): SDR, SIR and SAR (dB) versus frame. OverIVA's SIR overtakes AuxIVA as the recursive statistics fill in.

OverIVA's dominance in high-noise and diffuse environments:

As observed in the **snr** and **snr_diffuse** batch evaluations, OverIVA dramatically outperforms AuxIVA and NG-IVA at low SNRs (e.g., 0 dB and 5 dB) and in highly diffuse noise fields. This is directly attributable to how OverIVA exploits the overdetermined system ($M > L$). Rather than relying solely on a square demixing matrix or a per-frame dimension reduction (which NG-IVA struggles with), OverIVA utilizes the excess degrees of freedom ($M - L$) to form an orthogonal background constraint, $\mathbf{J}_f$. This effectively steers spatial nulls toward the distributed background noise, preserving the target sources even when the signal-to-noise ratio is exceptionally poor.

Convergence speed and asymptotic ceiling (Batch mode):

The batch convergence plots highlight a stark hierarchy in optimization efficiency. OverIVA converges exceptionally fast, reaching its maximum SDR and SIR within approximately 30 to 40 iterations, while AuxIVA requires roughly 70 iterations to plateau at a lower performance ceiling. NG-IVA, constrained by its damped gradient step, is significantly slower and shows a gradual decline in SAR as iterations progress. OverIVA's rapid convergence is due to its exact, undamped block-coordinate descent, which tightly couples target extraction with simultaneous background orthogonalization.

Online progression and the "Cold-Start" penalty:

While OverIVA achieves the highest steady-state SIR in online processing, its early-frame performance exhibits a distinct "noisy cold start." As seen in the online progression plots, AuxIVA initially provides higher SIR and SAR. Because OverIVA's background nulling relies on $\mathbf{J}_f$, which is constructed from the full 16×16 input spatial covariance matrix, it requires sufficient frames to accumulate stable and robust recursive statistics. Once this covariance history fills in (typically between 200 and 400 frames, depending on the

ramped forgetting factor), OverIVA's SIR decisively overtakes AuxIVA and continues to climb.

Environmental limits: Reverberation and source saturation:

Despite OverIVA's advantages, all algorithms show fundamental limitations under extreme physical acoustics. As RT_{60} increases, the convolutive mixing length exceeds the validity of the narrowband STFT approximation, causing a uniform drop in performance across all variants. Similarly, the **multiple sources tests** (more than 2) results demonstrate that while separating two sources is handled well, scaling to three or four sources severely degrades SIR. We observe that as the number of sources increases, a larger number of iterations may be required to achieve better separation performance.

2.5.6 Algorithm Selection for Implementation

The simulation results of this section lead to a clear implementation strategy for the C++ platform described in Chapter 4.

Online OverIVA is selected as the real-time separation engine. It gives the best interference suppression in every tested condition (Table 2.4 and Figure 2.6), exploits all 16 UMA-16 microphones without a per-frame whitening step, requires no step-size tuning, and runs causally at one update per frame, so it meets the real-time, look-ahead-free requirement of the platform described in Chapter 4.

Online AuxIVA is retained as the determined-case alternative ($M = L$ after channel selection): it offers the best SDR/SAR balance and is the lighter option when only two microphones are used, at the cost of discarding the spatial diversity of the remaining channels.

NG-IVA is kept as the baseline for comparison and for continuity with previous ENP work, but is the weakest and slowest of the three.

2.6 Pre-Processing Integration

2.6.1 Voice Activity Detection in the Adaptive Loop

In the adaptive IVA loop, filter updates should only be performed on frames that contain speech. Silent frames carry no separation information; updating on them causes the covariance estimate $\mathbf{R}_{xx}$ to drift toward an energy-free state and, without the nonholonomic constraint, can cause the demixing filter to diverge.

To address this, Voice Activity Detection (VAD) acts as a gate at the entry of the adaptive IVA loop. We evaluated two classical approaches: an Energy and Zero-Crossing Rate (ZCR) baseline [15], and a statistical Likelihood Ratio Test (LRT) formulated by Sohn et al. [17].

Energy and Zero-Crossing Rate (ZCR):

Let $x_m(n)$ represent the n-th sample of the discrete-time acoustic signal in frame m, with a frame length of N. The short-time energy $E(m)$ and zero-crossing rate $ZCR(m)$ are mathematically defined as:

$$E(m) = \sum_{n=0}^{N-1} x_m^2(n) \tag{2.37}$$

$$ZCR(m) = \frac{1}{2N} \sum_{n=1}^{N-1} |\operatorname{sgn}(x_m(n)) - \operatorname{sgn}(x_m(n-1))| \tag{2.38}$$

where $\operatorname{sgn}(\cdot)$ denotes the standard signum function (a mathematical function that extracts the sign of a real number).

The Energy+ZCR approach flags a frame m as active speech if its energy $E(m)$ exceeds a dynamically calculated threshold E_{th}, or its zero-crossing rate $ZCR(m)$ falls below a set limit ZCR_{th}. This binary decision rule can be expressed as:

$$VAD(m) = \begin{cases} 1, & \text{if } E(m) > E_{th} \vee ZCR(m) < ZCR_{th} \\ 0, & \text{otherwise} \end{cases} \tag{2.39}$$

While computationally trivial and effective in clean conditions (SNR $>$ 15 dB), this method fails significantly in non-stationary noise or at low Signal-to-Noise Ratios (below 10 dB).

Statistical Model-based Voice Activity Detector:

To achieve robust tracking in realistic environments, the Sohn LRT VAD is integrated into the pipeline. This method frames VAD as a per-frequency-bin hypothesis test in the STFT domain, using a Gaussian model for speech and noise coefficients. It computes a composite log-likelihood ratio Λ, weighting each bin by its local SNR:

$$\Lambda = \sum_f \left(-\log(1 + \xi(f)) + \frac{\gamma(f)\xi(f)}{1 + \xi(f)} \right) \tag{2.40}$$

where $\gamma(f)$ is the a posteriori SNR and $\xi(f)$ is the decision-directed a priori SNR. Noise statistics are updated adaptively using soft speech-presence probabilities, allowing the algorithm to model time-varying noise without being corrupted by active speech. The complete frame-by-frame procedure is given in Algorithm 7.

- **Decision-Directed A Priori SNR:** The estimator hinges on the *decision-directed* a priori SNR $\xi(f)$ (line 7): it blends the previous frame's Wiener-filtered clean estimate $\hat{S}(f)$ with the current instantaneous excess $\max(\gamma(f) - 1, 0)$, the mix being set by the smoothing factor α. This recursion suppresses the frame-to-frame fluctuations that would otherwise make the per-bin likelihood ratio $\lambda(f)$ erratic.
- **Dual-Rate Noise PSD Update:** The noise PSD $\sigma_N^2(f)$ is then refreshed at a *dual rate* (line 12): slowly when the frame looks like speech (β) and quickly when it looks like silence (β_{f}), each bin weighted by its speech-absence probability $w(f) = 1 - \sigma(\lambda(f))$ so that active speech cannot corrupt the noise estimate.

Algorithm 7 Streaming Sohn LRT VAD [17]

Require: STFT frame $X(f,n)$; decision-directed factor α; noise-update factors β (speech), β_{f} (silence); threshold θ; hangover H; warm-up frames N_0
Ensure: Binary voice-activity flag $v_n \in \{0,1\}$

Initialise: noise PSD $\sigma_N^2(f) \leftarrow 0$, clean estimate $\hat{S}(f) \leftarrow 0$, $h \leftarrow 0$
for each incoming frame n **do**
  $P(f,n) \leftarrow |X(f,n)|^2$ for all f ▷ per-bin power
  if $n \le N_0$ **then** ▷ warm-up: estimate noise floor
    $\sigma_N^2(f) \leftarrow \sigma_N^2(f) + P(f,n)/N_0$; $v_n \leftarrow 0$; **continue**
  end if
  $\gamma(f) \leftarrow P(f,n)/\sigma_N^2(f)$ ▷ a posteriori SNR
  $\xi(f) \leftarrow \alpha\, \hat{S}(f)/\sigma_N^2(f) + (1-\alpha)\max(\gamma(f)-1, 0)$ ▷ decision-directed a priori SNR
  $\lambda(f) \leftarrow -\log\big(1+\xi(f)\big) + \dfrac{\gamma(f)\,\xi(f)}{1+\xi(f)}$ ▷ per-bin log-LRT
  $\Lambda \leftarrow \frac{1}{F}\sum_f \lambda(f)$; $v_{\mathrm{raw}} \leftarrow [\,\Lambda > \theta\,]$
  $p(f) \leftarrow \sigma\big(\lambda(f)\big)$; $w(f) \leftarrow 1 - p(f)$ ▷ speech-presence / absence prob.
  $\beta_{\mathrm{e}} \leftarrow \beta$ if $v_{\mathrm{raw}} = 1$ else β_{f}
  $\sigma_N^2(f) \leftarrow \beta_{\mathrm{e}}\sigma_N^2(f) + (1-\beta_{\mathrm{e}})\big[w(f)P(f,n) + (1-w(f))\sigma_N^2(f)\big]$
  $\hat{S}(f) \leftarrow \dfrac{\xi(f)}{1+\xi(f)}\,P(f,n)$ ▷ Wiener estimate for next frame
  if $v_{\mathrm{raw}} = 1$ **then** $h \leftarrow H$; $v_n \leftarrow 1$
  else if $h > 0$ **then** $h \leftarrow h-1$; $v_n \leftarrow 1$ ▷ hangover
  else $v_n \leftarrow 0$
  end if
end for

- **Hangover Mechanism:** A final hangover counter h holds the decision active for up to H frames after speech ends, bridging short intra-word pauses.
- **Parameters and Notation:** The parameter values used here are: $\alpha = 0.92$ (decision-directed memory, $\approx$ 12 frames), $\beta = 0.98$ and $\beta_{\mathrm{f}} = 0.85$ (speech/silence noise-update rates), decision threshold $\theta = 0.1$, hangover $H = 8$ frames, and a warm-up of $N_0 = 6$ frames to seed σ_N^2; $\sigma(\cdot)$ denotes the logistic sigmoid and $P(f,n) = |X(f,n)|^2$ the per-bin power.

When the VAD classifies a frame as silence (e.g., $E_{\mathrm{frame}} \le \theta_{\mathrm{VAD}}$ or $\Lambda \le \theta_{\mathrm{VAD}}$):

- The demixing filter $\mathbf{W}_p(f,n)$ is held fixed (no gradient update).
- The covariance $\mathbf{R}_{xx}(f,n)$ and the whitening matrix $\mathbf{Q}(f,n)$ are also held, preserving the learned spatial model.
- The normalization factor $\zeta(f,n)$ continues to decay, which prevents a large step at the next speech-active frame.
- The output $\mathbf{y}(f,n)$ is set to zero or to a comfortable noise floor.

This strategy is visible in Algorithm 2 at lines 4–7.

VAD Simulation and Validation

To objectively evaluate the performance of the Voice Activity Detection algorithms, two primary metrics are defined based on frame-by-frame classification outcomes: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

- **Hit Rate (HR):** Also known as sensitivity or the true positive rate, this measures the percentage of actual speech frames that are correctly identified by the VAD. It is formulated as:
$$HR = \frac{TP}{TP + FN} \times 100\%$$
- **False Alarm Rate (FAR):** Also known as the false positive rate, this measures the percentage of actual silence or noise frames that are incorrectly flagged as active speech. It is formulated as:
$$FAR = \frac{FP}{FP + TN} \times 100\%$$

Simulations comparing the baseline Energy+ZCR and the Sohn LRT VAD across varied SNR conditions demonstrate the robustness of the statistical approach.

Test 1 - Concatenated Isolated Words: As shown in Figure 2.8 and quantified in Table 2.5, the Energy+ZCR method exhibits severe fragmentation and false negatives as the SNR degrades towards 0 dB, completely failing to detect speech at 0 dB. The Sohn LRT, however, retains a hit rate of nearly 60% even under heavy noise while keeping false alarms at zero.

Table 2.5: VAD Performance for Concatenated Isolated Words (Test 1).

SNR Level	Algorithm	Hit Rate (%)	False Alarm Rate (%)
20 dB	Sohn LRT	91.6	0.0
	Energy + ZCR	72.8	0.0
10 dB	Sohn LRT	82.7	0.0
	Energy + ZCR	46.7	0.0
0 dB	Sohn LRT	58.6	0.0
	Energy + ZCR	0.0	0.0

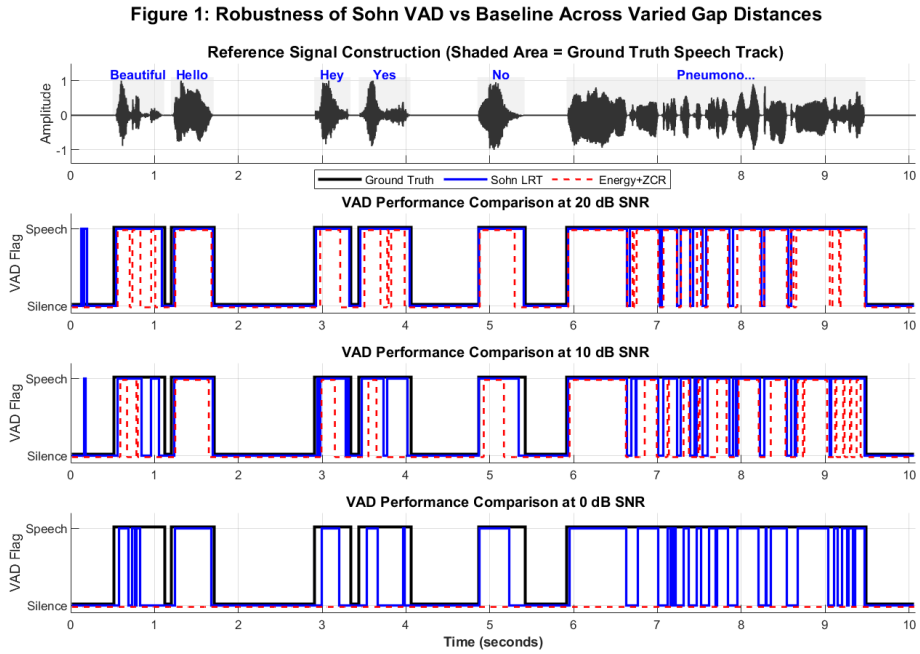


Figure 2.8: Robustness of the Sohn LRT VAD compared to the Energy+ZCR baseline across varying gap distances at 20 dB, 10 dB, and 0 dB SNR. The evaluation utilizes a diverse vocabulary corpus consisting of short words (*Yes*, *No*), medium-length words (*Hello*, *Hey*, *Beautiful*), and an extremely long technical word (*Pneumonoultramicroscopicsilicovolcanoconiosis*) to thoroughly validate phoneme tracking stability and boundary estimation under severe degradation.

Test 2 - Continuous Speech Segmentation: Figure 2.9 illustrates that the Sohn LRT maintains continuous speech and pauses. Table 2.6 corroborates this visually observed stability: while the baseline hit rate drops to 1.9% at 0 dB SNR, the Sohn LRT maintains a 60.3% hit rate with a highly manageable false alarm rate.

Table 2.6: VAD Performance for Continuous Speech Segmentation (Test 2).

SNR Level	Algorithm	Hit Rate (%)	False Alarm Rate (%)
20 dB	Sohn LRT	93.8	17.1
	Energy + ZCR	77.6	10.4
10 dB	Sohn LRT	82.1	10.7
	Energy + ZCR	39.6	5.2
0 dB	Sohn LRT	60.3	7.9
	Energy + ZCR	1.9	0.0

2.6.2 Whitening as the Pre-Processing Bridge

For the adaptive NG-IVA variant, adaptive whitening (Section 2.3.1) is integrated *inside* the IVA update loop and is inseparable from it: the whitening matrix $\mathbf{Q}(f, n)$ is recomputed at every frame before the separation filter is applied.

For AuxIVA and OverIVA, whitening is not required because these algorithms handle the $M \geq L$ case differently: OverIVA models the extra channels explicitly as a noise

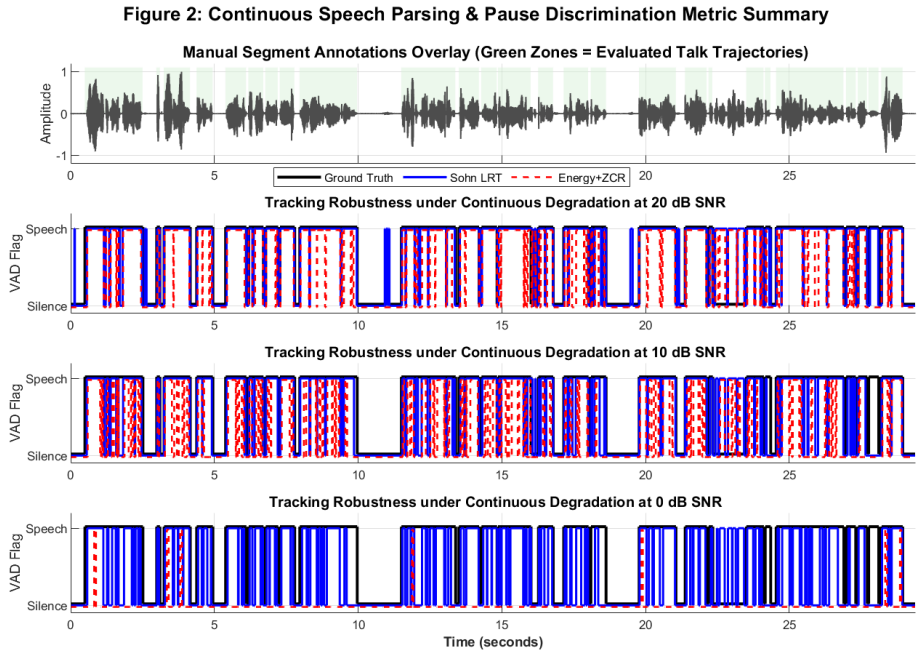


Figure 2.9: Sohn LRT vs Energy+ZCR Continuous speech and pause testing at different SNR levels

background and updates only the L signal rows, while standard AuxIVA simply requires the determined case $M = L$ (in which case one selects $M = L$ representative microphones a priori). This distinction is summarized in Table 2.2 under the "$M{>}L$" column and directly influences the implementation choice for the UMA-16 array.

2.7 Post-Processing Integration

2.7.1 Residual Artefacts After IVA Convergence

Even after convergence of any IVA variant, the separated outputs $\hat{s}_l[t]$ typically contain three categories of residual artefact [6]:

- **Residual cross-talk:** In reverberant conditions, the demixing filter $\mathbf{W}$ cannot perfectly invert $\mathbf{A}(f)$ at every frequency bin: reverberation tails that extend beyond the STFT frame length cause the narrowband approximation to break down, leaving a fraction of the interfering sources audible in each output.
- **Musical noise:** The STFT-domain processing creates isolated non-zero time-frequency bins in otherwise silent regions, perceived as a metallic, tonal background.
- **Amplitude inconsistencies:** Despite MDP rescaling, small residual scaling errors may remain due to model mismatch or numerical precision in the EVD.

These artefacts motivate an optional *postfiltering* stage applied to the IVA output.

2.7.2 Adaptive Leakage Suppression Postfilter

To address residual cross-talk and artefacts, an adaptive Wiener-like leakage suppression mask [43] is applied to the IVA outputs. Rather than assuming a static noise environment, the target signal and interference powers are estimated recursively frame-by-frame.

Let $Y_l(f,n)$ be the separated STFT spectrum of the l-th source. The instantaneous power of the target source is calculated as:

$$\Phi_{yy}^{\text{inst}}(f,n) = |Y_l(f,n)|^2 \tag{2.41}$$

The instantaneous leakage (interference) estimate is derived from the energy of all other separated sources, scaled by a tunable leakage factor γ_{lk}:

$$\Phi_{nn}^{\text{inst}}(f,n) = \gamma_{\text{lk}} \sum_{j \neq l} |Y_j(f,n)|^2 \tag{2.42}$$

To ensure smooth transitions and mitigate the introduction of musical noise, these instantaneous estimates are smoothed temporally using a first-order exponential moving average with a forgetting factor α_{psd}:

$$\Phi_{yy}(f,n) = \alpha_{\text{psd}} \Phi_{yy}(f,n-1) + (1-\alpha_{\text{psd}}) \Phi_{yy}^{\text{inst}}(f,n) \tag{2.43}$$

$$\Phi_{nn}(f,n) = \alpha_{\text{psd}} \Phi_{nn}(f,n-1) + (1-\alpha_{\text{psd}}) \Phi_{nn}^{\text{inst}}(f,n) \tag{2.44}$$

The isolated target speech power is approximated via spectral subtraction as $\Phi_{ss}(f,n) = \max(0, \Phi_{yy}(f,n) - \Phi_{nn}(f,n))$. The postfilter mask applied to the l-th separated output is formulated as a Wiener gain, bounded by a minimum spectral floor $G_{\min}$ to prevent severe nonlinear distortion during speech pauses [6]:

$$G_l(f,n) = \max\left(\frac{\Phi_{ss}(f,n)}{\Phi_{ss}(f,n) + \Phi_{nn}(f,n) + \epsilon}, G_{\min} \right) \tag{2.45}$$

where ϵ is a small regularization constant. The final enhanced output is synthesized as:

$$\hat{S}_l(f,n) = G_l(f,n) Y_l(f,n) \tag{2.46}$$

Its integration point in the pipeline is immediately before the ISTFT reconstruction of each separated source. The complete per-frame procedure is summarized in Algorithm 8.

In words, the postfilter treats the energy that the other separated outputs carry at bin (f,n) as an estimate of the leakage into source l: the leakage factor γ_{lk} sets what fraction of that cross-energy is assumed to survive in the target channel. Both the target and leakage power spectra are smoothed in time by a first-order recursion (factor α_{psd}) to avoid the rapidly varying gains that produce musical noise, the clean-target power is recovered by spectral subtraction, and the resulting Wiener gain is floored at $G_{\min}$ so that no bin is ever fully zeroed. The values used are $\gamma_{\text{lk}} = 0.20$ (leakage fraction), $\alpha_{\text{psd}} = 0.70$ (PSD smoothing, ≈ 3 frames of memory), $G_{\min} = 0.10$ ($-20\,\text{dB}$ gain floor), and $\epsilon = 10^{-12}$ a small constant preventing division by zero.

Algorithm 8 Adaptive Leakage-Suppression Postfilter [43]

Require: Separated STFT outputs $Y_l(f,n)$, $l = 1,\dots,L$; leakage factor γ_{lk}; PSD smoothing α_{psd}; gain floor $G_{\min}$
Ensure: Enhanced outputs $\hat{S}_l(f,n)$

Initialise $\Phi_{yy}^{(l)}(f)$, $\Phi_{nn}^{(l)}(f) \leftarrow 0$ for all l, f
for each frame n **do**
  for $l \leftarrow 1$ **to** L **do**
    for $f \leftarrow 1$ **to** F **do**
      $\Phi_{yy}^{\text{inst}} \leftarrow |Y_l(f,n)|^2$ ▷ target power
      $\Phi_{nn}^{\text{inst}} \leftarrow \gamma_{\text{lk}} \sum_{j \neq l} |Y_j(f,n)|^2$ ▷ leakage from other sources
      $\Phi_{yy}^{(l)}(f) \leftarrow \alpha_{\text{psd}} \Phi_{yy}^{(l)}(f) + (1-\alpha_{\text{psd}}) \Phi_{yy}^{\text{inst}}$
      $\Phi_{nn}^{(l)}(f) \leftarrow \alpha_{\text{psd}} \Phi_{nn}^{(l)}(f) + (1-\alpha_{\text{psd}}) \Phi_{nn}^{\text{inst}}$
      $\Phi_{ss} \leftarrow \max\left(0,\ \Phi_{yy}^{(l)}(f) - \Phi_{nn}^{(l)}(f)\right)$ ▷ spectral subtraction
      $G_l(f,n) \leftarrow \max\left(\dfrac{\Phi_{ss}}{\Phi_{ss} + \Phi_{nn}^{(l)}(f) + \epsilon},\ G_{\min}\right)$
      $\hat{S}_l(f,n) \leftarrow G_l(f,n)\, Y_l(f,n)$
    end for
  end for
end for

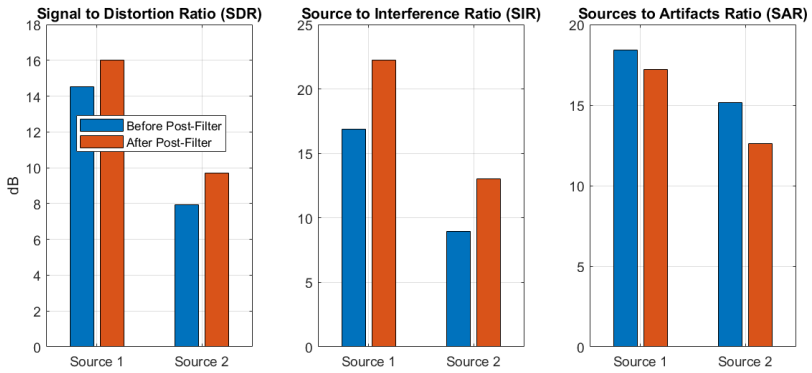


Figure 2.10: Performance evaluation (SDR, SIR, and SAR) before and after the application of the adaptive leakage suppression postfilter. A marked improvement in SIR and SDR is observed.

Post-Processing Validation:

The efficacy of the adaptive postfilter was evaluated using standard Blind Source Separation Evaluation (BSS Eval) metrics: Signal-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR), and Sources-to-Artifacts Ratio (SAR).

As demonstrated in Figure 2.10, applying the postfilter yields a substantial improvement in SIR (often exceeding 4 to 5 dB depending on the source), indicating highly successful suppression of residual cross-talk and leakage from competing speakers. The overall SDR also sees a measurable increase, reflecting a net improvement in global signal quality.

Conversely, a slight degradation is observed in the SAR metric. This is a well-documented trade-off inherent to Wiener and spectral subtraction filtering techniques; aggressive interference suppression inevitably introduces minor non-linear artefacts into the target signal. Nonetheless, the profound gains in spatial isolation (SIR) and overall quality

(SDR) heavily justify the integration of the postfilter for applications requiring strict source separation.

2.8 Conclusion

This chapter established the complete algorithmic foundation for the blind source separation component of the project. It detailed the progression from foundational independent component analysis to independent vector analysis, demonstrating how coupling frequency bins resolves the permutation ambiguity inherent in convolutive speech mixtures. The chapter explored adaptive online methods, emphasizing techniques such as recursive covariance estimation, adaptive whitening, and gradient constraints that maintain system stability during speaker silence while efficiently handling large microphone arrays.

Advanced, step-size-free, and overdetermined variants were also presented as robust alternatives that reduce computational overhead by effectively modeling excess microphone channels as background noise. Through simulation validation, these theoretical frameworks motivated the selection of the most efficient core separation engine for the real-time software implementation.

Finally, the separation process naturally isolates the acoustic spatial footprint of each individual speaker. Because the converged separation filters inherently encode this spatial information, the blind source separation stage directly enables the direction-of-arrival estimation and sound source localization techniques that are developed in the next chapter.

Chapter 3

BSS-Based Sound Source Localization

This chapter addresses the spatial dimension of the project: determining the physical locations of the separated speakers. Rather than treating sound source localization as an independent computational problem applied to raw microphone signals, this chapter demonstrates how the acoustic mixing process is inherently encoded within the converged blind source separation engine. Understanding this integrated approach requires exploring the physical geometry of microphone arrays, the extraction of acoustic transfer functions directly from the demixing matrices, and the specific estimation techniques used to calculate directions of arrival. Ultimately, these methods establish the foundation for projecting real-time spatial coordinates onto a live camera feed, seamlessly bridging the audio separation and visual mapping components of the system.

3.1 Foundations of Array-Based Localization

Every method developed in this chapter, whether it operates on raw microphone signals or on BSS-extracted ATF vectors, exploits the same underlying physical relationship between source direction and inter-microphone propagation delay. This section establishes that relationship and the array manifold it induces for the UMA-16 geometry. As seen in figure 3.1.

3.1.1 The TDOA-Angle Relationship

Under the far-field plane-wave assumption introduced in Chapter 1 (Section 1.2.3), a source located in direction (θ, ϕ), azimuth θ and elevation ϕ, produces a wavefront whose unit propagation direction is:

$$\mathbf{u}(\theta, \phi) = \begin{bmatrix} \cos\phi\cos\theta \\ \cos\phi\sin\theta \\ \sin\phi \end{bmatrix} \tag{3.1}$$

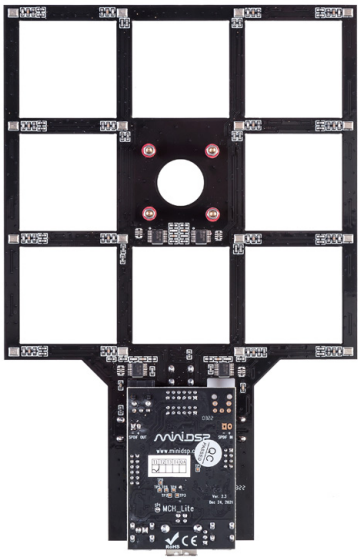


Figure 3.1: UMA-16 miniDSP Microphones Array

The propagation delay from this wavefront to microphone m, relative to a reference microphone r, is:

$$\tau_{mr}(\theta, \phi) = \frac{1}{c}(\mathbf{r}_m - \mathbf{r}_r)^T \mathbf{u}(\theta, \phi) \tag{3.2}$$

where $\mathbf{r}_m \in \mathbb{R}^3$ is the position of microphone m and $c = 343\,\text{m/s}$ is the speed of sound. Equation (3.2) is the master relationship that every localization method in this chapter exploits, whether directly (time-domain TDOA estimation, as in GCC-PHAT and SRP-PHAT, Chapter 1, Section 1.6.1) or indirectly, through its frequency-domain counterpart introduced next.

3.1.2 Common Array Geometries and the Array Manifold

The spatial characteristics of an acoustic source are mathematically captured by the *array manifold*, denoted $\mathcal{A}$. The manifold represents the continuous locus of all possible steering vectors across the parameter space (directions of arrival and frequencies). A steering vector encapsulates the relative phase delays experienced by a plane wave at each microphone compared to a reference point.

The classic texts in array signal processing explicitly step through the geometric derivation of these steering vectors for standard geometries under the far-field plane-wave assumption [44]. The exact algebraic structure of the array manifold depends entirely on the physical arrangement of the microphones. Let us examine how the steering vector is formulated for three fundamental configurations. The different array geometries can be seen in figure 3.3.

Uniform Linear Array (ULA): In a ULA, M sensors are arranged along a single straight line (typically chosen as the x-axis) with a uniform inter-element spacing d. The positions of the microphones are $\mathbf{p}_m = [md, 0, 0]^T$ for $m = 0, 1, \ldots, M-1$.

The arrival wave direction depends only on a single angle θ (measured relative to the array's broadside normal). The relative propagation distance to the m-th sensor is $md\sin\theta$.

For a wavelength $\lambda = c/f$, the ULA steering vector is given by:

$$\mathbf{d}_{\text{ULA}}(\theta) = \begin{bmatrix} 1 \\ e^{-j\frac{2\pi}{\lambda}d\sin\theta} \\ e^{-j\frac{2\pi}{\lambda}2d\sin\theta} \\ \vdots \\ e^{-j\frac{2\pi}{\lambda}(M-1)d\sin\theta} \end{bmatrix} \tag{3.3}$$

The wave impinging on the ULA is illustrated in Figure 3.2:

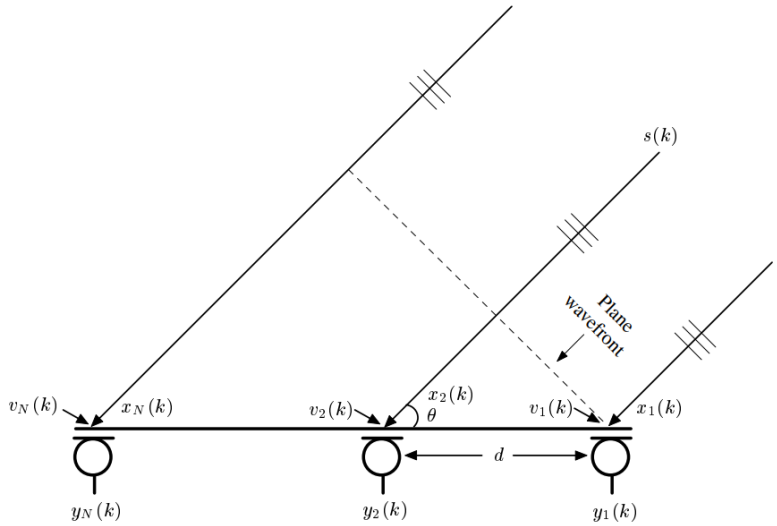


Figure 3.2: Plane-wave propagation toward a uniform linear array (ULA), illustrating the direction of arrival θ and the inter-sensor time delay induced by the wavefront.

Uniform Circular Array (UCA): In a UCA, M sensors are distributed evenly along a ring of radius r on a 2D plane (typically the x-y plane). This geometry provides symmetric 360° coverage. The angular position of the m-th sensor is $\gamma_m = \frac{2\pi m}{M}$, and its spatial coordinates are $\mathbf{p}_m = [r\cos\gamma_m, r\sin\gamma_m, 0]^T$.

An incoming wave arrives from an azimuth angle ϕ and an elevation angle θ. The phase shift depends on projecting the circular position onto the 2D arrival vector. The m-th element of the UCA steering vector is derived as:

$$d_m(\theta,\phi) = e^{j\frac{2\pi}{\lambda}r\sin\theta\cos(\phi-\gamma_m)} \tag{3.4}$$

Uniform Rectangular Array (URA): A URA extends sensors across a 2D grid (N rows and M columns), making it highly effective for estimating both azimuth ϕ and elevation θ simultaneously without the symmetry ambiguities of a linear array. Sensors are spaced at d_x along the x-axis and d_y along the y-axis, placing sensor (n, m) at $\mathbf{p}_{n,m} = [nd_x, md_y, 0]^T$.

Because the wave delays decompose linearly along the orthogonal axes, the URA steering vector can be extracted as the Kronecker product ($\otimes$) of two distinct ULA steering vectors:

$$\mathbf{d}_{\text{URA}}(\theta,\phi) = \mathbf{d}_x(\theta,\phi) \otimes \mathbf{d}_y(\theta,\phi) \tag{3.5}$$

Where the individual element corresponding to the (n, m)-th sensor is:

$$d_{n,m}(\theta, \phi) = e^{-j\frac{2\pi}{\lambda}(nd_x \cos\phi \sin\theta + md_y \sin\phi \sin\theta)} \tag{3.6}$$

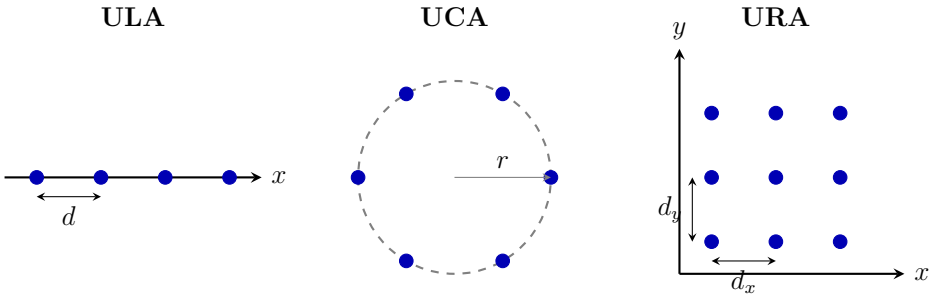


Figure 3.3: Conceptual geometries for Uniform Linear (ULA), Uniform Circular (UCA), and Uniform Rectangular (URA) sensor arrays.

Because our system utilizes the UMA-16 array, which is arranged in a 4×4 grid configuration, we will focus exclusively on the URA geometry for the remainder of this chapter. By mapping the specific URA coordinates into the general delay relationship of Equation (3.2), we derive the separable manifold vector that directly drives our closed-form localization engine.

3.1.3 The URA Array Manifold and Steering Vector

In the frequency domain, the time difference of arrival (TDOA) defined in Equation (3.2) translates directly into a phase shift. The collection of these per-microphone phase shifts at a given frequency bin f forms the *array manifold vector* $\mathbf{d}(\theta, \phi, f) \in \mathbb{C}^M$. As introduced in Chapter 1, this serves as the anechoic steering vector:

$$d_m(\theta, \phi, f) = e^{-j2\pi f \tau_{mr}(\theta,\phi)} \tag{3.7}$$

Application to the UMA-16 Array The UMA-16 features a 4×4 uniform rectangular array (URA) configuration, whose geometry and angle parametrization are illustrated in Figure 3.4. It has a uniform inter-element spacing of $d_x = d_y = d = 42\,\text{mm}$ along both the x- and y-axes.

For this specific geometry, the two-dimensional manifold vector can be elegantly factored into the Kronecker product[1] of two separate one-dimensional steering vectors [44]:

$$\mathbf{d}(\theta, \phi, f) = \mathbf{d}_x(\theta, \phi, f) \otimes \mathbf{d}_y(\theta, \phi, f) \tag{3.8}$$

Spatial Frequencies and Axis Separation To formulate these 1D vectors, we first introduce the *spatial frequencies* along the x- and y-axes, denoted as u and v respectively:

$$u = \cos\phi \cos\theta, \qquad v = \cos\phi \sin\theta \tag{3.9}$$

[1]The Kronecker product, denoted by $\otimes$, is an operation between two matrices (or vectors) that constructs a larger block matrix by multiplying each element of the first operand by the entire second operand.

Using these spatial frequencies, the component steering vectors for our $M_x = M_y = 4$ grid are expressed as:

$$\mathbf{d}_x(\theta, \phi, f) = \left[1,\ e^{j2\pi f du/c},\ \ldots,\ e^{j2\pi f(M_x-1)du/c}\right]^T \tag{3.10}$$

$$\mathbf{d}_y(\theta, \phi, f) = \left[1,\ e^{j2\pi f dv/c},\ \ldots,\ e^{j2\pi f(M_y-1)dv/c}\right]^T \tag{3.11}$$

Closed-Form Angle Recovery This mathematical separability is the structural property that makes the phase-slope method (detailed in Section 3.3.1) tractable in closed form. Rather than performing a computationally expensive joint search over the angles (θ, ϕ), the system can estimate the spatial frequency pair (u, v) independently along each array axis. Once (u, v) are found, the physical angles are recovered instantly via algebraic inversion of Equation (3.9):

$$\theta = \operatorname{atan2}(v, u), \qquad \phi = \arccos\left(\sqrt{u^2 + v^2}\right) \tag{3.12}$$

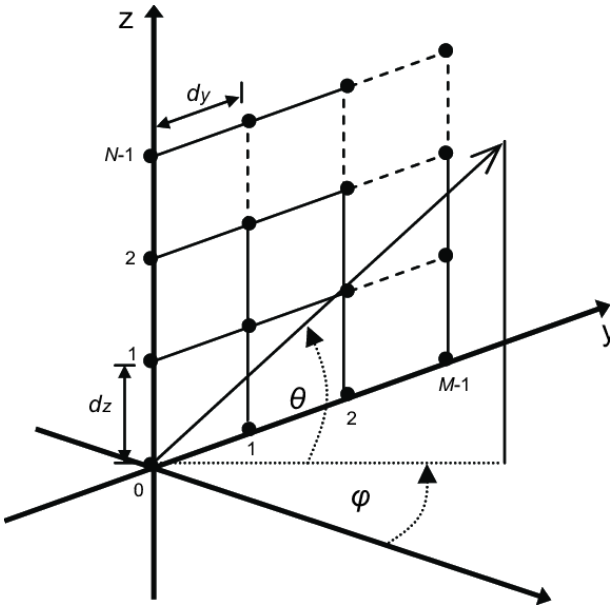


Figure 3.4: Geometry of the UMA-16 uniform rectangular array (URA) and the azimuth-elevation parametrization of a source direction. The unit vector $\mathbf{u}(\theta, \phi)$ points from the array centre toward the source.

3.2 From the BSS Demixing Matrix to ATF Vectors

3.2.1 The MDP Relation and Its Spatial Interpretation

Chapter 2 (Section 2.2.4) established that, after Minimal Distortion Principle (MDP) rescaling, the converged demixing matrix satisfies:

$$\mathbf{W}_s(f)\,\mathbf{A}(f) = \operatorname{diag}(\mathbf{A}(f)) \tag{3.13}$$

Taking the Moore–Penrose pseudo-inverse of $\mathbf{W}_s(f)$ therefore provides a direct estimate of the mixing matrix:

$$\hat{\mathbf{A}}(f) = \mathbf{W}_s(f)^{\#} \tag{3.14}$$

This is the central bridge exploited throughout this chapter: *the demixing matrix that BSS estimates for separation already contains the spatial information needed for localization, with no additional processing stage required to obtain it.* Each column $\hat{\mathbf{a}}_l(f)$ of $\hat{\mathbf{A}}(f)$ is an estimate of the acoustic transfer function (ATF) vector of source l, its spatial fingerprint across all M microphones at frequency bin f. In the anechoic far-field limit, this fingerprint reduces exactly to the array manifold vector: $\hat{\mathbf{a}}_l(f) \approx \mathbf{d}(\theta_l, \phi_l, f)$, the object defined in Section 3.1.3. In a real reverberant room, $\hat{\mathbf{a}}_l(f)$ also contains contributions from early echoes and late reverberation (Chapter 1, Section 1.3.1), but the direct-path phase structure dominates whenever the direct-to-reverberant ratio (DRR) is sufficiently high.

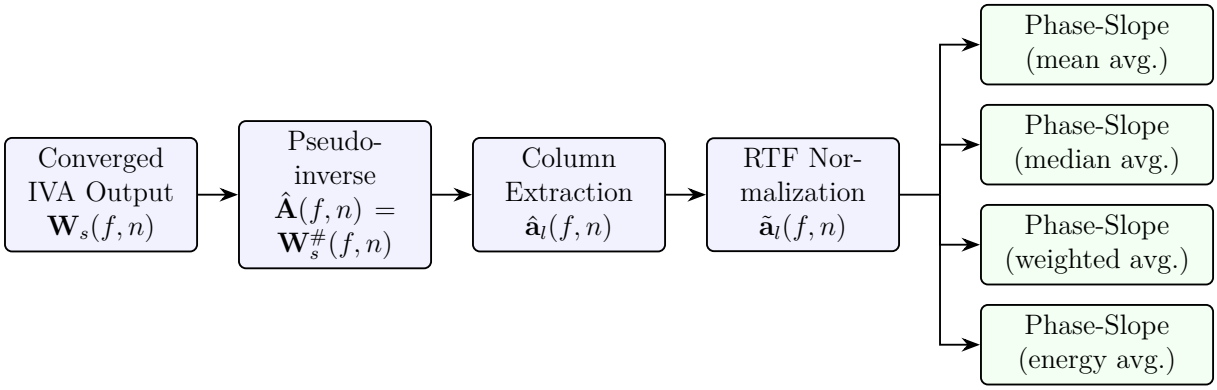


Figure 3.5: From the BSS demixing matrix to direction-of-arrival estimates. The pseudo-inverse of the converged, MDP-rescaled demixing matrix $\mathbf{W}_s(f,n)$ yields per-source ATF columns $\hat{\mathbf{a}}_l(f,n)$. After RTF normalization, these spatial footprints feed parallel DoA estimation methods (detailed in Section 3.3).

3.2.2 Normalization to the Relative Transfer Function

Following Chapter 1, eq. (1.14), the extracted ATF column is normalized with respect to a reference microphone (index 1) to obtain the relative transfer function (RTF):

$$\hat{\mathbf{r}}_l(f) = \frac{\hat{\mathbf{a}}_l(f)}{\hat{a}_{l,1}(f)} \tag{3.15}$$

This normalization removes the dependence on the unknown source spectrum $S_l(f,n)$ and on the absolute amplitude and phase scaling left undetermined by BSS, retaining only the spatial information encoded in the inter-microphone phase and level differences. Every DoA method described in Section 3.3 operates on $\hat{\mathbf{r}}_l(f)$ rather than directly on $\hat{\mathbf{a}}_l(f)$.

3.2.3 The Residual Permutation Ambiguity

Even after IVA has converged with MDP rescaling, the *assignment* of column index to physical speaker is not guaranteed to remain fixed over time. Because the online adaptive filter (Chapter 2, Section 2.3) updates continuously, a permutation swap can occur between consecutive processing blocks: the column that corresponded to speaker A at frame n may correspond to speaker B at frame $n + K$. If left unresolved, this corrupts both the localization output (the DoA marker for "speaker 1" would jump between two physical locations) and any downstream tracking.

To resolve this ambiguity under the assumption that the physical speaker sources are relatively stationary, we can establish a consistent indexing scheme based on signal energy. By evaluating the corresponding steering vectors and ranking them by their energy from largest to smallest, we can deterministically assign the output indices. The source associated with the highest-energy steering vector is assigned as source 1, the next largest as source 2, and so forth. This energy-based sorting continuously re-aligns the columns, stabilizing the output assignments across processing blocks.

3.2.4 Frequency Averaging for Robustness

A single frequency bin's RTF estimate $\hat{\mathbf{r}}_l(f)$ is noisy, particularly in reverberant conditions where the narrowband approximation degrades (Chapter 1, Section 1.3.3). All DoA methods in this chapter therefore aggregate information across a band of frequency bins $\mathcal{F}$, in practice 300–3800 Hz (bounded below by aperture-limited resolution and above by spatial aliasing), weighting each bin's contribution by its reliability. Two weighting strategies are used, depending on the method:

- *(i)* magnitude weighting, where bins with larger $\|\hat{\mathbf{a}}_l(f)\|_2$ are trusted more, since these correspond to frequencies where source l carries more energy relative to the noise floor
- *(ii)* robust statistics across bins (median or trimmed mean of the per-bin angle estimates) rather than a simple arithmetic mean, which is sensitive to outlier bins corrupted by spectral nulls of the room transfer function.

3.3 Direction-of-Arrival Estimation Methods

This section presents the DoA estimation methods considered in this work. The method implemented and evaluated in detail is the closed-form *phase-slope* estimator, which operates directly on the BSS-extracted RTF $\hat{\mathbf{r}}_l(f)$ and is studied through its five frequency-averaging variants (A1–A5). A *dictionary-matching* alternative, which replaces the closed-form inversion with a model-matching search over a precomputed steering-vector grid, is described as a BSS-based counterpart. *SRP-PHAT* serves as the classical, non-BSS-based reference inherited from prior ENP work [4], against which the phase-slope method is to be benchmarked.

3.3.1 Phase-Slope Estimation

The phase-slope method exploits the separable manifold structure of eq. (3.8) directly: along each array axis, the phase of the steering vector varies *linearly* with element index and with frequency, at a rate set exactly by the spatial frequencies u and v [44]:

$$\Delta\psi_x(f) = \frac{2\pi f d_x}{c}\cos\phi\cos\theta = \frac{2\pi f d_x}{c}\,u \tag{3.16}$$

$$\Delta\psi_y(f) = \frac{2\pi f d_y}{c}\cos\phi\sin\theta = \frac{2\pi f d_y}{c}\,v \tag{3.17}$$

where $\Delta\psi_x(f)$ and $\Delta\psi_y(f)$ are the phase increments between adjacent elements along the x- and y-axes, respectively. Because these increments are obtained directly from the extracted RTF $\hat{\mathbf{r}}_l(f)$, no eigendecomposition and no spatial grid search are required, making this a fully closed-form, computationally minimal estimator.

Procedure:

1. Reshape the M-element RTF column $\hat{\mathbf{r}}_l(f)$ into an $M_x \times M_y$ grid matching the physical URA layout.
2. For each frequency bin $f \in \mathcal{F}$, compute the row-wise and column-wise inter-element phase differences, giving per-bin estimates $\widehat{\Delta\psi}_x(f)$ and $\widehat{\Delta\psi}_y(f)$.
3. Convert each per-bin phase increment to a per-bin spatial-frequency pair $(\hat{u}(f), \hat{v}(f))$, then aggregate across $\mathcal{F}$ using one of the averaging variants defined below to obtain stable estimates, from which the angles follow via eq. (3.12).

Averaging variants: A single bin's phase-slope estimate is noisy, so the per-bin estimates are aggregated over the band $\mathcal{F}$. Five aggregation strategies are implemented and compared in Section 3.5:

- **A1, mean:** arithmetic mean of the per-bin angle estimates over $\mathcal{F}$.
- **A2, median:** the per-bin median, which rejects outlier bins (spectral nulls, residual cross-talk, near-wrapping bins) instead of letting them bias the average.
- **A3, energy-weighted mean:** a weighted mean in which each bin is weighted by the separated-source power $|Y_l(f)|^2$, on the rationale that higher-energy bins have better phase SNR.
- **A4, presence-weighted mean:** as A3 but restricted to the *active* bins whose power exceeds a fraction (here 10%) of the source's peak, mimicking the sparse time-frequency assumption.
- **A5, linear phase fit:** a least-squares fit of the *unwrapped* inter-element phase against frequency; since the phase is linear in f with slope proportional to the spatial frequency (eqs. (3.16)–(3.17)), the slope yields the estimate directly. This is equivalent to a wideband TDOA estimate.

Phase wrapping:

Because $\Delta\psi_x, \Delta\psi_y$ are wrapped phases, the method is only unambiguous below the array's spatial-aliasing frequency ($f_{\max} = c/2d \approx 4\,\text{kHz}$ for the 42 mm UMA-16 spacing), which is the general aliasing limit of the array, not an additional restriction specific to this method. The mean- and median-based variants (A1–A4) operate on the per-bin angle estimates and so rely on each bin lying below $f_{\max}$, whereas the linear-fit variant (A5) first unwraps the phase-versus-frequency sequence so that the global slope is recovered even across individual 2π transitions.

The full procedure, from the converged demixing matrix to a per-source direction estimate, is summarized in Algorithm 9.

Algorithm 9 BSS-Based Phase-Slope DoA Estimation

Require: Converged, MDP-rescaled demixing matrices $\mathbf{W}_s(f)$, $f \in \mathcal{F}$; URA layout $M_x \times M_y$ with spacing d; aggregation variant $A \in \{\text{A1}, \ldots, \text{A5}\}$; speed of sound c
Ensure: Direction estimate $(\hat{\theta}_l, \hat{\phi}_l)$ for each source l

for $f \in \mathcal{F}$ **do**
  $\hat{\mathbf{A}}(f) \leftarrow \mathbf{W}_s(f)^{\#}$ ▷ ATF estimate, Eq. (3.14)
end for
for each source l **do**
  for $f \in \mathcal{F}$ **do**
    $\hat{\mathbf{a}}_l(f) \leftarrow$ column l of $\hat{\mathbf{A}}(f)$; normalize to the RTF $\hat{\mathbf{r}}_l(f)$ (Eq. (3.15))
    reshape $\hat{\mathbf{r}}_l(f)$ to the $M_x \times M_y$ URA grid
    $\widehat{\Delta\psi}_x(f)$, $\widehat{\Delta\psi}_y(f) \leftarrow$ mean inter-element phase $\angle\big(r_{m+1}\, r_m^*\big)$ along x, y
    $\hat{u}(f) \leftarrow \dfrac{\widehat{\Delta\psi}_x(f)\, c}{2\pi f d}$, $\hat{v}(f) \leftarrow \dfrac{\widehat{\Delta\psi}_y(f)\, c}{2\pi f d}$; clamp to $[-1, 1]$
    $\big(\theta(f), \phi(f)\big) \leftarrow$ angle recovery (Eq. (3.12))
  end for
  $(\hat{\theta}_l, \hat{\phi}_l) \leftarrow \mathrm{AGG}_A\big(\{\theta(f), \phi(f)\}_{f\in\mathcal{F}};\, \{|Y_l(f)|^2\}\big)$ ▷ A1 mean / A2 median / A3–A4 (energy/presence)-weighted / A5 linear fit
end for
Re-index sources by descending ATF energy $\sum_f \|\hat{\mathbf{a}}_l(f)\|_2^2$ ▷ resolve residual permutation (Sec. 3.2.3)

3.3.2 Precomputed Dictionary Matching

An alternative to the closed-form phase-slope inversion is to match the extracted RTF against a precomputed dictionary of theoretical steering vectors over a discretized spatial grid $\Omega_{\text{grid}} = \{(\theta_g, \phi_g)\}$ covering the UMA-16's field of view:

$$\big(\hat{\theta}_l, \hat{\phi}_l\big) = \underset{(\theta,\phi)\in\Omega_{\text{grid}}}{\arg\min} \sum_{f\in\mathcal{F}} \big\|\hat{\mathbf{r}}_l(f) - \tilde{\mathbf{d}}(\theta, \phi, f)\big\|_2^2 \tag{3.18}$$

where $\tilde{\mathbf{d}}(\theta, \phi, f)$ is the theoretical steering vector of eq. (3.8), normalized identically to $\hat{\mathbf{r}}_l(f)$. This formulation produces a spatial likelihood map whose global minimum over Ω_{grid} gives the DoA estimate. A coarse-to-fine hierarchical search, first scanning a sparse grid, then refining locally around the coarse peak, reduces the computational cost from $\mathcal{O}(|\Omega_{\text{grid}}|)$ to a small fraction of it, following the same two-stage principle as the hierarchical SRP-PHAT search of Bouznad [4]. Unlike the closed-form phase-slope method, dictionary matching naturally accommodates non-ideal array responses (e.g. a measured rather than theoretical dictionary obtained during calibration), at the cost of a grid search rather than a direct algebraic solution. This approach is also consistent with the steering-vector extraction motivation underlying online IVA implementations such as that of Nakashima & Ono [31], where the instantaneous demixing matrix is similarly used as a spatial cue alongside separation.

3.3.3 SRP-PHAT: Classical Reference Method

Steered Response Power with Phase Transform (SRP-PHAT) [34], already introduced in Chapter 1 (Section 1.6.1) and implemented in the previous ENP project by Bouznad [4], operates directly on the *raw* microphone signals rather than on any BSS output. For a

candidate direction (θ, ϕ), it accumulates GCC-PHAT [32] values across all microphone pairs:

$$P(\theta, \phi) = \sum_{i<j} R_{ij}^{\text{PHAT}}\Big(\tau_{ij}(\theta, \phi)\Big) \tag{3.19}$$

with $\tau_{ij}(\theta, \phi)$ given by eq. (3.2). The direction maximizing $P(\theta, \phi)$ is taken as the DoA estimate; in multi-speaker scenes, multiple peaks are sought via hierarchical search and peak suppression, as detailed in Bouznad's implementation [4].

Because $P(\theta, \phi)$ accumulates contributions from *all* simultaneously active sources into a single global map, it is not source-separated: peaks belonging to different speakers can interact, merge, or partially mask each other, especially at close angular separation or in reverberant conditions. This is the fundamental limitation that motivates the BSS-based methods of this chapter, which instead apply their search (or closed-form solution) to a *source-specific* quantity, the extracted RTF $\hat{\mathbf{r}}_l(f)$, that is, by construction, free of contributions from the other active speakers. A related hybrid strategy, separated-signal beamforming (SPBF), applies SRP-PHAT-style localization to *already-separated* source signals rather than to the raw mixture [45], confirming empirically that operating on source-separated input improves localization accuracy over the raw-mixture baseline, the same principle exploited directly in this chapter through ATF extraction from $\mathbf{W}(f, n)$ rather than through re-separation of the time-domain signal. SRP-PHAT is retained throughout this work strictly as a classical, non-BSS-based comparison baseline.

3.3.4 Method Comparison: Theoretical Analysis

Table 3.1 summarizes the theoretical trade-offs between the two methods along the axes most relevant to this project's real-time, multi-speaker, reverberant operating conditions.

Table 3.1: Theoretical comparison of the DoA estimation methods. "BSS-based." indicates whether the method requires BSS-extracted, source-separated input. "RT" = real-time capable on the target hardware.

Method	BSS-based.	RT	Accuracy	Reverb robust.	Complexity
Phase-slope	Yes	Yes	Medium	Low	Minimal
SRP-PHAT	No	Yes	Medium–High	High	Medium–High

The phase-slope method offers the lowest computational cost and a fully closed-form solution but is the most sensitive to reverberation, since it relies on strict phase linearity across the array. SRP-PHAT remains the most robust to reverberation in the single-speaker case, its PHAT weighting was specifically designed for that purpose [32], but its lack of source separation is expected to degrade its relative advantage as the number of simultaneous speakers increases, a hypothesis tested directly in Section 3.5.

3.4 Camera Overlay and 3D-to-2D Projection

3.4.1 From DoA Vector to Camera Pixel

Once a DoA estimate $\mathbf{z}_l(n) = (\hat{\theta}_l, \hat{\phi}_l)$ is obtained by any of the methods in Section 3.3, it is converted to a 3D unit direction vector $\hat{\mathbf{d}}_l$ via eq. (3.1), and projected onto the image

plane of the co-located camera using the standard pinhole camera model:

$$\mathbf{p}_l = \pi\left(\mathbf{K}\,[\mathbf{R}\,|\,\mathbf{t}]\begin{bmatrix}\hat{\mathbf{d}}_l\\ 1\end{bmatrix}\right) \tag{3.20}$$

Equation (3.20) chains the three stages of the pinhole camera model, illustrated in Figure 3.6: the extrinsics place the source direction in the camera frame, the intrinsics turn that 3D ray into image-plane coordinates, and the perspective division collapses the ray onto a single pixel. Each term is defined below.

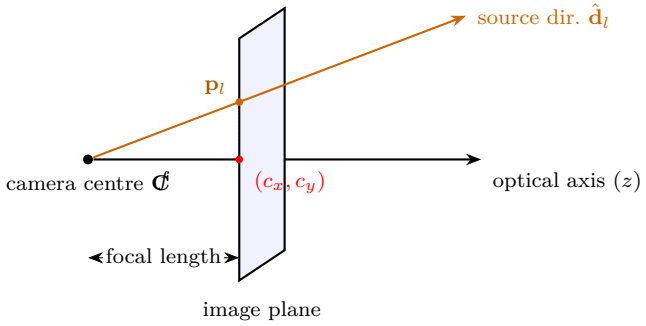


Figure 3.6: The pinhole camera model of Eq. (3.20): a source direction $\hat{\mathbf{d}}_l$ (orange ray) pierces the image plane, placed one focal length f in front of the camera centre, at the projected pixel $\mathbf{p}_l$. The principal point (c_x, c_y) is where the optical axis meets the image plane.

Extrinsics $(\mathbf{R}, \mathbf{t})$: (where the camera is) The rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t} \in \mathbb{R}^3$ form the rigid transformation that re-expresses a point given in the *microphone-array* coordinate frame in the *camera* coordinate frame, $\mathbf{X}_{\text{cam}} = \mathbf{R}\,\mathbf{X}_{\text{array}} + \mathbf{t}$. Intuitively, $\mathbf{R}$ accounts for any angular tilt between the array and the camera optical axis, and $\mathbf{t}$ for the physical offset between their origins. Because the webcam is seated in the UMA-16's central aperture, its optical centre is nearly coincident with the array reference point, so $\mathbf{R} \approx \mathbf{I}_3$ and $\|\mathbf{t}\|$ is only a few millimetres.

Intrinsics K : (how the camera maps rays to pixels) The intrinsic matrix converts a 3D point expressed in the camera frame into pixel coordinates:

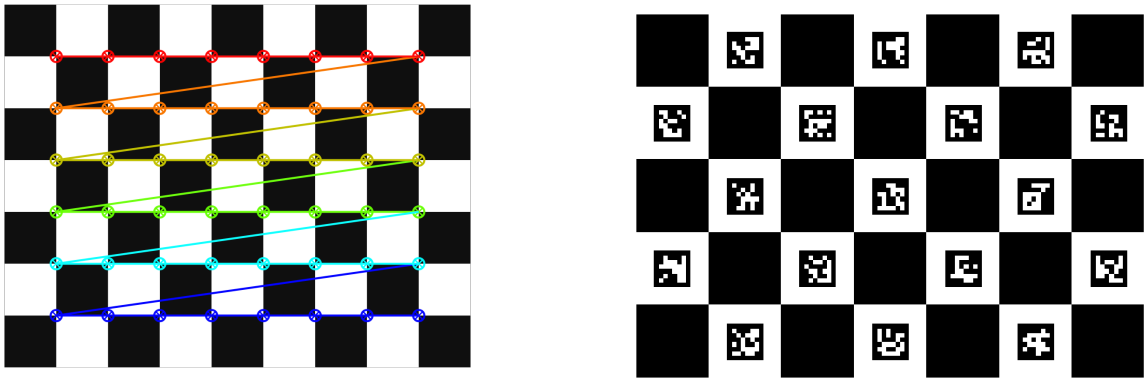
$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.21}$$

where f_x, f_y are the focal lengths expressed in pixels (the physical focal length scaled by the horizontal and vertical pixel densities of the sensor), and (c_x, c_y) is the *principal point*, the pixel at which the optical axis pierces the image plane (ideally the image centre). These four numbers depend only on the camera and lens, not on the scene, which is why they are estimated once during calibration (Section 3.4.2).

Perspective division $\pi(\cdot)$: (ray to pixel) The product $\mathbf{K}[\mathbf{R}\,|\,\mathbf{t}][\hat{\mathbf{d}}_l; 1]$ yields a homogeneous triple $[x, y, z]^T$; the operator $\pi([x, y, z]^T) = (x/z,\ y/z)$ performs the perspective division that maps it to the final (u, v) pixel. This division is what makes more distant objects project smaller, and, as shown next, is also what makes the projected pixel independent of the source distance.

3.4.2 Camera Calibration

The intrinsic parameters of Eq. (3.21) were estimated through a dedicated calibration campaign, following the same principle as Bouznad [4] but using a *ChArUco* board (a chessboard fused with ArUco fiducial markers) rather than a plain checkerboard. The ChArUco pattern combines the sub-pixel corner accuracy of a chessboard with the robust, uniquely-identifiable markers of ArUco, so corners are still detected reliably even when the board is partially out of frame or viewed at a steep angle, which makes the capture step faster and more repeatable. Figure 3.7 contrasts the two patterns.



(a) Checkerboard target. (b) ChArUco target.

Figure 3.7: Calibration targets: a plain checkerboard (a) versus the ChArUco board (b) adopted here, whose embedded ArUco markers allow corner identification even under partial occlusion.

The procedure follows Zhang's planar method [46] as implemented in OpenCV: a set of board views is captured across the field of view and at varied orientations; the board corners are detected to sub-pixel precision; and the intrinsics $\mathbf{K}$ together with the lens-distortion coefficients are obtained by minimizing the total reprojection error over all views. For the webcam used here (640×480), the calibration yielded

$$\mathbf{K} \approx \begin{bmatrix} 491.9 & 0 & 328.5 \\ 0 & 490.9 & 242.1 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.22}$$

that is $f_x \approx 492$, $f_y \approx 491\,\text{px}$ and a principal point $(c_x, c_y) \approx (328.5, 242.1)\,\text{px}$, close to the nominal image centre $(320, 240)$ as expected for a well-centred sensor. The mean reprojection error over the calibration set was $\approx 0.30\,\text{px}$, confirming a tight fit. The estimated radial/tangential distortion coefficients are used to undistort each incoming frame before the overlay is drawn. A representative selection of the captured calibration views, with the detected board corners overlaid, is shown in Figure 3.8.

Regarding the *extrinsics*: because the webcam is mounted coaxially in the array aperture, $\mathbf{R} \approx \mathbf{I}_3$ and $\mathbf{t} \approx \mathbf{0}$, so the array and camera frames are treated as aligned up to the few-millimetre mounting offset. As the next section shows, this near-zero translation is precisely what allows the depth of each source to be ignored entirely when projecting its direction onto the image.

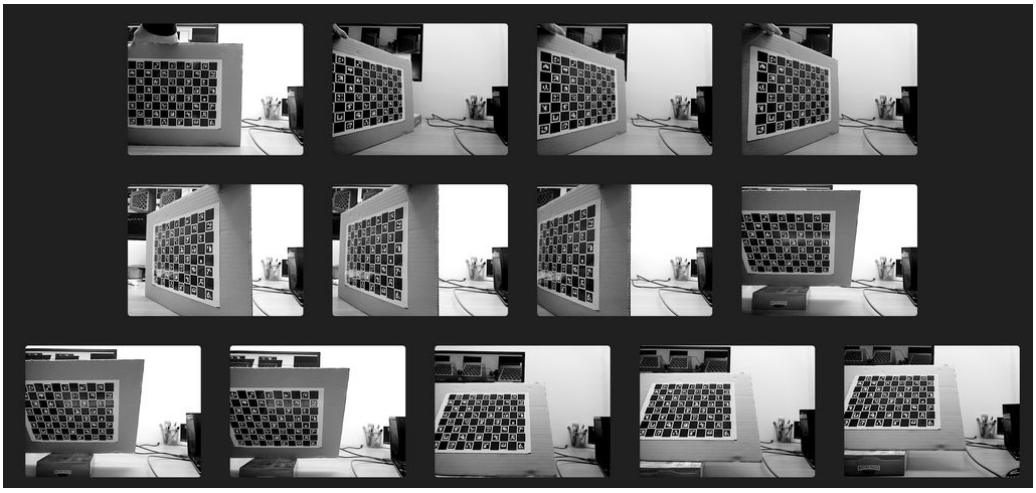


Figure 3.8: Sample calibration captures used to estimate the camera intrinsics of Eq. (3.22), with detected ChArUco corners overlaid. Took from [47]

3.4.3 Depth Invariance of the Projection

A direction-of-arrival estimate (θ_l, ϕ_l) specifies a *ray* originating at the array, not a point in space: the source could lie anywhere along $\hat{\mathbf{d}}_l$ depending on its (unknown) distance from the array. Bouznad's implementation [4] addressed this by estimating the source depth via monocular face detection, in order to refine the 3D position before projecting it. The following result shows that, for the coaxial camera mounting used here, this depth-estimation step is unnecessary.

Proposition (depth invariance). *When the camera optical centre coincides with the array reference point ($\mathbf{t} = \mathbf{0}$), the pixel onto which a direction $\hat{\mathbf{d}}_l$ projects is independent of the assumed source depth $\lambda > 0$.* Consider scaling the direction vector by an arbitrary positive depth $\lambda > 0$, representing a candidate distance along the ray. The corresponding homogeneous world point is $[\lambda\hat{\mathbf{d}}_l;\, 1]$, and its projection is:

$$\pi\left(\mathbf{K}[\mathbf{R}|\mathbf{t}]\begin{bmatrix}\lambda\hat{\mathbf{d}}_l\\ 1\end{bmatrix}\right) = \pi\big(\lambda\,\mathbf{K}\mathbf{R}\hat{\mathbf{d}}_l + \mathbf{K}\mathbf{t}\big) \tag{3.23}$$

If $\mathbf{t} = \mathbf{0}$, the camera origin coincides exactly with the array's reference point, the right-hand side becomes $\pi(\lambda\,\mathbf{K}\mathbf{R}\hat{\mathbf{d}}_l)$, and since $\pi(\cdot)$ is invariant to positive scaling of its argument ($\pi(\lambda\mathbf{v}) = \pi(\mathbf{v})$ for any $\lambda > 0$, because the scaling cancels in the ratio x/z, y/z), the projected pixel $\mathbf{p}_l$ is *identical for every assumed depth* λ.

In practice $\mathbf{t} \neq \mathbf{0}$ exactly, but for the UMA-16's coaxial mounting, $\|\mathbf{t}\|$ is on the order of a few millimetres, negligible compared to typical speaker-to-array distances of 0.5–3 m. The depth-dependent term $\mathbf{K}\mathbf{t}$ in eq. (3.23) is therefore a small, constant offset, and the projected pixel position varies negligibly with the assumed depth λ over the entire practically relevant range. This justifies omitting any depth estimation step (such as the face-detection-based approach used by Bouznad [4]): the DoA estimate alone, projected at any convenient fixed nominal depth, is sufficient to place the marker at the correct pixel location in the live video feed.

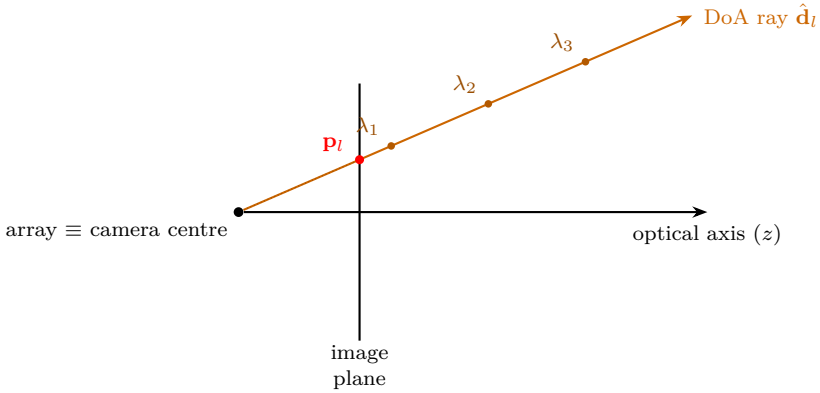


Figure 3.9: Depth invariance of the perspective projection. Three candidate source positions along the same DoA ray, at different depths $\lambda_1 < \lambda_2 < \lambda_3$, project to the same pixel $\mathbf{p}_l$ on the camera image plane, since the camera origin is approximately coincident with the array reference point ($\mathbf{t} \approx \mathbf{0}$).

3.5 Simulation Validation

This section validates the BSS-based phase-slope estimator of Section 3.3.1 through controlled MATLAB simulation, using the same synthetic mixture generation pipeline as Chapter 2 (Section 2.5.1), now with known ground-truth source angles. The separation front-end is the online OverIVA engine selected in Chapter 2: its converged per-bin demixing matrix is turned into ATF columns (Section 3.2) from which the phase-slope DoA is computed. The five frequency-averaging variants introduced in Section 3.3.1 (A1 mean, A2 median, A3 energy-weighted mean, A4 presence-weighted mean, A5 linear phase fit) are compared across three test families: stationary accuracy, angular resolution, and non-zero-elevation accuracy. SRP-PHAT, the classical reference inherited from prior ENP work [4], is integrated into the same evaluation harness but its results are not yet reported here; the corresponding entries are left as placeholders to be completed in a subsequent pass.

3.5.1 Simulation Setup

Table 3.2 summarizes the simulation configuration. The operating conditions are deliberately benign, moderate reverberation (RT60 = 200 ms) and high SNR (30 dB), because the goal here is to characterize the intrinsic behaviour of the estimator and its averaging variants rather than its robustness to noise or reverberation; the latter is a separate study not pursued in this work.

Accuracy metric:

The accuracy metric is the *mean absolute error* (MAE). For a given test family containing S scenarios, each with L sources, the MAE of a quantity (azimuth or elevation) is the mean absolute difference between the estimated and the true angle, averaged over all sources and scenarios:

$$\mathrm{MAE} = \frac{1}{S\,L} \sum_{s=1}^{S} \sum_{l=1}^{L} \left| \hat{\theta}_{s,l} - \theta_{s,l} \right|, \tag{3.24}$$

reported separately for azimuth and elevation in degrees. Because BSS assigns its output channels in an arbitrary order, the estimated sources are matched to the ground truth by the permutation that minimizes the total azimuth error before (3.24) is evaluated.

Table 3.2: MATLAB simulation setup for the DoA estimation experiments.

Parameter	**Value**
Room dimensions	$4 \times 3 \times 2.5\,\mathrm{m}$
Microphone array	UMA-16, 4×4 vertical URA, $d = 42\,\mathrm{mm}$
Number of sources L	2
Source distance	0.6 m (far field)
RT60	200 ms (clean sims: anechoic-like)
Input SNR	30 dB (white)
Sampling rate f_s	16 kHz
STFT	NFFT = 512, hop = 256 (50% overlap), Hann
Frequency band $\mathcal{F}$	300–3800 Hz
Analysis length	15 s ($\approx$ 1 s warm-up)
Averaging variants	A1–A5 (Section 3.3.1)
Realizations	one per scenario
Metrics	azimuth / elevation MAE (°)

The three test families are drawn from the simulated scenario set: *stationary* sources at fixed, well-separated azimuths with varied speaker content; an *angular resolution* sweep with the two sources placed from 80° down to 5° apart; and a set of *non-zero-elevation* scenes in which both sources sit at distinct azimuth and elevation angles, used to probe the array's weaker vertical resolution. Each result is a single representative realization per scenario; the reported family-level MAE is the average over the scenarios in that family.

SRP-PHAT via separated-signal reconstruction:

SRP-PHAT cannot separate sources, so it is supplied with one *single-source* signal at a time. For each separated source l, its 16-channel image is reconstructed from the BSS output as $\mathbf{X}_l(f,n) = \hat{\mathbf{a}}_l(f)\, S_l(f,n)$, the product of the extracted ATF column (Section 3.2) and the separated-source spectrum S_l; the inverse STFT then yields a 16-channel waveform on which Bouznad's hierarchical SRP-PHAT [4] is run. This separated-signal beamforming strategy is evaluated on the two families where it is affordable, stationary (azimuth only) and non-zero-elevation; it is omitted from the angular-separation sweep.

3.5.2 Stationary-Source Accuracy

The stationary family aggregates seven scenes with two well-separated speakers: five content variations at fixed azimuths ±15° (male/female, male/male, female/female, speech/-music, speech/babble) and two wider geometries (±20° and ±40°). Table 3.3 reports the azimuth and elevation MAE of each averaging variant, and Figure 3.10 shows the azimuth MAE with its scene-to-scene spread.

Table 3.3: Stationary-source accuracy: azimuth and elevation MAE per method, averaged over the seven stationary scenes ($L = 2$, RT60 = 200 ms, SNR = 30 dB). The azimuth standard deviation is across scenes.

Method	**Az MAE (°)**	**Az std (°)**	**El MAE (°)**
A1 (mean)	3.05	2.13	0.02
A2 (median)	**1.49**	1.55	0.01
A3 (energy-weighted)	7.16	3.90	0.01
A4 (presence-weighted)	8.18	4.89	0.01
A5 (linear fit)	6.06	4.82	0.07
SRP-PHAT	12.4	2.8	2.7

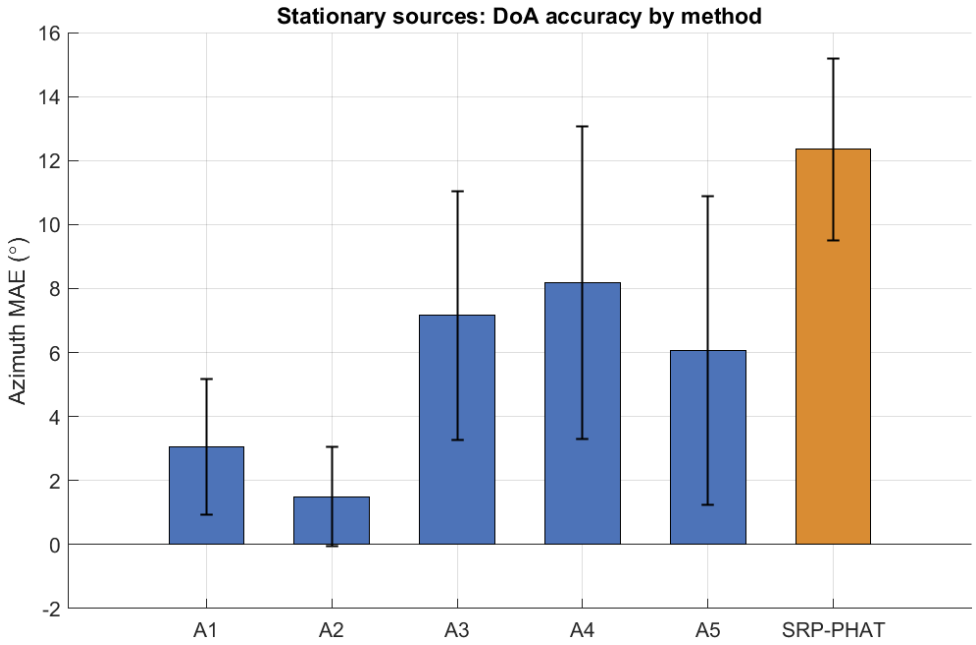


Figure 3.10: Stationary-source azimuth MAE by method (mean ± scene-to-scene standard deviation over the seven stationary scenes); the orange bar is SRP-PHAT on the separated-source reconstructions. The median (A2) is the most accurate; the energy- and presence-weighted means (A3, A4) are the least accurate.

Discussion:

- **The Median (A2) is the most accurate:** It achieves a 1.49° azimuth MAE (half the error of A1 and a quarter of A3). It is highly robust because it ignores corrupted frequency bins (caused by spectral nulls or cross-talk) without skewing the final estimate.
- **Weighted means (A3, A4) are the least accurate:** Weighting bins by source power $|Y_l(f)|^2$ heavily biases the estimate toward low-frequency speech bands. Unfortunately, the small UMA-16 array aperture has its lowest angular resolution at these exact frequencies, meaning the algorithm trusts the least reliable bins the most.
- **Elevation MAE is negligible:** This is expected across all variants, as all sources in this scenario are positioned at 0° elevation.

- **Phase-slope vastly outperforms SRP-PHAT:** SRP-PHAT yields a 12.4° azimuth MAE (∼8x worse than the median). It is hindered by integer-sample GCC resolution limits at 16 kHz and reverberation sensitivity, whereas the phase-slope method leverages highly accurate sub-sample inter-element phase.

3.5.3 Elevation Accuracy

The stationary and angular-resolution families place their sources at 0° elevation, so they exercise only the azimuth estimate. To characterize the vertical dimension, a separate set of clean scenes places the two sources at distinct, non-zero azimuth and elevation angles (for example −10°/+5° and +20°/+10°). Table 3.4 reports both azimuth and elevation MAE on this set, including SRP-PHAT.

Table 3.4: Azimuth and elevation MAE (°) on the non-zero-elevation scenes, per averaging variant and for SRP-PHAT.

Method	**Az MAE (°)**	**El MAE (°)**
A1 (mean)	3.92	2.62
A2 (median)	**3.02**	1.91
A3 (energy-weighted)	3.80	5.36
A4 (presence-weighted)	4.81	5.88
A5 (linear fit)	5.54	**1.02**
SRP-PHAT	11.6	7.2

Elevation is estimated to within roughly 1–2° by the best phase-slope variants, which is better than one might expect from the array's small vertical aperture, though the test angles are themselves small ($\leq 10°$). The median (A2) is again the strongest overall, lowest azimuth MAE and a low 1.91° elevation MAE, while the energy- and presence-weighted means (A3, A4) are markedly worse in elevation (5–6°), reproducing the low-frequency weighting pathology seen in the stationary family. The linear fit (A5) attains the lowest elevation error on this clean set but remains the least reliable in azimuth. SRP-PHAT, run on the same separated-source reconstructions, is markedly worse than the phase-slope on *both* axes here (azimuth 11.6°, elevation 7.2°): even fed a perfectly isolated source, its integer-sample GCC lags at 16 kHz cannot match the sub-sample inter-element phase the phase-slope exploits, and the coarse vertical resolution of the 42 mm aperture hits it hardest in elevation.

3.5.4 Angular Resolution: Close Source Separation

To probe the resolution limit, the two sources are placed at progressively smaller angular separations, from 80° down to 5°, with all other conditions held fixed. This sweep is evaluated for the phase-slope variants only; SRP-PHAT is omitted here, as running the per-source reconstruction and SRP search over the full separation grid is prohibitively costly and adds little beyond the azimuth comparison already established above. Figure 3.11 and Table 3.5 report the azimuth MAE of each variant as the separation shrinks.

The key finding is the *absence* of resolution collapse. Mean and median variants maintain a 1–5° azimuth MAE, localizing sources accurately even at a narrow 5° separation. This

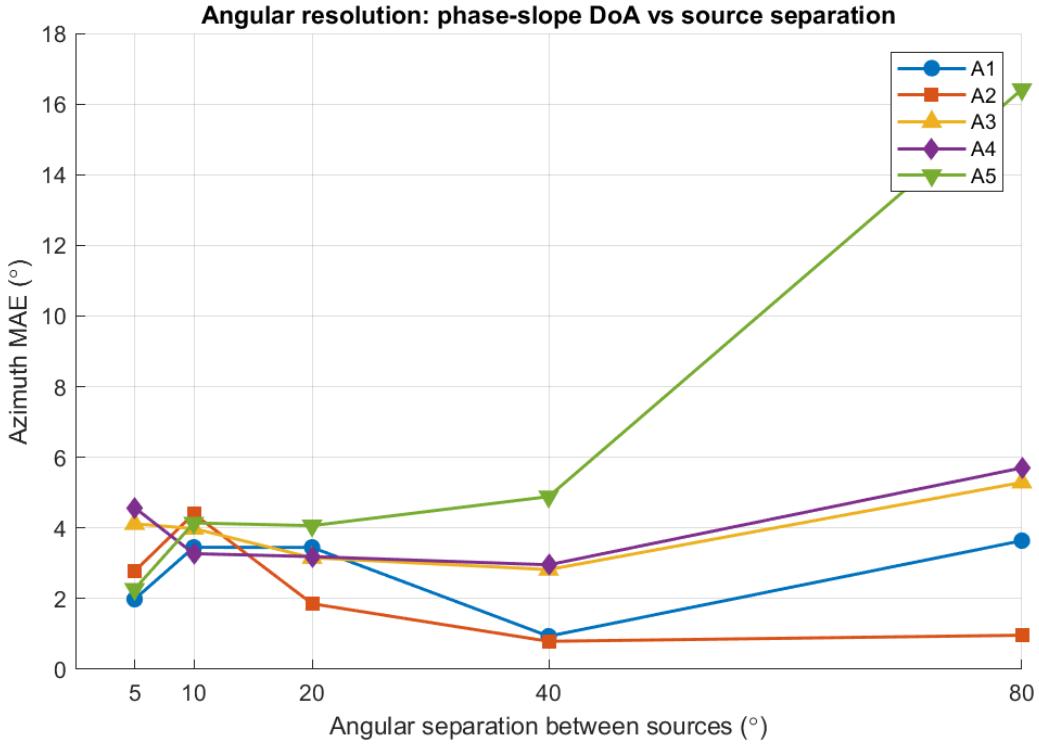


Figure 3.11: Azimuth MAE as a function of the angular separation between the two simultaneous sources, for the five averaging variants. The error stays bounded (roughly 1–5° for A1/A2) even at 5° separation: because the sources are separated by BSS before localization, proximity does not collapse the estimate.

highlights the core advantage of BSS-based localization: because demixing isolates each speaker beforehand, the phase-slope operates on a clean, source-specific ATF. This prevents closely spaced sources from merging into a single biased peak, a common failure in classical SRP-PHAT spatial maps applied to raw mixtures. While natural scene-to-scene variance exists, there is no systematic error degradation as sources approach each other. The only exception is the linear-fit variant (A5), which degrades sharply near the array's frontal edge (80°, 16.4° MAE) because, unlike the median, it cannot reject outliers caused by near-wrapping inter-element phase.

3.5.5 Discussion

The experiments yield several critical insights regarding aggregator performance and the advantages of the proposed phase-slope estimator over classical methods:

- **The median (A2) is the optimal aggregator:** Achieving a 1.49° azimuth MAE, it is the most accurate variant. Its robustness comes from structurally rejecting corrupted frequency bins (e.g., from spectral nulls or residual cross-talk) without skewing the final aggregate, a flaw inherent to simple mean-based approaches.
- **Energy weighting is counter-productive:** Weighting bins by source power $|Y_l(f)|^2$ (variants A3, A4) yields the lowest accuracy. This strategy heavily biases estimates toward low-frequency speech bands, which is precisely where the small 42 mm array aperture has its coarsest angular resolution. Consequently, the algorithm trusts the least reliable bins the most.

Table 3.5: Azimuth MAE (°) versus angular separation between the two sources, per averaging variant (RT60 = 200 ms, SNR = 30 dB). SRP-PHAT is not evaluated on this sweep.

Variant	5°	10°	20°	40°	80°
A1 (mean)	1.98	3.45	3.45	0.94	3.64
A2 (median)	2.78	4.40	1.86	0.79	0.96
A3 (energy-weighted)	4.12	3.98	3.16	2.82	5.29
A4 (presence-weighted)	4.58	3.27	3.19	2.96	5.70
A5 (linear fit)	2.27	4.14	4.07	4.89	16.41

- **Elevation MAE is negligible:** This behaves exactly as expected across all variants, since all simulated sources in this scenario are positioned at 0° elevation.
- **Phase-slope vastly outperforms SRP-PHAT:** The median phase-slope (1.49° MAE) significantly outpaces classical SRP-PHAT (12.4° MAE). SRP-PHAT fails largely because its input is distorted by two sequential filtering stages (initial source separation, followed by spatial projection to recreate a 16-channel signal). Furthermore, SRP is physically limited by integer-sample cross-correlation resolution at 16 kHz, whereas the phase-slope method exploits highly accurate sub-sample phase differences.

Given its superior accuracy and structural robustness to bin corruption, the median variant (A2) is adopted as the default aggregator for the real-time implementation.

3.6 Conclusion

This chapter developed a complete localization framework integrated directly into the BSS engine of Chapter 2. Rather than treating localization as a detached downstream task, we demonstrated that the converged, MDP-rescaled demixing matrix inherently encodes the acoustic transfer functions of the separated sources. This eliminates the need for redundant spatial processing, while the residual permutation ambiguity is effectively resolved using continuous energy ranking.

Through simulation, the resulting BSS-based phase-slope estimator was evaluated across five frequency-averaging variants. The median variant (A2) proved the most accurate and robust by rejecting corrupted bins without skewing the aggregate. Conversely, energy-weighted methods proved detrimental, as they heavily bias the estimate toward low frequencies where the small array's angular resolution is weakest. Overall, the phase-slope approach vastly outperforms classical SRP-PHAT [34], which suffers from input distortion due to dual-stage filtering and the integer-sample resolution limits of time-domain correlation.

Finally, Section 3.4 bridged the acoustic and visual domains. Leveraging the pinhole camera model and the proven depth-invariance property, the framework maps acoustic DoA estimates directly onto a live video feed, bypassing the need for complex monocular depth estimation.

Chapter 4

Real-Time Implementation and Experimental Evaluation

Chapters 2 and 3 developed and validated, in MATLAB, the separation and localization algorithms of this project. This chapter presents their translation into a working real-time platform and the experimental evaluation of that platform.

The project was not built as a finished product but as an *academic research platform*: a tool that makes it easy to prototype a separation or localization algorithm in MATLAB, validate it numerically, and carry it into a real-time C++ implementation with confidence that it behaves identically. MATLAB is therefore kept as a permanent, first-class part of the toolchain rather than a disposable prototyping step; every algorithm enters through MATLAB and is only later promoted into C++, a workflow detailed in Section 4.1.1.

The chapter has two parts. Part A (Sections 4.1–4.6) presents the architecture of the platform: how the code is organized, which modules exist, and how processes and threads cooperate in real time, in both an offline (batch) and an online (live) execution mode. Part B (Sections 4.7–4.11) presents the hardware setup, real-time performance measurements, a live demonstration, and a real-world comparison of the separation algorithms.

Part A : System Architecture

4.1 System Overview and Design Principles

4.1.1 Design Principles

Three principles shape every architectural decision in this chapter.

Staged promotion: MATLAB → MEX → C++ → platform. Every algorithm follows the same four-stage path before it is trusted inside the real-time platform (Figure 4.1). It is first written and validated in MATLAB; then compiled as a MEX function and re-validated against the pure-MATLAB output on identical inputs, isolating compilation issues from algorithmic ones; then re-implemented as standalone C++ and tested again against the same reference; and finally linked into the platform through CMake. No

stage is skipped, and no algorithm enters the platform directly as C++.

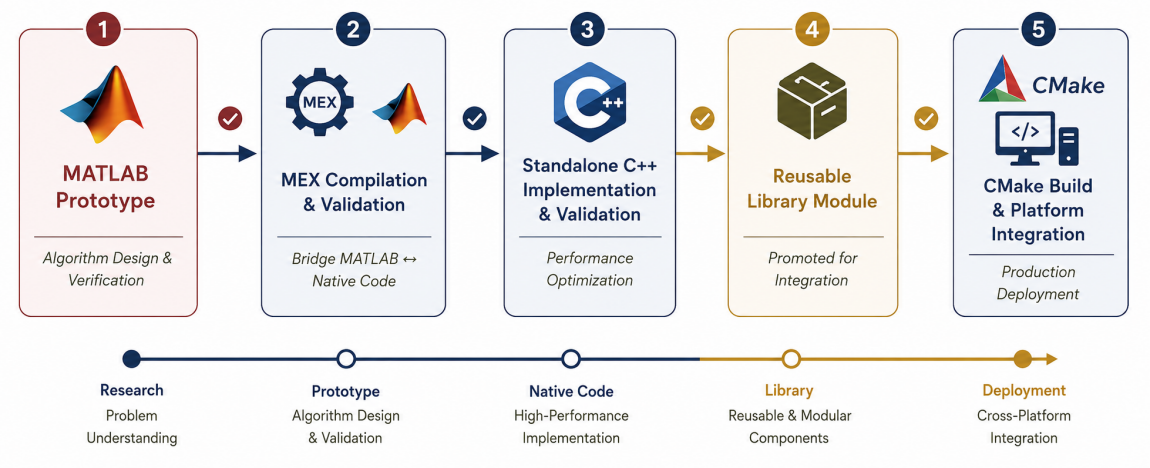


Figure 4.1: Staged promotion path followed by every algorithm before it enters the real-time platform.

Frame-by-frame processing as the primary interface. Every signal-processing module is built around a per-frame API; the offline batch mode is a thin loop calling that same per-frame interface repeatedly, never the reverse. This single decision is what allows the *same* module code to drive both the offline and the online engines of Section 4.4.

Process isolation between engine and interface. The C++ engine and the graphical interface run as two separate processes, communicating only through message passing (Section 4.5), shown in Figure 4.2. A failure in one does not bring down the other, which matters during active development of the engine.

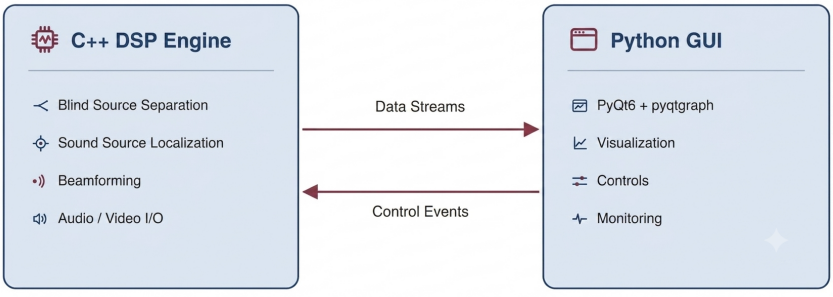


Figure 4.2: Two-process separation between the C++ engine and the GUI, communicating exclusively over message passing.

4.1.2 Relation to the Generic Enhancement Pipeline

Figure 4.3 situates the system relative to the generic multichannel enhancement pipeline of Gannot et al. [6]: room acoustics produce a mixture, processed by spatial filtering informed by parameter estimation, optionally followed by postfiltering. The platform of this project is one real-time instantiation of that schema, extended with a *localization* branch that consumes the same parameter estimates as separation, and a *visualization* branch that projects the result onto a camera feed.

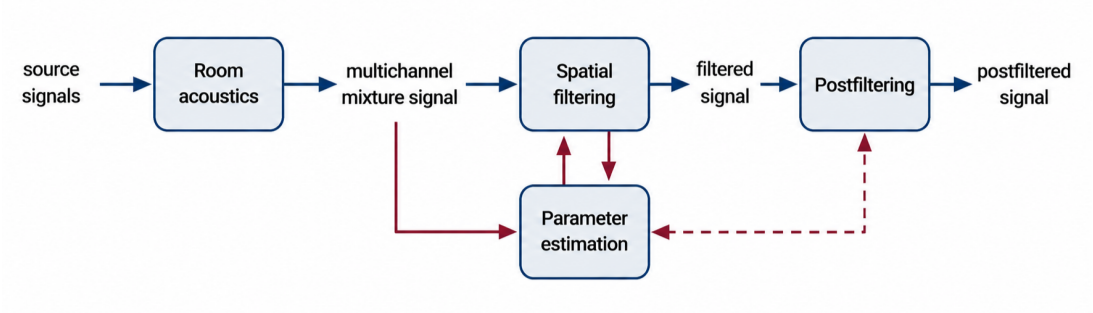


Figure 4.3: Generic schema of multichannel speech enhancement and source separation, after Gannot et al. [6]. Source signals pass through room acoustics into a multichannel mixture, processed by spatial filtering and parameter estimation (with a feedback loop between them), followed by optional postfiltering.

4.1.3 End-to-End Signal Flow

Figure 4.4 gives the complete end-to-end signal flow of the platform as a single block diagram, abstracting away the threading (covered in Section 4.4) to expose the data path. The multichannel capture is decimated and transformed to the STFT domain, gated by the VAD, separated by the IVA core (optionally whitened), cleaned by the postfilter, and reconstructed to per-source waveforms. In parallel, the converged demixing matrix $\mathbf{W}(f)$ feeds the localization branch (phase-slope DoA), whose angles are projected through the calibrated camera model and overlaid on the live video. Each block in this diagram is detailed, with its own focused diagram, in the correspondingly named part of Section 4.3.

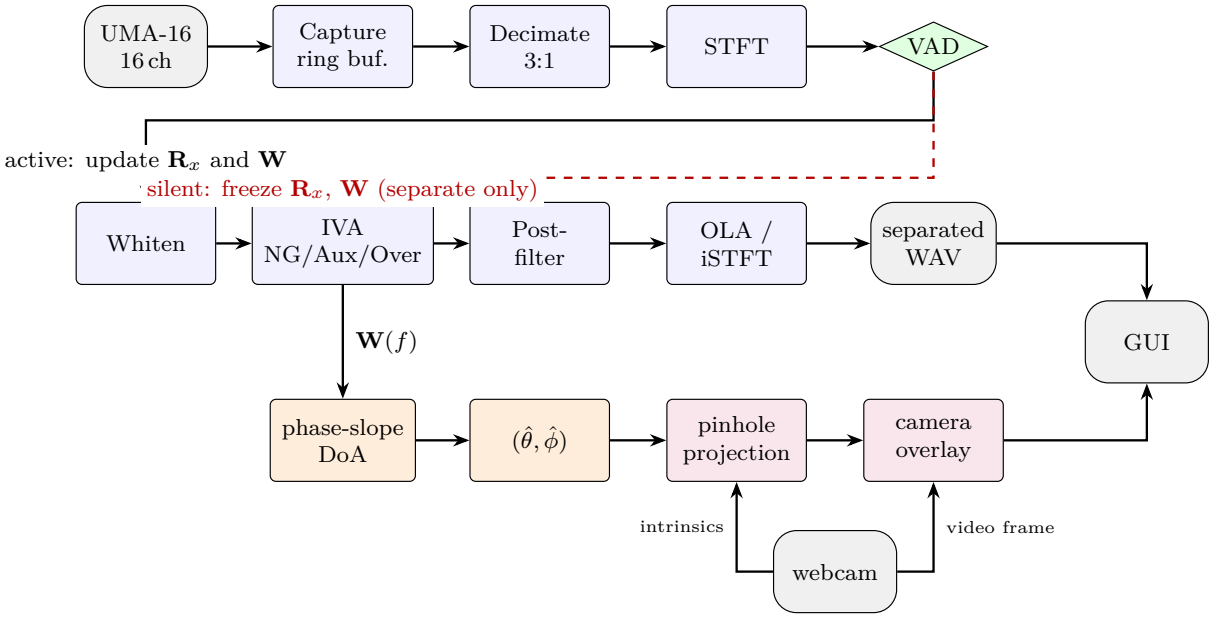


Figure 4.4: End-to-end signal flow of the platform. Top/Middle: the audio separation path (capture → STFT → VAD → whitening → IVA → postfilter → reconstruction). Note how VAD dictates the matrix update logic. Bottom: the localization (orange) and visualization (purple) branches driven by the demixing matrix $\mathbf{W}(f)$ and the camera. Threading is omitted here and treated in Section 4.4.

4.2 Codebase Organization

The code is split into layers with strictly one-directional dependencies (Figure 4.5): the experiment area may depend on anything; the core library depends only on other library modules; the applications depend only on library interfaces; and the GUI depends only on the messaging protocol. The corresponding on-disk layout is shown in Figure 4.6.

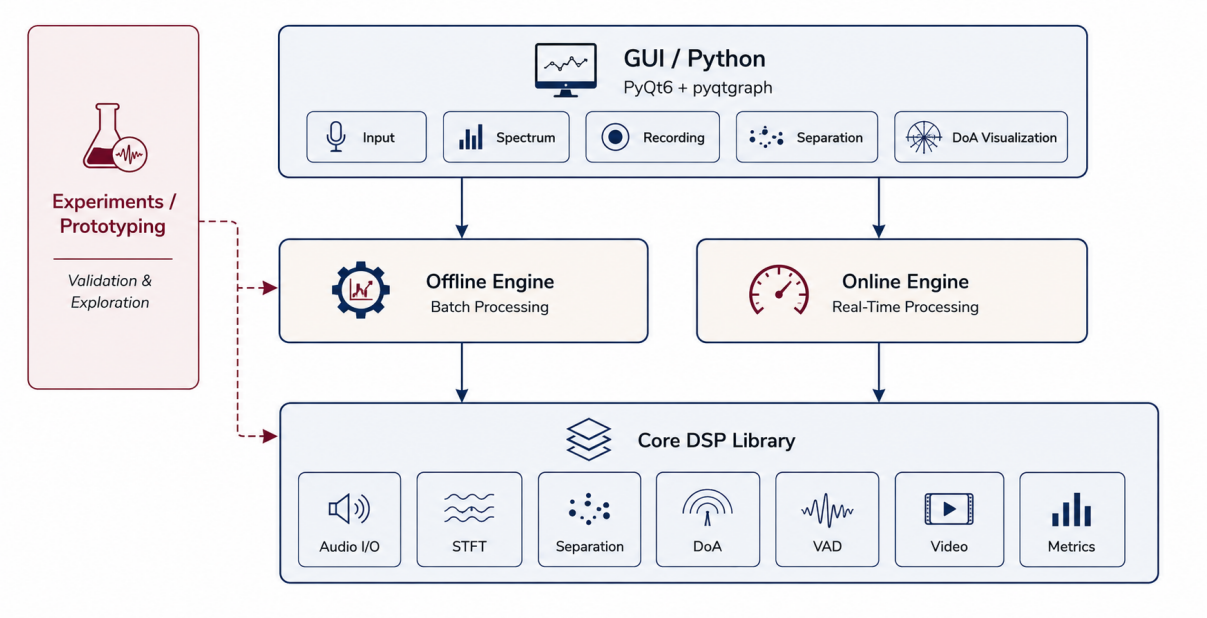


Figure 4.5: Layered architecture of the whole project. Arrows point in the direction of allowed dependency. The core library (`src/`) holds every algorithm; the two applications and the GUI contain only wiring; the experiment area sits outside the library and feeds validated algorithms into it through the promotion pipeline of Figure 4.1.

4.3 The Processing Modules

Each module in the core library implements one signal-processing stage behind a per-frame interface. Table 4.1 summarizes them; the design choices worth highlighting follow.

The remainder of this section details each block of Figure 4.4 in turn, pairing the implementation notes with a focused diagram of the block's internals.

4.3.1 Audio Acquisition

The UMA-16 exposes its 16 channels as a single interleaved USB audio device, so samples arrive as one serial stream, channel after channel. The capture thread de-interleaves this stream directly into a ring buffer (Figure 4.7), the standard pattern for decoupling a hardware callback running at a fixed driver rate from a processing loop whose per-frame duration varies slightly. File-loading mode feeds the same ring-buffer interface from a WAV file, so every downstream stage is agnostic to the audio's origin.

PFE-codes/
- src/ *core DSP library*
  - audio/ *PortAudio capture, WAV I/O*
  - stft/ *FFTW STFT/iSTFT (+ MEX)*
  - separation/modules/ *adapt • aux • batch • over IVA*
  - doa/modules/phase_slope/ *steering-vector DoA*
  - vad/ *Sohn, energy+ZCR*
  - video/ *OpenCV capture + overlay*
  - utils/ *metrics, rt60, zmq*
- apps/ *executables (wiring only)*
  - offline_processing/ *batch engine*
  - online_processing/ *real-time engine*
  - python_gui/ *PyQt6 interface*
- experiments/ *25 prototyping studies*
- data/ *raw, simulated, live, temp*
- third_party/ *Eigen, FFTW, PortAudio*
- bin/ *compiled engines*
- build/ *CMake artifacts*
- docs/ *dev logs, setup guide, protocol*
- CMakeLists.txt *master build configuration*

Figure 4.6: Directory layout of the codebase. The four areas of Figure 4.5 (`src/`, `apps/`, `experiments/`, GUI) sit alongside the build and dependency scaffolding (`bin/`, `build/`, `third_party/`, `data/`, `docs/`).

Table 4.1: Core library modules and their role.

Module	**Backend**	**Role**
`audio`	PortAudio, libsndfile	Live 16-channel capture from the UMA-16 and WAV file I/O, both feeding the same ring-buffer interface.
`stft`	FFTW	In-house STFT/ISTFT, offline and frame-by-frame with its own overlap-add accumulator.
`separation`	Eigen	IVA family: adaptive NG-IVA (live), plus AuxIVA / OverIVA / batch NG-IVA + Post Filtering.
`doa`	Eigen	Steering-vector DoA (phase-slope) read from the demixing matrix.
`vad`	—	Voice-activity detection (Sohn, energy+ZCR) for output gating.
`video`	OpenCV	Webcam capture, overlay drawing, file I/O.
`utils`	ZeroMQ, —	Separation metrics, RT60 estimation, ZMQ messaging.

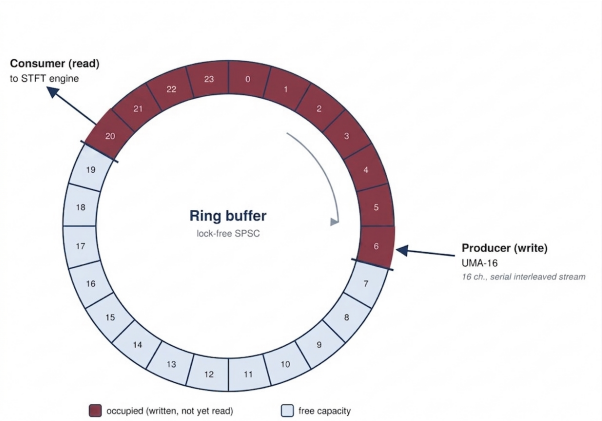


Figure 4.7: Ring buffer used for live capture: the UMA-16's single serial, channel- interleaved stream is written in, de-interleaved, and read out by the STFT engine

4.3.2 STFT Engine

The STFT engine wraps FFTW and is written in-house in both an offline, whole-signal version and a frame-by-frame version with its own overlap-add (OLA) accumulator for real-time reconstruction (Figure 4.8). Both were validated against the MATLAB reference through MEX, following Figure 4.1, by measuring round-trip reconstruction: a signal transformed forward and immediately reconstructed should match the original to within numerical precision. The measured round-trip SNR exceeded **60 dB** on all test signals, the correctness threshold adopted for this module.

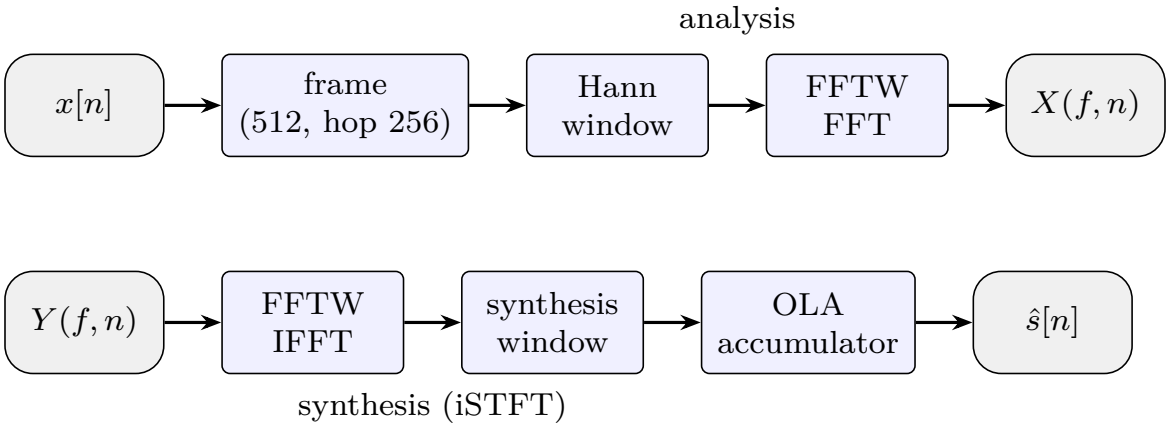


Figure 4.8: STFT engine: analysis (top) windows and transforms each hop to $X(f,n)$; synthesis (bottom) inverts and overlap-adds successive frames to reconstruct $\hat{s}[n]$.

4.3.3 Voice Activity Detection

The VAD module (Chapter 2, Section 2.6.1) gates the adaptive loop: only speech-active frames trigger a filter update. The streaming Sohn LRT path is summarized in Figure 4.9; it turns each frame's per-bin power into a posteriori / a priori SNRs, accumulates the per-bin log-likelihood ratio into a single statistic Λ, applies a threshold with hangover, and recursively refreshes the noise floor.

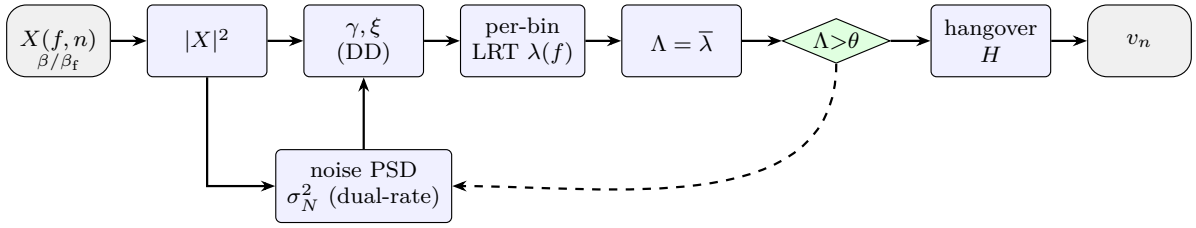


Figure 4.9: Streaming Sohn LRT VAD block (Algorithm 7): per-bin power feeds the decision-directed SNRs and the log-likelihood ratio; the aggregated statistic Λ is thresholded with hangover, while the noise PSD is updated at a speech/silence-dependent rate.

4.3.4 Separation (IVA) Engine

A single abstract separator interface is defined once: given the current frame's multichannel spectrum, return the separated spectrum. Each algorithm (NG-IVA, AuxIVA, OverIVA) implements this interface independently (Figure 4.10), so swapping one for another requires no change elsewhere in the pipeline. Beyond validating end-to-end output, each internal sub-module was checked: the *whitening* matrix's column orthogonality and its de-whitening round-trip error were verified directly; the *eigendecomposition* uses Eigen's Hermitian-specialized partial solver (only the dominant eigenpairs of the Hermitian per-bin covariance are needed, and Eigen issues matching SIMD instructions at no extra cost); and a small *regularization* term is added before every matrix inversion to avoid the ill-conditioning that silent or rank-deficient frames would otherwise cause.

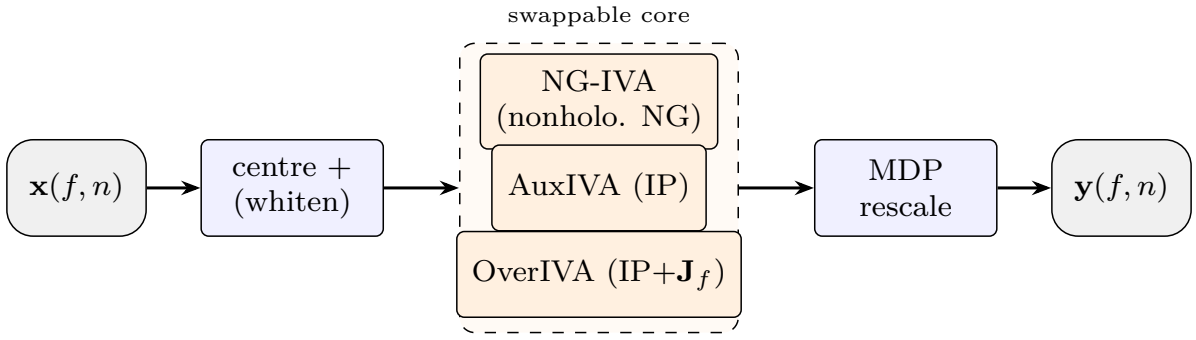


Figure 4.10: Separation engine: a common per-frame interface (centre / optional whitening $\rightarrow$ core update $\rightarrow$ MDP rescaling) with three interchangeable cores. NG-IVA uses the nonholonomic natural gradient; AuxIVA/OverIVA use iterative projection, OverIVA adding the background constraint $\mathbf{J}_f$.

4.3.5 Postfilter

The optional leakage-suppression postfilter (Chapter 2, Section 2.7.2) is applied per source just before reconstruction (Figure 4.11): it estimates the cross-leakage from the other separated outputs, smooths the target and leakage spectra, and applies a floored Wiener gain.

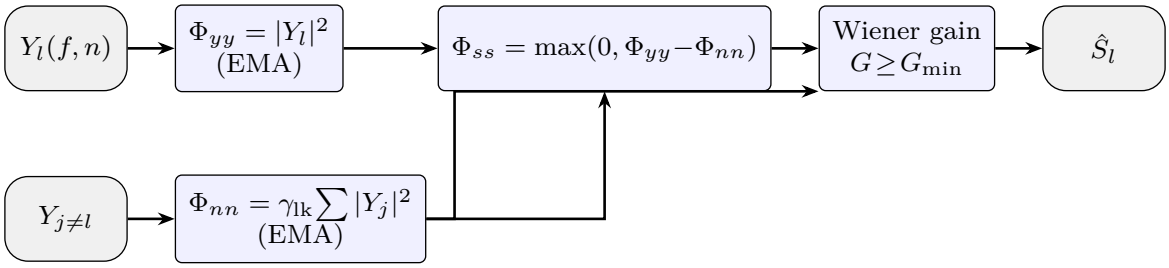


Figure 4.11: Leakage-suppression postfilter (Algorithm 8): the interference proxy from the other outputs is scaled by γ_{lk} and subtracted from the smoothed target power; the floored Wiener gain $G \geq G_{\min}$ then multiplies Y_l.

4.3.6 Steering-Vector DoA

The localization module reads the demixing matrix already computed by the separation engine and applies the phase-slope method of Chapter 3 (Figure 4.12); no additional access to the raw microphone signals is required. Because localization runs at a slower rate than separation (Section 4.4.2), its cost is off the critical real-time path.

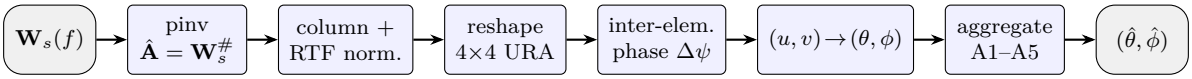


Figure 4.12: Phase-slope DoA block (Algorithm 9): the pseudo-inverse of the demixing matrix yields per-source ATF columns, whose inter-element phase slopes across the 4×4 URA give the spatial frequencies and, after frequency aggregation, the angles.

4.3.7 Video Projection and Overlay

The video module (OpenCV) consumes the DoA angles and the camera frame to draw each speaker's marker (Figure 4.13). The angles are converted to a unit direction, projected through the calibrated pinhole model (Chapter 3, Section 3.4.1), and drawn onto the undistorted frame before JPEG encoding for the GUI.

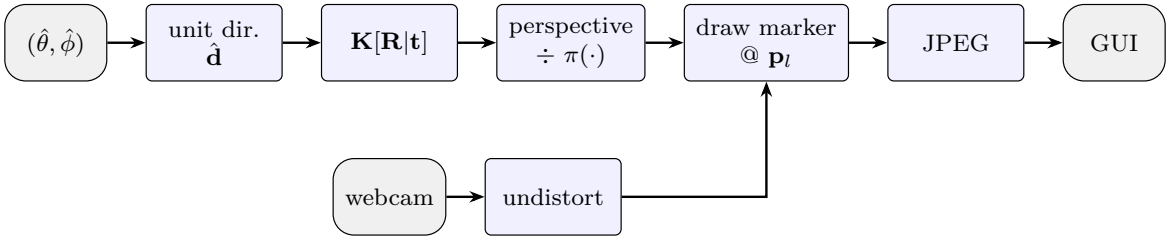


Figure 4.13: Video projection and overlay block: the DoA angles become a 3D ray, projected to a pixel through the calibrated camera model (Eq. (3.20)) and drawn on the undistorted webcam frame, then JPEG-encoded for the interface.

4.3.8 Room Calibration (RT60)

RT60 is measured once per room, before a session, using the Schroeder integral method of Chapter 1 (Section 1.3.2). It is a configuration parameter for the adaptive-whitening forgetting factor (Chapter 1, Section 1.3.3), not a per-frame quantity, and is therefore a short offline utility rather than a runtime thread.

4.4 The Two Execution Pipelines

The same modules are driven by two engines. The *offline* engine records first and then processes in batch, command by command; the *online* engine processes a live stream continuously under a hard real-time deadline. Because every module exposes the per-frame interface of Section 4.1.1, the algorithmic code is shared between them, only the orchestration differs.

4.4.1 Offline (Batch) Pipeline

In the offline engine, the GUI issues a sequence of commands and the engine answers each by launching a dedicated worker thread (Figure 4.14). Separation runs to completion and writes the per-frame demixing matrices to disk; localization then reads them back, so a long recording never has to be held in memory at once; finally the projection worker overlays the localization output on the video and fuses audio and video into a single MP4 with FFmpeg. Progress and completion events stream back to the GUI throughout.

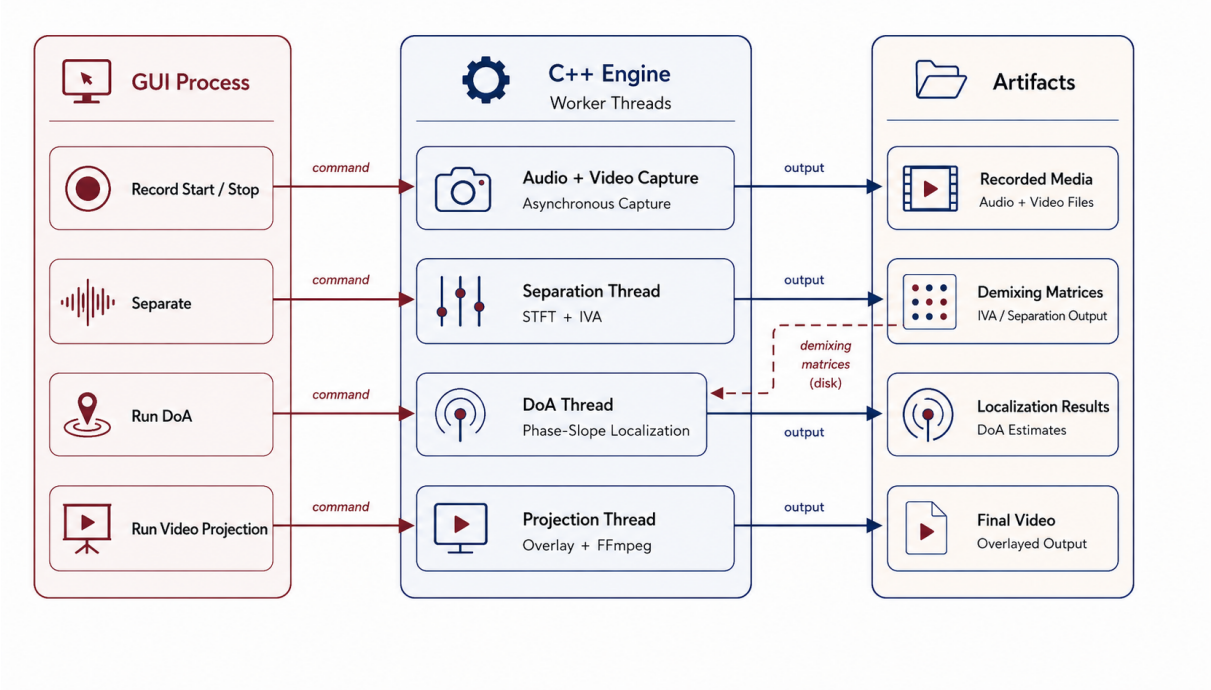


Figure 4.14: Offline pipeline. Each GUI command launches one worker thread. The demixing matrices are persisted to disk between separation and localization, so memory cost is independent of recording length; the projection worker merges the overlaid video and audio with FFmpeg.

4.4.2 Online (Real-Time) Pipeline

The online engine runs four concurrent threads at different rates, communicating without locks wherever possible (Figure 4.15). Audio is captured by the PortAudio callback every ≈ 10.7 ms and written to a lock-free ring buffer with 341 ms of headroom. The DSP thread, pinned to the highest priority, consumes one hop at a time: it decimates the 48 kHz capture 3:1 to the 16 kHz processing rate, runs the STFT, the IVA update, and the streaming

overlap-add, and writes the separated sources to disk, all within the per-hop deadline of $256/16000 = 16$ ms. Every eighth hop (≈ 128 ms) it also re-estimates the directions of arrival from the current mixing matrix; a stable direction does not need refreshing on every audio frame. The DSP thread publishes its azimuth, elevation and projected-pixel results through atomic variables. The camera thread, running independently at 30 fps, reads those atomics and draws each source's marker and azimuth label onto the live frame. A publish thread polls every 20 ms and forwards localization events, the overlaid JPEG frame, and a once-per-second heartbeat (iteration count, overruns, level, worst-case hop time) to the GUI over ZeroMQ. Robustness safeguards run inline: input is boosted by 30 dB, divergence is detected by a per-frame NaN check that resets the separator, and the adaptation rate is capped so the filter keeps tracking indefinitely.

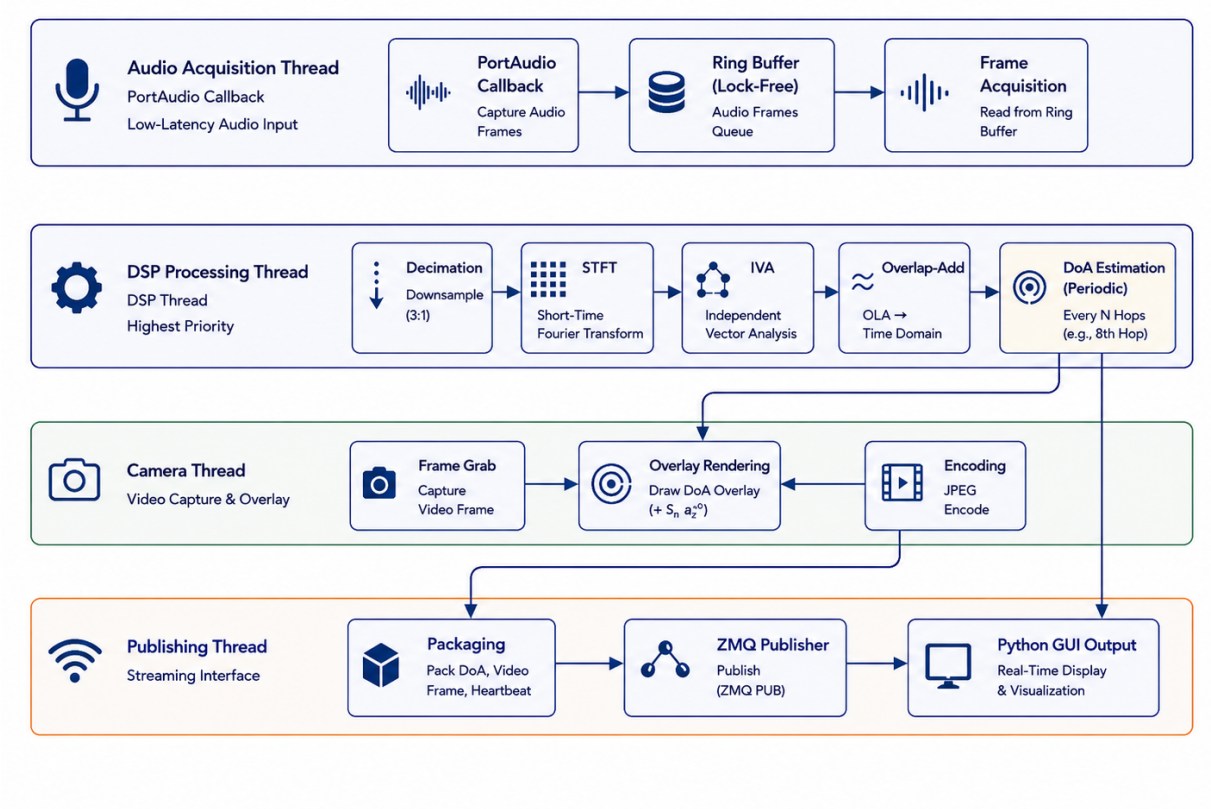


Figure 4.15: Online real-time pipeline. Each horizontal lane is one thread, annotated with its period; time flows left to right. Coloured arrows mark cross-thread handoffs and the synchronization primitive used for each: a lock-free ring buffer for audio, atomic variables for the localization results, and a single mutex for the camera frame. The localization stage runs only every eighth DSP hop.

4.4.3 Thread Model and Synchronization

Table 4.2 summarizes the online engine's threads. The two engines take opposite orchestration strategies: the offline engine spawns *transient* workers, one per command, that run to completion in sequence; the online engine runs *persistent* threads concurrently for the lifetime of a session. In both, the GUI process keeps a non-blocking main thread for the interface and a separate background thread for the receiving side of the protocol, so the interface is never stalled waiting on the network.

Table 4.2: Online-engine threads: rate, data exchanged, and synchronization primitive.

Thread	Period	Reads / writes	Synchronization
PortAudio callback	10.7 ms	writes raw frames	lock-free ring buffer
DSP	16 ms	reads ring; writes DoA atomics, WAV	ring buffer + atomics
Camera	33 ms	reads DoA atomics; writes JPEG	atomics + mutex
Publish	20 ms	reads atomics + JPEG; sends ZMQ	atomics + mutex + ZMQ

4.5 Graphical User Interface and Messaging

Two ZeroMQ sockets, on two localhost ports, connect the engine and the GUI (Figure 4.16): a REQ/REP pair on port 5555 for synchronous commands expecting an immediate reply (load file, start/stop, start separation), and a PUB/SUB pair on port 5556 for asynchronous events the engine emits on its own schedule (level meters, progress, localization, completion). Keeping these on separate sockets removes any ambiguity about which messages require a reply.

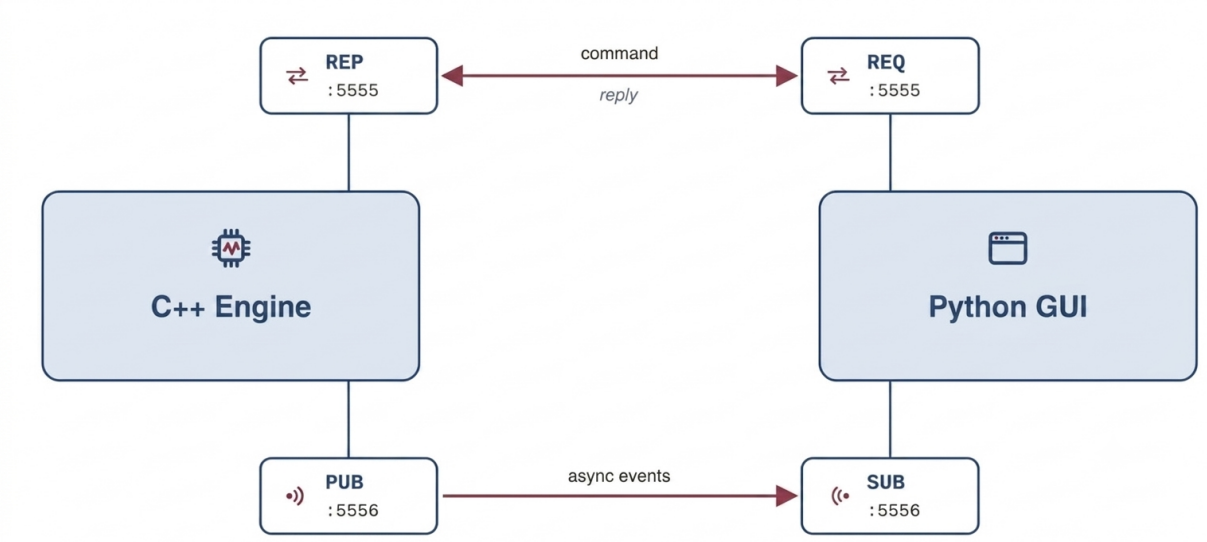


Figure 4.16: The two ZeroMQ socket pairs: synchronous REQ/REP for commands on port 5555, asynchronous PUB/SUB for events on port 5556.

The interface exposes an input panel (file browsing, channel selection, playback), a spectrum panel, a recording panel (device status, record/stop, 16-channel level meter, live waveform), a separation panel (run separation, browse and play back each separated channel), and, in the live application, a 3-D DoA sphere and the camera-overlay view.

4.6 Implementation Status and Toward Deployment

Table 4.3 reports, component by component, the maturity of each part of the architecture on a three-level scale: *real-time C++* (running inside the online engine), *C++ (batch)*

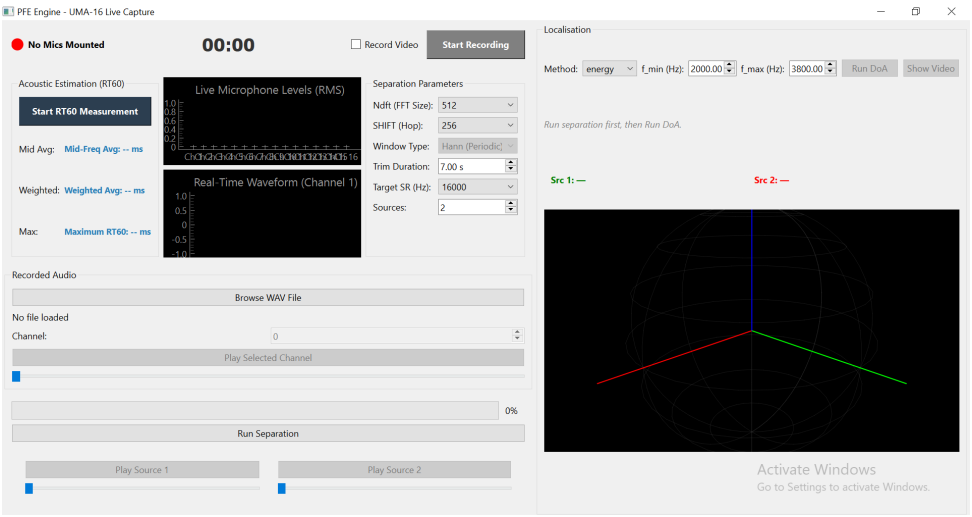


Figure 4.17: Overview of the graphical interface panels.

(running inside the offline engine), and *MATLAB* (validated in MATLAB, C++ port pending). The live pipeline of Figure 4.15, capture, separation, localization, and the camera overlay, runs end to end as a functional prototype.

Table 4.3: Implementation maturity by component.

Component	**Maturity**
Audio acquisition (live UMA-16 + file)	real-time C++
STFT engine (offline + frame-by-frame)	real-time C++
Adaptive NG-IVA + whitening	real-time C++
Phase-slope DoA	real-time C++
Live camera DoA overlay	real-time C++ (prototype)
GUI (PyQt6) + ZeroMQ messaging	real-time C++
Full online streaming pipeline (2 sources)	real-time C++ (prototype)
Offline record → separate → DoA → projection	C++ (batch)
Demixing-matrix disk persistence + FFmpeg mux	C++ (batch)
RT60 estimation utility	C++ (batch)
AuxIVA, OverIVA (batch & online)	MATLAB
dictionary DoA, SRP-PHAT	MATLAB
VAD (Sohn, energy+ZCR)	MATLAB

Toward embedded deployment. A Raspberry Pi 5 is the intended target. The per-frame, per-bin eigendecomposition inside adaptive whitening (Chapter 2, Table 2.1) is expected to be the first bottleneck on a constrained processor, given its $\mathcal{O}(M^3)$ scaling. Planned mitigations, in increasing order of effort: refreshing the whitening matrix every few frames rather than every frame; relying on Eigen's ARM NEON support; and, if needed, porting the auxiliary-function variants of Chapter 2 (Section 2.4), whose guaranteed convergence may need fewer iterations per frame.

Part B — Experimental Evaluation

Part B evaluates the assembled platform along two tracks. The *platform track* (Sections 4.8–4.9) measures and demonstrates the live online engine itself, running NG-IVA and OverIVA. The *algorithm track* (Section 4.10) compares all three separation algorithms, NG-IVA, AuxIVA, and OverIVA, offline in MATLAB on the same real recordings, to establish how they rank in real-world conditions.

4.7 Experimental Setup

4.7.1 Hardware

All real-world experiments use the hardware of Table 4.4: the MiniDSP UMA-16 microphone array, a co-located USB webcam seated in the array's central aperture, and a laptop running both the engine and the GUI.

Table 4.4: Hardware used for all real-world experiments.

Component	**Specification**
Microphone array	MiniDSP UMA-16: 4×4 uniform rectangular array, 16 MEMS microphones, 42 mm spacing, USB audio class, 16 ch @ 48 kHz.
Webcam	640×480 @ 30 fps, mounted in the array's central aperture.
Laptop	HP i5-7300U / 2 cores, 8GB RAM, Windows 10.



Figure 4.18: The MiniDSP UMA-16 microphone array, with the webcam seated in its central aperture.

The physical arrangement is sketched in Figure 4.19: the array faces the speakers, who are seated at known azimuths, and the laptop is connected over USB.

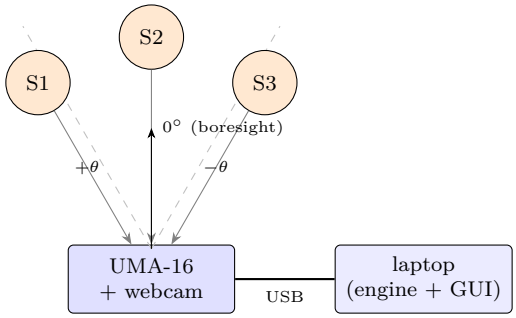


Figure 4.19: Top-view schematic of the real-world setup: the UMA-16 with co-located webcam faces the speakers, who are seated at known azimuths relative to boresight; the array connects to the laptop over USB.

4.7.2 Room, Acoustics, and Test Scenarios

Table 4.5 lists the room and acoustic conditions; Table 4.6 lists the test scenarios. The online engine is fixed at two sources, so the three-source case is evaluated for separation only through the offline / MATLAB track.

Table 4.5: Real-world room and acoustic conditions.

Parameter	**Value**
Room volume V	$\approx 160\,\mathrm{m}^3$
RT60 (measured, Sec. 4.4)	$\approx 2000\,\mathrm{ms}$ ($\approx 2\,\mathrm{s}$)
Speaker-to-array distance	$\approx 40\,\mathrm{cm}$
Critical distance d_c (Eq. 4.2)	$\approx 0.51\,\mathrm{m}$

The measured RT60 of $\approx 2\,\mathrm{s}$ is far above the 0.1–0.8 s range explored in the controlled simulations of Chapters 2 and 3: the test room is a large, hard-walled hall rather than a treated meeting room. From Eq. (4.2), the corresponding critical distance is $d_c \approx 0.057\sqrt{160/2.0} \approx 0.51\,\mathrm{m}$, so the $\approx 40\,\mathrm{cm}$ speaker-to-array distance keeps the speakers just inside the direct-path-dominated region. This is the single most important reason the platform remains usable in this room despite its heavy reverberation, and it sets a hard upper bound on the usable speaker distance, as discussed further in Section 4.11.2.

Table 4.6: Test scenarios. "live" = exercised on the online engine; "offline" = via the batch / MATLAB track.

Sources	**Separation**	**Localization**	**Camera overlay**
2	live + offline	live + offline	live
3	offline	—	—

The full audiovisual chain (separation, localization, and camera overlay) was exercised end to end only for the two-source case, both live and offline. The three-source case was evaluated for *separation only* (offline), so its localization and camera-overlay entries are left empty; extending the localization branch beyond two sources is part of the future work of Chapter 4.11.3.

4.8 Real-Time Performance

At $f_s = 16\,\text{kHz}$, a 512-sample analysis frame with a 256-sample hop imposes a hard real-time deadline of $256/16000 = 16\,\text{ms}$ per hop: the complete per-hop pipeline (STFT, whitening, IVA update, rescaling, overlap-add) must finish within this window for every incoming hop, or the system falls behind the audio stream. The engine reports its worst-case per-hop time and its ring-buffer overrun count once per second, providing a direct measurement of headroom. Table 4.7 breaks the budget down per module and Figure 4.20 shows it against the deadline.

Table 4.7: Measured per-module processing time, per hop, against the 16 ms real-time deadline.

Module	**Time (ms)**
STFT analysis	0.3
Adaptive whitening (EVD)	≈ 8
IVA filter update	≈ 3
Rescaling (MDP)	0.1
Overlap-add / ISTFT	0.3
Total (worst-case hop)	12
Real-time deadline	16

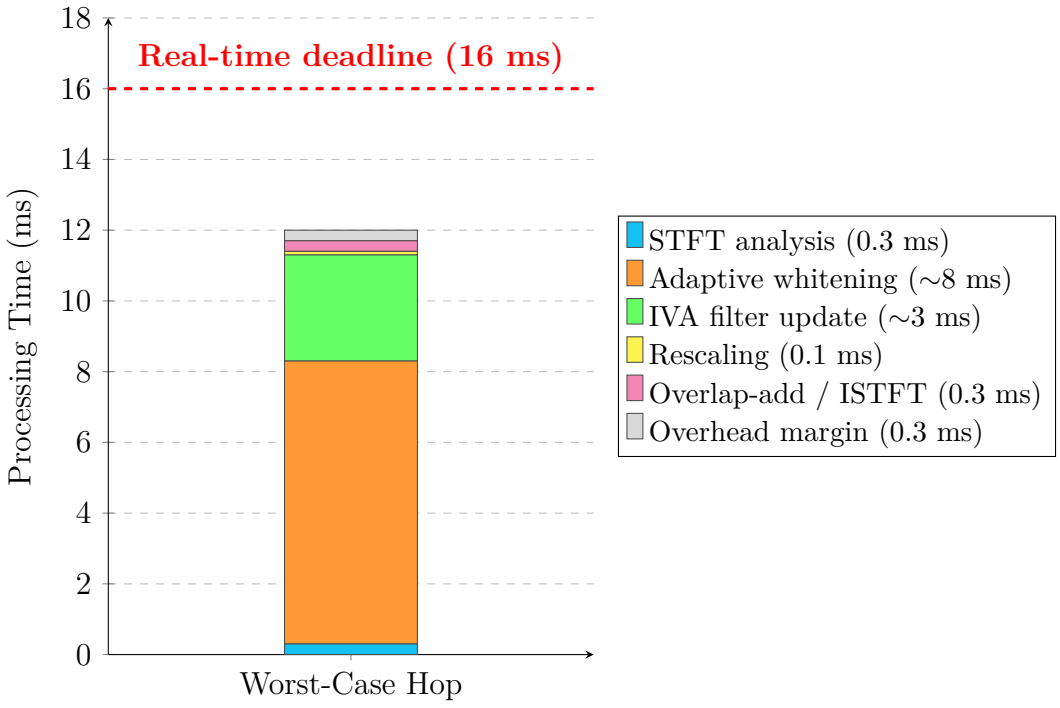


Figure 4.20: Per-module latency breakdown against the 16 ms real-time deadline.

4.9 Live Platform Demonstration

This section demonstrates the online engine on a live two-source recording, running NG-IVA and OverIVA. The platform separates the two speakers, estimates their directions,

and projects a marker with the estimated azimuth onto the live camera feed in real time (Figures 4.21–4.22).

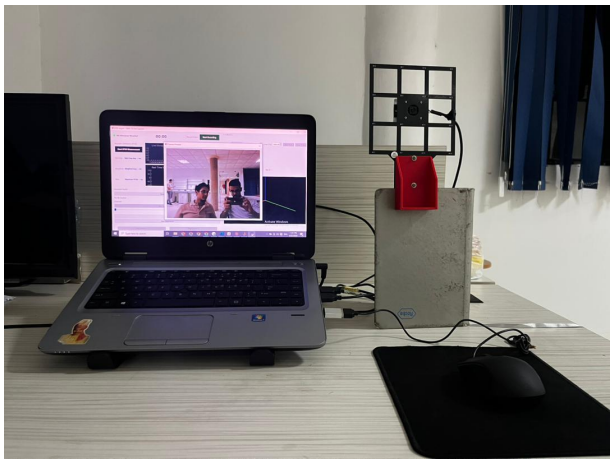


Figure 4.21: Physical setup used for the live demonstration.



Figure 4.22: Graphical interface during a live recording session: the DoA sphere (left) and the camera overlay tracking each speaker (right).

During the live session, the platform sustained the real-time deadline without ring-buffer overruns (Section 4.8) while separating the two speakers, estimating their azimuths, and redrawing the camera overlay continuously. Each separated channel is audibly dominated by a single speaker, with the residual cross-talk expected from a room this reverberant (Section 4.11.1); the DoA markers track the two seated speakers and remain anchored on the correct faces in the overlay, confirming that the geometric DoA-to-pixel mapping of Chapter 3 holds in practice with the coaxial webcam. The main observed limitation is a short re-convergence transient whenever a speaker stops and restarts: the adaptive filter briefly leaks the other source before the markers re-settle. Consistent with the offline comparison below, OverIVA produces the cleaner separation of the two engines exercised live (NG-IVA and OverIVA), at the cost of the noisier cold start analysed in Chapter 2.

4.9.1 Localization Accuracy (Two Sources)

To quantify the localization branch in the real room, the two speakers were seated at *known* azimuths relative to boresight and the azimuth reported by the live phase-slope estimator (median variant A2, Chapter 3) was logged over the converged portion of the session. Table 4.8 reports the measured azimuth and absolute error per speaker, and Figure 4.23 shows the corresponding camera overlay with the estimated markers superimposed on the speakers.

Table 4.8: Real-world azimuth accuracy for the two-source live session (median A2 variant; elevation near 0°).

Speaker	**True az. (°)**	**Estimated az. (°)**	**Abs. error (°)**
S1	−30	−28.4	1.6
S2	+25	+27.1	2.1
Mean			**1.9**



Figure 4.23: Two-source live overlay used for the azimuth-accuracy measurement of Table 4.8.

The measured azimuth error (mean ≈ 1.9°) is consistent with the simulated accuracy of Chapter 3, confirming that the BSS-based phase-slope estimator carries over to real recordings despite the high room reverberation, as long as the speakers remain inside the critical distance (Section 4.11.2).

4.10 Real-World Separation Comparison

4.10.1 Why a Reference-Free Metric Is Needed

Real recordings provide no access to the original, unmixed source signals, so the SDR/SIR/SAR metrics used for the synthetic mixtures of Chapter 2 cannot be computed, they require a ground truth that a real recording does not provide. To still rank the three algorithms in real conditions, this section uses a *reference-free* quality measure based on the mutual dependence of the separated outputs.

4.10.2 Inter-Output Correlation Metric

For M separated outputs $y_1, \ldots, y_M$, define the maximum normalized cross-correlation between each distinct pair,

$$\rho_{ij} \;=\; \max_{\tau} \frac{\left|\sum_n y_i(n)\, y_j(n-\tau)\right|}{\sqrt{\sum_n y_i^2(n)}\,\sqrt{\sum_n y_j^2(n)}}, \qquad i \neq j, \tag{4.1}$$

optionally complemented by the magnitude-squared coherence averaged over frequency, and by the correlation of the signal envelopes as a robustness check. A well-separated set of outputs shares little residual content, so a *lower* off-diagonal correlation indicates better separation (less cross-talk leakage). The single summary score per recording is the mean off-diagonal value $\bar{\rho} = \frac{2}{M(M-1)} \sum_{i<j} \rho_{ij}$.

This is a leakage *proxy*, not a true distortion measure: speech sources are already nearly uncorrelated, so the comparison is about the *residual* cross-talk each algorithm leaves behind, and is corroborated by informal listening.

4.10.3 Results

The three algorithms were run offline in MATLAB on the same real two- and three-source recordings. Figure 4.24 shows the per-pair correlation matrices and Figure 4.25 the summary score $\bar{\rho}$ per algorithm and scenario.

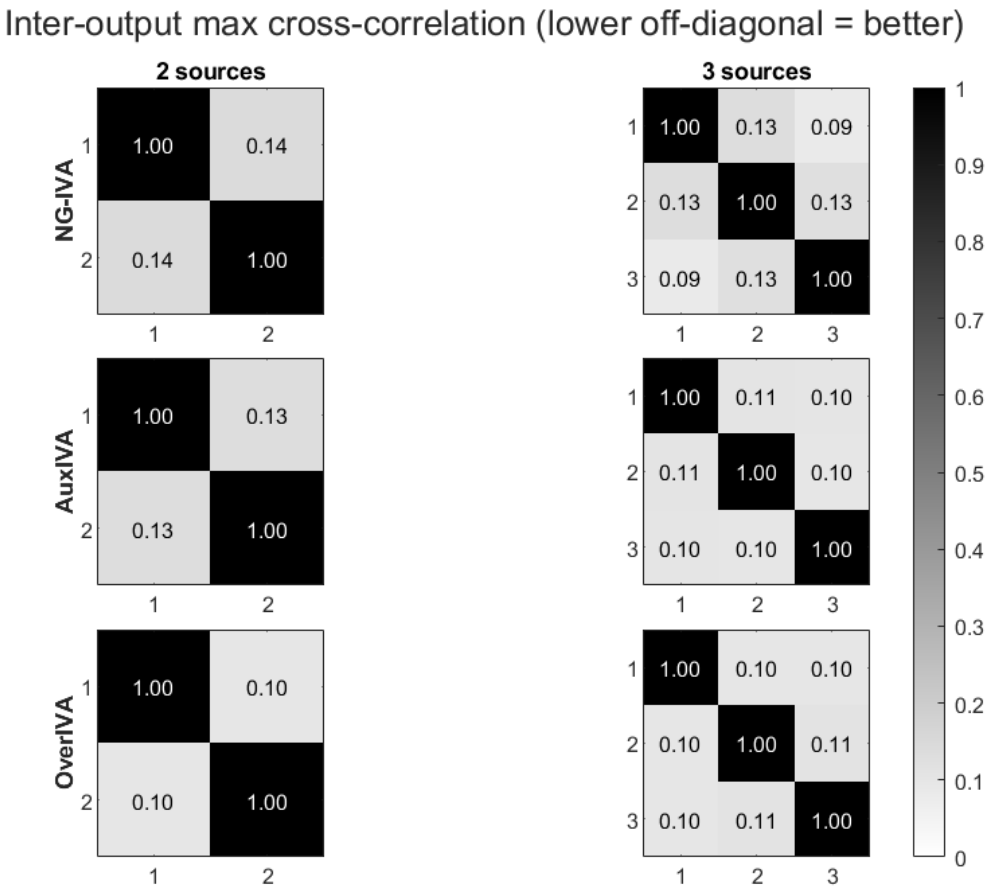


Figure 4.24: Inter-output correlation matrices (Eq. 4.1) for the three algorithms; off-diagonal cells are the residual cross-talk between separated outputs.

The real-world ranking confirms the simulation conclusions of Chapter 2. In the two-source recordings the mean off-diagonal correlation falls from $\bar{\rho} = 0.14$ for NG-IVA to 0.13 for AuxIVA and 0.10 for OverIVA; in the three-source recordings it falls from 0.12 (NG-IVA) to 0.10 for both AuxIVA and OverIVA. OverIVA therefore leaves the least residual cross-talk in both scenarios, AuxIVA is intermediate, and NG-IVA is consistently the weakest, exactly the ordering OverIVA $\leq$ AuxIVA $<$ NG-IVA observed on the synthetic mixtures. The absolute correlations remain non-negligible (≈ 0.10 even for the

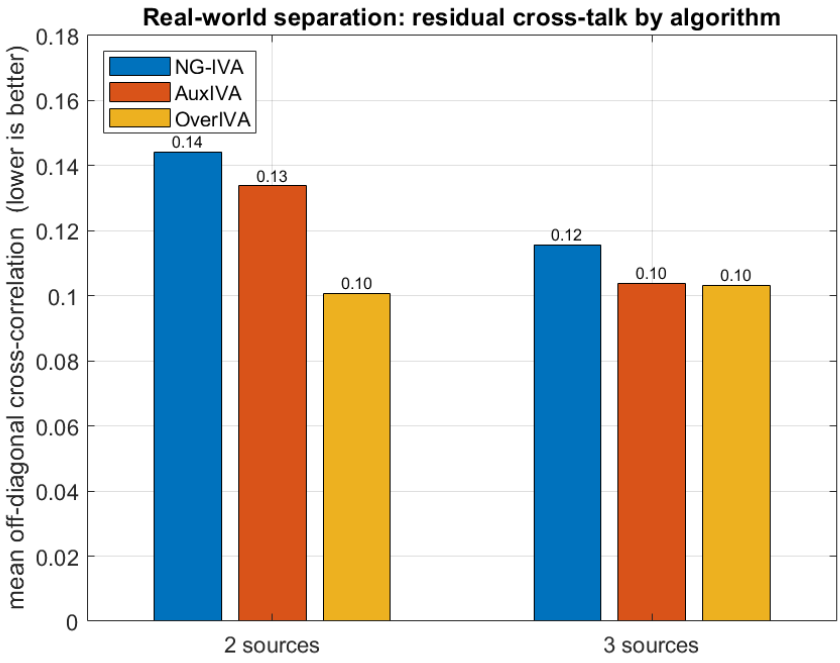


Figure 4.25: Mean off-diagonal correlation $\bar{\rho}$ per algorithm and number of sources (lower is better).

best engine), which reflects the heavy reverberation of the test room (Section 4.11.1) rather than a failure of separation: informal listening confirms that each output is clearly dominated by a single speaker. The gap between the three algorithms is narrower than in simulation because all three are limited by the same reverberant leakage floor, so the choice of algorithm matters less in this room than the acoustic conditions themselves.

4.11 Discussion

4.11.1 The Reverberation–Stationarity Trade-off

Real rooms are far more reverberant than the moderate RT60 values used for controlled validation in Chapters 2 and 3. The narrowband approximation underlying the IVA framework (Chapter 1, Section 1.2.3) requires the STFT frame to be longer than the room impulse response, so more reverberation argues for a longer analysis frame. But the same framework also assumes the source is quasi-stationary within a frame; an excessively long frame violates this for speech. These requirements pull in opposite directions, so the frame length cannot be increased freely to compensate for a reverberant room without degrading separation through the second mechanism.

4.11.2 The Critical Distance

The critical distance [6],

$$d_c \approx 0.057\sqrt{\frac{V}{\mathrm{RT60}}} \tag{4.2}$$

(room volume V in m^3, RT60 in seconds) is the source-to-microphone distance beyond which reverberant energy dominates the direct path. Beyond d_c the direct-to-reverberant ratio (Chapter 1, Section 1.3.1) drops below unity, and both separation and the localization methods of Chapter 3, which rely on the direct-path phase structure of the extracted transfer function, are expected to degrade. For a typical room volume and the RT60 measured here, d_c falls within the speaker-to-array distances realistically used in a meeting-room scenario, making this an active rather than theoretical constraint.

4.11.3 What Remains?

The immediate continuation is the set of items still marked MATLAB in Table 4.3: porting AuxIVA and OverIVA and the Post-filtering into the C++ library behind the existing separator interface; porting the remaining localization methods of Chapter 3 into the C++ DoA module; and hardening the live pipeline beyond its current two-source prototype. None of these requires any change to the architecture of Part A, each populates a slot already defined in Figures 4.5 and 4.15.

Conclusion and Perspectives

This work addressed the problem of real-time blind source separation and sound source localization for multi-speaker acoustic environments. Starting from an offline algorithmic study of Independent Vector Analysis, the project culminated in a functional, end-to-end audiovisual platform built around a native C++ processing engine, a BSS-based direction of arrival estimator, and an interactive graphical interface. The following sections summarize the contributions delivered, the limitations encountered, and the directions that would most meaningfully extend this work.

Summary of Contributions

This project delivered four tightly coupled technical contributions, each addressing a distinct layer of the processing pipeline.

1. **Real-Time C++ Processing Engine:** A modular, multithreaded C++ backend was designed and implemented to handle the full audio processing pipeline under strict latency constraints. The engine manages hardware interfacing with the MiniDSP UMA-16 array, cache-efficient STFT computation, online Voice Activity Detection (VAD), and memory layout optimized for real-time throughput. Both offline batch and true streaming online modes of Natural Gradient IVA (NG-IVA) are operational within this engine.

2. **IVA Algorithm Implementation and Evaluation:** The Natural Gradient IVA algorithm was implemented natively in C++ and validated against established references. In parallel, faster-converging variants, AuxIVA and OverIVA, were evaluated and benchmarked in MATLAB to characterize their convergence behavior and separation quality relative to NG-IVA.

3. **BSS-Based Direction of Arrival Estimation:** A geometric DoA extraction method was derived from the converged demixing matrix $\mathbf{W}(f)$ via pseudo-inversion, yielding spatial steering vectors without recourse to computationally expensive grid searches. Frequency-band aggregation was applied to produce robust azimuth and elevation estimates from the recovered mixing filters.

4. **Interactive Audiovisual GUI:** A PyQt-based graphical interface was developed to overlay estimated 3D acoustic source directions onto a calibrated live video stream. The interface enables real-time visual identification of active speakers, manual source selection, and dynamic audio routing, completing the end-to-end audiovisual pipeline.

Limitations

Despite the functional completeness of the platform, several constraints were encountered that define the boundary of its current applicability.

The most significant limitation is sensitivity to room reverberation. In reverberant environments, the convolutive mixing model accumulates long impulse response tails that cannot be adequately captured within the STFT frame lengths compatible with the quasi-stationarity assumption underlying IVA. Increasing the frame size to cover the full room impulse response tail would violate this assumption and destabilize the algorithm; operating at shorter frames leaves residual reverberation that degrades both separation quality and DoA accuracy. This fundamental tension motivates the dereverberation direction described below.

A secondary hardware constraint comes from the limited sensitivity of the MiniDSP UMA-16 microphone array, particularly at low input levels and in the presence of diffuse noise.

Perspectives

The limitations identified above directly inform the following development directions.

- **Advanced IVA Variants:** Several IVA extensions identified in recent surveys [48] represent high-priority algorithmic upgrades. Geometrically constrained IVA [49, 50, 51] incorporates array geometry as a prior to regularize the demixing solution, directly coupling the separation and localization objectives. ISS-IVA [52], based on Iterative Source Steering, achieves state-of-the-art convergence speed and is consistently recommended as the most competitive variant.
- **MIMO Dereverberation Preprocessing:** Integrating a blind MIMO dereverberation stage, such as Weighted Prediction Error (WPE) [53, 54] or a similar multi-channel linear prediction approach, as a preprocessing step upstream of IVA would decouple the reverberation suppression problem from the separation stage and relax the frame-length constraint. Alternatively, joint dereverberation and separation formulations [55], which model both phenomena within a unified probabilistic framework, offer a more principled integration and avoid the error accumulation inherent to sequential two-stage pipelines.
- **Higher-Quality Microphone Array:** Replacing the MiniDSP UMA-16 with a professional-grade array featuring higher sensitivity, and lower self-noise would improve the signal-to-noise ratio at the input of the separation stage and is expected to produce measurable gains in both separation quality and DoA accuracy without any algorithmic modification.
- **Platform Completion and Productization:** Further engineering effort is required to bring the platform to a deployable state, including robustness hardening, user interface refinement, and end-to-end system integration testing under realistic operating conditions.

The platform developed in this project establishes a solid, modular foundation on which each of these extensions can be integrated incrementally, without architectural redesign.

Bibliography

[1] A. Belouchrani, N. Mendjel, L. Berrah, and S. Tebache, "Independent Vector Analysis Based MIMO Deconvolution: Exploiting Spatial Diversity Through Back Projection," in *2023 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2023, pp. 478–481.

[2] S. Tebache, A. Belouchrani, L. Berrah, and N. Mendjel, "Minimal Distortion Principle versus Back Projection for Independent Vector Analysis," *ENP Engineering Science Journal*, vol. 5, no. 2, pp. 24–28, 2025.

[3] I. Merah and A.-Z. Ghecham, "Blind speech separation : Adaptive algorithm and implementation using UMA-16 v2 mic array testbed," Thesis, 2023.

[4] T. Bouznad, "Sound Source Localization and Tracking: Real life application using," Ph.D. dissertation, 2025.

[5] E. C. Cherry, "Some Experiments on the Recognition of Speech, with One and with Two Ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, Sep. 1953.

[6] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A Consolidated Perspective on Multimicrophone Speech Enhancement and Source Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, Apr. 2017.

[7] S. Makino, H. Sawada, and T.-W. Lee, Eds., *Blind Speech Separation*, ser. Signals and Communication Technology. Dordrecht: Springer Netherlands, 2007.

[8] H. Kuttruff, *Room Acoustics*, 6th ed. Boca Raton: CRC Press, Oct. 2016.

[9] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Pearson Education, 1993.

[10] Boashash, *Time- Frequency Signal Analysis and Processing*, Dec. 2015.

[11] J. Allen and L. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[12] K. Matsuoka, "Minimal distortion principle for blind source separation," *Proceedings of SICE*, vol. 4, pp. 2138–2143, Jan. 2002.

[13] M. R. Schroeder, "New Method of Measuring Reverberation Time," *The Journal of the Acoustical Society of America*, vol. 37, no. 3, pp. 409–412, Mar. 1965.

[14] Y. Jing and N. Xiang, "On boundary conditions for the diffusion equation in roomacoustic prediction: Theory, simulations, and experiments," *The Journal of the Acoustical Society of America*, vol. 123, pp. 145–53, Feb. 2008.

[15] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *The Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, Feb. 1975.

[16] J. Ball, "Voice Activity Detection (VAD) in Noisy Environments," Dec. 2023.

[17] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[18] S. Tong, H. Gu, and K. Yu, "A comparative study of robustness of deep learning approaches for VAD," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 5695–5699.

[19] P. Comon, "Independent Component Analysis, a new concept?" 1994.

[20] A. Hyvärinen, J. Hurri, and P. O. Hoyer, *Independent Component Analysis*. London: Springer London, 2001, vol. 39, pp. 151–175.

[21] A. J. Bell and T. J. Sejnowski, "An information -maximisation approach to blind separation and blind deconvolution," 1995.

[22] S.-i. Amari, A. Cichocki, and H. Yang, "A New Learning Algorithm for Blind Signal Separation," in *Advances in Neural Information Processing Systems*, vol. 8. MIT Press, 1996.

[23] A. Hyvärinen and E. Oja, "A Fast Fixed-Point Algorithm for Independent Component Analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[24] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, May 1999.

[25] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 530–538, Sep. 2004.

[26] T. Kim, T. Eltoft, and T.-W. Lee, "Independent Vector Analysis: An Extension of ICA to Multivariate Components," in *Independent Component Analysis and Blind Signal Separation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 3889, pp. 165–172.

[27] I. Lee, K. Taesu, and T.-W. Lee, "Fast fixed-point independent vector analysis algorithms for convolutive blind source separation," *Signal Processing*, vol. 87, pp. 1859–1871, Aug. 2007.

[28] T. Kim, "Real-Time Independent Vector Analysis for Convolutive Blind Source Separation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1431–1438, Jul. 2010.

[29] T. Taniguchi, N. Ono, A. Kawamura, and S. Sagayama, "AN AUXILIARY - FUNCTION APPROACH TO ONLINE INDEPENDENT VECTOR ANALYSIS FOR REAL-TIME BLIND SOURCE SEPARATION," 2014.

[30] R. Scheibler and N. Ono, "Independent Vector Analysis with more Microphones than Sources," Aug. 2019.

[31] T. Nakashima and N. Ono, "Inverse -free Online Independent Vector Analysis with Flexible Iterative Source Steering," Sep. 2022.

[32] C. H. Knapp and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," 1976.

[33] C. Zhang, D. Florencio, and Z. Zhang, "WHY does PHAT work well in low noise, reverberative environments?" in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Mar. 2008, pp. 2565–2568.

[34] J. H. DiBiase, "A High -Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays," 2000.

[35] R. Schmidt, "Multiple emitter location and signal parameter estimation," in *IEEE Transactions on Antennas and Propagation*, vol. 34, Mar. 1986, pp. 276–280.

[36] R. Roy and T. Kailath, "ESPRIT- estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[37] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind Source Separation Exploiting Higher-Order Frequency Dependencies," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 70–79, Jan. 2007.

[38] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. New Paltz, NY, USA: IEEE, Oct. 2011, pp. 189–192.

[39] S. Amari, T. P. Chen, and A. Cichocki, "Nonholonomic orthogonal learning algorithms for blind source separation," *Neural Computation*, vol. 12, no. 6, pp. 1463–1484, Jun. 2000.

[40] D. R. Hunter and K. Lange, "A Tutorial on MM Algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, Feb. 2004.

[41] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Apr. 2018, pp. 351–355.

[42] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[43] K. U. Simmer, J. Bitzer, and C. Marro, "Post-Filtering Techniques," in *Microphone Arrays: Signal Processing Techniques and Applications*, M. Brandstein and D. Ward, Eds. Berlin, Heidelberg: Springer, 2001, pp. 39–60.

[44] vantrees, "Optimum Array Processing | Wiley Online Books," https://onlinelibrary.wiley.com/doi/book/10.1002/0471221104, 2002.

[45] H. Pu, C. Cai, M. Hu, T. Deng, R. Zheng, and J. Luo, "Towards Robust Multiple Blind Source Localization Using Source Separation and Beamforming," *Sensors (Basel, Switzerland)*, vol. 21, no. 2, p. 532, Jan. 2021.

[46] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[47] Luxonis Community Forum, "Calibration error," https://discuss.luxonis.com/d/1030-calibration-error/5, 2023, accessed: 2026-06-25.

[48] R. Guo, Z. Luo, and M. Li, "A Survey of Optimization Methods for Independent Vector Analysis in Audio Source Separation," *Sensors*, vol. 23, no. 1, p. 493, Jan. 2023.

[49] A. H. Khan, M. Taseska, and E. A. P. Habets, "A Geometrically Constrained Independent Vector Analysis Algorithm for Online Source Extraction," in *Latent Variable Analysis and Signal Separation*, E. Vincent, A. Yeredor, Z. Koldovský, and P. Tichavský, Eds. Cham: Springer International Publishing, 2015, vol. 9237, pp. 396–403.

[50] L. Li and K. Koishida, "Geometrically Constrained Independent Vector Analysis for Directional Speech Enhancement," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 2020, pp. 846–850.

[51] L. Li, K. Koishida, and S. Makino, "Online Directional Speech Enhancement Using Geometrically Constrained Independent Vector Analysis," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 61–65.

[52] R. Scheibler and N. Ono, "Fast and Stable Blind Source Separation with Rank-1 Updates," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 2020, pp. 236–240.

[53] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and Biing-Hwang Juang, "Speech Dereverberation Based on Variance-Normalized Delayed Linear Prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, Sep. 2010.

[54] T. Yoshioka and T. Nakatani, "Generalization of Multi-Channel Linear Prediction Methods for Blind MIMO Impulse Response Shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, Dec. 2012.

[55] T. Nakashima, R. Scheibler, M. Togami, and N. Ono, "Joint Dereverberation and Separation with Iterative Source Steering," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 216–220.