Room Impulse Response estimation in Noisy Environments using Music as an Excitation Signal

MONIKA ŠERPATAUSKAITE'

Master's thesis
2026:E38

LUND UNIVERSITY

Faculty of Science
Centre for Mathematical Sciences
Mathematical Statistics

CENTRUM SCIENTIARUM MATHEMATICARUM

Master's Theses in Mathematical Sciences 2026:E38
ISSN 1404-6342

LUNFMS-1404-2026

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/

Abstract

This thesis investigates non-intrusive Room Impulse Response (RIR) estimation using music as an excitation signal in an acoustic environment with background noise present. The work studies the structure of room impulse responses, sparse background noise, babble noise, and music excitation signals in the context of RIR estimation. Building on the recently proposed AnyRIR estimator, which explored similar structures, weighted estimation methods incorporating background noise covariance are investigated under different background noise conditions.

Robustness to sparse background disturbances is achieved through the Huber objective function formulation, reducing sensitivity to sporadic interference and resulting in high restored speech intelligibility. In babble noise conditions, incorporating background noise covariance matrix as weight within the Huber loss function increases accuracy of the estimated RIR. However, this did not lead to substantial improvements in restored speech intelligibility, suggesting that intelligibility is limited more by remaining structured babble interference than by RIR estimation accuracy itself. Results show that covariance weighting alone with weighted least squares is insufficient for RIR estimation in the presence of babble noise, producing higher normalized mean squared error (NMSE) of the estimated RIR compared to covariance weighted Huber estimation.

Acknowledgements

I would like to thank my supervisors Andreas Jakobsson and David Sundström for their guidance. Thank you for consistently dedicating time to overview my progress, provide insights and brainstorm ideas as well as for your patience, encouragement, and flexibility throughout the thesis process.

I would also like to thank researchers K. Y. Lee, N. Meyer-Kahlen, K. Prawda, S. J. Schlecht and V. Välimäki. Their work on robust non-intrusive room impulse response estimation served as a major source of inspiration and their published ideas and code provided a foundation for this thesis.

Introduction

Room Impulse Response (RIR) describes the propagation of an impulse from a point source to a point receiver. It acts as an acoustic fingerprint of the room, encoding information about the room size, wall materials, furniture, and other sound-absorbing objects, as well as air absorption which depends on the temperature and humidity of the room.

The estimation of room impulse response plays a vital role in many acoustical signal processing problems. Acoustic echo cancellation uses room impulse response to predict the echo component in the recorded signal, commonly applied in telecommunications, suppressing unwanted echoes in video calls and hands-free systems [8]. Source localization problem aims to identify the sound source position based on how the acoustics of a room transform the source signal to the recorded one [7]. Another common research area in acoustic signal processing is speech dereverberation, where RIR allows reverberation to be removed or reduced from recorded speech signals, improving speech intelligibility [26].

A common method to measure the room impulse response is to use a known input signal and measure the output. An ideal sound source is one with infinite energy at time point 0 [15]. In practice, that is not possible to achieve, hence various methods have been developed that imitate such sound, having large amount of energy in a very short amount of time. Such excitation signals include gunshots, popping of balloons or a hand clap, which lack low-frequency content resulting in high-variance RIR estimates ([5], [21]) and cannot be perfectly reproduced.

These problems motivated the use of generated signals with rich frequency content. One commonly used method is the Maximum Length Sequence (MLS), which uses a periodic pseudo-random signal as the excitation signal [42]. However, the MLS method does not account for the non-linear distortions introduced by the measurement device or loudspeaker. An alternative approach of using sine sweep as the excitation signal overcomes such limitations [42]. The input signal is defined as a sine wave with exponentially growing frequency.

However, such approaches are intrusive and require controlled recording conditions. Uncontrolled environments, such as cafes, restaurants or live events often play background music, which inspired using music signals as excitation signals to evaluate room acoustics [36, 18].

Particularly, estimating room impulse response and using it to enhance speech signals masked by loud music is a subject of interest in forensics [18, 19]. Suspects of some criminal activity choose to have incriminating conversations in person, in a place where loud music conceals the contents of their dialogue. In this situation, if a microphone is present in the room, audio surveillance recording would include interfering music signal, speech from non-target speakers as well as other noises like clanking or door slamming, drowning the speech of interest, making it difficult to understand. Music identification software such as Shazam [49] can identify the song being played. After identifying and downloading the music track,

it can be subtracted from the recorded audio. However, if only the music is removed from the recording, the resulting audio will have an echo-like sound of the song due to the impulse response component still being present, corresponding to reflections of the music signal within the room. When the room impulse response is estimated accurately it can be removed from the recorded signal alongside the music track, resulting in a recording where the target speech is more intelligible.

This thesis focuses on room impulse response estimation in cafe-like environments where interfering background noise and music are present and the song playing is used as an input signal. Particular focus is placed on how different noise levels and structures, such as sparse disturbances, babble noise affect robustness of the estimation and recovered speech intelligibility. Recent approaches such as AnyRIR improve robustness to sparse background disturbances through Huber function formulations in the time-frequency domain. Building on this framework, this thesis investigates covariance weighted estimation methods designed for temporally correlated babble noise. Since applications of such methods often involve unveiling speech signals masked by music, speech intelligibility after removing the music and reverberation components is evaluated alongside accuracy of the estimated RIR and computational complexity.

Contents

1. Literature Review

Research investigating impulse response estimation in systems corrupted by additional noise has proposed a range of methods to improve estimation accuracy in noisy environments. This chapter overviews approaches addressing colored noise, sparse impulsive disturbances, and poor spectral excitation, which form the basis for the methodology developed in this thesis.

A method for estimating impulse response when system is corrupted with colored noise is explored in an article "Regularized impulse response estimation for systems with colored output noise" by E. C. Boeira and D. Eckhard [11]. Accounting for temporal correlation of the disturbing noise, the authors proposed Regularized Weighted Least Squares (RWLS) estimator, where regularization matrix controls the bias-variance trade-off and improving conditioning of the matrix to be inverted and the estimated covariance is used as a weighting matrix. The experiment was conducted using $N = 500$ samples per realization and a signal-to-noise ratio (SNR) of 10 dB. The identification procedure was repeated over 1000 Monte Carlo simulations. In each run, new realizations of the input signal and disturbance process were generated, and the impulse response was estimated using three different approaches: conventional Least Squares (LS), Regularized Least Squares (RLS), and the proposed Regularized Weighted Least Squares (RWLS) method. The performance of the different estimators was assessed using several metrics, including the norm of the bias, the trace of the covariance matrix, the trace of the mean-square error (MSE) matrix. The results demonstrated that the proposed RWLS approach achieved lower estimation variance and lower MSE than both the conventional LS and the standard regularized least-squares methods, highlighting the benefits of explicitly accounting for the covariance structure of colored disturbances during impulse response estimation.

J. Kim and J. Lavaei in article "Huber-based Robust System Identification with Near-Optimal Guarantees Across Independent and Adversarial Regimes" [34] explored the robustness of a Huber function based estimator for system identification and evaluated its performance under both persistent and sparse additive noise conditions. The proposed method was compared against conventional least-squares and ℓ_1-norm estimators. Under persistent noise, the Huber estimator achieved lower estimation errors than both competing methods when a sufficient number of observations N was available ($N \geq 500$). Under sparse noise conditions, Huber estimator consistently outperformed the least-squares estimator, resulting in significantly smaller errors. However, the ℓ_1-norm estimator achieved the highest estimation accuracy and was able to recover the true system with negligible error for sufficiently large sample sizes $N \geq 100$. The results suggest that Huber-based estimation provides a compromise between least-squares and ℓ_1-norm formulations, offering improved robustness to outliers while retaining high accuracy in the presence of persistent noise.

Huber loss function is also used in weights of adaptive filters in system identification tasks,

where system is disturbed by sparse, impulsive noise [37, 32]. S. M. Jung and P. Park proposed a modified Normalized Least Mean Square (NLMS) algorithm [32] in which a modified Huber function is incorporated into the adaptive filter update to reduce the influence of large residuals caused by impulsive noise. The method was evaluated using a randomly generated 32-coefficient system excited by white Gaussian noise. The observations were corrupted by three types of disturbances: additive noise on the input signal, additive Gaussian noise on the output signal, and sporadic high-amplitude impulsive disturbances modeled using a Bernoulli-Gaussian process. Averaged over 100 independent trials, the proposed algorithm achieved lower normalized mean square deviation (NMSD) than the conventional NLMS algorithm, demonstrating improved robustness to impulsive interference.

Article "AnyRIR: Robust non-intrusive room impulse response estimation method in the wild" explores room impulse response estimation in sparse noise conditions when the input signal is a music signal [36]. Similar to the previously discussed system identification methods in sparse noise conditions, the proposed approach employs a Huber-type objective function to reduce the influence of outliers in impulse response estimation. However, unlike the reviewed approaches, the residuals are computed in the time-frequency domain, exploiting the sparsity of audio signals in the spectral domain. The proposed algorithm is compared with least-squares estimate of the RIR over 50 simulations under both stationary noise conditions and stationary noise with additional sparse impulsive disturbances. When background noise is stationary the proposed algorithm improved accuracy marginally compared with least-squares estimate. However, in the presence of impulsive disturbances, the Huber-based formulation substantially outperformed the least-squares estimator, achieving an average impulse response estimation error of approximately -36 dB, compared to approximately -10 dB obtained using least-squares estimation. Additionally, problem of possible poor spectral excitation of the input signal is addressed. Music signals typically do not have flat power spectral density, which may lead to ill-conditioned Toeplitz matrix of the music signal. To flatten the spectrum of the input signal, inverse linear prediction filter is applied. Experimental results demonstrated that this preprocessing step significantly improves the conditioning of the estimation problem, reducing the number of iterations required for convergence from approximately 5000 to fewer than 600.

The reviewed literature highlights several challenges relevant to room impulse response estimation in noisy environments using music input signal. Since the excitation signal considered in this thesis is a music signal, the preprocessing strategy proposed in the AnyRIR framework is adopted to improve numerical stability and accelerate convergence of the estimation algorithm. Furthermore, both temporally correlated background noise and sparse impulsive disturbances are considered. To address these challenges, covariance matrix weighting is incorporated to account for the temporal correlation of the disturbance process, while a Huber objective function is employed to improve robustness to sparse outliers. Finally, because a common application of room impulse response estimation is speech enhancement, a speech signal is included as part of the observed output signal. From the perspective of system identification, the speech component acts as an interfering disturbance. To exploit the sparse spectral structure of speech signals and improve robustness to such interference, the residuals are evaluated in the time-frequency domain.

2. Theory

2.1 Sound

Sound can be described by kinetic energy made by vibrations of molecules of the medium it is traveling in [53]. A medium of interest in room impulse response estimation is air. Air molecules are moved by a vibrating object - a sound source (e.g. human speaking, music playing through a speaker) disturbing the neighboring particles from their resting position (equilibrium) causing air molecules to collide with each other. As air molecules are pushed together in the region closest to vibrating object (region "A"), it creates an area of high density, called compression. The neighboring region (region "B") has relatively lower density, not yet disturbed by vibrations of the sound source. The increase in molecule density in region "A", increases air pressure which pushes the molecules to lower pressure and density regions to restore equilibrium. Due to inertia of the particles they travel too far and create a low density region - rarefaction. The particles pushed to region "B" result in high density area. The process repeats, pushing particles away, creating a high pressure area in further region "C". As the particles move back and forth, they create a wave of compression and rarefaction transferring energy away from the sound source, resulting in sound traveling as seen in Figure 2.1.

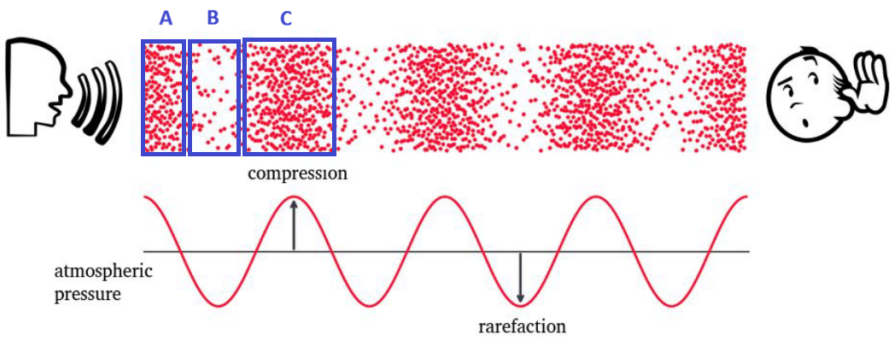


Figure 2.1: Representation of a sound wave: air pressure oscillation [50]

2.2 Sine waves

Cyclic behavior of air pressure variations mathematically can be represented as a function of time $x(t)$ described by a sine wave or sinusoid. A sinusoid function is expressed as

$$x(t) = A\sin(2\pi ft + \phi), \tag{2.2.1}$$

where A - amplitude, peak deviation of the function from zero, f - frequency, number of complete cycles per second, t - time, ϕ - phase, defines where in a cycle the sinusoid starts at time $t = 0$, where a full cycle is 2π. An example of a sine wave and how it shifts when phase, amplitude, and frequency are changed can be seen in Figure 2.2.

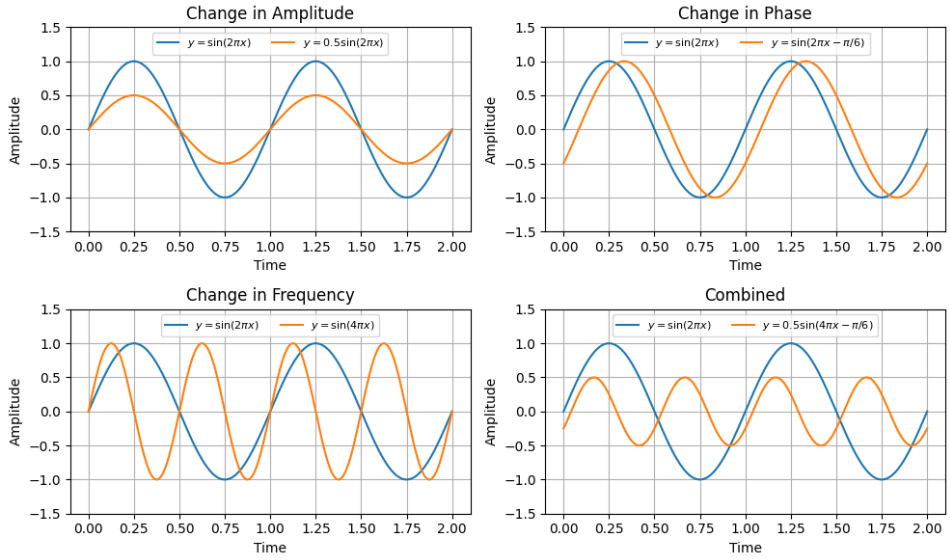


Figure 2.2: Affect of phase, amplitude and frequency change in a sine function

On the lower left plot, blue line repeats pattern every second, hence frequency, $f = 1$, while orange line repeats every 0.5 seconds or in other words has identical 2 periods in 1 second. Hence frequency f can be expressed as

$$f = \frac{1}{T}, \tag{2.2.2}$$

where T is length of 1 period.

Real-world audio signals are generally more complicated than a single sine wave. Instead, signal can include multiple oscillations in one period at different frequencies, amplitudes, and phases. Thus, a more complex signal can be expressed as a sum of sinusoids:

$$x(t) = A_0 + \sum_{k=1}^{K} A_k \sin(2\pi f_k t + \phi_k), \tag{2.2.3}$$

where each term corresponds to a sinusoidal component with its own amplitude, frequency, and phase, and A_0 denotes the DC (Direct Current) component, representing the constant offset (mean value) of the signal. One such signal is presented in Figure 2.3:

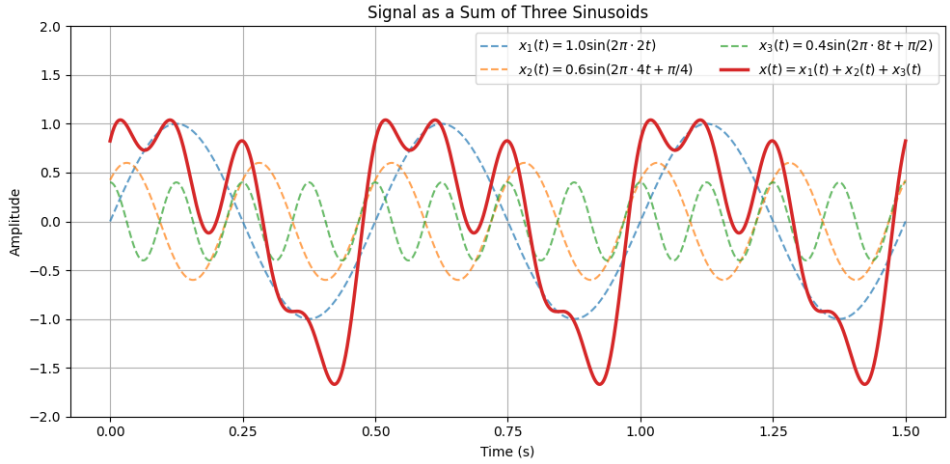


Figure 2.3: Sum of sine waves with fundamental frequency $f_0 = 2$

The resulting signal of three sinusoids is periodic, repeating it's pattern every 0.5 seconds. This is called the fundamental period and is noted T_0. The fundamental frequency is calculated using formula 2.2.2 and is noted as f_0.

The underlying process $x_1(t) = 1.0\sin(2\pi \cdot 2t)$, shown as the blue dashed line, completes two cycles per second and therefore has frequency $f_1 = 2$, $x_2(t) = 0.6\sin(2\pi \cdot 4t + \pi/4)$ (orange) - $f_2 = 4$, $x_3(t) = 0.4\sin(2\pi \cdot 8t + \pi/2)$ (green) - $f_3 = 8$. The underlying frequencies are integer multiples of the fundamental frequency f_0 and are referred to as harmonics. Integer multiples of f_0 not present in the signal, such as $3f_0$ in function $x(t)$ displayed in Figure 2.3, are still harmonic frequencies, but with zero amplitude coefficients.

Frequency components present in a signal are often represented using a spectrum. It displays the amplitudes of the individual sinusoids as a function of frequency as seen in Figure 2.4

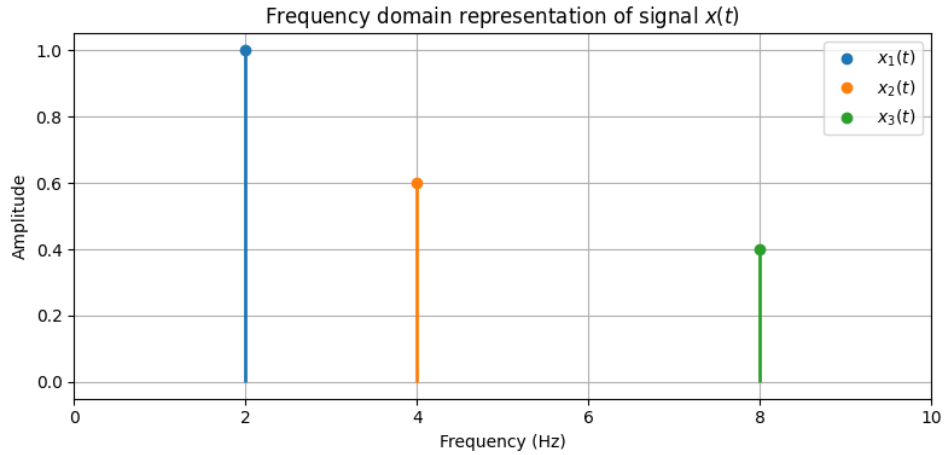


Figure 2.4: Frequency domain representation of signal $x(t)$

2.3 Fourier Analysis

2.3.1 Fourier Series

The french mathematician and physicist Jean Baptiste Joseph Fourier showed that any periodically repeating waveform can be expressed as a sum of sinusoids [53]. Hence, the purpose of Fourier analysis is to represent a signal as a sum of sine and cosine waves, revealing which frequencies are present and how strongly they contribute to the signal.

A general sinusoid can be expressed as a sum of sine and cosine functions

$$x(t) = A\sin(2\pi(ft+\phi)) = A_1\sin(2\pi ft) + A_2 cos(2\pi ft). \tag{2.3.1}$$

where amplitudes A_1 and A_2 satisfy

$$A = \sqrt{A_1^2 + A_2^2}, \qquad \phi = \arctan\left(\frac{A_2}{A_1}\right) \tag{2.3.2}$$

Thus, any real periodic signal $x(t)$ can be expressed as sum of sines and cosines

$$x(t) = \frac{A_0}{2} + \sum_{k=1}^{\infty} A_k \sin(2\pi k f_0 t) + \sum_{k=1}^{\infty} B_k \cos(2\pi k f_0 t), \tag{2.3.3}$$

This expression is called a Fourier series. The set of coefficients $\{A_k, B_k\}$ are referred to as the Fourier coefficients and are determined using formulas:

$$A_k = \frac{2}{T_0}\int_0^{T_0} x(t)\cos(2\pi k f_0 t)\,dt, \quad k = 1, 2, 3, \ldots \tag{2.3.4}$$

$$B_k = \frac{2}{T_0}\int_0^{T_0} x(t)\sin(2\pi k f_0 t)\,dt, \quad k = 1, 2, 3, \ldots \tag{2.3.5}$$

If the signal has zero mean, the DC component $A_0 = 0$. Otherwise, the value of the DC term is

$$A_0 = \frac{2}{T_0}\int_0^{T_0} x(t)dt. \tag{2.3.6}$$

Using Euler's formula [22]:

$$e^{\pm i\theta} = \cos(\theta) \pm i\sin(\theta), \tag{2.3.7}$$

sine and cosine function can be expressed in complex form as:

$$\cos(2\pi k f_0 t) = \frac{1}{2}\left(e^{i2\pi k f_0 t} + e^{-i2\pi k f_0 t}\right) \tag{2.3.8}$$

$$\sin(2\pi k f_0 t) = \frac{1}{2i}\left(e^{i2\pi k f_0 t} - e^{-i2\pi k f_0 t}\right) \tag{2.3.9}$$

Substituting sine and cosine definitions in complex form into equation 2.3.3 results in:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{i2\pi k f_0 t}, \tag{2.3.10}$$

This equation is called the synthesis equation of a Fourier Series, constructing the signal $x(t)$ using complex exponential basis functions. Here, X_k is defined as:

$$X_{+k} = \frac{A_k - iB_k}{2}, \quad X_{-k} = \frac{A_k + iB_k}{2}, \quad X_0 = \frac{A_0}{2}. \tag{2.3.11}$$

Substituting the real form Fourier Series equations for A_k and B_k (2.3.4, 2.3.5) into 2.3.11 yields:

$$X_k = \frac{1}{T_0} \int_0^{T_0} x(t) \left[\cos(2\pi k f_0 t) - i \sin(2\pi k f_0 t)\right] dt. \tag{2.3.12}$$

Using Euler's formula 2.3.7, the complex form of the Fourier series becomes:

$$X_k = \frac{1}{T_0} \int_0^{T_0} x(t)\, e^{-i2\pi k f_0 t}\, dt, \quad k = 0, \pm 1, \pm 2, \pm 3, \ldots \tag{2.3.13}$$

This equation is known as the analysis equation of a Fourier Series, as it allows to analyze how a signal can be represented by complex exponential basis functions. The complex coefficient X_k encodes both amplitude and phase information of the corresponding sinusoidal component. The real and imaginary parts of X_k are related to the Fourier coefficients A_k and B_k as:

$$A_k = 2\,\mathrm{Re}(X_k), \qquad B_k = -2\,\mathrm{Im}(X_k). \tag{2.3.14}$$

The magnitude and phase of the k-th frequency component can be obtained from X_k using:

$$|X_k| = \sqrt{\mathrm{Re}(X_k)^2 + \mathrm{Im}(X_k)^2} = \frac{1}{2}\sqrt{A_k^2 + B_k^2}, \tag{2.3.15}$$

$$\phi_k = \arg(X_k) = \arctan\left(\frac{\mathrm{Im}(X_k)}{\mathrm{Re}(X_k)}\right) = \arctan\left(\frac{-B_k}{A_k}\right). \tag{2.3.16}$$

The complex form of Fourier Series is most often used in research and computer algorithms due to computational efficiency and notation conciseness [48] .

2.3.2 Fourier Transform

While the Fourier series provides a sum of sines and cosines representation of periodic signals, defined over interval $(0, T_0)$, most real-world signals are not periodic, requiring a more general framework.

Extending Fourier series analysis to aperiodic signals involves treating those signals as periodic, but with a period $T_0 \to \infty$. Consequently, the spacing between the fundamental frequency and harmonics $\frac{k}{T_0}$ decreases. Hence, as $T_0 \to \infty$, the frequency spectrum becomes

continuous. This leads to the Fourier transform, which represents a signal as a continuous superposition of complex exponentials:

$$X(f) = \int_{-\infty}^{\infty} x(t)\, e^{-i2\pi ft}\, dt. \tag{2.3.17}$$

The complex function of the frequency f: $X(f)$, is equivalent to complex number X_k of the Fourier series and has encoded magnitude and phase:

$$X(f) = |X(f)|e^{i\theta(f)} \tag{2.3.18}$$

where $|X(f)|$ is the magnitude and is defined:

$$|X(f)| = \sqrt{(\mathrm{Re}(X(f)))^2 + (\mathrm{Im}(X(f)))^2}. \tag{2.3.19}$$

whereas the phase $\theta(f)$ is formed as:

$$\theta(f) = \arctan\left(\frac{\mathrm{Im}(X(f))}{\mathrm{Re}(X(f))}\right). \tag{2.3.20}$$

The inverse Fourier transform, converting signal from frequency domain representation $X(f)$ to original time-domain representation, may be expressed as:

$$x(t) = \int_{-\infty}^{\infty} X(f)\, e^{i2\pi ft}\, df \tag{2.3.21}$$

2.3.3 Sampling

Most Fourier analysis is performed using digital computers with finite memory, therefore signals are typically represented in discrete-time form before numerical processing [17]. To achieve this, the signal is sampled at regular time intervals T_s, known as the sampling period. The resulting sequence of samples forms a discrete-time signal, denoted as $x[n] = x(nT_s)$.

From a mathematical perspective, sampling can be described as the multiplication of the continuous-time signal $x(t)$ with a sampling function $p(t)$:

$$x_s(t) = x(t)p(t). \tag{2.3.22}$$

In digital systems, the sampling function is modeled as a periodic impulse train:

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s), \tag{2.3.23}$$

where $\delta(t)$ denotes the Dirac delta function. It is defined by

$$\delta(t) = 0, \quad t \neq 0, \qquad \text{and} \qquad \int_{-\infty}^{\infty} \delta(t)\, dt = 1. \tag{2.3.24}$$

The function is non-zero only when $t = 0$. At that point there is a spike such that, the area of the function equals 1, and this area is obtained at infinitesimal interval of time. Hence,

the $p(t)$ function has infinite amount of such spikes spaced at intervals T_s. Using the shifting property of the Dirac delta function:

$$\int_{-\infty}^{\infty} x(t)\,\delta(t-t_0)\,dt = x(t_0), \tag{2.3.25}$$

which allows it to extract the value of a signal at a specific time instant t_0, the sampled signal can be expressed as

$$x_s(t) = x(t)\sum_{n=-\infty}^{\infty}\delta(t-nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s)\,\delta(t-nT_s). \tag{2.3.26}$$

In the frequency domain, sampling results in a periodic repetition of the signal spectrum:

$$X_s(f) = \sum_{k=-\infty}^{\infty} X(f-kf_s), \tag{2.3.27}$$

where f_s is the sampling frequency defined as

$$f_s = \frac{1}{T_s}. \tag{2.3.28}$$

If the sampling frequency is not sufficiently large, these replicated spectra overlap, resulting in reconstructed signal to have artifacts - unwanted sounds, distortions. To avoid this effect, the sampling frequency must satisfy the Nyquist criterion:

$$f_s \geq 2f_{\max}, \tag{2.3.29}$$

where $f_{\max}$ is the highest frequency present in the signal.

2.3.4 Discrete Fourier Transform

If the signal of interest is observed over a finite duration $T = NT_s$, where N is the number of samples in the discrete-time signal $x[n]$, the signal can be analyzed using the Discrete Fourier Transform (DFT). The DFT is defined as

$$X_k = \sum_{n=0}^{N-1} x[n]e^{-i2\pi kn/N}, \quad k = 0, 1, \ldots, N-1. \tag{2.3.30}$$

The frequency domain representation X_k is evaluated at discrete frequency intervals $\Delta f = \frac{1}{T} = \frac{f_s}{N}$, resulting in evaluated frequencies to be:

$$0, \frac{1}{N}f_s, \frac{2}{N}f_s, \ldots, \frac{N-1}{N}f_s. \tag{2.3.31}$$

2.3.5 Fast Fourier Transform

The Fast Fourier Transform (FFT) is an algorithm that computes the same transform computationally efficiently. Computing the discrete Fourier transform of n samples will require about n^2 operations. If the length of the data sequence, n, happens to be a power of two, i.e., $n = 2^m$

for some integer m, the structure of the FFT allows X_k to be evaluated in merely $nlog_2n$ operations [39]. However, likely, the data length n is not a power of two. In such cases, one may add $N-n$ zeros at the end of the sequence, such that N is a power of two. The procedure to add zeros at the end of the sequence prior to computing the FFT is termed zero-padding.

Most common FFT method is the Cooley-Tukey algorithm [16], which exploits the symmetry of the DFT function. The value of X_{N+k} (2.3.30) can be expressed as:

$$X_{N+k} = \sum_{n=0}^{N-1} x_n \, e^{-i2\pi(N+k)n/N} \tag{2.3.32}$$

$$= \sum_{n=0}^{N-1} x_n \, e^{-i2\pi n} \, e^{-i2\pi kn/N} \tag{2.3.33}$$

$$= \sum_{n=0}^{N-1} x_n \, e^{-i2\pi kn/N} \tag{2.3.34}$$

where, using Euler's equation 2.3.7, $e^{2\pi in} = 1$ holds for any integer n. Hence, $X_k = X_{k+N}$ and for any integer l, $X_k = X_{k+lN}$.

The authors of the FFT algorithm suggested splitting DFT into two terms: one with even-number indices and the other with odd-number indices.

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n \, e^{-i2\pi kn/N} \\ &= \sum_{m=0}^{N/2-1} x_{2m} \, e^{-i2\pi k(2m)/N} + \sum_{m=0}^{N/2-1} x_{2m+1} \, e^{-i2\pi k(2m+1)/N} \\ &= \sum_{m=0}^{N/2-1} x_{2m} \, e^{-i2\pi km/(N/2)} + e^{-i2\pi k/N} \sum_{m=0}^{N/2-1} x_{2m+1} \, e^{-i2\pi km/(N/2)} \end{aligned} \tag{2.3.35}$$

Since components

$$E_k = \sum_{m=0}^{N/2-1} x_{2m} \, e^{-i2\pi km/(N/2)}, \qquad \text{and} \qquad O_k = \sum_{m=0}^{N/2-1} x_{2m+1} \, e^{-i2\pi km/(N/2)} \tag{2.3.36}$$

also use DFT formulation 2.3.30, $E_k = E_{k+N/2}$ and $O_k = O_{k+N/2}$. Using the result of 2.3.32, the component $e^{-i2\pi k/N}$ in 2.3.35 can be expressed as:

$$e^{-i2\pi(k+N/2)/N} = e^{-i2\pi k/N} \cdot e^{-i\pi} \tag{2.3.37}$$

Using Euler's equation: $e^{-i\pi} = -1$, the component becomes

$$e^{-i2\pi(k+N/2)/N} = -e^{-i2\pi k/N}. \tag{2.3.38}$$

Thus, the DFT can be expressed as

$$X_k = E_k + e^{-i2\pi k/N} O_k, \tag{2.3.39}$$

$$X_{k+N/2} = E_k - e^{-i2\pi k/N} O_k. \tag{2.3.40}$$

Using symmetry properties above, redundant computations are avoided. As long as the smaller DFT components have even number of indices m, and the decomposition is applied recursively, the computational complexity is reduced from $O(N^2)$ to $O(N \log N)$.

2.3.6 Short-Time Fourier Transform

Assumption of stationarity

The DFT and FFT provides frequency spectrum of a signal over a provided period of time. This implicitly assumes that signal's frequency components are constant over the length of the data provided - signal is approximately stationary over time.

(Weakly) stationary process is defined by a constant mean function and covariance function $r(s,t)$ depends only on the time difference $\tau = t - s$,where t and s are time points. However, audio signals like speech or music are not stationary processes [6], having changing statistical properties over time. By splitting the signal into segments that are approximately stationary and performing the Fourier transform on each segment, it can be seen how frequency components change over the duration of the signal. This approach is know as time-frequency analysis.

Windowing

Splitting signal into short temporal segments is called windowing. The simplest window function is the rectangular window defined as:

$$w[n] = 1, \qquad 0 \leq n \leq N_w - 1, \tag{2.3.41}$$

where N_w is the size of the window. Multiplying the signal $x[n]$ by $w[n]$ retains the signal within the window and sets it to zero elsewhere.

However, the rectangular window introduces abrupt changes and discontinuities at the endpoints, producing artifacts in the spectrum. To avoid this an alternative window function is applied in this thesis - Hann window [39]. The Hann window function is expressed as

$$w[n] = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{N_w - 1}\right), \qquad 0 \leq n \leq N_w - 1. \tag{2.3.42}$$

The function has a bell shape, smoothly tapering to zero at the endpoints, making discontinuities invisible and improving spectral representation.

Short-Time Fourier Transform

Evaluating how frequency components change over the duration of the signal is done by Short-Time Fourier transform. The signal is segmented using a window function $w[n]$, and the Fourier transform is applied to each segment:

$$X(m,k) = \sum_{n=0}^{N-1} x[n+m]\, w[n]\, e^{-i2\pi kn/N}, \qquad m = 0, 1, ..., M-1 \tag{2.3.43}$$

where m indexes the time frames and k indexes the frequency bins. This yields a time-frequency representation of the signal.

Spectrogram

The Short-Time Fourier Transform produces a matrix of complex values $X(m, k)$, where m indexes time frames and k indexes frequency bins. To visualize the result, the log-magnitude of the coefficients is generally computed:

$$20 \log_{10} (|X(m, k)|) .$$

This two-dimensional representation is commonly visualized as a heat map, referred to as a spectrogram. The logarithmic scaling is used to reflect the perceptual sensitivity of human hearing and expresses the magnitude on a decibel (dB) scale. The spectrogram of an example speech signal is shown in Figure 2.5.

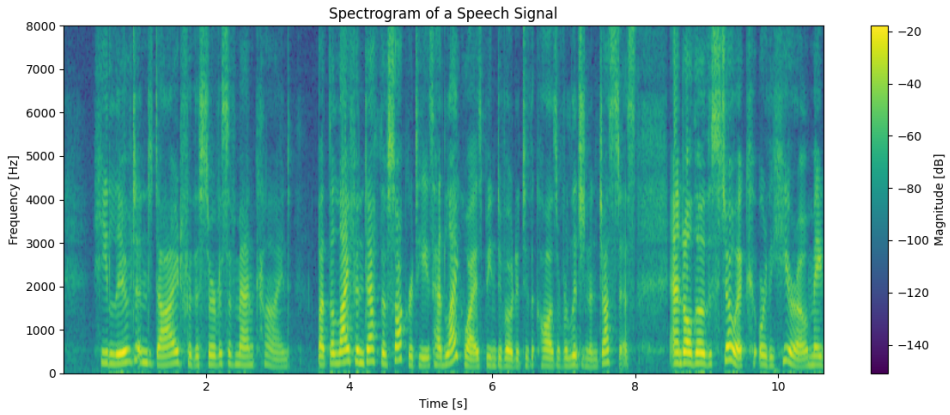


Figure 2.5: Spectrogram of a speech signal

In the figure, the signal of interest was transformed using Short-Time Fourier Transform with a Hann window of length 256 samples. The x axis of the plot represents time, y axis - frequency and the color indicates the magnitude in decibels. Most of the signal's energy is concentrated in the lower frequency bands and harmonic structures corresponding to the repeating horizontal bands, with the first half of the signal in the frequency range 3000-6000 Hz exhibiting relatively lower energy as compared to the later segments. Additionally, there are low-energy regions, pictured as darker vertical bands, indicating pauses or silence in the signal. These pauses and change in middle-range frequency component energy levels motivate the use of time–frequency analysis methods such as the Short-Time Fourier Transform.

2.4 Linear time-invariant system

A system is a process that transforms an input signal into an output signal. The objective of system analysis is to describe how a system modifies a signal.

2.4.1 Properties

A system $f(\cdot)$ maps an input signal x to an output signal y, with $y = f(x)$ being called linear if it satisfies the principle of superposition:

$$f(ax_1 + bx_2) = af(x_1) + bf(x_2), \tag{2.4.1}$$

for all constants a, b and signals x_1, x_2.

A system is called time invariant if a time shift of the input signals causes the same time shift of the output signals. That is, if $y[n]$ is the response to an input $x[n]$, for any shift in the input, $x[n-k]$, the response of the system will be $y[n-k]$:

$$x[n] \to y[n], \qquad x[n-k] \to y[n-k]. \tag{2.4.2}$$

Although no real-life is system is exactly linear and time invariant, LTI model proves to be a powerful tool, providing good approximations in practice [40].

2.4.2 Convolution

When the input signal $x[n]$ is a discrete time signal, it can be decomposed into sequence of individual impulses using the Dirac delta function 2.3.24:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]. \tag{2.4.3}$$

If the LTI system is f and the impulse function $\delta[n]$ is the input signal, the output of the system is

$$h[n] = f(\delta[n]). \tag{2.4.4}$$

Due to time-invariance property of the LTI system the following holds:

$$h[n-k] = f(\delta[n-k]) \tag{2.4.5}$$

The function $h[n]$ is referred to as the impulse response of the system. Applying the linearity property of the system 2.4.1, the output signal can be expressed as:

$$y[n] = \sum_{k=-\infty}^{\infty} x[n]h[n-k]. \tag{2.4.6}$$

This result is referred to as the convolution sum. The operation on the right-hand side of the equation is know as convolution and commonly written as:

$$y[n] = x[n] * h[n]. \tag{2.4.7}$$

2.4.3 Toeplitz matrix

Convolution can be rewritten with Toeplitz matrix [27]. A Toeplitz matrix is $n \times n$ matrix $T_n = \{t_{k,j}; k, j = 0, 1, \ldots, n-1\}$, where if $T_{k,j}$ is the k, j element of matrix T_n, then $T_{k,j} = T_{k+1,j+1} = t_{k-j}$, i.e., a matrix of the form:

$$T_n = \begin{bmatrix} t_0 & t_{-1} & t_{-2} & \cdots & t_{-(n-1)} \\ t_1 & t_0 & t_{-1} & \cdots & t_{-(n-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & \cdots & t_2 & t_1 & t_0 \end{bmatrix} \tag{2.4.8}$$

When impulse response is expressed as a Toeplitz matrix, one needs to take into account the properties of matrix operations. If the length of impulse response h and the input signal x is N_1 and N_2, respectively, the result of the convolution, the output signal y, has length $N = N_1 + N_2 - 1$. Usually $N_1 < N_2$, therefore the Toeplitz matrix of impulse response is zero padded to be of dimensions $N \times N_2$, so that

$$\mathbf{y} = H\mathbf{x} = \begin{bmatrix} h_0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & \cdots & 0 \\ h_2 & h_1 & h_0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ h_{N_1-1} & \cdots & h_2 & h_1 & h_0 \\ 0 & h_{N_1-1} & \cdots & h_2 & h_1 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & h_{N_1-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N_2-1} \end{bmatrix}, \tag{2.4.9}$$

where H is the impulse response Toeplitz matrix and $\mathbf{x}$ is vector representation of input signal x. An equivalent formulation can be obtained by constructing the Toeplitz matrix from the input signal instead of the impulse response. In this case, the Toeplitz matrix X is of $N \times N_1$ dimension and convolution is expressed as:

$$X\mathbf{h} = \begin{bmatrix} x_0 & 0 & 0 & \cdots & 0 \\ x_1 & x_0 & 0 & \cdots & 0 \\ x_2 & x_1 & x_0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ x_{N_2-1} & \cdots & x_2 & x_1 & x_0 \\ 0 & x_{N_2-1} & \cdots & x_2 & x_1 \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & \cdots & 0 & x_{N_2-1} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{N_1-1} \end{bmatrix}. \tag{2.4.10}$$

2.4.4 Frequency domain representation

The convolution operation in the time domain has an equivalent representation in the frequency domain. Applying the (DFT) to both sides of (2.4.7) yields

$$Y[k] = X[k]H[k], \tag{2.4.11}$$

where $X[k]$, $H[k]$, and $Y[k]$ are the frequency-domain representations of the input signal, impulse response and output signal, respectively.

2.4.5 System identification

The system identification problem seeks to estimate the impulse response $h[0], h[1], ..., h[L-1]$, using the input and output data.

Ordinary Least Squares

One method to estimate the impulse response of a system is ordinary least squares. Assuming that the output can be explained by a linear model:

$$\mathbf{y} = X\mathbf{h} + \mathbf{e}, \tag{2.4.12}$$

where the vector $\mathbf{e}$ denotes an additive noise, here assumed to be zero-mean white noise process with variance σ_e^2. The goal of ordinary least squares is to minimize the difference between the output signal $\mathbf{y}$ and the model prediction $X\hat{\mathbf{h}}$, where $\hat{\mathbf{h}}$ denotes the vector impulse response. The objective function to be minimized, also referred to as the loss function, is then given by

$$\hat{\mathbf{h}}_{LS} = \arg\min_{\mathbf{h}} ||\mathbf{y} - X\mathbf{h}||_2^2. \tag{2.4.13}$$

where the notation $||\mathbf{r}||_2^2 = \mathbf{r}^\top\mathbf{r}$ refers to the ℓ_2- norm of the vector $\mathbf{r}$. The estimate $\hat{\mathbf{h}}_{LS}$ is obtained using the normal equations yielding

$$\hat{\mathbf{h}}_{LS} = (X^\top X)^{-1}X^\top\mathbf{y}. \tag{2.4.14}$$

Weighted Ordinary Least Squares

When the system involves noise that does not exhibit white noise properties, the least squares method 2.4.13 can be modified to account for the disrupting sound. Defining the covariance matrix of noise as $\Sigma_{\mathbf{e}} = E\left\{\mathbf{e}\mathbf{e}^\top\right\}$, the weighted least squares estimate is

$$\hat{\mathbf{h}}_{WLS} = \arg\min_{\mathbf{h}} ||L(\mathbf{y} - X\mathbf{h})||_2^2, \tag{2.4.15}$$

where L comes from inverse of the covariance matrix $\Sigma_{\mathbf{e}}^{-1}$ decomposition:

$$\Sigma_{\mathbf{e}}^{-1} = L^\top L. \tag{2.4.16}$$

forming new variables $\tilde{\mathbf{y}} = L\mathbf{y}$, $\tilde{X} = LX$, and $\tilde{\mathbf{e}} = L\mathbf{e}$. Then,

$$\Sigma_{\tilde{\mathbf{e}}} = E\left\{\hat{\mathbf{e}}\hat{\mathbf{e}}^\top\right\} = E\left\{L\mathbf{e}\mathbf{e}^\top L\right\} = L\Sigma_{\mathbf{e}}L^\top = I, \tag{2.4.17}$$

meaning modified noise $\hat{\mathbf{e}}$ is a white process. Hence, adding a weighting with the covariance matrix of the noise is called prewhitening.

Resulting WLS estimate is expressed as

$$\hat{\mathbf{h}}_{WLS} = (X^\top\Sigma_{\mathbf{e}}^{-1}X)^{-1}X^\top\Sigma_{\mathbf{e}}^{-1}\mathbf{y}. \tag{2.4.18}$$

Covariance matrix

If $\gamma_{i,j} = Cov(e_i, e_j)$, where e_i denotes the additive noise sample at time index i, then for a stationary process e, $\gamma_{i,j} = \gamma_{i-j}$. Hence, element γ_k can be estimated by

$$\hat{\gamma}_k = \frac{1}{n}\sum_{i=1+|k|}^{n} X_i X_{i-|k|}, \qquad 1-n \le k \le n-1. \tag{2.4.19}$$

In case the mean is unknown, the estimate is modified to

$$\tilde{\gamma}_k = \frac{1}{n}\sum_{i=1+|k|}^{n}\left(X_i - \bar{X}_n\right)\left(X_{i-|k|} - \bar{X}_n\right), \qquad \text{where } \bar{X}_n = \frac{\sum_{i=1}^{n} X_i}{n}. \tag{2.4.20}$$

Then, the covariance matrix $\Sigma_{\mathbf{e}}$ is constructed as a Toeplitz matrix using the estimated auto-covariance sequence up to a chosen maximum lag order.

Iterative Reweighted Least Squares

If the system includes sparse noise, i.e., isolated bursts of sound, the ℓ_1-norm $||\mathbf{r}||_1 = \sum_{i=1}^{n} |r_i|$ is recommended to be used as a loss function [29].

Iterative Reweighted Least Squares is an algorithm used to minimize the l_1-norm objective function by solving a sequence of quadratic functions with appropriate weights [1]. When the objective function is an l_1-norm of the residual: $||\mathbf{y} - X\mathbf{h}||_1 = ||\mathbf{r}||_1$, the IRLS algorithm is formed as follows:

(i) Set initial $\hat{\mathbf{h}}^{(0)}$.

(ii) For k = 0, 1, 2, ...:

  (a) Calculate $\mathbf{r}^{(k)} = \mathbf{y} - X\hat{\mathbf{h}}^{(k)}$

  (b) Set $W^{(k)} = diag\{1/|\mathbf{r}^{(k)}|\}$

  (c) Calculate $\hat{\mathbf{h}}^{(k+1)} = (X^\top W^{(k)} X)^{-1}(X^\top W^{(k)}\mathbf{y})$

  (d) Repeat until some convergence check (relative change in parameter estimates $\hat{\mathbf{h}}$ or residual $\mathbf{r}$ ([41],[28]).

Generally, the weight function $W^{(k)} = w(\mathbf{r}^{(k)})$ is expressed as:

$$w(\mathbf{r}) = 2g'(\mathbf{r}^2), \tag{2.4.21}$$

where $g(\mathbf{r}) = \rho(\sqrt{\mathbf{r}})$ and $\rho(\mathbf{r})$ is the objective loss function [41]. Hence, the weight function can be written as:

$$w(\mathbf{r}) = \frac{\rho(\mathbf{r})}{\mathbf{r}}. \tag{2.4.22}$$

Huber loss function

The Huber loss function is a robust loss function, combining properties of the loss functions ℓ_1 and ℓ_2. It is defined as

$$\rho_\delta(\mathbf{r}) = \begin{cases} \frac{1}{2}|\mathbf{r}|^2, & \text{if } |\mathbf{r}| \le \delta, \\ \delta\left(|\mathbf{r}| - \frac{1}{2}\delta\right), & \text{if } |\mathbf{r}| > \delta, \end{cases} \tag{2.4.23}$$

where $\delta > 0$ is a threshold parameter that determines the transition between quadratic (ℓ_2-like) and linear (ℓ_1-like) behavior.

Least-Squares Minimal Residual

Solving the minimization problem

$$\hat{\mathbf{h}}^{(k+1)} = \arg\min_{\mathbf{h}} ||W^{(k)\frac{1}{2}}(\mathbf{y} - X\mathbf{h})||_2^2 \tag{2.4.24}$$

using normal equations can be memory intensive, especially if the input and output signals are evaluated in both time and frequency domain using STFT. To increase efficiency and decrease memory usage the least squares problem can be solved using the Least-Squares Minimal Residual (LSMR) algorithm [23].

The goal of the LSMR algorithm is to solve the least squares problem

$$\min_{\mathbf{h}} ||A\mathbf{h} - \mathbf{b}||_2^2, \tag{2.4.25}$$

by iteratively minimizing the residual norm $||A^\top \mathbf{r}||_2^2$, where $\mathbf{r} = A\mathbf{h} - \mathbf{b}$. The full LSMR algorithm can be found in Appendix A. The outline of the algorithm is as follows:

(i) LSMR uses the Golub–Kahan bidiagonalization process to construct the orthonormal bases U_{k+1} and V_k such that

$$AV_k = U_{k+1}B_k,$$

where B_k is a lower bidiagonal matrix. This reduces the original least squares problem to a smaller subproblem involving the bidiagonal matrix B_k.

(ii) Recursive QR factorization is applied to the bidiagonal matrices. QR factorization is a decomposition technique of a matrix B into a product of an orthonormal matrix Q and an upper triangular matrix R:

$$B = QR.$$

The QR factorizations transform the bidiagonal matrices into triangular form, allowing efficient recursive updates of the solution.

(iii) Plane (Givens) rotations are used to incrementally update the QR factorizations of the bidiagonal matrices. The orthonormal matrix Q_k is represented as a product of plane rotation matrices:

$$Q_k = P_{k-1} \dots P_2 P_1,$$

where each rotation matrix P_l acts only on rows l and $l+1$.

(iv) The estimate of h is updated iteratively using quantities obtained from the upper triangular matrices R_k.

(v) Checking some stopping criteria: stop if $||\mathbf{r_k}||_2$ or $||A^\top \mathbf{r_k}||_2$ is lower than some threshold.

It is worth noting that, the Python function `scipy.sparse.linalg.LinearOperator` enables the use of functions describing the operations $A\mathbf{h}$ and $A^\top \mathbf{v}$ (or, in the complex case, $A^H \mathbf{v}$, where A^H denotes the Hermitian transpose of A) instead of explicitly forming the matrix A.

By iteratively solving the least squares problem, avoiding storing entire matrix A, LSMR significantly reduces memory requirements and provides an efficient solution of the large-scale weighted least squares problem within each IRLS iteration.

2.5 Preprocessing

2.5.1 Linear Prediction Filter

Linear prediction models a signal as a linear combination of its past samples, i.e.,

$$x[n] \approx \sum_{k=1}^{p} a_k x[n-k], \tag{2.5.1}$$

where $\{a_k\}_{k=1}^{p}$ are the linear prediction coefficients and p is the model order.

The corresponding prediction error signal can be obtained by filtering the signal with the inverse prediction filter

$$e[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k]. \tag{2.5.2}$$

This operation effectively removes predictable (correlated) structure from the signal, resulting in a residual that is closer to white noise.

2.6 Room Impulse Response

When the linear time-invariant system in question is a room, the impulse response is referred to as the room impulse response (RIR). In room acoustics, RIR characterizes how sound propagates from sound source to receiver (e.g. microphone) in a specific room.

2.6.1 Room Impulse Response components

The impulse response is created by a sound propagating from the excitation source (in LTI systems referred to as the input signal) in wave form and bouncing around the room.

Sound traveling straight from a source to a receiver is called direct sound and its' trajectory - direct path. The time that it takes for the direct sound from the sound source to reach the measurement position is denoted the propagation delay time.

Sound propagating in other directions strikes a surface (e.g. walls, ceilings, floor) and part of the wave energy reflects in the form of a wave with a different amplitude and phase from the original wave. The fraction of wave energy lost during the reflection, where energy is proportional to amplitude squared, is called the absorption coefficient and it depends on the material of the surface [35]. After the direct sound reaches the microphone, the next most prominent features to be recorded is the sound arriving after the lowest amount of reflections or lowest order reflections. These waves are referred to as the early reflections.

A collection of reflected copies of the original sound that arrive later with lower amplitudes is called late reverberation, forming an exponential decay in magnitude of recorded sound.

One of room acoustic parameters is the RT60 Reverberation time - this is the duration of time it takes for reverberant sound in a room to decay by 60 dB after the sound source has stopped emitting sound. The rate at which the sound decays is related to the volume of the room, the materials of the reflecting surfaces and their absorption coefficients. This relationship is typically modeled using the Sabine equation [46]:

$$RT_{60} = \frac{0.161V}{S_{tot}a_{tot}}, \tag{2.6.1}$$

where V is the volume of the room in m^3, S_{tot} - area of all surfaces in the room, and a_{tot} is the total absorption coefficient of the room, calculated as follows:

$$a_{tot} = \frac{\sum_i a_i S_i}{S_{tot}}, \tag{2.6.2}$$

where a_i denotes the absorption coefficient for a particular area S_i. Typical values of reverberation times range from about 0.3 seconds (living rooms) up to 10 seconds (large churches). Large halls often have RT60 values between 0.7 and 2 seconds [35].

2.6.2 Room Simulation

Recording room impulse responses is tedious and time consuming work. Thus, there have been many efforts to simulate acoustics of a room.

Image Source Method

The Image Source Method (ISM) introduced by Allen and Berkley, is one of the most used room acoustic simulation methods [31]. It models sound reflection paths between the source and receiver, where wall reflections are modeled as virtual sound sources as well - image sources. The real source S is mirrored across the wall, creating the virtual image source S'. The mirrored source is placed at equal perpendicular distance from the wall as the original source as shown in figure 2.6. ISM computes a straight path from S to M, where M is the microphone. The point, where the straight line intersects the wall is the reflection point R. The reflected acoustic path S→R→M has exactly the same total propagation distance as the straight virtual path S'→M. This allows the reflection delay to be computed simply from the Euclidean distance between image source and microphone.

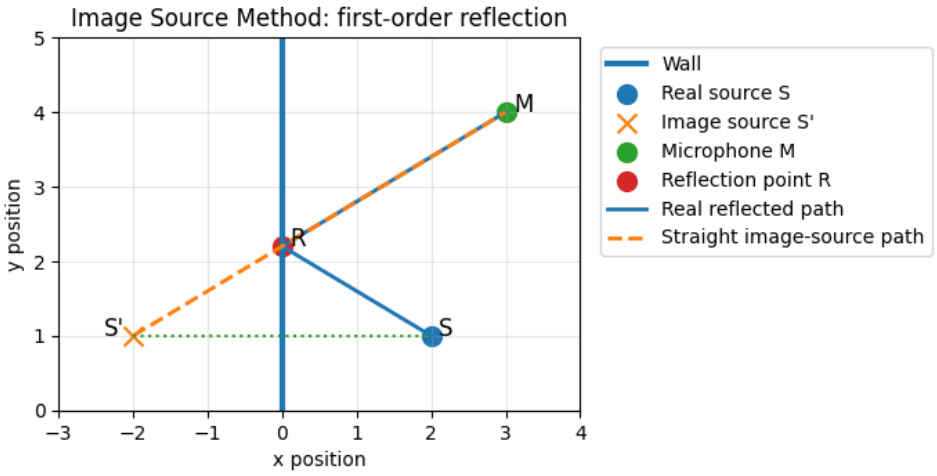


Figure 2.6: Image Source method with one reflection

The same idea extends to multiple reflections. For a second-order reflection the image source is mirrored across two walls successively, resulting point S" is seen in Figure 2.7. The point on the line between S" and M, where it intersects with the y axis becomes a reflection

point. The second reflection point is the intersection between the second wall (x axis) and the straight line from first image source S' behind this wall and already the determined reflection point R2. For higher order reflections the process is repeated, including other perpendicular walls as well as floors and ceiling.

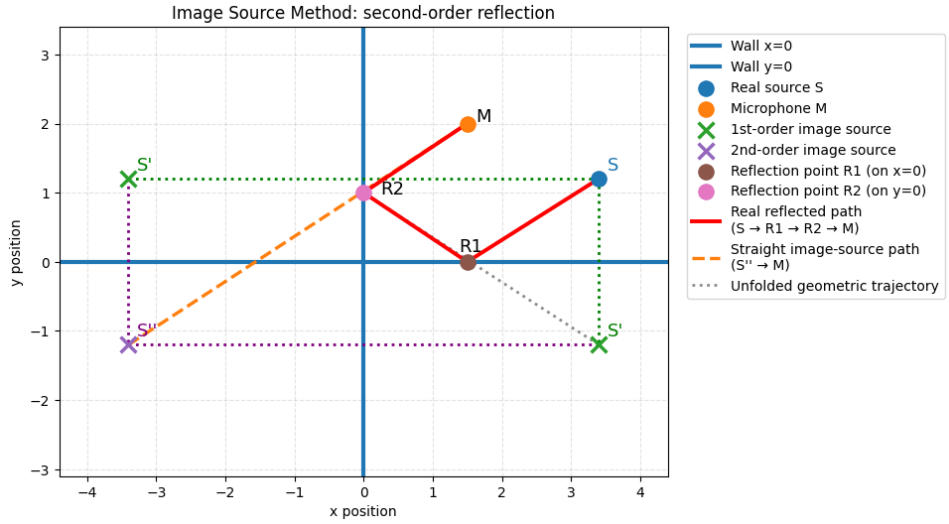


Figure 2.7: Image source method modeling of 2 reflections

The RIR is formed as a sum of delayed and scaled signal copies. The delay of each impulse is determined by the propagation distance and speed of sound, while the amplitude depends on absorption of the walls involved in the reflection path. Consequently, early reflections appear as distinct impulses in the RIR, whereas higher-order reflections become increasingly dense and form the reverberant tail.

The image source method is often used in a shoe-box room - a pallelepipedic rooms with 4 or 6 walls (in 2D and 3D respectively), all at right angles and polyhedral rooms [12]. In highly symmetric shoebox rooms, the regularity of the image sources' positions leads to a monotonic convergence in the time arrival of high order reflection image sources. This causes artifacts called sweeping echoes [24]. The randomized image method adds a small random displacement to the image source positions, so that they are no longer time-aligned, thus reducing sweeping echoes.

Ray Tracing

Ray tracing (RT) is an alternative method based on idea of emitting many rays in random directions from the source position. Each ray propagates through the room, reflecting from surfaces and losing energy. The microphone is modeled as a small capture region, and whenever a ray passes through this region, its arrival time and remaining energy are recorded. The room impulse response is then constructed by accumulating the energies of all detected rays into time bins corresponding to their propagation delays. An example of the ray tracing method is provided in Figure 2.8, where 15 rays are emitted from source point S in random directions. The ray path in red crosses the radius of microphone M and at that point is recorded, but does not stop emitting up to a set time, reflection order or energy threshold.

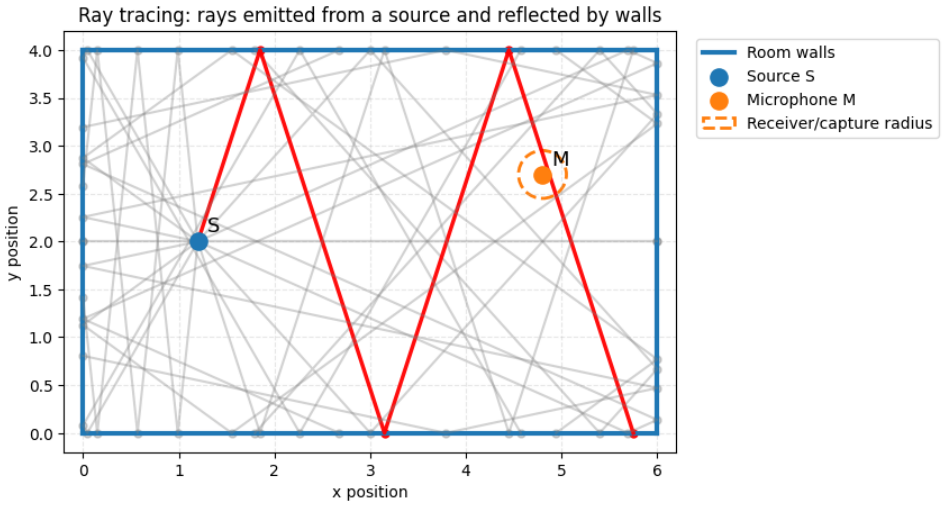


Figure 2.8: Ray tracing simulation

In real rooms, late reverberation behaves as a dense and diffuse sound field. Because ray tracing method considers not only specular reflections, but also diffuse ones, launching many rays that gradually spread throughout the room, ray tracing method produces a more realistic reverberation tail than finite-order image source simulations [47].

Using the strengths of both room acoustic simulation methods, the hybrid ISM/RT simulator was introduced. It uses ISM to simulate the early reflections in the room impulse response and RT for the diffuse tail [51].

2.7 Evaluation Metrics

2.7.1 Normalized Mean Square Error

Evaluating the accuracy of impulse response estimation can be done with the normalized mean square error (NMSE) metric [9, 4]. When $\mathbf{h}$ and $\hat{\mathbf{h}}$ are true and the estimated impulse response respectively, the NMSE is defined as

$$NSME = \frac{||\mathbf{h} - \hat{\mathbf{h}}||_2^2}{||\mathbf{h}||_2^2}. \tag{2.7.1}$$

2.7.2 Short-time objective intelligibility

A common goal of identifying the room impulse response is to enhance speech signals present in a room. To evaluate how well a speech signal was recovered, one may use the Short-time objective intelligibility (STOI) [30] metric. It evaluates speech intelligibility and has been shown to be strongly correlated with results of subjective listening tests.

STOI operates on the clean speech signal x and the processed speech signal y, which are first resampled to 10 kHz and aligned in time. Both signals are segmented into 50% overlapping Hann-windowed frames of length 256 samples. Each frame is zero-padded to 512 samples and transformed using the discrete Fourier transform.

Neighboring frequency bins are grouped into one-third octave bands. An octave corresponds to a doubling of frequency, while a one-third octave band divides this interval into three smaller frequency ranges. In STOI, 15 one-third octave bands are used, starting from a center frequency of 150 Hz. The energy of the j-th one-third octave band in frame m is defined as

$$X_j(m) = \sqrt{\sum_{k=k_1(j)}^{k_2(j)-1} |\hat{x}(k,m)|^2}, \tag{2.7.2}$$

where $\hat{x}(k,m)$ denotes the k-th DFT coefficient of frame m, while $k_1(j)$ and $k_2(j)$ represent the lower and upper DFT-bin indices of the corresponding one-third octave band. The processed speech representation $Y_j(m)$ is computed analogously.

For each one-third octave band, STOI evaluates short temporal segments consisting of N consecutive frames (best choice of N is proved to be 30, corresponding to approximately 400 ms of speech). The processed speech segment is first normalized such that its energy matches the energy of the clean speech segment, scaling with a factor α:

$$\alpha = \sqrt{\frac{\sum_n X_j(n)^2}{\sum_n Y_j(n)^2}}. \tag{2.7.3}$$

After normalization, $\alpha Y_j(n)$ is clipped to lower bound the signal-to-distortion ratio (SDR):

$$SDR_j(n) = 10 \log_{10} \left(\frac{X_j(n)^2}{(\alpha Y_j(n) - X_j(n))^2} \right) \tag{2.7.4}$$

Clipping prevents severely distorted time-frequency components from dominating the intelligibility estimate. The clipped signal is defined as

$$\tilde{Y} = \max \left(\min \left(\alpha Y, X + 10^{-\beta/20} X \right), X - 10^{-\beta/20} X \right),$$

where β denotes the minimum allowed SDR level. Maximum correlation with human listening test intelligibility is obtained with $\beta = -15$.

The intermediate intelligibility measure is then computed as the linear correlation coefficient between the clean and processed time-frequency units:

$$d_j(m) = \frac{\sum_n (X_j(n) - \mu_X)(\tilde{Y}_j(n) - \mu_Y)}{\sqrt{\sum_n (X_j(n) - \mu_X)^2 \sum_n (\tilde{Y}_j(n) - \mu_Y)^2}},$$

where μ_X and μ_Y denote the mean values of the clean and processed time-frequency units within the considered time segment. This correlation measures how similarly the clean and processed speech vary over time within each one-third octave band.

Finally, the STOI score is obtained by averaging the intermediate correlation values across all time frames and one-third octave bands:

$$d = \frac{1}{JM} \sum_{j,m} d_j(m),$$

where J is the number of one-third octave bands and M is the number of frames.

While the theoretical range returned from STOI is [-1,1], the practical range is closer to [0.4,1]. When the STOI metric approaches 0.4 the speech approaches unintelligibility [43].

3. Methodology

3.1 AnyRIR

3.1.1 Algorithm

AnyRIR is a room impulse response estimation method presented in the paper "AnyRIR: Robust non-intrusive room impulse response estimation method in the wild" [36] focusing on system identification in an uncontrolled environment. The setting used involves a room with music track playing alongside sparse noise - footsteps, door getting shut, chair scraping the floor and silverware clanking. Because such background noise is non-stationary and is not normally distributed, l_2-based deconvolution methods are not applied, as sparse noise would break assumptions of stationarity and Gaussian distribution of residuals. To estimate impulse response corrupted by sparse noise a regression based approach is adopted using the l_1-norm loss to suppress the influence of outliers in impulse response estimation algorithm.

The formulation of the problem becomes:

$$\hat{h} = \arg\min_{h} ||y - x * h||_1, \tag{3.1.1}$$

where h is room impulse response, x - excitation signal (song) and y is the resulting signal captured by the microphone, that can be expressed as $y = x*h+n$, n representing background noise. Additionally, the background noise is assumed to be sparse in frequency domain, suggesting the problem

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} ||S\mathbf{y} - SX\mathbf{h}||_1, \tag{3.1.2}$$

where S denotes the short-time Fourier transform 2.3.43 with Fast Fourier transform applied to each segment.

The IRLS algorithm is employed to minimize the ℓ_1 norm objective function in3.1.2. Huber loss function for weight update scheme is chosen to make the estimation robust to non-stationary and non-Gaussian noise. Using formula 2.4.22, where $\rho(r)$ is the Huber loss function, the weights of the IRLS algorithm are

$$w(\mathbf{r}) = \begin{cases} 1, & \text{if } |\mathbf{r}| \leq \delta, \\ \frac{\delta}{|\mathbf{r}|}, & \text{if } |\mathbf{r}| > \delta. \end{cases} \tag{3.1.3}$$

The resulting algorithm can be written as

$$\mathbf{h}^{(i+1)} = \arg\min_{\mathbf{h}} \left\| \mathbf{w}^{(i)} \circ \big(S(\mathbf{y} - X\mathbf{h}^{(i)})\big) \right\|_2^2, \tag{3.1.4}$$

$$\mathbf{r}^{(i+1)} = S\big(\mathbf{y} - X\mathbf{h}^{(i+1)}\big), \tag{3.1.5}$$

$$\mathbf{w}^{(i+1)} = \frac{1}{\max\big(|\mathbf{r}^{(i+1)}|, \delta\big)}, \tag{3.1.6}$$

where $\circ$ denotes the Hadamard (elementwise) product. In the implemented code, δ is set to be equal to the standard deviation of background noise.

The IRLS algorithm requires a stopping algorithm to be chosen. The stopping criteria is based on the relative change in the estimated impulse response in consecutive iterations:

$$\frac{||\hat{\mathbf{h}}^{(\mathbf{k})} - \hat{\mathbf{h}}^{(\mathbf{k-1})}||_1}{||\hat{\mathbf{h}}^{(\mathbf{k-1})} + 1 \cdot 10^{-6}||_1} < \epsilon, \tag{3.1.7}$$

where $\epsilon = 1 \cdot 10^{-6}$, and $1 \cdot 10^{-6}$ in the denominator of the expression is set for numerical stability, to avoid division by zero.

The matrix $A^{(i)} = \mathbf{w}^{(i)}SX$ is very large (if the length of the signal x is l, n_{fft} - number of frequency bins per time segment, n - number of time points per time segment, the matrix is of dimension $M \times n_{fft}$, where $M = \lceil l/n \rceil$). Therefore, to solve the minimization problem in algorithm 3.1.4, LSMR is used to reduce memory usage. Additionally, the matrix $A^{(i)}$ is never explicitly constructed. Instead the `LinearOperator` function is used with functions of operations $A\mathbf{h}$ and $A^H\mathbf{v}$ as arguments. More specifically, the functions are defined as:

$$A^{(i)}\mathbf{h} = W^{(i)}\, S(X\mathbf{h}), \tag{3.1.8}$$

$$(A^{(i)})^H\mathbf{v} = X^H S^H (W^{(i)}\mathbf{v}), \tag{3.1.9}$$

where $\mathbf{h}$ and $\mathbf{v}$ are input vectors of the function.

3.1.2 Preprocessing

When a linear system is ill-conditioned, iterative methods solving the least squares problem $||A\mathbf{h} - \mathbf{b}||_2^2$ can often diverge, leading to large amounts of computations or inaccurate results. Thus, it is common to use preconditioning of the matrix X for faster convergence [33].

In the present setting, X corresponds to a convolution operator, built from shifted copies of the input signal. If the signal has high energy in narrowband frequency range, i.e., has poor spectral excitation, the columns of the Toeplitz matrix X become highly correlated with one another. For example, if the signal is a single sinusoid, the shifted column values differ only by phase, hence columns are close to being linearly dependent. In contrast, if the signal is spectrally rich or approximately white, neighboring columns of the matrix X have smaller correlation. When columns are highly correlated, multiple impulse responses may produce nearly identical convolutions $X\mathbf{h}$. Consequently, different impulse response estimates can yield similar residual errors. Since iterative solvers use the residual error to determine update directions, similar residuals provide limited information about how the estimated impulse response should be adjusted. This may lead to slow convergence, numerical instability, and increased sensitivity to noise.

Music signals typically exhibit strong harmonic and rhythmic structure, with energy concentrated around the fundamental frequency and the harmonics rather than being uniformly distributed across frequencies [20], leading to to an ill-conditioned system matrix X, slowing down convergence and poor estimates of RIR.

To mitigate this issue, a preprocessing step is applied to both the excitation signal x and the measurement signal y in order to flatten their spectrum. The authors achieve this by downsampling the signals, injecting high-frequency noise to compensate for missing spectral content, and applying a shared inverse linear prediction filter (2.5.2) of order 200. The linear prediction filter acts as a whitening transformation. Using the same filter for both signals ensures that the convolutional relationship between them is preserved.

3.2 Room simulation

A simulated room was created using the `pyroomacoustics` library. The scenario of interest includes music and speech, which is a usual setting for coffee shops, restaurant, or bars. The total area of a medium sized cafe ranges from 70-150 m^2 [44], thus width and length of a room is set to be 8 and 9 meters, making the the area to 72 m^2. The height of the room is set to be 2.7 meters, reflecting standard height of a commercial building [13]. Thus, the dimensions of the room where set to be $9 \times 8 \times 2.7$ meters.

In the library `pyroomacoustics`, having set the dimensions of the room, the absorption coefficients can be specified by specifying materials of the walls (e.g. concrete, marble, ceramic tiles), or choosing the reverberation time RT60 and using the inverse of Sabine's formula (2.6.1) to calculate the total absorption coefficient of the room. In this setting, RT60 is set to 0.6 seconds [54]. The inverse Sabine function also provides the maximum number of order of reflections needed to achieve the desired RT60 value. In the AnyRIR algorithm, the length of the impulse response needs to be specified. Since RT60 is 0.6 seconds, length of the estimated RIR is set to be 0.7s.

To simulate a room, a hybrid method of image source method and ray tracing is used to achieve the most realistic room impulse response.

The music source is placed at a random width and length point of the room with specified height of 2.5m, since speakers emitting music are usually put near the ceiling of the space. What is more, a person speaking is placed at a random choice of length and width coordinate of the room with 1.3 meter height position, which is the average distance from the the floor to human head, when a person is sitting [52].

The microphone position is chosen to be in a corner of the room at a human speech level height, with coordinates [8.8m, 7.4m, 1.3m].

3.3 Speech and Music Signals

Speech signal is taken from LibriSpeech - a large English speech recording library, where each recording has sampling frequency of 16kHz [14]. Music audio files are commonly stored using a sampling frequency of 44.1 kHz, which originates from the Compact Disc (CD) digital audio standard introduced by Sony and Philips in the early 1980s [45]. Hence, the music signal used in this simulation is originally provided with 44.1kHz sampling frequency and is then

downsampled to the speech sampling frequency of 16kHz. This is chosen to be the sampling frequency of the room simulation as well.

The speech and music are peak-normalized by dividing each signal by its maximum absolute amplitude. Peak normalization removes differences in signal amplitude while keeping the original timing and frequency content unchanged. Consequently, the relative loudness between speech and music sources is determined explicitly through the applied scaling factors rather than the original recording amplitudes. The music signal is multiplied by a factor of 3. Thus, before room simulation, the target speech has peak amplitude 1, the music signal has peak amplitude 3. This produces a setting where the interfering music is louder than the target speech.

The input x - the music signal and the output y are preprocessed by calculating the short time Fourier transform, with frame size of 256 sample points cut by Hann window, each of these segments is zero-padded to 512 points and fast Fourier transform is applied. Then, the signals are injected with high-frequency noise and a shared inverse linear prediction filter is applied to flatten the spectrum.

The weight function requires information about the standard deviation of background noise, δ. In case of no additional background noise added, the standard deviation is estimated by using the last 0.5 seconds of preprocessed output signal, which is a time segment where the music and the speech signals are quiet.

3.4 Babble Noise

To simulate a more realistic setting of an uncontrolled environment or coffee shop environment, babble noise is simulated, using 20 different speech signals from the library LibriSpeech. The signals are placed at random positions in the room with a 30 cm margin, so that the speech signal does not come from a position right at the boundary of the wall. The height parameter is bounded by 1.3 and 2 meters simulating a speaker either from a standing or a sitting position.

Babble noise sources are peak-normalized alongside speech and music signals and multiplied by a factor of 0.3. Hence, each individual babble source is quieter than both the music and the target speech before room propagation.

To calculate the standard deviation and the covariance matrix of the background noise to use in weight update equation of IRLS and weighted least squares problem, respectively, a second room simulation is generated. The background noise room simulation has the same dimensions with only 20 different speech signals scattered around the room at the same positions. The 20 speech signals are of the same speakers used in first room to simulate babble noise of the same people talking, with different contents of speech.

3.5 Sparse Noise

A recording with sparse background noise of 1 minute and 1 second was obtained. The speech signal of interest is approximately 10 seconds long, then when simulating a room with sparse background noise present, a random snippet of length equal to the speech of interest length is chosen from the sparse background noise recording. The unused recording parts are used to calculate the standard deviation and covariance matrix.

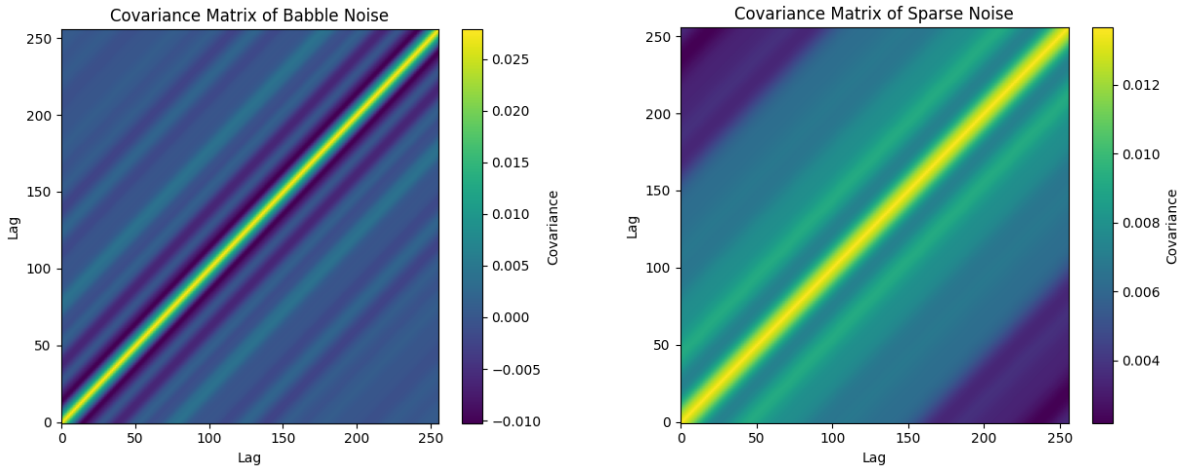
The source point of sparse background noise is placed in the middle of the simulated room, with coordinates [4.5, 4, 1.35].

The recording contains different noises: a chair being pulled and scraping the floor, utensils clanking, a person clearing their throat and other banging sounds. Because the choice of snippet to use in the simulation is random: one snippet can be mostly silent, other with a quiet noise and other may contain constant disruption, 10 simulations are performed to decrease bias introduced by randomness and results are averaged.

3.6 Covariance matrix

It is known that when a linear system $y = Ah$ is corrupted by an additive colored noise, the weighted least squares is implemented, by using noise covariance matrix as weight of normal equations.

In the short-time Fourier analysis, signals are commonly assumed to be approximately stationary within short local segments, as discussed in Section 2.3.6. Consistent with this assumption, the background noise process was modeled as locally stationary within blocks of 256 samples corresponding to the STFT frame length. Thus, the time domain covariances of babble and sparse noise are constructed using formula 2.4.20. The resulting covariance matrices are visualized in Figure 3.1a.



(a) Babble Noise

(b) Sparse Noise

Figure 3.1: Covariance matrix of Babble Noise and Sparse Noise up to lag 256

The estimated babble noise covariance matrix exhibits an oscillatory behavior, indicating periodic correlations in the signal, whereas the sparse noise covariance matrix has a pattern of strong dependence with nearby samples and gradually fades as lag increases. Strong off-diagonal covariance values indicate that neighboring samples are correlated, violating the white noise assumption of ordinary least squares and motivating the use of weighted least squares. Therefore, WLS is implemented in both babble and sparse background noise conditions using covariance matrices Σ_{BN} and Σ_{SN} constructed from the corresponding background noise processes, as described in equation 2.4.18.

Covariance whitening accounts for temporal dependence in the background noise by reducing correlations between residual components. However, whitening does not equalize residual magnitudes, and the whitened residual still contains components with substantially different amplitudes. Since an ℓ_2-norm objective emphasizes large residual values, the IRLS weighting used in AnyRIR remains relevant after covariance whitening. Hence, a covariance-weighted extension of the AnyRIR method is introduced, referred to as Weighted AnyRIR. The proposed modification combines the two mechanisms by applying covariance whitening to the residual before the IRLS weighting step. The modified update of IRLS algorithm in 3.1.4 is written as

$$\mathbf{h}^{(i+1)} = \arg\min_{\mathbf{h}} \|\mathbf{w}^{(i)} \circ LS(\mathbf{y} - X\mathbf{h})\|_2^2, \tag{3.6.1}$$

where

$$\Sigma^{-1} = L^H L.$$

4. Results

4.1 Room 1

The simulation of room 1 includes only the speech and music signals. Since there is no background noise, only the AnyRIR method is performed in this setting.

To visualize the estimated room impulse response, spectrograms are used to showcase the change in frequency content over time. The spectrogram of the estimated RIR by the AnyRIR method is provided in Figure 4.1 alongside the true impulse response for direct comparison.

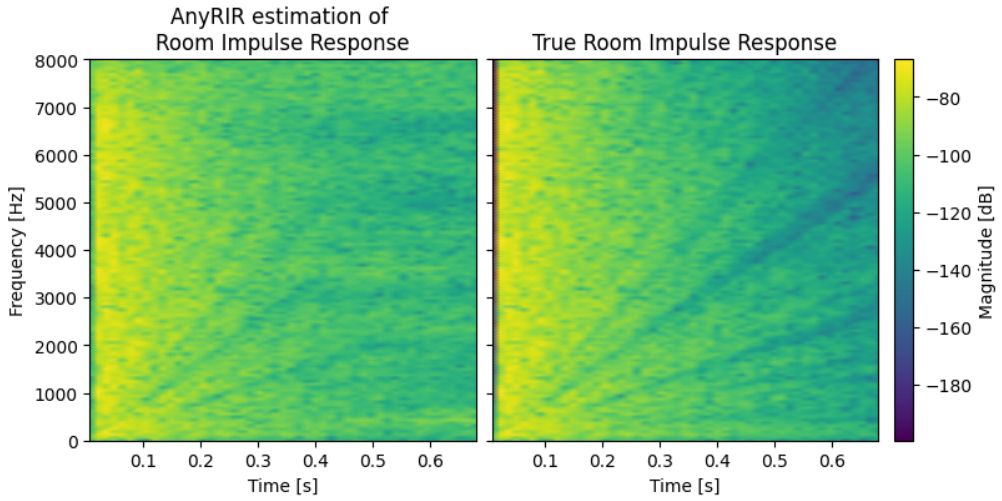


Figure 4.1: Spectrogram of true RIR and AnyRIR estimation of RIR in Room 1

Overall, the estimate appears visually accurate, capturing the direct sound and early reflections within the first 0.2 seconds, followed by a gradual decay. Some discrepancies can be observed in the reverberation tail, where the estimated RIR exhibits higher magnitude values in the 0.5-0.6 second range as compared to the true impulse response. The Numerical accuracy of the RIR estimate is $NMSE \approx 0.0216$.

Additionally, speech restoration is performed by convolving the music signal with the estimated RIR and subtracting the result from the output signal y. The result is used as an argument in the STOI metric function alongside the clean speech signal.

The clean speech signal is simulated by creating a room with equal room parameters and using the same positions of the speech source and the microphone as defined in room 1. The speech signal used in the simulation contains low frequency content and audible pauses. A

visual representation of recovered and true speech is shown in Figure 4.2.

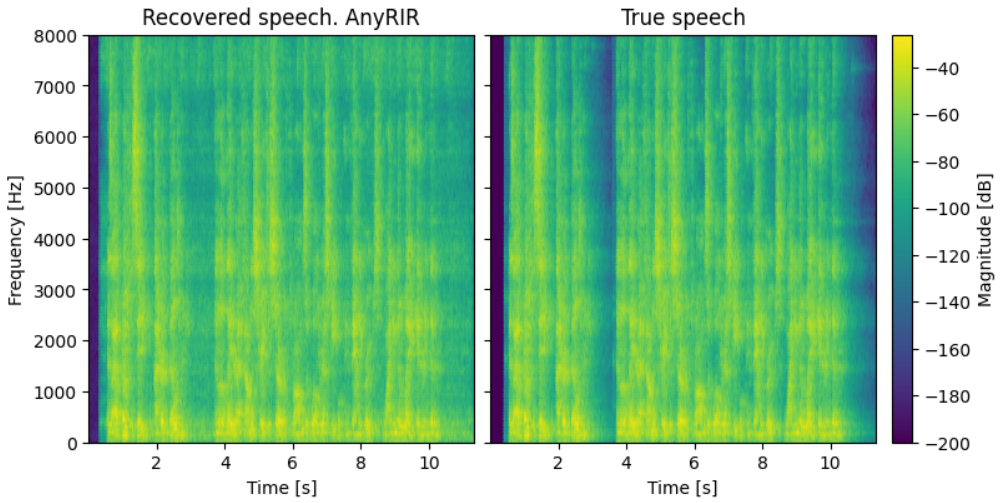


Figure 4.2: Restored speech signal with AnyRIR in Room 1

The resulting plot of the reconstructed speech appears to be more uniform in magnitude. The recovered speech has visibly higher magnitude values in 3-4s range as well as the end of the signal as compared to the true speech signal magnitude, where pauses and silence is. Nevertheless the low frequencies have high magnitude values coinciding with the true speech signal. The discrepancies present do not have a significant impact, resulting in high speech intelligibility score of $STOI \approx 0.98$. This is a substantial improvement compared with the recorded mixture before music removal, where the intelligibility score was only $STOI \approx 0.46$.

4.2 Room 2. Sparse Background Noise

Room 2 includes the speech signal of interest, a music signal, and sparse background noise. Since this simulation involves background noise, the weighted AnyRIR and the weighted least squares methods are evaluated alongside the AnyRIR. The estimated room impulse responses of a single simulation are presented in Figure 4.3.

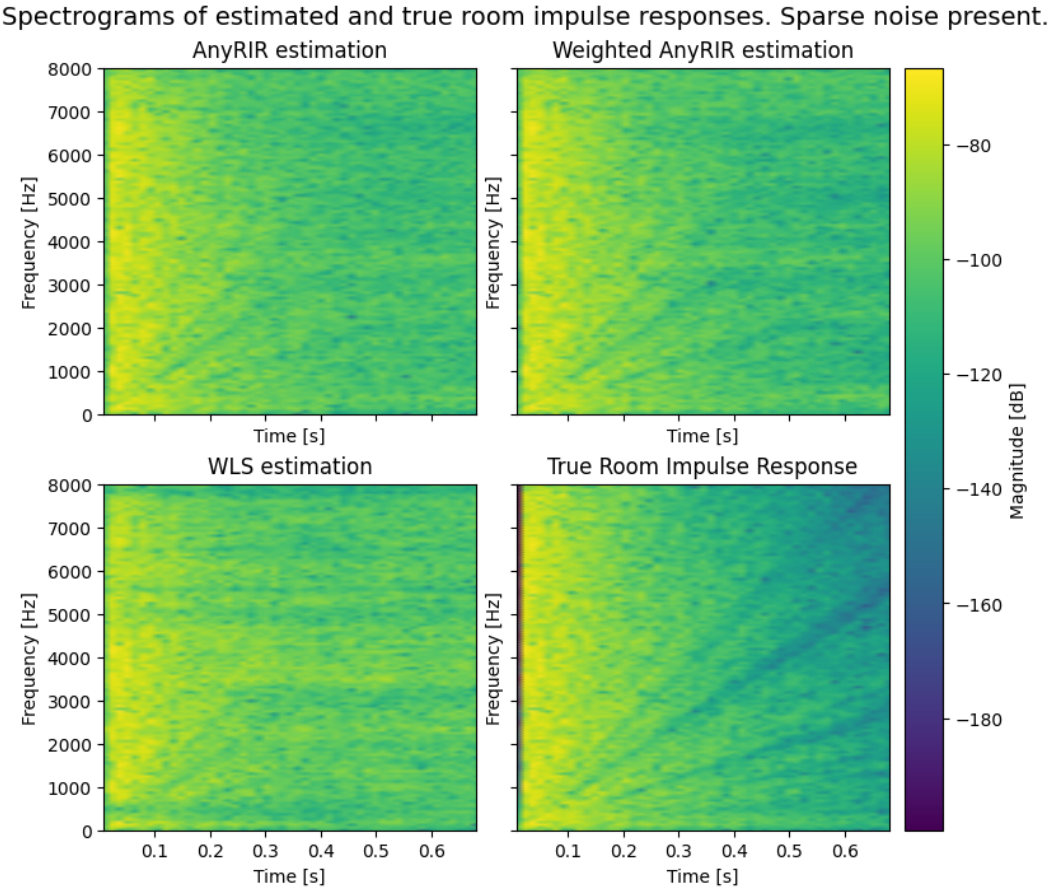


Figure 4.3: AnyRIR, Weighted AnyRIR and WLS room impulse responses estimates. Room with sparse noise present.

All methods produce similar room impulse response estimates, with no visually significant differences. As in the room 1 simulation, the methods capture the direct sound and the early reflections, although the problem of higher magnitude in the reverberation tail persists. What is more, in the 0-0.1 second range, the WLS estimate has lower magnitude for the low frequency components (0-1 kHz) and the highest frequency components (6 kHz - 8 kHz) as compared to the true impulse response.

Spectrograms of the recovered speech signals of all three methods are shown in Figure 4.4. Similar problems are observed as in the room 1 simulation: low magnitude in areas of silence are estimated to be higher, making the signal more uniform in magnitude. This is especially apparent in the weighted least squares estimate, with the resulting estimate lacking visually distinct segments of speech. Additionally, the WLS method seems to preserve the

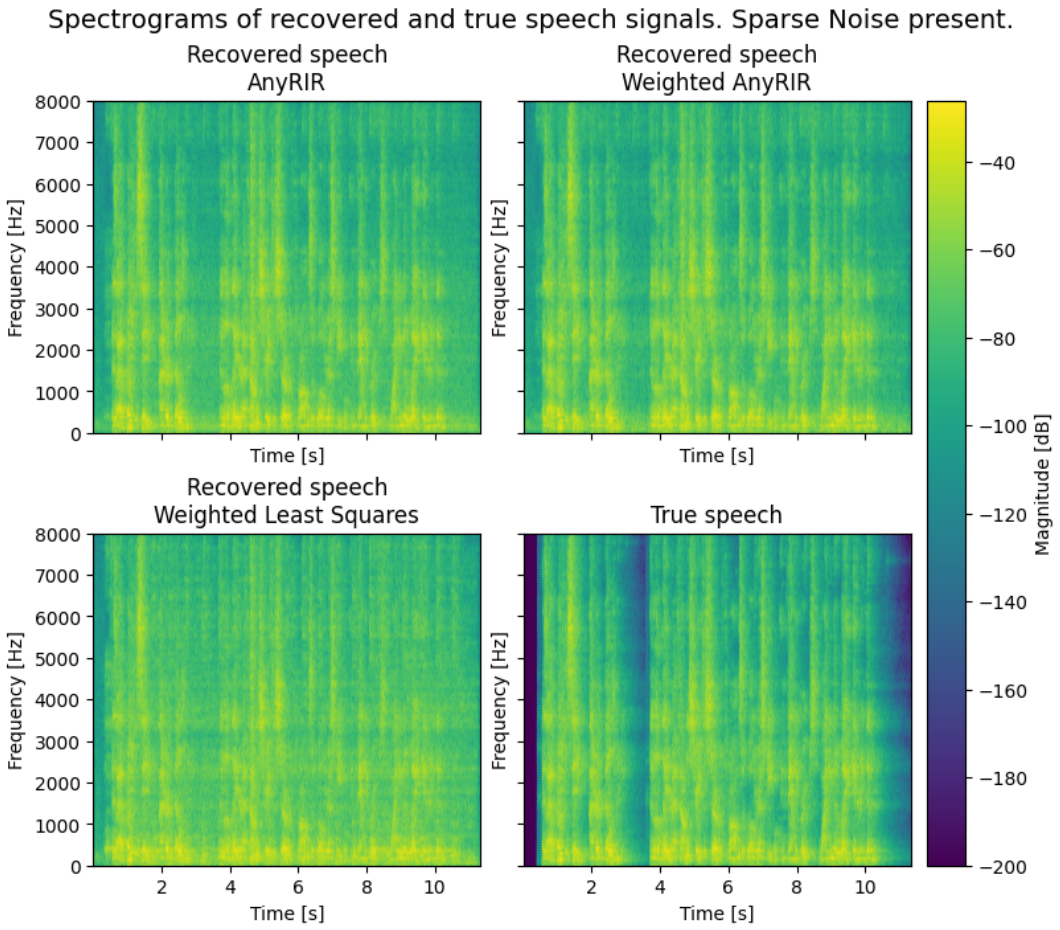


Figure 4.4: Recovered speech signal of AnyRIR, Weighted AnyRIR, and WLS. Room with sparse noise present.

low-frequency components of the speech signal well, whereas higher frequencies (over 1kHz) appear to be dimmed.

The simulation is repeated 10 times with different sparse background noise snippets. The mean STOI of the recordings before removing music signal and RIR is 0.453 with standard deviation 0.005. Alongside NSME of the room impulse response estimates and the recovered speech STOI after removing music signal convolved with estimated RIR, the computational costs as time of estimation in seconds are evaluated. The resulting mean and standard deviation values of all metrics over 10 simulations are as follows:

Table 4.1: Performance comparison of RIR estimation methods in Room 2

Model	**STOI Mean**	**STOI St. dev.**	**NMSE Mean**	**NMSE St. dev.**	**Time Mean**	**Time St. dev.**
AnyRIR	0.906	0.030	0.024	0.001	51.107	3.56
Weighted AnyRIR	0.878	0.040	0.026	0.003	391.552	39.908
WLS	0.792	0.018	0.229	0.017	1.651	0.237

After removing the music convolved with room impulse response, the average STOI metric increased significantly for all applied methods, although not reaching the levels of clean room results of 0.98. The original AnyRIR method achieves the best overall balance between speech intelligibility and RIR estimation accuracy in this setting. The weighted AnyRIR has a slightly lower average STOI of 0.878 and substantially higher computation time. This suggests that the covariance-based weighting does not provide robustness to the sparse noise conditions. The WLS method is computationally much faster, with an average time of only 1.651 seconds. However, this speed comes at a considerable cost. The WLS produces a larger NMSE of 0.229 and a lower STOI of 0.792, indicating both reduced RIR estimation accuracy and lower recovered speech intelligibility.

4.3 Room 3. Babble Noise

Room 3 involves the speech of interest, a music signal, and 20 non-target speakers scattered in a room resulting in babble background noise. As stated previously, for a baseline setting of a room with babble noise present, each amplitude of source signal of babble noise is multiplied by a factor of 0.3, making it a more quiet signal than the music and the speech signal before the room acoustics are applied. The resulting room impulse response estimates using a 0.3 amplitude factor babble noise setting are shown in spectrograms in Figure 4.5.

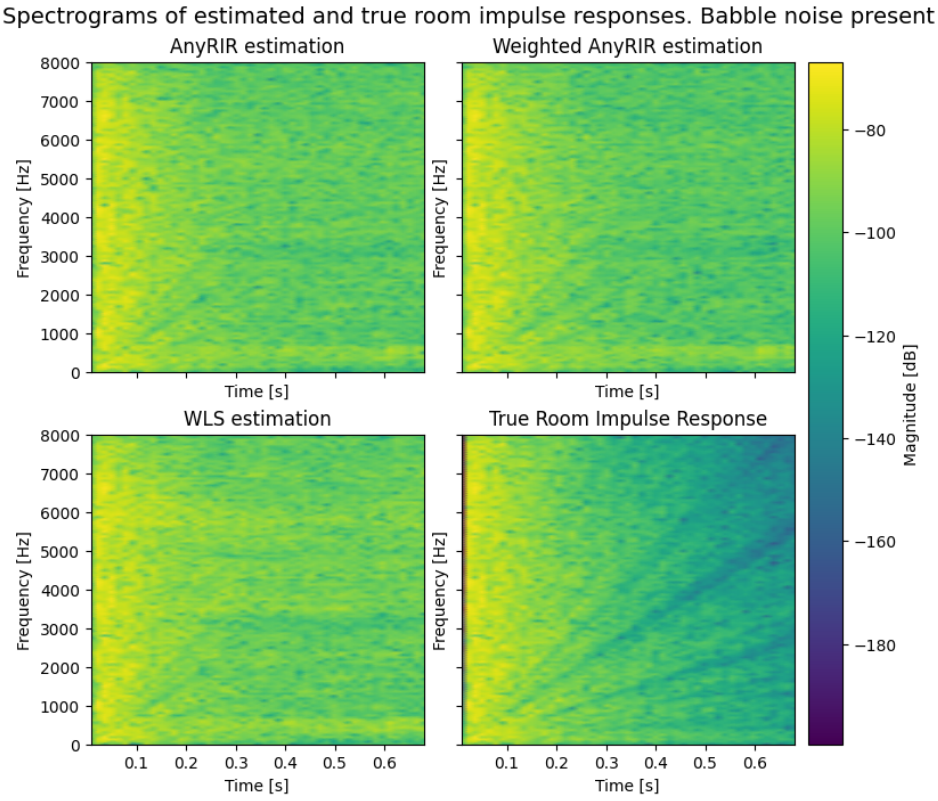


Figure 4.5: AnyRIR, Weighted AnyRIR, and WLS room impulse responses estimates. Room with babble noise present.

Compared to the sparse-noise setting in Section 4.2, the spectrograms obtained in the presence of babble noise appear visually smoother and less structured. In particular, the diagonal reverberation patterns visible in the true room impulse response are less distinguishable in all estimated RIRs. The estimation methods produce spectrograms with more uniform magnitude distributions and reduced visibility of the reverberation patterns.

The recovered speech spectrograms are displayed in Figure 4.6

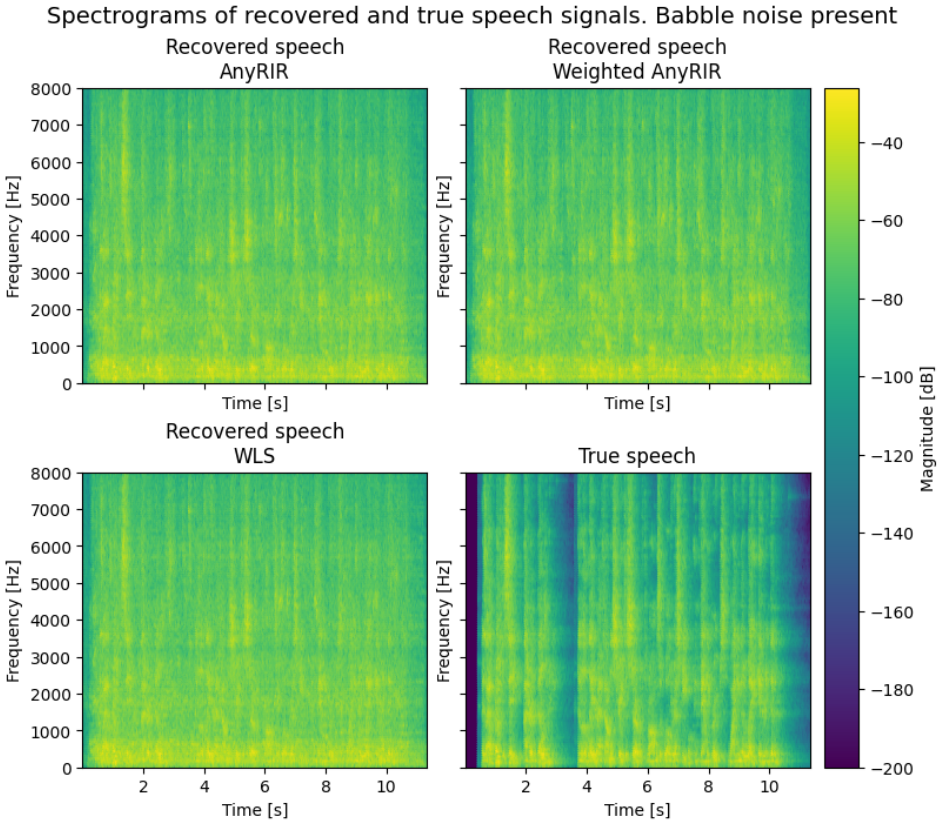
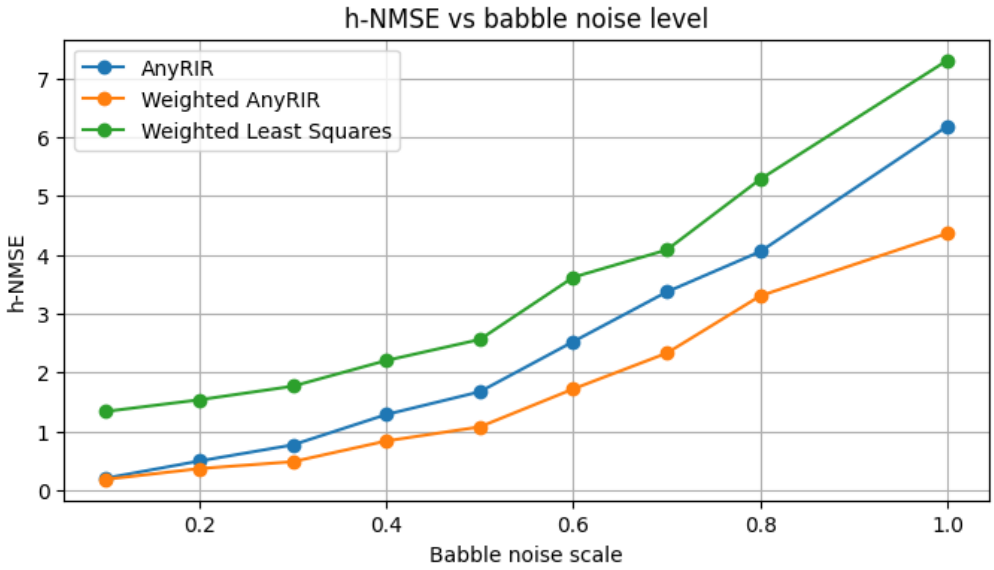


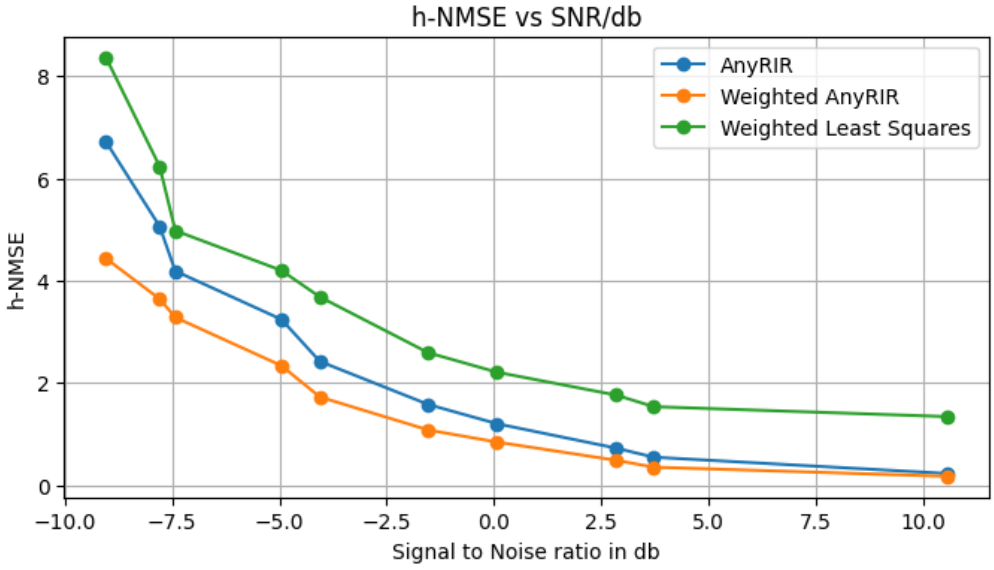
Figure 4.6: Recovered speech signal of AnyRIR, Weighted AnyRIR and WLS. Babble noise.

In the babble-noise setting, all recovered speech spectrograms appear noticeably more uniform as compared to the true speech signal. The pauses and low-magnitude regions visible in the true speech spectrogram become less distinguishable after restoration. Since babble noise consists of multiple overlapping speech signals, it overlaps strongly with the target speech across both time and frequency, making the restoration problem more difficult. As a result, more residual components remain after the subtraction step.

Numerical evaluation of the method performance is done over increasing amplitude of babble noise, raising factor from 0.1 to 1, which corresponds to signal-to-noise ratio (SNR) values decreasing from approximately 13 to -7 decibels. Here SNR means speech signal of interest to babble noise ratio. The results of RIR estimation accuracy are visualized using line graphs (Figure 4.7), where the x axis of the left and right graphs denote increasing amplitude of babble noise sources and increasing signal-to-noise ratio in decibels, respectively, while the y axis denotes the NMSE.



(a) Increasing Babble Noise Amplitude Factor

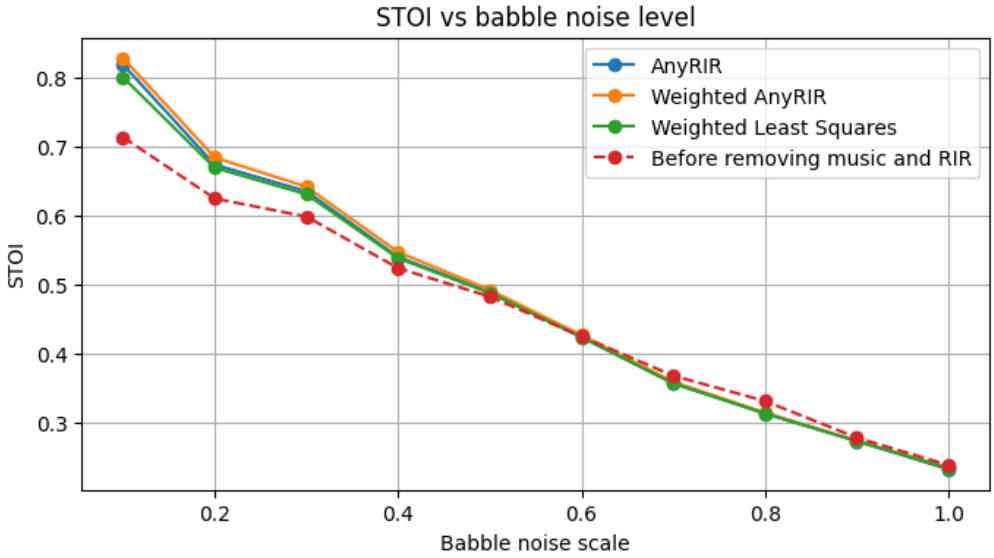


(b) Increasing SNR

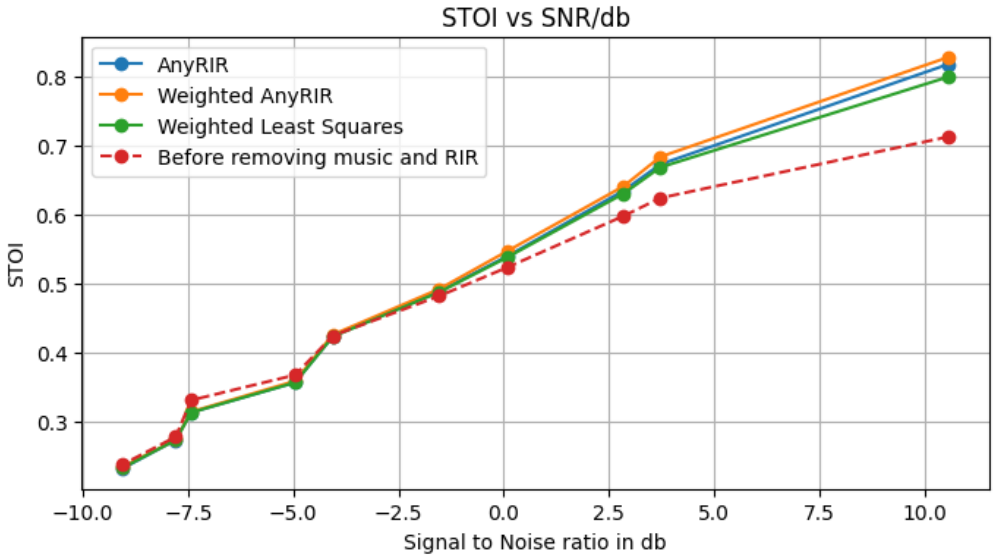
Figure 4.7: AnyRIR, Weighted AnyRIR, and WLS RIR estimation accuracy in NSME versus babble noise level

The estimation error increases for all methods as the babble-noise amplitude factor increases and as SNR becomes smaller. This indicates that stronger background noise makes accurate room impulse response estimation more difficult. Among the evaluated methods, the Weighted AnyRIR consistently achieves the lowest NMSE values across all babble-noise levels, suggesting that the covariance-based weighting improves robustness of the RIR estimation in the presence of babble noise. In contrast, the WLS method produces the largest errors for all tested noise levels.

In contrast, speech intelligibility graphs in Figure 4.8 show small performance differences between the three methods.



(a) Increasing Babble Noise Amplitude Factor

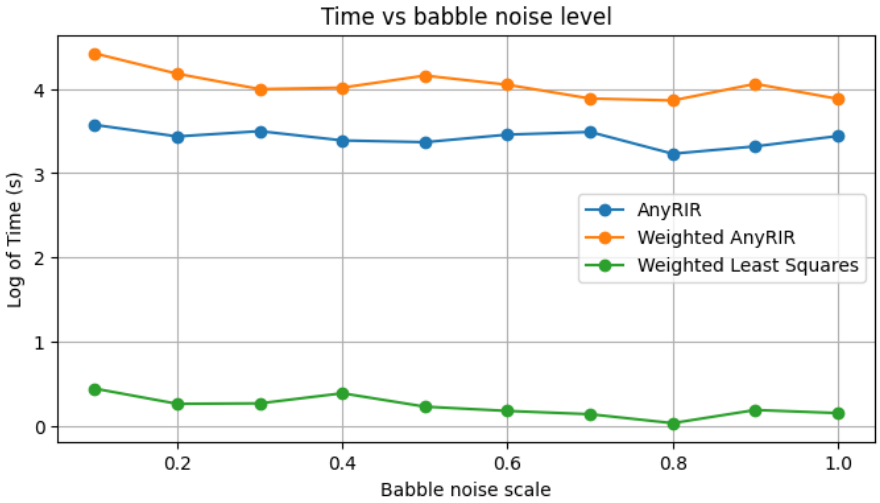


(b) Increasing SNR

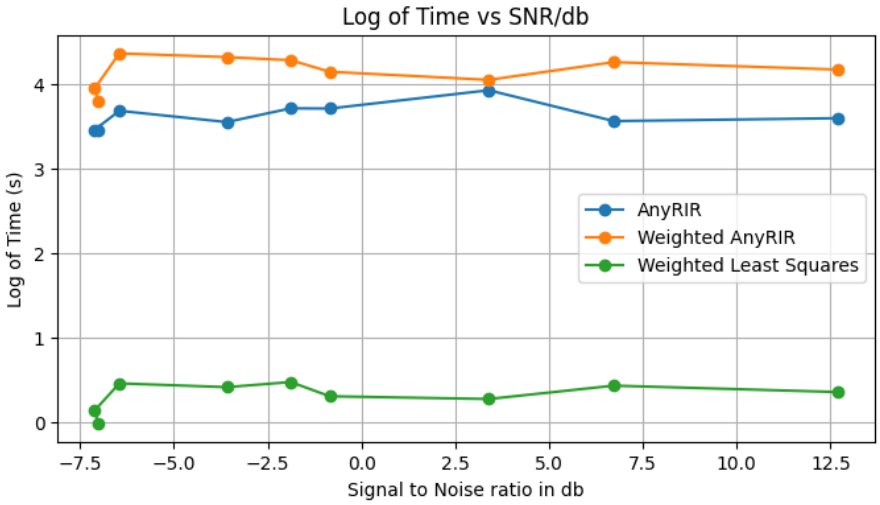
Figure 4.8: AnyRIR, Weighted AnyRIR, and WLS restored speech STOI versus babble noise level

From SNR levels -8 dB to 0 dB removal of music track convolved with estimated room impulse response offers no improvement in speech intelligibility, where background babble noise masks the speech of interest. Only when SNR level is above 0 dB, some improvement in intelligibility can be seen with Weighted AnyRIR method having the highest scores, although the differences between the three methods are marginal.

The computational cost results in Figure 4.9 highlight estimation time difference between the methods.



(a) Increasing Babble Noise Amplitude Factor



(b) Increasing SNR

Figure 4.9: AnyRIR, Weighted AnyRIR, and WLS computational costs versus babble noise level

Weighted AnyRIR is consistently the most computationally expensive approach. AnyRIR also shows high computation time, although the lower than the weighted AnyRIR method. In contrast, WLS remains computationally very efficient across all tested noise conditions, requiring only a few seconds.

5. Conclusions and Future Work

This thesis evaluated room impulse response estimation in acoustic environments with background noise present with the goal of improving estimation accuracy and speech intelligibility after signal recovery. The work focused on investigating music as an excitation signal and sparse background noise interference in room impulse response estimation, following ideas explored in the recently published AnyRIR method. Considering another type of background noise commonly present in cafes and restaurants - babble noise, the AnyRIR method was extended to a weighted AnyRIR formulation using a babble noise covariance matrix as a weighting term. Additionally, taking into account the computationally expensive iterative approach of AnyRIR, a weighted least squares was implemented to assess the trade-off between estimation accuracy and computational time.

In the baseline setting - a room without any background noise, AnyRIR produces an accurate estimate of the room impulse response, resulting in low NMSE and high recovered speech intelligibility. When sparse background noise was added, AnyRIR's performance slightly dropped, although still retaining STOI at a high level. Moreover, it achieved better results than weighted AnyRIR and WLS both in RIR estimation accuracy and recovered speech intelligibility, suggesting that the covariance weighting used in the thesis does not provide robustness when sparse background noise is present. Although difference in AnyRIR and weighted AnyRIR performance is marginal, the latter method required, on average, more than seven times longer estimation time.

The WLS method is substantially faster than both iterative methods, but this came at a cost of significant drop in NMSE and STOI. In a sparse background noise setting, visual inspection showed that WLS underestimated the magnitudes of some early low-frequency and high-frequency components of the RIR. Since these components correspond to the direct sound and early reflection regions of the RIR, their underestimation may have negatively affected speech restoration.

Using babble noise as background noise proved to be more challenging in both RIR estimation and speech unmasking. Babble noise is constructed using overlapping speech signals, meaning it shares time-frequency properties with the target speech. As a result, the recovered speech spectrograms appeared more uniform, low-magnitude regions present in true speech signal became barely distinguishable. Increasing the babble noise amplitude factor caused the RIR estimation error to increase for all methods. Weighted AnyRIR consistently outperformed the other two methods and as the babble noise amplitude increased, the difference between the NMSE values of AnyRIR and weighted AnyRIR increased. This indicates that covariance weighting can improve robustness when the background noise has a stronger continuous structure, such as babble noise. Nevertheless, the improvements in NMSE did not lead to improved STOI, as all methods exhibited a similar rate of STOI degradation with increasing babble noise

amplitude. This implies that speech intelligibility is limited more by the remaining obstructing babble noise rather than accuracy of the RIR estimation.

The computational cost results showed that the weighted AnyRIR method was the most computationally expensive, especially when babble noise level was high. The AnyRIR method was faster than the weighted AnyRIR, but still significantly slower than WLS, which kept estimation time under 5 seconds for all babble noise levels. Thus, although WLS produced worse RIR estimates, it may be useful when dealing with long signals and speech recovery depends heavily on background noise suppression.

The main limitation of this thesis is that experiments were conducted using simulated environments and short audio segments. Applying the methods discussed in practice requires consideration of other factors. For example, the target speaker may move through the room, since a person is unlikely to remain in a fixed position throughout the duration of the dialogue of interest [38]. Additionally, when music is used as an input signal, one needs to consider the problem of accurately identifying and time aligning the music track with recorded signal [3] as well as spectral richness of the music signal. If the music track excites only a limited range of frequencies, the resulting Toeplitz matrix X may become ill-conditioned. Consequently, the RIR estimation problem may become unstable and produce high-variance estimates of the impulse response [10, 21].

The results of babble noise experiment show that even when RIR estimation NMSE is low, speech intelligibility cannot be improved due to residual babble noise present in the signal. Future work could incorporate techniques such as spatial filtering and beamforming, which use the known or estimated location of the target speaker to enhance signals arriving from the target direction while suppressing interfering background noise [2, 25].

In conclusion, in a simulated room with spectrally rich music track playing, the AnyRIR method performs best in sparse background noise settings, providing a balance between accuracy, intelligibility and computational costs. The weighted AnyRIR provides the best RIR estimation accuracy in babble background noise setting, but does not improve speech intelligibility, motivating the use of additional background-noise suppression methods in future work.

Bibliography

[1] C.J. Adcock and N. Meade. "A comparison of two LP solvers and a new IRLS algorithm L1 estimation". In: *Statistical Procedures and Related Topics* Vol. 31 (1997), pp. 119–132.

[2] H. Adel et al. "Beamforming Techniques for Multichannel audio Signal Separation". In: *International Journal of Digital Content Technology and its Applications* 6.20 (2012), pp. 659–667.

[3] A. Alexander, O. Forth, and D. Tunstall. "Music and noise fingerprinting and reference cancellation applied to forensic audio enhancement". In: *Audio Engineering Society Iinternational Conference*. 2012.

[4] F. Antonacci et al. "DiffusionRIR: Room Impulse Response Interpolation using Diffusion Models". In: *Proceedings of the 11th Convention of the European Acoustics Association*. 2025.

[5] M. Arana et al. "Impulse source versus dodecahedral loudspeaker for measuring parameters derived from the impulse response in room acoustics". In: *The Journal of the Acoustical Society of America* 134.1 (2013), pp. 275–284.

[6] T. Bäckström et al. *Introduction to Speech Processing*. 2nd ed. 2022.

[7] H. Barfuss et al. "The LOCATA Challenge: Acoustic Source Localization and Tracking". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 1620–1643.

[8] J. Benesty et al. *Advances in Network and Acoustic Echo Cancellation*. Springer, 2001.

[9] A. Björkman et al. "Room Impulse Response Estimation Using Optimal Transport: Simulation-Informed Inference". In: *Proceedings of the 32nd European Signal Processing Conference (EUSIPCO)*. 2024.

[10] T. Blumensath et al. "Sparse Representations in Audio and Music: from Coding to Source Separation". In: *The Institute of Electrical and Electronics Engineers (IEEE)* 98.6 (2009), pp. 995–1005.

[11] E. C. Boeira and D. Eckhard. "Regularized impulse response estimation for systems with colored output noise". In: *Australian New Zealand Control Conference (ANZCC)*. 2021.

[12] J. Borish. "Extension of the image model to arbitrary polyhedra". In: *The Journal of the Acoustical Society of America* 75.6 (1984), pp. 1827–1836.

[13] Ceiling Tiles UK. *A Guide to Standard Ceiling Height Requirements (UK)*. URL: https://www.ceilingtilesuk.co.uk/standard-ceiling-height-uk/ (visited on 05/02/2026).

[14] G. Chen et al. "LibriSpeech: an ASR corpus based on public domain audio books". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015.

[15] C. Lynge Christensen, G. Koutsouris, and A. Richard. "Sine Sweep Optimization for Room Impulse Response Measurements". In: *Forum Acusticum*. 2020.

[16] J. W. Cooley and J. W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". In: *Mathematics of Computation* 19.90 (1964), pp. 297–301.

[17] J. Ding et al., eds. *Modelling, Uncertainty and Data for Engineers (MUDE) Textbook*. Accessed: 2026-05-02. 2025. URL: https://mude.citg.tudelft.nl/book.

[18] R. A. Dobre, C. Negrescu, and D. Stanomir. "Development and testing of an audio forensic software for enhancing speech signals masked by loud music". In: *Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies*. 2016.

[19] R. A. Dobre et al. "A Method for Recovering Speech Signals Heavily Masked by Music Based on the Affine Projection Algorithm". In: *International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking*. 2018.

[20] D. P. W. Ellis et al. "Signal Processing for Music Analysis". In: *IEEE Journal of Selected Topics in Signal Processing* 5.6 (2011), pp. 1088–1110.

[21] F. Elvander, M. Jälmby, and T. van Waterschoot. "Low-Rank Room Impulse Response Estimation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), pp. 957–969.

[22] L. Euler. *Introduction to analysis of the infinite*. M.M. Bousquet, 1748.

[23] D. Ching-Lund Fong and M. Saunders. "LSMR: And Iterative Algorithm for Sparse Least-Squares Problems". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2950–2971.

[24] K. Furuya, Y. Kaneda, and K. Kiyohara. "Sweeping echoes perceived in a regularly shaped reverberation room". In: *The Journal of the Acoustical Society of America* 111.2 (2002), pp. 925–930.

[25] G. Del Galdo et al. "Spatial filtering using directional audio coding parameters". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2009.

[26] N. D. Gaubitch and P. A. Naylor. *Speech Dereverberation*. Springer, 2010.

[27] R. M. Gray. *Toeplitz and circulant matrices: a review*. Now Publishers Inc., 2006.

[28] P. J. Green. "Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some Robust and Resistant Alternatives". In: *Journal of the Royal Statistical Society Series B* 46.2 (1984), pp. 149–192.

[29] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, 2015.

[30] R. C. Hendriks et al. "A Short-Time Objective Intelligibility Measure for Time-Frequency Weighted Noisy Speech". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010.

[31] J.Allen and D.Berkley. "Image method for effciently simulating small-room acoustics". In: *The Journal of the Acoustical Society of America* 65.4 (1979), pp. 943–950.

[32] S. M. Jung and P. Park. “Normalised least-mean-square algorithm for adaptive filtering of impulsive measurement noises and noisy inputs”. In: *Electronics Letters* 49.20 (2013), pp. 1270–1272.

[33] V. Kalantzis, M. S. Squillante, and C. Wah Wu. “Stable Iterative Solvers for Ill-conditioned Linear Systems”. In: *IEEE Conference on High Performance Extreme Computing*. 2025.

[34] J. Kim and J. Lavaei. “Huber-based Robust System Identification with Near-Optimal Guarantees Across Independent and Adversarial Regimes”. In: *arXiv preprint arXiv:2603.27586* (2025).

[35] H. Kuttruff. *Room Acoustics*. CRC Press, 2016.

[36] K. Y. Lee et al. “AnyRIR: Robust Non-Intrusive Room Impulse Response Estimation in the Wild”. In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2026.

[37] M. Lee, P. Park, and T. Park. “Bias-Compensated Normalized Least Mean Fourth Algorithm for Adaptive Filtering of Impulsive Measurement Noises and Noisy Inputs”. In: *Asian Control Conference (ASCC)*. 2019.

[38] S. Lin. “Reverberation-Robust Localization of Speakers Using Distinct Speech Onsets and Multichannel Cross Correlations”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.11 (2018), pp. 2098–2111.

[39] G. Lindgren, H. Rootzén, and M. Sandsten. *Stationary Stochastic Processes for Scientist and Engineers*. Chapman and Hall, 2013.

[40] L. Ljung. “Linear Time-Invariant Models of Non-linear Time-varying Systems”. In: *The European Journal of Control* 7.2-3 (2001), pp. 203–219.

[41] R. A. Maronna. *Robust statistics theory and methods*. Wiley Online Library Online Books, 2019.

[42] I. Mateljan and K. Ugrinovic. “Comparison of Different Impulse Response Measurement Techniques”. In: *Journal of the Audio Engineering Society* 50.4 (2002), pp. 249–262.

[43] MathWorks. *Measuring Speech Intelligibility and Perceived Audio Quality with STOI and ViSQOL*. URL: https://se.mathworks.com/help/audio/ug/measure-speech-intelligibility-and-perceived-audio-quality-with-stoi-and-visqol.html (visited on 05/02/2026).

[44] Plan7Architect. *How Big Should a Café Be? – Size Guide*. URL: https://plan7architect.com/how-big-should-a-cafe-be-size-guide-ai2/ (visited on 05/02/2026).

[45] A. Pras and C. Guastavino. “Sampling Rate Discrimination: 44.1 kHz vs. 88.2 kHz”. In: *Proceedings of the 128 th convention of the Audio Engineering Society*. 2010.

[46] W. C. Sabine. *Collected Papers on Acoustics*. Harvard University Press, 1922.

[47] L. Savioja and U. P. Svensson. “Overview of geometrical room acoustic modeling techniques”. In: *The Journal of the Acoustical Society of America* 138.2 (2015), pp. 708–730.

[48] J. Semmlow. *Circuits, Signals and Systems for Bioengineers: A MATLAB-Based Introduction, Third Edition*. Academic Press, 2018.

[49] Shazam. *Music discovery, charts and song lyrics*. 2025. URL: https://www.shazam.com/ (visited on 05/02/2026).

[50] V. Velardo. *Sound and Waveforms*. Teaching slides accessed Januray 2026. 2020. URL: https://github.com/musikalkemist/AudioSignalProcessingForML/tree/master.

[51] M. Vorländer. "Simulation of the transient and steady-state sound propagation in rooms using a new combined raytracing/image-source algorithm". In: *The Journal of the Acoustical Society of America* 86.1 (1989), pp. 172–178.

[52] E. Walshaw. *Average Heights / Dimensions of Person Sitting*. URL: https://www.firstinarchitecture.co.uk/average-heights-dimensions-of-person-sitting/ (visited on 05/02/2026).

[53] T. Wempe. *Into Sound*. Brave new books, 2018.

[54] Woodcoustics. *Recommended Reverberation Times*. URL: https://share.google/lv4dK0Z5nv13xuspM (visited on 05/02/2026).

A. LSMR Algorithm

The goal of the LSMR algorithm is to solve the least squares problem

$$\min_{\mathbf{h}} ||A\mathbf{h} - \mathbf{b}||_2^2,$$

while iteratively minimizing $||A^\top \mathbf{r}||_2$, where

$$\mathbf{r} = A\mathbf{h} - \mathbf{b}.$$

Let $\mathbf{b} = \beta_1 \mathbf{u_1}$ ($\beta_1 = ||\mathbf{b}||_2$, $\mathbf{u_1} = \mathbf{b}/\beta_1$) and $A^\top \mathbf{u_1} = \alpha_1 \mathbf{v_1}$, where β_1 and α_1 are some scalars and $\mathbf{u_1}$, $\mathbf{v_1}$ - orthonormal vectors.

For k = 1, 2, ...

$$\beta_{k+1}\mathbf{u_{k+1}} = A\mathbf{v_k} - \alpha_k \mathbf{u_k}$$

$$\alpha_{k+1}\mathbf{v_{k+1}} = A^\top \mathbf{u_{k+1}} - \beta_{k+1}\mathbf{v_k}$$

After k steps the following matrix expressions are obtained: $AV_k = U_{k+1}B_k$ and $A^\top U_{k+1} = V_{k+1}L_{k+1}^\top$, where

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}, \qquad L_{k+1} = \begin{pmatrix} B_k & \alpha_{k+1}\mathbf{e_{k+1}} \end{pmatrix}.$$

Then, the objective function becomes:

$$\min_{\mathbf{y_k}} \|A^\top \mathbf{r_k}\| = \min_{\mathbf{y_k}} \left\| \bar{\beta}_1 \mathbf{e_1} - \begin{pmatrix} B_k^\top B_k \\ \bar{\beta}_{k+1}\mathbf{e_k}^\top \end{pmatrix} \mathbf{y_k} \right\|,$$

where $\bar{\beta}_{k+1} = \alpha_k \beta_k$. Then, QR factorization is applied:

$$Q_{k+1}B_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \qquad R_k = \begin{pmatrix} \rho_1 & \theta_2 & & \\ & \rho_2 & \ddots & \\ & & \ddots & \theta_k \\ & & & \rho_k \end{pmatrix}$$

allowing the problem to be rewritten as:

$$\min_{\mathbf{y_k}} ||A^\top \mathbf{r_k}||_2 = \min_{\mathbf{t_k}} \left\| \bar{\beta}_1 \mathbf{e_1} - \begin{pmatrix} R_k^\top \\ \varphi_k \mathbf{e_k}^\top \end{pmatrix} \mathbf{t_k} \right\|_2,$$

where $\mathbf{h_k} = V_k \mathbf{y_k}$, $\mathbf{t_k} = R_k \mathbf{y_k}$, $R_k^\top \mathbf{q_k} = \bar{\beta}_{k+1} \mathbf{e_k}$, $\mathbf{q_k} = \varphi_k \mathbf{e_k}$.

Performing QR factorization on the resulting matrix:

$$\begin{pmatrix} R_k^\top & \bar{\beta}_1 \mathbf{e_1} \\ \varphi_k \mathbf{e_k}^\top & 0 \end{pmatrix}$$

yields the QR factorization:

$$\bar{Q}_{k+1} \begin{pmatrix} R_k^\top & \bar{\beta}_1 \mathbf{e_1} \\ \varphi_k \mathbf{e_k}^\top & 0 \end{pmatrix} = \begin{pmatrix} \bar{R}_k & \mathbf{z_k} \\ 0 & \zeta_{k+1} \end{pmatrix}, \qquad \bar{R}_k = \begin{pmatrix} \bar{\rho}_1 & \bar{\theta}_2 & & \\ & \bar{\rho}_2 & \ddots & \\ & & \ddots & \bar{\theta}_k \\ & & & \bar{\rho}_k \end{pmatrix},$$

where $\bar{R}_k \mathbf{t_k} = \mathbf{z_k}$, and

$$\mathbf{z_k} = \begin{pmatrix} \zeta_1 & \zeta_2 & \cdots & \zeta_k \end{pmatrix}^\top.$$

Let W_k and $\bar{W}_k$ be computed from $R_k^\top W_k^\top = V_k^\top$ and $\bar{R}_k^\top \bar{W}_k^\top = W_k^\top$. Then, $\mathbf{h_k}$ can be expressed as:

$$\mathbf{h_k} = \mathbf{h_{k-1}} + \zeta_k \bar{\mathbf{w}}_\mathbf{k}.$$

Hence, when k increases to $k+1$, all quantities remain the same except for one additional term. The QR factorizations, together with R_k and $\bar{R}_k$, are updated using plane rotations.

$$P_l = \begin{pmatrix} I_{l-1} & & & \\ & c_l & s_l & \\ & -s_l & c_l & \\ & & & I_{k-l-1} \end{pmatrix}$$

and

$$Q_{k+1} = P_k \cdots P_2 P_1,$$

then

$$Q_{k+1} B_{k+1} = Q_{k+1} \begin{pmatrix} B_k & \alpha_{k+1} \mathbf{e_{k+1}} \\ & \beta_{k+2} \end{pmatrix} = \begin{pmatrix} R_k & \theta_{k+1} \mathbf{e_k} \\ 0 & \bar{\alpha}_{k+1} \\ 0 & \beta_{k+2} \end{pmatrix},$$

$$Q_{k+2} B_{k+1} = P_{k+1} \begin{pmatrix} R_k & \theta_{k+1} \mathbf{e_k} \\ 0 & \bar{\alpha}_{k+1} \\ 0 & \beta_{k+2} \end{pmatrix} = \begin{pmatrix} R_k & \theta_{k+1} \mathbf{e_k} \\ 0 & \rho_{k+1} \\ 0 & 0 \end{pmatrix}.$$

where

$$\theta_{k+1} = s_k \alpha_{k+1}, \qquad \bar{\alpha}_{k+1} = c_k \alpha_{k+1},$$
$$c_k \bar{\alpha}_k + s_k \beta_{k+1} = \rho_k, \qquad -s_k \bar{\alpha}_k + c_k \beta_{k+1} = 0.$$

If $\bar{Q}_{k+1} = \bar{P}_k \ldots \bar{P}_2 \bar{P}_1$, the second QR factorization update becomes:

$$\bar{Q}_{k+2} \begin{pmatrix} R_{k+1}^\top \\ \theta_{k+2} \mathbf{e_{k+1}}^\top \end{pmatrix} = \bar{P}_{k+1} \begin{pmatrix} \bar{R}_k & \bar{\theta}_{k+1} \mathbf{e_k} \\ 0 & \bar{c}_k \rho_{k+1} \\ 0 & \theta_{k+2} \end{pmatrix} = \begin{pmatrix} \bar{R}_k & \bar{\theta}_{k+1} \mathbf{e_k} \\ 0 & \bar{\rho}_{k+1} \\ 0 & 0 \end{pmatrix}.$$

Here the following holds:

$$\bar{c}_k\bar{c}_{k-1}\rho_k + \bar{s}_k\theta_{k+1} = \bar{\rho}_k, \qquad \bar{s}_k\rho_{k+1} = \bar{\theta}_{k+1}, \qquad \bar{c}_k\ \bar{\zeta}_k = \zeta_k,$$

$$-\bar{s}_k\bar{c}_{k-1}\rho_k + \bar{c}_k\theta_{k+1} = 0, \qquad -\bar{s}_k\ \bar{\zeta}_k = \bar{\zeta}_{k+1}.$$

Considering the last row of the matrix equation $R_{k+1}^\top W_{k+1}^\top = V_{k+1}^\top$ and the last row of $\bar{R}_{k+1}^\top \bar{W}_{k+1}^\top = W_{k+1}^\top$, the updating equations are obtained:

$$\theta_{k+1}\mathbf{w_k}^\top + \rho_{k+1}\mathbf{w_{k+1}}^\top = \mathbf{v_{k+1}}^\top,$$

$$\bar{\theta}_{k+1}\bar{\mathbf{w}}_\mathbf{k}^\top + \bar{\rho}_{k+1}\bar{\mathbf{w}}_\mathbf{k+1}^\top = \mathbf{w_{k+1}}^\top.$$

Substituting $\mathbf{d_k} = \rho_k\mathbf{w_k}$ and $\bar{\mathbf{d}}_\mathbf{k} = \rho_k\bar{\rho}_k\bar{\mathbf{w}}_\mathbf{k}$, the update of $\mathbf{h_k}$ becomes:

$$\bar{\mathbf{d}}_\mathbf{k} = \mathbf{d_k} - \left(\bar{\theta}_k\rho_k/(\rho_{k-1}\bar{\rho}_{k-1})\right)\bar{\mathbf{d}}_\mathbf{k-1},$$

$$\mathbf{h_k} = \mathbf{h_{k-1}} + \left(\zeta_k/(\rho_k\bar{\rho}_k)\right)\bar{\mathbf{d}}_\mathbf{k},$$

$$\mathbf{d_{k+1}} = \mathbf{v_{k+1}} - \theta_{k+1}/\rho_k)\mathbf{d_k}.$$