Multichannel Wiener Filtering for Dual-Microphone Speech Processing in a Car Environment

by
Juhi Khalid

Thesis Submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Applied Sciences in Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

With advancements in automotive electronics and sensors, the sound pick-up using multiple microphones has become feasible for hands-free telephony and voice command in-car applications. Adding a second microphone so that both driver and passenger can have dedicated microphones can help in better sound pickup. However, challenges remain in effectively processing multiple microphone signals due to bandwidth or processing limitations. If a single resulting microphone signal is to be transmitted or processed, adding (mixing) the microphone signals or switching between them can be efficient and cost effective solutions. However, switching between microphone signals can cause sudden changes in the acoustic path between loudspeakers and microphones, leading to performance decay in the acoustic echo canceling module. And mixing the two microphone signals can lead to echoes and notch filtering effects (spectral holes) in the resulting mixed signal. This work explores the use of the Multichannel Wiener Filter algorithm with a two-microphone in-car system to enhance speech quality for driver and passenger voice, i.e., more specifically to mitigate notch-filtering effects caused by echoes and improve background noise reduction. We evaluate its performance under various noise conditions using modern objective metrics such as the Deep Noise Suppression Mean Opinion Score. The effect of head movements of driver/passenger is also investigated. The proposed method is shown to provide significant improvements over a simple mixing of microphone signals.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Martin Bouchard, for his unwavering support, insightful guidance, and thoughtful suggestions throughout my research journey. His passion for excellence and dedication to his field have been a constant source of inspiration and motivation. I feel truly fortunate to have had the opportunity to work under his supervision, and I sincerely thank him for the invaluable knowledge and mentorship he has provided. This thesis would not have been possible without his continuous encouragement and trust in my abilities.

I am deeply grateful to Michael Tetelbaum and Hala As'ad for giving me the opportunity to work with you on the in-car communication system, which immensely helped me develop and complete my thesis.

I would also like to extend my heartfelt thanks to the members and staff of the Faculty of Engineering, whose knowledge, assistance, and support have contributed significantly to my academic journey.

I am profoundly grateful to my parents for their unconditional love, sacrifices, and endless encouragement. Their belief in my abilities has been a constant source of strength. I am equally thankful to my sisters and in-laws for their support and motivation throughout this journey.

To my loving husband, Hisham Veeran, thank you for your patience, understanding, and unwavering support during the challenging moments. Your encouragement has helped me stay focused and determined.

Lastly, I dedicate this work to my children, the light of my life, Izna and Kian.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AEC	Acoustic Echo Canceller
AECMOS	Acoustic Echo Cancellation Mean Opinion Score
ANC	Adaptive Noice Canceller
COLA	Constant Overlap Add
DNSMOS	Deep Noise Suppression Mean Opinion Score
DNSMOS BAK	Deep Noise Suppression Mean Opinion Score for Background Noise Component
DNSMOS SIG	Deep Noise Suppression Mean Opinion Score for Speech Signal Component
IR	Impulse Response
ITU	International Telecommunication Union
MOS	Mean Opinion Score
MWF	Multichannel Wiener Filter
MWF-AC	Multichannel Wiener Filter with Continuous Adaptation
MWF-AS	Multichannel Wiener Filter with Adaptation Stopped
OLA	Overlap and Add
PESQ	Perceptual Evaluation of Speech
RIR	Room Impulse Response
SIR	Signal to Interference Ratio
SNR	Signal to Noise Ratio
STFT	Short-Time Fourier Transform
VAD	Voice Activity Detection

Chapter 1 Introduction

1.1 Motivation

With advancements in automotive electronics and sensors, the sound pick-up using multiple microphones has become feasible for hands-free telephony and voice command in-car applications. For example, in addition to a primary microphone mostly dedicated to sound-pick up from the driver, to improve sound pick-up for the front seat passenger a secondary microphone can be added closer to the passenger (or a secondary differential microphone can be steered towards the passenger).

When more than one microphone signal is used as the input for processing, we have multichannel filtering. This can be viewed in different ways. Beamforming with distributed microphones in the cars is a possibility. Normally, beamforming is based on knowing a propagation model of sound coming from the direction of a target source (e.g. driver), as well as knowing the multichannel correlation matrix of the signals to be minimized (e.g. background noise and interferers). Instead of relying on a propagation model and direction of arrival for a target (e.g. driver), the Multichannel Wiener filter (MWF) formulation is based on estimating correlations between the different sensor signals and the target speaker signal to extract. The resulting correlations are used to build a multichannel correlation matrix and multichannel correlation vector, from which the MWF solution is computed. The correlations can be estimated if it is possible to detect when the different sources are active, separately or jointly. This can be done by comparing the signal powers at the mic or using the correlation between mics and past signals. For some applications this can be challenging and this is the main drawback of the MWF method, but the method does not require any propagation model or direction of arrival estimation, unlike beamformers.

Challenges remain in effectively processing multiple microphone signals due to bandwidth or processing limitations. As such, for telephony applications or for voice command, it may still not be possible to fully process or transmit the two microphone signals separately. Therefore, some solution such as mixing (adding) the microphone signals or switching between the two microphone signals needs to be developed. But each of these solutions comes with drawbacks.

Switching between the microphone signals can be done based on which speech source is active (driver or passenger). The main drawback of this approach is that it creates sudden changes in the

acoustic echo path between loudspeakers and the selected microphone to be used (primary or secondary), which will affect the performance of acoustic echo cancellation (AEC) systems. AEC systems are a requirement to prevent that a "far-end" voice played in car loudspeakers is fed back to the microphones before transmission or processing. Acoustic echo cancellation can also be used during media playback, to help with the voice command functionality by removing the media component picked up by the microphone(s). A second drawback of microphone switching is its inability to properly deal with the (less frequent) case of driver and passenger speaking simultaneously.

Mixing the microphone signals has its own drawbacks. Since each speech source reaches the summed microphone signal through two acoustic paths (one for each microphone), the sum of two signals with different delays can lead to "notch filtering" effects in the spectrum of the resulting signal, i.e., distortion because some frequencies are attenuated at the spectral notches. The background noise power is also increased (typically doubled) when two signals are added in mixing.

In this thesis, we revisit the multichannel Wiener filter (MWF) adaptive filtering algorithm in the specific context of a two-microphone system for the in-car voice pick-up of two speakers (a driver and front seat passenger). While the use of Wiener filtering for multichannel speech enhancement has been considered before in car environments [1], in this work we are considering the use of MWF for tasks such as mitigation of notch-filtering effect and source extraction in the case of simultaneous driver and passenger talk, in addition to assessing the performance of MWF for noise reduction in different noise types and conditions. We also make use of recently introduced metrics such as AECMOS and DNSMOS. AECMOS and DNSMOS are metrics recently developed by Microsoft for evaluating echo suppression and noise suppression, respectively.

1.2 Thesis contributions and publication

The main contributions of this thesis are:

- for an in-car environment, we propose the use of the multichannel Wiener filter for the (possibly simultaneous) tasks of notch-filtering effect mitigation (echo removal) in the sum of microphone signals and multichannel background noise reduction, with both tasks

performed with either one or multiple active speakers. Since the MWF extracts the speech sources before mixing/adding them, it can also be used for interference cancellation or source extraction (i.e., cancelling of competing talkers);

- through simulations of a case study, we demonstrate experimentally the drop of performance that the simple method of microphone switching can bring to an echo canceling system;
- through simulations of a case study, we demonstrate experimentally the notch filtering effect caused by the simple method of microphone signal mixing;
- through simulations of a case study, we demonstrate the benefit of having a dedicated (secondary) microphone for the front passenger.

A 6-page conference paper publication with title "Improved in-car sound pick-up using multichannel Wiener filter" has been finalized and will be submitted to a conference once approval is received from the company collaborating on this research (i.e., after clearance following an invention disclosure submission). The paper will also be published in arXiv.

1.3 Thesis organization

A brief description of the remaining chapters of the thesis is given here.

Chapter 2 provides a short literature review of material relevant to the thesis topic.

Chapter 3 discusses the experimental setup for the car environment, including the physical components and impulse responses. It lays down the framework on which all the tests performed and results obtained are valid.

Chapter 4 goes through the different AEC setups that can be used in the presence of more than one microphone (with two microphones considered in this work) and how they impact the speech quality. It describes switching and mixing the signals in the time domain and also introduces different metrics for evaluating AEC performance. It also compares the cases of one and two microphone systems through performance evaluation and puts forth the justification for having two microphones instead of one.

Chapter 5 introduces the mathematical foundation on which the Multichannel Wiener Filter is designed for the system considered. It goes into the implementation of the designed MWF.

Chapter 6 explains how notch filtering is removed with the assistance of MWF and compares its performance for different settings.

Chapter 7 dives into the performance of adaptive MWF in the presence of different types of noise.

Chapter 8 introduces a case of head movement of the driver/passenger into the system and evaluates its impact.

Chapter 9 concludes the thesis and suggests some ways in which this work could be extended in the future.

Chapter 2 Literature Review

Unlike simple rooms in some communication systems, achieving high-quality speech signals in a moving vehicle is challenging due to the complex and non-stationary nature of in-car noise. Therefore, enhancing speech in automotive environments is crucial for hands-free communication, voice-controlled systems, and in-car passenger interactions. Here, we review various approaches to overcome these challenges, focusing on microphone arrays, beamforming techniques, and noise suppression algorithms.

2.1 Wiener filtering in signal enhancement

Wiener filtering is a powerful technique for noise removal in speech processing, known for its optimal performance in minimizing the mean square error between the original and estimated signals. It is quite effective in suppressing noise while preserving speech quality and is extensively used in many applications such as telecommunications, hearing aids, and voice-controlled systems. It can be used as the main noise removal step [1] or as an additional step in the entire noise removal process [2].

Generally, in a car cabin environment noise is additive and slowly varying, which allows us to compute noise characteristics when speech is absent. Even if the assumption about the noise is not valid, partial noise reduction is still possible. Even though in this work we delve into noise reduction in the case of multiple sensors, single-channel noise suppression still holds great importance due to its wide variety of applications. There is always a trade-off between noise suppression and speech degradation when it comes to noise removal in speech processing. In [1], a single Wiener filter is used as a basis to find the relation between the two, which shows that the amount of noise attenuation is, in general, proportional to the amount of speech degradation. This calls for something more than SNR gain or similar metrics to gauge the effect of noise removal algorithms.

A Wiener filter can be implemented in the time and frequency domains. Even though time domain filtering is effective, most cases implement it in the frequency domain since it helps with easier computation. This paved the way for using two-dimensional Fourier Transform to make a two-dimensional Wiener filter [3], along with a one-dimensional Wiener filter to make a hybrid filter

[4] which exploits the 2D speech and noise features in a two-dimensional spectrogram, considering correlations between the different time frames. This was shown to be more effective than spectral subtraction or a one-dimensional Wiener filter.

Adaptive filtering plays a critical role in noise reduction due to its ability to handle a wide range of input types, including deterministic or stochastic signals that may be stationary or time-varying. Unlike fixed filters, adaptive filters can adjust their parameters in real time based on the incoming data, making them highly effective in dynamic environments where noise characteristics change over time. Theoretical foundations, such as Wiener solutions, demonstrate that adaptive filters can asymptotically approach optimal performance for stationary stochastic inputs, achieving high output signal-to-noise ratios without requiring prior knowledge of the signal or noise. When the reference input is free of the desired signal, and certain conditions are met, adaptive filtering can eliminate noise from the primary input without distorting the signal [5], [6]. Adaptive Wiener filter also performs well at SNRs 5-10dB as compared to other noise reduction methods such as beamforming, adaptive line enhancer, spectral subtraction, and gamma tone filters [7].

2.2 Wiener filtering in automobile audio signal processing

Wiener filters are used in different ways in automobile audio systems, whether it be various configurations of mics or variations of filtering techniques. It can be used to reduce noise as well as to reduce the presence of interfering speech components for sound source separation when multiple passengers are speaking [8]. As mentioned before, adaptive Wiener filtering provides an efficient way to keep up with the statistics of desired or undesired signals [9],[10]. Even if other forms of signal enhancement algorithms are used, Wiener filtering still proves to be a practical additional step in the process [10],[11], [12]. For model-based Wiener filter, the statistics can also be estimated using a prior model and applied to the Wiener filter [3].

2.3 MWF in signal enhancement and automobile audio systems

MWF can be used with microphone arrays to avoid interference and reduce noise [13]. The MWF has some equivalences with beamforming algorithms [14], where the term beamforming here is not restricted to a microphone array of closely located microphones, i.e., microphones can be distributed in a car (e.g., the primary and secondary microphones for driver and front seat passenger). Beamformers are typically designed based on knowing a propagation model for a sound coming from the direction of a target speech source, as well as knowing the multichannel

(multi-microphone) correlation matrix of the signals to be minimized (i.e., background noise and other competing speech sources). The correlation matrix can also be estimated based on propagation models for the noise and competing sources. Beamformers may thus require knowing the direction of arrival of the target and competing sources. Instead of relying on a propagation model and direction of arrival for the sources, the MWF formulation is entirely based on estimating correlations between the different microphone signals and the target speaker signal to extract (e.g., driver voice). The resulting correlations are used to build a multichannel correlation matrix and multichannel correlation vector, from which the MWF solution is computed. The required correlations can be estimated if it is possible to detect when the different sources are active, separately or jointly. Therefore, no acoustic propagation model or direction of arrival estimation is required in the MWF approach.

The use of the MWF algorithm has been suggested in several applications (e.g., for hearing aids [15],[16]). When it comes to automobiles, the multichannel Wiener filter (MWF) for speech enhancement has been considered before in car environments, but it has been either for solely removing background noise from a single talker [17], [18], [19], for positioning virtual acoustic sources during conference calls performed in cars [20], or in the context of SONAR-like driver assistance [21]. Variations of MWF such as a minimum variance distortionless response (MVDR) is used in [22] to remove wind noise. But it has not been used yet for multi-sources processing such as to avoid notch filtering effect when mic summing or to extract individual sources, while reducing background noise. These are effects that need to be considered when assessing the advantages of having a dedicated mic for each person in the front car cabin.

2.4 Different microphone setups in automobile audio systems

Single mic, multiple mics, or an array of mics can be used in signal enhancement. The positioning of the mics also plays a major role in speech enhancement. There can be separate mics dedicated to speech and noise, to get the characteristics of noise separately [9], or each mic can be dedicated to a speaker [23], or multiple microphones can be used to define a desired signal space beyond which sound pick-up is considered to be noise [10]. An array of mics is also considered in [12],[22] to improve the in-car communication system by reducing noise.

2.5 Speech enhancement for automobiles

The car cabin is a small space that constitutes a relatively compact acoustic environment. Even though most surfaces inside the car are poor reflectors of sound, they do not substantially impair the quality of speech captured by in-car microphones. In contrast, the primary challenge in automotive speech processing stems from the presence of broadband acoustic noise, which overlaps significantly with the frequency range of human speech. This noise becomes particularly problematic at higher vehicle speeds, where it can severely degrade speech quality. The dominant sources of in-car noise [24] include the engine and its drive train operation, tire-road interaction, and turbulent airflow over the vehicle body, as shown in Figure 2.1. Of these, wind noise tends to increase at a faster rate with vehicle speed compared to tire-generated noise, making it a particularly challenging component to suppress in high-speed driving conditions.

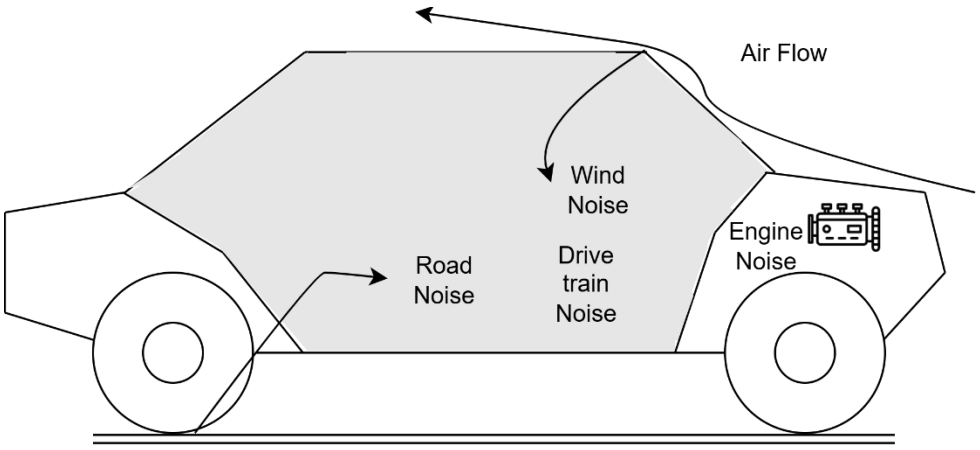

Figure 2.1 Car cabin noises

Chapter 3 Car Set-up

This section goes through the car cabin environment in the front and how different components are set up for the experiments considered in this thesis. This is derived from the existing car models and microphone-loudspeaker setups. The work of this thesis was performed in collaboration with the audio group of a car manufacturer company.

3.1 Speech signal capturing

One of the primary considerations when designing a car environment for audio processing is selecting the appropriate type of microphone. In many of the previous works, it can be seen that an array of microphones, as well as individual microphones with different placements, have been investigated. In our system, we chose cardioid microphones directed toward (the expected position of) each person sitting in the front. This decision can be justified as follows:

- Directional Sensitivity and Noise Reduction: Cardioid microphones have an audio pick-up pattern. They have a direction of arrival (DOA) dependent and frequency dependent response. They predominantly capture sound from the front and attenuate sound from the sides and back. This directionality ensures that the dominant speech captured by the mic is the person sitting in front of it while attenuating surrounding sounds such as noise or interfering speech.
- System Complexity and Integration: An alternative to using standalone microphones is using a microphone array. This involves multiple microphones and requires beamforming with signal processing, increasing complexity. Furthermore, with individual cardioid mics it is easier to change the location of microphones while integrating, depending on different car measurements. The individual cardioid mics setup can also be easily extended for multiple passengers.
- Cost-Effectiveness and Maintenance: Microphone arrays are comparably more expensive from an initial and maintenance standpoint. This is because they require multiple components and more complex processing. Therefore, cardioid microphones present a cost-effective alternative to microphone arrays.
- Performance in Dynamic Acoustic Environments: Considering the constantly changing acoustic conditions in vehicles, which depend on external environmental factors, cardioid mics are be able to provide more consistent performance.

Figure 3.1 shows the general polar plot of a cardioid mic. When humans speak, they do not emit sound equally in all directions [25]. The directional distribution of speech for a human head is more concentrated in the front, which can be efficiently picked up by the directed microphone. The primary microphone is directed towards the driver and the secondary microphone is directed towards the passenger.

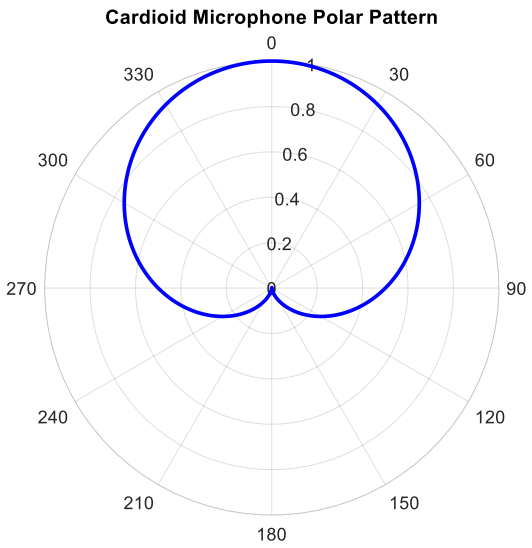

Figure 3.1 Polar pattern of a Cardioid mic

3.2 Car model

To design our car set-up, we used the Ford Explorer 2025 as the approximate car model. The interior dimensions are:

- Length: 5m
- Width: 2m
- Height: 1.78 m.

The environment for the experiments is set up as explained below.

3.3 Impulse responses

The impulse responses are generated using rir_generator() from [26]. It is a MEX-function designed for MATLAB™, and it allows users to adjust parameters such as reflection order, room

dimensions, and microphone directivity. It uses the image method proposed in [27]. It is frequently used in acoustic signal processing applications to generate synthetic room impulse responses. The room is assumed to be a rectangular enclosure with a source-to-receiver impulse response. This can approximate a car setup for our experiments.

The driver and passenger mics are placed in front of the driver and passenger respectively, pointing towards them. Cardioid mics, which are directional, are used. The mics are kept 0.8m apart. A reverberation time of 0.07 seconds is selected. The exact positions of the mics and sources with respect to the car dimensions are given in Table 3.1.

Table 3.1 Mic and source positions in the car

Microphone/Source	Distance from the front(m)	Distance from the left side(m)	Distance from the floor(m)
Primary Microphone	1.65	0.6	1.7
Secondary Microphone	1.65	1.4	1.7
Driver	2.5	0.6	0.75
Passenger	2.5	1.4	0.75
Left loudspeaker	2.2	0.15	0.3
Right loudspeaker	2.2	1.85	0.3

It should be noted that the dimensions and coordinates used in the rectangular room simulations are based on the car exterior dimensions. In retrospect, it would have been preferable to use the dimensions and coordinates of the car cabin instead. However, the distances between the different key elements (driver, passenger, loudspeakers, microphones) remain the same, and it was possible to adjust the reverberation level to the desired level for the simulations (set to a T60 reverberation

time of 70ms). Therefore, the validity of the results provided in this thesis should not be significantly affected by the choice of using the car exterior dimensions.

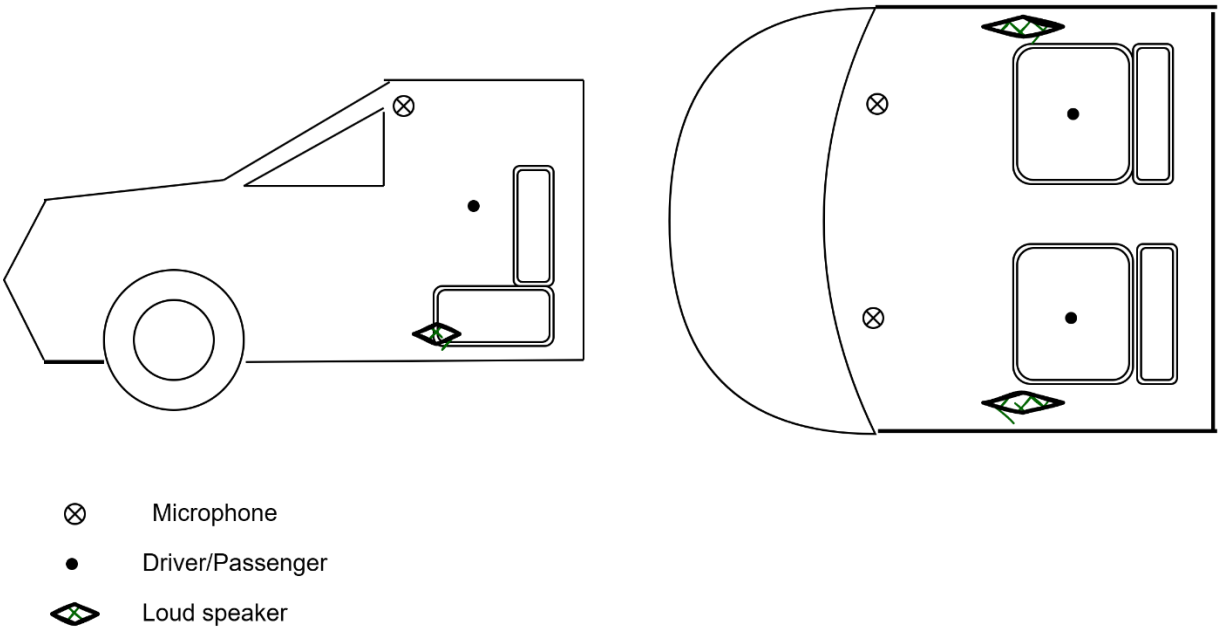

Figure 3.2 Positions of mics and audio sources in the car

Since we use a simulated environment for the car cabin for our system, few things should be noted. There are key differences between simulated environments and real-world car cabins that impact the performance of in-vehicle speech and acoustic processing systems. We assume some simple reflections and surface behaviors, whereas real car interiors can affect sound propagation unpredictably due to its geometry or materials used. Moreover, real-world environments introduce diffraction effects, where sound waves bend around physical obstacles such as seats, dashboards, and passengers. Background noise in simulations is typically controlled and stationary, while real driving conditions introduce dynamic noise that can vary with speed and environment. Additionally, environmental factors like temperature and humidity can affect acoustics.

All the simulations for testing are carried out in MATLAB, except for AECMOS and DSNMOS calculations which use a virtual environment to call python functions. The AECMOS and DSNMOS methods and calculations will be defined in a later section.

3.4 AEC

Acoustic echo cancellers were initially developed for telecommunications to maintain duplex communication by canceling the signals reflected from the caller. These systems model the echo path and use this model to cancel the echo from the received signal. Over the years, various adaptive algorithms have been employed for this purpose, such as LMS (Least Mean Squares), RLS (Recursive Least Squares), and NLMS (Normalized Least Mean Squares).As cars began to be equipped with hands-free telephony and modern audio systems, acoustic echo cancellers became necessary. In addition to the echo from the remote caller, cars also have loudspeakers playing music or other audio, which should not interfere with the speech of the passengers or driver. Whenever loudspeakers are active, acoustic echo cancelling (AEC) is required to remove loudspeaker content picked up by the microphone(s). For each microphone where AEC is applied, it may be mono AEC or multichannel AEC, depending on the audio played by the loudspeakers. Further processing called acoustic echo suppression (AES) can also be performed after each AEC unit to further attenuate the residual echo. AEC and AES processing have access to the loudspeaker signal(s) (Rin), in addition to the microphone signals. Acoustic echo cancellers in vehicles must make use of AEC and AES processing to ensure clear communication while eliminating external and internal echoes.

The AEC used is a proprietary algorithm provided by the industry partner. It is based on FDAF (Frequency-domain Adaptive Filter) algorithm [28], which implements classical NLMS adaptive filtering in the frequency domain.

3.5 Metrics used for performance evaluation

In this work, different metrics are used to evaluate different aspects of signal improvement, i.e., canceling of notch filtering effect and noise reduction.

3.5.1 PESQ

The Perceptual Evaluation of Speech Quality (PESQ) score is an objective method standardized by the International Telecommunication Union (ITU) under Recommendation ITU-T P.862 [29]. It is widely used to assess the end-to-end speech quality of telephone networks and speech codecs. It is the result of integrating two earlier algorithms: the Perceptual Analysis Measurement System (PAMS) and PSQM99, an enhanced version of the Perceptual Speech Quality Measure (PSQM). PESQ was developed to assess speech quality across a broader range of network conditions.

PESQ takes two main audio signals as input:

- The Reference Signal: This is the original, clean, unprocessed speech signal that serves as the benchmark for quality assessment.
- The Degraded Signal: This is the speech signal that has passed through the system under test (e.g., a telephone network or codec) and may contain various distortions such as noise, coding artifacts, packet loss, filtering, or delay.

For normal subjective test material, the PESQMOS value returned by PESQ typically lies between 1.0 (bad) and 4.5 (perfect, no distortion). Although this is uncommon, the PESQMOS may fall below 1.0 in cases of extremely high distortion. Since PESQ requires both the reference (original) and the degraded signal, it is not intended for non-intrusive measurements where the original signal is unavailable.

The scale -0.5 to 4.5 (or similar scales) used by PESQ is derived from the original MOS scale (0-5), where a score above 4.0 is considered "toll quality" (good quality comparable to original uncompressed signal telephone quality). For wideband speech, a wideband version of PESQ is used, i.e., wPESQ [30], and we use it for our research.

3.5.2 AECMOS

AECMOS [31] was developed by researchers from Microsoft to offer an accurate, efficient, and scalable metric for assessing speech quality impacted by echo. It is trained using ground-truth human ratings in accordance with the guidelines from ITU-T Rec. P.831 [32], ITU-T Rec. P.832 [33], and ITU-T Rec. P.808 [34].

Common methods for evaluating AEC models involve intrusive objective measures such as Echo Return Loss Enhancement (ERLE) and PESQ. However, commercially available objective metrics like 3QUEST [35] and ACOPT 32 [36] also have their limitations. 3QUEST is designed to measure only single talk echo performance, while ACOPT 32 requires the same near-end signal to be played twice, limiting its applicability in real call scenarios. Furthermore, the outputs of ACOPT 32 can be difficult to interpret.

The AECMOS model operates with the following inputs:

- Near-End Microphone Signal: This audio is captured by the microphone at the near-end. It typically includes near-end speech, background noise, and any far-end signal leaked through as an echo.
- Far-End Signal: This audio signal originates from the far end of the communication, which is played through the near-end loudspeaker.
- Enhanced Signal: This is the audio signal processed by the acoustic echo canceller to remove the echo component while preserving the near-end speech.
- Optional Scenario Marker:
    a. Near-End Single Talk: Only the near-end user is speaking.
    b. Far-End Single Talk: Only the far-end signal is present at the near end
    c. Double talk: Both the near-end and far-end users are (or can be) speaking simultaneously.

The AECMOS model aims to assess the quality of the enhanced signal, focusing particularly on residual echo and other potential degradations. The AECMOS model produces two Mean Opinion Score (MOS) predictions on a scale of 1 to 5, each reflecting assessments in distinct categories. The first MOS prediction pertains to echo. This score gauges the anticipated human subjective rating of call quality degradation specifically attributed to echo. A higher score (closer to 5) indicates less echo impairment, and a lower score (closer to 1) suggests a greater level of annoyance due to echo, categorized as "Very Annoying." This prediction aims to evaluate the extent of annoyance caused by echo to a human listener. The second MOS prediction addresses other types of degradations. This score predicts the quality impairments that arise from sources other than echo. Such degradations may include noise, missing audio, distortions, or cut-outs. Similar to the echo MOS, a higher score (closer to 5) indicates fewer other degradations, whereas a lower score (closer to 1) reflects higher annoyance to the listener. An AECMOS score of 3.5 and above is considered good.

3.5.3 Signal Interference Ratio Gain (SIR gain)

This can be used to measure the decoupling of the signals extracted by the MWF filters. It can be defined from the power ratios of desired signal over interference signal, with the output ratio divided by the input ratio. Equivalently, it can be defined by the power gain of the desired signal from input to output, over the power gain of the interference signal from input to output. Here, for

the primary mic, the desired signal is the driver speech and the interference is passenger speech, and vice versa for the secondary mic. So, the SIR gain $SIR\ gain_{dB}$ can be calculated as:

$$Gain_A = 10 * \log_{10}\left(\frac{P_{Ao}}{P_{A1}}\right) \tag{3.1}$$

P_{Ao} is the power of the output extracted signal A when only source A is present (active) and P_{A1} is the power of the signal from source A at input microphone 1 (primary).

$$Gain_B = 10 * \log_{10}\left(\frac{P_{Bo}}{P_{B2}}\right) \tag{3.2}$$

P_{Bo} is the power of the output extracted signal B when only source B is present (active) and P_{B2} is the power of the signal from source B at input microphone 2 (secondary).

$$SIR\ gain_{dB} = Gain_A - Gain_B \tag{3.3}$$

3.5.4 Signal to Noise Ratio (SNR) Gain

The SNR gain is used to measure how much the background noise is attenuated by the system. It is calculated from the ratio of signal power to noise power at the input and output of the system, with the output ratio divided by the input ratio. Equivalently, it can be defined by the power gain of the desired signal from input to output, over the power gain of the noise signal from input to output. The noise power gain is given as:

$$Gain_{noise} = 10 * \log_{10}\left(\frac{P_{No}}{P_{Ni}}\right) \tag{3.4}$$

Where P_{No} is the power of noise at the output when only noise is present, and P_{Ni} is the power of noise at the input reference microphone i=1 or 2. The SNR gain $SNR\ gain_{dB}$ for the MWF extracting source A (driver) is then:

$$SNR\ gain_{dB} = Gain_A - Gain_{noise} \tag{3.5}$$

Similarly, an SNR gain can be defined for the MWF extracting source B (passenger).

3.5.5 DNSMOS

DNSMOS (Deep Noise Suppression Mean Opinion Score) is a non-intrusive perceptual objective speech quality metric developed by Microsoft to evaluate noise suppressors. DNSMOS P.835 [37] was developed based on P.835 [38] human ratings.

To get the DNSMOS scores, an audio input sampled at 16 kHz is required. There is no need for a clean or reference signal, making this entirely non-intrusive. It gives the following MOS scores:

- Speech quality (SIG): This score reflects the quality of the speech component in the audio clip. The ratings vary from very poor (MOS=1) to excellent (MOS=5).
- Background noise quality (BAK): This score reflects the quality of the background noise in the audio clip. These ratings also vary from very poor (MOS=1) to excellent (MOS=5).
- Overall quality (OVRL): This score represents the overall quality of the audio clip, considering both speech and background noise. Similar to SIG and BAK, the ratings range from very poor (MOS=1) to excellent (MOS=5)

All these scores are intended to align with human subjective evaluations on a scale from 1 to 5. A DNSMOS score of 3.5 and above is considered good.

3.6 Different noises, their spectra, and characteristics

White noise is the most common noise used for testing signals and systems. It has a flat frequency spectrum with equal power at all frequencies, making it useful to model random disturbances.

Hoth noise was developed by Daniel F. Hoth [39] to be used as a standard background noise model derived from real-world measurements of room noise spectra. It was defined by pooling data from multiple locations. The curve may be raised or lowered according to the sound level as the general shape remains the same over different locations. Hoth noise is frequently used for speech quality evaluations and the performance of communication systems. Hoth noise has high energy in the lower frequency ranges and tapers off to higher frequencies. It is assumed to be relatively stationary or slowly varying and can be used as steady-state noise for testing. To adapt the noise to the modern wideband and super wideband telephone standards, which extend beyond 4 kHz, extended noise profiles for Hoth noise were used. According to ITU-T P.800, the Hoth noise power density spectrum is defined from 100 Hz to 8000 Hz with band edges aligned to IEC 61260 octave and fractional-octave filters, and a -15 dB/octave slope is applied beyond 8000 Hz.

Any non-white noise signal with power spectrum density (PSD) varying as a function of noise is referred to as colored noise. Even if this can be a good real-world representation of room noise and can be used as a baseline, in cars it should be noted that wind noise can be quite different, being non-stationary and possibly imbalanced at different microphones (i.e., with different levels).

Other types of non-white (colored) noises are red and pink noises, generated by processing i.i.d., zero-mean, additive white Gaussian noise [40]. This is carried out by applying DFT on the white noise signal and manipulating the complex spectral coefficients to obtain the new spectrum of the coloured noise. Signal conditioning of the coloured noise is performed in the time domain to ensure zero mean and a standard deviation of 1. All colored noises with $1/f^n$ spectrum must be high pass filtered to avoid infinite power at very low frequency. Also, a high-pass filter is required if we want that, for a given global SNR, the noise content is significant at speech frequencies (i.e., otherwise most of the noise is just at irrelevant low frequencies and does not affect speech). Hence, the colored noises are passed through a high pass filter with cut off frequency 50Hz before using them.

For red noise, the PSD slope varies as a function of $\frac{1}{f^2}$ with a spectral slope rate of -6dB/oct or -20dB/dec. For pink noise, the PSD slope varies as a function of $\frac{1}{f}$ with a spectral slope rate of -3dB/oct or -10dB/dec. These signals are stationary. Green noise can be generated to have a PSD profile matching the human ear sensibility to sound loudness, and it is approximated by inverting grey noise response [41] and modifying it to prevent increase of high frequency.

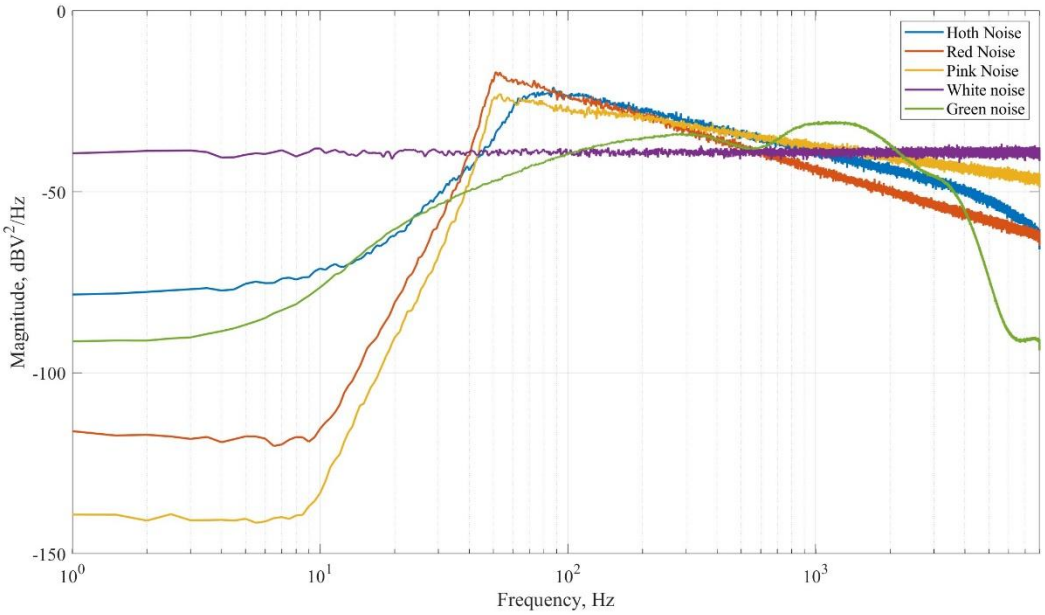

Figure 3.3 Power Spectra of different noises used for testing

We also do some tests with wind noise. As previously mentioned, wind noise can be non-stationary and may not be balanced at the microphones. Wind noise is also highly unpredictable as it can change with car speed, wind direction, weather or even vehicle shape. This variability makes it hard to perform repeatable and replicable tests, so this can pose some challenges while testing. Figures 3.4 and 3.5 show the power spectrum and the time domain plot of the wind noise used for testing.

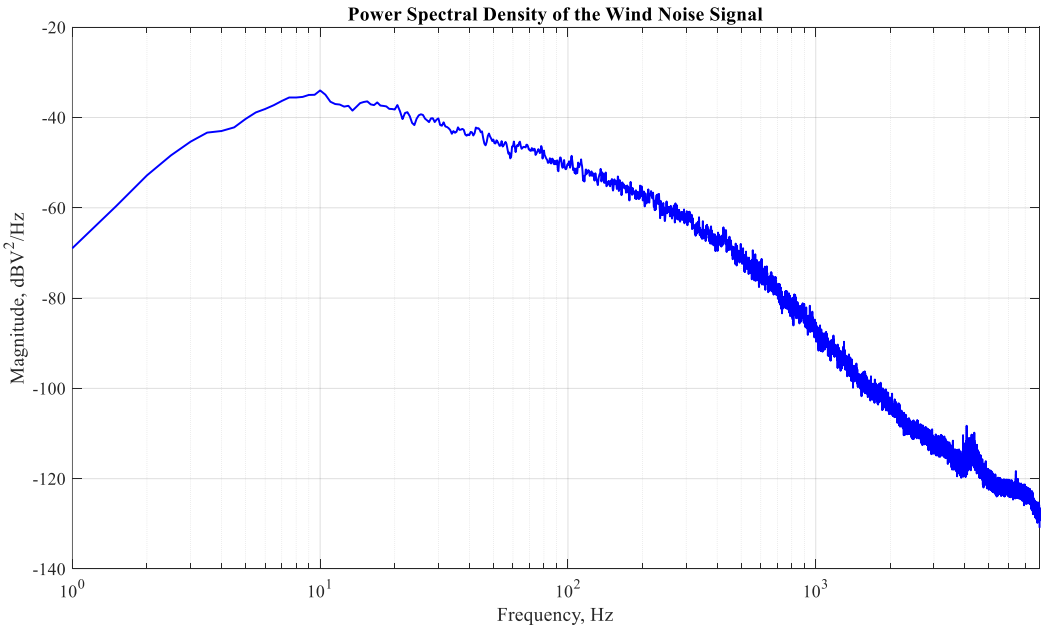

Figure 3.4 Power Spectrum of wind noise used for testing

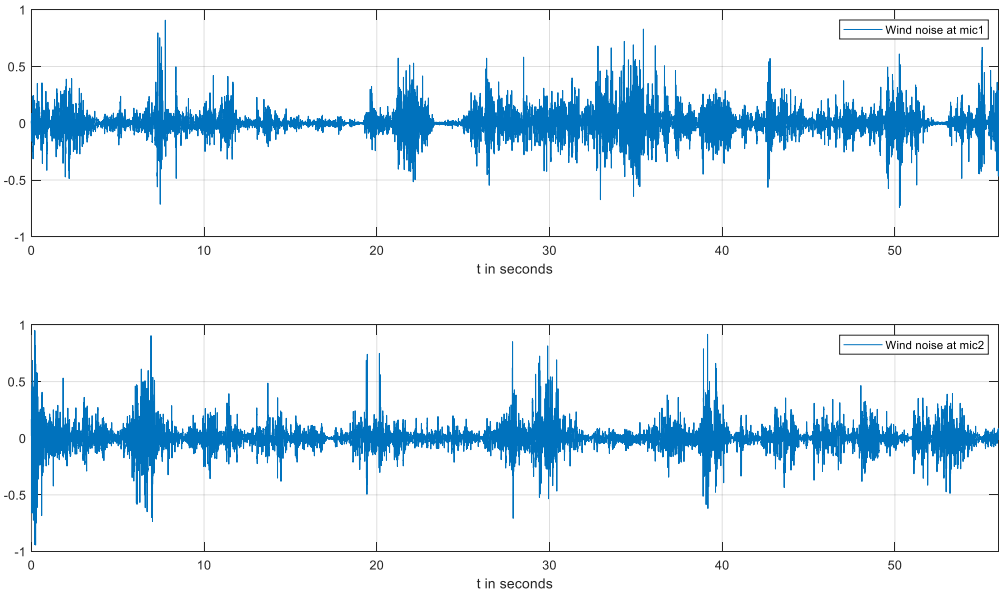

Figure 3.5 Time domain wind noise signals at two microphones.

Chapter 4 Microphone and Acoustic Echo Cancellers Setups

Acoustic echo cancellation is one of the most important parts of communication systems. In a car, this is quite relevant since there can be music playing through loudspeakers, or feedback from the "far-end" speech played through loudspeakers. In both cases, these are undesirable components picked up by the microphone(s). Most cars nowadays are equipped with a single mic/single array mic whose purpose is mostly to pick up the driver speech. We consider a 2-mic system that will help improve the quality of driver and passenger speech as each mic system will be located/directed closer/towards each speech source. When there are two mics, there is an option of using 2 AECs or 1 AEC. So, if the simple switching and mixing options are considered to generate one resulting signal from the two mics signals, we can have 4 scenarios:

- Switching before a single AEC (Figure 4.1): In this case, switching is done between the mics signals, depending on which active speaker is detected. This scenario has low computational complexity. However, problems can arise while detecting the speaker in cases of unbalanced noise, increased head movement by the speaker, time delay between the mics, etc.
- Switching after two AECs (Figure 4.2): Each mic signal is passed to an AEC, and the switching is done between the two signals from the AECs. This scenario calls for two AECs, which significantly increases the computational complexity because of an additional AEC. At any time, one extra AEC is working whose output is not used because of the switching.
- Mixing/Adding before a single AEC (Figure 4.3): This keeps the computational complexity of the system to a minimum as the signals just have to be added. The problems that can arise are that the notch filtering effect when adding two signals with different propagation time delays between the mics. Also, the noises at the two mics will be added as well, which will double the noise level at the AEC (if the noise is balanced between mics).
- Adding after two AECs (Figure 4.4): Each mic signal is passed to an AEC, and the signals are added after the AECs. Here as well, using two AECs significantly increases the computational complexity. And as in the previous case, there can be a notch filtering effect.

In our simulations, these scenarios can be chosen by changing the values of the variables in a configuration file as shown in Table 4.1. Imm_enable, Imm_order, and Mode are the variables that can be changed in the code to choose between different scenarios.

Table 4.1 Settings for different scenarios

Scenario	Imm_enable	Imm_order	Mode	Switching/adding action
1	1	1	1	Switching before AEC
2	1	2	1	Switching after 2 AECs
3	1	1	2	Adding before 1 AEC
4	1	2	2	Adding after 2 AECs

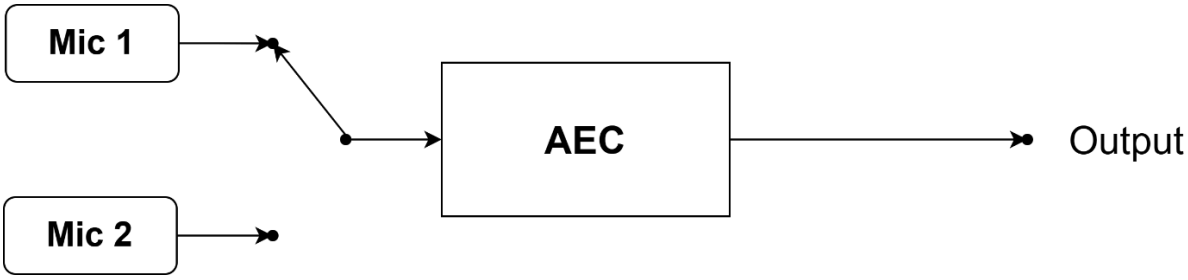

Figure 4.1 Switching before AEC

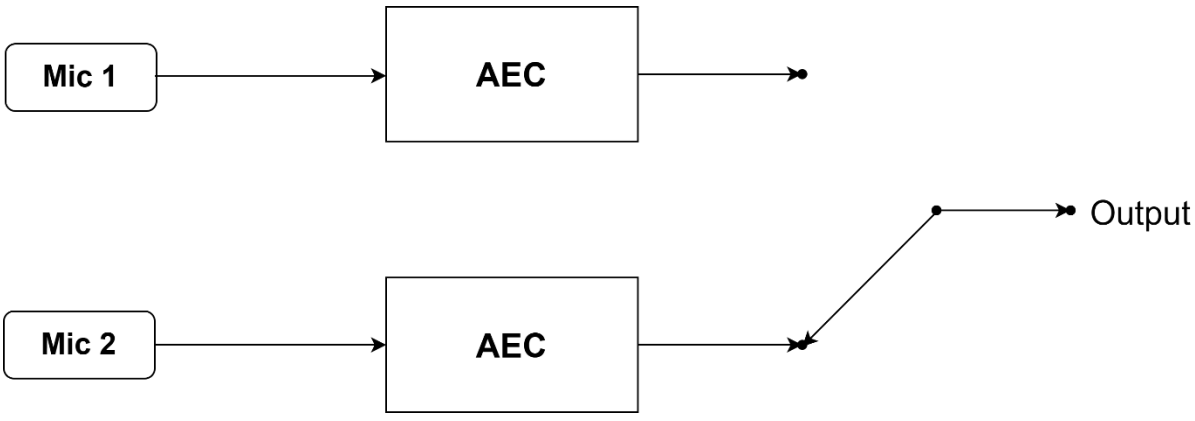

Figure 4.2 Switching after AECs

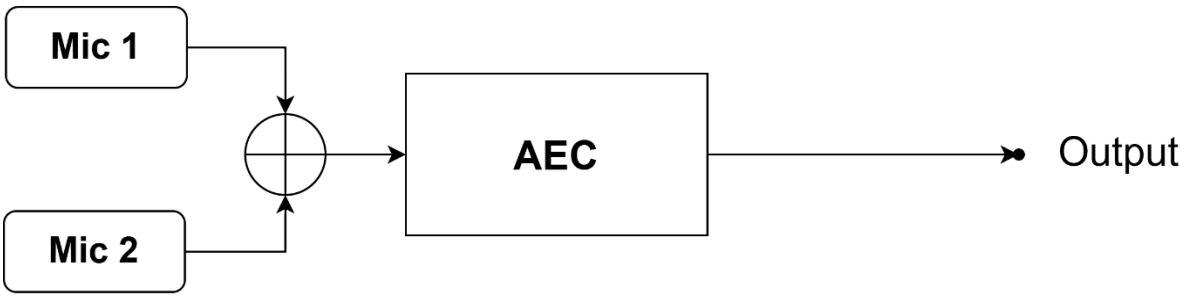

Figure 4.3 Mixing/Adding before AEC

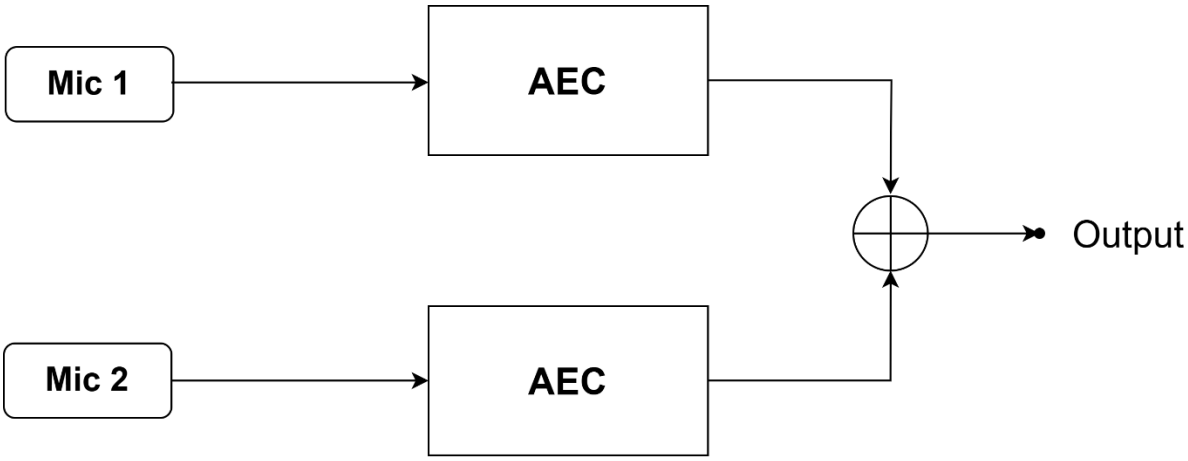

Figure 4.4 Mixing/Adding after AECs

4.1 Switching

The smoothed power of the signals is used to make decisions for switching. A recursive moving average is used for the signal powers. To determine the forgetting factor λ, a time constant of τ_{frames} = 20 frames is used. So, the corresponding time constant τ in ms is,

$$\tau = 20 \text{ frames} \times 8 \frac{\text{ms}}{\text{frame}} = 160 \text{ ms} \tag{4.1}$$

Where 8 ms is the frame size.

$$\lambda = (\tau_{frames} - 1)/\ \tau_{frames}) = 0.95 \tag{4.2}$$

$$P_{mic1} = \lambda \times P_{mic1} + (1 - \lambda) \times x_1 \tag{4.3}$$

$$P_{mic2} = \lambda \times P_{mic2} + (1 - \lambda) \times x_2 \tag{4.4}$$

$$P_{Rin} = \lambda \times P_{Rin} + (1 - \lambda) \times y \tag{4.5}$$

where x_1, x_2, y are the power values for the current frames of the mic 1, mic 2 and Rin (loudspeaker, "far-end") signals.

When $P_{Rin} < threshold,$ (the threshold is chosen to be 0.0008 in our specific case), i.e., when there is no far-end signal, our system switches to the signal of the microphone with higher power. If switching happens after the AECs, switching chooses between the signals at the output of 2 AECs.

4.2 Mixing/Adding

If only 1 AEC is used, the signals from the mic are added and fed to the AEC. This raises the possibility of increased power from the far-end signal and noise signals. If 2 AECs are used, the outputs from the AECs are summed, in which case, the far-end will already be suppressed. Mixing/adding is carried out in the time domain.

4.3 Speech Signals

The speech signals used for several of our experiments are as shown in Figure 4.5. There is intermittent speech segments of the driver and the passenger starting from time 4 seconds. The first 4 seconds have only far end loudspeaker signal (to allow sufficient time for the AEC module to converge) and the time segment 4-8 seconds has only driver and passenger ("near-end") speech. The last 4 seconds have both driver or passenger speech and far end loudspeaker signals. It can be observed in Figure 4.5 that the same speech source was used for the driver and passenger speech. This was used to reduce the fluctuations caused by different speech source material (e.g. specific voice, or specific words) on some of the metrics used. It was possible to do this because the scenarios of driver and passenger simultaneously talking was not considered in this case. Even though a specific set of speech signals are used to compare metrics, the results are generalizable to other speech signals since the system performance mostly depends on the acoustic paths.

The input signal at both left and right speakers (Rin) is the same, corresponding to a "mono" far-end source (as in most voice calls). For the case of media playback, there can be stereo or multichannel sources in the loudspeakers, and then a different multichannel AEC is used. But for MWF we have assumed no loudspeaker signal (or we have assumed that some AEC was enabled at all microphones, such that only the mixture of near end sources remains after the AEC processing). The simulated impulse responses have an inherent cross-side attenuation of 2 dB. This

means that driver speech reaching the secondary microphone has an attenuation of 2 dB compared the passenger speech, similarly passenger speech reaching the primary microphone has an attenuation of 2dB compared to driver speech. In several of our experiments, an additional cross-side attenuation factor will be added to the impulse responses for different test settings and this will be mentioned for each case.

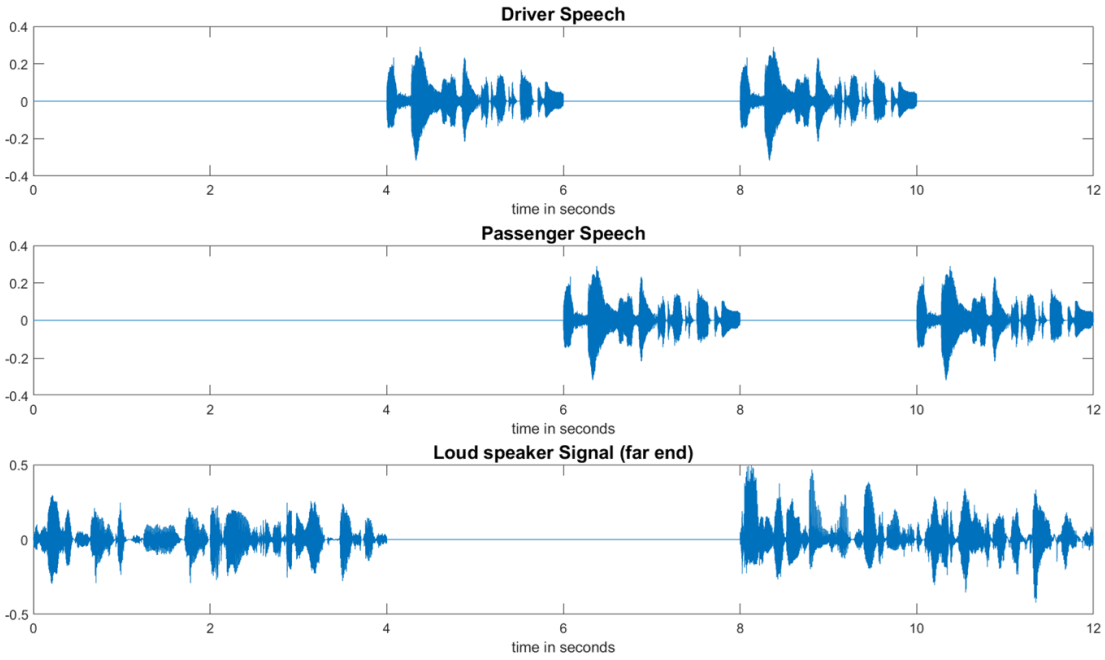

Figure 4.5 Speech signals used for testing

4.4 Comparison of One-mic Performance with Two-mics Performance

In this chapter we wish to first demonstrate some effects such as the lower quality of passenger speech pick-up when only a single primary mic is used, the notch filtering effect when mixing/adding of two mic signals is used, and AEC performance degradation when two mic signal switching is used before a single AEC module. For this, we compare the cases using one mic vs two mics. The two mics are used in 4 setups, as mentioned before: switching before two AECs, switching after a single AEC, adding after two AECs, and adding before a single AEC. It should be noted again that using two AECs adds to the computational complexity. The cross-side attenuation used in this experiment is 10 dB (2dB from the original simulated impulse responses + additional 8dB), from the feedback provided by the collaborating car manufacturer company.

Tables 4.2-4.4 show the performance scores for different setups in terms of wPesq and AECMOS echo scores.

The scores are calculated on the whole duration of the signals. The driver, passenger, and loudspeakers sources are those shown in Figure 4.5, unless they are disabled. For the single mic case in Table 4.2, the weaker score of 3.1968 for passenger speech compared to 3.9727 for driver speech shows clearly that the single mic case leads to a degradation of quality for passenger speech.

Table 4.2 wPesq scores for single mic

Active sources	Single mic
Driver and loudspeakers	3.97
Passenger and loudspeakers	3.30
Driver, passenger and loudspeakers	3.38

Table 4.3 wPesq scores for 2 mics

Active sources	Switching before 1 AEC	Switching after 2 AECs	Adding before 1 AEC	Adding after 2 AECs
Driver and loudspeakers	3.97	3.97	3.12	3.75
Passenger and loudspeakers	3.68	3.68	3.30	3.34
Driver, passenger and loudspeakers	3.69	4.04	2.91	3.66

In Table 4.3, the overall reduced performance observed when mixing/adding signals from two microphones (last two column) compared to the switching approach (first two columns of scores), can be explained by the notch filtering effect. Switching when only the passenger is speaking (Table 4.3) also achieves better performance than the single-microphone setup (Table 4.2), because the secondary microphone is selected for the passenger's speech, thus the passenger speech is no attenuated as in the primary microphone.

With the switching and mixing methods, it could be expected that the scores for driver speech should be the same as the scores for passenger speech; however, we see in Table 4.3 that it is not

the case. As can be seen from Figure 4.5, the far-end signal (played through loudspeakers and mostly removed by the AEC module) is different in the 8-10 sec segment (during driver speech) and in the 10-12 sec segment (during passenger speech). This leads to different wPESQ scores, because wPESQ has signal-dependencies.

The last row of Tables 4.2-4.3 appears in grey, because they are less reliable. This is because they correspond to the case with alternating driver & passenger speech, which causes time-varying alignment issues between the reference source signals and microphone signals required by the wPesq method, so the scores become less reliable.

The AECMOS echo scores can be used to detect the presence of remaining echoes in the AEC processed signal. As such, they should detect the performance degradation caused by switching in the case of alternating driver & passenger speech, because of the sudden echo path change caused by switching. Table 4.4 shows the resulting AECMOS echo scores for the 2-mics configurations. However, the expected negative impact of switching before a single AEC, compared to switching after two AECs or compared to mixing/adding, was not observed in this experiment (i.e., the scores in Table 4.4 don't show any correlation with the expected behaviour). As we will further demonstrate later, this is due to the fact that the simulated setup has perfectly symmetric impulse responses (IRs) between loudspeakers on one side and microphones on the other side. Consequently, when switching from the (primary) mic signal close to the driver to the (secondary) mic signal close to the passenger, the expected sudden echo path change is absent, i.e., the echo path remains the same because of the symmetry in the simulation. Therefore, all the scores found in Table 4.4 actually represent a good performance (all above 4.2 scores, out of 4.5), with low residual echo component after AEC. Later in this chapter, we will perform experiments without the echo path symmetry in the simulations.

Table 4.4 AECMOS echo scores for 2 mics

Active sources	Switching before 1 AEC	Switching after 2 AECs	Adding before 1 AEC	Adding after 2 AECs
Driver and loudspeakers	4.40	4.40	4.24	4.37
Passenger and loudspeakers	4.41	4.41	4.37	4.41
Driver, passenger and loudspeakers	4.47	4.48	4.36	4.52

Figures 4.6-4.9- show the source signals, microphone signals and output signals for the 4 different two-mics scenarios and with both near end sources (driver and passenger) enabled as well as far-end source (loudspeaker). The input in the first subplot corresponds to the input at primary mic and the input in second sublot refers to the input at the secondary mic. The output in the third subplot is the output from the AEC. If the switch flag is high, then the primary mic is chosen, and secondary mic is chosen otherwise. In these figures, a good performance is indicated in the bottom subplot when the blue signal (output signal) in the interval 8-12 sec (with far-end signal attenuated by AEC module) is the same as the blue signal in the interval 4-8 sec (without far-end signal).

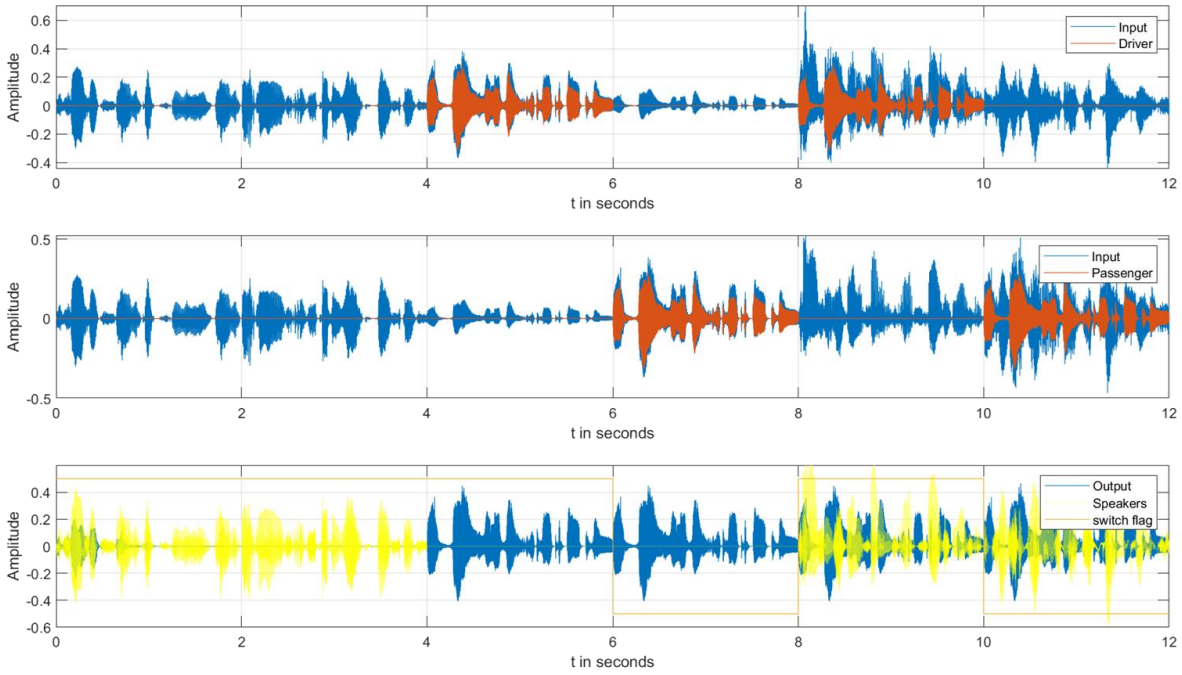

Figure 4.6 Switching before single AEC

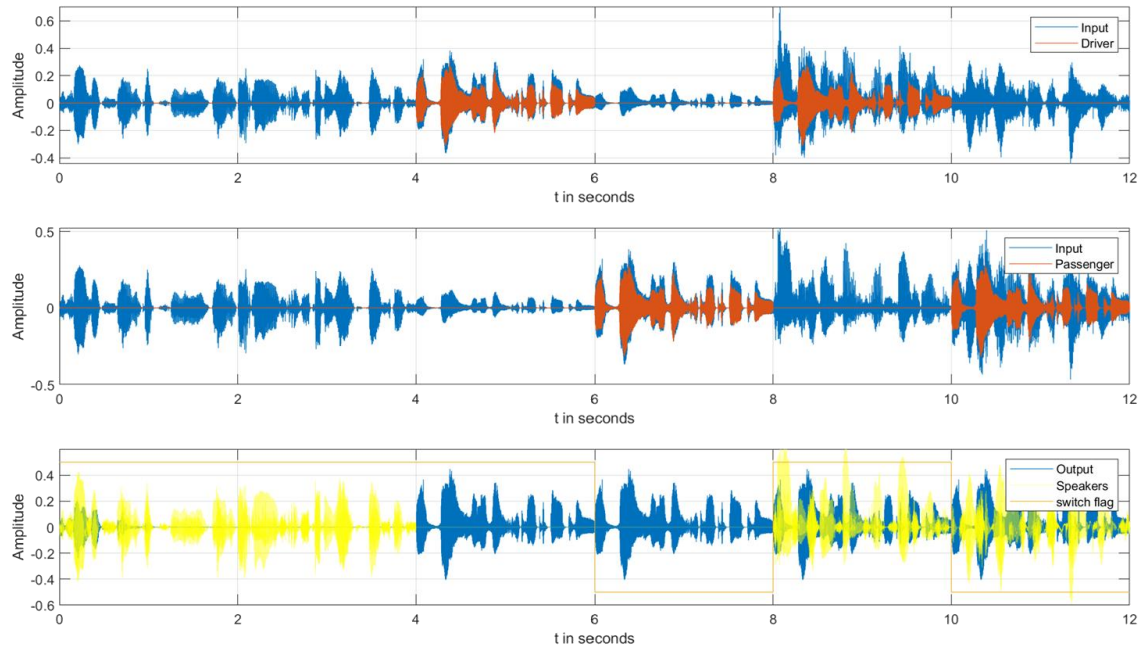

Figure 4.7 Switching after two AECs

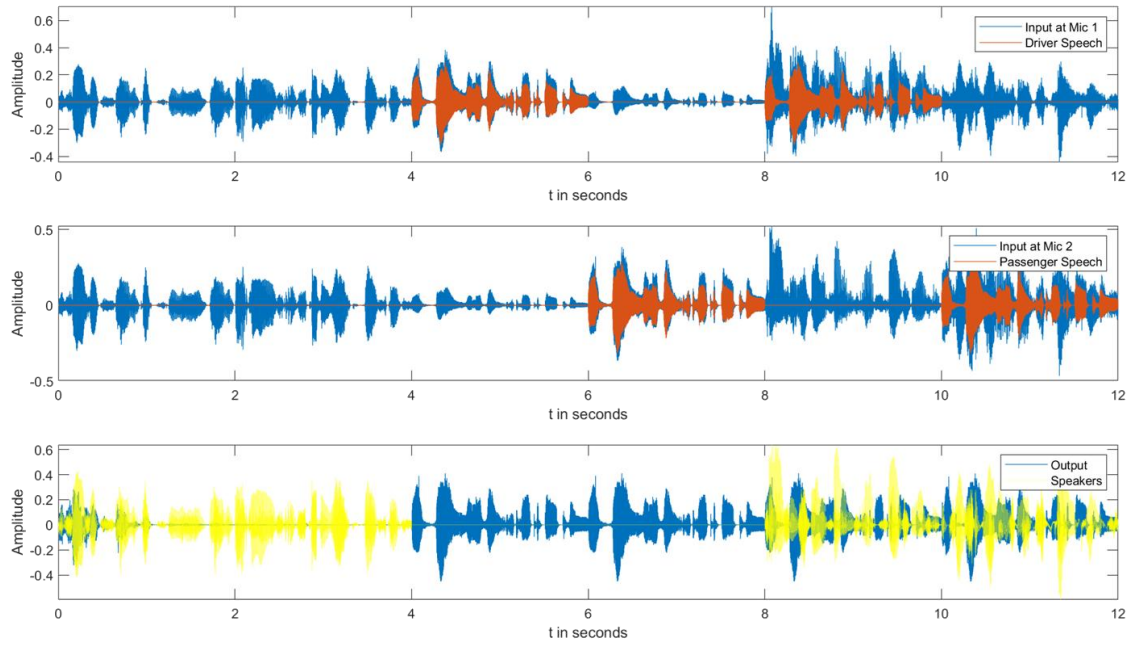

Figure 4.8 Adding before single AEC

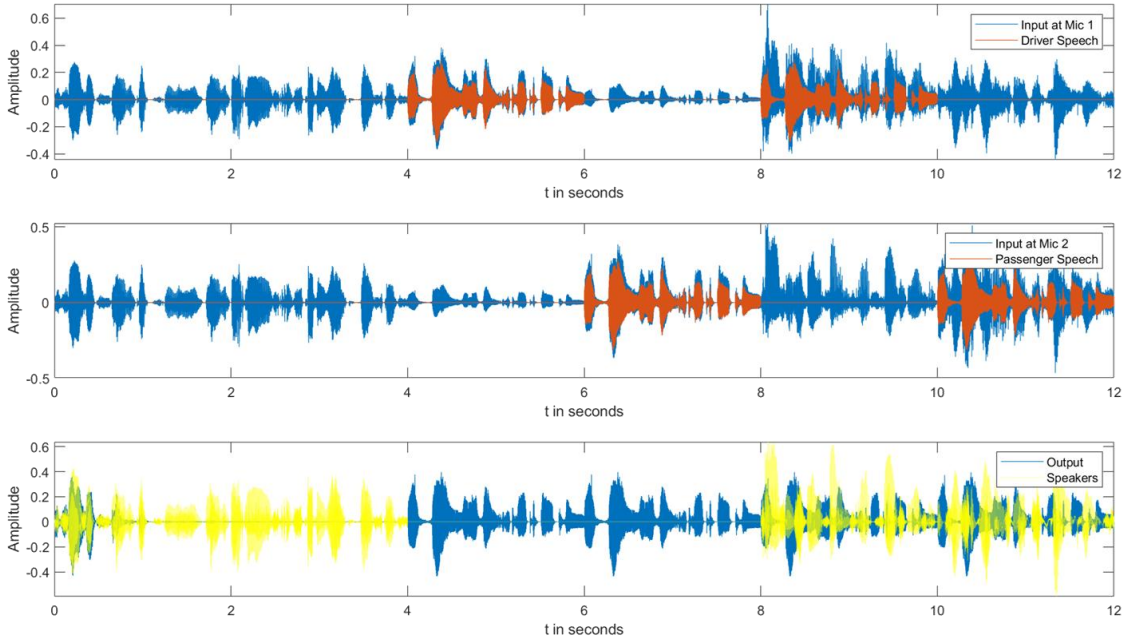

Figure 4.9 Adding after two AECs

In Figures 4.6-4.7 and for the rest of this thesis, ideal switching was used to determine if we have mostly driver speech or passenger speech. This is done in order to decouple the effect of a practical switching algorithm from the other aspects investigated in this thesis, and since the thesis does not focus on developing a practical switching scheme. In a real-life scenario, a switching scheme based

on microphone levels could be used (either before a single AEC or after two AECs), to determine which mic signal should be used. This can be done in various ways, including:

- Comparing the instantaneous or smoothed power of microphone signals;
- Comparing correlations between microphone signals and past switching output signal samples.

4.4.1 Effect of symmetry of microphones on switching

As previously explained, by default for our simulated setup IRs are symmetric, and this is an important factor to consider while switching. To see the impact of switching for the more realistic case of asymmetric IRs and echo path changes, we changed the (x,y,z) position of the secondary microphone from (1.65m, 1.4m, 1.7m) to (1.65m, 1.8m, 1.7m), which corresponds to moving the microphone 40cm to the right.

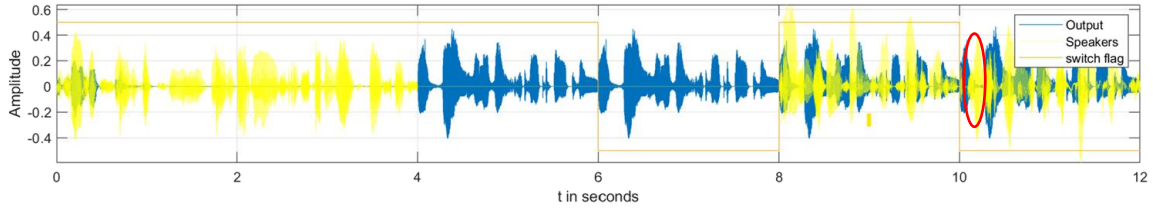

Figure 4.10 Switching before single AEC, symmetric simulated IRs

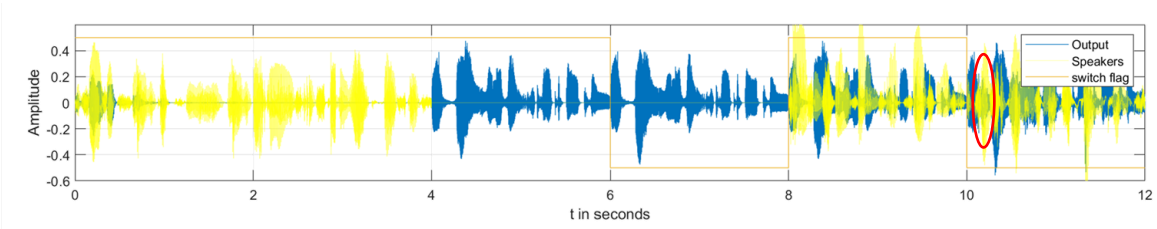

Figure 4.11 Switching before single AEC, asymmetric simulated IRs

In Figure 4.11 showing the result of switching before single AEC with asymmetric simulated IRs, we can see remaining transients from the far end signal in the circled portion (signal in blue, after the speech source switch which occurs at 10 sec). Those transients are caused by the adaptation required by the AEC module when it is active (in presence of far-end signal) and when a sudden echo path change occurs. In contrast, if we compare with the result of switching before single AEC

with symmetric simulated IRs in Figure 4.10 (i.e., same figure as the bottom subplot in Figure 4.6), we see that the blue signal is much smaller in the circled portion (i.e., it corresponds more closely to the output signal found after 6 sec when there is no far end signal).

The impact of the sudden echo path change while the AEC unit turned on at 10 sec is also reflected in the AECMOS scores and wPESQ scores, where we can observe in Table 4.5 a clear performance drop when asymmetric IRs are used.

Table 4.5 Performance for switching with and without symmetric IRs

	wPesq	AECMOS echo score	AECMOS degradation score
switching symmetric IRs	3.6941	4.4739	4.0505
switching asymmetric IRs	1.9051	4.3049	3.8264

Chapter 5 Multichannel Wiener Filter

Wiener filtering is used for linear estimation of a desired signal based on a reference or input signal. The signals are treated as random processes, and the filter design is based on the statistics obtained through ensemble averaging. Assuming that the random processes are also jointly ergodic and wide sense stationary, the time and ensemble averages are asymptotically the same.

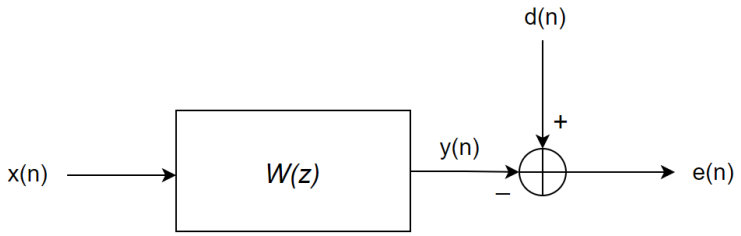

Figure 5.1 Block diagram of Wiener filter

Wiener solutions can be extended to multichannel scenarios, i.e., multiple inputs. In our case, we have two inputs, the primary mic signal and the secondary mic signal.

Consider that there are 2 inputs, $x_1(n)$ and $x_2(n)$ used to predict a desired signal $d(n)$ producing an error signal $e(n)$

$$e(n) = d(n) - \sum_{k=0}^{N-1} w_{1k} x_1(n-k) - \sum_{k=0}^{N-1} w_{2k} x_2(n-k) \tag{5.1}$$

In matrix form, it can be written as:

$$e(n) = d(n) - \begin{bmatrix} w_{1,0} \cdots w_{1,N-1}\, w_{2,0} \cdots w_{2,N-1} \end{bmatrix} \begin{bmatrix} x_1(n) \\ \vdots \\ x_1(n-N+1) \\ x_2(n) \\ \vdots \\ x_2(n-N+1) \end{bmatrix} \tag{5.2}$$

Or in vector notation form:

$$y(n) = \boldsymbol{w}^H \boldsymbol{x}(n) \tag{5.3}$$

$$e(n) = d(n) - \boldsymbol{w}^H \boldsymbol{x}(n) \tag{5.4}$$

Where

$$\boldsymbol{w} = \left[w_{1,0}^* \cdots w_{1,N-1}^* w_{2,0}^* \cdots w_{2,N-1}^*\right]^T = \left[w_{1,0} \cdots w_{1,N-1}\, w_{2,0} \cdots w_{2,N-1}\right]^H \tag{5.5}$$

The mean square error (MSE) is given by:

$$MSE = \xi(\boldsymbol{w}) = \phi_{ee}(0) = \sigma_e^2 = E[e(n)e^H(n)] \tag{5.6}$$

$$= E\left[\left(d(n) - \boldsymbol{w}^H \boldsymbol{X}(n)\right)\left(d(n) - \boldsymbol{w}^H \boldsymbol{X}(n)\right)^H\right] \tag{5.7}$$

$$= E[d(n)d^H(n)] - \boldsymbol{w}^H E[\boldsymbol{x}(n)d^H(n)] - E[\boldsymbol{x}^H(n)d(n)]\boldsymbol{w} + \boldsymbol{w}^H E[\boldsymbol{x}(n)\boldsymbol{x}^H(n)]\,\boldsymbol{w} \tag{5.8}$$

$$\boldsymbol{p} = E[\boldsymbol{x}(n)d^H(n)] \tag{5.9}$$

$$\boldsymbol{R} = E[\boldsymbol{x}(n)\boldsymbol{x}^H(n)] \tag{5.10}$$

$$\xi(\boldsymbol{w}) = \sigma_d^2 - \boldsymbol{w}^H \boldsymbol{p} - \boldsymbol{w}^{T\boldsymbol{p}^*} + \boldsymbol{w}^H \boldsymbol{R}\boldsymbol{w} \tag{5.11}$$

$$\frac{\partial \xi(\boldsymbol{w})}{\partial \boldsymbol{w}^*} = \boldsymbol{0}_{2N\times 1} - \boldsymbol{p} - \boldsymbol{0}_{2N\times 1} + \boldsymbol{R}\boldsymbol{w}_{opt} = \boldsymbol{0}_{2N\times 1} \tag{5.12}$$

$$\boldsymbol{w}_{opt} = \boldsymbol{R}^{-1}\boldsymbol{p} \tag{5.13}$$

The performance at the optimum is the minimum MSE, or MMSE:

$$MMSE = \xi\left(\boldsymbol{w}_{opt}\right) = \xi_{min} = \sigma_d^2 - \boldsymbol{p}^H \boldsymbol{R}^{-1}\boldsymbol{p} \tag{5.14}$$

At the optimum error, the error signal is uncorrelated/orthogonal to the input signals and to the output signal.

These equations form the foundation on which the Wiener solution for multichannel scenarios, i.e., multiple inputs, is developed. Even though it can be extended to any number of inputs, we have two inputs for our system: the two microphone signals. We also have two sources to extract

and mix/add: the driver source and the passenger source, and each of these sources requires a different multichannel Wiener filter. This is further explained in the next section. The equations for MWF are not new, although we will customize the way to estimate the statistics to fit our application.

5.1 Frequency domain MWF to extract the source S_A as received at the primary microphone

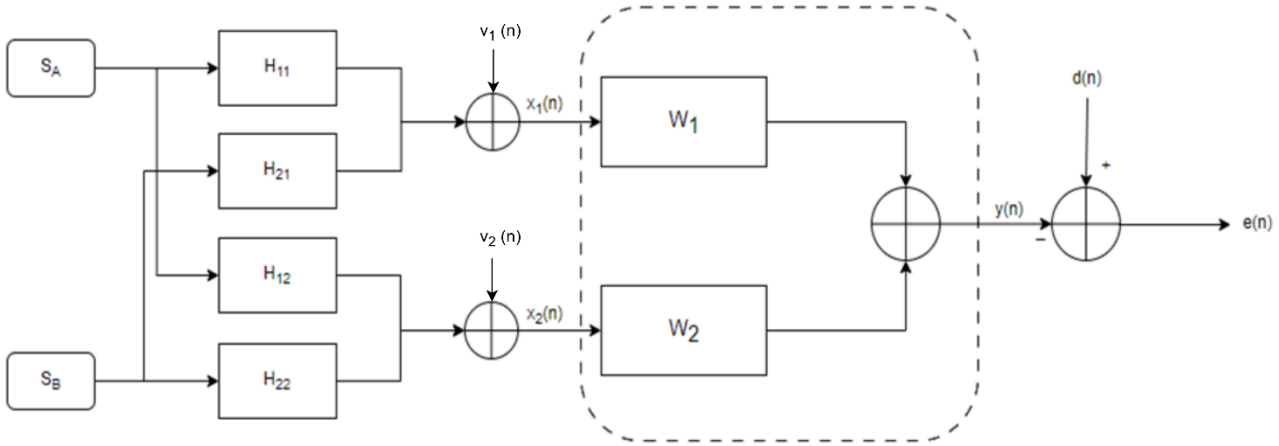

Figure 5.2 Multichannel Wiener filter for two inputs and desired signal d

Consider the driver and passenger sources, $s_A$and s_B, with background noise v. Then at the two microphones 1 and 2, this can be written as:

$$x_1(n) = s_{1A}(n) + s_{1B}(n) + v_1(n) \tag{5.15}$$

$$x_2(n) = s_{2A}(n) + s_{2B}(n) + v_2(n) \tag{5.16}$$

Where the sources are uncorrelated to each other as well as the noise, i.e.,

$$E\left[s_{jA}(n)s_{iB}(n)\right] = 0;\ i,j = 1,2 \tag{5.17}$$

$$E\left[s_{jk}(n)v_i(n)\right] = 0;\ i,j = 1,2; k = A,B \tag{5.18}$$

In the frequency domain, the estimated source $\hat{s}_A$ at mic 1 from the MWF is :

$$\hat{s}_A(\omega) = \boldsymbol{w}^H(\omega)\boldsymbol{x}(\omega) = [w_1(\omega)w_2(\omega)][x_1(\omega)x_2(\omega)]^T \tag{5.19}$$

Where

$$\boldsymbol{w}(\omega) = \boldsymbol{R}^{-1}(\omega)\boldsymbol{p}(\omega) \tag{5.20}$$

$$\boldsymbol{w}(\omega) = [w_1^*(\omega) w_2^*(\omega)]^T$$

$$\boldsymbol{R}(\omega) = \begin{bmatrix} \phi_{x_1x_1}(\omega) & \phi_{x_1x_2}(\omega) \\ \phi_{x_2x_1}(\omega) & \phi_{x_2x_2}(\omega) \end{bmatrix} = \begin{bmatrix} \phi_{x_1x_1}(\omega) & \phi_{x_1x_2}(\omega) \\ \phi^*_{x_1x_2}(\omega) & \phi_{x_2x_2}(\omega) \end{bmatrix} \tag{5.21}$$

$$\boldsymbol{p}(\omega) = [\phi_{x_1d}(\omega) \quad \phi_{x_2d}(\omega)]^T = [\phi^*_{dx_1}(\omega) \quad \phi^*_{dx_2}(\omega)]^T \tag{5.22}$$

(with $\phi_{xy}(\omega) = E[y(\omega)x^*(\omega)]$)

The elements of the **R** matrix are given by,

$$\phi_{x_jx_i}(\omega) = \phi_{s_{jA}s_{iB}}(\omega) + \phi_{s_{jB}s_{iB}}(\omega) + \phi_{v_jv_i}(\omega) \tag{5.23}$$

$\phi_{x_jx_i}(\omega)$ can be estimated as a sum of components when single sources are active (only driver or passenger, with noise) and subtracting noise-only statistics:

$$\phi_{x_jx_i}(\omega) = \left(\phi_{s_{jA}s_{iB}}(\omega) + \phi_{v_jv_i}(\omega)\right) + \left(\phi_{s_{jB}s_{iB}}(\omega) + \phi_{v_jv_i}(\omega)\right) - \phi_{v_jv_i}(\omega) \tag{5.24}.$$

The elements of **p** are given by,

$$\phi_{x_jd}(\omega) = \phi_{s_{jA}s_{1A}}(\omega) \tag{5.25}$$

$\phi_{x_jd}(\omega)$ can be estimated when only the source s_A is active (driver with noise) and subtracting the noise-only statistics:

$$\phi_{x_jd}(\omega) = \phi_{s_{jA}s_{1A}}(\omega) = \left(\phi_{s_{jA}s_{1A}}(\omega) + \phi_{v_jv_1}(\omega)\right) - \phi_{v_jv_1}(\omega) \tag{5.26}$$

5.2 Implementation of Adaptive Multichannel Wiener Filter

As illustrated in Figure 5.3, for mixing/adding microphone signals without notch filtering effect (without switching), multichannel Wiener Filters (MWF) are implemented between the microphones and a single Acoustic Echo Canceller module (AEC). The signals from the microphones are directly fed into each MWF (one for each source, i.e., driver and passenger). These filters independently extract each source while mitigating the effects of two-path propagation (i.e., notch filtering effect). The MWFs keep adapting to update the filter coefficients with respect to the changes in the statistics of the speech signals from the driver and passenger and background noise.

Multichannel LTI filtering can introduce frequency dependent level changes but no non-linear audible distortions (like musical noise or other types of artefacts introduced by single channel Wiener filtering noise suppression based on time-varying frequency-dependent gains). If the MWF filters need to adapt (and therefore they are slowly time-variant and not LTI), the output signals may include some audible effects from the non-constant filtering, but normally the adaptation to the MWF coefficients is sufficiently slow and the output does not suffer from significant distortion.

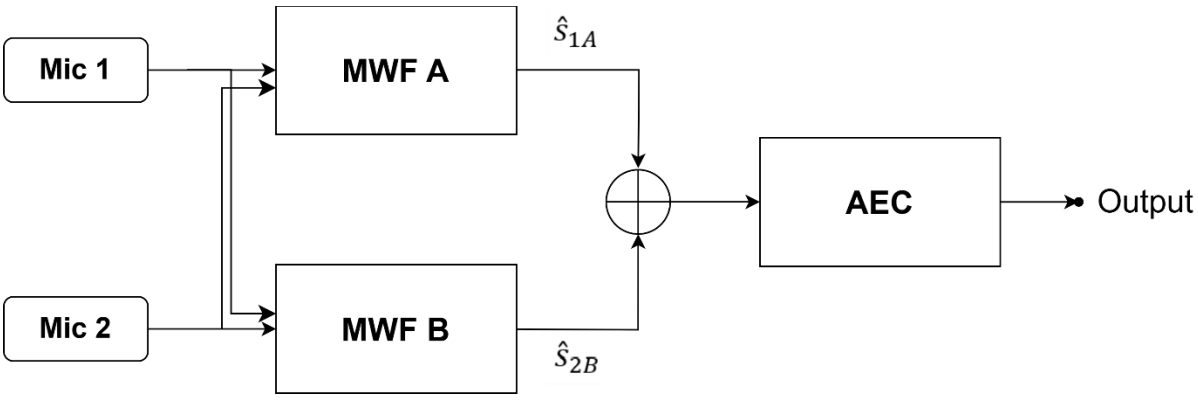

Figure 5.3 Sum of outputs from two MWF filters (sum of driver and passenger signal predictions), before single AEC module.

5.2.1 Overlap-Add and Windowing

The computations for MWF are carried out in the frequency domain. This includes calculating the convolutions in the frequency domain using Fourier transform multiplications. The overlap and add method allows the signal to be divided onto small overlapping blocks, processing each block separately. This helps with the real time processing of the signals.

One of the main concerns during windowing is the perfect reconstruction of the signal. This calls for the Constant Overlap Add (COLA) constraint [42], which ensures that overlapping windowed segments sum to a constant value during STFT and inverse STFT operations. This prevents distortion and allows for perfect reconstruction.

A window function satisfies the COLA constraint if:

$$\sum_{m} w(n - mR) = constant, \forall n \tag{7.6}$$

Where R is the overlap size and $w(n)$ is the window. In MATLAB™, this can be checked using the iscola() function. A Hanning window with option 'periodic' when the length is even and the shift (or overlap) is 50% satisfies the COLA constraint and gives perfect reconstruction. Iscola() can also be used to verify perfect reconstruction if a window is to be applied at both analysis and synthesis (which is preferable, to minimized block processing effects in the output signal). Then we use the function with option 'wola'. The use of the iscola() function is illustrated below.

```
len=128;
shift = floor((len+1)/2);
overlap=len-shift;

win = hann(len,'symmetric');
[tf,m,maxDeviation] = iscola(win,overlap,'ola')
tf=0,m =0.9913, maxDeviation =0.0085

win = hann(len,'periodic');
[tf,m,maxDeviation] = iscola(win,overlap,'ola')
tf =   1,m =  1,maxDeviation =   2.2204e-16

win = sqrt(hann(len,'symmetric'));
[tf,m,maxDeviation] = iscola(win,overlap,'wola')
tf =   0, m =    0.9913, maxDeviation = 0.0085

win = sqrt(hann(len,'periodic'));
[tf,m,maxDeviation] = iscola(win,overlap,'wola')
tf=1, m=1, maxDeviation= 2.2204e-16
```

A result of tf=1, m=1 indicates that the window with specified length and overlap is COLA compliant.

From Figure 5.4, it can be seen that even though both windows start at 0, the “symmetric” option ends at 0 and the “periodic” option ends just before 0, avoiding duplication of the first point. This shows that the end points don’t overlap for the periodic window. The effect of both windows during overlap and add is shown in Figure 5.5, where with the periodic window the result is perfectly flat in the middle portion (i.e., discarding transient effects at the beginning and the end).

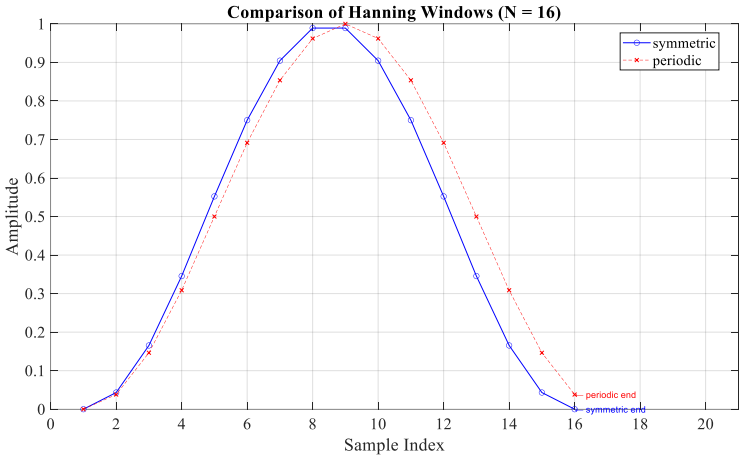

Figure 5.4 Symmetric and Periodic Hanning windows

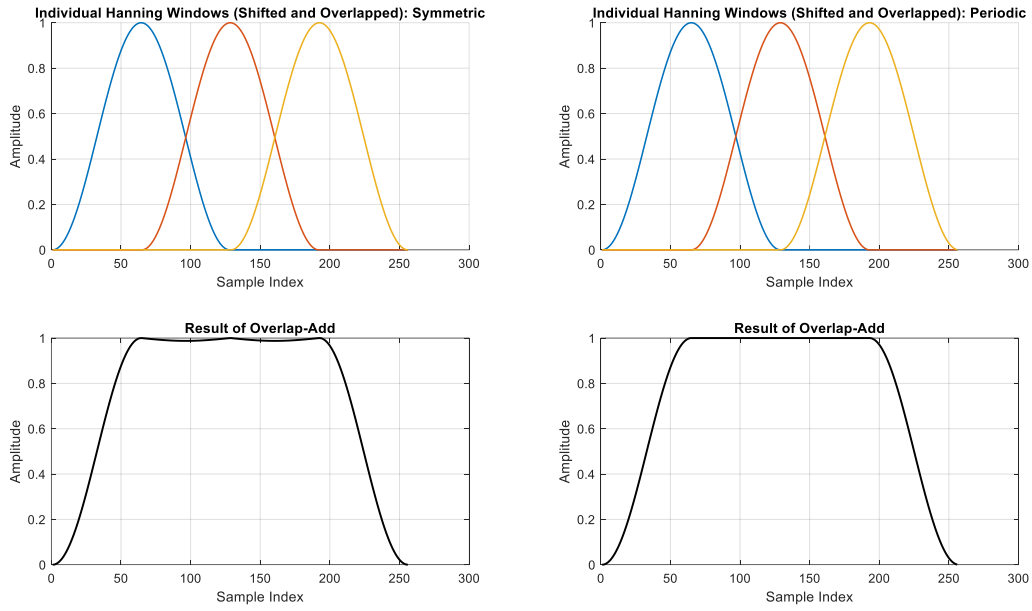

Figure 5.5 OLA with symmetric and periodic Hanning windows

5.2.2 Regularization factor and moving average factor

While applying MWF, it is imperative to choose the parameters that affect the performance of MWF properly. The parameters to be adjusted are:

- δ: regularization factor.
- λ: moving average factor for the correlation coefficients.

It should be noted that these parameters can change with frame size and the frequency bands considered. Generally, δ values as small as 0.01 were found to be appropriate. For larger frame sizes, larger δ values were found to be required. Typically, the value of δ should not exceed 1.0, since the regularization is applied in a normalized fashion in (7.7) below. For instance, when δ =10.0, the regularization is 10 times the trace value in the original matrix. This leads to small coefficients and small output values that need to be amplified. On the other hand, higher values can be used at some frequencies if we know there is more concentration of noise power in those frequencies, and less speech content.

We need the inverse of the correlation matrix to find the Wiener filter coefficients. This can give errors if R is poorly conditioned because it is nearly singular or in the presence of large noise. This calls for the regularization of the matrix with the regularization factor δ chosen accordingly:

$$w = \left(R + I * \delta * \frac{trace(R)}{length(R)}\right)^{-1} p \quad (5.7).$$

Here, $\frac{trace(R)}{length(R)}$ is the average power of the input signals. This method scales the regularization factor proportionally to the mic signals' power. A smaller δ is preferred in the case of low noise and higher δ when the noise is higher. The regularization factor in Wiener filtering is crucial for achieving a balance between restoring the signal without instabilities and suppressing a large amount of interference and noise.

λ acts as the forgetting factor determining how much the old correlation values can affect the new filter values. They can range from 0 (no history) to 1 (no forgetting). The λ value determines how many milliseconds of previous data are used, depending on the frame size.

The correlation factors are updated as shown

$$\phi_{AB} = \lambda * \phi_{AB} + (1 - \lambda) A_{freq} .* conj(B_{freq}) \quad (5.8)$$

Since (7.8) is applied in the frequency domain on a frame by frame basis, the λ value is also affected by the frame size used (in sec). Therefore, it is important to map properly the time constant $1/(1 - \lambda)$ (in frames) to a corresponding desired value in seconds.

Chapter 6 MWF and Notch Filtering Effect

In Chapter 5, it was seen that while adding the microphone signals before a single AEC or after two AECs, the performance (speech quality) can be reduced due to the notch filtering effect. Notch filtering effect occurs when two signals are out of phase at some frequencies, leading to destructive interference and attenuation (or complete cancellation) of those frequencies. This can negatively affect the quality of driver/passenger signals even when they have dedicated mics, since each mic picks up the signal from a given source. This two-paths propagation can in principle be eliminated by using a delay, so that the source signals picked up by each mic are time aligned, with no interference at any frequency. However, this gives rise to the following issues:

- If the resulting time aligned signals are mixed/added, the resulting signal can contain either (or both) driver and passenger signals, and these two speech sources require different delays to be time aligned. Therefore, the source for which the wrong delay is being used will still suffer from notch filtering effect (worsened, because the resulting total delay will be increased by the added delay, which leads to more notch locations in the frequency spectrum).
- If two different delays are used depending on which source is active (driver or passenger), this requires some switching mechanism, and switching can cause the problem of sudden echo path change which is detrimental to AEC performance.

Hence, we reached the solution of using a multichannel Wiener filter to remove two-path propagation and the resulting notch filtering effect. The MWF is implemented in the frequency domain, using Overlap-Add (OLA) with 50% overlap, with computations of the required correlations derived from microphone signals (with mixture of sources and background noise) rather than from ideal driver or passenger components at the microphones. This approach requires accurate detection of instances when only the driver or only the passenger is active. In a first phase, noise-free scenarios are considered, and scenarios with background noise are considered in the next chapter.

As previously mentioned, the cross-side attenuation is the attenuation found in a mic signal from a source on the opposite side, e.g., the attenuation to the driver signal while being received at the secondary mic (on the passenger side). The simulated impulse responses here are symmetrical

(because of symmetric loudspeaker and microphone locations in the simulated setup). Hence, both driver and passenger source signals suffer from the same cross-side attenuation. When the cross-side attenuation is higher, the effect of source signals from the opposite side reduces, which also reduces the notch filtering effect (i.e., because of smaller echo when mixing/adding the signals of both mics). Therefore, the worst case of the notch filtering effect is when there is the least cross-side attenuation. The simulated IRs used here have around 2dB cross-side attenuation, to which more cross-side attenuation can be added for testing. For testing, the original IRs are used without any additional cross-side attenuation, as they are the ones most affected by the notch filtering effect. This means the passenger's speech level is close to the driver's at the primary mic, and the driver's speech level is close to the passenger's at the secondary mic. To compare the performance, IRs with 8dB additional cross-side attenuation are also used with MWF.

To clearly understand the effect of MWF on the notch filtering effect, we used white noise as input, as it has a flat frequency spectrum with equal power over all frequencies. We also used different frame sizes to see how they can change the performance of MWF filters and the resulting cancellation of two-path propagation.

6.1 Additional cross attenuation: 0dB

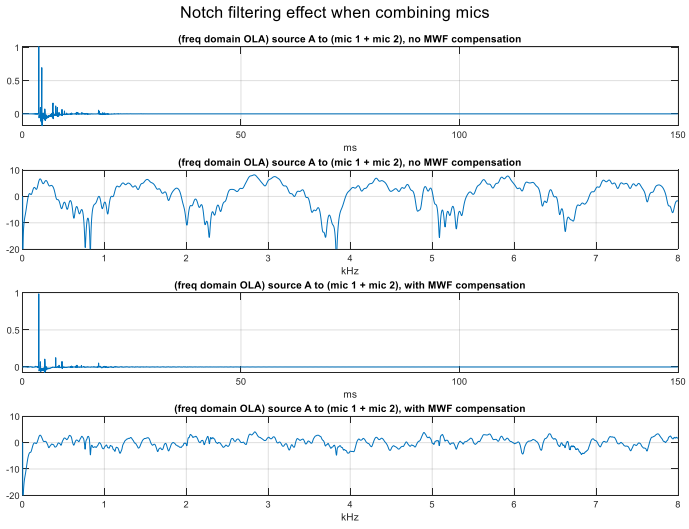

Figure 6.1 Notch filtering effect for 100ms frame size and 0dB additional cross-side attenuation

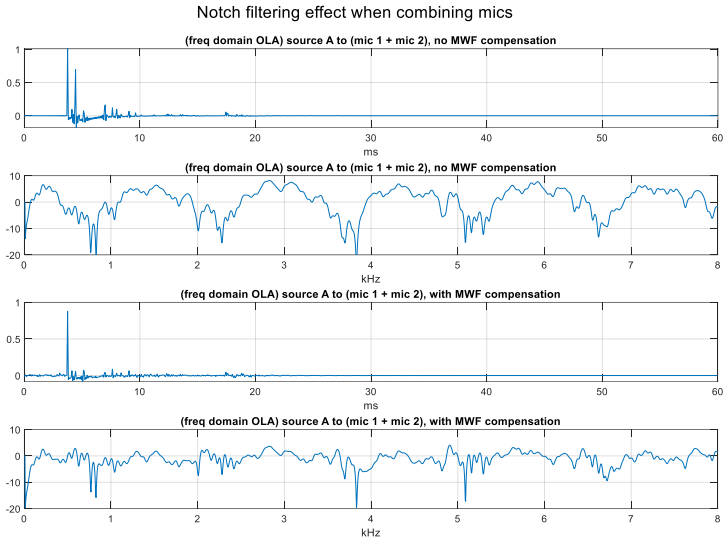

Figure 6.2 Notch filtering effect for 20ms frame size and 0dB additional cross-side attenuation

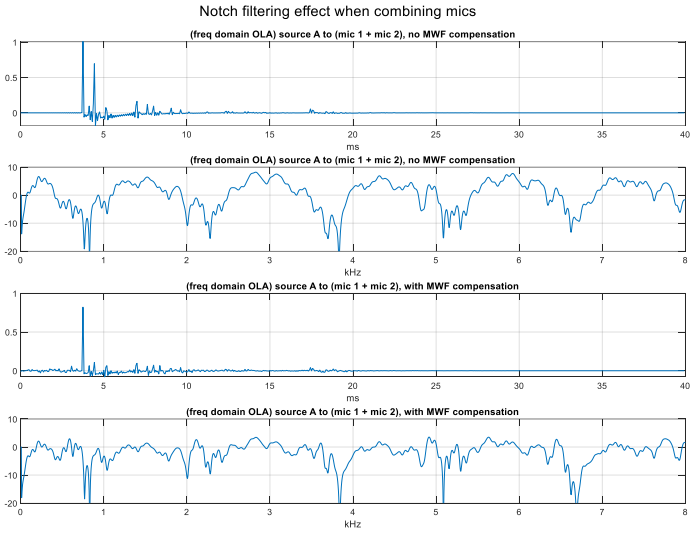

Figure 6.3 Notch filtering effect for 8ms frame size and 0dB additional cross-side attenuation

Figure 6.1 shows the result of adding the IRs from a source to the two mics (i.e., corresponding to signal mixing/adding for that source) with and without MWF. Without MWF, there are multiple notches, with big ones near 1, 2, 4, 5 and 7 kHz frequencies. When the frame size is 100ms, almost all the notches are removed for the MWF, giving the best results. This is because the MWF filter size (in coefficients) is the same as the frame size (in samples), therefore large frame sizes mean

longer MWF filters, which can better model the acoustic paths or their inverses. When the frame size is 20ms (Figure 6.2), there are some residual notches with the big ones around 1, 4, and 5 kHz frequencies. The performance for reducing notch filtering effect is further reduced when the frame size is 8ms (Figure 6.3); many notches are still present even after the filtering by MWF. This shows that a higher frame size is better in order to remove the notches from adding the IRs.

6.2 Additional cross-side attenuation: 8dB

Figures 6.4-6.5 show that there are hardly any deep notches when adding the simulated IRs with 8 dB additional attenuation. This is because the cross-side attenuation makes the opposite source's speech level low compared to the main source, i.e., the speech level of the passenger at the primary mic is low, and the speech level of the driver is low at the secondary mic. Therefore, even though there is improvement in the notch filtering effect with MWF, there is not a strong difference, and the frame size has less impact on the removal of the notches.

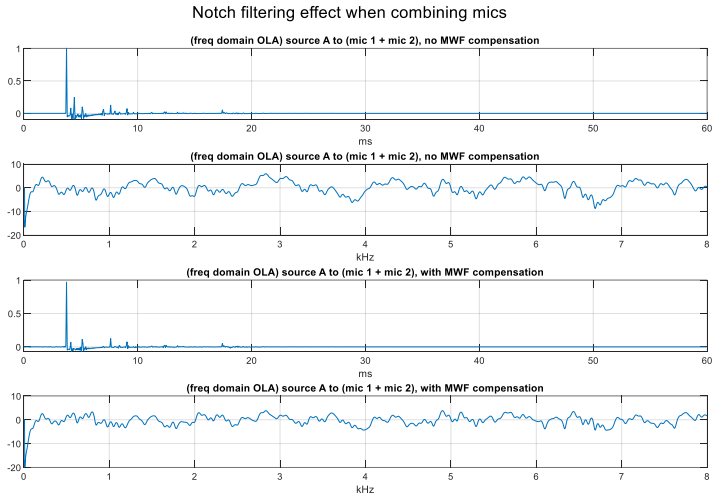

Figure 6.4 Notch filtering effect for 20ms frame size and 8dB cross-side attenuation

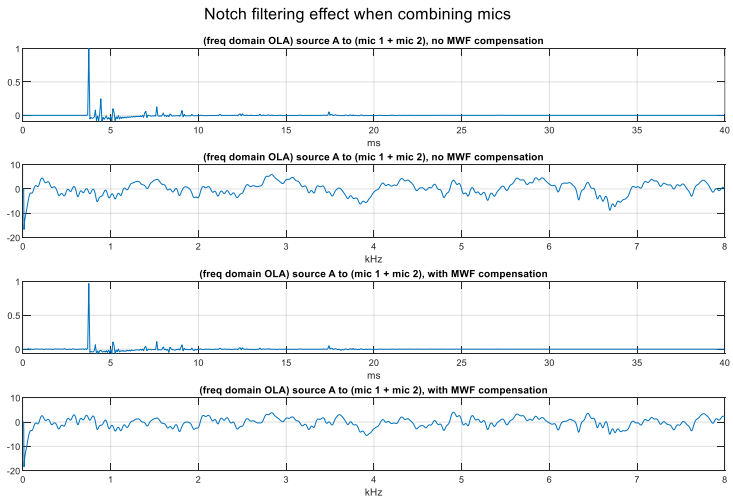

Figure 6.5 Notch filtering effect for 8ms frame size and 8dB additional cross-side attenuation.

It should be noted that in practical speech processing systems, frame size is typically chosen based on latency requirements. While our results may mention a frame size of 100 ms, this does not imply a recommendation to use such a size in real-time applications. In practice, shorter frame sizes, typically ranging from 8 ms to 30 ms, are preferred to reduce algorithmic processing delay and latency. The choice of frame size depends on the specific application and its latency constraints, which is why real-time speech communication systems often adopt smaller frame sizes. Thus, for our system, a frame size of 8ms was found to be apt for processing.

Chapter 7 MWF and Noise Reduction

This chapter investigates the performance of summing the outputs of MWF filters in the presence of speech sources (driver, passenger) and with background noise (white and colored noises).

7.1 Regularization factor and forgetting factor for different noises

The input SNR is fixed at 13dB and different regularization factors and forgetting factors are applied to see which would be the best fit for any noise condition. The SIR and SNR gains are calculated using the primary mic signal as input reference for the driver speech source and the secondary mic signal as input reference signal for the passenger speech source. The DNSMOS values are calculated on the sum of the output signals from the MWF.

The tables with the results are given in Appendix, section A.1. For white noise, Table A.1 shows the performance of the MWF with different frame size, λ and δ values. When λ is 0.96, which is the lowest value considered, it requires a δ of 1.0 to get a good performance i.e., where the DNSMOS values for both signal and noise are above 3.0. I also give the best SNR gain of around 9.0 and SIR gain values near 14.0 and 16.0. Even if higher λ is considered for a frame size of 8ms, it requires a regularization factor of 1.0 for good performance. It can be seen that with higher frame size, we need higher λ values as well as a δ value of at least 1.0 for acceptable metrics. Higher λ means that it needs a longer history to calculate the correlation coefficients. This is not desirable as speech signal statistics are time-varying, and it will affect the performance. This can be seen from the fact that overall the best performance is when the frame size is the smallest, i.e., 8ms.

When the red noise is used (Table A.2), and the frame size is 8ms, δ has to be 1.0 for good performance with DNSMOS values for signal and noise components above 3.0, even if higher λ values are used. As the frame size increases, there is a requirement for higher λ values, and δ cannot go below 1.0. When the frame size is higher than 24ms, the DNMOS for noise cannot reach a score of 3.0 anymore.

With pink noise, from Table A.3, there is less speech degradation at the output as indicated by DNSMOS signal score. But when it comes to noise removal, it requires a smaller frame size of 8ms and λ value of 0.96 and δ value of 1.0, as indicated by SNR gain as well as DNSMOS noise score.

Similarly, with green noise (Table A.4), noise removal becomes harder as the frame size increases. It requires a δ of 1.0 for a DNSMOS score of 3.0 and above. As higher frame sizes require higher λ values, it is evident that here as well, 8ms frame size with δ =1.0 and λ = 0.96 proves to be the preferred choice.

Hoth noise shows a trend different from the above noises, as no acceptable output could be obtained using a single δ for the entire frequency range. It required a higher δ for lower frequencies to suppress noise and a δ of 1.0 for higher frequencies to avoid distortion. Thus, the Table A.5 shows the results using different frame sizes and λ values with a δ of 100.0 for frequencies lower than 312.5 Hz and 1.0 for higher frequencies. Here also, a frame size of 8ms shows the best performance.

It is clear from these results that for stable performance it is most efficient to use a frame size of 8ms with λ of 0.96 and δ of 1 for all the noises (except for Hoth noise at low frequencies).

7.2 Active Speaker Detection

To compute the correlations required by the MWF filters, it is important to be able to detect when the different sources are active (i.e., a classifier is required to detect the conditions: no source (except background noise), only driver source, only passenger source, and both driver and passenger sources active). This can be done in various ways, including:

- Comparing the instantaneous or smoothed power of microphone signals
- Comparing correlations between microphone signals and past MWF output signal samples.

For the experiments performed in this thesis and as stated in an earlier chapter, in order to decouple the effect of active speaker detection from the performance achievable by MWF filtering, a perfect classification assumption is made, i.e., perfect knowledge of when each of the driver and passenger voice sources is active. Figure 7.1 shows the speech signals used for testing. From 0 to 20 seconds, the MWF is allowed to train and the metrics are calculated starting from 20 seconds. The SIR and SNR gains are calculated during the time segment 20-56 seconds and DNSMOS scores are computed during the time segment 20-36 seconds

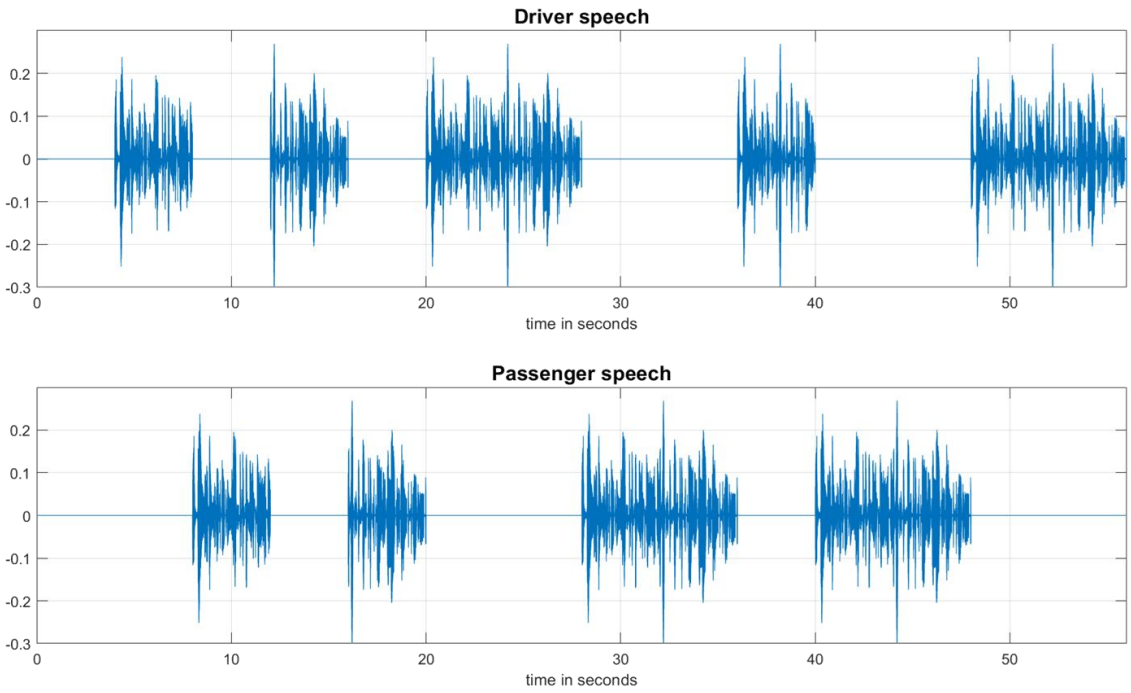

Figure 7.1 Speech signals used for testing

7.3 MWF with different types and levels of noise

This section evaluates the performance of mixing/adding the MWF filters outputs under different levels of noise. The types of noises used for testing are white, red, pink, green and Hoth noises. The frame size is fixed at 8ms for best noise reduction. The moving average factor of $\lambda = 0.96$ and regularization factor of $\delta = 1.0$. The only exception is Hoth noise, where a regularization factor of 100.0 is used for frequencies lower than 312.5 Hz (and 1.0 for higher frequencies). The original simulated IRs with 2dB cross-side attenuation are used, with no additional cross attenuation. The input SNR is varied from 10dB to 0dB.

While comparing with the sum of mic signals simple method, it can be noted that the SNR gain for this method is -1.0 dB and SIR gain is -1.26 dB. The SIR gain and SNR gain values with MWF at different input SNRs are given in Tables A.6 to A.10 in the appendix section A.2. It can be seen that using MWF provides significant improvements for the SIR gain and SNR gain scores. The SIR gain ranges from 7 dB to 9 dB for white noise, and the SNR gain is around 10 dB. The SIR gain for red and pink noises is similar to white noise, but the SNR gain is around 6 dB. For green noise, SIR gain and SNR gain are always close to 8 dB or higher. The performance with Hoth

noise is also good with SIR gain varying from 6 dB to 11 dB and SNR gain varying from 6 dB to 8 dB for different input SNRs.

Figures 7.2 and 7.3 provide the DNSMOS-signal and DNSMOS-noise scores, for the different types of noise and noise levels (input SNR). In the case of white noise, there is good noise reduction with an SNR gain of about 10dB in all cases. This is reflected in the DNSMOS values, where the noise scores stay above 2.0 even when the input SNR is as low as 1 dB. There is a significant improvement when compared with the sum of mics method (without MWF). But below 5dB SNR, there is considerable noise in the output even though there is an improvement in speech quality.

Compared to pink and green noise, red noise performs better since the DNSMOS noise score is higher than 2.0 even at input SNR close to zero. Pink noise has an SNR gain score similar to red noise, but the DNSMOS signal score indicates better speech quality in the case of red noise. There is only a slight improvement in the DNSMOS noise score for green and pink noise as input SNR gets closer to zero. For Hoth noise, there is a decent improvement in the DNSMOS noise scores and a significant improvement in the DNSMOS signal scores with MWF.

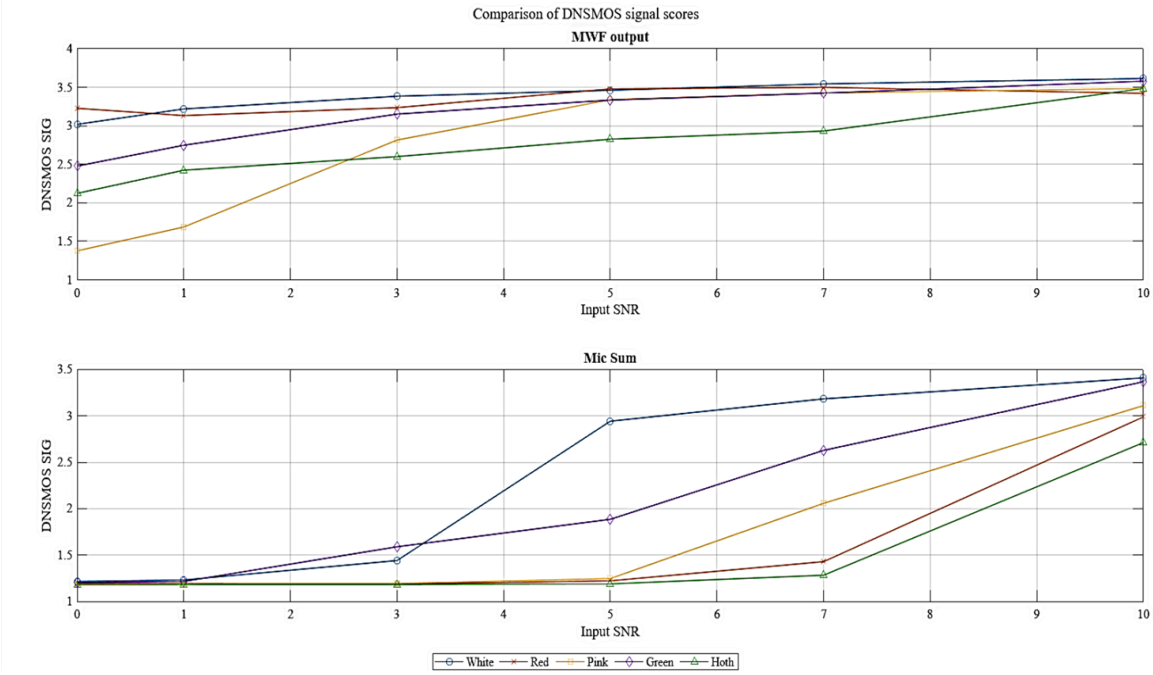

Figure 7.2 Comparing the performances of MWF and signal mixing (DNSMOS signal score)

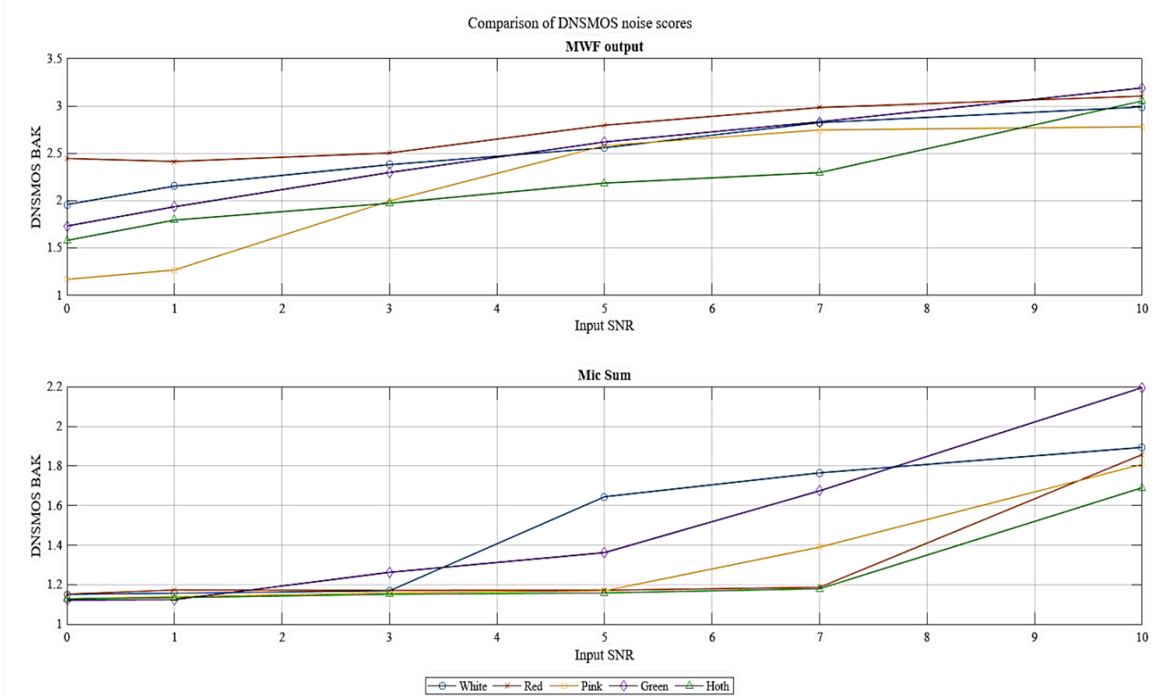

Figure 7.3 Comparing the performances of MWF and signal mixing (DNSMOS noise score)

The data tables used to generate the figures are given in the Appendix section A.2. It should be noted that the simulation results depend more on the acoustic paths (and their static or time-varying nature) than on the choice of audio source content. However, it is true that the absolute values of the different metrics used would change with different audio source content (but the ranking of the different scores for different conditions should not change). To show this, Figure 7.4 has been generated with a different set of speech signals but with the same activity pattern. It shows how there is consistent improvement in the DNSMOS signal and noise scores with MWF as compared to direct mixing, especially with increase in input SNR. Compared to the corresponding curves for white noise in Figures 7.2-7.3, we can see that the even though the absolute values of the scores are different, the ranking of scores for different conditions does not change.

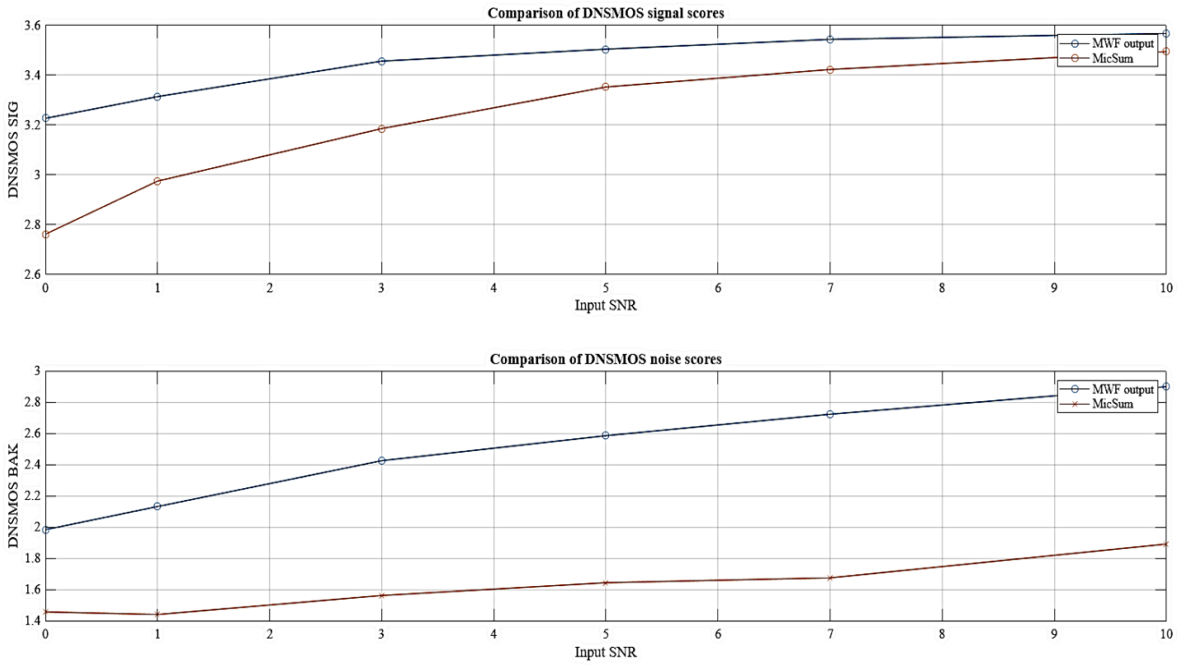

Figure 7.4 DNSMOS scores for different set of speech signals with white noise

7.4 MWF with wind noise

When testing with wind noise, it must be noted from Fig. 7.4 that the noise is highly nonstationary in amplitude levels and also with time varying level imbalance between the two mics. The results for wind noise can be found in Table A.11 from the Appendix and in Figure 7.5. Compared with the SNR gain and SIR gain for the simple mic signals mixing method which are -1.0 dB and -1.26 dB, respectively, the SNR gain values in Table A.11 range from 1.5 to 2 dB for driver and 4.6 to 5.2 dB for passenger and the SIR gain values are all above 6 dB. So even though these gain values are more modest than for the previous types of noise, the method of summing the MWF outputs is still beneficial. In terms of DNSMOS scores in Figure 7.5, the MWF method shows some improvement in the DNSMOS-noise scores, but a slight decrease in the DNSMOS-signal scores. So again, the performance is weaker than for the previous types of noise considered, because of the noise level and imbalance fluctuations. Therefore, for wind noise the MWF filters would need more time to adapt in order to deliver the same performance as for other noises.

Under no noise conditions, the ideal solutions that the MWFs can reach are entirely based on the acoustic transfer functions between the sources and the microphones. So, variations in source signals (e.g. time fluctuations or the non-stationary nature of their PSD) don't affect the ideal

performance achievable by the MWFs, i.e., the ideal MWF filters are static. The ideal MWFs also remain static when additive noise is present at the microphones and when the level of the different sources relative to the noise remain the same, i.e., when the signal to noise ratios between the desired source signals and additive noise remains the same. However, when either the levels of the source signals change or the levels of the background additive noise changes (at either microphone), the ideal MWF solution changes and the practical MWFs need to adapt to the new conditions. This is what happens in the case of wind noise, whose level at each microphone is highly non-stationary.

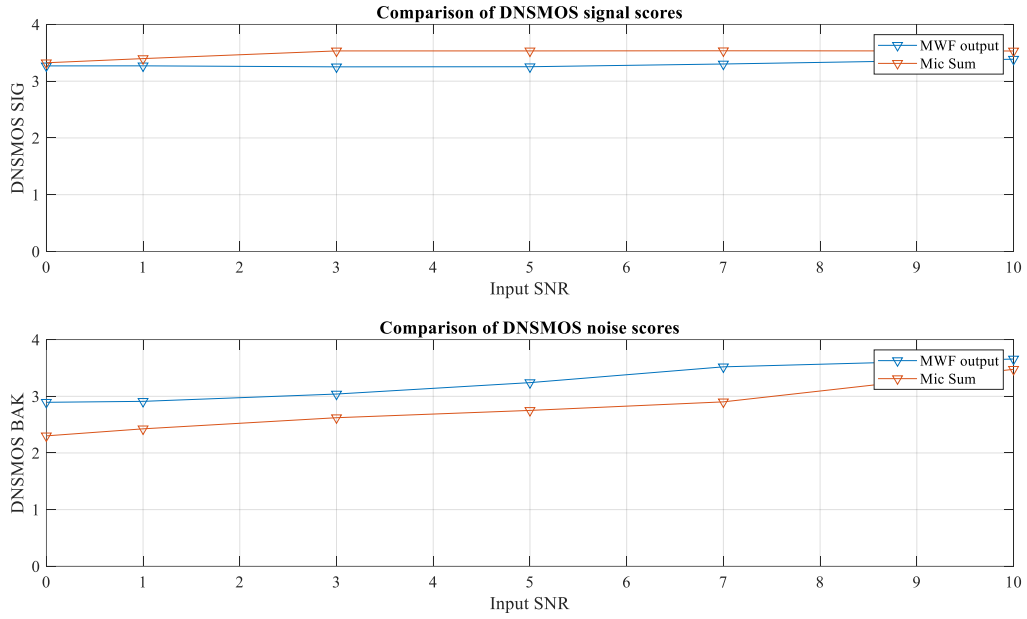

Figure 7.5 Comparing the performances of MWF and signal mixing for wind noise (DNSMOS signal and noise scores)

Chapter 8 MWF and Head Movement

A driver or a front seat passenger may have to turn or move their head during turns, lane changes, or out of general human behaviour. This calls for the evaluation of performance of the MWF system during such movements. The impulse response between a source and the microphones changes as the head moves, and this affects the values of the MWF filters coefficients, which need to adapt.

Our simulations assume point sources, i.e., the sources are non-directional. As a result, source rotation does not impact the simulated outcomes. To model changes in the source-to-microphone acoustic path, we chose to implement a translation of the source. The primary goal was not to capture variations in signal level or energy at the microphones, but rather to replicate the changes in the phase content across different frequencies that occur in the acoustic echo path when the source position varies. This phase variation was the key aspect we aimed to reproduce in our simulations.

In the configuration file used to generate the impulse responses, the positions of the speech/audio sources are as given below:

Table 8.1 Positions of different signal sources in the car

	Distance in front (m)	Distance from left door (m)	Distance from floor (m)
Driver	2.5	d	0.75
Passenger	2.5	1.4	0.75
Left speaker	2.2	0.15	0.3
Right speaker	2.2	1.85	0.3

The value of d used to generate the original (default) impulse responses is 0.6 m. During testing, the value of d was changed to 0.5 and 0.45 to simulate head movement by 10 cm and 15 cm.

The speech used for testing are divided into 3 segments after training, as shown in Table 8.2 and Figure 8.1.

Table 8.2 Changes in d during different time segments

	Segment 1	Segment 2	Segment 3
Time duration	20-36s	36-54s	54-68s
d (m)	0.6	0.5/0.45	0.6

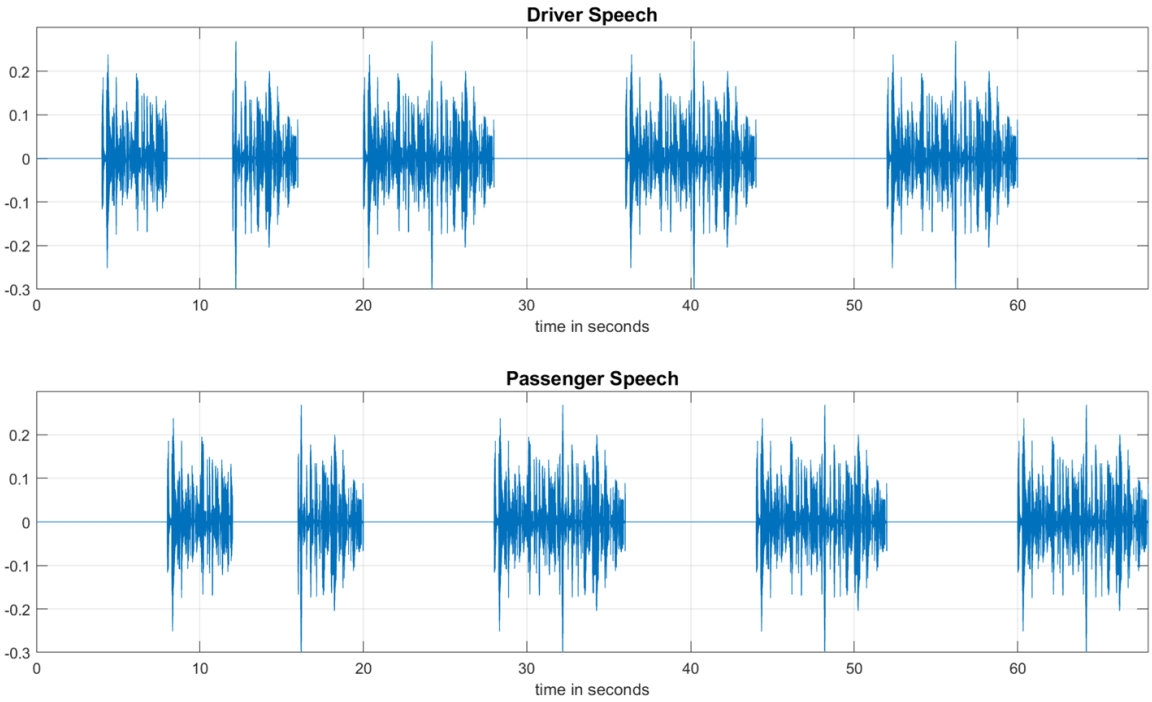

Figure 8.1 Speech signals used for testing

From the previous chapters/testing, it was observed that an 8ms frame size gave the best noise reduction for all noise types among the different frame sizes. For the frame size of 8ms, the regularization factor (δ) of 1.0 and the moving average factor (λ) of 0.96 are fixed for testing. The input SNR is varied from 10dB to 5dB to 1dB. This SNR is calculated based on the first segment of speech where the original impulse response without head movement is used. The metrics used for testing for this case are DNSMOS-signal (speech quality), DNSMOS-noise (background noise reduction), SIR gain and SNR gain.

The noises used for the tests are: white, red, pink, green, and Hoth. As before, for Hoth noise the regularization factor of 100.0 is used for frequencies less than 312.5 Hz and 1.0 for higher frequencies.

From the tables in Appendix section A.3, for white, coloured and Hoth noises, the pattern is similar. There is an improvement in both DNSMOS scores in the sum of MWF outputs method compared to the sum of 2 mic signals, even when there is head movement. This is more significant with lower SNR. If the adaptation of MWF is stopped after 36 s, i.e., the first speech segment, the performance scores suffer, especially when the noise level is higher. This shows that there needs to be continuous adaptation of MWF coefficients regardless of head movement.

8.1 Continuous vs paused adaptation of MWF

To see if we require continuous adaptation of MWF, we pause the adaptation after 36 seconds and see the effects in the last two segments of speech. So, we have 3 segments of speech with 2 setups (Table 8.3).

- MWF with continuous adaptation (MWF-AC)
- MWF with adaptation paused after 36 seconds (MWF-AS)

Table 8.3 Time segments during which MWF adapts and do not adapt

	20-36 seconds	36-52 seconds	52-68 seconds
MWF-AC	Adapting	Adapting+ head movement	Adapting
MWF-AS	Adapting	No Adapting +head movement	No Adapting

For each of the above cases, head displacements of 0.1m and 0.15m are considered by changing the value of d to 0.5m and 0.45m from 0.6m. But it can be seen that there is only a minute difference between their performances, meaning that a displacement of 0.1m or 0.15m has similar effect.

In the first segment, the MWF is adapting in both cases so that they start out with the same performance. The second segment helps evaluate the MWF in situations with head movement and assesses its performance with and without continuous adaptation. In the third segment, the head reverts to the original position. This helps us evaluate whether the performance can go back to the one from the first time segment, i.e., without adaptation.

DNSMOS is calculated for 16-second time segments with both driver and passenger speech. SIR and SNR gain are calculated for every 8 seconds of driver and passenger speech, separately in each of the 16 second time segments.

Figures 8.2 to 8.13 show the performance for different types of noise with input SNRs of 1dB, 5dB and 10dB for the above-mentioned cases. When the input SNR is high, there is less noise in the beginning; hence, there is less noise for the MWF to remove. This means that the SNR gain would be less for higher input SNR. Whereas DNSMOS evaluates perceptual speech quality, unlike raw SNR. Even though the SNR gain might be higher for low input SNR cases, residual distortions and artifacts from aggressive denoising can lower the perceptual quality. This can be noted in the figures, where the SIR and SNR gain may be higher for lower cases of lower input SNR, but the perceptual quality measured by DNSMOS is higher for higher input SNR for the same noise.

8.1.1 White noise

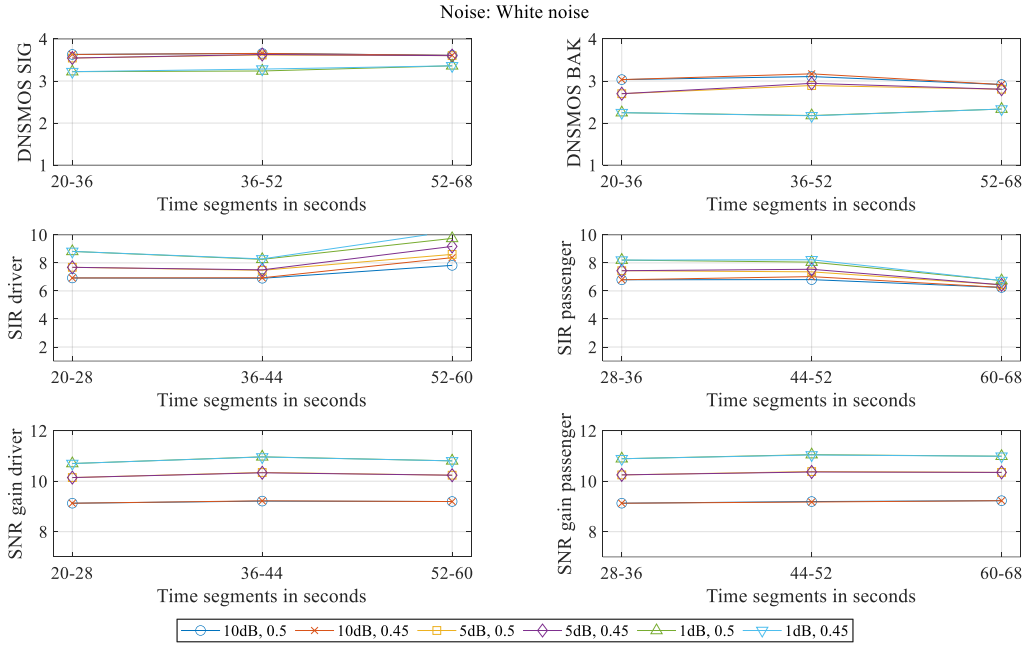

Figure 8.2 DNSMOS, SIR gain and SNR gain for MWF-AC for white noise

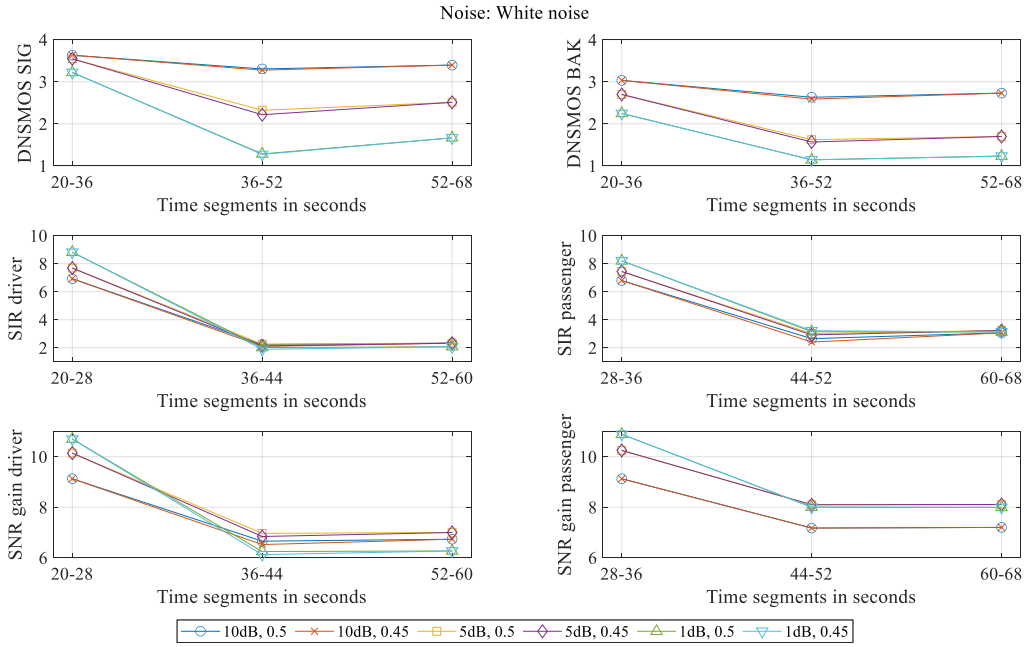

Figure 8.3 DNSMOS, SIR gain and SNR gain for MWF-AS for white noise

8.1.2 Red noise

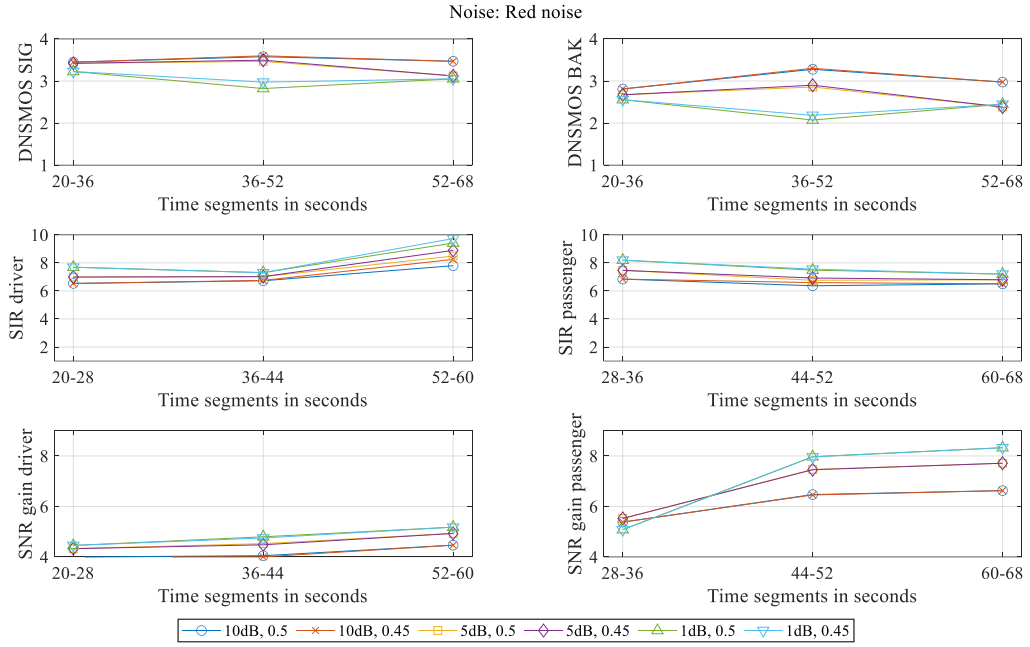

Figure 8.4 DNSMOS, SIR gain and SNR gain values for MWF-AC for red noise

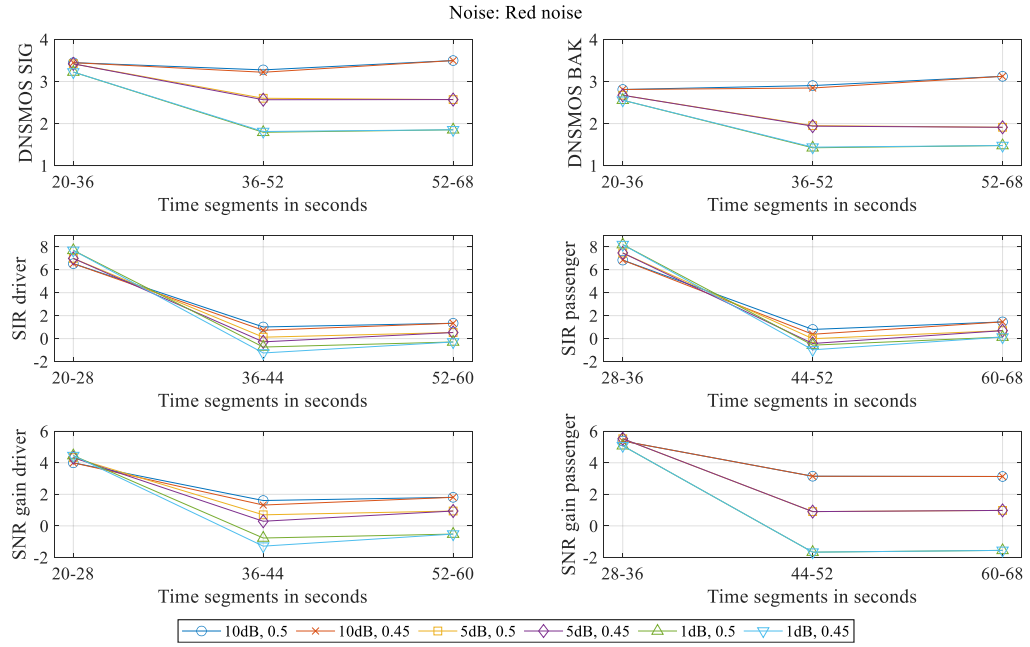

Figure 8.5 DNSMOS, SIR gain and SNR gain for MWF-AS for red noise

8.1.3 Pink noise

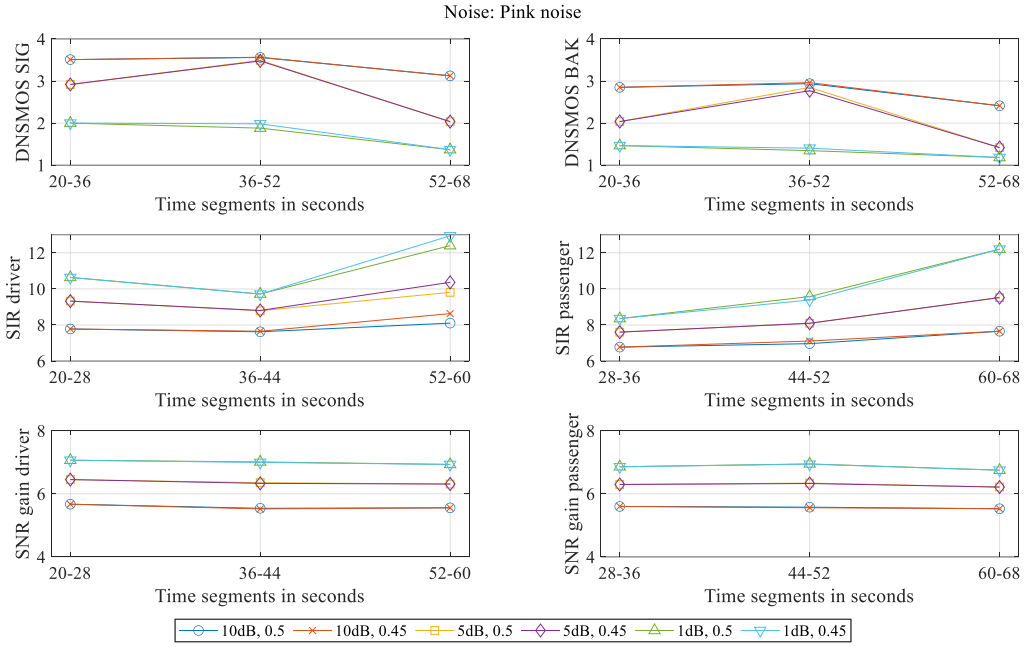

Figure 8.6 DNSMOS, SIR gain and SNR gain values for MWF-AC for pink noise

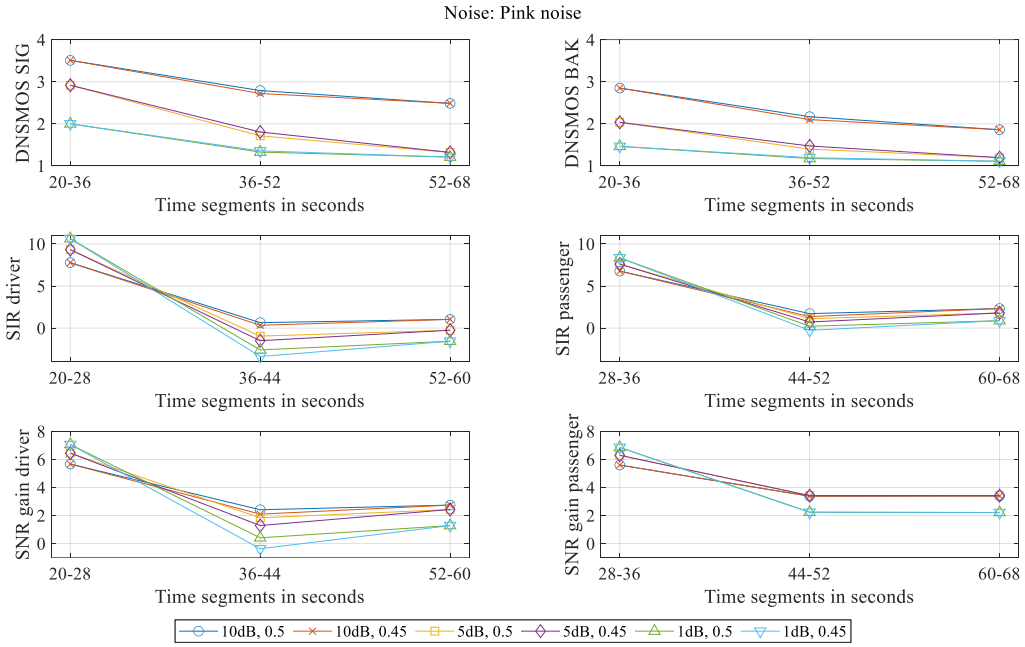

Figure 8.7 DNSMOS, SIR gain and SNR gain values for MWF-AS for pink noise

8.1.4 Green noise

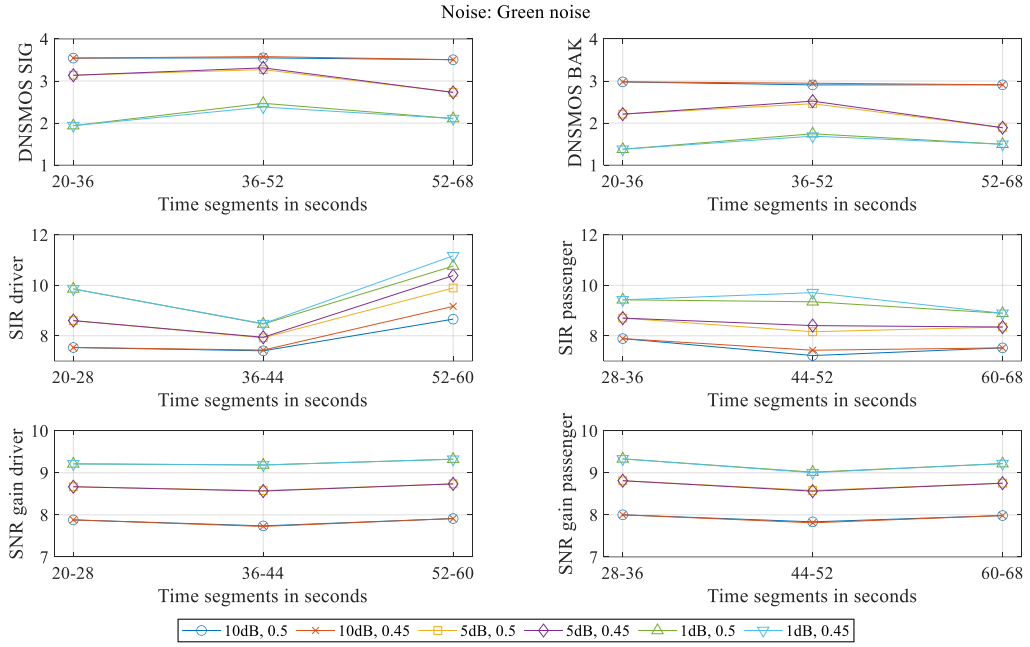

Figure 8.8 DNSMOS, SIR gain and SNR gain values for MWF-AC for green noise

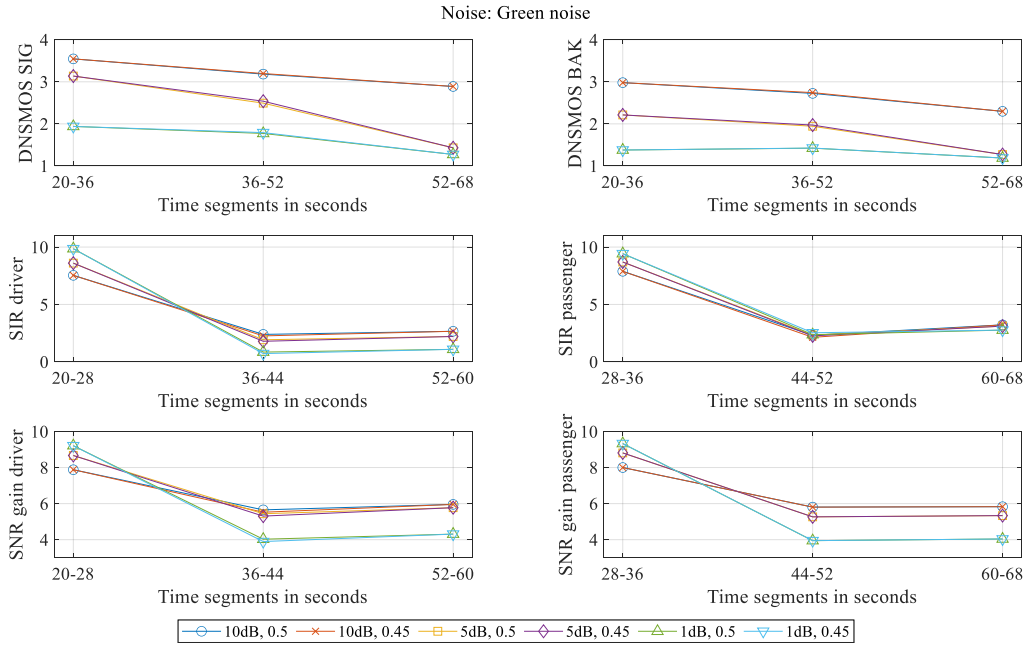

Figure 8.9 DNSMOS, SIR gain and SNR gain values for MWF-AS for green noise

8.1.5 Hoth noise

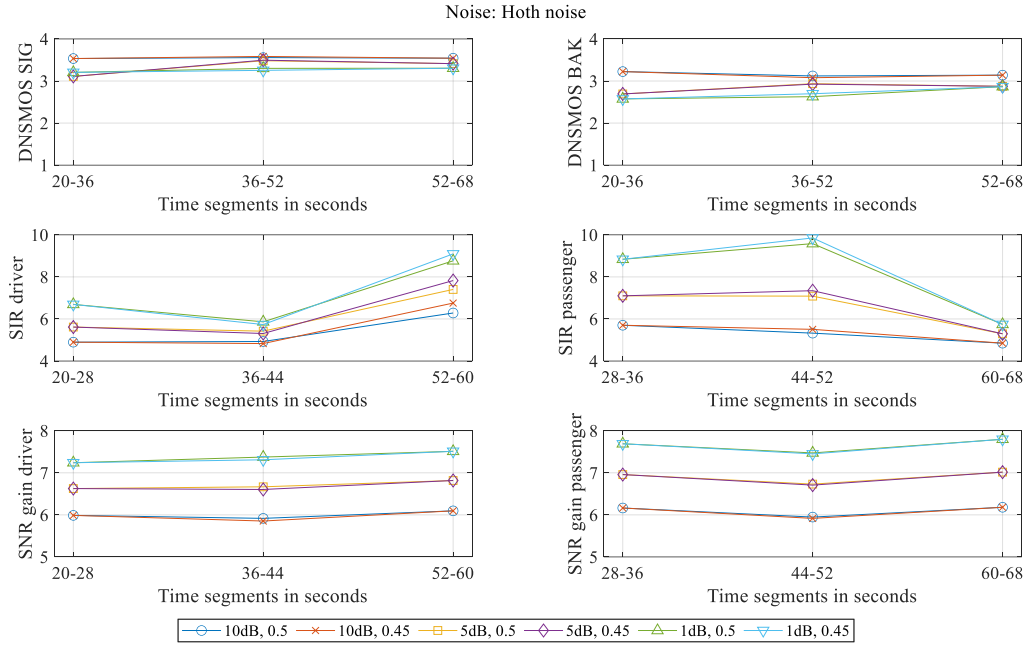

Figure 8.10 DNSMOS, SIR gain and SNR gain values for MWF-AC for Hoth noise

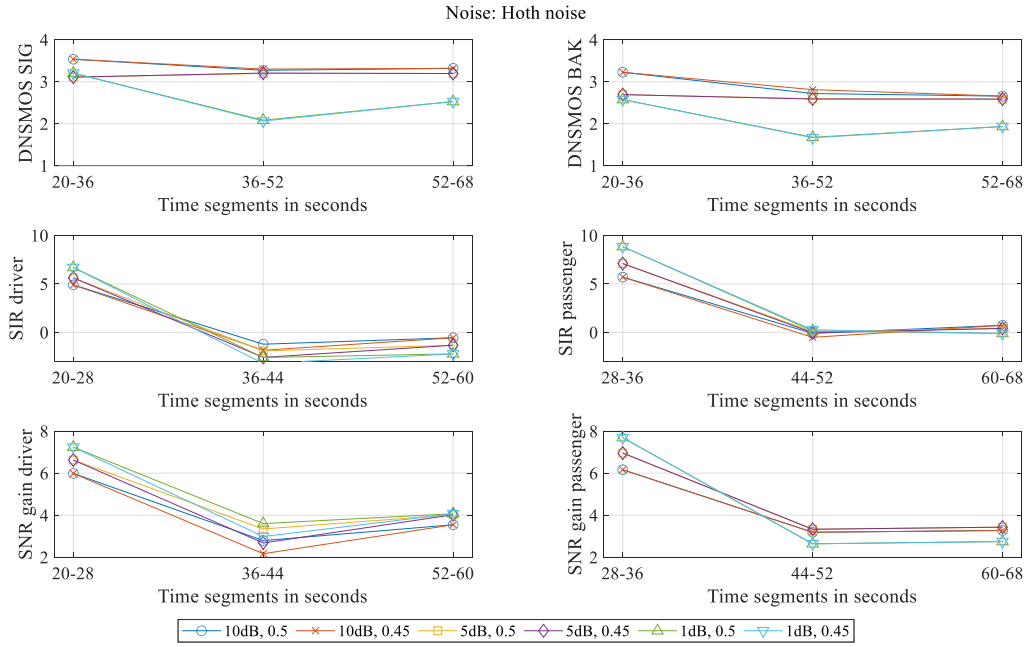

Figure 8.11 DNSMOS, SIR gain and SNR gain values for MWF-AS for Hoth noise

From the figures, it can be seen that with white, red, pink, green, and Hoth noise, the MWF shows similar trends. With continuous adaptation, the MWF holds steady performance for all the input SNRs as well as d values. In the case of MWF-AS, the performance falls once the adaptation is stopped after 36 seconds, regardless of the input SNR or head movement. This fall in metrics, especially perceptual quality, is further exacerbated by the increase in noise power. MWF adaptation is thus found to be imperative for good performance.

8.2 Comparison of MWF with direct mic signals mixing

In order to see if it is computationally worth to have MWF rather than direct mixing, we compare the performance when:

- MWF continuously adapts (MWF-AC)
- MWF stops adapting after 36 seconds (MWF-AS)
- No MWF (direct mic signal mixing) (MicSum).

To evaluate the performance, different noises with input SNRs 10db, 5dB and 1dB are used. DNSMOS-signal and DNSMOS-noise scores are plotted in the figures to visualise how the performance changes with different setups. For simulating the head movement, the value of d is changed from 0.6m to 0.5m, i.e., a 0.1 m displacement is made during the time segment 36-52 seconds. Experiments were carried out with d=0.45m as well, and the DNSMOS values were close to the ones with d=0.5m. The input SNR is calculated during 20-28 seconds for the driver and 28-36 seconds for the passenger, because these are the time durations when the driver and passenger are speaking, respectively.

The following figures compare the performances for the 3 mentioned cases with a head displacement of 0.1m (d=0.5m) for each type of noise.

8.2.1 White noise

When there is adaptation (MWF-AC), the signal and noise DNSMOS values for all 3 input SNRs are higher than for the sum of mic signals method. For 10dB input SNR, the signal score for MWF-AS is comparable to MicSum, but the noise score is better with MWF-AS, even though there is no adaptation. As the input SNR decreases, there is a stronger effect of noise on the performance, as the noise and signal scores fall when adaptation is stopped. For 5dB input SNR, it might better to directly mix the signal than have an MWF if there is no continuous adaptation (MWF-AS).

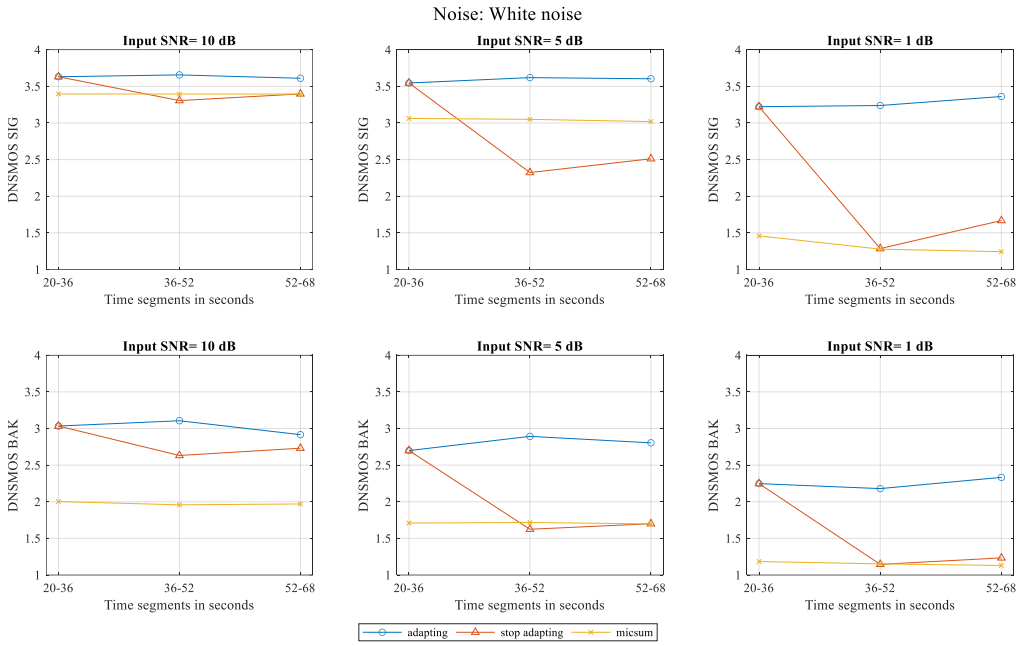

Figure 8.12 DNSMOS values for MWF-AC, MWF-AS, and MicSum for white noise

8.2.2 Red noise

For red noise with an input SNR of 10dB, the signal and noise scores slightly fall for MWF-AS during the head movement as compared to MWF-AC and is able to pick up after the head moves back to the original position. Even when the input SNR is 5dB and 1dB, and the signal and noise scores for MWF-AS fall considerably after the adaptation is stopped, it is still better than MicSum. This indicates that for red noise, the MWF outperforms MicSum regardless of adaptation.

8.2.3 Pink noise

For pink noise, irrespective of the input SNR, the signal and noise scores for MWF-AS is higher than for MicSum, and MWF-AC is has higher scores than both. This shows that MWF makes a difference in the performance in the presence of pink noise and continuous adaptation is required for steady performance.

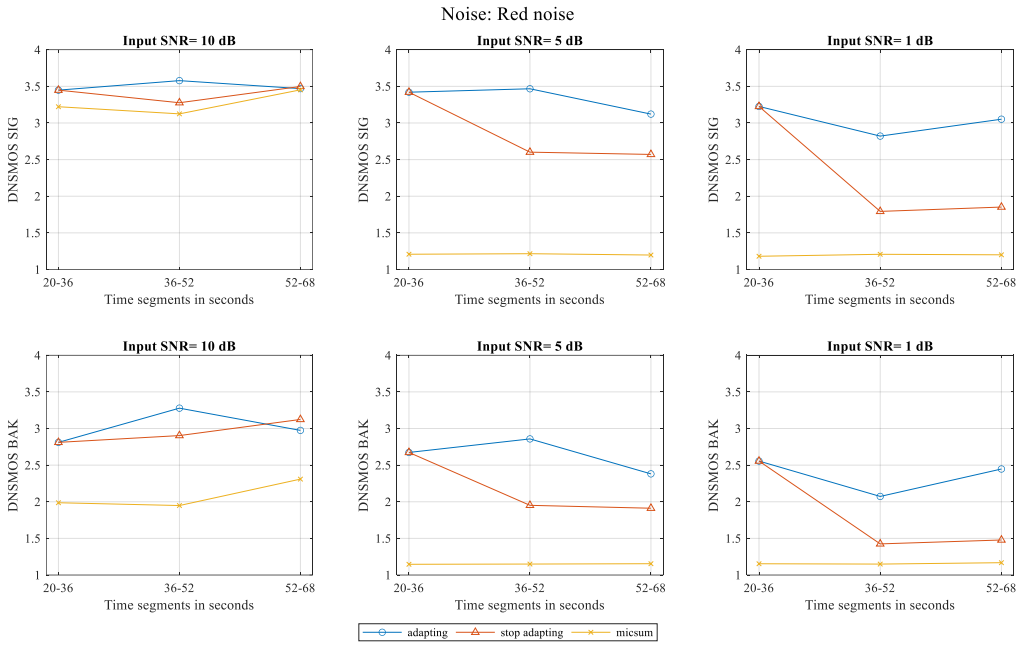

Figure 8.13 DNSMOS values for MWF-AC, MWF-AS, and MicSum for red noise

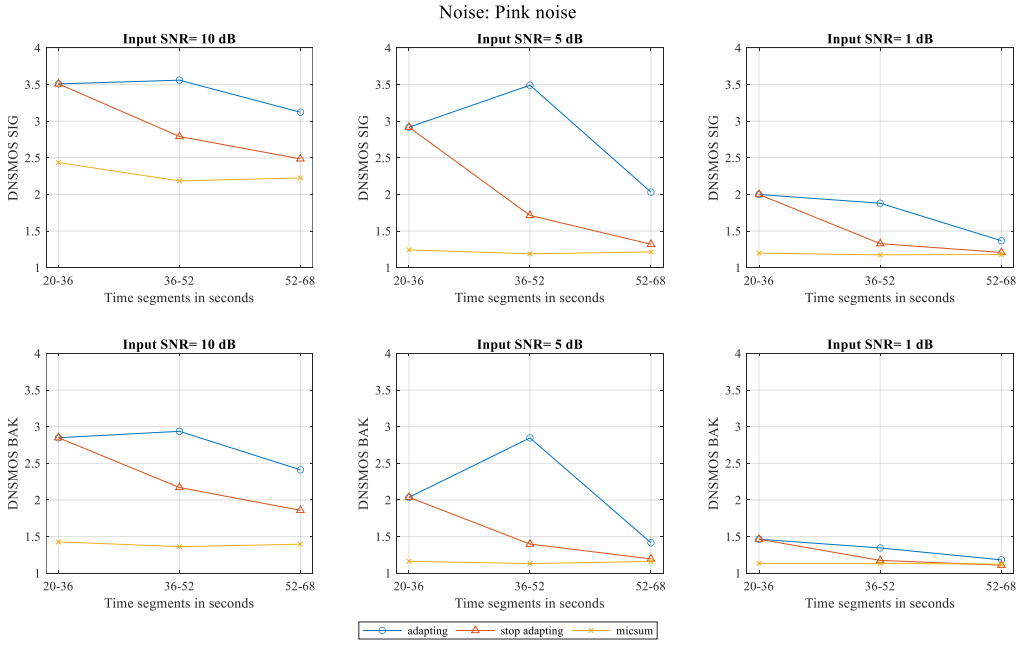

Figure 8.14 DNSMOS values for MWF-AC, MWF-AS, and MicSum for pink noise

8.2.4 Green noise

For green noise with input SNR 10dB, the signal score for MicSum is better than MWF-AS, even when there is head movement, but the noise score is lower. With higher noise power, the

performance of MicSum and MWF-AS get closer but is much poorer than MWF-AC. This shows that not only MWF, but continuous adaptation is also required with MWF for good performance with green noise.

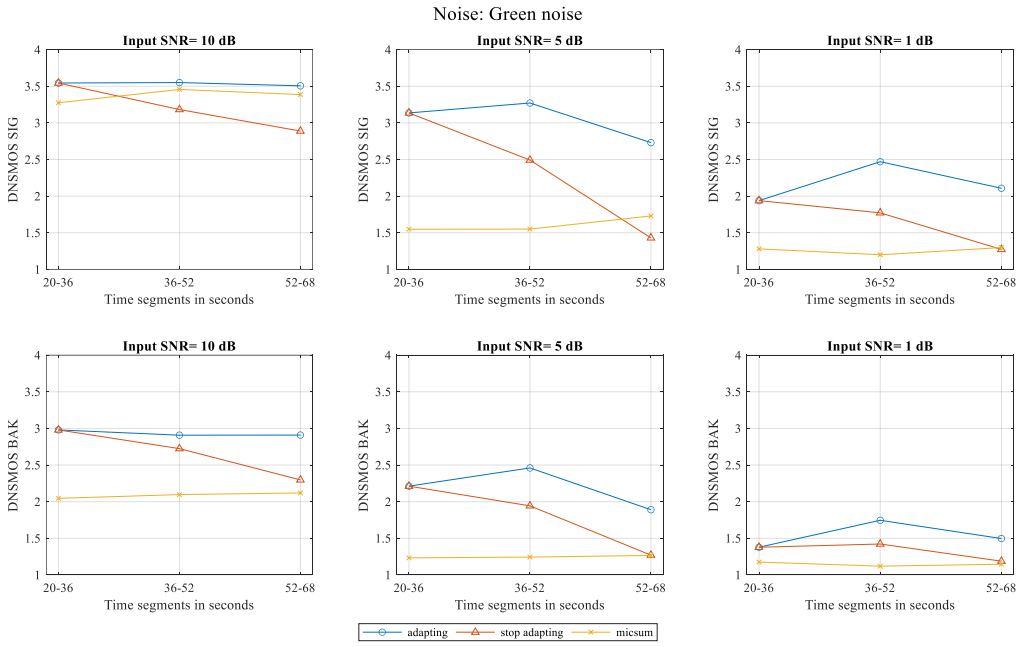

Figure 8.15 DNSMOS values for MWF-AC, MWF-AS, and MicSum for green noise

8.2.5 Hoth noise

The performance of the 3 setups with Hoth noise is shown in the Figure 8.18. In every case, it can be seen that the presence of MWF increases the performance considerably. Furthermore, for good, steady performance, continuous adaptation is required. MWF-AC easily outperforms the other methods at any noise level. This is significant since Hoth noise gives a type of noise that is generally present in communication systems.

8.3 Discussion

In summary, considering the performance of MWF for different types of noise, we can say that introducing an MWF with continuous adaptation in the presence of two microphones (each dedicated to driver and passenger) improves the speech quality compared to direct mixing of signals. The extra processing required for the MWF filters may be justified since it allows to use a single AEC module afterwards. There are other benefits, such as having access to individual

extracted driver and passenger signals if interference cancellation (source separation) is to be performed.

Chapter 9 Conclusion and Future work

9.1 Conclusion

In this thesis, the simple mixing (adding) and switching methods to combine two microphone signals into a single signal were first investigated, which showed drawbacks such as, the notch filtering effects and increased noise in the mixed/added signal, and decreased AEC performance when switching two microphone signals before a single AEC. To resolve these issues, we investigated the use adaptive MWF filters to extract separate driver and passenger signals without multi-path propagation effects before mixing them, thereby reducing notch filtering effects and minimizing background noise. The effect of head movements of driver/passenger was also investigated.

The MWF method proved effective in a simulated car environment, where each front microphone is mostly dedicated to a front seat speaker, which makes the task of estimating the signal correlations easier (i.e., it becomes easier to detect when each source is active and to estimate the related correlations). The MWF method successfully removed notch filtering effects, particularly with larger frame sizes. Through multichannel linear filtering, which is nearly LTI, the MWF effectively reduced background noise with minimal distortion (unlike single channel speech enhancement methods, which introduce more distortion). Since the number of simultaneously active acoustic sources (talkers or loudspeakers) that MWF filters can extract is limited by the number of microphones, MWFs can be disabled when signals are played through the loudspeaker signals, or, alternatively and at higher cost, AEC could be performed on each microphone signal to remove loudspeaker signal components, before the remaining driver and passenger source signals are processed by the MWFs. The MWF filters can perform driver and passenger speech extraction whether only one of them is talking or whether they are talking simultaneously. In the latter case, since the MWF extracts the speech sources before mixing them, it can also be used for interference cancellation (i.e., extracting one speech source and cancelling the competing talker(s)).

Talker movements can potentially impact MWF performance, but in such cases the MWF has shown to adapt quickly without divergence. The performance of MWF filtering can also be limited by highly non-stationary background noise such as wind noise, with time-varying spectra and time-varying signal level imbalance between microphones.

9.2 Future work

Wind noise is one of the unpredictable, level-imbalanced time-varying noises that pose a challenge to noise reduction in certain high-noise environments. Since MWF depends heavily on the computed statistics of the signal, its performance can deteriorate in such situations. A possible approach to solve this would be using deep learning with an ample amount of recorded wind noise to train the noise suppressor.

It is essential to validate the proposed approach in real-world automotive environments. This includes conducting experiments within actual vehicle cabins with real time audio sources to account for the complex and variable acoustic conditions present in practical scenarios. Additionally, system performance has to be evaluated under varying driving conditions, such as different vehicle speeds and accompanying noise levels, including road, wind, and engine noise.

The different components of the speech/audio processing pipeline in cars, such as acoustic echo canceller (AEC), acoustic echo suppressor (AES), noise suppression (NS), and the proposed multichannel Wiener filters rely mostly on traditional filtering or noise suppression. Some approaches have been proposed to replace them fully with deep learning methods that have the potential to cope better with non-linearities and non-stationarities (e.g. if a model is already trained to be robust to different types of environments and signal statistics). This includes the so-called end-to-end (E2E) methods [43]. Other methods have been proposed based on hybrid approaches, where the machine learning part is dedicated to performing some specific tasks in traditional signal processing. For example, in AEC some methods have been proposed to take care of the control unit and step size adjustment. For beamforming (or equivalently, MWF), some methods have been proposed to perform the estimation of the noise covariance matrix, etc. The hybrid approach is likely to lead to methods that have lower complexity and can be applied in the near future.

References

[1] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction Wiener filter," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1218–1234, Jul. 2006, doi: 10.1109/TSA.2005.860851.

[2] G. Wang, C. Li, and L. Dong, "Noise Estimation Using Mean Square Cross Prediction Error for Speech Enhancement," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1489–1499, Jul. 2010, doi: 10.1109/TCSI.2010.2054930.

[3] I. Y. Soon and S. N. Koh, "Speech enhancement using 2-D Fourier transform," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 717–724, Nov. 2003, doi: 10.1109/TSA.2003.816063.

[4] H. Ding, I. Y. Soon, S. N. Koh, and C. K. Yeo, "A spectral filtering method based on hybrid wiener filters for speech enhancement," *Speech Communication*, vol. 51, no. 3, pp. 259–267, Mar. 2009, doi: 10.1016/j.specom.2008.09.003.

[5] B. Widrow *et al.*, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975, doi: 10.1109/PROC.1975.10036.

[6] M. A. Abd El-Fattah *et al.*, "Speech enhancement with an adaptive Wiener filter," *Int J Speech Technol*, vol. 17, no. 1, pp. 53–64, Mar. 2014, doi: 10.1007/s10772-013-9205-5.

[7] K. Garg and G. Jain, "A comparative study of noise reduction techniques for automatic speech recognition systems," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2016, pp. 2098–2103. doi: 10.1109/ICACCI.2016.7732361.

[8] M. Fukui, T. Watanabe, and M. Kanazawa, "Sound Source Separation for Plural Passenger Speech Recognition in Smart Mobility System," *IEEE Trans. Consumer Electron.*, vol. 64, no. 3, pp. 399–405, Aug. 2018, doi: 10.1109/TCE.2018.2867801.

[9] Y.-H. Chen, S.-J. Ruan, and T. Qi, "An automotive application of real-time adaptive Wiener filter for non-stationary noise cancellation in a car environment," in *2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)*, Hong Kong, China: IEEE, Aug. 2012, pp. 597–601. doi: 10.1109/ICSPCC.2012.6335628.

[10] T. Z. Qi and T. J. Moir, "A hybrid noise canceller with a real-time adaptive Wiener filter and a geometric-based voice–activity detector for an automotive application," *Adaptive Control & Signal*, vol. 24, no. 6, pp. 508–522, Jun. 2010, doi: 10.1002/acs.1146.

[11] M. Tsujikawa, T. Arakawa, and R. Isotani, "In-car speech recognition using model-based wiener filter and multi-condition training," in *Interspeech 2008*, ISCA, Sep. 2008, pp. 972–975. doi: 10.21437/Interspeech.2008-284.

[12] J.-T. Chien and P.-Y. Lai, "Car Speech Enhancement Using a Microphone Array" *International Journal of Speech Technology,* 8 (2005), pp.79-91.

[13] M. B. Trawicki and M. T. Johnson, "Multichannel MMSE Wiener Filter Using Complex Real and Imaginary Spectral Coefficients for Distributed Microphone Speech Enhancement," *International Journal of Theoretical and Applied Mathematics (IJTAM)*, vol. 2, no. 2, Dec. 2016, pp.115-120.

[14] S. Gannot and I. Cohen, "Adaptive Beamforming and Postfiltering," in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. A. Huang, Eds., Berlin, Heidelberg: Springer, 2008, pp. 945–978. doi: 10.1007/978-3-540-49127-9_47.

[15] T. Van den Bogaert, S. Doclo, J. Wouters, and M. Moonen, “Speech enhancement with multichannel Wiener filter techniques in multimicrophone binaural hearing aids,” *J Acoust Soc Am*, vol. 125, no. 1, pp. 360–371, Jan. 2009, doi: 10.1121/1.3023069.
[16] N. Modhave, Y. Karuna, and S. Tonde, “Design of multichannel wiener filter for speech enhancement in hearing aids and noise reduction technique,” in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, Coimbatore, India: IEEE, Nov. 2016, pp. 1–4. doi: 10.1109/GET.2016.7916626.
[17] J. Meyer and K. U. Simmer, “Multi-channel speech enhancement in a car environment using Wiener filtering and spectral subtraction,” in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany: IEEE Comput. Soc. Press, 1997, pp. 1167–1170. doi: 10.1109/ICASSP.1997.596150.
[18] C. Fox, G. Vitte, M. Charbit, J. Prado, R. Badeau, and B. David, “A subband hybrid beamforming for in-car speech enhancement”, *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, pp. 11-15, Aug. 2012.
[19] S. Stenzel and J. Freudenberger, “On the Speech Distortion Weighted Multichannel Wiener Filter for diffuse Noise,” in *Speech Communication; 10. ITG Symposium*, Sep. 2012, pp. 1–4.
[20] M. Gimm, F. Kühne, and G. Schmidt, “10 A Multichannel Spatial Hands-Free Application for In-Car Communication Systems,” in *Towards Human-Vehicle Harmonization*, De Gruyter, 2023, pp. 129–140.
[21] B. Kaulen, J. Abshagen and Gerhard Schmidt, "Multichannel Wiener filter in active sound-navigation-and-ranging systems—A joint beamformer and matched filter approach", *IET Radar, Sonar & Navigation*, vol. 18, no. 9, Sept. 2024, pp.1554-1569.
[22] S. Grimm and J. Freudenberger, “Wind noise reduction for a closely spaced microphone array in a car environment,” *J Audio Speech Music Proc.*, vol. 2018, no. 1, p. 7, Dec. 2018, doi: 10.1186/s13636-018-0130-z.
[23] T. Matheja, M. Buck, and T. Fingscheidt, “A dynamic multi-channel speech enhancement system for distributed microphones in a car environment,” *EURASIP J. Adv. Signal Process.*, vol. 2013, no. 1, p. 191, Dec. 2013, doi: 10.1186/1687-6180-2013-191.
[24] A.R. George, “Automobile Aeroacoustics,” *Proc. of AIAA 12th Aeroacoustic Conference*, San Antonio, TX, U.S.A., pp.1-6, Apr. 1989 https://doi.org/10.2514/6.1989-1067
[25] H. Kuttruff, *Room acoustics*, 4th ed., Transferred to digital printing. London New York: Taylor & Francis, 2006.
[26] “RIR Generator.” Accessed: Apr. 17, 2025. [Online]. Available: https://github.com/ehabets/RIR-Generator
[27] J.B. Allen and D. A. Berkley. "Image method for efficiently simulating small-room acoustics." The Journal of the Acoustical Society of America, vol. 65, no. 4, pp.943-950, Apr. 1979.
[28] J. Shynk, Frequency Domain and Multirate Adaptive Filtering, IEEE Signal Processing, 1992
[29] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, Salt Lake City, UT, USA: IEEE, 2001, pp. 749–752. doi: 10.1109/ICASSP.2001.941023.

[30] ITU-T, Recommendation P.862.2 Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs, International Telecommunication Union, Geneva, Switzerland, Jan. 2007.
[31] M. Purin, S. Sootla, M. Sponza, A. Saabas, and R. Cutler, "AECMOS: A speech quality assessment metric for echo impairment," Jan. 27, 2022, *arXiv*: arXiv:2110.03010. doi: 10.48550/arXiv.2110.03010.
[32] ITU-T Recommendation P.831 Subjective performance evaluation of network echo cancellers. Accessed: Apr. 11, 2025. [Online]. Available: http://rfc.nop.hu/itu7/P/P0831e.pdf
[33] ITU-T Recommendation P.832 Subjective performance evaluation of hands-free terminals, Accessed: Apr. 11, 2025. [Online]. Available: https://img.antpedia.com/standard/pdf/L61/1705/ITU-T%20P.832-2000_en.pdf
[34] ITU-T Recommendation P.808 Subjective evaluation of speech quality with a crowdsourcing approach Accessed: Apr. 11, 2025. [Online]. Available: https://www.itu.int/rec/dologin_pub.asp?lang=f&id=T-REC-P.808-202106-I!!PDF-E&type=items
[35] 3QUEST: Applications & Use in Practice, Accessed: Apr. 11, 2025. [Online]. Available: https://cdn.head-acoustics.com/fileadmin/data/global/Application-Notes/Telecom/3QUEST-Applications-and-Use-In-Practice-Application-Note.pdf
[36] HEAD Acoustics Data Sheet ACOPT 32(Code 6859) Speech-based Double Talk. Accessed: Apr. 18, 2025. [Online]. Available: https://cdn.head-acoustics.com/fileadmin/data/global/Datasheets/Analysis_Software/ACQUA_Options/ACOPT-32-Speech-based-Double-Talk-ACQUA-Option-6859-Data-Sheet.pdf
[37] C. K. A. Reddy, V. Gopal, and R. Cutler, "Dnsmos P.835: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 886–890. doi: 10.1109/ICASSP43922.2022.9746108.
[38] ITU-T Recommendation P.835 Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm, Accessed: Apr. 11, 2025. [Online]. https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.835-200311-I!!PDF-E&type=items
[39] D. F. Hoth, "Room Noise Spectra at Subscribers' Telephone Locations," *The Journal of the Acoustical Society of America*, vol. 12, no. 4, pp. 499–504, Apr. 1941, doi: 10.1121/1.1916129.
[40] H. Zhivomirov, "A Method for Colored Noise Generation", *Romanian journal of acoustics and vibration*, vol. 15, no. 1, pp.14-19, Aug. 2018.
[41] Oscillator and Signal Generator. Accessed: Apr. 19, 2025. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/37376-oscillator-and-signal-generator
[42] C. Borss and R. Martin, "On the construction of window functions with constant-overlap-add constraint for arbitrary window shifts," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan: IEEE, Mar. 2012, pp. 337–340. doi: 10.1109/ICASSP.2012.6287885.
[43] T. Haubner, A. Brendel and W. Kellermann, "End-to-end deep learning-based adaptation control for linear acoustic echo cancellation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 227-38, Oct. 2023.

Appendix A: Data Tables

A.1 Regularization factor and forgetting factor for the MWF filters

This section covers the data tables used to find the appropriate regularization factor δ and forgetting factor λ for the MWF filters.

Each table is for a different type of noise. It goes through the different values of frame size, λ and δ , as defined before. In the case of Hoth noise, the δ values are fixed at 100.0 for frequencies less than 312.5 Hz and 1.0 for higher frequencies.

Table A.1 Effect of δ and λ values for white noise

Frame size	$\boldsymbol{\delta}$	$\boldsymbol{\lambda}$	**SIR gain driver**	**SIR gain passenger**	**SNR gain driver**	**SNR gain passenger**	**DNSMOS overall**	**DNSMOS signal**	**DNSMOS noise**
0.008	1	0.96	14.4	16.85	9.16	9.17	2.68	3.58	3.07
0.008	0.01	0.984	16.52	17.74	6.34	6.59	2.63	3.61	2.85
0.008	0.1	0.984	15.24	17.16	7.09	7.28	2.68	3.63	2.93
0.008	1	0.984	11.42	14.07	8.56	8.66	2.73	3.62	3.11
0.008	0.01	0.9886	15.86	17.07	6.07	6.36	2.61	3.60	2.81
0.008	0.1	0.9886	14.49	16.26	6.76	7.00	2.67	3.63	2.91
0.008	1	0.9886	10.56	12.82	8.19	8.34	2.73	3.63	3.11
0.008	0.01	0.9911	15.33	16.5	5.87	6.18	2.59	3.60	2.8
0.008	0.1	0.9911	13.86	15.45	6.51	6.78	2.66	3.63	2.9
0.008	1	0.9911	9.83	11.65	7.86	8.09	2.72	3.63	3.1
0.024	1	0.9657	11.87	13.64	8.41	8.61	2.68	3.64	3.01
0.024	0.1	0.9733	16.3	17.61	6.6	6.86	2.58	3.63	2.75
0.024	1	0.9733	11.12	12.61	8.15	8.43	2.68	3.63	3.02
0.024	0.1	0.976	16.08	17.41	6.58	6.85	2.58	3.62	2.75
0.024	1	0.976	10.77	12.11	8.03	8.34	2.72	3.64	3.07
0.024	0.01	0.984	18.46	19.42	5.49	5.72	2.53	3.61	2.65
0.024	0.1	0.984	15.03	16.24	6.21	6.50	2.57	3.61	2.75
0.024	1	0.984	9.25	10.06	7.48	7.90	2.72	3.62	3.08
0.024	0.01	0.988	17.95	18.89	5.28	5.49	2.50	3.60	2.6
0.024	0.1	0.988	14.27	15.35	5.92	6.21	2.57	3.61	2.75
0.024	1	0.988	8.12	8.68	7.04	7.53	2.72	3.61	3.09
0.04	1	0.9733	9.54	10.04	7.44	7.95	2.65	3.57	3.05
0.04	1	0.98	8.48	8.86	7.09	7.62	2.66	3.58	3.01
0.064	1	0.9744	7.72	7.81	6.12	7.2	2.55	3.50	2.9
0.064	1	0.9872	5.66	5.82	5.64	6.30	2.72	3.60	3.09
0.08	1	0.984	5.67	5.72	5.59	6.31	2.6	3.53	2.99

Table A.2 Effect of δ and λ values for red noise

Frame size	δ	λ	SIR gain driver	SIR gain passenger	SNR gain driver	SNR gain passenger	DNSMOS overall	DNSMOS signal	DNSMOS noise
0.008	0.01	0.96	18.09	18.92	3.93	4.19	2.56	3.53	2.81
0.008	0.1	0.96	17.26	18.66	4.72	5.13	2.65	3.55	2.97
0.008	1	0.96	14.59	16.8	6.27	7	2.71	3.44	3.29
0.008	0.01	0.984	16.13	17.39	3.50	3.89	2.52	3.53	2.70
0.008	0.1	0.984	14.83	16.68	4.31	4.83	2.57	3.55	2.82
0.008	1	0.984	11.43	13.94	5.89	6.77	2.77	3.45	3.35
0.008	0.01	0.9886	15.52	16.74	3.38	3.75	2.54	3.54	2.74
0.008	0.1	0.9886	14.10	15.79	4.18	4.66	2.59	3.54	2.85
0.008	1	0.9886	10.52	12.66	5.77	6.57	2.79	3.47	3.36
0.024	1	0.9657	11.95	13.57	8.48	8.85	2.59	3.40	3.12
0.024	0.1	0.9733	16.31	17.45	6.14	6.34	2.67	3.57	2.98
0.024	1	0.9733	11.17	12.52	8.46	8.88	2.62	3.42	3.14
0.024	0.01	0.976	19.56	20.19	4.57	4.70	2.48	3.49	2.69
0.024	0.1	0.976	16.07	17.23	6.17	6.36	2.64	3.56	2.93
0.024	1	0.976	10.80	12.02	8.43	8.85	2.59	3.39	3.12
0.024	0.01	0.984	18.94	19.9	4.62	4.77	2.45	3.46	2.64
0.024	0.1	0.984	15.03	16.11	5.98	6.19	2.54	3.51	2.8
0.024	1	0.984	9.25	9.96	8.20	8.62	2.48	3.30	2.97
0.024	0.01	0.988	18.45	19.37	4.49	4.64	2.44	3.43	2.65
0.024	0.1	0.988	14.27	15.23	5.75	5.96	2.5	3.45	2.77
0.024	1	0.988	8.10	8.59	7.93	8.36	2.38	3.24	2.83
0.064	1	0.9744	7.79	7.76	0.05	3.69	2.10	2.92	2.51
0.064	1	0.9872	5.64	5.78	7.11	7.54	2.05	2.87	2.43
0.08	1	0.984	5.69	5.73	6.06	6.75	2.63	3.54	2.85

Table A.3 Effect of δ and λ values for pink noise

Frame size	δ	λ	SIR gain driver	SIR gain passenger	SNR gain driver	SNR gain passenger	DNSMOS overall	DNSMOS signal	DNSMOS noise
0.008	1	0.96	14.30	17	6.35	6.91	2.63	3.54	3.02
0.008	0.01	0.984	16.04	17.47	2.83	3.15	2.34	3.46	2.40
0.008	0.1	0.984	14.72	16.78	3.83	4.22	2.4	3.47	2.51
0.008	1	0.984	11.3	14.06	5.60	6.21	2.71	3.56	3.13
0.008	0.01	0.9886	15.43	16.82	2.54	2.86	2.30	3.44	2.33
0.008	0.1	0.9886	14.00	15.89	3.52	3.89	2.36	3.47	2.42
0.008	1	0.9886	10.43	12.77	5.31	5.88	2.66	3.54	3.09
0.024	1	0.952	12.59	14.49	6.56	7.2	2.36	3.32	2.65
0.024	1	0.9657	11.74	13.62	6.74	7.08	2.41	3.35	2.70
0.024	0.01	0.9733	18.81	19.82	3.22	3.28	2.27	3.40	2.34
0.024	0.1	0.9733	15.83	17.39	4.53	4.68	2.41	3.51	2.50
0.024	1	0.9733	10.99	12.58	6.47	6.85	2.44	3.4	2.73
0.024	0.01	0.976	18.67	19.79	3.17	3.24	2.29	3.43	2.36
0.024	0.1	0.976	15.67	17.15	4.41	4.57	2.4	3.50	2.50
0.024	1	0.976	10.64	12.07	6.35	6.73	2.46	3.44	2.76
0.024	0.01	0.984	18.04	19.28	2.81	2.93	2.28	3.43	2.30
0.024	0.1	0.984	14.64	15.98	3.92	4.11	2.37	3.48	2.46
0.024	1	0.984	9.13	10.01	5.81	6.23	2.46	3.44	2.76
0.024	0.01	0.988	17.54	18.71	2.55	2.67	2.27	3.42	2.31
0.024	0.1	0.988	13.91	15.08	3.58	3.77	2.35	3.47	2.43
0.024	1	0.988	8.00	8.63	5.39	5.85	2.36	3.36	2.61
0.04	1	0.9733	9.4	9.98	5.97	6.47	2.41	3.38	2.70
0.04	1	0.98	8.34	8.8	5.63	6.12	2.15	3.1	2.32
0.04	0.1	0.984	13.73	14.98	3.49	3.65	2.27	3.4	2.33
0.04	1	0.984	7.52	7.9	5.31	5.82	2.14	3.1	2.34
0.064	0.1	0.9872	12.4	12.76	2.3	2.86	2.05	3.12	2.06
0.064	1	0.9872	5.57	5.77	4.42	5	2.03	3.03	2.2
0.08	1	0.984	5.59	5.65	4.38	5.02	2.06	3.05	2.22

Table A.4 Effect of δ and λ values for green noise

Frame size	$\boldsymbol{\delta}$	$\boldsymbol{\lambda}$	**SIR gain driver**	**SIR gain passenger**	**SNR gain driver**	**SNR gain passenger**	**DNSMOS overall**	**DNSMOS signal**	**DNSMOS noise**
0.008	0.01	0.96	18.69	19.38	5.58	5.86	2.4	3.45	2.59
0.008	0.1	0.96	17.71	19.13	6.48	6.84	2.56	3.51	2.88
0.008	1	0.96	14.54	17.00	8.05	8.57	2.78	3.59	3.23
0.008	0.01	0.984	16.54	17.91	4.90	5.72	2.39	3.46	2.57
0.008	0.1	0.984	15.18	17.19	5.72	6.67	2.54	3.52	2.82
0.008	1	0.984	11.43	14.08	7.15	8.43	2.67	3.59	3.05
0.008	0.01	0.9886	15.86	17.23	4.67	5.55	2.37	3.45	2.54
0.008	0.1	0.9886	14.42	16.28	5.44	6.46	2.48	3.5	2.71
0.008	1	0.9886	10.55	12.8	6.80	8.18	2.65	3.59	3
0.024	1	0.952	12.54	12.46	2.84	7.56	2.53	3.48	2.90
0.024	1	0.9657	11.85	13.59	7.29	8.65	2.66	3.57	3.05
0.024	0.1	0.9733	16.21	17.67	5.66	6.63	2.43	3.49	2.65
0.024	1	0.9733	11.09	12.55	7.02	8.43	2.67	3.59	3.05
0.024	0.01	0.976	19.17	20.17	4.60	5.32	2.23	3.37	2.37
0.024	0.1	0.976	15.97	17.43	5.57	6.54	2.42	3.49	2.62
0.024	1	0.976	10.73	12.05	6.89	8.31	2.66	3.59	3.04
0.024	0.01	0.984	18.45	19.8	4.30	4.96	2.22	3.38	2.34
0.024	0.1	0.984	14.93	16.24	5.12	6.03	2.34	3.47	2.49
0.024	1	0.984	9.20	109	6.32	7.77	2.6	3.56	2.93
0.024	0.01	0.988	17.9	19.21	4.00	4.59	2.20	3.36	2.3
0.024	0.1	0.988	14.18	15.34	4.77	5.59	2.35	3.46	2.53
0.024	1	0.988	8.07	8.63	5.88	7.32	2.63	3.56	2.95
0.04	1	0.9733	9.49	9.99	6.36	7.82	2.59	3.53	2.95
0.04	1	0.98	8.42	8.8	6	7.43	2.56	3.53	2.88
0.04	0.1	0.984	13.94	15.25	4.52	5.26	2.27	3.42	2.4
0.04	1	0.984	7.59	7.9	5.66	7.08	2.59	3.54	2.92
0.064	1	0.9744	7.71	7.74	4.89	6.88	2.41	3.42	2.66
0.064	1	0.9872	5.6	5.78	4.65	5.96	2.49	3.45	2.86
0.08	1	0.984	5.66	5.56	4.64	5.95	2.44	3.4	2.78

Table A.5 Effect of λ values for Hoth noise

Frame size	$\boldsymbol{\lambda}$	**SIR gain driver**	**SIR gain passenger**	**SNR gain driver**	**SNR gain passenger**	**DNSMOS overall**	**DNSMOS signal**	**DNSMOS noise**
0.008	0.96	14.53	16.11	7.23	7.56	2.66	3.51	3.11
0.008	0.984	10.82	13.03	6.34	6.85	2.68	3.56	3.1
0.024	0.96	12.33	14.22	7.50	8.12	2.44	3.41	2.8
0.024	0.984	9.20	10.01	6.54	6.93	2.33	3.32	2.62
0.04	0.984	7.58	7.9	6.09	6.47	1.71	2.52	1.75
0.064	0.984	6.23	6.36	5.49	5.93	1.48	2.11	1.47
0.08	0.984	5.97	4.76	5.29	5.52	1.34	1.84	1.34

A.2 Performance metrics MWF with different types and levels of noise

This section includes the data tables containing the performance metrics for the method mixing/adding the MWF outputs, with different types and levels of noise.

Table A.6 Performance metrics at different input SNRs for white noise

Input SNR (dB)	SIR gain driver (dB)	SIR gain passenger (dB)	SNR gain driver (dB)	SNR gain passenger (dB)	DNSMOS: signal, noise (MWF extracted signal)		DNSMOS: signal, noise (sum of mic signals)	
10	7.32	6.84	9.05	9.25	3.61	2.99	3.41	1.89
7	7.83	7.21	9.74	9.87	3.54	2.82	3.18	1.77
5	8.28	7.53	10.14	10.21	3.46	2.56	2.94	1.64
3	8.84	7.93	10.48	10.47	3.38	2.38	1.44	1.17
1	9.55	8.41	10.76	10.66	3.22	2.15	1.23	1.16
0	9.96	8.7	10.87	10.73	3.02	1.96	1.22	1.15

Table A.7 Performance metrics at different input SNRs for red noise

Input SNR (dB)	SIR gain driver (dB)	SIR gain passenger (dB)	SNR gain driver (dB)	SNR gain passenger (dB)	DNSMOS: signal, noise (MWF extracted signal)		DNSMOS: signal, noise (sum of mic signals)	
10	7.02	6.60	4.9	5.21	3.42	3.10	2.99	1.86
7	7.48	6.77	5.42	5.73	3.5	2.98	1.43	1.19
5	7.91	6.96	5.78	6.11	3.48	2.8	1.22	1.17
3	8.47	7.26	6.13	6.47	3.23	2.50	1.19	1.17
1	9.19	7.71	6.45	6.80	3.13	2.41	1.253	1.17
0	9.62	8.00	6.58	6.94	3.23	2.45	1.19	1.15

Table A.8 Performance metrics at different input SNRs for pink noise

Input SNR (dB)	SIR gain driver (dB)	SIR gain passenger (dB)	SNR gain driver (dB)	SNR gain passenger (dB)	DNSMOS: signal, noise (MWF extracted signal)		DNSMOS: signal, noise (sum of mic signals)	
10	7.27	7.17	5.62	5.46	3.48	2.78	3.11	1.81
7	7.95	7.84	6.05	5.86	3.42	2.75	2.06	1.39
5	8.58	8.45	6.36	6.14	3.34	2.58	1.25	1.17
3	9.34	9.17	6.68	6.4	2.81	1.99	1.19	1.16
1	10.15	9.88	6.97	6.62	1.68	1.27	1.19	1.14
0	10.52	10.18	7.11	6.7	1.38	1.17	1.18	1.12

Table A.9 Performance metrics at different input SNRs for green noise

Input SNR (dB)	SIR gain driver (dB)	SIR gain passenger (dB)	SNR gain driver (dB)	SNR gain passenger (dB)	DNSMOS: signal, noise (MWF extracted signal)		DNSMOS: signal, noise (sum of mic signals)	
10	7.95	7.76	8.06	7.85	3.58	3.19	3.37	2.2
7	8.68	8.35	8.59	8.33	3.42	2.83	2.63	1.67
5	9.31	8.82	8.92	8.64	3.33	2.62	1.89	1.36
3	10.07	9.38	9.23	8.92	3.15	2.3	1.59	1.26
1	10.99	10.06	9.51	9.18	2.75	1.93	1.22	1.12
0	11.48	10.46	9.63	9.29	2.48	1.73	1.20	1.12

Table A.10 Performance metrics at different input SNRs for hoth noise

Input SNR (dB)	SIR gain driver (dB)	SIR gain passenger (dB)	SNR gain driver (dB)	SNR gain passenger (dB)	DNSMOS: signal, noise (MWF extracted signal)		DNSMOS: signal, noise (sum of mic signals)	
10	5.83	6.27	6.20	5.88	3.48	3.05	2.71	1.69
7	6.60	7.33	6.67	6.29	2.93	2.3	1.28	1.18
5	7.36	8.33	7.04	6.61	2.82	2.18	1.19	1.16
3	8.38	9.511	7.45	6.96	2.6	1.97	1.18	1.15
1	9.7	10.58	7.87	7.31	2.42	1.79	1.18	1.13
0	10.42	10.86	8.08	7.47	2.12	1.58	1.18	1.13

Table A.11 Performance metrics at different input SNRs for wind noise

Input SNR (dB)	SIR gain driver (dB)	SIR gain passenger (dB)	SNR gain driver (dB)	SNR gain passenger (dB)	DNSMOS: signal, noise (MWF extracted signal)		DNSMOS: signal, noise (sum of mic signals)	
10	6.53	6.48	2.06	5.19	3.39	3.66	3.53	3.48
7	6.62	6.44	1.96	5.2	3.30	3.52	3.54	2.90
5	6.68	6.39	1.87	5.13	3.25	3.24	3.53	2.75
3	6.74	6.31	1.77	4.98	3.25	3.040	3.53	2.62
1	6.79	6.20	1.64	4.76	3.27	2.91	3.4	2.43
0	6.81	6.15	1.57	4.61	3.27	2.89	3.32	2.30

A.3 SIR gain and SNR gain values during head movement

This section gives the data tables for SIR gain and SNR gain values during head movement, for both continuous MWF adaptation as well as paused MWF adaptation.

Table A.12 SIR gain, SNR gain values for white noise with head movement and continuous adaptation

		20-28 sec		**28-36 sec**		**36-44 sec**		**44-52 sec**		**52-60 sec**		**60-68 sec**	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	6.92	9.13	6.8	9.13	6.90	9.21	6.81	9.19	7.82	9.2	6.25	9.23
0.45	10	6.92	9.13	6.8	9.13	6.94	9.22	7.02	9.18	8.37	9.19	6.25	9.23
0.5	5	7.68	10.14	7.44	10.25	7.47	10.35	7.36	10.38	8.61	10.24	6.43	10.35
0.45	5	7.68	10.14	7.44	10.25	7.50	10.33	7.54	10.36	9.17	10.24	6.43	10.35
0.5	1	8.82	10.70	8.2	10.89	8.25	10.9752	8.05	11.05	9.74	10.81	6.74	10.99
0.45	1	8.82	10.70	8.2	10.89	8.28	10.96	8.22	11.04	10.26	10.81	6.74	10.99

Table A.13 SIR gain, SNR gain values for white noise with head movement and paused adaptation

		20-28 sec		**28-36 sec**		**36-44 sec**		**44-52 sec**		**52-60 sec**		**60-68 sec**	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	6.92	9.13	6.8	9.13	2.23	6.65	2.65	7.17	2.33	6.73	3.07	7.2
0.45	10	6.92	9.13	6.8	9.13	2.10	6.52	2.41	7.17	2.33	6.73	3.07	7.2
0.5	5	7.68	10.14	7.44	10.25	2.26	6.96	3.01	8.1	2.33	6.7	3.23	8.11
0.45	5	7.68	10.14	7.44	10.25	2.14	6.84	2.92	8.1	2.33	7	3.23	8.11
0.5	1	8.82	10.70	8.2	10.89	2.02	6.24	3.15	8.00	2.08	6.27	3.13	7.99
0.45	1	8.82	10.70	8.2	10.89	1.89	6.12	3.22	8.00	2.08	6.27	3.13	7.99

Table A.14 SIR gain, SNR gain values for red noise with head movement and continuous adaptation

		20-28 sec		28-36 sec		36-44 sec		44-52 sec		52-60 sec		60-68 sec	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	6.53	4	6.84	5.38	6.72	4.05	6.37	6.47	7.8	4.46	6.51	6.63
0.45	10	6.53	4	6.84	5.38	6.75	4.00	6.6	6.45	8.25	4.46	6.51	6.63
0.5	5	6.99	4.33	7.47	5.52	7.01	4.53	6.76	7.46	8.49	4.93	6.8	7.71
0.45	5	6.99	4.33	7.47	5.52	7.02	4.48	6.92	7.45	8.89	4.93	6.8	7.71
0.5	1	7.69	4.45	8.19	5.1	7.30	4.8	7.48	7.97	9.41	5.18	7.2	8.33
0.45	1	7.69	4.45	8.19	5.1	7.3	4.75	7.55	7.96	9.73	5.18	7.2	8.33

Table A.15 SIR gain, SNR gain values for red noise with head movement and paused adaptation

		20-28 sec		28-36 sec		36-44 sec		44-52 sec		52-60 sec		60-68 sec	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	6.53	4	6.84	5.38	1.01	1.61	0.8	3.15	1.34	1.81	1.46	3.13
0.45	10	6.53	4	6.84	5.38	0.73	1.32	0.38	3.15	1.34	1.81	1.46	3.13
0.5	5	6.99	4.33	7.47	5.52	0.12	0.70	-0.01	0.90	0.53	0.94	0.69	0.98
0.45	5	6.99	4.33	7.47	5.52	-0.28	0.29	-0.43	0.90	0.53	0.94	0.69	0.98
0.5	1	7.69	4.45	8.19	5.1	-0.74	-0.77	-0.57	-1.67	-0.29	-0.52	0.13	-1.55
0.45	1	7.69	4.45	8.19	5.1	-1.25	-1.28	-0.98	-1.67	-0.29	-0.52	0.13	-1.55

Table A.16 SIR gain, SNR gain values for pink noise with head movement and continuous adaptation

		20-28 sec		**28-36 sec**		**36-44 sec**		**44-52 sec**		**52-60 sec**		**60-68 sec**	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	7.78	5.67	6.77	5.60	7.62	5.54	6.96	5.58	8.10	5.56	7.65	5.53
0.45	10	7.78	5.67	6.77	5.60	7.65	5.52	7.11	5.56	8.63	5.56	7.65	5.53
0.5	5	9.32	6.45	7.60	6.3	8.79	6.35	8.1	6.34	9.81	6.31	9.52	6.21
0.45	5	9.32	6.45	7.60	6.3	8.81	6.33	8.194	6.33	10.36	6.31	9.52	6.21
0.5	1	10.64	7.06	8.35	6.86	9.72	7.01	9.58	6.95	12.40	6.93	12.21	6.75
0.45	1	10.64	7.06	8.35	6.86	9.71	6.99	9.38	6.94	12.95	6.93	12.21	6.75

Table A.17 SIR gain, SNR gain values for pink noise with head movement and paused adaptation

		20-28 sec		**28-36 sec**		**36-44 sec**		**44-52 sec**		**52-60 sec**		**60-68 sec**	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	7.78	5.67	6.77	5.60	0.65	2.41	1.72	3.38	1.03	2.75	2.32	3.4
0.45	10	7.78	5.67	6.77	5.60	0.33	2.1	1.37	3.38	1.03	2.75	2.32	3.4
0.5	5	9.32	6.45	7.60	6.3	-0.95	1.83	1.12	3.42	-0.25	2.43	1.8	3.42
0.45	5	9.32	6.45	7.60	6.3	-1.5	1.29	0.72	3.42	-0.25	2.43	1.8	3.42
0.5	1	10.64	7.06	8.35	6.86	-2.6	0.41	0.22	2.24	-1.56	1.28	0.88	2.21
0.45	1	10.64	7.06	8.35	6.86	-3.38	-0.37	-0.27	2.24	-1.56	1.28	0.88	2.21

Table A.18 SIR gain, SNR gain values for green noise with head movement and continuous adaptation

		20-28 sec		28-36 sec		36-44 sec		44-52 sec		52-60 sec		60-68 sec	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	7.54	7.88	7.89	8.00	7.41	7.74	7.22	7.83	8.66	7.91	7.52	7.98
0.45	10	7.54	7.88	7.89	8.00	7.43	7.73	7.43	7.81	9.17	7.91	7.52	7.98
0.5	5	8.60	8.67	8.70	8.81	7.92	8.58	8.16	8.58	9.89	8.74	8.35	8.75
0.45	5	8.60	8.67	8.70	8.81	7.94	8.57	8.40	8.56	10.387	8.74	8.35	8.75
0.5	1	9.86	9.21	9.43	9.33	8.47	9.18	9.33	9.02	10.76	9.32	8.9	9.22
0.45	1	9.86	9.21	9.43	9.33	8.48	9.19	9.71	9.00	11.17	9.32	8.9	9.22

Table A.19 SIR gain, SNR gain values for green noise with head movement and paused adaptation

		20-28 sec		28-36 sec		36-44 sec		44-52 sec		52-60 sec		60-68 sec	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	7.54	7.88	7.89	8.00	2.39	5.65	2.32	5.81	2.66	5.96	3.2	5.83
0.45	10	7.54	7.88	7.89	8.00	2.26	5.52	2.14	5.81	2.66	5.96	3.2	5.83
0.5	5	8.60	8.67	8.70	8.81	1.92	5.44	2.33	5.27	2.21	5.77	3.11	5.33
0.45	5	8.60	8.67	8.70	8.81	1.79	5.31	2.26	5.27	2.21	5.77	3.11	5.33
0.5	1	9.86	9.21	9.43	9.33	0.85	4.03	2.37	3.95	1.09	4.31	2.76	4.04
0.45	1	9.86	9.21	9.43	9.33	0.72	3.9	2.53	3.95	1.09	4.31	2.76	4.04

Table A.20 SIR gain, SNR gain values for Hoth noise with head movement and continuous adaptation

		20-28 sec		28-36 sec		36-44 sec		44-52 sec		52-60 sec		60-68 sec	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	4.9	5.99	5.70	6.16	4.93	5.92	5.33	5.95	6.28	6.1	4.85	6.18
0.45	10	4.9	5.98	5.70	6.16	4.8	5.85	5.51	5.92	6.76	6.1	4.85	6.18
0.5	5	5.62	6.62	7.1	6.96	5.42	6.67	7.09	6.73	7.40	6.82	5.29	7.02
0.45	5	5.62	6.62	7.1	6.96	5.31	6.60	7.34	6.71	7.83	6.82	5.29	7.02
0.5	1	6.69	7.24	8.8	7.69	5.86	7.37	9.58	7.47	8.76	7.51	5.74	7.8
0.45	1	6.69	7.24	8.8	7.69	5.73	7.31	9.85	7.45	9.09	7.51	5.74	7.8

Table A.21 SIR gain, SNR gain values for Hoth noise with head movement and paused adaptation

		20-28 sec		28-36 sec		36-44 sec		44-52 sec		52-60 sec		60-68 sec	
d	Input SNR	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger	SIR gain driver	SNR gain driver	SIR gain passenger	SNR gain passenger
0.5	10	4.9	5.99	5.70	6.16	-1.21	2.79	-0.1	3.19	-0.55	3.54	0.73	3.27
0.45	10	4.9	5.98	5.70	6.16	-1.84	2.17	-0.51	3.19	-0.55	3.54	0.73	3.27
0.5	5	5.62	6.62	7.1	6.96	-1.90	3.35	0.08	3.33	-1.32	4.02	0.41	3.43
0.45	5	5.62	6.62	7.1	6.96	-2.57	2.68	-0.04	3.33	-1.32	4.02	0.41	3.43
0.5	1	6.69	7.24	8.8	7.69	-2.58	3.60	0.15	2.64	-2.21	4.07	-0.11	2.75
0.45	1	6.69	7.24	8.8	7.69	-3.20	2.99	0.27	2.64	-2.21	4.07	-0.11	2.7