KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Joint Multi-Channel Noise Reduction and Echo Cancellation

Real-Time Implementation and Evaluation

Sven Verret

Thesis submitted for the degree of
Master of Science in
Electrical Engineering, im

Thesis supervisor:
Prof. dr. ir. Marc Moonen

Assessors:
Prof. dr. ir. Patrick Wambacq
Prof. dr. ir. Alexander Bertrand

Mentor:
ir. Robbe Van Rompaey

Academic year 2018 – 2019

© Copyright KU Leuven

Without written permission of the thesis supervisor and the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to Departement Elektrotechniek, Kasteelpark Arenberg 10 postbus 2440, B-3001 Heverlee, +32-16-321130 or by email `info@esat.kuleuven.be`.

A written permission of the thesis supervisor is also required to use the methods, products, schematics and programmes described in this work for industrial or commercial use, and for submitting this publication in scientific contests.

Abstract

In the domain of signal processing, whenever we record a speech signal in a crowded room, we are often confronted with microphone signals that are both contaminated by excessive background noise and an interfering audio signal, also called the echo signal. We want to increase the quality of our recordings by reducing the power of both these unwanted components.

There are multiple possible solutions for this issue, but the most straight-forward is a two-stage process, consisting of two cascading filters. The first filter reduces the ambient background noise and the second filter cancels out the interfering echo signal.The noise reduction step is implemented through different variants of the Multi-channel Wiener Filter (MWF). A Mean Squares (MS) echo filter takes care of the echo signal.

It is also possible to achieve both noise reduction and echo cancellation in one stage of processing. These are the so-called joint noise reduction and echo cancellation systems. In this thesis, we evaluate these experimental integrated solutions. We compare the performances and computational requirements of these algorithms against classical two-stage solutions that serve as benchmarks. To test the hypothesis that an integrated method can be as efficient and effective as multiple dedicated components, we conduct both virtual and real-life experiments. We evaluate and compare their performances with relevant performance measures.

The two-step option we have chosen as a reference (MS filters on all channels followed by an MWF) remains a good choice, due to its relatively fast execution and stable performance. Remarkable is the very effective operation in case of a loud echo signal.The computational more efficient solution (MWF followed by one MS filter) does not meet our expectations. The gain in execution time is too limited to compensate for the loss of performance compared to the reference method. Equipping the MWF with the clean interference signal in order to improve echo suppression is also disappointing. This method achieves slightly worse results than the reference, but now without the gain in calculation time that the previous method boasts.

One integrated method offers a good possible replacement for our reference. It becomes clear that the current form of the subspace projection method is too inefficient, with no equivalent increase in performance. On the other hand, the integrated echo-subtraction method is efficient enough and gives good results, comparable to the reference method. Further research is needed, with more performance measures to be evaluated.

Contents

Abstract i

Contents iii

List of Figures v

Samenvatting vi

Nomenclature xi

1 Introduction 1
- 1.1 Overview of Thesis . . . 2

2 Overview of System and Signal Model 5
- 2.1 Research Scenario . . . 5
- 2.2 Acoustic Signal Model . . . 7
- 2.3 Combined Solutions . . . 10

3 Audio Signal Processing Techniques 11
- 3.1 Introduction . . . 11
- 3.2 Noise Reduction . . . 11
  - 3.2.1 Measurement of NR Performance . . . 11
  - 3.2.2 Multi-channel Wiener Filtering (MWF) . . . 12
  - 3.2.3 Generalized Eigenvalue Decomposition (GEVD)-based MWF 14
- 3.3 Echo Cancellation . . . 15
  - 3.3.1 Measurement of EC Performance . . . 15
  - 3.3.2 Mean Squares Filtering (MS) . . . 16
- 3.4 Combined Noise Reduction and Echo Cancellation Techniques . . . 17
  - 3.4.1 Two-step MS and MWF Filtering . . . 17
  - 3.4.2 Far End Dimension Extension . . . 17
  - 3.4.3 Far End Signal Subtraction . . . 18
  - 3.4.4 Subspace Projection MWF . . . 19
  - 3.4.5 Time-Series Segmentation MWF . . . 23

4 Experimental Setups and Expectations 25
- 4.1 Introduction . . . 25
- 4.2 Discrete Setups . . . 26
  - 4.2.1 Two-step MS + MWF . . . 26
  - 4.2.2 Two-step MWF + MS . . . 27
  - 4.2.3 Two-step MWF with Extension + MS . . . 27

4.3 Integrated Setups . . . 28
4.3.1 MWF with FE Subtraction . . . 28
4.3.2 MWF with Subspace Projection . . . 29

5 Evaluation by Offline Simulation 31
5.1 Introduction . . . 31
5.2 General Code and Settings . . . 31
5.3 Additional Code . . . 32
5.4 Implementation of Performance Measures . . . 32
5.5 Results and Discussion . . . 33
5.5.1 Introduction . . . 33
5.5.2 Noise Reduction Performance . . . 34
5.5.3 Echo Cancellation Performance . . . 36
5.5.4 Intelligibility Performance . . . 37
5.6 Conclusion . . . 38

6 Evaluation by Real-Time Implementation 41
6.1 Introduction . . . 41
6.2 Phase 1: Offline Experiments with Real Data . . . 42
6.3 Phase 2: Real-Time Experiments . . . 42
6.4 Implementation of Performance Measurements . . . 44
6.4.1 Creation of Suitable Input Scenarios . . . 45
6.4.2 Calculating Output Measures . . . 47
6.5 Results and Discussion . . . 47
6.5.1 Introduction . . . 47
6.5.2 Noise Reduction Performance . . . 48
6.5.3 Echo Cancellation Performance . . . 48
6.5.4 Intelligibility Performance . . . 50
6.5.5 Computational Requirements . . . 52
6.6 Conclusion . . . 54

7 Conclusion and Further Research 57
7.1 Conclusion . . . 57
7.2 Further Research . . . 58
7.2.1 Better Implementation of Subspace Projection MWF Algorithm 58
7.2.2 Speech Intelligibility Weighted Measures . . . 59
7.2.3 Multiple Desired or Interfering Sources . . . 59
7.2.4 Speech Presence Probability VAD . . . 59

Bibliography 61

IEEE paper 65

List of Figures

2.1 Illustration of Research Scenario. 7

3.1 Possible Input Segmentations of Recorded Signals. 23

4.1 Block Diagram of Setup: MS filter followed by GEVD-MWF. 26
4.2 Block Diagram of Setup: GEVD-MWF followed by MS filter. 28
4.3 Block Diagram of Setup: Extended GEVD-MWF followed by MS filter. 29
4.4 Block Diagram of Setup: Integrated MWF Setup. 29

5.1 Noise Reduction Performances of Matlab Simulations. 34
5.2 Echo Cancellation Performances of Matlab Simulations. 37
5.3 Intelligibility Results: SINR values of Matlab Simulations. 38

6.1 Picture of the Real-Time Setup: loudspeakers and microphone array. . . 43
6.2 Picture of the Real-Time Setup: close-up of the microphone array. . . . 43
6.3 Picture of the Real-Time Setup: PC as playback device and Speedgoat. 44
6.4 Noise Reduction Performances of Real-Time Implementations. 49
6.5 Echo Cancellation Performances of Real-Time Implementations. 50
6.6 Intelligibility Results: PESQ scores of Real-Time Implementations. . . . 51
6.7 Intelligibility Results: SINR values of Real-Time Implementations. . . . 51
6.8 Execution Times of Real-Time Implementations. 53

Samenvatting

Introductie

In het domein van de signaalverwerking worden we vaak geconfronteerd met vervuilde microfoonsignalen wanneer we een spraaksignaal willen opnemen in een drukke ruimte. Vaak zijn deze signalen vervuild door achtergrondruis en door storende geluidssignalen van ongewenste spraaksignalen of luidsprekersignalen. Indien men ook toegang heeft tot deze onvervuilde, ongewenste spraaksignalen, noemen we deze echo signalen. We willen de kwaliteit van onze opnames verhogen door het volume van zowel de ruis als de echo te verminderen, zonder het gewenste spraaksignaal overmatig te vervormen of reduceren. Er zijn heel wat domeinen die dit soort audioverwerking vereisen.

Een belangrijke toepassing die we nader bestuderen is een betrouwbare en hoogwaardige audioconferentie-installatie. Het doel van de audioverwerking is te vergelijken met bijvoorbeeld een hoorapparaat, namelijk om zich te kunnen focussen op specifieke spraakbronnen door de ruis in alle andere richtingen te dempen. Het wegwerken van dit omgevingslawaai wordt ruisonderdrukking genoemd.

Conferentiesystemen zijn eveneens uitgerust met luidsprekers om de spraak van de gesprekspartner uit te zenden, zodat tweewegscommunicatie mogelijk is. Een dergelijke opstelling vormt echter een aanzienlijke bijkomende uitdaging. Indien de benodigde audioverwerking niet aanwezig is, wordt het geluid dat door de luidsprekers klinkt onmiddellijk opgenomen door de microfoons en teruggestuurd. Hetzelfde zal gebeuren bij het andere apparaat, waardoor een terugkoppeling lus ontstaat die het systeem onbruikbaar maakt. Dit effect is niet te verwaarlozen, want bij de meeste conferentiesystemen bevinden de luidsprekers zich in de buurt van de microfoons. Hierdoor zal het stoorsignaal een aanzienlijk vermogen hebben in het opgenomen signaal. Om dit probleem op te lossen, moet het stoorsignaal uit de geluidsopnames verwijderd worden. Dit type audioverwerking wordt echo-onderdrukking genoemd.

Hetzelfde probleem doet zich voor wanneer we onze smartphones beschouwen, die ook uitgerust zijn met microfoons en luidsprekers voor telefoongesprekken. Uw smartphone kan met één tik worden omgezet in een conferentiesysteem indien we de handsfree functie gebruiken, waarbij er meteen ook een nood ontstaat voor een ruis- en echo-onderdrukkingsalgoritme.

In deze thesis bespreken en testen we methodes die zowel ruis- als echo-onderdrukking kunnen bereiken in één geïntegreerde fase. We vergelijken de prestaties en computationele vereisten van deze algoritmes met klassieke tweestapsoplossingen, die als referentie dienen. Vervolgens zullen we op basis van zowel Matlab als real-time experimenten concluderen of het mogelijk is om deze geïntegreerde alternatieven te gebruiken zonder prestatieverlies.

Gezamenlijke ruis- en echo-onderdrukking

Er zijn meerdere oplossingen mogelijk voor dit probleem, maar de eenvoudigste optie is een tweestaps-algoritme bestaande uit twee opeenvolgende filters. De eerste filter verzwakt de omgevingsruis en de tweede filter neutraliseert het storende echosignaal.

De ruisreductiestap wordt uitgevoerd door middel van verschillende varianten van de Meerkanaals Wiener Filter (MWF). Een normale MWF verzamelt informatie van zowel spraak- als ruissegmenten in de microfoonsignalen en maakt afzonderlijke correlatiematrices. Het onderscheid is gebaseerd op een spraakactiviteitsdetector (Voice Activity Detector - VAD), die controleert of er in het huidige segment spraak actief is. Deze twee correlatiematrices worden vervolgens gecombineerd om de vereiste spraakcorrelatiematrix te vormen met behulp van een gegeneraliseerde eigenwaardeontbinding (Generalized Eigenvalue Decomposition - GEVD), die een robuuste manier biedt om de ruis te verminderen. Op basis van de spraak- en ruiscorrelatiematrices worden de MWF-coëfficiënten berekend.

Een echofilter verwijdert het echo-onderdeel uit een opgenomen spraaksignaal. Zulke filter benadert eerst het akoestische pad van de echobron naar elke microfoon en trekt in elk kanaal een gefilterde versie van het onvervuilde echosignaal af, met als doel de energie van het interferentiesignaal in de uitgang te minimaliseren.

Elke combinatie van een aparte ruisfilter en een echofilter noemen we een tweestapmethode. Naast deze bekende aanpak is het ook mogelijk om zowel ruis- als echo-onderdrukking in één bewerkingsstap te bereiken. Dit zijn de zogenaamde gezamenlijke ruis- en echo-onderdrukkingssystemen. In deze thesis evalueren we deze experimentele geïntegreerde oplossingen.

We bespreken en testen twee mogelijke opties: de subruimteprojectiemethode en de echo-aftrekkingsmethode. Deze methodes vereisen allebei VAD-informatie van zowel de gewenste als de interfererende spraak (echobron) om de verschillende correlatiematrices correct te schatten.

De projectiemethode beschouwt een aparte matrix voor elk van de vier mogelijke VAD gevallen, zodat er geen gegevens ongebruikt blijven. Het maakt gebruik van de goed gedefinieerde gezamenlijke subruimte als een manier om de kwaliteit van de spraak- en echo-subruimtes te verbeteren, met behulp van een projectiemethode.

Al deze informatie wordt vervolgens gezamenlijk gebruikt om een zo goed mogelijke schatting te geven van de gewenste subruimte die ten slotte gebruikt wordt om de MWF-coëfficiënten te berekenen.

De echo-aftrekkingsmethode gebruikt een soort ingebouwde echo-filter, die de echo-informatie tijdens het MWF proces kan extraheren en meteen aftrekken. Er is dus geen nood meer aan een aparte stap die de transferfunctie tussen echobron en microfoon probeert te benaderen en dit onderdeel probeert weg te werken uit de mix.

Voorgestelde opstellingen

Deze sectie geeft een overzicht van welke oplossingen we onderzoeken tijdens zowel de Matlab simulaties als de real-time experimenten. Elk algoritme dat we in deze thesis bespreken, werkt in het discreet frequentiedomein. Hierdoor bevat elke opstelling een onderdeel om de signalen om te zetten van het tijdsdomein naar het frequentiedomein en visa versa. Door het real-time karakter van de probleemopstelling hebben we nood aan een Fourier transformatie met schuivend venster, zodat deze continue kan werken, in deze thesis geïmplementeerd door de gevensterde Fourier transformatie (Short-Time Fourier Transformation - STFT).

De eerste oplossing (MS+MWF) gaat eerst het echosignaal in elk kanaal dempen, door de overdrachtsfuncties tussen de echobron en elke microfoon te benaderen waarna deze benadering van het signaal wordt afgetrokken. Dit wordt in onze implementatie door een Mean Squares (MS) filter gedaan. Na de aftrekking worden de kanalen naar een MWF gestuurd om de achtergrondruis te onderdrukken. Dit is het eerste twee-staps algoritme dat we bespreken en testen.

De tweede setup (MWF+MS) gaat na of het omwisselen van de volgorde van beide componenten een grote invloed heeft op de prestaties van het algoritme als geheel. We zijn er zeker van dat deze opstelling gunstiger is qua complexiteit dan de vorige, aangezien deze aanpak slechts één echofilter vereist in plaats van één voor elk microfoonkanaal. We bestuderen ook het onderscheid tussen een rang 1 of een rang 2 MWF benadering. Bij gebruik van de rang 2 versie moet de MWF zowel de gewenste als storende spraak intact laten. Dit geeft de MS-filter een gelijkaardige opdracht als de referentieopstelling, aangezien de interferentie nog niet gefilterd is en dus een vergelijkbaar volume heeft. Daarom verwachten we dat de echo-onderdrukking ongeveer even effectief zal zijn als de MS+MWF setup. We zijn niet zeker of de rang 2 benadering een grote impact heeft op de ruisonderdrukking. Bij gebruik van de rang 1 MWF wordt de echo echter ook als ruis beschouwd en wordt deze al onderdrukt door de MWF zelf. Het lagere volume zou het voor de MS-filter moeilijker kunnen maken om de overdrachtsfuncties te bepalen. Er bestaat bovendien ook een mogelijkheid dat de rang 1 benadering een verkeerde subruimte selecteert.

De laatste twee-staps opstelling die we bespreken is de MWF+MS met extensie (MWF ext+MS). We hebben toegang tot het onvervuilde interferentiesignaal in elke

fase van de audioverwerking, dus we kunnen deze informatie wellicht gebruiken om betere MWF-coëfficiënten te ontwikkelen, die ook de echo actief onderdrukken. We geven de filter toegang tot het bronsignaal van de echo door deze toe te voegen aan de ingangsset van de MWF. Opnieuw maken we het onderscheid tussen de rang 1 en rang 2 benadering.

Ten slotte bespreken we twee geïntegreerde oplossingen die slechts één verwerkingsfase vereisen: de subruimteprojectiemethode en de echo-aftrekkingsmethode. Beide opstellingen zijn vrij eenvoudig, aangezien beide enkel bestaan uit de desbetreffende geïntegreerde algoritmes samen met de Fourier transformaties.

Resultaten uit experimenten

De experimenten bestaan uit drie fases. Eerst voeren we Matlab-simulaties uit in een gesimuleerde ruimte met zelf gecreeërde signalen. Vervolgens gebruiken we een opstelling in het audio laboratorium om de bronsignalen in een echte ruimte af te spelen. Met behulp van onze microfonen nemen we de ruimte op. Dan gebruiken we deze opnames als ingangssignalen voor de verschillende methodes, die we als Simulink modellen geïmplementeerd hebben. Ten slotte passen we de Simulink modellen aan zodat deze volledig in real-time werken en we deze kunnen gebruiken in het audio laboratorium. Op basis van de resultaten uit de experimenten van deze drie fases kunnen we volgende besluiten trekken.

De eerste optie (MS+MWF) die we testen is een initiële echo filter installeren op alle microfoonkanalen om het interfererende signaal te dempen. Vervolgens gebruiken we deze signalen om de MWF van rang 1 aan te sturen, zodat deze de resterende ruissignalen kan onderdrukken. Zowel op vlak van ruis- als echo-onderdrukking presteert deze methode goed, met een uitzonderlijk resultaat indien het volume van de echo zeer luid is. Dit zorgt ervoor dat de echofilter toegang heeft tot een duidelijk signaal, met als gevolg een goede benadering van de transferfunctie tussen bron en microfoon wat voor een zeer effectieve werking zorgt. Aangezien dit een gevestigde techniek is die over het algemeen goede prestatiecijfers oplevert, gebruiken we deze als referentie.

We testen ook de efficiëntere versie (MWF+MS) van de tweestaps-oplossingen, namelijk de opstelling die eerst de MWF toepast om het ruisvolume te verminderen en vervolgens de resterende echo uit het enkelvoudig uitgangskanaal filtert. Deze oplossing heeft volgens de experimenten zoals verwacht minder berekeningen nodig, maar ze toont echter ook een aanzienlijke negatieve impact op zowel de ruis- als echo-onderdrukking, een impact die groter is dan de relatief beperkte efficiëntiewinst.

Vervolgens proberen we een betere echo-onderdrukking te bereiken door ook de MWF te voorzien van het schone stoorsignaal (MWF ext+MS). We bereiken dit door het stoorsignaal bij de rest van de microfoonsignalen te voegen als input voor de MWF. De experimenten tonen aan dat deze toevoeging de prestaties wel verhoogt,

maar er nog steeds niet voor kan zorgen dat deze methode op hetzelfde niveau als de referentiemethode werkt. De toevoeging van het echosignaal doet ook het rekenvoordeel ten opzichte van de referentie teniet, doordat we de gebruikte matrices en vectoren met één dimensie moeten verhogen. We concluderen dat deze specifieke implementatie niet bruikbaar is voor onze probleemstelling, omdat zowel de prestaties als de efficiëntie ondermaats zijn aan de referentie.

Laten we tot slot de resultaten van het eigenlijke onderwerp van deze thesis bespreken: de geïntegreerde oplossingen. We stellen twee experimentele methoden voor: de subruimteprojectiemethode en de echo-aftrekkingsmethode. De realtime experimenten tonen aan dat de echo-aftrekkingsmethode voor alle onderzochte prestatie-indicatoren beter presteert dan de subruimteprojectiemethode, terwijl deze ook veel minder berekeningen vereist. Na het bekijken van deze resultaten lijkt er weinig stimulans te zijn om in de praktijk de subruimteprojectiemethode te gebruiken. De echo-aftrekkingmethode presteert daarentegen in sommige gevallen zelfs beter dan de referentiemethode. De methode bereikt een betere ruis-onderdrukking, maar iets slechtere resultaten voor echo-onderdrukking in vergelijking met de referentie. Deze techniek vraagt ook meer berekeningen dan de referentie, maar het verschil is niet erg drastisch.

Besluit

De experimenten suggereren dat er meerdere effectieve manieren zijn om gezamenlijke ruis- en echo-onderdrukking te bereiken. De tweestaps-optie die we als referentie hebben gekozen (MS+MWF) blijft een goede keuze, door de relatief snelle uitvoering en stabiele prestaties. Opmerkelijk is de zeer effectieve werking in geval van een luid echosignaal.

De computationele meer efficiënte oplossing (MWF+MS) voldoet niet aan onze verwachtingen. De winst in uitvoeringstijd is te beperkt om het prestatieverlies ten opzichte van de referentiemethode te compenseren. Het idee om de MWF te voorzien van het schone stoorsignaal (MWF ext+MS) met als doel de echo-onderdrukking te verbeteren heeft ook teleurgesteld. Deze methode behaalt nog steeds iets slechtere resultaten dan de referentie, maar nu zonder de voorgaande winst in nodige rekentijd.

Eén van onze geïntegreerde methodes biedt een goede mogelijke vervanging voor onze referentie. Na het evalueren van onze resultaten is het duidelijk dat de huidige vorm van de subruimteprojectiemethode te inefficiënt is, zonder dat de prestaties een evenwaardige verhoging kennen. De echo-aftrekkingsmethode daarentegen is wel efficiënt genoeg en geeft mooie resultaten, vergelijkbaar met de referentiemethode. Verder onderzoek moet deze methode grondiger vergelijken met de referentie, waarbij nog meer prestatie-indicatoren geëvalueerd worden. Ook andere scenario's kunnen onderzocht worden.

Nomenclature

Acronyms

NR	noise reduction
EC	echo cancellation
MWF	multi-channel Wiener filter
LMMSE	linear minimum mean square error
VAD	voice activity detector
GEVD	generalized eigenvalue decomposition
MS	mean squares (filter)
NE	near end
FE	far end
SNR	signal-to-noise ratio
SIR	signal-to-interference ratio
SINR	signal-to-interference-plus-noise ratio
PESQ	perceptual evaluation of speech quality
RIR	room impulse response
STFT	short-time Fourier transformation
ISTFT	inverse short-time Fourier transformation
LCMV	linearly constrained minimum variance

Abbreviations

dB	decibel
Hz	hertz
i.e.	*id est*, that is
e.g.	*exempli gratia*, for example

Symbols

$Y[l]$	discrete time domain signal
$\boldsymbol{Y}[l]$	discrete array of time domain signals with time index l
$y[m, w]$	short-time frequency domain signal
$\boldsymbol{y}[m, w]$	array of short-time frequency domain signals with frequency bin index m and window shift w
A	matrix
$\tilde{A}$	approximation of matrix A
$\mathrm{diag}\{\cdot\}$	diagonal matrix with arguments as diagonal elements
$\mathrm{GEVD}_{r1}(A, B)$	rank 1 generalized eigenvalue decomposition of matrix A and B
$\mathrm{GEVD}_{r2}(A, B)$	rank 2 generalized eigenvalue decomposition of matrix A and B
$\{\cdot\}^*$	complex conjugate
$\{\cdot\}^H$	complex conjugate transpose
$\{\cdot\}^T$	non-conjugate transpose
$\{\cdot\}^{-1}$	matrix inverse
$E\{\cdot\}$	mathematical expectation
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert^n$	n-norm
$\lVert\cdot\rVert_F$	Frobenius norm
$\mathrm{vec}(\cdot)$	matrix vectorization
$\otimes$	Kronecker product
$\min_a$	minimize over a
$\arg\min_a$	argument of the minimum

Chapter 1

Introduction

Why do we need efficient signal processing systems for speech enhancement? Nowadays, more and more devices are equipped with multiple high-quality microphones. This available hardware increases the complexity of these issues, but also potentially increases the quality of the resulting speech. There are multiple scenarios where extra processing benefits the quality and usability of a multimedia system.

Medical applications are for example an important field of study [1] [2], especially the development of hearing aids is an important point for speech enhancement [3]. People who suffer from hearing loss require a greater separation in volume between the actual speech of the person that they are conversing with and any other interfering noise in their environment. These noises can be generated by different kinds of sources, e.g. static monotonous noise created by a ceiling fan, more dynamic noise from a television that is playing in the background or dispersed babble noise from other people in the room that are talking.

For this application, it is also of vital importance that the system works fast enough to allow for real-time processing, preferably with a minimal fixed delay. Having a big delay on your sense of hearing is very disorientating and uncomfortable, so this needs to be avoided at all cost. Also the audio quality of the speech has to be high enough for the user to understand the conversation. There exists a trade off in the type of noise reduction we are going to discuss, namely the better the noise suppression, the more the recorded speech will be distorted. If this distortion is too dominant, the user will begin to have trouble understanding what is being said. Minimizing this distortion while keeping the noise suppression performance constant is definitely another point of research. In this thesis, we will also look at real-time implementations of the different algorithms. It is important to check the computational requirements of the different options. In practice when developing a hearing aid, using an algorithm with a high computational intensity will require faster hardware and higher capacity battery cells, resulting in a heavier and larger hearing aid that is less convenient to wear.

Another important application is a reliable and high-quality audio conference system [4]. The goal of the audio processing here is similar to a hearing aid system, namely being able to focus on target speech sources by attenuating noise in all other directions. Getting rid of this noise in the recorded environment is called noise reduction.

However, audio conference systems are also equipped with speakers to broadcast the speech of the other party into the room, so that two-way communication is made possible. This kind of setup presents a considerable extra challenge though. Without any audio processing, the sound being played by the speakers immediately gets recorded by the microphones and sent back. The same happens by the other device, creating a feedback loop that makes the system unusable. This effect is far from ignorable, because with most conference systems, either a dedicated system or e.g. a laptop, the speakers are located near the microphones. Because of this, the interfering signal will have considerably power in the recorded signal. To solve this issue, the interfering signal has to be removed from the recordings. This type of audio processing is called echo cancellation.

The same problem arises when looking at our smartphones, which are also equipped with microphones and speakers for phone calls, be it hands free or not. Your smartphone can be converted into a conference system by one tap, immediately requiring an algorithm for noise reduction and echo cancellation.

Speech enhancement can be very useful as pre-processing for speech recognition systems [5] [6]. The higher the quality of the processed recording, the less likely is it for the speech recognition system to make a wrong decision. These two systems together can then be used as a voice interface with different kinds of devices, like personal computers and gaming consoles.

In this thesis, we discuss and test methods that are capable of achieving both noise reduction and echo cancellation in one integrated stage of signal processing. We compare the performances and computational requirements of these algorithms against classical two-stage solutions that serve as benchmarks. Then we draw a conclusion based on both Matlab and real-time experiments whether it is compelling to use these integrated alternatives for scenarios similar to our research setup.

1.1 Overview of Thesis

In Chapter 2 we introduce the research environment together with a suitable acoustic signal model to enable the discussion of goals and results. We also talk about the high level ideas and components of the research. In Chapter 3, we go over commonly used audio signal processing techniques which will be used in the different experimental setups. We discuss their theoretical foundations together with some of their properties. The techniques are sorted by their application purposes, first noise reduction techniques, then echo cancellation algorithms and finally techniques

that provide an integrated solution for both issues. When we have established this foundation, we discuss the different experimental configurations and some of our expectations in Chapter 4. These setups are tested both in Matlab simulations and in real-time implementations, which we discuss in Chapter 5 and Chapter 6 respectively. Here we focus on the practical implementations and comparing the results in order to draw our conclusions. In the final Chapter 7, we give a high level overview of the conclusions from all the experiments and discuss which aspects require further research.

Chapter 2

Overview of System and Signal Model

2.1 Research Scenario

In order to tackle our research goal, we start off by constructing an appropriate scenario in which we can unambiguously formalize the problem statement, propose different solutions and compare the performances of these solutions. To achieve this, the chosen scenario has some requirements:

- The scenario should be a representation of a realistic application where both noise reduction and echo cancellation are required. It is unreasonable to research and draw conclusions based on a situation which is not relevant in real life.
- The exact setup in our context is not explored before in other research.
- It is possible to implement the chosen scenario in the audio lab to conduct the real-time experiments.
- The scenario allows for an accurate representation of the performances of our different algorithms. This means there should be a minimal amount of uncontrollable external variables and the results should be similar in different variations of the setup.
- The setup allows for easy adjustment of the relative volumes and positions of the sound sources, in order to test the robustness of the proposed solutions.

The scenario that we picked, coined the 'Robust conference call' setup, is based on the setup discussed in [7]. This setup is very suitable for our research, although it has to be slightly adjusted in order to fully represent our particular problem statement.

The original paper describes following scenario: our hearing impaired subject is located in a room together with multiple sound sources. Some of these sources are

desirable for the subject to understand so this person can interact with them. Some are unwanted and should be suppressed. As usual when talking about hearing aids, there is one desired speech signal coming from a person speaking to our subject. The paper makes a more general case, considering a second desired audio signal which originates from an external audio device. The hearing aid can receive the clean playback version of this second source through a wireless connection between the hearing aid and the source of this signal (such as the TV set). Finally, there are also undesired sources in the room: people whom our subject is not conversing with.

The setup used in this thesis is based on the preceding scenario, but with some modifications.

- First and foremost, we do not consider the external reference signal as continuously active, instead it also has an on/off character. The assumed continuity of the reference signal makes the solution to noise reduction problem more straightforward, because it only considers two input scenarios: desired speech signal on and off, with the reference signal continuously on. In our scenario, we work with four different segments: each combination of the desired speech on and off with the reference signal on and off. When dealing with only two segments, the standard noise reduction algorithms can be applied, requiring a Voice Activity Detector (VAD) to check whether the speech is active or not. Now, some of the algorithms require two VADs to check the activity of both desired and reference signal.

- Another modification is the nature of our reference signal. In the paper, this external audio device is considered a second desired signal, whereas we consider this an undesired source and want to actively cancel this component out.

- The last modification is the type of noise reduction algorithms used in the proposed solutions. Since the paper was written, there have been a paradigm shift in the technique used to subtract the background noise from the input signals of the model. We use the more modern equivalent of this subtraction, which is based on Generalized Eigenvalue Decompositions (GEVD) and is more thoroughly explained in section 3.2.3.

These modifications result into the research scenario that we use throughout this thesis. It consists of a closed room where the conference call is set up. Our subject calls the correspondent on a screen that is part of full audiovisual setup. From the perspective of our system, the talking subject is considered the desired speech. Groups of people speaking to each other in the same room produce ambient noise, which is modelled by multiple babble noise sources. In order for our correspondent to understand what the subject is saying, different arrays of microphones are placed in the room, recording all the auditory activity. To establish the bidirectional nature of the conference call, there are two loudspeakers in the room, playing the audio of our correspondent. These broadcasts also get picked up again by the microphones, creating the echo scenario.

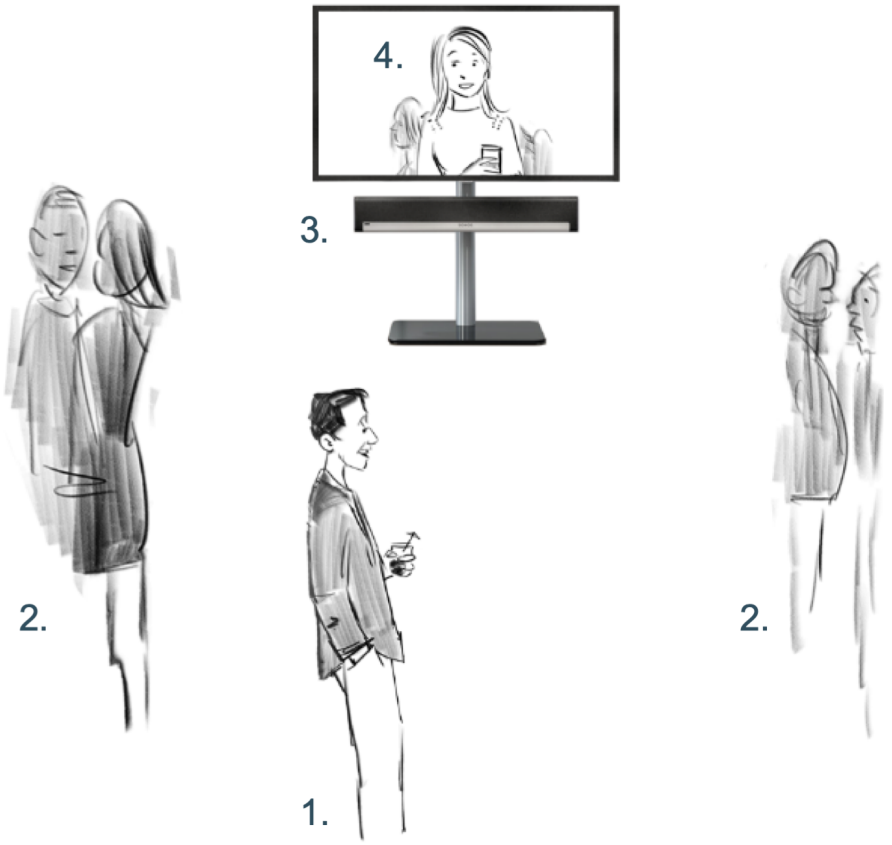


FIGURE 2.1: Illustration of Research Scenario.

Figure 2.1 gives an oversight of the proposed research setup. The user on our side of the system (nr.1) is considered the speaker with desired speech. Other people in the room (nr.2) are sources of babble noise. Microphones are set up in an array (nr.3) to record the room and act as inputs for the audio processing algorithms. Our correspondent (nr.4) talks back to our user. Whenever they respond, the speech is played back through our TV speakers so it is audible for our user. This speech signal also gets picked up by the microphones (which are usually positioned close to the TV speakers), creating the echo scenario.

2.2 Acoustic Signal Model

First off, in order to explain the techniques used in this thesis, we start off with introducing the signal model. Signals recorded by the sensors contain three types of data.

The desired speech signal of which we want to boost its clarity is called the Near

End (NE) signal. In our applications this is usually the target speech spoken by one person, meaning it is very variant in time plus if the target moves around in the room, it is also spatially variant. From now on, we will refer to the NE signal as $S_p^{src}[l]$, where l denotes the time domain index. The index p is added to select one of the desired speech sources in the case of multiple desired speech sources.

Next, our model contains different types of acoustic sources emitting signals of which we want to minimize their energy in the final output signal. It is important to note that we have no information whatsoever about some of these sources, neither the position in space, nor the acoustic path to the sensors nor the data that is being transmitted. We collect all the sources that meet these specifications under one umbrella term: noise signals. We will refer to noise signals as $N_r^{src}[l]$, where l denotes the time domain index and r selects one of the noise sources.

In practice, we consider two types of noise signals: namely Additive white Gaussian noise (AWGN) and Babble noise. Babble noise is generated by people talking in the vicinity of the target speech. This type of noise is particularly challenging to attenuate by signal processing, because of its highly time evolving structure and the fact its properties are very similar to the desired Near End signal.

Finally, the model contains an acoustic source of which the processing unit has access to the clean playback signal. We will refer to this signal as the Far End (FE) signal, the interference signal or $I_q^{src}[l]$, where l denotes the time domain index and q selects one of the interfering speech sources in case of multiple interference sources. In a video conference setting, this interfering signal usually is the speech of our correspondent being played back by our device's speakers. The processing has access to this clean signal because it has been transmitted to our device for playback.

If these time domain signals have superscript src, they are a clean source file. When we record the signals with the microphones, we can either look at the array of signals $\boldsymbol{Y}[l]$ or at one microphone channel $Y_k[l]$.

The recorded signal can contain all of the above mentioned types of signals in an additive manner. We formalize this as:

$$Y_k[l] = S_k[l] + I_k[l] + N_k[l] \tag{2.1}$$

where k selects one of the microphones en l is the time domain sample index. Each of these components can be written as a sum of convolutions of sources and transfer functions:

$$Y_k[l] = \sum_p \left(\boldsymbol{a}_{p,k}^S * S_p^{src} \right) + \sum_q \left(\boldsymbol{a}_{q,k}^I * I_q^{src} \right) + \sum_r \left(\boldsymbol{a}_{r,k}^N * N_r^{src} \right) + N_{other,k}[l] \tag{2.2}$$

where p iterates over all the desired speech sources S_p^{src}, q over all the interfering speech sources I_q^{src} and r over all the localized noise sources N_r^{src}. $\boldsymbol{a}$ contains the individual transfer function for a sound source p, q or r to microphone k, which convolutes with the corresponding source signals. $N_{other,k}[l]$ contains other relevant sources of noise, e.g. sensor noise, quantization noise and diffused noise.

In case of one desired and one interfering speech source ($p = 1$ and $q = 1$) and if we omit these indices, this formula reduces to:

$$Y_k[l] = \boldsymbol{a}_k^S * S^{src} + \boldsymbol{a}_k^I * I^{src} + \sum_r \left(\boldsymbol{a}_{r,k}^N * N_r^{src} \right) + N_{other,k}[l] \tag{2.3}$$

Since we exclusively discuss signal processing techniques in the frequency domain, we have to convert this signal model to the frequency domain. The model has to consider real-time applications, so we can not use a plain discrete Fourier transformation on the full signal. Instead, we use a windowed short-time Fourier transform (STFT) [8] in order to extract frequency and phase information of local sections of the signals [9].

$$y_k[m, w] = \mathrm{STFT}\left\{ Y_k[l] \right\} \tag{2.4}$$

where m is the frequency bin index and w is the window shift.

Similar to the time domain, when we talk about the array of all microphone signals, we use $\boldsymbol{y}[m, w]$. Otherwise, if we focus on one channel, we use $y_k[m, w]$. Since we use this notation frequently, we omit the indices and write $\boldsymbol{y}$ or y_k instead.

In the discrete frequency domain, formula (2.3) turns into:

$$y_k = \boldsymbol{a}_k^s \; s^{src} + \boldsymbol{a}_k^i \; i^{src} + \sum_r \left(\boldsymbol{a}_{r,k}^n \; n_r^{src} \right) + n_{other,k} \tag{2.5}$$

where s^{src} is the desired speech source in frequency domain, i^{src} the interfering speech source and n_r^{src} all the localized noise sources with index r. $\boldsymbol{a}$ contains the respective steering vectors. $n_{other,k}$ contains other relevant sources of noise. This equation is only valid by approximation, since the transformation to frequency domain is done by an STFT with a limited window length.

To verify how effective the noise reduction and echo cancellation of a processing unit in a certain scenario is, we have to introduce some qualitative measures. These are explained in the next chapter.

2.3 Combined Solutions

The classical way of combining echo cancellation and noise reduction in a network of acoustics devices is straightforward: install an echo cancellation unit, e.g. an adaptive (MS) filter followed by a noise reduction unit, e.g. a Multi-channel Wiener Filter (MWF). The MS filter will attenuate the interfering signal, also indicated as the Far end (FE) speech signal, while trying to leave the useful Near end (NE) signal unchanged. The MWF filters and combines the recorded microphone signals in order to attenuate the non-desired components of the signal, which we simply call noise. Next to the well established two-step approaches, it is also possible to perform both noise reduction and echo cancellation with one single filter. These are the so-called joint noise reduction and echo cancellation systems and are the main subject of research in this thesis.

Chapter 3

Audio Signal Processing Techniques

3.1 Introduction

In this chapter, we discuss the different audio signal processing techniques that are relevant for our experimental setups. First we go over the algorithms that exclusively focus on reducing the noise volume in the microphone signals. After that, we discuss techniques that work solely on cancelling out the echo signal. Finally, we propose and discuss possible solutions that have both a noise reduction and echo cancellation effect on the recorded signals.

3.2 Noise Reduction

3.2.1 Measurement of NR Performance

The noise reduction is measured here by calculating the difference between the Signal-to-noise ratios (SNR) of the input and output signals. This SNR is calculated by dividing the power of our NE signal by the power of the noise and converting it to the logarithmic scale. Formally:

$$SNR_{dB} = 10 \log_{10} \frac{E\left\{\left|S_{ref}\right|^2\right\}}{E\left\{\left|N_{ref}\right|^2\right\}} \tag{3.1}$$

$E\{.\}$ denotes the expected value operator. For SNR measures on the input side of the filter, S_{ref} and N_{ref} are the desired speech signal component $(\boldsymbol{a}_k^S * S^{src})[l]$ and noise signals components $\sum_r \left(\boldsymbol{a}_{r,k}^N * N_r^{src}\right)[l] + N_{other,k}[l]$ of a reference microphone $(k = ref)$ in the time domain. For output SNR measures, it is the desired speech signal and noise component in the output signal after the total filter.

If the difference in SNR is high, the noise of the environment is being attenu-

ated considerably and the noise reduction is effective.

3.2.2 Multi-channel Wiener Filtering (MWF)

The algorithm that we use for noise reduction is called Multi-channel Wiener filtering (MWF). Multi-channel means that the algorithm requires multiple channels of audio signals. We can supply this data by either directly connecting our algorithm with the raw recordings of the microphones, or by first pulling these signals through an MS filter. In any case, the input to this filter is an array of M signals called $\boldsymbol{y}[m, w]$. (Bold type means that $\boldsymbol{y}[m, w]$ is a vector containing the M different microphone signals $y_k[m, w]$ in short-time frequency domain, we will omit the indices for the sake of brevity). This type of filter does not differentiate between noise and interfering signals, they are both considered unknown noise sources. Therefore:

$$y_k = s_k + n_k \tag{3.2}$$

with $k = 1 \ldots$ M denoting the microphone channels. s_k contains the speech component of microphone k in the short-time frequency domain, n_k contains all signals that are considered noise.

The goal of MWF is to try and recover the useful desired signal from the recorded signals that are corrupted by noise. This is done by filtering and combining the $\boldsymbol{y}$ signals with the filter $\boldsymbol{w}$ in order to minimize the mean squared difference between the filtered output $\boldsymbol{w}^H\boldsymbol{y}$ and the desired output s_{ref} (from the point of view of one microphone, the reference microphone). Formally, this is a Linear Minimum Mean Square Error (LMMSE) estimator given by:

$$\boldsymbol{w}_{opt} = \arg\min_{\boldsymbol{w}} E\left\{\left|s_{ref} - \boldsymbol{w}^H\boldsymbol{y}\right|^2\right\} \tag{3.3}$$

$E\{.\}$ once again denotes the expected value operator and the superscript H denotes the complex conjugate transpose.

This estimator has a fixed solution which was deducted by Nobert Wiener in [10] with the multi-channel case explained in [11]:

$$\boldsymbol{w}_{opt} = R_{\boldsymbol{yy}}^{-1} R_{\boldsymbol{ss}} \boldsymbol{e}_{ref} \tag{3.4}$$

where:
$R_{\boldsymbol{yy}} = E\{\boldsymbol{yy}\}^H$ is the correlation matrix of the observed signals
$R_{\boldsymbol{ss}} = E\{\boldsymbol{ss}\}^H$ is the correlation matrix of the desired signal components
$R_{\boldsymbol{nn}} = E\{\boldsymbol{nn}\}^H$ is the correlation matrix of the noise components
$\mathbf{e}_{ref} = [0 \ldots 0\ 1\ 0 \ldots 0]^T$ selects the reference microphone in the array

We require an approximation of $R_{\boldsymbol{yy}}$ for each frequency bin m and during each short-time Fourier transformation (STFT) window w. In practice, we cannot calculate the true expected value of these correlation matrices. Instead we can estimate these matrices by signal averaging. In our implementation of the different tested algorithms, we use a moving average to update correlation matrices:

$$\tilde{R}_{\boldsymbol{yy}}[m,w] = \lambda\ \tilde{R}_{\boldsymbol{yy}}[m,w-1] + (1-\lambda)\ \boldsymbol{y}[m,w]\boldsymbol{y}^H[m,w] \tag{3.5}$$

where $\boldsymbol{y}$ is the array of input signals for all microphones in frequency bin m and STFT window w. The forgetting factor for the moving average is represented by λ. A value of λ close to zero gives more weight to the current observation, while a value close to one gives more weight to the previous smoothed estimate.

These different correlation matrices have to be estimated using only the relevant data, meaning $\tilde{R}_{\boldsymbol{nn}}$ should only be updated when the desired speech is inactive. To estimate whether the speech is active or not, we use a voice activity detector (VAD) [12].

We need to calculate the speech correlation matrix, which is required for calculating the optimal filter coefficients $\boldsymbol{w}_{opt}$ and to finally give our estimation of the desired speech signal $\boldsymbol{w}_{opt}^H\boldsymbol{y}$. In the case of regular MWF, we assume that the speech and noise signals are completely uncorrelated, meaning the correlation matrices are additive:

$$R_{\boldsymbol{yy}} = R_{\boldsymbol{ss}} + R_{\boldsymbol{nn}} \tag{3.6}$$

Estimating the speech correlation matrix is extremely easy in this case and done by subtracting the noise data from the observed matrix:

$$R_{\boldsymbol{ss}} = R_{\boldsymbol{yy}} - R_{\boldsymbol{nn}} \tag{3.7}$$

However, the assumption of uncorrelated noise is quite substantial in most practical scenarios. This assumption is only entirely satisfied in cases where the signature of the noise is completely white. Other scenarios are possible where the correlation is small enough to be ignored. However, most practical applications involve the appearance of babble noise and since this type of noise is basically dispersed speech, the correlation with the desired speech signal is likely to be quite high.

Since we have to calculate estimations of these correlation matrices, these values are never completely correct. On top of that, if the noise contains babble, it is even possible that $\tilde{R}_{\boldsymbol{nn}}$ is greater than $\tilde{R}_{\boldsymbol{yy}}$, leading to a useless $\tilde{R}_{\boldsymbol{ss}}$ matrix that is not positive semidefinite. This phenomenon is called oversubtraction. To overcome these flaws, we consider a specific estimation procedure to obtain $\tilde{R}_{\boldsymbol{ss}}$, together with

a formulation of the resulting MWF.

3.2.3 Generalized Eigenvalue Decomposition (GEVD)-based MWF

It is possible to jointly diagonalize the correlation matrices $R_{\boldsymbol{yy}}$ and $R_{\boldsymbol{nn}}$, using the generalized eigenvalue decomposition (GEVD) of the matrix pencil $\left\{R_{\boldsymbol{yy}}, R_{\boldsymbol{nn}}\right\}$. This operation results in:

$$R_{\boldsymbol{yy}} = Q\Sigma_{\boldsymbol{y}}Q^H \tag{3.8}$$

$$R_{\boldsymbol{nn}} = Q\Sigma_{\boldsymbol{n}}Q^H \tag{3.9}$$

$$R_{\boldsymbol{ss}} = Q(\Sigma_{\boldsymbol{y}} - \Sigma_{\boldsymbol{n}})Q^H \tag{3.10}$$

where Q is an invertible matrix, the columns of which are normalized and define the generalized eigenvectors. $\Sigma_{\boldsymbol{y}}$, $\Sigma_{\boldsymbol{n}}$ are real-valued diagonal matrices with:

$$\Sigma_{\boldsymbol{y}} = \text{diag}\left\{\sigma_{\boldsymbol{y}}^1 \ ... \ \sigma_{\boldsymbol{y}}^M\right\} \tag{3.11}$$

$$\Sigma_{\boldsymbol{n}} = \text{diag}\left\{\sigma_{\boldsymbol{n}}^1 \ ... \ \sigma_{\boldsymbol{n}}^M\right\} \tag{3.12}$$

The estimated $\tilde{R}_{\boldsymbol{ss}}$ has in practice generally a greater rank than the amount of desired speech sources, and it may even not be positive semidefinite. In this case, it has been demonstrated [13][14] that incorporating a low rank approximation based on the generalized eigenvalue decomposition (GEVD) of $\tilde{R}_{\boldsymbol{yy}}$ and $\tilde{R}_{\boldsymbol{nn}}$ enhances the estimation performance of the MWF, especially in low-SNR conditions. The GEVD-based rank R approximation has been shown to deliver the best performance, as it effectively selects the R modes corresponding to the highest SNR. This technique first decomposes the problem into generalised eigenvalues and then subtracts the noise from the rest of the desired subspaces. This makes the problem of oversubtraction irrelevant.

Rank 1 Approximation

In some of the setups, we use a rank 1 approximation which selects only the subspace that contains the single desired speech signal. This mean that we only consider the eigenvalue with the highest ratio of $\frac{\sigma_{\boldsymbol{y}}}{\sigma_{\boldsymbol{n}}}$ and its corresponding eigenvector. This method selects the mode with the highest SNR of $\tilde{R}_{\boldsymbol{ss}}$, which is assumed to refer to the desired speech signal.

$$\tilde{R}_{\boldsymbol{ss}} = \begin{bmatrix} \boldsymbol{q}_a & \mathbf{0} & \dots \end{bmatrix} \begin{bmatrix} \sigma_{\boldsymbol{y}}^a - \sigma_{\boldsymbol{n}}^a & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_a & \mathbf{0} & \dots \end{bmatrix}^H \tag{3.13}$$

where index a is the mode with the highest SNR ratio of $\sigma_{\boldsymbol{y}}/\sigma_{\boldsymbol{n}}$ together with its corresponding eigenvector $\boldsymbol{q}_a$. Based on the relative input volumes and VAD information, we assume that this mode selects the near end speech.

However, this technique is prone to misjudgement. In scenarios where the power of the interference signal is larger than the desired speech and the VAD information of the two signals overlap often, it is possible that the rank 1 approximation selects the far end speech instead or a combination of both. Even when the VAD information is clear enough and puts the interference in $\tilde{R}_{\boldsymbol{nn}}$ (considering the echo signal as just another noise source), in some cases this large approximation can result in a worse filter, compromising the noise reduction performance of the MWF.

Rank 2 Approximation

In other setups, we want to leave both the desired and interfering signals intact while still reducing the background noise of the microphone signals. This is done by a rank 2 approximation after the GEVD.

$$\tilde{R}_{\boldsymbol{ss+ii}} = \begin{bmatrix} \boldsymbol{q}_a & \boldsymbol{q}_b & \dots \end{bmatrix} \begin{bmatrix} \sigma_{\boldsymbol{y}}^a - \sigma_{\boldsymbol{n}}^a & 0 & \dots & 0 \\ 0 & \sigma_{\boldsymbol{y}}^b - \sigma_{\boldsymbol{n}}^b & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_a & \boldsymbol{q}_b & \dots \end{bmatrix}^H \tag{3.14}$$

where index a is the mode with the highest SNR ratio of $\sigma_{\boldsymbol{y}}/\sigma_{\boldsymbol{n}}$ together with its corresponding eigenvector $\boldsymbol{q}_a$. Similarly, b selects the mode with the second highest SNR ratio and its eigenvector $\boldsymbol{q}_b$. Based on the relative input volumes and VAD information, we assume that these two modes contain both the near end and far end signal.

It is important to note that a rank 2 approximation requires the VAD information of both desired subspaces. $R_{\boldsymbol{yy}}$ has to be built upon the complete speech signals of both NE and FE sources. If some parts of these signals end up in $R_{\boldsymbol{nn}}$, they are considered noise for that segment of processing and the subtraction causes the signal to get suppressed by the resulting filter.

3.3 Echo Cancellation

3.3.1 Measurement of EC Performance

The measure for echo cancellation is similar but considering the power of the interfering signal instead of the noise. This ratio is called the Signal-to-interference (SIR) ratio and is calculated by this formula:

$$SIR_{dB} = 10 \log_{10} \frac{E\left\{\left|S_{ref}\right|^2\right\}}{E\left\{\left|I_{ref}\right|^2\right\}} \tag{3.15}$$

$E\{.\}$ denotes the expected value operator. For SIR measures on the input side of the filter, S_{ref} and I_{ref} are the desired speech signal component $(\boldsymbol{a}_k^S * S^{src})[l]$ and interference signal component $(\boldsymbol{a}_k^I * I^{src})[l]$ of a reference microphone ($k = ref$) in the time domain. For output SIR measures, it is the desired and interference component in the output signal after the total filter.

As in resemblance to the SNR, when the processing delivers a high gain in SIR, the interfering signal played by the device speakers is attenuated well and the echo cancellation is effective.

3.3.2 Mean Squares Filtering (MS)

An established way of applying echo cancellation is an acoustic Mean Squares (MS) filter. To cancel out the undesired echo present in a microphone signal, the filter looks for an approximation of the acoustic path between the interference source and the microphone. It then filters the source signal with this approximation and then subtracts it from the microphone signal. This problem is formalized as a following minimization for each of the recorded channels. Since we assume that there is only one interference source signal active, we write i_{src} instead of the general i_q^{src}:

$$w_k = \arg\min_{w_k} E\left\{\left|i_k - w_k^H i_{src}\right|^2\right\} \tag{3.16}$$

where i_k contains the interference frequency data for one frequency bin during a single STFT window, recorded by the k-th microphone. i_{src} contains the clean unfiltered version of the interference signal and w_k contains the filter coefficient which is an approximation of the acoustic path between the interference source and the k-th microphone in one frequency bin.

This problem has a fixed optimal solution for the filter coefficient:

$$w_k = R_{ii}^{-1} R_{i_{src} i_k} \tag{3.17}$$

$R_{ii} = E\{i_{src}\ i_{src}\}$ is the energy of the interference speech channel.
$R_{i_{src} i_k} = E\{i_{src}\ i_k\}$ is the correlation between the clean interference speech i_{src} and the recorded version of the interference at microphone k called i_k.

The R_{ii} and $R_{i_{src} i_k}$ correlation matrices are calculated based on the y_k signal when s_k is inactive, in order to avoid that the MS filter also cancels out desired speech.

The update is only done when the VAD signals indicate that the desired speech is inactive in the selected STFT window. This implies that the MS filter does not use all the information contained in the full echo signal.

When the MS filter is applied, every output channel now contains $s_k + n_k + i_k - w_k^H i_{src}$. These signals are now an echo-reduced version of the recorded microphone signals. In order to also reduce the n_k portion of the recordings, these signals can then be fed into a noise reduction algorithm. We call these techniques two-step methods for noise reduction and echo cancellation. The goal of this thesis is to consider these two-step algorithms as benchmarks for evaluating joint algorithms, meaning one-step variations, by developing real-time implementations.

3.4 Combined Noise Reduction and Echo Cancellation Techniques

3.4.1 Two-step MS and MWF Filtering

The most straight-forward way to create a solution that performs both noise reduction and echo cancellation, is to combine an algorithm that reduces the noise with another block that suppresses the echo. We use a MWF implementation as the noise reduction component and an MS filter for echo cancellation. This results in multiple methods that depend on the order of the different stages. However, we use this particular order as a performance benchmark throughout the thesis. In section 4.2, we go over the different configurations used in the experiments, accompanied with a clarification of each version and what our expectations are in the tests. It is possible to first have the MS filter handle the interference on each channel separately and then let the MWF work on cancelling out the remaining noise. An alternative setup switches these two stages around, first applying the MWF and then taking care of the remaining interference by the MS filter. Both these variants are discussed further on in the text.

3.4.2 Far End Dimension Extension

The MWF provides us with a very robust algorithm for noise reduction. However, we have access to the clean interference signal at every stage of the audio processing, so we might be able to use it in order to develop better MWF coefficients that also actively suppress the echo. We give the filter access to the far end signal by concatenating it to the input array. The rank 1 GEVD-MWF based on the VAD of the desired speech signal considers the echo signal as just another noise source. Now that the filter has access to the source of this noise, it should be able to cancel it out more effectively. The full mathematical derivation is found in section 4.1 and 4.2 of [7]. The input to our extended MWF is:

$$\boldsymbol{y} = \begin{bmatrix} y_1 & \dots & y_M & i_{src} \end{bmatrix}^T \tag{3.18}$$

where $\boldsymbol{y}$ is the input array for the MWF, $y_1 \ \dots \ y_M$ are all the microphone signals in short-time frequency domain and i_{src} is the far end signal.

The usual MWF processing of section 3.2.3 still applies, however, all the correlation matrices now have an extra column and row. The GEVD subtraction consequently has an increased size, which hinders performance significantly since its mathematical complexity scales with $\mathrm{O}(mn^2)$ for an $m \times n$ matrix [15]. The optimal filter equation is still the same, but now has an extra dimension:

$$\boldsymbol{w}_{opt} = R_{\boldsymbol{yy}}^{-1} \tilde{R}_{\boldsymbol{ss}} \mathbf{e}_{ref} \tag{3.19}$$

3.4.3 Far End Signal Subtraction

We know that it is possible to suppress the echo component of a microphone recording by applying an MS filter: given the clean far end signal and a microphone signal, the adaptive filter subtracts the best possible linear estimation of the interference information in each channel. But what if we can suppress the interference signal directly based on the information that is available in the MWF algorithm? If we apply the far end extension as explained in section 3.4.2, this direct subtraction is possible.

As usual, our MWF collects the microphone information and builds $R_{\boldsymbol{yy}}$ and $R_{\boldsymbol{nn}}$, but now based on the information of both NE and FE VAD signals. $R_{\boldsymbol{yy}}$ contains information about both our desired and interfering speech, and also about the ambient noise sources. $R_{\boldsymbol{nn}}$ contains exclusively noise information since we update it only when both NE and FE are inactive. First, the GEVD subtraction part of the MWF still suppresses the noise by subtracting $R_{\boldsymbol{nn}}$ from $R_{\boldsymbol{yy}}$ and selecting the left-over desired subspaces. However, since we also want access to the far end signal in order to subtract it from the remaining correlation matrix, we select both the NE and FE subspace to remain. This subtraction results in a rank 2 matrix we call $\tilde{R}_{\boldsymbol{ss+ii}}$.

We realised that after the rank 2 GEVD subtraction, the last column of $\tilde{R}_{\boldsymbol{ss+ii}}$ always contains the coefficients associated with the clean far end signal.

$$\tilde{R}_{\boldsymbol{ss+ii}} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1(m+1)} \\ r_{21} & r_{22} & \dots & r_{2(m+1)} \\ \vdots & \vdots & \ddots & \\ r_{(m+1)1} & r_{(m+1)2} & & r_{(m+1)(m+1)} \end{bmatrix} \tag{3.20}$$

We extract the far end component in the last column of $\tilde{R}_{\boldsymbol{ss+ii}}$ and normalize it by

$r_{(m+1)(m+1)}$, which contains the power of the far end subspace:

$$\tilde{\boldsymbol{a}}^I = \begin{bmatrix} r_{1(m+1)} & r_{2(m+1)} & \dots & r_{(m+1)(m+1)} \end{bmatrix}^T / \sqrt{r_{(m+1)(m+1)}} \tag{3.21}$$

To enable the FE subtraction, this basis is then transformed into a correlation matrix, so it is compatible with $\tilde{R}_{\boldsymbol{ss+ii}}$:

$$\tilde{R}_{\boldsymbol{ii}} = \tilde{\boldsymbol{a}}^I \ (\tilde{\boldsymbol{a}}^I)^H \tag{3.22}$$

The FE correlation matrix is then subtracted from the joint subspace, which yields an approximated NE correlation matrix:

$$\tilde{R}_{\boldsymbol{ss}} = \tilde{R}_{\boldsymbol{ss+ii}} - \tilde{R}_{\boldsymbol{ii}} \tag{3.23}$$

The NE matrix is then used to calculate the MWF filter coefficients with the Wiener formula:

$$\boldsymbol{w}_{opt} = R_{\boldsymbol{yy}}^{-1} \tilde{R}_{\boldsymbol{ss}} \boldsymbol{e}_{ref} \tag{3.24}$$

This algorithm provides a more robust echo cancellation method than the implicit suppression a rank 1 approximation supplies, which in case of loud interference leaves open the possibility of selecting a non-desired subspace to remain in the result of the subtraction. It also uses all the information contained in the clean echo signal, as opposed to an MS filter, which only updates its echo information when the NE is inactive.

3.4.4 Subspace Projection MWF

Our objective is to be able to use all the possible information in the different recorded signals in order to get the best approximation of $R_{\boldsymbol{ss}}$. The extended MWF implementation already utilizes most of all available data, however it still disregards the on-off behaviour of the Far End signal. Since we have access to the clean signal, we can retrieve very reliable VAD data.

One way to use all the available data, i.e. the complete microphone signals and all VAD information, is based on Subspace Projection techniques [16]. It describes a method to estimate the $R_{\boldsymbol{ss}}$ and $R_{\boldsymbol{ii}}$ subspaces more effectively than using a pure GEVD-based algorithm. In the paper, these results are used to design an accurate linearly constrained minimum variance (LCMV) beamformer, however the same data can be used to design a MWF filter instead.

We start from $R_{\boldsymbol{yy}}$, which has abundant data about all the different subspaces

(NE, FE and noise). As a reminder, R_{yy} is the correlation matrix of all the recorded microphone signals, which is continuously updated while the algorithm runs. We subtract the noise information R_{nn} through a rank 2 GEVD subtraction. This subtraction results in $\tilde{R}_{ss+ii}$, the approximated correlation matrix of the NE and FE subspaces, and $\tilde{Q}_{ss+ii}$, the eigenvectors which make up the basis for our NE and FE subspaces. These estimations are reliable because they are based on well-defined and abundant input data.

$$\left[\tilde{R}_{ss+ii}, \tilde{Q}_{ss+ii}\right] = \mathrm{GEVD}_{r2}\left(R_{yy}, R_{nn}\right) \tag{3.25}$$

In the classical MWF implementation, the GEVD subtraction sorts the eigenvalues by signal-to-noise ratio based on R_{yy} and R_{nn}. It also sorts the columns of the resulting R and Q matrices in the same way. If we want to calculate R_{ss} by rank 1 approximation, we simply select the first column of R_{ss+ii}, assuming that the eigenvalue of the desired subspace has a higher signal-to-noise ratio than the interfering subspace, which is considered just another noise source. We can see that the classical approach takes a big approximation, because this choice can be wrong in case of a loud interference source and overlapping VAD information. However, the projection algorithm approaches this issue much more accurately by calculating a precise estimation of both R_{ss} and R_{ii}.

First, we estimate Q_{ss} and Q_{ii} based upon the incoming signals during (NE active, FE inactive) and (NE inactive, FE active) periods. These are the matrices which contain the eigenvectors that define the respective subspaces. Since there are only a limited amount of these specific segments available, the estimations are quite inaccurate.

$$\tilde{Q}_{ss} = \mathrm{GEVD}_{r1}\left(R_{ss+nn}, R_{nn}\right) \tag{3.26}$$

$$\tilde{Q}_{ii} = \mathrm{GEVD}_{r1}\left(R_{ii+nn}, R_{nn}\right) \tag{3.27}$$

To reduce the computational cost of the projections, only the valuable information is kept in the eigenvector matrices. A GEVD subtraction returns a full $M \times M$ matrix Q, but since there are only the same number of relevant eigenvectors as there are relevant subspaces contained in the matrix, we omit the rest of the eigenvectors. We define these submatrices:

$$Q^{sub}_{ss+ii} = \text{ The first 2 columns of } \tilde{Q}_{ss+ii} \tag{3.28}$$

$$Q^{sub}_{ss} = \text{ The first column of } \tilde{Q}_{ss} \tag{3.29}$$

$$Q^{sub}_{ii} = \text{ The first column of } \tilde{Q}_{ii} \tag{3.30}$$

If we design a MWF filter based on only the data in Q^{sub}_{ss}, there are two issues:

the scarcity of relevant data leading up to a specific point in time results in an imprecise estimation of the NE-subspace $Q_{\boldsymbol{ss}}^{sub}$, producing an inaccurate filter with sub-par noise reduction and echo cancellation performance. Second issue is that the coefficients have a slow response rate to temporal changes in the recording setup. These alterations can happen by the speech sources moving around in the recording room, the microphones getting displaced (e.g. the laptop is moved), or the acoustic properties of the room itself changing (e.g. somebody opens up a window). When there is an absence of relevant segments for a longer period of time, the filter becomes outdated quickly, degrading the performance.

We can improve the quality of $Q_{\boldsymbol{ss}}^{sub}$ and $Q_{\boldsymbol{ii}}^{sub}$ however. In theory, the columns of $Q_{\boldsymbol{ss+ii}}^{sub}$ and $\left[Q_{\boldsymbol{ss}}^{sub}, Q_{\boldsymbol{ii}}^{sub}\right]$ span the same subspace, however because we are working with estimations based on limited segments of data, this is not valid any more. This discrepancy is corrected by projecting the poorly estimated $Q_{\boldsymbol{ss}}^{sub}$ and $Q_{\boldsymbol{ii}}^{sub}$ onto the well defined joint subspace $Q_{\boldsymbol{ss+ii}}^{sub}$.

$$Q_{\boldsymbol{ss}}^{proj} = proj_{Q_{\boldsymbol{ss+ii}}^{sub}}\left(Q_{\boldsymbol{ss}}^{sub}\right) \tag{3.31}$$

$$Q_{\boldsymbol{ii}}^{proj} = proj_{Q_{\boldsymbol{ss+ii}}^{sub}}\left(Q_{\boldsymbol{ii}}^{sub}\right) \tag{3.32}$$

The projection is defined by following expression:

$$Q_{\boldsymbol{ss}}^{proj} \triangleq Q_{\boldsymbol{ss+ii}}^{sub}\left((Q_{\boldsymbol{ss+ii}}^{sub})^H\ Q_{\boldsymbol{ss+ii}}^{sub}\right)^{-1}(Q_{\boldsymbol{ss+ii}}^{sub})^H\ Q_{\boldsymbol{ss}}^{sub} \tag{3.33}$$

$$Q_{\boldsymbol{ii}}^{proj} \triangleq Q_{\boldsymbol{ss+ii}}^{sub}\left((Q_{\boldsymbol{ss+ii}}^{sub})^H\ Q_{\boldsymbol{ss+ii}}^{sub}\right)^{-1}(Q_{\boldsymbol{ss+ii}}^{sub})^H\ Q_{\boldsymbol{ii}}^{sub} \tag{3.34}$$

Note that the projection matrix of both expressions is the same, so it only has to be calculated once.

After applying these transformations, we now have better estimates of the separate subspaces. To find the actual required value of $R_{\boldsymbol{ss}}$ though, we still require corresponding coefficients for our basis vectors. The resulting coefficients alongside the calculated basis vectors should approximate the joint correlation matrix we found earlier as well as possible. We can express this task as a minimization problem:

$$\begin{bmatrix} P_s \\ P_i \end{bmatrix} = \arg\min_{P_s,P_i} \left\| \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj} \end{bmatrix} \begin{bmatrix} P_s & 0 \\ 0 & P_i \end{bmatrix} \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj} \end{bmatrix}^H - \tilde{R}_{\boldsymbol{ss+ii}} \right\|_F^2 \tag{3.35}$$

In order to find a solution for P_s and P_i, we can rewrite this equation as a least-squares problem which can easily be solved by a numerical solver. We first apply a vectorization to both matrices of the subtraction:

$$\min_{P_s,P_i} \left\| \text{vec}\left(\begin{bmatrix} Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj} \end{bmatrix} \begin{bmatrix} P_s & 0 \\ 0 & P_i \end{bmatrix} \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj} \end{bmatrix}^H \right) - \text{vec}\left(\tilde{R}_{\boldsymbol{ss+ii}} \right) \right\|_F^2 \tag{3.36}$$

To write this equation as a linear transformation of matrices, we now use following property of vectorization: $\text{vec}(ABC) = (C^{\text{T}} \otimes A)\text{vec}(B)$ [17]. This introduces the Kronecker product $\otimes$, which returns the tensor product of two matrices.

$$\min_{P_s,P_i} \left\| \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj} \end{bmatrix}^* \otimes \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj} \end{bmatrix} \begin{bmatrix} P_s \\ 0 \\ 0 \\ P_i \end{bmatrix} - \text{vec}\left(\tilde{R}_{\boldsymbol{ss+ii}} \right) \right\|_F^2 \tag{3.37}$$

Now we can omit the zeros in the coefficient matrix, together with the corresponding values of the Kronecker product:

$$\min_{P_s,P_i} \left\| \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj*} \otimes Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj*} \otimes Q_{\boldsymbol{ii}}^{proj} \end{bmatrix} \begin{bmatrix} P_s \\ P_i \end{bmatrix} - \text{vec}\left(\tilde{R}_{\boldsymbol{ss+ii}} \right) \right\|_F^2 \tag{3.38}$$

This equation now assumes the form of a normal least-squares problem, which we can give to a Matlab or Simulink numerical solver:

$$\begin{bmatrix} P_s \\ P_i \end{bmatrix} = \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj*} \otimes Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj*} \otimes Q_{\boldsymbol{ii}}^{proj} \end{bmatrix} \setminus \text{vec}\left(\tilde{R}_{\boldsymbol{ss+ii}} \right) \tag{3.39}$$

This least-squares problem is solved by following formula:

$$Z \triangleq \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj*} \otimes Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj*} \otimes Q_{\boldsymbol{ii}}^{proj} \end{bmatrix} \tag{3.40}$$

$$\begin{bmatrix} P_s \\ P_i \end{bmatrix} = (Z^H Z)^{-1}\ Z^H\ \text{vec}\left(\tilde{R}_{\boldsymbol{ss+ii}} \right) \tag{3.41}$$

Combining the resulting coefficients with the basis vectors yields our new approximation of $R_{\boldsymbol{ss}}$.

$$\tilde{R}_{\boldsymbol{ss}} = Q_{\boldsymbol{ss}}^{proj}\ P_s\ (Q_{\boldsymbol{ss}}^{proj})^H \tag{3.42}$$

Finally, we use $\tilde{R}_{\boldsymbol{ss}}$ to design the MWF filter:

$$\boldsymbol{w}_{opt} = R_{\boldsymbol{yy}}^{-1} \tilde{R}_{\boldsymbol{ss}} \mathbf{e}_{ref} \tag{3.43}$$

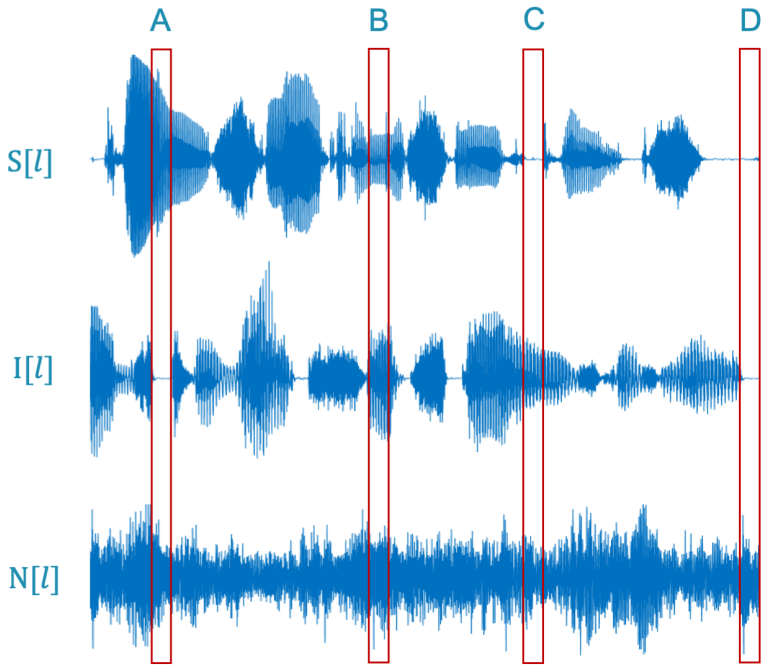


FIGURE 3.1: Possible Input Segmentations of Recorded Signals.

3.4.5 Time-Series Segmentation MWF

The idea of this technique is to update the relevant correlation matrices during every possible situation and based on this data, create two filters: one calculated specifically with the interfering signal in mind, which is made with data gathered when this signal was active. The other filter is tuned for situations when the interference is inactive, resembling the structure of a regular MWF filter. To ensure that the system applies the correct filter and updates the relevant correlation matrices at the right time, we need access to the VAD activity of both NE and FE speech. We have four different situations: In these distinct situations, R_{yy} contains unique information:

$$R_{ss+ii+nn} = R_{ss} + R_{ii} + R_{nn} \tag{3.44}$$

$$R_{ss+nn} = R_{ss} + R_{nn} \tag{3.45}$$

$$R_{ii+nn} = R_{ii} + R_{nn} \tag{3.46}$$

$$R_{nn} = R_{nn} \tag{3.47}$$

Each of these matrices is updated during specific segments to ensure there is no leakage of data. $R_{ss+ii+nn}$ is updated during every segment. R_{ss+nn} is updated during segment A and D and R_{ii+nn} during C and D. Finally, R_{nn} is only updated in segment D, during noise-only periods.

Our goal is of course to retrieve R_{ss}. In figure 3.1, we can see that in segment A and D, where there is no far end signal active, we can retrieve R_{ss} by subtracting

$R_{\boldsymbol{nn}}$ from $R_{\boldsymbol{ss+nn}}$ (preferably with GEVD subtraction, for robustness).

$$\tilde{R}_{\boldsymbol{ss}} = \text{GEVD}_{r1}\left(R_{\boldsymbol{ss+nn}}, R_{\boldsymbol{nn}}\right) \tag{3.48}$$

$$\boldsymbol{w}_{opt} = R^{-1}_{\boldsymbol{ss+nn}}\tilde{R}_{\boldsymbol{ss}} \tag{3.49}$$

In the other case, segment B and C, the far end signal is active and $R_{\boldsymbol{ss}}$ is retrieved by subtracting $R_{\boldsymbol{ii+nn}}$ from $R_{\boldsymbol{ss+ii+nn}}$, again with a GEVD subtraction.

$$\tilde{R}_{\boldsymbol{ss}} = \text{GEVD}_{r1}\left(R_{\boldsymbol{ss+ii+nn}}, R_{\boldsymbol{ii+nn}}\right) \tag{3.50}$$

$$\boldsymbol{w}_{opt} = R^{-1}_{\boldsymbol{ss+ii+nn}}\tilde{R}_{\boldsymbol{ss}} \tag{3.51}$$

In this way, we are training two different filters, one very effective for pure noise reduction and one taking in consideration the far end signal, so it is more capable of echo cancellation.

However, if we allow ourselves to switch between filters based on the VAD information, we open ourselves up to more possibilities that might lead to better SNR and SIR measures, but would impact the comfort of the listening experience. An extreme example is that we could install a noise gate which sets all the signal information during noise-only periods to zero. This would lead to a very low total energy level of the output noise and thus a good SNR score, however, this type of noise reduction does not add any qualitative processing and is not comfortable to listen to.

Since we want to evaluate fully integrated solutions, we limit ourselves to one continuous stream of filter coefficients. This is why we did not implement this solution in the next chapters.

Chapter 4

Experimental Setups and Expectations

4.1 Introduction

This chapter gives an overview of which solutions we select for further experimentation. Using block diagrams, we dissect the multiple solutions into the techniques we discussed in Chapter 3. Based on the way these techniques are used and combined, we also discuss possible expectations in performance or computational requirements.

Every diagram in this chapter has a few common entities. At the top we have the desired speech source $S^{src}[l]$ and the noise sources $N_r^{src}[l]$. This is a representation of different people talking in the room, with one of them using our conference call system. Below that, we have two loudspeakers broadcasting the interfering speech signal $I^{src}[l]$. In all of our experiments, we assume that both loudspeakers play the same signal, i.e. the far end is a mono signal. We record the room with microphone arrays, resulting in the $Y_k[l]$ time domain signals, one for each of the M microphones.

Since our algorithms all work in the discrete frequency domain, we first execute a short-time Fourier transformation (STFT) on the incoming signals, resulting in $y_k[m, w]$, where m represents the discrete frequency bin index and w the window during which the Fourier transformation is calculated. At that point, the different algorithms all do their work and end up with their approximation of the near end speech signal in frequency domain $\tilde{s}[m, w]$, which is transformed to time domain by an inverse short-time Fourier transformation (ISTFT), giving us the final result $\tilde{S}[l]$.

The filled-up microphone icon on the right represents the reference microphone. The dotted line shows which part of the setup is inaccessible for the algorithms (above) and which part can be used as inputs (below). Each solution also requires VAD information from the near end speech, some of them also require VAD data for the far end speech. The figures illustrate the conversion from signal to binary VAD

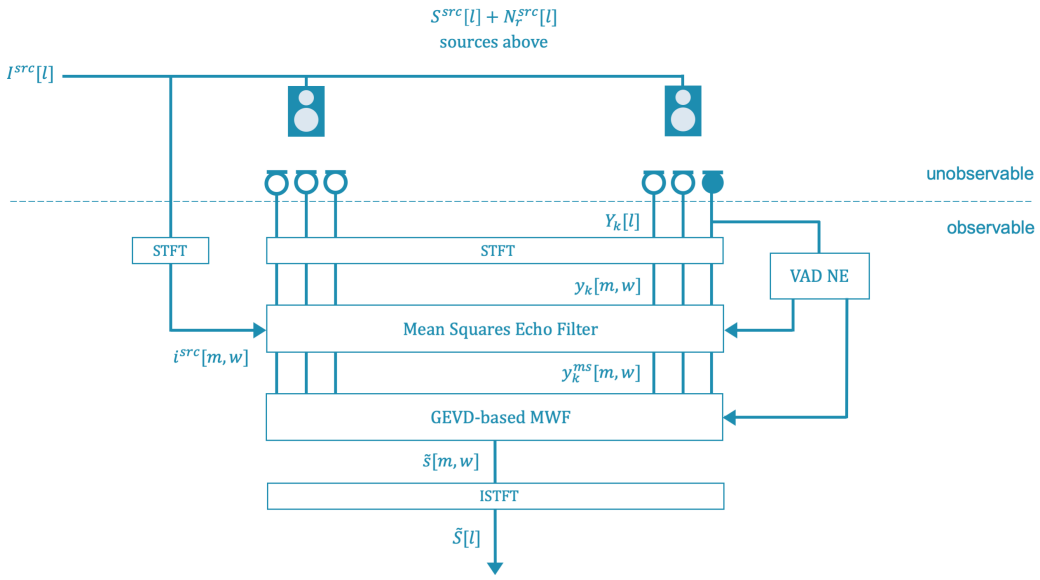


Figure 4.1: Block Diagram of Setup: MS filter followed by GEVD-MWF.

data as a VAD block.

In the first section of this chapter, we discuss the setups that have discrete components of processing, more specifically that combine two algorithms, one for noise reduction and one for echo cancellation. In the second part, we talk about the integrated solutions, which only required one step of processing to achieve both a reduction in noise and echo volume.

4.2 Discrete Setups

4.2.1 Two-step MS + MWF

This setup is extremely similar to one of the two schemes that was worked out in [7], namely scheme B. This solution boils down to first cancelling out the audio device signal (i.e. the echo signal) in each channel by approximating the transfer functions between the audio device source and each microphone. After the subtraction, the channels are concatenated and fed to a MWF in order to suppress background noise. Afterwards, the clean audio device signal is added back into the output mix. The only difference with our solution is that we do not add the filtered audio device signal back in to the output mix, since we consider this reference signal as unwanted. Figure 4.1 is a graphical representation of the setup that we use in the experiments.

We expect this setup to have a good overall performance and serve as the benchmark for other solutions. The MS filter has direct access to the (potentially loud) interference component of the microphone signals in order to determine the transfer

function between source and microphone. After this, the MWF has relatively clean signals to work with and the R_{yy} should approach rank 1, leading to a good noise reduction performance.

However, from a computational standpoint it may seem redundant to let the MS filter work on all of the microphone signals. Since the MWF is a linear operation, the MS filter should be perfectly able to cancel out the echo in only the output signal of the MWF. This brings us to the next section, where we switch the components from position.

4.2.2 Two-step MWF + MS

This setup tests if switching up the order of both components has a big impact on the performance of the algorithm as a whole. First, we collect the microphone signals and feed them into the MWF algorithm. Then, we also compare two versions of the MWF algorithm: one that outputs a rank 1 $\tilde{R}_{ss}$ and one that allows the result of the GEVD subtraction to be rank 2. When using a rank 2 approximation, the result of the subtraction is called $\tilde{R}_{ss+ii}$, since the correlation matrix has two remaining modes: the desired speech and the interfering signal. More details can be found in section 3.2.3. This technique allows the MS filter afterwards to cancel out the most clean version of the echo possible at that stage. The setup is illustrated in figure 4.2. The rank 2 implementation also requires VAD information of the FE source, which is illustrated by the dashed line on the figure.

We are sure this setup is computationally more beneficial than the previous one, but is there a trade-off in performance and if so, is it worth it? Using the rank 2 MWF, the noise reduction component should leave both desired and interfering speech intact, resulting in a similar task for the MS filter compared to the previous setup. This is why we expect the echo cancellation to have a similar performance to the MS + MWF setup, however we are not sure if the rank 2 approximation has a big impact on noise reduction performance. The experiments should clarify this. When using the rank 1 MWF however, the echo is also considered noise and is already suppressed before feeding into the MS filter. The lower volume could make it more difficult for the MS filter to determine the transfer functions. The experiments will clear up if the accumulation of echo suppression by both components is less effective than the benchmark.

4.2.3 Two-step MWF with Extension + MS

Since both components of the algorithm in case of a rank 1 MWF suppress the echo, it seems to be a good idea to give the MWF all ingredients to achieve the best echo cancellation possible. We do this by increasing the dimension of the MWF by one and adding the clean far end signal as an input for the algorithm, as explained in

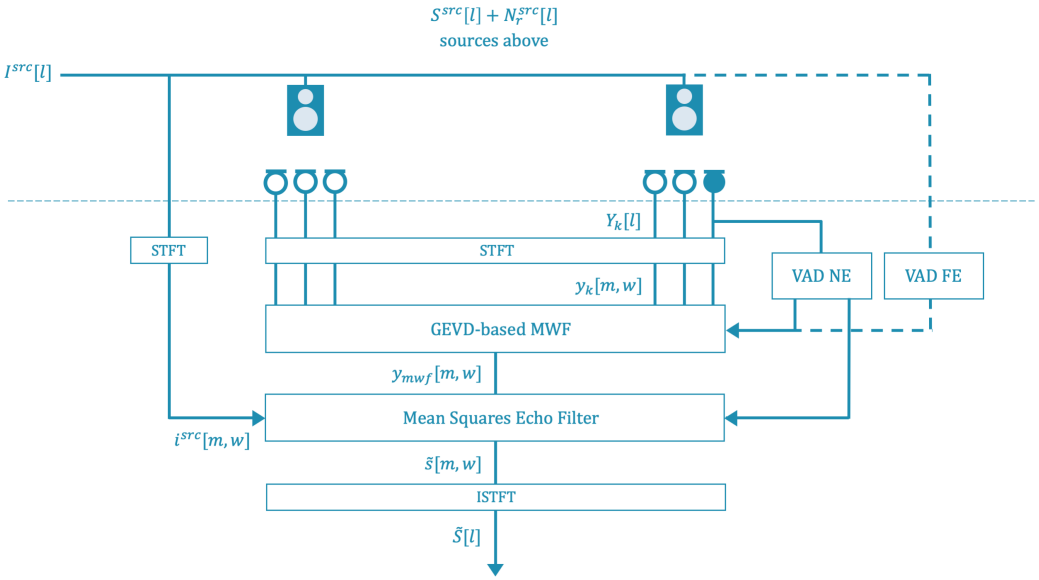


Figure 4.2: Block Diagram of Setup: GEVD-MWF followed by MS filter.

3.4.2. This leads to the setup illustrated in figure 4.3. Again, when we use the rank 2 implementation, the setup requires VAD information of the FE source, which is illustrated by the dashed line on the figure.

We expect an increase in echo cancellation performance compared to previous section because the MWF now also has access to the clean far end signal for filtering. The trade-off is the increased dimension of the problem, resulting in a heavier computational load. It is unclear how this affects the performance of the MS filter afterwards. We expect the noise reduction performance to lay in line with previous sections.

4.3 Integrated Setups

4.3.1 MWF with FE Subtraction

The first of the integrated setups is an implementation of the far end subtraction technique, as previously explained in chapter 3.4.3. It requires an increase in dimension, similar to the 'MWF with extension + MS setup', also increasing the computational requirements for the MWF. The setup itself is straight-forward and is illustrated in figure 4.4.

We expect this method to have a better performance than all of the rank 1 techniques, especially in scenarios were we have a loud interference signal. Since the selection of our desired subspace does not rely solely on signal power and VAD information (as it is in case of the rank 1 methods), there is no chance on a misinterpretation. Also, the huge approximation of the correlation matrix approaching rank 1 does not apply

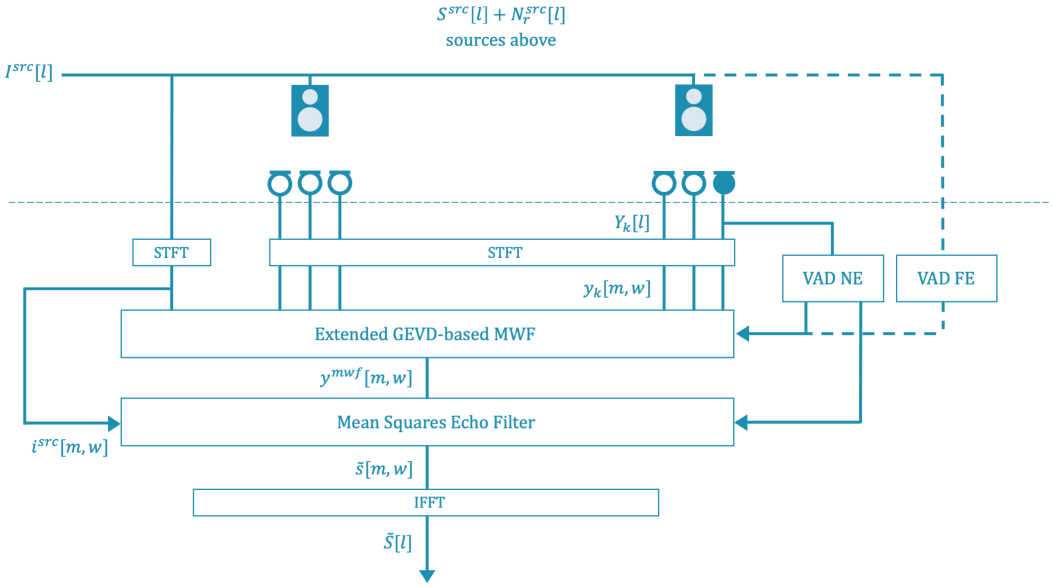


FIGURE 4.3: Block Diagram of Setup: Extended GEVD-MWF followed by MS filter.

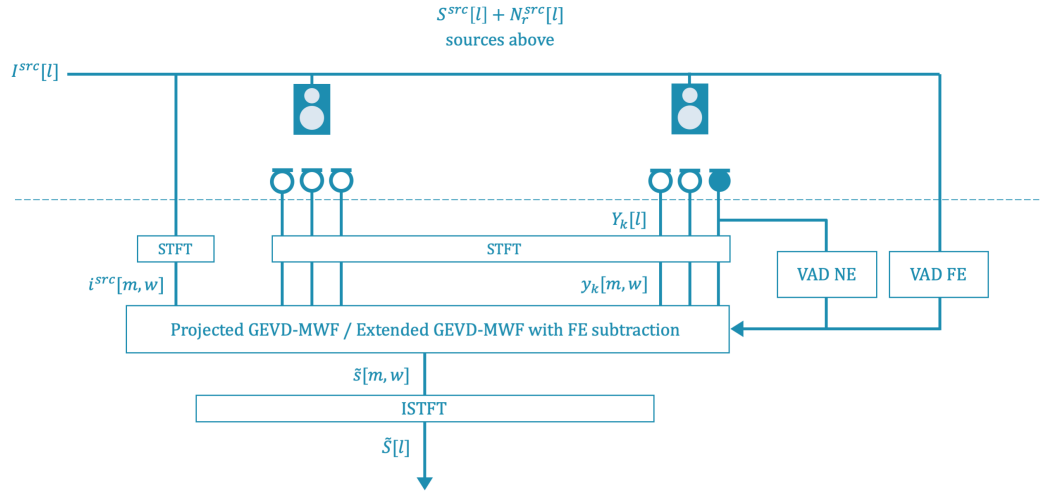


FIGURE 4.4: Block Diagram of Setup: Integrated MWF Setup.

for this solution, most likely resulting in better MWF coefficients.

4.3.2 MWF with Subspace Projection

The second integrated setup is the implementation of MWF with subspace projection (chapter 3.4.4). It also requires VAD information of the far end speech signal, in order

to build the four different correlation matrices: one for each of the possible input cases. It is computationally much more intense compared to the other algorithms, because each frequency bin requires three instead of one GEVD subtractions. However, we expect a good performance and hope to outperform the two-step benchmark, making this an interesting solution if the computational load is not an issue. The implementation is also illustrated in figure 4.4.

Chapter 5

Evaluation by Offline Simulation

5.1 Introduction

We now have an overview of the different setups we want to compare in the experiments. To start with the experiments, we have to convert all of the techniques into Matlab scripts. The algorithms are all based on an existing Matlab implementation of the GEVD-based MWF algorithm. In order to conduct our Matlab experiments, we need some additional code and settings to fully implement all of our theorized solutions.

5.2 General Code and Settings

All of the experiments (Matlab, Simulink and real-time implementation) are done at a sampling frequency of $f_s = 16$ kHz. This is high enough for speech signals to be perfectly comprehensible and low enough to be computationally acceptable. For the Fourier transform we use Hann-windowed STFT with a window size of 512 samples and with the consecutive Hann-windows overlapping by 50%.

The source files used as inputs for the Matlab models are speech files selected from the Harvard word list database, as recommended by [18]. They are then filtered by a Room Impulse Response (RIR) generator, proposed by Allen and Berkley [19], which is a very commonly used method to create room impulse responses by means of software. The function gives the user control over multiple design variables, e.g. the size of the virtual room, the reverberation time inside of the space and the coordinates of the different sound sources and microphones.

To get realistic results, we have to select a few variables in order to design the virtual setup. To achieve a close match in acoustic response in accordance to the audio lab where we conduct the real-time experiments, the shape of the room is set to a cube with an edge length of 5 meters. The reverberation time is set to a T60 of 0.22 seconds. Using this information, together with the source audio files, we use the RIR-generator to calculate the relevant impulse responses. Filtering the different

audio files with the corresponding impulse responses gives us the input audio for every microphone. These signals are then used as inputs for the algorithms that are being tested.

5.3 Additional Code

A few pieces of additional code are required because they are used by some of the experimental setups:

- Implementation of a block based Mean Squares filter that is compatible with our MWF code.
- Extending the dimension of the MWF by one, through concatenating the clean far end signal to the array of the microphone signals. This extension also needs to be considered when calculating the performance measurements, but we will explain this in the next section.
- An implementation of the far end subtraction: selection of the last column of $R_{\boldsymbol{yy}}$ after noise subtraction (selecting the far end signal dimension), normalizing it, expanding it by self-multiplication into a correlation matrix and subtracting it from the rest of $R_{\boldsymbol{yy}}$.

The subspace projection setup in itself needs quite a few additions and modifications:

- Four VAD cases instead of two, considering both NE and FE VAD information.
- For each of these cases there exists a correlation matrix that has to be updated.
- Three instead of one GEVD subtractions per frequency bin: one subtraction for equation (3.25), one for equation (3.26) and finally one for equation (3.27).
- Code that implements the projection of the lower quality subspaces $Q_{\boldsymbol{ss}}$ and $Q_{\boldsymbol{ii}}$ onto the well-defined joint subspace $Q_{\boldsymbol{ss+ii}}$, as defined in equation (3.33) and (3.34).
- Code that calculates $R_{\boldsymbol{ss}} = Q_{\boldsymbol{ss}}^{proj}\ P_s\ (Q_{\boldsymbol{ss}}^{proj})^H$.

5.4 Implementation of Performance Measures

To compare the performances of our selected algorithms, we iterate over different values of signal-to-noise ratio (SNR) and signal-to-interference ratio (SIR) at the reference microphone. We let the different algorithms process these mixes and then calculate the performance measures at the output level. Based on these values, we create plots visualizing this data to give the reader a quick overview of the performance of each algorithm.

In order to calculate the SNR, SIR and SINR values of each iteration, we need

access to the separate tracks so we can calculate the power of the processed desired, noise and interference signals separately. The ratios of these powers are our measures of noise and echo suppression. During the simulations, the algorithms obviously take in and process the actual microphone signals with filter coefficients derived from the calculated correlation matrices. Since we consider all signals additive, it does not matter if the filter coefficients are applied to the individual subspaces of the input, or to the entire mix. If we apply the calculated filters to the individual tracks (desired speech, noise and interference), we can measure the power of each subspace at the output and get reliable performance measures.

It is important to note that when evaluating an MS solution, only the far end track should be filtered with the mean squares solution. A MS filter executes a subtraction and since the signals are additive and independent, only one subtraction should be made on all of the individual tracks, otherwise we would introduce a filtered interference in the near end and/or noise track.

5.5 Results and Discussion

5.5.1 Introduction

Now that it is clear how we practically calculate the different performance measures, we look at the plots illustrating the performance of the selected algorithms, generated by the Matlab code of these solutions. Section 5.5.2 discusses the noise reduction performances by evaluating the SNR gains of our solutions. Next up, section 5.5.3 discusses the echo cancellation performances by looking at the SIR gains. Finally, in section 5.5.4, we cover the measures that illustrate the understandability of the solutions. Unlike in the real-time implementation, we do not evaluate the PESQ scores because of some practical issues with the PESQ function. However, we still discuss the SINR values.

The execution times of our Matlab algorithms depend on a lot of external variables (CPU load, other programs running, etc.). This leads to inconclusive data about the different trade-offs between performance and computational requirements. However, the execution times of the Simulink models are a lot more reliable, so comparing and discussing this data will be done in the Chapter 6.

Every subsection is accompanied with two plots. Each left sub-plot illustrates the measures when looking at situations with different volumes of noise, while keeping the interference volume constant at iSIR = 0dB. This means that at the reference microphone the interfering speech has an equal amount of power compared to the desired speech while the noise volume is variable between different input scenarios. On the left-hand side of this sub-plot we evaluate low SNR situations, meaning the babble noise in the recording environment is very loud and often overpowers both desired and interfering speech. In the middle of the graph, the noise has an equal amount of power compared to the other sound sources. On the right-hand side we

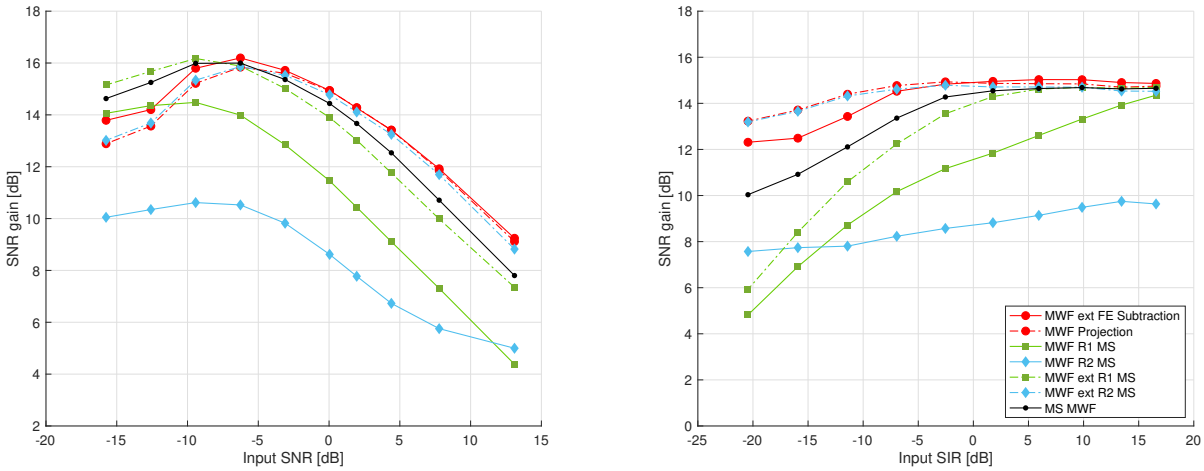


Figure 5.1: Noise Reduction Performances of Matlab Simulations.

evaluate high SNR situations, where the noise settles more quietly in the background.

The right sub-plot visualizes the measures over different interference volumes, while the noise stays at a constant iSNR = 0dB. Again, this means that at the point-of-view of the reference microphone the noise has an equal amount of power compared to the desired speech, while we change the volume of the interference for each scenario. Left-hand side of this sub-plot evaluates low SIR situations, where the interference overpowers both the desired speech and the ambient noise in the room. On the right-hand side we look at situations where the interference has a low volume and therefore approaches a pure noise reduction problem.

The figures were generated by running the algorithms for 80 seconds. The first 20 seconds of the output is not used in the calculation of the measures, since we allow the algorithms to collect enough data in order to achieve a representable solution.

5.5.2 Noise Reduction Performance

Let us first discuss the noise reduction performances of our Matlab implementations. We pay special attention to the low iSNR scenarios, since the amount of noise reduction makes a bigger perceptual impact on the quality of the processed speech in scenarios where the room is very noisy. If the noise is already low in volume at the microphone level, echo cancellation performance becomes much more important.

- The black line, representing the MS+MWF algorithm, acts as our benchmark. We can see that its performance is overall quite steady, with a good amount of noise reduction in the low iSNR scenarios (up to 16dB of gain). It does not

have any unexplainable valleys or peaks, so our expectations have been met. Some of the other solutions do better, some do worse.

- Next, we look at the MWF+MS algorithms (marked by the full green and blue lines) to see if there is a trade-off in performance when using these computationally more beneficial solutions. It is clear that the noise reduction performance suffers from switching the sequence of the MS and MWF components. Especially the rank 2 implementation does not do well in all of the input scenarios, as it is about 5dB lower than the benchmark in all of the noise scenarios and about 2-4dB lower in the interference scenarios. We discuss the computational requirements in the next chapter, to see if the decrease in execution time is great enough to justify these figures.

- The MWF+MS solutions with the extended far end dimension (dotted green en blue lines) are up next. These perform a lot better than the previous ones without the extra dimension. Both perform at about the same level as the benchmark. The rank 2 implementation even outperforms the benchmark at low iSIR values with up to 3dB of extra gain. Apparently, supplying the MWF with this extra information allows it to suppress the noise more effectively. However, since we have to increase the dimension of the problem, the computational advantage over the benchmark might be lost. This will be clarified in the next chapter.

- Both rank 1 MWF+MS implementations (full and dotted green lines) perform quite inferior on the right sub-plot, especially in low iSIR scenarios. This behaviour is expected and was already discussed in section 3.2.3: if the interference is lower in volume than the near end signal (iSIR >0dB), the approximation that the interference source can be treated as just another noise source is fair. $R_{\boldsymbol{yy}}$ after noise subtraction naturally approaches rank 1 so we do not lose much information when setting the other eigenvalues to zero. There is also very little chance that the rank 1 approximation chooses the wrong subspace, since the desired speech signal has the highest SNR. When the interference gets louder however, this misinterpretation becomes plausible. Also, even when the right subspace gets chosen, a lot of information is still thrown away by considering echo signal as noise, resulting in a bigger approximation and a worse filter. This is not the case with the rank 2 variant, where the performance is steadier (although still worse).

- Finally, the noise reduction of our integrated solutions (the full and dotted red lines) hold up really well. The only area where they get slightly outperformed by the benchmark is at very low iSNR, however the difference in gain never exceeds 1-2dB. The solution with subspace projection has a very steady and high performance in all scenarios, however we should keep the high computational cost in mind. The same steady and respectable figures are achieved by the FE subtraction method, performing slightly better at low iSNR and slightly worse at low iSIR than the projection method. This implementation is much more

efficient however, even possibly outperforming the benchmark. Once again, the next chapter will clarify this claim.

5.5.3 Echo Cancellation Performance

Next up, we discuss the echo cancellation performance of our Matlab implementations. Similar to previous section, we pay special attention to the low iSIR areas, since this is where the echo reduction is required the most and has the biggest impact on the perceptual quality of the output signal.

- Once again, we start with our benchmark MS+MWF algorithm (in black). It is clear that none of the other algorithms have the same effective and steady echo suppression than the benchmark. Especially at low iSIR, the performance is unquestionably, with SIR gains up to 28dB! As we expected, the loud interference volume in these areas gives a good input for the MS filter to determine the transfer functions, resulting in a very good approximation of the impulse responses and a very effective subtraction of the filtered echo signal. Let us compare the benchmark against the other algorithms.

- The MWF+MS solutions (full green and blue lines) are up next. On the left sub-plot, they both follow a similar trend as the benchmark, although performing a quite bit worse overall. Note that the y-scale is different than in the previous plot, a seemingly small distance on the plot translates in a fairly big difference in performance. For low iSNR scenarios the benchmark achieves SIR gains around 20dB, while these methods start out under 10dB. However, for higher iSNR the performance does increase, but they are still not up to par with the benchmark.

- The extended MWF+MS solutions (dotted green and blue lines) are next. The rank 1 method has a clear advantage over the rank 2 variant. It was not clear in advance what the interaction between the echo suppression capabilities of the rank 1 MWF and the MS filter would be. The rank 1 version actively helps cancelling out the echo, while the rank 2 leaves the interference alone. It is conceivable that the MS filter would work more effectively on the clean far end signal that the rank 2 MWF puts out, resulting in an overall better echo cancellation performance. However, the experiments clearly show that this is not the case. Comparing them to the other algorithms, the rank 2 method has quite a terrible performance overall, especially compared to the benchmark. The rank 1 variant holds up better because of its echo suppression capabilities but in general, its performance, especially in very noisy environments, is not up to par.

- Finally, we discuss the integrated solutions (in full and dotted red). The projection method has an overall good performance. On the left sub-plot, there is not a single other algorithm except for the benchmark that that achieves the same stability and effectiveness. Also on the right sub-plot it outperforms all

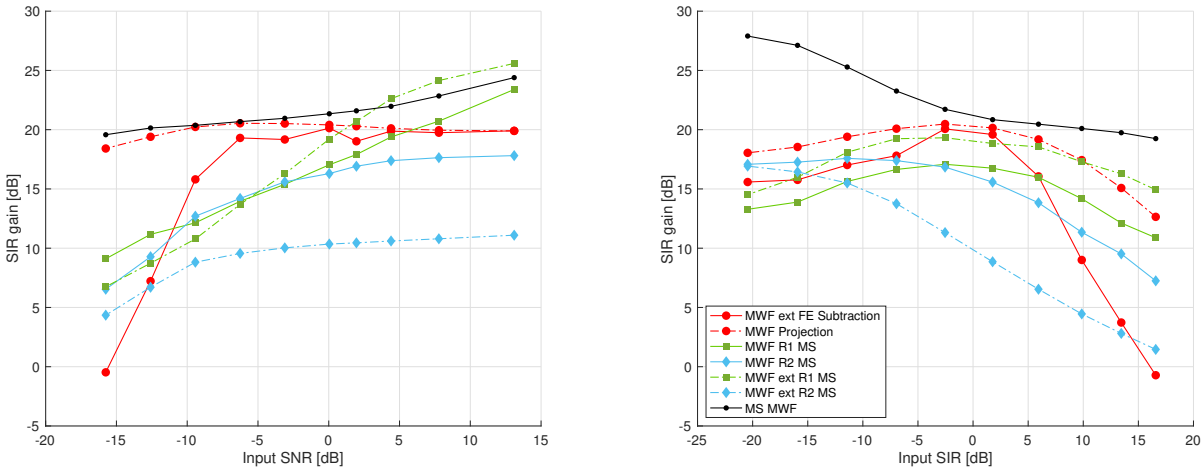


FIGURE 5.2: Echo Cancellation Performances of Matlab Simulations.

other solutions, again with the exception of the benchmark. The FE subtraction method however seems to be more inconsistent. Especially in in very noisy environments its performance starts to break down. It also seems to lack effectiveness in high iSIR scenarios, but this is not as much an issue since the interference is already low in volume at the microphone level.

5.5.4 Intelligibility Performance

As the final part of the discussion about the simulation results, we take a look at the intelligibility of the output signals. Normally, there are two measurements that we would like to discuss: the PESQ score and the SINR values. However, there was a practical problem with implementing the PESQ measurement for the Matlab algorithms. Since the code we use for calculating this score is contained in a protected Matlab file, debugging the issue could not be done in time for the thesis deadline. The reader can find a discussion on the calculated PESQ scores of the Simulink based signals in chapter 6.5.4.

We still discuss the SINR values though, giving us insight in the relative volume of the desired speech compared to all the other subspaces.

- As usual, we start off with the MS+MWF benchmark in black. The outstanding echo suppression performance is decisive for the SINR measurement as well. The great performance of the algorithm in low iSIR scenarios is also very evident in these figures. The noise reduction performance in these areas is also good enough for it not to have a big negative impact on the joint measure. We see that no other algorithm outperforms our benchmark in any of the scenarios presented.

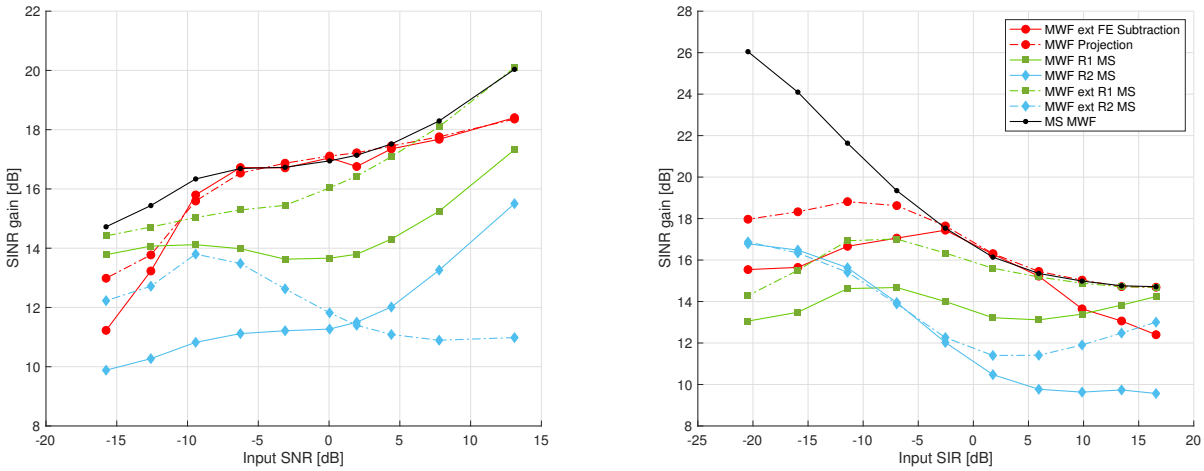


Figure 5.3: Intelligibility Results: SINR values of Matlab Simulations.

- The MWF+MS (full green and blue lines) and the extended MWF+MS implementations (dotted green and blue lines) have the same trend as in the previous figures. Their performance is not up to par compared to the benchmark. The difference in effectiveness is quite great and it seems unlikely that the gain in performance will make up for it (in case of the normal MWF+MS solutions). When looking at these figures, it seems that both blue and both green lines seem to cluster together, where in most cases both the rank 1 solutions outperform the rank 2 solutions. However, there is a flip side to this statistic: in the next chapter where we look at PESQ scores, the rank 2 solutions clearly outperform the rank 1 solutions, alongside the rest of the algorithms. This can be explained: the rank 2 solutions have a less aggressive reduction in volume of both noise and echo, but they do not throw as much information away when selecting the remaining subspace(s), resulting in less distortion of the output signal.

- Our integrated solutions (both red lines) have an overall better performance than all of the MWF+MS algorithms. Only in really noisy environments their performance drops a bit. We see the same trend in previous sections, so it is to be expected that it also shows up in this figure. In the low iSIR scenarios, they both perform well but unfortunately not up to par with our benchmark.

5.6 Conclusion

In general, the Matlab experiments show that first applying the Mean Squares filter on all the microphone recordings to reduce the echo volume in each of the channels and then feeding these signals into a rank 1 GEVD-based MWF for noise reduction is a very effective way of joint noise and echo reduction in an acoustic system.

A computationally more efficient solution is to first apply the MWF and filter out the residual echo in the single output channel. This way we save on the computational power of multiple (M - 1) MS filters. However, these implementations have a worse noise and especially echo reduction performance. The implementation where the MWF only selects the highest SNR subspace, which results in a R_{ss} that is rank 1, achieves a slightly better echo cancellation than the implementation where the MWF selects both the desired and interfering subspace. The next chapter clarifies how high the computational gain is, but it has to be substantial in order to justify the loss in performance.

Better results are achieved by extending the clean far end signal to the MWF input and solving the problem with an extra dimension. This modification increases the amount of noise reduction achieved by the MWF, but the echo suppression is less consistent. The rank 2 implementation especially has a terrible echo cancellation performance, with at least 10dB less SIR gain than the benchmark in every tested scenario. Also, we expect that the computational benefit over the benchmark is lost by increasing the dimension of the MWF.

Both of our integrated solutions have a better noise reduction performance than the benchmark in most of the tested input scenarios. The subspace projection method also has a good echo cancellation performance and only gets outperformed by the benchmark in extremely low iSIR scenarios. However, the computational cost of the projection method is far higher than any of the other solutions. The FE subtraction method has a similar performance, but it lacks stability: in very noisy scenarios the effectivity of echo cancellation seriously falls apart.

Chapter 6

Evaluation by Real-Time Implementation

6.1 Introduction

After getting a first impression of the potential of the different tested algorithms, it is now time to check their performance in real-time implementations.

We first convert the Matlab models of the different noise reduction and echo cancellation techniques into Simulink models. These blocks are used to program our Speedgoat real-time target machine we use in the audio lab to run real-time experiments. After also building the Simulink models for the input/output handling of the signals, together with the building block for the actual processing, the C compiler converts the models into C code, which can run on the target machine.

Since the lab equipment imposes certain limitations on the amount of speakers and microphones we can use, the real-time setup is slightly different compared to previous setups of this thesis. The main limitation is the amount of available I/O on the Speedgoat, which is limited because the other department researchers also have real-time audio setups that they require to be operational. We lowered the amount of microphones to three (instead of six), which is still enough to differentiate between three subspaces: desired signal, interference signal and noise. This also avoids running into computational problems, since the Speedgoat has enough processing power to run each algorithm with these dimensions. We have one loudspeaker for the near end speech, one for the far end speech (instead of two speakers playing the same source file) and one speaker for babble noise.

In order to make the switch from local Simulink simulations to the real-time implementation using the Speedgoat as smooth as possible and make potential troubleshooting more straightforward, we organize the real-time experiments in two steps.

6.2 Phase 1: Offline Experiments with Real Data

First up, we create recorded signals using a playback and record setup and use these signals to test the Simulink models, making sure that the offline simulations also work with actual real-life data. This setup is very straightforward: play all of the relevant sound clips (selected from the Harvard word list database [18]) through the different loudspeakers available in the lab and record the mix of signals with our set of microphones. The playback is done with the lab computer, which has Adobe Audition installed. This software enables us to route the different signals (near end, far end and noise signals) to the audio interface and according to the settings of the sound card, they will get played through specific loudspeakers in the experimental environment.

The audio interface between the computer and the Speedgoat is an RME Fireface UCX sound card. Three of the output channels are reserved for the loudspeakers to play the corresponding speech or noise signal.

During the first stage of our experiments, the lab computer captures and saves the raw microphone signals. These recordings are then used as inputs for the offline Simulink models, in order to check whether the algorithms are still stable with non-synthetic data and to double check for any mistakes in the models themselves. During this phase, the Speedgoat is configured to simply route the different microphone signals into the sound card of the lab computer, where they can be captured and saved using Adobe Audition. The required VAD information is calculated by Simulink blocks based on the clean source files. After running and checking the simulations for any discrepancies, we can now assume that everything is in order with the Simulink models and that real data does not cause instability in the algorithms.

6.3 Phase 2: Real-Time Experiments

We can now move on to the actual real-time implementation. The playback portion of our experimental setup remains the same, but there are some modifications in the recording and processing setup. The real-time setup requires two extra output channels of the RME sound card to be connected to the Speedgoat, in order to supply the real-time algorithms with VAD information or a clean version of the far end signal while it is also playing through one of the loudspeakers. This source of clean audio is required for some of the evaluated solutions. It is assumed that this clean signal is accessible because of the teleconference setup we are using to test the different algorithms, where we receive both audio and video of the correspondent through an internet link.

During an experiment, the lab computer plays different source files through the loudspeakers, whilst the microphones record the mix of these signals. These streams



FIGURE 6.1: Real-time setup with three loudspeakers, one for each subspace. Below we see the microphone array that records the input signals for the algorithms.

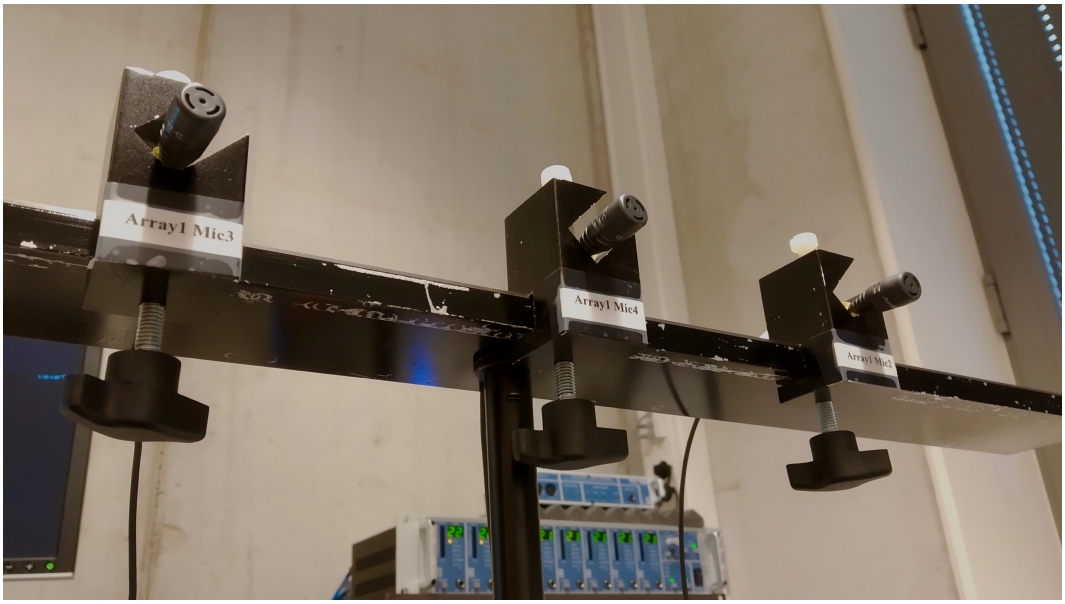


FIGURE 6.2: A close-up of the microphone array, which consists of three units.



Figure 6.3: On the left we see the computer that acts a playback and recording device for the setup. On the right we can see the Speedgoat, a Samson amplifier for the passive loudspeaker and on top of that a pre-amplifier for the microphones.

are fed into the Speedgoat where the compiled Simulink models use the VAD data to calculate the relevant correlation matrices and compute the filter coefficients. Consequently, it is no longer required to have access to all of the raw microphone inputs at the lab computer, since these signals get processed directly by the code loaded onto the Speedgoat. It is however necessary to have access to the output of the algorithms, so we can monitor if the algorithm is operating correctly and afterwards calculate the performance measures. The results of these experiments are captured in a video clip demonstrating the setup and the performance of our algorithms, which is presented during the thesis defence presentation.

6.4 Implementation of Performance Measurements

A slightly different and more efficient experimental setup was used in order to generate the performance figures.

To have the same input statistics (iSNR, iSIR) for the different algorithms, every scenario needs the same impulse responses and loudspeaker volumes. This is difficult to achieve in a real environment. Since the audio lab is shared between multiple students and researchers who all require unique setups for their experiments, it is very unlikely that the impulse responses and mixing volumes of the setup remains exactly the same in between sessions.

One option to achieve this equivalence would be to run the different algorithms in parallel on the Speedgoat and adjust the volume of either the noise or interference source for every scenario, so that every method operates with exactly the same external variables (iSNR,iSIR). However, the computational requirements for running all of our solutions in parallel are too great for the Speedgoat to handle, so this option is not feasible in practice.

An alternative approach is to record the different signals separately using the lab computer and mix them according to the different input scenarios we want to test our algorithms for. In this way we are sure that there are no differences between the input scenarios when comparing the different algorithms.

It is also important to note that since the different correlation matrices needed to calculate the filter coefficients require information from multiple VAD situations, the algorithms do not perform to their full potential immediately after starting up. To have a fair comparison between the different solutions, we allow the algorithms to settle in for a certain amount of time before calculating the performance measures. Empirically, we found that a start-up time of 20 seconds gives the solutions enough data to estimate well-conditioned filter coefficients.

6.4.1 Creation of Suitable Input Scenarios

First, we go back to the phase 1 setup in order to record and save the signals captured by all of our three microphones. Then we play back the different source files separately (desired speech, interference speech and noise) through their respective speaker and record the microphones. We now have nine source files which we will call tracks: for every subspace (in our case desired $S[l]$, interference $I[l]$ and noise $N[l]$) there are three recordings, one for every unit of the microphone array. These tracks are time-domain signals denoted with the superscript *rec* and subscript k, which is the microphone index. They are used to create the different input scenarios, giving a certain power to each of the tracks.

Mixing is done in a Matlab script, which requires an array of iSNR values in decibels, an array of input SIR values in decibel and all of the nine individual recorded tracks. First, we normalize the input tracks so each iSNR and iSIR value for every microphone is 0dB.

$$I_k^{rec}[l] := I_k^{rec}[l] \sqrt{\frac{\text{var}(S_k^{rec}[l])}{\text{var}(I_k^{rec}[l])}} \tag{6.1}$$

$$N_k^{rec}[l] := N_k^{rec}[l] \sqrt{\frac{\text{var}(S_k^{rec}[l])}{\text{var}(N_k^{rec}[l])}} \tag{6.2}$$

with $k = 1...3$ equal to the amount of microphones. Recall that the variance of a signal (denoted by var(.))calculates its power. The first equation equalizes the power

of the interference signal recorded by microphone k to the power of the desired speech signal recorded by the same microphone. Similarly, the second equation equalizes the power of the noise signal to the desired speech signal.

Now, given the required values of iSNR and iSIR together with the normalized signals, we can calculate the mixing coefficients directly based on these required values. We build three track matrices, one for every microphone, with in every column one of the subspaces.

$$\boldsymbol{T}_k[l] = \begin{bmatrix} S_k^{rec}[l] & I_k^{rec}[l] & N_k^{rec}[l] \end{bmatrix} \tag{6.3}$$

with $k = 1...3$ equal to the amount of microphones.

There are three mixing coefficients for every required input scenarios, one for each subspace. In order to sweep the noise source volume while keeping the desired and interference volume fixed, the coefficients of the desired and interference signal are equal to 1 (keep a steady iSIR = 0dB) while the power of the noise changes based on the given iSNR values. Accordingly, when sweeping the interference volume we keep the desired and noise volume steady (iSNR = 0dB) while adjusting the interference power. To calculate the multiplicands, we first convert the decibel values to absolute values:

$$\mathrm{iSNR}_a = 10^{\frac{\mathrm{iSNR_{dB}}}{10}} \tag{6.4}$$

$$\mathrm{iSIR}_a = 10^{\frac{\mathrm{iSIR_{dB}}}{10}} \tag{6.5}$$

To vary the noise power for the iSNR plots, we calculate $\boldsymbol{C}_n$ to mix the separate tracks. $\boldsymbol{C}_n$ mixes the desired speech and the interference equally loud (their coefficients set to 1) and varies the noise power volume based on the current required value of input SNR. Similarly, to create the plots with varying interference volumes, we use $\boldsymbol{C}_i$. These coefficients mix the desired and noise tracks equally loud and vary the power of the interference based on the required input SIR value.

$$\boldsymbol{C}_n = \begin{bmatrix} 1 & 1 & \sqrt{1/\mathrm{iSNR}_a} \end{bmatrix} \tag{6.6}$$

$$\boldsymbol{C}_i = \begin{bmatrix} 1 & \sqrt{1/\mathrm{iSIR}_a} & 1 \end{bmatrix} \tag{6.7}$$

These coefficients are then matrix multiplied with the tracks to mix them:

$$Y_k[l] = \boldsymbol{T}_k[l]\ \boldsymbol{C}_n^T \text{ (while sweeping noise volume)} \tag{6.8}$$

$$Y_k[l] = \boldsymbol{T}_k[l]\ \boldsymbol{C}_i^T \text{ (while sweeping interference volume)} \tag{6.9}$$

These precisely mixed signals serve as inputs to iterate over the different Simulink models in order to create the performance plots.

6.4.2 Calculating Output Measures

Similar to the Matlab simulations, the separate tracks are also used to calculate the SNR, SIR and SINR of the output signals. While the algorithm is running, it produces filter coefficients during every window which are applied to the input signal in the frequency domain. Since we assume that all the signals are additive, it does not matter if the filter coefficients are applied to the individual subspaces of the input, or to the entire mix. By also applying the calculated filters to the individual tracks, we can measure the power of each subspace at the output and get reliable performance measures.

It is important to note that when evaluating an MS solution, only the far end track should be filtered with the mean squares solution. An MS filter executes a subtraction and since the signals are additive, only one subtraction should be made on all of the individual tracks, otherwise we would introduce a filtered interference in the near end and/or noise track.

Perceptual Evaluation of Speech Quality (PESQ)

In practice, this score is more straightforward to calculate because the PESQ measure only requires the processed output instead of the individual subspace signals. Once one of the algorithms is done processing an entire input file, the output is compared to the clean near end signal by a Matlab function [20] which implements the PESQ measure based on the ITU standard P.862 [21]. This function returns a value between -0.5 (processing has highly degraded the speech signal) and 4.5 (no difference between reference speech and processed speech). [22] claims that the PESQ measure is a good indication for subjective quality and comprehensibility of speech signals.

6.5 Results and Discussion

6.5.1 Introduction

Using the setup described in section 6.4 we have generated multiple plots illustrating the performance of the selected algorithms. Section 6.5.2 discusses the noise reduction performances by evaluating the SNR gains of our solutions. Next up, section 6.5.3 discusses the echo cancellation performances by looking at the SIR gains. In section 6.5.4, we cover the measures that illustrate the understandability of the solutions. Finally, section 6.5.5 gives us insight into the computational requirements of the different algorithms.

The test scenarios were generated by mixing individual tracks as discussed in 6.4.1, with a total signal length of 80 seconds and a start-up time before the measurements of 20 seconds.

6.5.2 Noise Reduction Performance

First, we take a look at the noise reduction performance of the selected algorithms. We pay special attention to the low iSNR situations. A high amount of noise reduction makes a bigger perceptual impact on the quality of the processed speech in scenarios where the room is very noisy, compared to situations where the noise is already low enough in volume to sit in the background.

- First, we take a look at our benchmark, the MS+MWF algorithm in black. Similar as in the Matlab simulations, the performance is steady in all scenarios, with especially good results in noisy environments .
- On both graphs we can immediately see two distinct groups forming: the MWF algorithms that utilize rank 2 approximations (in blue) have considerable weaker noise suppression in comparison to the rest of the solutions. This trend returns in the other sections as well, with the trade-off appearing in the PESQ scores, which we discuss in a later section. The seemingly inferior performance is caused by the rank 2 approximation itself. R_{nn} is only updated while both near end and far end speech is inactive, a situation which is relatively rare in any conversation. Less access to relevant data can have an negative impact on the estimation quality of R_{nn} and might lead to a poorly conditioned matrix.
- The standard rank 1 algorithms (in green) have a very similar profile as the MS+MWF solution, with an overall slightly worse performance than the benchmark. The rank 1 MWF+MS has the worst performance, because the MWF has to deal with the far end signal without having access to the source file of this interference through the dimension extension. Forcing the R_{ss} to be rank 1 in this case is a bigger approximation, resulting in a slightly worse noise reduction performance.
- Looking at the integrated solutions (in red), we can observe that these methods in most cases outperform the rest of the algorithms. The MWF with FE subtraction has a slightly better profile than the subspace projection MWF algorithm across-the-board, especially in low iSNR and iSIR. However, it is only a slight difference which is never greater than 1dB of extra gain. However, keep in mind that the computational requirements for the subspace projection method are a lot greater.

6.5.3 Echo Cancellation Performance

Now we evaluate the results of the echo cancellation performance. Similar to previous section, good echo cancellation is important for low iSIR scenarios where the interference overpowers the desired speech considerably. That is why we pay special attention to low SIR data points.

- Overall, the MS+MWF algorithm reigns supreme in the echo suppression category. Especially in low iSIR scenarios, the areas that require high quality

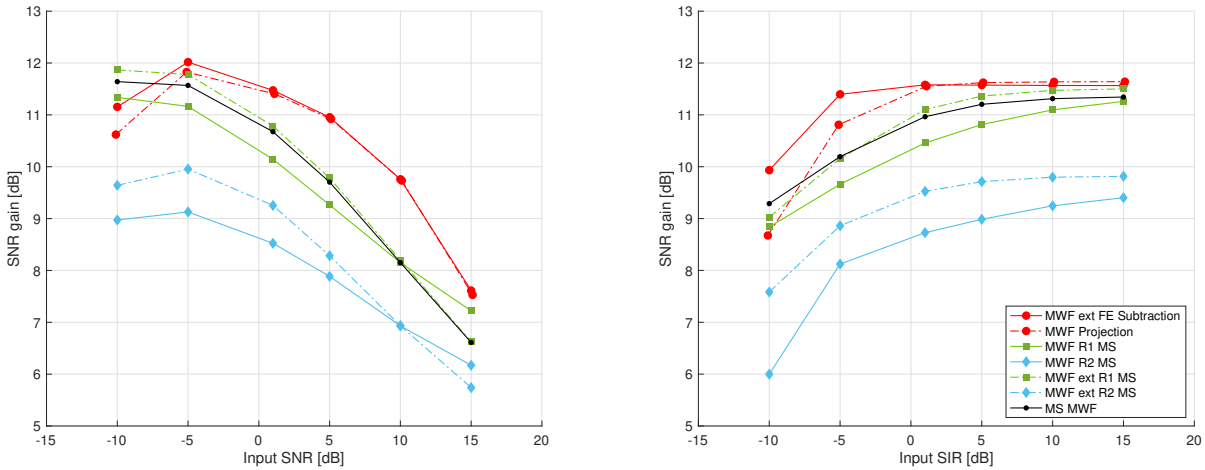


FIGURE 6.4: Noise Reduction Performances of Real-Time Implementations.

echo cancellation the most as we previously discussed, we can see very high gains (up to 17dB). The loud volume of the interference gives the MS filter a very clear input to identify the impulse responses between the source of interference and each microphone. Because of this situation, the filter coefficients are a very good approximation of these impulse responses and the MS filter can work optimally, resulting in a good suppression of the interference. In this case, the input to the MWF is already well conditioned and approaches rank 1, meaning the approximation is more justified. The residual interference is low in power, so it can be seen as just another noise source the MWF is well capable of suppressing. These two factors result in a very effective echo cancellation. Consequently, this solution acts as the benchmark for echo cancellation performance.

- Similar to the noise reduction figures, we see both the rank 1 solutions (in green) and both the rank 2 solutions (in blue) grouping together. Both rank 2 based methods have a clear disadvantage compared to not only the rank 1 algorithms, but all of the tested solutions. The rank 1 MWF component actively helps cancelling out the echo, while the rank 2 implementation leaves the interference alone. It was conceivable that the MS filter could work much more effectively on the intermediate echo signal when the MWF does not tries to suppress it first. However, both the Matlab and real-time experiments show the same result: the echo volume gets reduces more if both the MWF (rank 1) and MS filter work on suppressing the echo.

- The integrated methods work better than the rank 1 based discrete solutions, and seem to be more robust against differences in noise volume than the rank 2 based methods. However, they do not perform up to par against our benchmark in any of the real-time scenarios. The FE subtraction method is a lot more

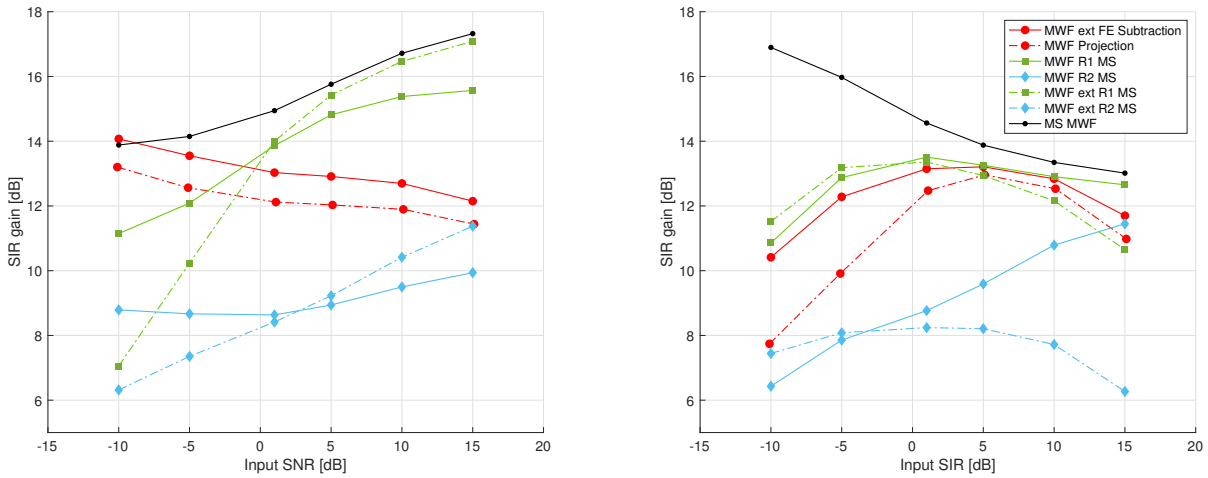


Figure 6.5: Echo Cancellation Performances of Real-Time Implementations.

stable than during the Matlab simulations, its performance in low iSNR and high iSIR scenarios does not fall apart any more. Between the two integrated solutions, the FE subtraction methods outperforms the subspace projection method slightly, similar to the noise reduction results.

6.5.4 Intelligibility Performance

In this section, we compare the understandability of the output signals produced by the different selected algorithms. We have two measures that can provide us with insight about this topic: the PESQ score and the singnal-to-interference-and-noise ratio (SINR) value. The PESQ score is a good indication for the subjective quality and comprehensibility of speech signals, while the SINR value purely compares the relative power of the desired signal to the noise and interference signals.

- First, let us talk about the benchmark in black, the MS+MWF algorithm. It achieves respectable PESQ scores which are once again, quite stable. We see that the PESQ performance is almost exactly the same as the rank 1 methods, a bit worse than the rank 2 methods and a bit better than the integrated solutions. Our benchmark achieves good SINR values, with no other algorithm consistently outperforming it. Especially the great echo cancellation capabilities of the MS+MWF algorithms show in this figure, in the low iSIR area.

- The rank 1 solutions (full and dotted green) have a very respectable and consistent performance, both in relative volumes and PESQ scores. We can see that rank 1 MWF+MS algorithm has very similar results as the benchmark (except for the SINR score in low iSIR scenarios), making it an interesting pick when computational requirements are strict. The extended rank 1 MWF+MS

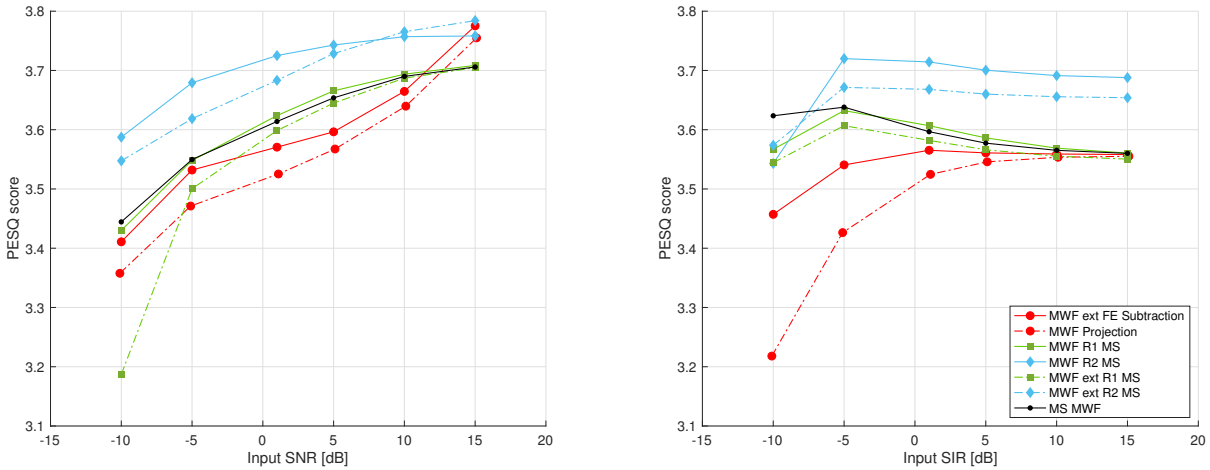


FIGURE 6.6: Intelligibility Results: PESQ scores of Real-Time Implementations.

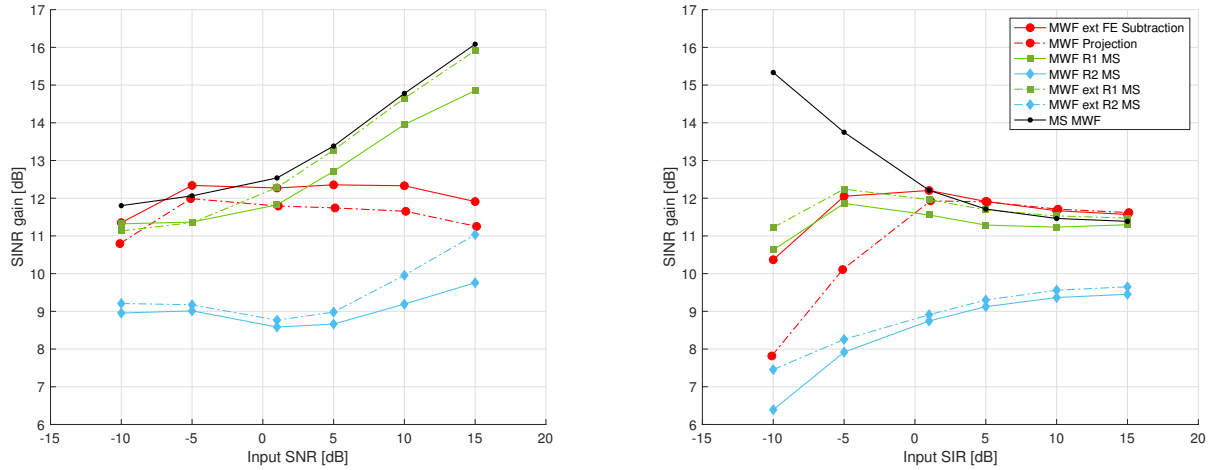


FIGURE 6.7: Intelligibility Results: SINR values of Real-Time Implementations.

algorithm has very similar scores as well, performing slightly better in the SINR plots than its lower dimensional counterpart.

- The rank 2 solutions (full and dotted blue) are probably the most interesting to evaluate, because they perform totally differently depending on which of the two measures we look at. They both achieved great PESQ scores, performing well above the benchmark and the other algorithms. The version without extra FE dimension performs best out of the two. However, when looking at the SINR values, they clearly perform way below average, sitting far below the benchmark and never even crossing into the territory of the other algorithms. This duality between both graphs is characterized by a low reduction in volume (sub-par SINR performance), but a low amount of distortion in the output signal (good PESQ scores). This behaviour is explained by the nature of the rank 2 MWF approximation. When the MWF selects the two subspaces with the highest SNR and setting the rest of the eigenvalues to zero, it does not forcefully throw away all the information about the far end signal. This results in less distortion of the output signal, but with no extra suppression of the echo signal.

- The integrated methods (full and dotted red) have the worst PESQ scores in general. Especially the subspace projection method achieves relatively sub-par scores, specifically in low iSIR areas it fails to perform well. This also becomes clear during the real time experiments: the desired voice in the output sounds heavily filtered in some cases, moreso than with the other algorithms. The output is still usable and understandable though. If the application can cope with the distortion and additional computational requirement, the subspace subtraction method is still a fair option. The FE subtraction method has a more stable performance, staying relatively close to the benchmark. The PESQ scores take a small hit in low iSIR scenarios.

6.5.5 Computational Requirements

In this section, we compare the execution time of the different Simulink implementations in order to gain insight into the computational requirements of the solutions. The average execution times for processing an 80 seconds long array of microphone signals by the Simulink models are displayed in figure 6.8.

- As we expected, the implementation where the MWF filter processes all the microphone signals and an MS filter afterwards gets rid of the residual echo, is computationally more beneficial than our MS+MWF benchmark. It only requires one instead of M MS filters. However, the gain in our setup is quite limited, resulting in a 8% faster execution in case of M = 3 microphones. We expect that this difference gets greater in setups with more microphones.

- The four right-most algorithms all require an extra dimension for the far end signal. This increases the execution time considerably. This is also the only

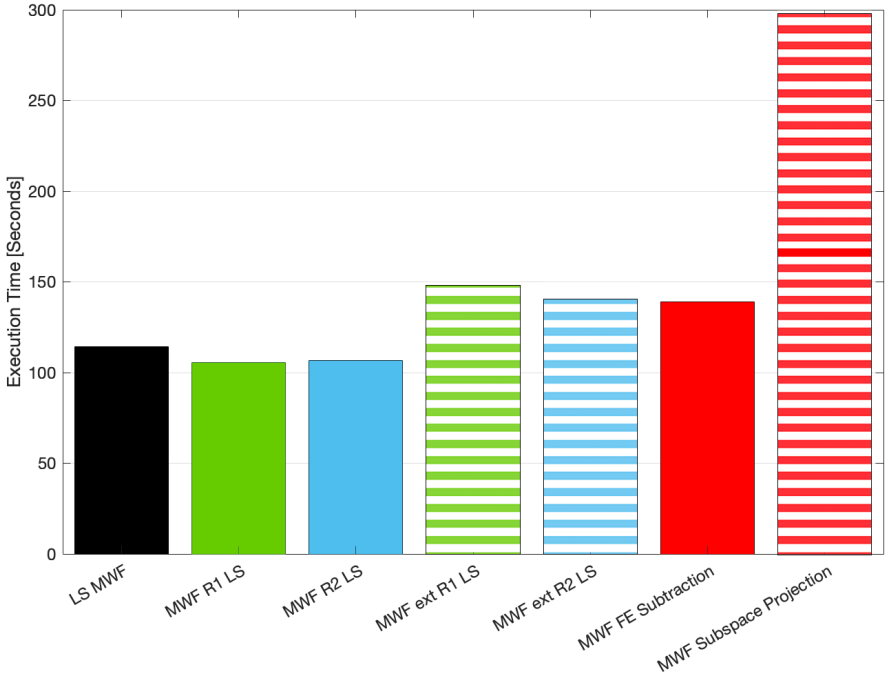


FIGURE 6.8: Execution Times of Real-Time Implementations.

reason why our MWF implementation with FE subtraction is slower than the benchmark. However, it is still a faster solution than applying an entire MS filter afterwards, as we can see by the execution times of the extended MWF MS implementations.

- It is obvious that the subspace projection method requires a substantial amount of extra processing power, because of the two extra GEVD subtractions in every frequency bin. In our tests it requires at least twice the amount of work compared to the others methods.

- The difference between the extended and normal methods is quite considerable, but since these measures are calculated with only three microphones, adding one extra dimension has a significant relative impact on the amount of calculations required. If the setup has more microphones however, the relative increase in execution time gets smaller. The subspace projection does not have this kind of benefit: the algorithm always requires two extra GEVD subtractions, independent of the amount of microphones in the setup.

6.6 Conclusion

Similarly to the Matlab experiments, our MS+MWF benchmark gives us a good starting point for the performance measures. It boasts a relatively stable and competitive noise reduction performance together with unmatched echo cancellation properties. The real-time experiments also shed light on the computational requirements of the different solutions. Once again, our benchmark offers a competitive baseline for this figure: some of the experimental alternatives perform better, some worse.

In the previous chapters it was already clear that it would be computationally interesting to switch the two stages around, so the adaptive echo filter only has to work on the single output channel of the MWF. However, the Matlab simulations suggest that performance suffers quite dramatically under this adjustment and that the computational benefits should be great enough to justify this trade-off. The real-time simulations now show that generally this is not the case. For a setup with three microphones, the decrease in execution time of a fixed length test signal is only 8%, while the performance loss is in most cases greater. Especially the rank 2 implementation performs much worse, with a decrease in SNR gain between 7% and 35% together with a decrease in SIR gain between 15% and 65%. On the flip side: the PESQ scores achieved by the rank 2 implementations are the only ones that outperform the benchmark.

The experiments shed light on this trade-off between volume reduction (as measured by SINR gain) and correlation based intelligibility measures (PESQ scores). We noticed that methods with a high increase in SINR gain achieved quite poor PESQ scores and visa versa. This trade-off becomes especially clear with the rank 2 MWF+MS implementations, since they offer the highest PESQ scores whilst having the lowest gain of relative volume. These findings are more thoroughly explained in section 6.5.4.

The two-stage methods that implement the FE dimension extension have better results than the regular MWF+MS options, but they fail to match the benchmark's performance. They also lose the computational advantage over the benchmark by adding an extra dimension, which leads to bigger correlation matrices and slower GEVD subtractions. In conclusion, there seems to be no reason to use these algorithms over the classical cascade of echo filters and MWF, since both the computational and qualitative characteristics are worse.

Finally, let us discuss the integrated solutions. Between the two investigated methods, there is a clear winner: the method with FE subtraction outperforms the projection method in all of the test scenarios for all of the measures and it boasts a much better computational profile. There seems to be no valid reason to prefer the subspace projection solution over the FE subtraction method, at least when considering the measures we used during our real-time experiments. The FE subtraction method requires more computations than the benchmark, however the increase is quite small and manageable. On top of that, the relative difference grows smaller if we increase

the number of microphones in the setup. It is also the fastest of the solutions that implement the FE dimension extension.

The FE subtraction algorithm has a higher SNR gain than the benchmark in most of the researched scenarios. The echo cancellation performance is stable and better than the other alternative setups, but still gets outperformed by the benchmark, especially in scenarios where the echo is very loud. PESQ scores are similar to the benchmark, with only a notable difference in favour of the benchmark during very noisy input scenarios. The SINR scores are also very stable, better than the alternatives but in some cases worse than the benchmark, again most noticeably in scenarios where the echo is loud.

Chapter 7

Conclusion and Further Research

7.1 Conclusion

In this thesis we investigate microphone signals that are both contaminated by excessive background noise and an interfering audio signal. Multiple methods for reducing the volume of both these noise and echo components are proposed and evaluated experimentally. We tested the performance of each algorithm in multiple test scenarios by changing the relative playback volumes of the sound sources. We pick a collection of performance measures to gain insight in both the volume reduction of unwanted signals as the intelligibility of the output signal.

The experiments show that first applying an echo filter on all the microphone channels in order to reduce the echo volume in each of the channels and then feeding these signals into a rank 1 GEVD-based MWF for noise reduction is a very effective way of joint noise and echo reduction in an acoustic system. Since this is an established technique that returns good performance figures, we use it as a benchmark.

We also test more efficient two-stage solutions, that first apply a MWF to reduce noise and then filter out the residual echo in the single output channel. Since this approach requires only one echo filter, instead of one on every microphone channel, it calls for less computations. However, the experiments show a substantial negative impact on both noise and especially echo reduction performance, an impact that is much greater than the relatively limited computational gain.

In the hope of achieving better echo cancellation performance, we supply the MWF with the clean far end signal by concatenating it to the rest of the microphone signals. The experiments show that the addition does increase the performance, but it still fails to match the effectivity of the benchmark method. It also loses the computational advantage over the benchmark by adding in the extra dimension. We conclude that there seems to be no use case for this particular implementation, since

both the performance and efficiency are worse than the benchmark.

Finally, let us discuss the actual subject matter of this thesis: the integrated solutions which only require one stage of processing. We propose two experimental methods, the subspace projection method and the FE subtraction method. The projection method uses the well-defined joint subspace as a way to improve the quality of the separate speech-only and echo-only subspaces, using a projection method. All this information is then used together to provide the best possible estimate of the desired speech subspace. The FE subtraction method uses a built-in echo filter, which can extract the echo information during the MWF process and subtract it immediately.

The real-time experiments show that the subtraction method outperforms the subspace projection solution for all of the reviewed measures, while also requiring much less computations. After looking at these results, there seems to be little incentive to use the subspace projection method. However, the FE subtraction method performs well and in some cases even better than the benchmark. It achieves higher noise reduction measures but slightly worse echo cancellation results. Also, it is computationally more demanding than the benchmark, however the difference is not very drastic. Further research should compare this method against the benchmark more thoroughly considering different measures and other scenarios.

7.2 Further Research

7.2.1 Better Implementation of Subspace Projection MWF Algorithm

In our implementation of the Subspace Projection algorithm, we have to solve following least squares problem in order to get a good approximation of $R_{\boldsymbol{ss}}$:

$$\begin{bmatrix} P_s \\ P_i \end{bmatrix} = \begin{bmatrix} Q_{\boldsymbol{ss}}^{proj*} \otimes Q_{\boldsymbol{ss}}^{proj} & Q_{\boldsymbol{ii}}^{proj*} \otimes Q_{\boldsymbol{ii}}^{proj} \end{bmatrix} \setminus \operatorname{vec}\left(\tilde{R}_{\boldsymbol{ss+ii}}\right) \tag{7.1}$$

We use the P_s value in order to calculate the approximation of $R_{\boldsymbol{ss}}$ that is used in the MWF filter design:

$$\tilde{R}_{\boldsymbol{ss}} = Q_{\boldsymbol{ss}}^{proj}\ P_s\ (Q_{\boldsymbol{ss}}^{proj})^H \tag{7.2}$$

However, we do not use the P_i coefficient at all. It might be possible to use this high quality value of P_i and consequently $\tilde{R}_{\boldsymbol{ii}}$ in order to increase the performance and stability of the algorithm, without many additional calculations. An option could be to consider this information to increase the quality of the initial rough approximation of $\tilde{Q}_{\boldsymbol{ii}}$ that is retrieved by the rank 1 GEVD subtraction, for example by applying a weighted average between $\tilde{Q}_{\boldsymbol{ii}}^{sub}$ and the eigenvector present in the previous high quality estimation of $\tilde{R}_{\boldsymbol{ii}}$.

7.2.2 Speech Intelligibility Weighted Measures

In this thesis, we evaluated and compared the performances of different signal processing algorithms. In order to make qualitative comparisons between the selected solutions, we used a variety of performance measures. Each measure reflects the competence of an algorithm in a separate way. Since the possible solutions have multiple goals to achieve, each aspect should be tested and measured. We used SNR, SIR and SINR values to measure the reduction in volume of both the noise and echo components of the recorded signals, together with PESQ values to quantify distortion and intelligibility.

This can be done in a more precise way. Since all the audio signals present in this thesis are either clean or diffused speech signals, these measures could be geared towards the perceived volume of speech signals, instead of total power. This can be achieved by different, more specialized performance measures. A possible adjustment could be considering the relative importance of the different frequency bands of speech signals [23]. These ratios should be set with respect to the speech intelligibility index (SII) standard [24], in order to reflect the importance of each frequency bin for speech signals. This way, the SI-weighted measures (SNR, SIR, SINR) are more suitable to measure the perceived effectiveness of both noise and echo suppression.

7.2.3 Multiple Desired or Interfering Sources

In our experiments, we only discussed setups with one desired source file (S_p^{src} with $p = 1$) and one interference source (I_q^{src} with $q = 1$). The algorithms we have discussed can be used in more general and complex cases of multiple desired or interfering source files. For example, if two interference sources are active, the classical two-stage setup would require an additional echo filter to deal with the second echo. Our integrated subspace projection method would not change apart from the dimensions of the matrices, possibly making it useful again in these more specific cases. The FE subtraction method should also easily scale to multiple desired sources. These sophisticated setups are a point of further research.

7.2.4 Speech Presence Probability VAD

The experiments used a binary VAD based on the clean source files of both desired and interfering speech. This was done because the VAD itself is not the focus of this thesis. In real-life systems however, there is no access to the clean near end speech signal. The NE VAD has to be based on microphone signals and since these recordings are contaminated by noise and echo signals, this is not an easy task. This issue leads to an entire other field of research. A solution to this problem is to introduce Speech Presence Probability, which calculates the chance that there is speech present during a certain window [25] [26], instead of using a binary VAD that simply implements a threshold on the signal power.

Bibliography

[1] S. Vorobyov and A. Cichocki, "Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis," *Biological cybernetics*, vol. 86, pp. 293–303, 05 2002.

[2] S. Satheesh and K. V. S. V. R. Prasad, "Medical image denoising using adaptive threshold based on contourlet transform," *CoRR*, vol. abs/1103.4907, 2011.

[3] K. Chung, "Challenges and recent developments in hearing aids: Part I. Speech understanding in noise, microphone technologies and noise reduction algorithms," *Trends in Amplification*, vol. 8, no. 3, pp. 83–124, 2004.

[4] S. Sabri and B. Prasada, "Video conferencing systems," *Proceedings of the IEEE*, vol. 73, pp. 671–688, April 1985.

[5] Zhengyou Zhang, Zicheng Liu, M. Sinclair, A. Acero, Li Deng, J. Droppo, Xuedong Huang, and Yanli Zheng, "Multi-sensory microphones for robust speech detection, enhancement and recognition," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. iii–781, May 2004.

[6] J. H. L. Hansen and M. A. Clements, "Constrained iterative speech enhancement with application to speech recognition," *IEEE Transactions on Signal Processing*, vol. 39, pp. 795–805, April 1991.

[7] A. Geusens, A. Bertrand, B. Cornelis, and M. Moonen, "Multi-channel noise reduction in hearing aids with wireless access to an external reference signal," in *Proc. of the International Workshop on Acoustic Signal Enhancement (IWAENC)*, (Aachen, Germany), Sep. 2011.

[8] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, pp. 235–238, June 1977.

[9] E. Sejdic, I. Djurovic, and J. Jiang, "Time-frequency feature representation using energy concentration: An overview of recent advances," *Digital Signal Processing*, vol. 19, no. 1, pp. 153 – 183, 2009.

[10] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.

[11] M. S. Brandstein and D. B. Ward, "Microphone arrays: Signal processing techniques and applications," pp. 41–57, 01 2001.

[12] B. Cornelis, M. Moonen, and J. Wouters, "Binaural voice activity detection for MWF-based noise reduction in binaural hearing aids," in *2011 19th European Signal Processing Conference*, pp. 486–490, Aug 2011.

[13] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Low-rank approximation based multichannel wiener filter algorithms for noise reduction with application in cochlear implants," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 785–799, April 2014.

[14] A. Hassani, A. Bertrand, and M. Moonen, "Distributed GEVD-based signal subspace estimation in a fully-connected wireless sensor network," in *Proc. European Signal Processing Conference (EUSIPCO)*, (Lisbon, Portugal), Sept. 2014.

[15] L. Trefethen and D. Bau, *Numerical Linear Algebra*, pp. 234–240. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 1997.

[16] A. Hassani, A. Bertrand, and M. Moonen, "LCMV beamforming with subspace projection for multi-speaker speech enhancement," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 91–95, March 2016.

[17] G. J. Simpson, *Identity Relations for Vectorization of Matrices and Tensors*, pp. 475–482. Cambridge Molecular Science, Cambridge University Press, 2017.

[18] "IEEE recommended practice for speech quality measurements," *IEEE Transactions on Audio and Electroacoustics*, vol. 17, pp. 225–246, Sep. 1969.

[19] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," in *Journal Acoustic Society of America*, p. 943, April 1979.

[20] Y. Hu and P. C. Loizou, "Speech enhancement: Theory and practice," (Boca Raton, FL, USA), 2007.

[21] I.-T. P.862, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Feb. 2001.

[22] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 229–238, Jan 2008.

[23] J. E. Greenberg, P. M. Peterson, and P. M. Zurek, "Intelligibility-weighted measures of speech-to-interference ratio and speech system performance," *The Journal of the Acoustical Society of America*, vol. 94, no. 5, pp. 3009–3010, 1993.

[24] A. S. of America. Standards Secretariat, A. N. S. Institute, A. S. of America, and B. Accredited Standards Committee S3, *American National Standard: Methods for Calculation of the Speech Intelligibility Index*. Acoustical Society of America, 1997.

[25] M. Souden, J. Chen, J. Benesty, and S. Affes, "Gaussian model-based multichannel speech presence probability," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1072–1077, 2009.

[26] T. Gerkmann, M. Krawczyk, and R. Martin, "Speech presence probability estimation based on temporal cepstrum smoothing," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4254–4257, March 2010.

IEEE paper

Joint Multi-Channel Noise Reduction and Echo Cancellation

Real-Time Implementation and Evaluation

Sven Verret

***Abstract*—In the field of signal processing, we are often confronted with contaminated microphone signals whenever we want to record a speech signal in a busy room. These signals are often affected by background noise and by disturbing audio signals generated by unwanted human speech or loudspeakers. If we also have access to the clean source signal of these unwanted audio signals, we call these audio signals echo signals. We want to increase the quality of our recordings by reducing the volume of both the noise and echo components, without excessively distorting or reducing the desired speech signal. In this paper, we discuss and test methods that are capable of achieving both noise reduction and echo cancellation in one integrated stage of signal processing. We compare the performances and computational requirements of these algorithms against classical two-stage solutions that serve as benchmarks. Then we will draw a conclusion based on both Matlab and real-time experiments whether it is compelling to use these integrated alternatives for scenarios similar to our research setup.**

***Index Terms*—noise reduction, echo cancellation, multi-channel Wiener filter (MWF), real-time, implementation.**

I. Introduction

THIS paper investigates how to clean up microphone signals that are both contaminated by excessive background noise and an interfering audio signal. In order to tackle our research goal, we start off by constructing an appropriate scenario in which we can formalize the problem statement, propose different solutions and compare the performances of these solutions. The scenario that we picked, coined the 'Robust conference call' setup, is based on the setup discussed in [1]. This setup is very suitable for our research since it has multiple noise sources and an audio signal with the source signal available, which acts as our echo signal.

An important field of applications where joint noise- and echo cancellation is needed are medical applications [2] [3], especially the development of hearing aids is an important point for speech enhancement [4]. People who suffer from hearing loss require a greater separation in volume between the actual speech of the person that they are conversing with and any other interfering noise in their environment. The same problem arises when looking at our smartphones, which are also equipped with microphones and speakers for phone calls, be it hands free or not. Your smartphone can be converted into a conference system by one tap, immediately requiring an algorithm for noise reduction and echo cancellation.

Speech enhancement can be very useful as pre-processing for speech recognition systems [5] [6]. The higher the quality of the processed recording, the less likely is it for the speech recognition system to make a wrong decision. These two systems together can then be used as a voice interface with different kinds of devices, like personal computers and gaming consoles.

There are multiple possible solutions for this issue, but the most straight-forward is a two-stage process, consisting of two cascading filters, where one filter reduces the ambient background noise and a second filter cancels out the interfering echo signal. The noise reduction step is implemented through different variants of the Multi-channel Wiener Filter (MWF). A Mean Squares (MS) echo filter takes care of the echo signal.

Next to the well established two-step approaches, it is also possible to perform both noise reduction and echo cancellation with one single filter. These are the so-called joint noise reduction and echo cancellation systems and are the main subject of research of this paper.

This paper first discusses the signal model needed to explain the possible solutions to this problem in section II. In section III, we present the audio signal processing techniques that are relevant for the experimental setups. After that, in section IV, we discuss the results from the experiments by comparing the performances of the different proposed algorithms. Based on these results, we draw conclusions in section V.

II. Acoustic Signal Model

We use multiple microphones to record and retrieve the input signals for our algorithms. When we record the signals with M microphones, we can either consider the array of signals $\boldsymbol{Y}[l]$ or focus on one microphone channel $Y_k[l]$.

The desired speech signal of which we want to boost its clarity is called the Near End (NE) or desired signal. We refer to the NE signal as $S^{src}[l]$, where l denotes the time domain index and the superscript src indicates that it is a clean source file. The model contains an unwanted acoustic source of which the processing unit has access to the clean playback signal. We refer to this signal as the Far End (FE) signal, the interference or echo signal denoted by $I^{src}[l]$. Finally, we refer to noise signals as $N_r^{src}[l]$, where r selects one of the noise sources, since we assume multiple sources are present. In practice, we consider two possible types of noise signals: namely Additive white Gaussian noise (AWGN) and Babble noise.

The recorded signals can contain transformed versions of all of the above mentioned components in an additive manner.

We formalize this for channel k as:

$$Y_k[l] = S_k[l] + I_k[l] + N_k[l] \quad (1)$$

$$= \boldsymbol{a}_k^S * S^{src} + \boldsymbol{a}_k^I * I^{src} + \sum_r \left(\boldsymbol{a}_{r,k}^N * N_r^{src}\right) + N_{other,k}[l] \quad (2)$$

where $\boldsymbol{a}$ contains the individual transfer function for a sound source to microphone k, which convolutes with the corresponding source signals. $S_k[l]$, $I_k[l]$ en $N_k[l]$ denote respectively the desired speech component, interfering speech component and noise component in microphone channel k. $N_{other,k}[l]$ contains other relevant sources of noise, e.g. sensor noise, quantization noise and diffused noise.

Since we exclusively discuss signal processing techniques in the frequency domain, we have to convert this signal model to the frequency domain. The model has to consider real-time applications, so we use a windowed short-time Fourier transform (STFT) [7] in order to continuously extract frequency and phase information of local sections of the signals. When we talk about the array of all microphone signals in the frequency domain, we use $\boldsymbol{y}[m, w]$, where m is the frequency bin index and w is the window shift. Otherwise, if we focus on one channel, we use $y_k[m, w]$. Since we use this notation frequently, we omit the indices and write $\boldsymbol{y}$ or y_k instead. In the discrete frequency domain, formula (1) and (2) turn into:

$$y_k = s_k + i_k + n_k \quad (3)$$

$$= \boldsymbol{a}_k^s \; s^{src} + \boldsymbol{a}_k^i \; i^{src} + \sum_r \left(\boldsymbol{a}_{r,k}^n \; n_r^{src}\right) + n_{other,k} \quad (4)$$

where s^{src}, i^{src}, n_r^{src} are the STFT transformations of $S^{src}[l]$, $I^{src}[l]$ and $N_r^{src}[l]$, with $\boldsymbol{a}_k^s$ $\boldsymbol{a}_k^i$ and $\boldsymbol{a}_{r,k}^n$ as their respective steering vectors.

The techniques that we discuss in the next section require multiple correlation matrices. These are the correlation matrices of following signals:

$R_{\boldsymbol{yy}} = E\{\boldsymbol{yy}\}^H$	(the observed input signals)
$R_{\boldsymbol{ss}} = E\{\boldsymbol{ss}\}^H$	(the desired signal components)
$R_{\boldsymbol{nn}} = E\{\boldsymbol{nn}\}^H$	(the noise signal components)
$R_{ii} = E\{i_{src} i_{src}\}$	(the clean interference source signal)
$R_{i_{src} i_k} = E\{i_{src} i_k\}$	(the clean and recorded interference)

III. Techniques and Experimental Setups

A. *MWF-based Noise Reduction*

The algorithm that we use for noise reduction is called Multi-channel Wiener filtering (MWF). Multi-channel means that the algorithm requires multiple channels of audio signals. The input to this filter is the array of M signals $\boldsymbol{y}$.

The goal of MWF is to try and recover the useful desired signal from the recorded signals that are corrupted by noise. This is done by filtering and combining the $\boldsymbol{y}$ signals with the filter $\boldsymbol{w}$ in order to minimize the mean squared difference between the filtered output $\boldsymbol{w}^H \boldsymbol{y}$ and the desired speech component s_{ref} in the reference microphone. Formally, this is a Linear Minimum Mean Square Error (LMMSE) estimator given by:

$$\boldsymbol{w}_{opt} = \arg\min_{\boldsymbol{w}} E\left\{\left|s_{ref} - \boldsymbol{w}^H \boldsymbol{y}\right|^2\right\} \quad (5)$$

$E\{.\}$ denotes the expected value operator, $\{.\}^H$ denotes the complex conjugate transpose and $|.|$ denotes the absolute value.

This estimator has an optimal solution, called the Wiener filter coefficients [8,9]:

$$\boldsymbol{w}_{opt} = R_{\boldsymbol{yy}}^{-1} R_{\boldsymbol{ss}} \boldsymbol{e}_{ref} \quad (6)$$

where $\mathbf{e}_{ref}$ is a vector which selects the reference microphone.

In practice, we cannot calculate the true expected value of these correlation matrices. Instead we estimate them by signal averaging over the different STFT windows by using a moving average (a tilde above the matrix means approximation). To estimate whether the desired speech is active and we should update $\tilde{R}_{\boldsymbol{yy}}$ or $\tilde{R}_{\boldsymbol{nn}}$, we use a voice activity detector (VAD) [10]. The interference signal is considered just another noise source, so its information is captured in both $\tilde{R}_{\boldsymbol{yy}}$ and $\tilde{R}_{\boldsymbol{nn}}$. To calculate $R_{\boldsymbol{ss}}$, we have to subtract the noise information from the input signals.

$$\tilde{R}_{\boldsymbol{ss}} = \tilde{R}_{\boldsymbol{yy}} - \tilde{R}_{\boldsymbol{nn}} \quad (7)$$

However, this is not a robust subtraction for several reasons: we work with estimations of the expected values, the noise is not uncorrelated and echo is captured in both matrices. This can lead to oversubtraction resulting in a useless $\tilde{R}_{\boldsymbol{ss}}$ matrix that is not positive semidefinite.

It has been demonstrated [11] that incorporating a low rank approximation based on the generalized eigenvalue decomposition (GEVD) of $\tilde{R}_{\boldsymbol{yy}}$ and $\tilde{R}_{\boldsymbol{nn}}$ enhances the estimation performance of the MWF, since it solves the oversubtraction issue. The GEVD of matrix pencil $\left\{R_{\boldsymbol{yy}}, R_{\boldsymbol{nn}}\right\}$ results in:

$$R_{\boldsymbol{yy}} = Q \Sigma_{\boldsymbol{y}} Q^H \quad (8)$$

$$R_{\boldsymbol{nn}} = Q \Sigma_{\boldsymbol{n}} Q^H \quad (9)$$

$$R_{\boldsymbol{ss}} = Q(\Sigma_{\boldsymbol{y}} - \Sigma_{\boldsymbol{n}}) Q^H \quad (10)$$

where Q is an invertible matrix, the columns of which are normalized and define the generalized eigenvectors. $\Sigma_{\boldsymbol{y}}$, $\Sigma_{\boldsymbol{n}}$ are real-valued diagonal matrices with eigenvalues:

$$\Sigma_{\boldsymbol{y}} = \text{diag}\left\{\sigma_{\boldsymbol{y}}^1 \; ... \; \sigma_{\boldsymbol{y}}^M\right\} \quad (11)$$

$$\Sigma_{\boldsymbol{n}} = \text{diag}\left\{\sigma_{\boldsymbol{n}}^1 \; ... \; \sigma_{\boldsymbol{n}}^M\right\} \quad (12)$$

In some of the setups, the MWF should only keep the single desired speech signal intact. This is done by a rank 1 approximation, which means that we only consider the eigenvalue with the highest ratio of $\sigma_{\boldsymbol{y}}/\sigma_{\boldsymbol{n}}$ and its corresponding eigenvector. This selects the subspace with the highest SNR of $\tilde{R}_{\boldsymbol{ss}}$ based on the VAD information, which is assumed to select the desired speech signal. The other eigenvalues are set to zero.

$$\tilde{R}_{\boldsymbol{ss}} = \begin{bmatrix}\boldsymbol{q}_a & \mathbf{0}\end{bmatrix} \begin{bmatrix}\sigma_{\boldsymbol{y}}^a - \sigma_{\boldsymbol{n}}^a & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix} \begin{bmatrix}\boldsymbol{q}_a^H \\ \mathbf{0}\end{bmatrix} \quad (13)$$

where index a is the mode with the highest SNR ratio together with its corresponding eigenvector $\boldsymbol{q}_a$. $\mathbf{0}$ denotes null vectors or matrices of the correct dimension.

In other setups, we want to keep both the desired and interfering subspace intact and only reduce actual background noise. This is done by a rank 2 approximation after the GEVD subtraction. We assume that these two modes contain both the NE and FE subspaces, based on the relative input volumes and VAD information of both sources.

$$\tilde{R}_{\boldsymbol{ss}+\boldsymbol{ii}} = \begin{bmatrix} \boldsymbol{q}_a^H \\ \boldsymbol{q}_b^H \\ \mathbf{0} \end{bmatrix}^H \begin{bmatrix} \sigma_{\boldsymbol{y}}^a - \sigma_{\boldsymbol{n}}^a & 0 & \mathbf{0} \\ 0 & \sigma_{\boldsymbol{y}}^b - \sigma_{\boldsymbol{n}}^b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_a^H \\ \boldsymbol{q}_b^H \\ \mathbf{0} \end{bmatrix} \quad (14)$$

where index a is the mode with the highest eigenvalue SNR ratio together with its corresponding eigenvector $\boldsymbol{q}_a$ and b selects the second highest SNR ratio with its eigenvector $\boldsymbol{q}_b$.

B. MS-based Echo Cancellation

An established way of applying echo cancellation is an acoustic Mean Squares (MS) filter. To cancel out the undesired echo present in a microphone signal, the filter looks for an approximation of the transfer function (TF) between the interference source and the microphone. It filters the source signal with this approximated TF and then subtracts it from the microphone signal.

$$w_k = \arg\min_{w_k} E\left\{ \left| i_k - w_k^H i_{src} \right|^2 \right\} \quad (15)$$

where i_k contains the interference frequency data for one frequency bin during a single STFT window, recorded by the k-th microphone. Since we formalized this problem in a similar manner as the MWF problem, it has a similar optimal solution:

$$w_k = R_{ii}^{-1} R_{i_{src} i_k} \quad (16)$$

$R_{ii} = E\{i_{src} i_{src}\}$ is the energy of the interference source signal i_{src}.

These correlation matrices are only updated when the NE speech is inactive, indicated by the VAD, to prevent desired speech cancellation in double talk scenarios.

After the MS filter is applied, every output channel now contains:

$$y_k = s_k + n_k + i_k - w_k^H i_{src} \quad (17)$$

These signals are now an echo-reduced version of the recorded microphone signals. However, they still require noise reduction in order to get rid of the n_k component, which can be done before or after the MS filter. We test both options to compare which is more efficient and effective.

C. Two-step Combined Noise and Echo Suppression

The most straight-forward way to create a solution that performs both noise reduction and echo cancellation, is to combine an algorithm that reduces the noise with another block that suppresses the echo. Now follows a summary of the different two-step solutions we consider and compare.

The first solution (MS+MWF) boils down to first cancelling out the echo signal in each channel. Then, the channels are concatenated and fed to a MWF in order to suppress background noise. We expect this setup to have a good overall performance, since the MS filter has direct access to the (potentially loud) interference component of the microphone signals in order to determine the transfer function between source and microphone. After this, the MWF has relatively clean signals to work with and the $R_{\boldsymbol{yy}}$ should approach rank 1, leading to a good noise reduction performance.

The second setup (MWF+MS) tests if changing the order of both components has a big impact on the performance of the algorithm. It is computationally more beneficial since it only requires one MS filter instead of one for every microphone channel. We also compare two versions of the MWF algorithm: the rank 1 and rank 2 approximation. The rank 1 MWF considers the echo as noise and also suppresses it before feeding the output into the MS filter. The lower volume could make it more difficult for the MS filter to determine the transfer functions. In addition, there is also a possibility that the rank 1 approximation selects the wrong subspace. The experiments will clear up if the accumulation of echo suppression by both components is effective.

The last two-step setup we discuss is the MWF+MS with extension (MWF_ext+MS). We have access to the clean interference signal everywhere, so we might be able to use this information to develop better MWF coefficients, which are also well suited to suppress the echo. We give the filter access to the source signal of the echo by adding it to the input set of the MWF. The input to our extended MWF is:

$$\boldsymbol{y} = \begin{bmatrix} y_1 & \dots & y_M & i_{src} \end{bmatrix}^T \quad (18)$$

where $\boldsymbol{y}$ is the input array, $y_1 \ \dots \ y_M$ are all the microphone signals in frequency domain and i_{src} is the far end signal. The GEVD subtraction consequently has an increased size, which hinders performance significantly since its mathematical complexity scales with O(mn^2) for an $m \times n$ matrix [12]. Once again, we test both the rank 1 and rank 2 approaches.

D. Integrated Noise and Echo Suppression

Now we discuss the methods that require only one stage of processing. There are some advantages that these integrated solutions have over the previous two-stage solutions. These algorithms should provide a more robust echo cancellation method than the implicit suppression a rank 1 approximation supplies, which in case of loud interference leaves open the possibility of selecting a non-desired subspace after the GEVD subtraction. They also use all the information contained in the clean echo signal, as opposed to an MS filter, which only updates its echo information when the NE is inactive.

*1) **Far End Signal Subtraction MWF**:* The FE subtraction method uses a built-in echo filter, which can extract the echo information during the MWF process and subtract it immediately. There is no need for a separate step that tries to estimate the transfer function between the echo source and the microphone and then removes the filtered signal from the

mix. If we apply the far end extension as explained previously, this direct subtraction is possible.

As usual, our MWF collects the microphone information and builds R_{yy} and R_{nn} based on the information of both NE and FE VAD signals. R_{nn} contains exclusively noise information since we update it only when both NE and FE are inactive. First, we apply a rank 2 GEVD subtraction of the noise, which results in the rank 2 $\tilde{R}_{ss+ii}$ matrix. The last column of $\tilde{R}_{ss+ii}$ always contains the coefficients associated with the clean far end signal.

$$\tilde{R}_{ss+ii} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1(m+1)} \\ r_{21} & r_{22} & \cdots & r_{2(m+1)} \\ \vdots & \vdots & \ddots & \\ r_{(m+1)1} & r_{(m+1)2} & & r_{(m+1)(m+1)} \end{bmatrix} \quad (19)$$

We extract the far end component in the last column of $\tilde{R}_{ss+ii}$ and normalize it by $r_{(m+1)(m+1)}$, which contains the power of the far end subspace:

$$\boldsymbol{a}_i = \begin{bmatrix} \frac{r_{1(m+1)}}{\sqrt{r_{(m+1)(m+1)}}} & \cdots & \frac{r_{(m+1)(m+1)}}{\sqrt{r_{(m+1)(m+1)}}} \end{bmatrix}^T \quad (20)$$

To enable the FE subtraction, this basis is then transformed into a correlation matrix compatible with $\tilde{R}_{ss+ii}$:

$$\tilde{R}_{ii} = \boldsymbol{a}_i \ (\boldsymbol{a}_i)^H \quad (21)$$

The FE correlation matrix is then subtracted from the joint subspace, which yields an approximated NE correlation matrix:

$$\tilde{R}_{ss} = \tilde{R}_{ss+ii} - \tilde{R}_{ii} \quad (22)$$

The NE matrix is then used to calculate the MWF filter coefficients with the Wiener formula (6).

*2) **Subspace Projection MWF**:* Another possible way to use all the available data (i.e. the complete microphone signals and all VAD information) in a fully integrated solution is based on a Subspace Projection technique [13]. It describes an effective method to estimate the R_{ss} and R_{ii} subspaces.

It starts from R_{yy}, which contains abundant data about all the different subspaces. We subtract the noise information R_{nn} through a rank 2 GEVD subtraction, which results in $\tilde{R}_{ss+ii}$, the approximated correlation matrix of the NE and FE subspaces, and $\tilde{Q}_{ss+ii}$, the eigenvectors which make up the basis for our NE and FE subspaces. These estimations are reliable because they are based on the well-defined and abundant input data collected in R_{yy}.

$$\left[\tilde{R}_{ss+ii}, \tilde{Q}_{ss+ii}\right] = \text{GEVD}_{r2}\left(R_{yy}, R_{nn}\right) \quad (23)$$

We then estimate the Q_{ss} and Q_{ii} eigenvector matrices based upon the incoming signals during desired-only and interference-once periods. Since there are only a limited amount of these specific segments available, the estimations of these individual subspaces are quite inaccurate.

$$\tilde{Q}_{ss} = \text{GEVD}_{r1}\left(R_{ss+nn}, R_{nn}\right) \quad (24)$$

$$\tilde{Q}_{ii} = \text{GEVD}_{r1}\left(R_{ii+nn}, R_{nn}\right) \quad (25)$$

A GEVD subtraction returns a full $M \times M$ matrix Q, but since there are only the same number of relevant eigenvectors as there are relevant subspaces contained in the matrix, we omit the rest of the eigenvectors. We annotate these submatrices by the superscript sub.

We can improve the quality of Q_{ss}^{sub} and Q_{ii}^{sub}. In theory, the columns of Q_{ss+ii}^{sub} and $\left[Q_{ss}^{sub}, Q_{ii}^{sub}\right]$ span the same subspace, however because we are working with estimations based on limited segments of data, this is not valid any more. This discrepancy is corrected by projecting the poorly estimated Q_{ss}^{sub} and Q_{ii}^{sub} onto the well defined joint subspace Q_{ss+ii}^{sub}. The projection is defined by following expressions, where J is defined as the projection matrix:

$$J \triangleq Q_{ss+ii}^{sub}\left((Q_{ss+ii}^{sub})^H \ Q_{ss+ii}^{sub}\right)^{-1}(Q_{ss+ii}^{sub})^H \quad (26)$$

$$Q_{ss}^{proj} = J \ Q_{ss}^{sub} \quad (27)$$

$$Q_{ii}^{proj} = J \ Q_{ii}^{sub} \quad (28)$$

To find the actual required value of R_{ss}, we still require corresponding coefficients for our basis vectors. The resulting coefficients alongside the calculated basis vectors should approximate the joint correlation matrix as well as possible. We can express this task as a minimization problem:

$$\min_{P_s,P_i} \left\| \begin{bmatrix} Q_{ss}^{proj} \\ Q_{ii}^{proj} \end{bmatrix}^T \begin{bmatrix} P_s & 0 \\ 0 & P_i \end{bmatrix} \begin{bmatrix} Q_{ss}^{proj} \\ Q_{ii}^{proj} \end{bmatrix}^* - \tilde{R}_{ss+ii} \right\|_F^2 \quad (29)$$

where $\|.\|_F$ is the Frobenius norm and is defined as the square root of the sum of the absolute squares of its elements.

This equation can be worked out and rewritten as a normal least-squares problem, which we can give to a Matlab or Simulink numerical solver in order to find P_s and P_i:

$$Z \triangleq \begin{bmatrix} Q_{ss}^{proj\,*} \otimes Q_{ss}^{proj} & Q_{ii}^{proj\,*} \otimes Q_{ii}^{proj} \end{bmatrix} \quad (30)$$

$$\begin{bmatrix} P_s \\ P_i \end{bmatrix} = (Z^H Z)^{-1} \ Z^H \ \text{vec}\left(\tilde{R}_{ss+ii}\right) \quad (31)$$

where $\otimes$ calculates the Kronecker product, vec(.) indicates the vectorization of a matrix and $.^*$ is complex conjugate.

The resulting P_s coefficient combined with the basis vector yields our new approximation for R_{ss}:

$$\tilde{R}_{ss} = Q_{ss}^{proj} \ P_s \ (Q_{ss}^{proj})^H \quad (32)$$

The $\tilde{R}_{ss}$ matrix is then used to calculate the MWF coefficients with formula (6).

IV. Results and Discussion

We test the performance of each algorithm in multiple test scenarios by changing the relative playback volumes of the sound sources. We pick a collection of performance measures to gain insight in both the volume reduction of unwanted signals and the intelligibility of the output signal. The performance measure that we use to evaluate the noise reduction is the signal-to-noise ratio (SNR). For echo cancellation performance we look at the signal-to-interference ratio (SIR). To compare the intelligibility of the different results, we look at the relative volume of the desired speech compared to all the unwanted signals and at a subjective perceptual measure

(PESQ score) [14]. The relative volume is measured by the signal-to-interference-and-noise ratio (SINR).

The methods are implemented and tested both in a Matlab environment for initial simulations, as well as in the Simulink environment, where they are used in real-time experiments with real microphones and loudspeakers as the sound sources. We use three loudspeakers in total, one for the desired speech signal, one for the interference speech signal and one for the babble noise signal. The source files for the speech signals are selected from the Harvard word list database, as recommended by [15].

The results in the form of plots are found in figures 1-4. The top graph illustrates different scenarios at various noise volumes, the bottom graph looks at different echo volumes.

A. *Noise Reduction Performance*

First, we take a look at our benchmark, the MS+MWF algorithm in black. The performance is relatively steady in all scenarios, with good results in noisy environments.

On both graphs we can immediately see two distinct groups forming: the MWF algorithms that utilize rank 2 approximations (in blue) have considerable weaker noise suppression in comparison to the rest of the solutions. The trade-off appears in the PESQ scores, where these methods perform stronger. The seemingly inferior performance is caused by the rank 2 approximation itself. R_{nn} is only updated while both near end and far end speech is inactive, a situation which is relatively rare in any conversation. Less access to relevant data can have an negative impact on the estimation quality of R_{nn} and might lead to a poorly conditioned matrix.

The standard rank 1 algorithms (in green) have a very similar profile as the MS+MWF solution, with an overall slightly worse performance than the benchmark. The rank 1 MWF+MS has the worst performance, because the MWF has to deal with the far end signal without having access to the interference source file. Forcing the R_{ss} to be rank 1 in this case is a bigger approximation.

Looking at the integrated solutions (in red), we can observe that these methods in most cases outperform the rest of the algorithms. The MWF with FE subtraction has a slightly better profile than the subspace projection MWF algorithm across-the-board, especially in low iSNR and iSIR.

B. *Echo Cancellation Performance*

Overall, the MS+MWF algorithm reigns supreme in the echo suppression category. Especially in low iSIR scenarios, the areas that require high quality echo cancellation the most as we previously discussed, we can see very high gains (up to 17dB). The loud volume of the interference gives the MS filter a very clear input to identify the impulse responses between the source of interference and each microphone. In this case, the input to the MWF is already well conditioned and approaches rank 1, meaning the approximation is more justified. The residual interference is low in power, so it can be seen as just another noise source the MWF is well capable of suppressing. These two factors result in a very effective echo cancellation.

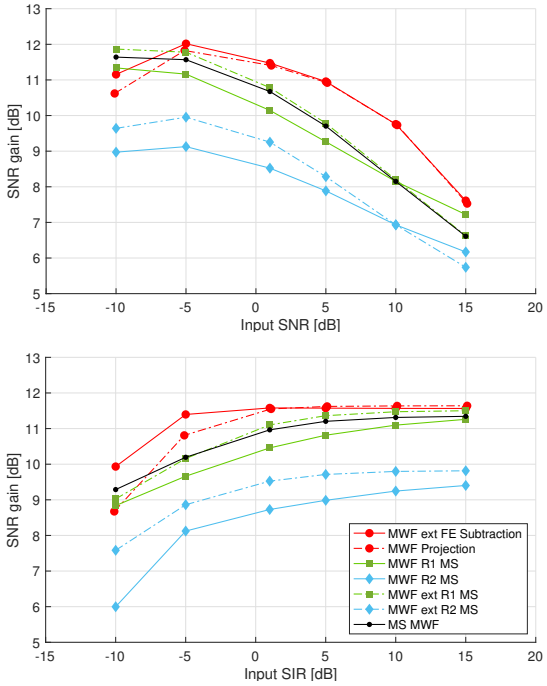


Fig. 1. Noise reduction performance measured by signal-to-noise ratio.

Similar to the noise reduction figures, we see both the rank 1 solutions (in green) and both the rank 2 solutions (in blue) grouping together. Both rank 2 based methods have a clear disadvantage compared to not only the rank 1 algorithms, but all of the tested solutions. The rank 1 MWF component actively helps cancelling out the echo, while the rank 2 implementation leaves the interference alone. It was conceivable that the MS filter could work much more effectively on the intermediate echo signal when the MWF does not tries to suppress it first. However, both the Matlab and real-time experiments show the same result: the echo volume gets reduces more if both the MWF (rank 1) and MS filter work on suppressing the echo.

The integrated methods work better than the rank 1 based discrete solutions, and seem to be more robust against differences in noise volume than the rank 2 based methods. However, they do not perform up to par against our benchmark in any of the real-time scenarios. Between the two integrated solutions, the FE subtraction methods outperforms the subspace projection method slightly, similar to the noise reduction results.

C. *Intelligibility Performance*

First, let us talk about the benchmark in black, the MS+MWF algorithm. It achieves respectable PESQ scores which are once again, quite stable over the different input scenarios. We see that the PESQ performance is almost exactly the same as the rank 1 methods, a bit worse than the rank 2 methods and a bit better than the integrated solutions. Our benchmark achieves good SINR values, with no other algorithm consistently outperforming it. Especially the great echo cancellation capabilities of the MS+MWF algorithms show in this figure, in the low iSIR area. The rank 1 solutions

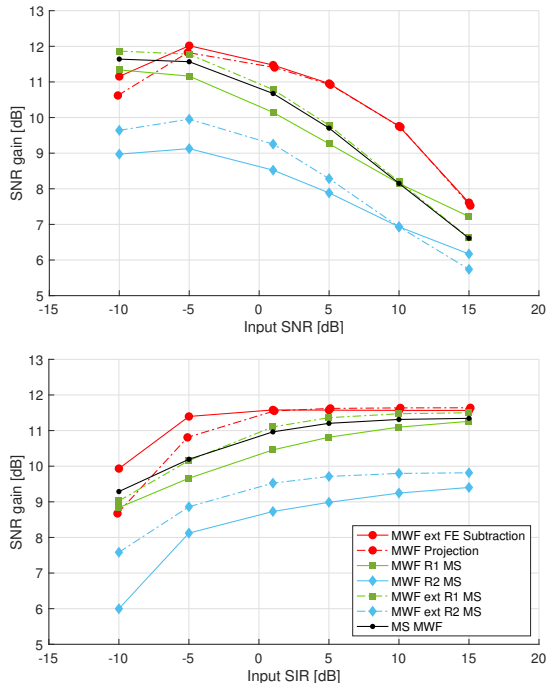


Fig. 2. Echo cancellation performance measured by signal-to-interference ratio.

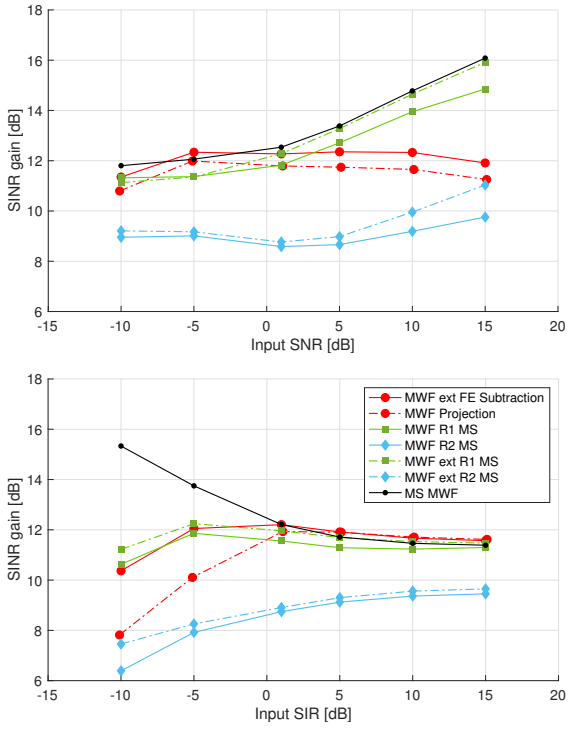


Fig. 3. Intelligibility performance measured by signal-to-interference-and-noise ratio.

(full and dotted green) have a very respectable and consistent performance, both in relative volumes and PESQ scores. We can see that rank 1 MWF+MS algorithm has very similar results as the benchmark (except for the SINR score in low iSIR scenarios), making it an interesting pick when computational requirements are strict. The extended rank 1 MWF+MS algorithm has very similar scores as well, performing slightly better in the SINR plots than its lower dimensional counterpart.

The rank 2 solutions (full and dotted blue) are probably the most interesting to evaluate, because they perform totally differently depending on which of the two measures we look at. They both achieved great PESQ scores, performing well above the benchmark and the other algorithms. The version without extra FE dimension performs best out of the two. However, when looking at the SINR values, they clearly perform way below average, sitting far below the benchmark and never even crossing into the territory of the other algorithms. This duality between both graphs is characterized by a low reduction in volume (sub-par SINR performance), but a low amount of distortion in the output signal (good PESQ scores). This behaviour is explained by the nature of the rank 2 MWF approximation. When the MWF selects the two subspaces with the highest SNR and setting the rest of the eigenvalues to zero, it does not forcefully throw away all the information about the far end signal. This results in less distortion of the output signal, but with no extra suppression of the echo signal.

The integrated methods (full and dotted red) have the worst PESQ scores in general. Especially the subspace projection method achieves relatively sub-par scores, specifically in low iSIR areas it fails to perform well. This also becomes clear during the real time experiments: the desired voice in the output sounds heavily filtered in some cases, moreso than with the other algorithms. The output is still usable and understandable though. If the application can cope with the distortion and additional computational requirement, the subspace subtraction method is still a fair option. The FE subtraction method has a more stable performance, staying relatively close to the benchmark. The PESQ scores take a small hit in low iSIR scenarios.

D. *Computational Requirements*

As we expected, the MWF+MS is computationally more beneficial than our MS+MWF benchmark. It only requires one instead of M MS filters. However, the gain in our setup is quite limited, resulting in a 8% faster execution in case of M = 3 microphones. We expect that this difference gets greater in setups with more microphones.

The subspace projection method requires a substantial amount of extra processing power, because of the two extra GEVD subtractions in every frequency bin. In our tests it requires at least twice the amount of work compared to the others methods.

The difference between the extended and normal methods is quite considerable, but since these measures are calculated with only three microphones, adding one extra dimension has a significant relative impact on the amount of calculations required. If the setup has more microphones however, the relative increase in execution time gets smaller. The subspace projection does not have this kind of benefit: the algorithm always requires two extra GEVD subtractions, independent of the amount of microphones in the setup.

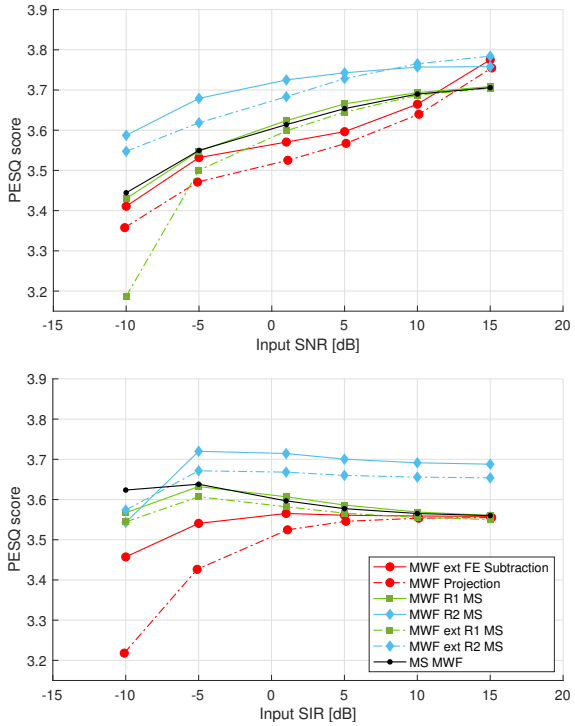


Fig. 4. Intelligibility performance measured by PESQ scores.

V. Conclusion

In this paper we investigate microphone signals that are both contaminated by excessive background noise and an interfering audio signal. Multiple methods for reducing the volume of both these noise and echo components are proposed and evaluated experimentally.

The experiments suggest that there are several effective ways to achieve joint noise and echo suppression. The two-step option we have chosen as a reference (MS+MWF) remains a good choice, due to its relatively fast performance and stable performance. Remarkable is the very effective performance in case of a loud echo signal.

The computational more efficient solution (MWF+MS) does not meet our expectations. The gain in execution time is too limited to compensate for the loss of performance compared to the reference method. The idea of equipping the MWF with the clean interference signal (MWF ext+MS) in order to improve echo suppression has also disappointed. This method still achieves slightly worse results than the reference, but now without the previous gain in computing time.

One of our integrated methods offers a good possible replacement for our reference. After evaluating our results, it is clear that the current form of the subspace projection method is too inefficient, with no equivalent gain in performance over the reference method. On the other hand, the integrated echo subtraction method is only slightly less efficient than the benchmark while achieving comparable and in some cases better results than the reference method. Further research is needed to compare it more closely with the reference method, with more performance indicators to be evaluated. Other scenarios can also be investigated.

References

[1] A. Geusens, A. Bertrand, B. Cornelis, and M. Moonen, "Multi-channel noise reduction in hearing aids with wireless access to an external reference signal," in *Proc. of the International Workshop on Acoustic Signal Enhancement (IWAENC)*, (Aachen, Germany), Sep. 2011.

[2] S. Vorobyov and A. Cichocki, "Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis," *Biological cybernetics*, vol. 86, pp. 293–303, 05 2002.

[3] S. Satheesh and K. V. S. V. R. Prasad, "Medical image denoising using adaptive threshold based on contourlet transform," *CoRR*, vol. abs/1103.4907, 2011.

[4] K. Chung, "Challenges and recent developments in hearing aids: Part I. Speech understanding in noise, microphone technologies and noise reduction algorithms," *Trends in Amplification*, vol. 8, no. 3, pp. 83–124, 2004.

[5] Zhengyou Zhang, Zicheng Liu, M. Sinclair, A. Acero, Li Deng, J. Droppo, Xuedong Huang, and Yanli Zheng, "Multi-sensory microphones for robust speech detection, enhancement and recognition," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. iii–781, May 2004.

[6] J. H. L. Hansen and M. A. Clements, "Constrained iterative speech enhancement with application to speech recognition," *IEEE Transactions on Signal Processing*, vol. 39, pp. 795–805, April 1991.

[7] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, pp. 235–238, June 1977.

[8] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.

[9] M. S. Brandstein and D. B. Ward, "Microphone arrays: Signal processing techniques and applications," pp. 41–57, 01 2001.

[10] B. Cornelis, M. Moonen, and J. Wouters, "Binaural voice activity detection for MWF-based noise reduction in binaural hearing aids," in *2011 19th European Signal Processing Conference*, pp. 486–490, Aug 2011.

[11] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Low-rank approximation based multichannel wiener filter algorithms for noise reduction with application in cochlear implants," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 785–799, April 2014.

[12] L. Trefethen and D. Bau, *Numerical Linear Algebra*, pp. 234–240. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 1997.

[13] A. Hassani, A. Bertrand, and M. Moonen, "LCMV beamforming with subspace projection for multi-speaker speech enhancement," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 91–95, March 2016.

[14] I.-T. P.862, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Feb. 2001.

[15] "IEEE recommended practice for speech quality measurements," *IEEE Transactions on Audio and Electroacoustics*, vol. 17, pp. 225–246, Sep. 1969.

Master's thesis filing card

Student: Sven Verret

Title: Joint Multi-Channel Noise Reduction and Echo Cancellation

Dutch title: Gezamenlijke meerkanaals ruis- en echo-onderdrukking

UDC: 621.3

Abstract:
In the field of signal processing, we are often confronted with contaminated microphone signals whenever we want to record a speech signal in a busy room. These signals are often affected by background noise and by disturbing audio signals generated by unwanted human speech or loudspeakers. If we also have access to the clean source signal of these unwanted audio signals, we call these echo signals. We want to increase the quality of our recordings by reducing the volume of both the noise and echo components, without excessively distorting or reducing the desired speech signal. In this paper, we discuss and test methods that are capable of achieving both noise reduction and echo cancellation in one integrated stage of signal processing. We compare the performances and computational requirements of these algorithms against classical two-stage solutions that serve as benchmarks. Then we will draw a conclusion based on both Matlab and real-time experiments whether it is compelling to use these integrated alternatives for scenarios similar to our research setup.

Thesis submitted for the degree of Master of Science in Electrical Engineering, im

Thesis supervisor: Prof. dr. ir. Marc Moonen

Assessors: Prof. dr. ir. Patrick Wambacq
Prof. dr. ir. Alexander Bertrand

Mentor: ir. Robbe Van Rompaey