COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

AUTOMATIC REMOVAL OF MUSIC TRACKS FROM TV PROGRAMMES

Carlos Pedro Vianna Lordelo

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Luiz Wagner Pereira Biscainho

Rio de Janeiro
Setembro de 2018

AUTOMATIC REMOVAL OF MUSIC TRACKS FROM TV PROGRAMMES

Carlos Pedro Vianna Lordelo

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Diego Barreto Haddad

Prof. Diego Barreto Haddad, D.Sc.

Wallace Alves Martins

Prof. Wallace Alves Martins, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2018

Lordelo, Carlos Pedro Vianna

Automatic Removal of Music Tracks from TV Programmes/Carlos Pedro Vianna Lordelo. – Rio de Janeiro: UFRJ/COPPE, 2018.

XVII, 69 p.: il.; $29,7$cm.

Orientador: Luiz Wagner Pereira Biscainho

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2018.

Referências Bibliográficas: p. 65 – 69.

1. sound source separation. 2. music extraction. 3. multimedia signals. 4. soundtrack. 5. film. 6. audio fingerprinting. I. Biscainho, Luiz Wagner Pereira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Primeiramente, gostaria de agradecer não só aos meus pais por todo o suporte dado durante meu longo período de graduação e mestrado, mas também aos meus dois irmãos por se manterem próximos e dispostos a me ajudar durante todo o tempo.

Agradeço também à minha namorada, Anna Carolina, que me atura há muitos anos e nunca deixou de entender a importância que dou a esta pesquisa e sempre me apoiou e me deu forças, mesmo nas horas mais difíceis (últimas semanas antes da data de defesa, principalmente).

Agradeço também aos amigos do curso, especialmente aos mais antigos e próximos que estão ao meu lado desde a graduação, Gustavo, Léo, Fonini, Petraglia, Helio, Mazza e Bandeira. As nossas "conversas de Burguesão" são sempre um bom motivo para aliviar um pouco a pressão do dia-a-dia. Ah, e não podia deixar de dar um encarecido obrigado ao Gustavo por me ajudar também na entrega da versão final do texto, pois me mudei para o exterior.

Uma consideração especial deve ser dada ao meu orientador, Luiz Wagner, que sempre acreditou na minha capacidade. Sua estrita exigência nesses últimos anos foi importantíssima para que obtivéssemos os ótimos resultados em todos os trabalhos que realizamos juntos. Obrigado também pelas matérias ministradas que tive a oportunidade de cursar, pelos necessários puxões-de-orelha (foram muitos) e pelos eventuais papos de psicologia infantil, como gosta de falar. Sou eternamente grato por tê-lo como orientador e amigo.

Devo agradeço também aos outros professores da banca, Wallace e Diego, pelas inquestionáveis dicas para melhoria do trabalho e pelas essenciais correções do meu texto.

Obrigado também aos alunos e professores do SMT pelas conversas e litros de café que dividimos durante todas essas tardes juntos no laboratório.

Por último, devo agradecer também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por me fornecer suporte financeiro durante o período de mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

REMOÇÃO AUTOMÁTICA DE TRILHAS MUSICAIS DE PROGRAMAS DE TV

Carlos Pedro Vianna Lordelo

Setembro/2018

Orientador: Luiz Wagner Pereira Biscainho

Programa: Engenharia Elétrica

Este trabalho está inserido na área de pesquisa de separação de fontes sonoras. Ele trata do problema de remover automaticamente segmentos de música de programas de TV. A tese propõe a utilização de uma gravação musical pré-existente, facilmente obtida em CDs oficialmente publicados relacionados à obra audiovisual, como referência para o sinal não desejado.

O método é capaz de detectar automaticamente pequenos segmentos de uma trilha musical específica espalhados pelo sinal de áudio do programa, mesmo que eles apareçam com um ganho variante no tempo, ou tenham sofrido distorções lineares, como processamento por filtros equalizadores, ou distorções não lineares, como compressão de sua faixa dinâmica.

O projeto desenvolveu um algoritmo de busca rápida usando técnicas de impressão digital de áudio e dados do tipo "*hash-token*" para diminuir a complexidade. O trabalho também propõe a utilização da técnica de filtragem de Wiener para estimar os coeficientes de um potencial filtro de equalização, e usa um algoritmo de "*template matching*" para estimar ganhos variantes no tempo para escalar corretamente os excertos musicais até a amplitude correta com que eles aparecem na mistura.

Os componentes-chaves para o sistema de separação são apresentados, e uma descrição detalhada de todos os algoritmos envolvidos é reportada. Simulações com trilhas sonoras artificiais e de programas de TV reais são analisadas e considerações sobre novos trabalhos futuros são feitas.

Além disso, dada a natureza única do projeto, é possível dizer que a dissertação é pioneira no assunto, tornando-se uma fonte de referência para outros pesquisadores que queiram trabalhar na área.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AUTOMATIC REMOVAL OF MUSIC TRACKS FROM TV PROGRAMMES

Carlos Pedro Vianna Lordelo

September/2018

Advisor: Luiz Wagner Pereira Biscainho

Department: Electrical Engineering

This work pertains to in the research area of sound source separation. It deals with the problem of automatically removing musical segments from TV programmes. The dissertation proposes the utilisation of a pre-existant music recording, easily obtainable from officially published CDs related to the audiovisual piece, as a reference for the undesired signal.

The method is able to automatically detect small segments of the specific music-track spread among the whole audio signal of the programme, even if they appear with time-variable gain, or after having suffered linear distortions, such as being processed by equalization filters, or non-linear distortions, such as dynamic range compression.

The project developed a quick-search algorithm using audio fingerprint techniques and hash-token data types to lower the algorithm complexity. The work also proposes the utilisation of a Wiener filtering technique to estimate potential equalization filter coefficients and uses a template matching algorithm to estimate time-variable gains to properly scale the musical segments to the correct amplitude they appear in the mixture.

The key components of the separation system are presented, and a detailed description of all the algorithms involved is reported. Simulations with artificial and real TV programme soundtracks are analysed and considerations about new future works are made.

Furthermore, given the unique nature of this project, it is possible to say the dissertation is pioneer in the subject, becoming an ideal source of reference for other researchers that want to work in the area.

Contents

List of Figures ix

List of Tables xi

List of Symbols xii

List of Abbreviations xvi

1 Introduction 1

1.1 Automatic Extraction of Specific Music Tracks 2

1.1.1 Music Rights in Brazilian Audiovisual Industry 3

1.2 Background & Related Works 4

1.2.1 Automatic Detection of Music Samples 6

1.2.2 Audio Fingerprinting 7

1.3 Objectives 7

1.4 Methods for Quality Assessment 8

1.5 Organisation of the Text 9

2 Detection and Synchronisation of Music Segments 11

2.1 Algorithm for Quick-Searching Music Segments 12

2.1.1 Audio Fingerprint Hash-Token Matrix Construction 13

2.1.2 Searching for Music Segments 16

2.2 Synchronisation of the Music Segments 17

2.3 Simulations for Quick-Search Method 19

2.3.1 Simulation 01 19

2.3.2 Simulation 02 21

2.3.3 Simulation 03 22

3 Time-Variable Gain Estimation 26

3.1 Fundamentals 26

3.2 Mathematical Analysis 27

3.3 Gain Estimation Simulations and Results 30

3.3.1 Simulation **01** 31
3.3.2 Simulation **02** 31
3.3.3 Simulation **03** 32

4 Estimation of the Filter Coefficients 39
4.1 Wiener Filtering Algorithm 39
4.1.1 Estimation of the Final Coefficients 43
4.2 Wiener Filtering Simulations 44
4.2.1 Low-pass Filtering Simulation 45
4.2.2 Band-pass Filtering Simulation 46
4.2.3 High-pass Filtering Simulation 46
4.2.4 Final Remarks 47

5 Non-linear Distortions 49
5.1 Dynamic Range Compression 49
5.2 Simulations 50

6 Removal of Specific Segments of the Music-Track 53
6.1 Complete Removal Algorithm 53
6.2 Artificial Signal Case Study 55
6.2.1 Simulation 01 – Constant Gain and no Filter 55
6.2.2 Simulation 02 – Variable Gain and No Filter 56
6.2.3 Simulation 03 – Constant Gain and Filter 57
6.2.4 Simulation 04 – Variable Gain and Filter 58
6.3 Real-life TV Programmes 59

7 Conclusion and Future Works 61
7.1 Final Remarks 61
7.2 Research Directions 62

A Objective Quality Assessment of Sound Source Separation Results 63

Bibliography 65

List of Figures

2.1 Part of the spectrogram of the soundtrack and the corresponding detected peaks. 14

2.2 Example of formation of spectrogram landmarks. 15

2.3 Example with the confirmed presence of an excerpt of a target music in the soundtrack signal. 18

2.4 Example with the confirmed absence of an excerpt of a target music in the soundtrack signal. 19

2.5 Variable gains applied to excerpts in Simulation 02. 21

2.6 Results of synchronisation using excerpts with 15-second duration. . . 24

2.7 Results of synchronisation using excerpts with 30-second duration. . . 25

3.1 Results of the time-variable gain estimation for $L = 25$ ms and $L = 50$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts. 33

3.2 Results of the time-variable gain estimation for $L = 100$ ms and $L = 200$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts. 34

3.3 Results of the time-variable gain estimation for $L = 25$ ms and $L = 50$ ms using a chirp as the reference gain curve. 35

3.4 Results of the time-variable gain estimation for $L = 100$ ms and $L = 200$ ms using a chirp as the reference gain curve. 36

3.5 Results of the time-variable gain estimation for $L = 25$ ms and $L = 50$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts. 37

3.6 Results of the time-variable gain estimation for $L = 100$ ms and $L = 200$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts. 38

4.1 Generic signal enhancement framework. 40

4.2 Dialogue-sfx enhancement framework. 40

4.3 Spectrogram of the first excerpt signal. 45

4.4 Spectrogram of the second excerpt signal. 45
4.5 Final low-pass Wiener filter estimation. 46
4.6 Final band-pass Wiener filter estimation. 47
4.7 Final high-pass Wiener filter estimation. 48

5.1 Example of soft-clipping/dynamic range compression functions. . . . 49
5.2 Results for the time-variable gain estimation procedure applied on compressed versions of soundtrack. 51

6.1 Time-variable gain curve applied to the excerpts during Simulations 02 and 04. 57
6.2 Frequency response of the original filter used in Simulation 02 and the respective response of the estimated Wiener filter. 57
6.3 Frequency response of the original filter used in Simulation 03 and the respective response of the estimated Wiener filter. 58
6.4 Frequency response of the original filter used in Simulation 04 and the respective response of the estimated Wiener filter. 59

List of Tables

2.1 Results of the quick-search method for Simulation 01. 20
2.2 Scores of the separation for Simulation 01 in dB. 20
2.3 Results of the quick-search method for linear gain curve in Simulation 02. 21
2.4 Results of the quick-search method for chirp gain curve in Simulation 02. 21
2.5 Scores of the separation for Simulation 02 in dB. 22

5.1 Separation results for both simulations in dB. 51

6.1 Separation results for Simulation 01 in dB. 55
6.2 Separation results for Simulation 02 in dB. 56
6.3 Separation results for Simulation 03 in dB. 58
6.4 Separation results for Simulation 04 in dB. 59

List of Symbols

$H(z)$	z-transform of the filter used in the generic signal enhancement framework, p. 39
M	size of the Wiener Filters, p. 41
N	total number of different Wiener filters estimated, p. 43
P	number of fixed spectral bases in PFNMF algorithm, p. 6
Q	number of non-fixed spectral bases in PFNMF algorithm, p. 6
$R_{\bullet\bullet}(n, n)$	statistical correlation of the sequences represented by bullet points at zero time lag, p. 29
T	length of the template signal, p. 27
$W(\mathrm{e}^{\mathrm{j}\Omega})$	DTFT of $\mathbf{w}(n)$, p. 44
$W(z)$	z-transform of the final Wiener filter used in the dialogue-sfx enhancement framework, p. 40
$W_i(\mathrm{e}^{\mathrm{j}\Omega})$	DTFT of $\mathbf{w}(n_i^\star)$, p. 43
Δf	frequency neighbourhood around an anchor point to look for landmarks, p. 14
Δt	number of forward frames to look for landmarks starting from an anchor point, p. 14
Δ	time instant where the template was added to the dialogue-sfx signal, p. 27
Δ_i	time instant where the excerpt of $m_i(n)$ appears in the soundtracks of some simulations, p. 19
$\mathbf{R}_{mm}(n^\star)$	statistical autocorrelation matrix of the reference signal $m(n^\star)$, p. 41

α constant gain applied to a short L-length template signal, p. 28

$\alpha(n)$ time-variable gain applied to the music excerpt, p. 27

δ time sample to cut a template signal from a reference music-track signal, p. 27

δ_i first sample of a segment of $m_i(n)$ that appears in some simulations' soundtrack at position Δ_i, p. 19

δ_i' last sample of a segment of $m_i(n)$ that appears in some simulations' soundtrack at position Δ_i, p. 19

$\epsilon(n)$ error signal of the generic signal enhancement framework, p. 40

$\eta_1(n)$ uncorrelated noise that is corrupting $x(n)$, p. 39

$\eta_2(n)$ another noise signal; different from, but correlated with $\eta_1(n)$, p. 39

$\hat{\alpha}$ estimated value for the constant gain α by the proposed algorithm, p. 30

$\hbar$ hash number calculated using the information of a single landmark, p. 16

$\mathbb{E}\left\{\bullet\right\}$ statistical expected value of the argument, p. 29

$\mathbb{L}$ hash-token matrix with the soundtrack landmark information, p. 16

$\mathbf{d}_{\scriptscriptstyle L}$ L-length vector with values from $d_{\scriptscriptstyle L}(0)$ to $d_{\scriptscriptstyle L}(L-1)$, p. 29

$\mathbf{m}(n^\star)$ vector comprised of M samples of the original music track signal starting from $n^\star$ until $n^\star + M - 1$, p. 41

$\mathbf{s}_{\scriptscriptstyle L}$ L-length vector with values from $s_{\scriptscriptstyle L}(0)$ to $s_{\scriptscriptstyle L}(L-1)$, p. 29

$\mathbf{t}_{\scriptscriptstyle L}$ L-length vector with values from $t_{\scriptscriptstyle L}(0)$ to $t_{\scriptscriptstyle L}(L-1)$, p. 29

$\mathbf{w}(n)$ vector with the coefficients of the final Wiener filter, p. 43

$\mathbf{w}(n^\star)$ vector with the coefficients of the Wiener filter estimated for the time sample $n^\star$, p. 41

$\mathbf{w}(n_i^\star)$	vector with the coefficients of the Wiener filter estimated for the time sample $n_i^\star$, p. 43
$\mathcal{A}\left\{\bullet\right\}$	average time-domain value of the argument, p. 29
$\mathcal{F}\left\{\bullet\right\}$	DTFT of the argument, p. 43
$\mathcal{R}_{\bullet\bullet}(0)$	temporal correlation of the sequences represented by bullet points at zero time lag, p. 29
$\mu(n)$	music-track signal of a TV programme, p. 2
$\overline{d_L}$	expected value of $d_L(n)$, p. 29
$\overline{t_L}$	expected value of $t_L(n)$, p. 29
$\mathbf{r}_{s\mathbf{m}}(n^\star)$	statistical correlation vector between $s(n^\star)$ and $\mathbf{m}(n^\star)$, p. 42
$\texttt{arctan}(\bullet)$	arctangent function of the argument, p. 50
$\texttt{std}(\bullet)$	function that returns the standard deviation of the vector passed as argument, p. 51
$\texttt{sum}(\bullet)$	function that returns the sum of the elements of the vector it receives as argument, p. 29
$\mathbf{w}(n^\star)$	vector with the Wiener filter coefficients calculated in the time of analysis $n^\star$, p. 42
$d(n)$	dialogue-sfx signal, p. 2
$d_L(n)$	realigned L-length version of the dialogue-sfx signal, p. 28
$d_T(n)$	T-length trimmed dialogue-sfx signal after having been properly aligned to the template, p. 28
$e(n)$	error signal of the dialogue-sfx enhancement task, p. 40
f_1	frequency bin of a landmark anchor point, p. 14
f_2	frequency bin of a landmark end-point, p. 14
$m'(n)$	$m(n)$ after being filtered by an unknown filter;, p. 40
$m(n)$	specific musical track that should be removed from a TV programme, p. 2
$m_i(n)$	different music signals used to create artificial soundtracks in some simulations, p. 19

$m_w(n^\star)$ output of the Wiener filter at time $n^\star$, p. 41

n time sample, p. 2

$n^\star$ particular time sample the Wiener filter coefficients are estimated on, p. 41

$n_i^\star$ one of the N time samples used to estimate the Wiener filters, p. 43

$s(n)$ soundtrack signal, p. 2

$s_L(n)$ realigned L-length version of the soundtrack signal, p. 28

$s_T(n)$ T-length trimmed soundtrack signal after having been properly aligned to the template, p. 28

$t(n)$ template signal to be searched in the soundtrack, p. 27

t_1 frame number of a landmark anchor point, p. 14

t_2 frame number of a landmark end-point, p. 14

$t_L(n)$ short L-length segment of the template signal, p. 28

$u(n)$ unit-step function, p. 19

$x(n)$ signal to be enhanced by the generic signal enhancement framework, p. 39

$y(n)$ output of the filter from the generic signal enhancement framework, p. 39

List of Abbreviations

BSS	Blind Source Separation, p. 1
DFT	Discrete Fourier Transform, p. 13
DTFT	Discrete-Time Fourier Transform, p. 43
EELD	Effective Excerpt Length Detected, p. 20
FDMV	Frequency-Domain Median Value, p. 44
FIR	Finite-length Impulse Response, p. 40
ICA	Independent Component Analysis, p. 5
MAPE	Mean Absolute Percentage Error, p. 30
MIR	Music Information Retrieval, p. 1
NMF	Non-negative Matrix Factorisation, p. 6
PEAQ	Perceptual Evaluation of Audio Quality, p. 9
PESQ	Perceptual Evaluation of Speech Quality, p. 9
PFNMF	Partially Fixed Non-negative Factorisation, p. 6
REPET	REpeating Pattern Extraction Technique, p. 5
SAR	Sources-to-Artefacts Ratio, p. 9
SDR	Signal-to-Distortion Ratio, p. 9
SIR	Signal-to-Interference Ratio, p. 9
STFT	Short-Time Fourier Transform, p. 13
TDAV	Time-Domain Average Value, p. 43
TDMV	Time-Domain Median Value, p. 44

TV	Television, p. 1
UCB	Brazilian Union of Composers, p. 3

Chapter 1

Introduction

In the era of information technology and with the advent of the Internet, we are able to easily access and share information using signals of different media. In a website, for example, some text materials are always present, even though its volume vary from page to page. Such texts are mixed with graphics, pictures, animations, videos, audio excerpts, and/or music, thus motivating the term 'multimedia signals' [1]. Therefore, a multimedia system can be viewed as a system that may include several time-dependent data (sound, video, and animation), spatially-dependent data (images, text. and graphics) or even other data types.

For instance, TV signals involve information coming from two different media, audio and video, which are mixed in a synchronised manner in order to properly convey the desired information. If the audio appears to be lagged behind the video or vice-versa, the results will not be pleasant. Moreover, there is also the possibility of finding text in this type of multimedia system because subtitles are often included in many broadcasting television channel signals. However, this project will only take into account the information obtainable through the analysis of the audio signal extracted from a television programme. No text or image data are going to be considered in any part of the signal processing techniques implemented and reported in this text.

This project is also closely related to an area of research known as Music Information Retrieval [2] (MIR). Explaining it in a few words, we may say that MIR researchers try to create methods to obtain any type of information they may need by performing detailed analysis of musical signals. Some examples of information of interest include multi-pitch estimation, [3], rhythm analysis [4], and the automatic music transcription [5]. For instance, Blind Source Separation (BSS) [6] methods commonly play an important part in MIR applications, and are also a frequent topic of discussion. Such type of methods are greatly welcomed in the music industry for their large grid of applications.

One of such applications is the automatic removal of a specific music track from a television programme.

1.1 Automatic Extraction of Specific Music Tracks

First of all, it is important to give the reader a better definition of the terms that are going to be used in this dissertation to avoid any type of ambiguity or misinterpretation. The term 'soundtrack' is going to be used throughout the text to represent the full set of sounds in a television programme, i.e., in addition to the music-track that may exist, it also includes the sound effects of the scene and the voices of the actors. On the other hand, the term 'music-track' is going to be used to identify only the set of musical signals present in the audio signal; it may include (or not) original music, created specifically for the audiovisual work, or other musical pieces, songs, and excerpts of pre-existing musical recordings. Also, there is a third term constantly used in the dissertation that represents every signal that is part of the soundtrack, but is not part of the music-track. The chosen term was 'dialogue-sfx' signal because it includes speech signals (dialogue of the characters) and the sound effects of the scene.

In general, we may mathematically define the soundtrack of any audiovisual piece as:

$$s(n) = d(n) + \mu(n), \tag{1.1}$$

where $s(n)$ is the soundtrack signal, $d(n)$ is the dialogue-sfx signal and $\mu(n)$ is the entire music-track signal of the programme, which can be understood as a non-simultaneous combination of many musical excerpts (without overlapping samples) coming from different recordings that have been put together to form a long music-track signal.

Thus, it is possible to define the idealised process of automatic removal of a specific music track $m(n)$ from TV programmes as the ability to analyse the whole soundtrack signal and remove as many information as possible from $\mu(n)$ related to $m(n)$, leaving behind the dialogue-sfx signal and other musical pieces that appear in $\mu(n)$ untouched.

It is worth noting that the set of music tracks of a television programme such as films, series, or soap operas is an essential part of the audiovisual work. The musical track is responsible for contributing to the setting of the scene by arousing a wide range of feelings in the spectators. It not only has the ability of provoking emotive or excitement sensations, but it is also able to improve the spectators' anguish and frightening experiences. Thus, why would a post-processing music-track extraction may be required if the music signal plays such an important role in the audiovisual material?

The answer lies in the copyright laws that involve the whole process of creation, reproduction and broadcasting of any audiovisual work.

1.1.1 Music Rights in Brazilian Audiovisual Industry

In Brazil, there is a non-profit organisation known as Brazilian Union of Composers (UCB) that executes the collective management of music-related rights of Brazilian creators, publishers, performers, and producers. In their website[1], it is possible to find useful information about many types of music-related copyrights.

According to [7], one of the published UCB guides, there are 3 different music-related rights concerning the utilisation of a musical piece in any sort of audiovisual work in the country.

- Right of Inclusion or Synchronisation: when film or TV producers want to include a specific music in an audiovisual material, they have to ask for an inclusion permit, also known as synchronisation permit, to the legal owners of the music copyrights. To get this legal authorisation there is a negotiation between the interested parts. Many factors can contribute to the amount of money the producers will have to pay; some examples are the period of time the music will be playing in the audiovisual material, and whether or not the music will have an important role in the final version of the work.

- Right of Reproduction: it refers to the physical or digital reproduction of the audiovisual work. Each time a new copy of the work is reproduced, the owners of the music rights also should receive a payment. Usually, the negotiation for the right of synchronisation already includes the right of reproduction.

- Right of Public Broadcasting: when an audiovisual work is exhibited for the general public by any means, i.e., cinema exhibition, TV broadcasting or re-broadcasting, etc., the creators have to pay for the right to broadcast the musical pieces present in their creation.

A special attention should be given to soap operas and TV series production. In Brazil and in many parts of the world, it is really common for a TV broadcasting channel to have their own soap operas and/or series productions. After having negotiated for the rights to include, to reproduce, and to broadcast the music tracks they want to use in their audiovisual work, they exhibit the complete piece for some time.

However, if they want to re-exhibit an old production that has been created long time ago, they have to make sure they still have the rights of synchronisation and

[1] `http://www.ucb.org.br/english`, accessed in 01/08/2018.

of public broadcasting of every song that is part of the original soundtrack. In some cases, the rights might have expired, and, therefore, some musical contents must be removed from the recent version so that the content can be exhibited.

In other words, there are cases where some years after the production of a soap opera, for example, one may have the rights to play the video signal, but at the same time may have lost the rights to play some of its original musical signals. If it is not possible to retake the scenes, it becomes mandatory a complete substitution of every appearance of specific music tracks in the video's soundtrack signal, without affecting the quality of the characters' dialogues and the other sound effects.

In that regard, source separation methods are very attractive because often there is no more access to each individual signal, i.e., the audio part of the audiovisual piece is often stored with the music-track already mixed with the other sound sources. There is no separated dialogue-sfx signal available to remaster the soundtrack.

1.2 Background & Related Works

As far as the author was able to verify, there is no work in the specialised literature that proposes a technique to execute the automatic removal of a specific music from audiovisual signals. However, [8] addresses a similar topic: the removal of the whole music-track signal along with the sound effects of a film. According to the article, it is possible to use many international versions (with different dubbing languages) of the same film to remove both signals. The main idea behind the method is the fact that the music-track and the sound effects remain unchanged in all the international versions. Thus, it would be possible to remove them from the mixture by making a time alignment between the versions followed by a simple median filtering procedure.

It is easy to realise that this type of technique for music-track removal cannot be used under the perspective of this dissertation, not only because the sound-effects signal must not be removed, but also, and more importantly, because there is only one mixture available, i.e., there is only a single version of the original soundtrack signal.

Notwithstanding, there are some parallels that can be drawn with other research topics. For instance, the separation of singing voice from the instrumental accompaniment in a music signal is a largely discussed topic in the sound source separation community, and it can also be seen as being a dual problem if compared to the music-track removal task.

In general, simple techniques that deal with this task try to mathematically model the instrumental accompaniment signal of a musical recording with the objective of removing it from the mixture, leaving the singing voice in the residual signal. They often use the concept of musical repetition [9], which is based on the fact that musical pieces are structured as a mixture where the singer add time-variant words over a repetitive instrumental accompaniment signal. In other words, it can be stated that a musical recording with a singing voice and an instrumental accompaniment has many different verses sung over the same repetitive chord progression. Hence, the usual principle of operation is to recognise the repetitive patterns in the audio signal and use them to separate the repeating background (instrumental accompaniment) from the non-repeating foreground (singing voice).

For example, the REpeating Pattern Extraction Technique (REPET), as shown in [10], is able to generate a 'beat spectrum' for the recording, a time-variable function that stores information about the music repetitive structure. The author proposes the creation of this signal by computing the autocorrelation function [11] of each frequency row in the squared magnitude of the music spectrogram [12] (supposing the columns represent the time-frames and the rows represent the frequency bins) followed by the average value in each column. After this, it is possible to estimate a 'general period' for the repetitive patterns in the song by analysing the peaks of the beat spectrum. In the end, the REPET algorithm is able to create a spectrogram that represents the repetitive patterns, which can then be utilised to separate the accompaniment from the singing voice.

Another similar method, proposed in [13], implements an adaptation of the REPET algorithm to handle time-variations in the repeating background. It models the repetitive patterns as periodically time-variant and generates a 'beat spectrogram' instead of a simple 'beat spectrum'. First, it tracks local periods for the repetitive structure, then estimates local models for the repeating background, and finally extracts the patterns.

There are also more complex and hybrid methods, where pitch estimation techniques are included in the algorithm to find the melodic contour of the voice signal. The proposal in [14] mixes analysis in the frequency domain by using Independent Component Analysis (ICA) [15] and a method denoted by the author as Amplitude Discrimination, with the posterior time domain analysis and pitch estimation. In article [16], one estimates a rhythm mask to represent the repetitive patterns of the signal using a similarity matrix of the magnitude spectrogram of the musical recording and identifies the pitch contour of the singing voice utilising a multi-pitch estimation algorithm. More recently, deep learning techniques, such as convolutional neural networks, started to flourish in the specialised literature related to the theme [17, 18].

Despite being a topic continuously improved by academic research, methods of singing voice and instrumental accompaniment separation were not considered by the author the best approach to address the music-track removal problem. The music-track signal often appears in small segments throughout the whole soundtrack of the audiovisual work, which makes the mixture to lose musical repetition, a fundamental concept utilised by those types of methods. Moreover, in the majority of cases, the music-track signal also has a singing voice that is an intrinsic part of it and, therefore, should also be removed from the mixture along with the instrumental accompaniment.

Before concluding the section, it is important to cite two more research topics with works that were judged by the author as the closest methods related to the current research project objective. They are known in the literature as automatic detection of music samples [19–22] and audio fingerprinting [23–26].

1.2.1 Automatic Detection of Music Samples

'Music sampling' is a common process in the music industry. It refers to the use of excerpts or segments of a pre-existing music in the creation of new music pieces, mash-ups, or other musical productions. The ability to automatically verify if a sampling of a particular music has been included in another production would not only help in studies about geographical and temporal influences of a certain artist, but also help in the detection of plagiarism.

It is possible to analyse the music-track of an audiovisual material as being a 'sampling' of small segments of many musical recordings which have been inserted into a signal with the characters' dialogues and the sound effects. Methods for automatic detection of music samples, therefore, are really attractive to be used in this dissertation.

An example of a recent work that addresses this problem is [19]. The article proposes the use of an algorithm of Partially Fixed Non-negative Matrix Factorisation (PFNMF) to perform the detection. In a few words, this algorithm first implements the standard NMF factorisation [27] of a reference segment, that has supposedly been used in another musical work, using P basic spectral components (number of columns of the spectral bases matrix or the number of rows of the activation matrix); afterwards, the music suspected of having the previous segment inserted into is also factorised, but, this time, using $P + Q$ components, where the first P are fixed during the factorisation. Only the other Q spectral components that will be iteratively adapted to model the remaining sound sources present in the music. If the sampling has been confirmed, the activation matrices of both signals will have the same sequence of activations in the positions associated with the first P spectral components.

1.2.2 Audio Fingerprinting

In this application, the objective is to extract any type of content signature from audio signals. Such signatures are called Audio Fingerprinting Signatures because they are unique and capable of identifying the audio signal [28], even in the presence of noise, or subjected to losses of Internet broadcasting and ambient reverberation [25]. They can also be robust to pitch shifts and changes in the music tempo [26]. These types of techniques are greatly used in systems for automatic recognition of music such as Shazam[2], but can also be used in the detection of music samples [21, 22].

The high robustness of the audio fingerprinting algorithms suggested they could be used for the automatic music-track removal. It was expected that these algorithms would not only be able to automatically detect the instants in a soundtrack where there are segments of a specific music signal, but also identify the segments themselves, even though they might have suffered modifications such as linear filtering, time-variant gain levels, and non-linear distortions. An adaptation of an audio fingerprinting algorithm was used in the detection step of the automatic music tracks removal system and is explained in detail in Chapter 2.

1.3 Objectives

In a few words, it is possible to state that the final objective of the research project is the development of signal processing techniques for the automatic removal of a specific music signal from audio content in an audiovisual material.

More specifically, the dissertation approach to the extraction problem is to use a pre-existing musical recording as a reference signal for the musical track that should be removed from the mixture. This signal is denoted by $m(n)$ in this dissertation. It should be noted that it can be easily obtained from CDs that are often officially published by the producers of the programme and contain with the full set of music tracks present in the audiovisual piece.

The reference signal is definitely an useful tool to help with the removal procedure; however, there are still many challenges that the research project will have to face:

- It is not possible to ensure which segments in the reference signal are effectively present in the soundtrack of the audiovisual piece. Also, it is not trivial to automatically check the correct instant of the soundtrack where each musical segment appears.

[2] `https://www.shazam.com`, accessed in 01/08/2018.

- The editors of the audiovisual piece might have applied different time-variable gains on each musical excerpt included in the television programme.

- The editors might have also used equalisation filters on the music-track before adding it to the dialogue-sfx signal. This procedure generates different versions for the musical segments that are not exactly the same as the original ones which we have access to.

- There is also the possibility of existence of non-linear distortions in the final musical segment that actually appears in the soundtrack.

Thus, the implemented algorithms, which are the main product of the research project, are proposed considering the ability to handle the majority of those problems while trying to get the best possible quality in the post-processed separated signal, which, in a real-life case scenario should include the dialogue-sfx signal $d(n)$ along with the parts of the programme music-track signal $\mu(n)$ that are not samples of $m(n)$.

However, it is important to say that, given the unique nature of the research project and all the complexity involved in this type of sources-separation problem, the goal of the dissertation is to give a detailed description of the separation problem, as well as the main challenges involved in it, and propose and implement new techniques to perform the separation in artificially created environments, where simplifications can be made.

The application of the proposed separation algorithm to soundtrack signals from real-life TV programmes is discussed in Chapter 6, and some guidelines for improving the results are also presented.

1.4 Methods for Quality Assessment

Evaluating the quality of musical signals is an essentially subjective task. Each person perceives the sound in a different way; some are capable of noticing small nuances between two similar signals, whereas, for others, they may pass unnoticed. In that regard, it is always recommended to perform subjective tests, asking people to give a score for the separated signals. In [29] the reader can find more information about how to proceed when making subjective tests and the large complexity they may involve.

Motivated by the fact that such tests demand a large amount of people and strictly controlled conditions, some automatic assessment methods have arisen in the literature with the objective to simulate the human perception and to make easier the comparison of the algorithm results. Differently from the subjective methods, the objective methods for quality assessment can be easily applied to the experimental results to instantaneously get scores representing their quality.

There are many objective methods available in the literature. Some of them are complex techniques specially designed to evaluate the quality of audio signals taking into account concepts of the psychoacoustic theory [29], which is basically the mathematical modelling of how humans effectively perceive the sound they hear. Some examples are the Perceptual Evaluation of Speech Quality (PESQ) [30], focusing on speech intelligibility, and its general counterpart, the Perceptual Evaluation of Audio Quality (PEAQ) [31], aiming at overall audio fidelity.

In the context of audio source separation, there is a set of largely used objective methods that (while not perceptually inspired) are much simpler to understand. Those methods were created based on the Signal-to-Noise Ratio (SNR) and actually give a realistic score for the expected quality of the separation results because they quantify the amount of interferences (Signal-to-Interference Ratio – SIR), distortion (Signal-to-Distortion Ratio – SDR) and artefacts (Sources-to-Artefacts Ratio – SAR) that are present in the separated signals. For this reason they represent an interesting metric to evaluate the algorithms proposed in this dissertation. They will be the major figure of merit for the analysis of the separation results and will continuously appear along the text. The interested reader can check Appendix A for more details about the SNR-related objective quality assessment methods.

They were introduced in [32], and in [33] it is possible to obtain a publicly available MATLAB implementation that performs the objective assessments.

Before continuing, it is important to point out that such methods can only be used when the reference signals for the separated results are available. Under the project perspective, the desired signal is the dialogue-sfx signal, which is not available in a real-life application. Therefore, only the artificial soundtracks that have been created for the simulations could be evaluated using those methods.

1.5 Organisation of the Text

Each challenge stated in Section 1.3 will have its own chapter in the dissertation explaining how it has been addressed by the author. The chapters will also present MATLAB simulations to test how the respective part of the separation algorithm performed. The text starts with Chapter 2 explaining how the first step of the removal procedure should be executed. It gives a detailed description of the quick-search method developed to look for musical segments in the soundtrack signal. Then, Chapter 3 shows how the algorithm deals with time-variable gains applied to the musical excerpts. It proposes the usage of a template matching technique to estimate the gain curve before executing the excerpt removal. Chapter 4 discusses a simple Wiener filter procedure to estimate a potential equalisation filter that could have been used in the excerpts before putting them in the soundtrack as well. Chapter 5 is related to the analysis of non-linear distortions applied to the musical excerpts, and Chapter 6 wraps everything up, putting together each step discussed in the previous chapters and explaining the logic behind the whole removal process. It also presents some results of the algorithm when applied to artificial signals. The final remarks considering soundtracks from real-life TV programmes are presented in Chapter 7.

Materials

All the necessary materials for the execution of the project were provided by the Signals, Multimedia and Telecommunications Laboratory (SMT), highlighting the use of original DVDs of a Brazilian soap-opera [34] and the associated officially published CDs [35] with the songs that are part of its music-track signal.

Chapter 2

Detection and Synchronisation of Music Segments

As previously stated in Equation (1.1), and rewritten here for convenience, the simplest way to mathematically model the soundtrack signal of a television programme is

$$s(n) = d(n) + \mu(n), \tag{2.1}$$

where the signal $s(n)$ is the soundtrack, $d(n)$ is the dialogue-sfx signal and the signal $\mu(n)$ is the music-track signal with all the musical information included in the programme.

If we take a deeper look into $\mu(n)$, it is possible to see it as a sum of many musical excerpts, coming from different music tracks. Some of them have been sampled from a particular music recording $m(n)$ and should be removed from the programme. However, before performing the removal procedure, the algorithm must firstly search in the soundtrack signal for small excerpts of the target music. An intuitive idea to accomplish that would be to divide the music signal $m(n)$ into small excerpts of a few seconds of duration and compute the cross-correlation coefficient [11] between delayed versions of the soundtrack signal. If somewhere its value gets closer to 1, one could conclude that the associated musical excerpt is present around that position in the mixture signal.

However, this approach is not only computationally exhaustive, but it would also not work as expected because it is really common to apply time-variable gains in the musical excerpt before adding it to the dialogue-sfx signal. This process might generate an unpredictable bias in the cross-correlation function, which makes harder to correctly set the condition to confirm the presence of a particular segment in the mixture.

In that regard, the author has decided to utilise a method based on an audio fingerprinting technique [25], even though the original algorithm is used for music recognition. The classic idea is to create a large hash-table using pairs of the most prominent peaks of the spectrogram of a reference musical recording. This hash-table is unique for each song; therefore, it is possible to say that the hash-table is a reproducible fingerprint that is able to uniquely identify a song. If there is a random musical excerpt we want to identify, the same hash-table creation procedure can be applied to it and its hashes can be compared with the hashes of each reference signal present in a database. If a large number of them are equal for a specific song, one can conclude that the excerpt comes from it.

In this dissertation, a MATLAB implementation of the classic audio fingerprinting algorithm, which can be freely obtained in [36], has been adapted to work as a quick-search method to be used in the first part of the music tracks removal procedure.

2.1 Algorithm for Quick-Searching Music Segments

According to [25], there are four main characteristics of an audio feature that make it ideal to be used as an audio fingerprint or signature. The ideal audio feature should be:

- Temporally localised;
- Translation-invariant;
- Degradation robust;
- Sufficiently entropic.

Being temporally localised means that each fingerprint hash should be calculated using audio samples from a small region in time, so that distant events do not affect the hash. The translation-invariant characteristic is important to guarantee the reproducibility of the fingerprint hash independently of its position within the audio file, as long as the temporal locality containing the data from which the hash is computed is contained within the file.

In the original article, degradation robust means the fingerprint hashes must be robust to noise, reverberation and possible losses of Internet broadcasting. Analogously, as our quick-search method, the audio fingerprinting algorithm must keep its level of robustness high, since it should be able to identify the musical excerpts, even if analysing a soundtrack signal where time-variable gains may be present and where there may exist the concurrent presence of speech signals coming from the characters' dialogues, which can be considered as noise for the method. The sufficiently entropic guideline is necessary to minimise the probability of false fingerprint matches at non-corresponding locations. On the other hand, too much entropy leads to fragility and non-reproducibility of fingerprint tokens in the presence of noise and distortion.

The peaks of the spectrogram are a simple feature that follows many of those guidelines. It is possible to note that they are temporally-localised, translation-invariant and robust to noise [25]. Moreover, they have another interesting property that makes them even more attractive: the peaks of the spectrogram have an approximate linear superposability [25], i.e., if a new source is added to the mixture, it could be approximated by new peaks in the spectrogram; it would not influence the previous peaks that were present before). The only problem is that they have a really low entropy because the same spectrogram peak can appear in many different excerpts and a fingerprint token calculated using just a single peak information would not work properly because many different songs would have the same signature. In order to solve that issue, there is another step that increases the entropy before calculating the hashes.

2.1.1 Audio Fingerprint Hash-Token Matrix Construction

Using a sampling frequency of 48000 Hz, the quick-search algorithm starts with the computation of the log-magnitude spectrogram of the mixture (soundtrack) signal. The Short-Time Fourier Transform (STFT) [12] is employed with a Discrete Fourier Transform (DFT) size of 2048 samples, Hamming windows of the same length and with 50% overlap. Then, the algorithm detects the most prominent peaks in the soundtrack log-magnitude spectrogram. The way the peak detection is performed does not change considerably the results of the search method; the most important is that even the smaller peaks in the frame should also be considered as candidates, so that, it is not recommended the utilisation of a global threshold for the peak detection step. Remember we are trying to find the peaks of the soundtrack spectrogram that are part of the music-track signal, and they usually have a lower energy if compared to the peaks that come from the dialogue-sfx signal. The algorithm accomplishes this using the same approach as in [36]: it applies a Gaussian spreading function on each candidate peak, from the highest to the lowest, with the objective of masking the possible presence of windowing oscillations in its neighbourhood. If the candidate is above the final Gaussian threshold, it is considered a prominent peak. In each frame, the maximum number of possible peaks is set to 15; if more are found, only the higher ones are considered for the creation of fingerprint hashes. After the peaks have been detected, it is possible to obtain a ‘Constellation Matrix’, which can be understood as being the “useful-part” of the spectrogram, i.e., the part where the prominent peaks are present. The other parts of the spectrogram can be discarded. Figure 2.1 shows an example of a 2-second log-magnitude spectrogram of a soundtrack signal, with the corresponding constellation matrix with the detected peaks on top. Observe that the amplitude information of the peaks have been ignored from now on; this makes the quick-search algorithm more robust to time-variable gains and noise.

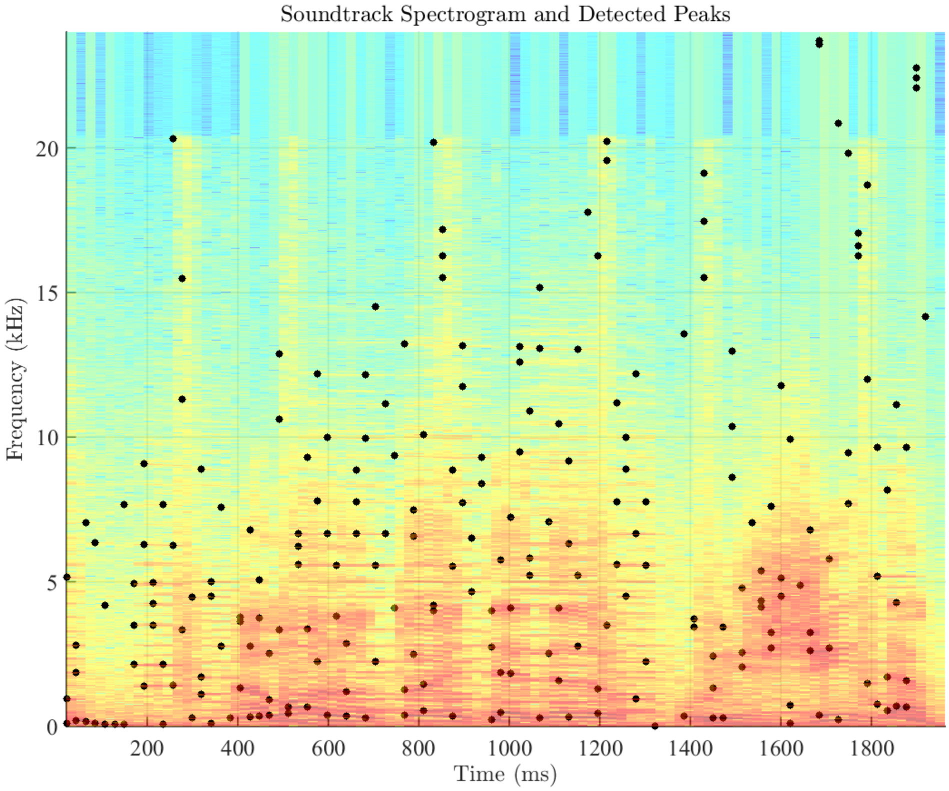


Figure 2.1: Part of the spectrogram of the soundtrack and the corresponding detected peaks.

With the objective of increasing the entropy of the system, instead of using the standard peaks as possible signatures for the signal, we are going to use pairs of peaks taken in a pre-defined region around its neighbourhood. Each peak in the frame is treated as an anchor point and, starting from it, a target region can be defined using a Δf of $\pm$ 63 frequency bins and a Δt of 31 frames forward. The other peaks that lie inside this region are linked to the anchor point to form what [25] calls 'landmarks'. Note that each landmark is uniquely defined using the coordinates of its anchor point (t_1, f_1) and the coordinates of its end-point (t_2, f_2). Furthermore, it is important to realise that t_1 and t_2 are actually the frame indexes of the spectrogram, but we can consider them as representing time offsets from the beginning of the soundtrack file. A total number of landmarks per anchor point is defined as 5, thus the algorithm does not spend time computing too much unnecessary information. It starts checking in the same frame where the anchor point is located for peaks that are inside the target region. Only if less than 5 are found the algorithm passes to the next frame. This search is done until it reaches a maximum of 5 landmarks or +31 frames forward. Figure 2.2 shows an illustration of how the landmarks are formed.

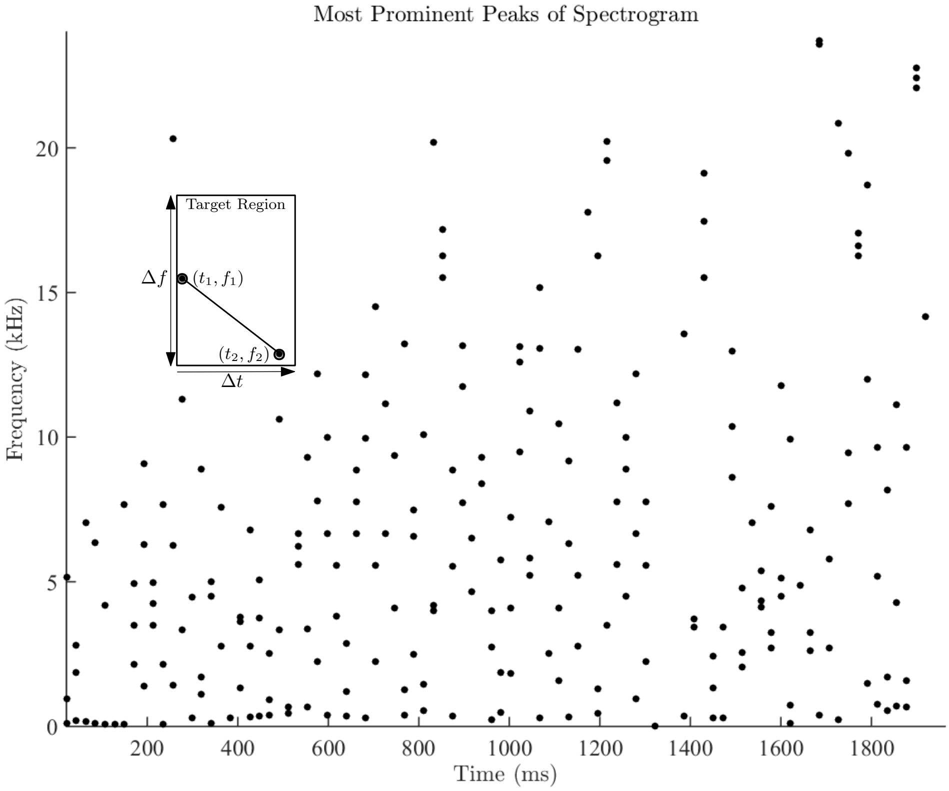


Figure 2.2: Example of formation of spectrogram landmarks. The anchor point has the coordinates (t_1, f_1) and the end-point has coordinates (t_2, f_2). If there were another point inside the target region, a new landmark would be created linking it to the same anchor point.

In a classic audio fingerprinting environment, there are hundreds of thousands of songs that pass by this procedure and have their landmarks stored in a large hash-table with the corresponding song ID. Later, if someone wants to identify a random musical excerpt, the algorithm computes the excerpt's landmarks and compare them with the landmarks stored in the hash-table. If there are sufficient matches originated by a single song, it is possible to say that the excerpt comes from the corresponding song. Counter-intuitively, in the project's perspective, it is proposed to store the landmarks of the mixture $s(n)$ signal in a hash-token matrix instead of storing the landmarks of the original music-track. The reason for this is better explained in Chapter 6, but it can be said that it makes the removal task an easier iterative procedure. The reference music signal $m(n)$ is divided into smaller excerpts without overlapping samples and each segment passes by the same landmark-hash-token process. Later, their fingerprint tokens are compared with the tokens stored in the database and, if, for a given excerpt, there are many matches with the same relative time offset, we can conclude that this particular excerpt is present in the mixture. Furthermore, using a simple procedure it is also possible to estimate a value for the time instant that the musical segment starts in the soundtrack signal.

Let's call $\mathbb{L}$ the hash-token matrix which will be storing the landmarks information related to the soundtrack signal. This matrix is first created as an empty row vector with a pre-defined length of 2^{22}. Considering each column index as a 22 bits hash number $\hbar$, it is possible to put the landmark information in $\mathbb{L}$ for quick-access later according to the following hash-token procedure:

- Put the value of f_1 in the first 10 bits of $\hbar$ (0 to 1023 if we use the right-hand side of the STFT);
- Put the value of $f_2 - f_1$ in the next 7 bits of $\hbar$ (0 to 63 if $f_2 \geq f_1$, and 65 to 127 if $f_2 < f_1$);
- Put the value of $t_2 - t_1$ in the last 5 bits (0 to 31);
- If the position $\mathbb{L}_{\hbar}$ is empty, store the time offset t_1 as a token there. Else, add a row in $\mathbb{L}$ and store t_1 as a new token in the next row, in the same column $\hbar$.

2.1.2 Searching for Music Segments

To perform a search, the fingerprinting operation above is performed on an excerpt of the musical recording we have available to generate a new set of hash-token values. Each hash from the excerpt can be used to search in $\mathbb{L}$ for matching hashes; if the excerpt is actually present in the mixture, the matches should occur at the same relative time offsets (the relative time offset can be defined as the time offset t_1 of the excerpt subtracted from the time offset t_1 of the matched landmark present in the database, which has been originated from the soundtrack signal). Therefore, a decision regarding the excerpt presence in the mixture can be made by analysing the values of the differences between their corresponding tokens.

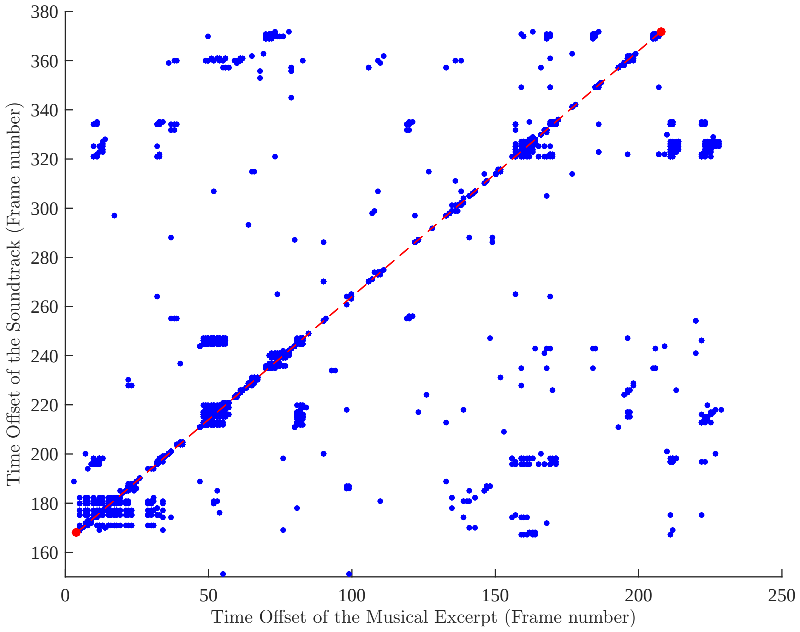
In other words, the problem of deciding whether or not a musical segment has been found in the mixture reduces to detecting a significant cluster of points forming a diagonal line within a scatter plot with their time offsets. Figure 2.3 shows an example of a confirmed presence of part of the musical excerpt in the soundtrack, while Figure 2.4 shows the processing results for an excerpt that is not present in the soundtrack.

2.2 Synchronisation of the Music Segments

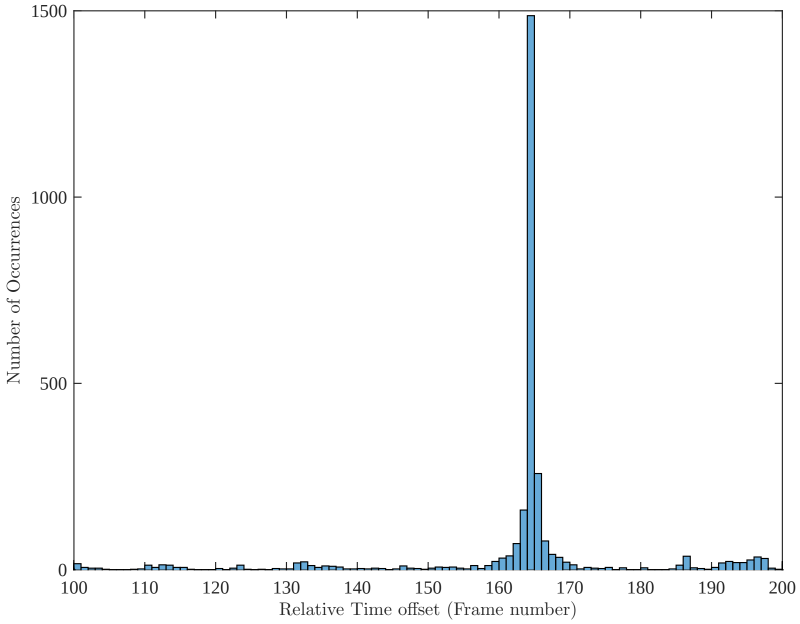
The synchronisation can be easily done once there is a confirmation of the presence of a musical excerpt in the soundtrack. Considering the points on top of the red dashed line in Figure 2.3a, it is possible to use the absolute time offsets of its edge points to create a segment of the soundtrack signal that is frame-by-frame matched with a segment of the musical excerpt. The red points of the Figure 2.3a shows the first and last frames to synchronise the signals.

However, it is important to realise we have been working with time offsets that have units of STFT frame number because we use information of a whole window to calculate the landmarks for a single time offset. Therefore, the synchronisation procedure should make a finer grain analysis after the signals have been frame matched in order to get the correct initial and final samples they should be aligned with. The sample-by-sample alignment is done using the autocorrelation function between the matched segments.

Before concluding, it should be noted that the quick-search algorithm, in theory, is capable of automatically trimming the correct 'window-size' from the musical excerpt. Analysing the Figure 2.3a, we can conclude that even though we used segments of the music-track signal with lengths of 250 frames, the red edge-points on the red line would permit us to fix the length of the musical segment and use a smaller and better matching signal excerpt. This leads us to believe that it would be better using larger excerpt sizes for detection.

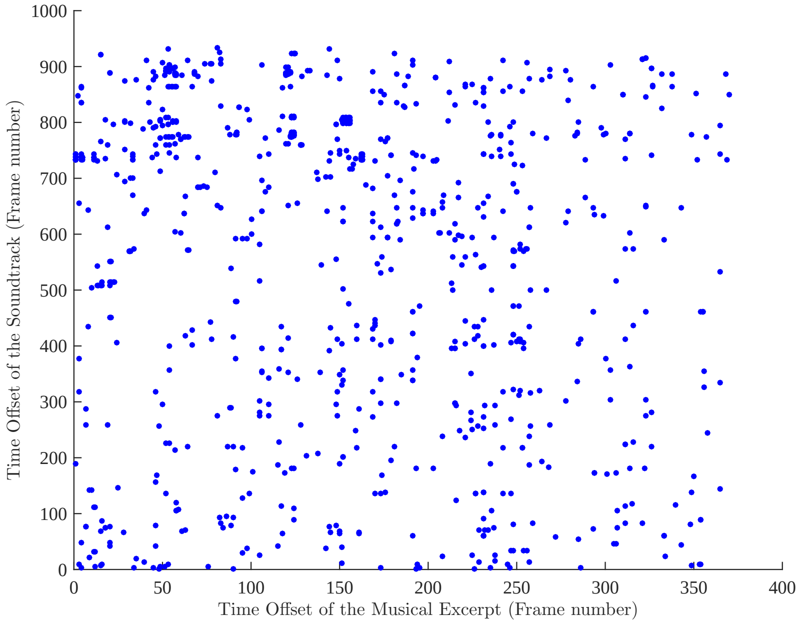


(a) Scatter plot with the music and soundtrack time offsets.

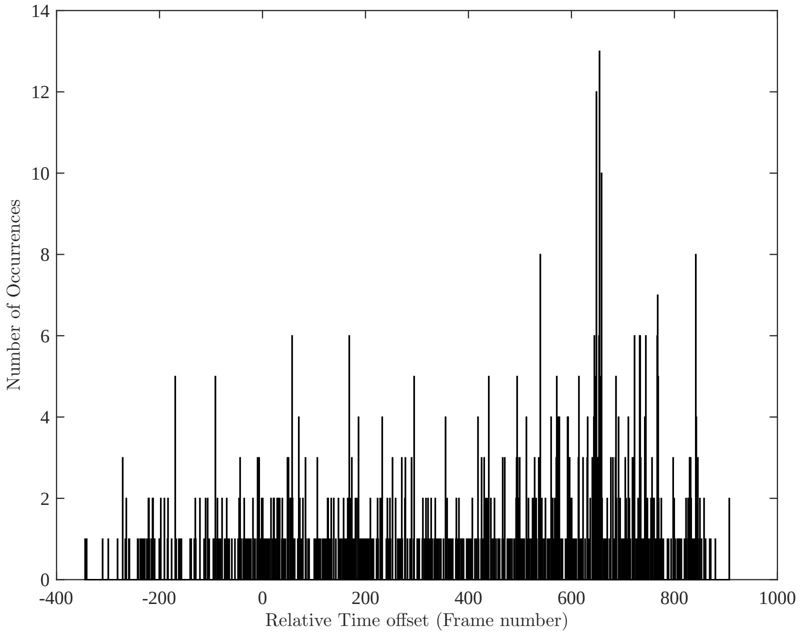


(b) Histogram of their relative time offsets.

Figure 2.3: Example with the confirmed presence of an excerpt of a target music in the soundtrack signal. Note there is a cluster of points forming a diagonal line (inclination of 45°) in the scatter plot. Those points were originated by landmark matches with the same relative time offset of 164 frames forward (linear coefficient of the line). The red points are used for synchronisation.



(a) Scatter plot with the music and soundtrack time offsets.



(b) Histogram of their relative time offsets.

Figure 2.4: Example with the confirmed absence of an excerpt of a target music in the soundtrack signal. Note there is no cluster of points forming a diagonal line in the scatter plot, neither a high number of the same relative time offsets in the histogram.

2.3 Simulations for Quick-Search Method

This section details the simulations the author implemented to test the performance of the method. In the simulations, musical excerpts are created from references music-track signals $m_i(n)$ according to

$$m_i(n)u(n-\delta_i) - m_i(n)u(n-\delta_i'), \tag{2.2}$$

where $u(n)$ is the unit-step function [37] and parameters δ_i and δ_i' represent the beginning and the end samples of the excerpt with respect to $m_i(n)$. They are added to a dialogue signal $d(n)$ starting at pre-determined time-instants Δ_i. The sampling frequency is 48000 Hz.

2.3.1 Simulation 01

In this simulation, 10-second duration excerpt signals were taken from 3 different songs, scaled by a constant 0.2 amplitude gain and added to a generic dialogue-sfx signal at the exact instants of 2, 14 and 26 seconds; then, the full quick-search method was performed in order to try to detect and synchronise each excerpt present in the mixture with their reference-signal counterparts.

Table 2.1: Results of the quick-search method for Simulation 01.

—	Δ_i	δ_i	δ'_i	EELD
Excerpt 1:	96257 (96000)	257 (0)	476159 (479999)	99.15 %
Excerpt 2:	673024 (672000)	1024 (0)	468736 (479999)	97.44 %
Excerpt 3:	1248257 (1248000)	257 (0)	458751 (479999)	95.52 %

Table 2.2: Scores of the separation for Simulation 01 in dB.

—	SDR	SIR	SAR
Excerpt 1	32.31	50.04	32.39
Excerpt 2	17.57	29.12	17.90
Excerpt 3	23.28	39.61	23.38

It should be noted that, in this simulation, the idea was only to check if the method would be able to find the correct initial and final time-samples where the excerpt appears in the mixture. Hence, it was used the exact excerpt with 10-second duration as the input for the quick-search method. The results of the detection are shown in Table 2.1; the values in parenthesis are the true values for each parameter, and the last column shows the Effective Excerpt Length Detected (EELD), which can be defined as the length of the detected excerpt divided by the original excerpt length, i.e.,

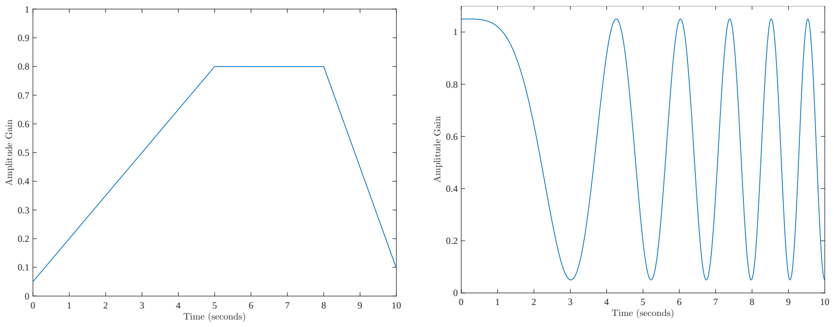
$$\mathrm{EELD}_i = \frac{\delta'_i - \delta_i + 1}{480000}. \tag{2.3}$$

Since the gain was previously known, it was possible to use it to scale the detected segment of the excerpt and remove it from the mixture, trying to retrieve the original dialogue signal. Table 2.2 shows the scores of each separated signal.

Note that, despite not being able to perfectly detect the values of Δ_i, δ_i and δ'_i, the algorithm detects almost the entire excerpts in the exact position they appear in the mixture. For more than 95 % of time where there were the concurrent presence of the dialogue and the musical excerpts, the algorithm was able to properly synchronise the reference signals with the mixture. There were only small errors at the beginning and at the end parts of each segment with some (less than 16300) samples undetected. If considering we are working with 48000 samples per second, those values represent less than half a second of error, which leaves the separated signal only with a slightly short time-localised pulse at the borders.

2.3.2 Simulation 02

The only difference from the last simulation is that, in this case, time-variable gains, depicted in Figure 2.5, were used in the excerpts before adding them to the dialogue-sfx at the same time positions.



(a) Piecewise-linear gain.

(b) Chirp-function gain.

Figure 2.5: Variable gains applied to excerpts in Simulation 02.

First, the gain in Figure 2.5a was applied to the 3 excerpts to simulate fade-in and fade-out effects that are commonly found in real-life music-track signals; the results of the quick-search method appear on Table 2.3. Then, another simulation was performed using a chirp function shown in Figure 2.5b, as the amplitude gain for the excerpts, so that we could check the robustness of the algorithm with respect to a highly varying gain. Once more, the exact excerpt was used as an input for the method. The idea here is to check if the algorithm would lose performance in the presence of those variable gains. After the detection had been finished, the synchronised segment of the excerpt was removed from the mixture using the original gain function and the quality assessment of the retrieved dialogue signals are shown on Table 2.5.

Table 2.3: Results of the quick-search method for linear gain curve in Simulation 02.

—	Δ_i	δ_i	δ'_i	EELD
Excerpt 1:	96257 (96000)	257 (0)	474111 (479999)	98.72 %
Excerpt 2:	673024 (672000)	1024 (0)	467712 (479999)	97.23 %
Excerpt 3:	1248257 (1248000)	257 (0)	458751 (479999)	95.52 %

Table 2.4: Results of the quick-search method for chirp gain curve in Simulation 02.

—	Δ_i	δ_i	δ'_i	EELD
Excerpt 1:	96257 (96000)	257 (0)	474112 (479999)	98.72 %
Excerpt 2:	672000 (672000)	0 (0)	470085 (479999)	98.08 %
Excerpt 3:	1248257 (1248000)	257 (0)	456704 (479999)	95.09 %

Table 2.5: Scores of the separation for Simulation 02 in dB.

—	Piecewise-Linear Gain			Chirp Gain		
—	SDR	SIR	SAR	SDR	SIR	SAR
Excerpt 1	34.45	62.48	34.46	24.19	45.89	24.22
Excerpt 2	19.71	41.64	19.74	17.92	37.99	17.97
Excerpt 3	23.70	49.19	23.71	10.07	23.47	10.30

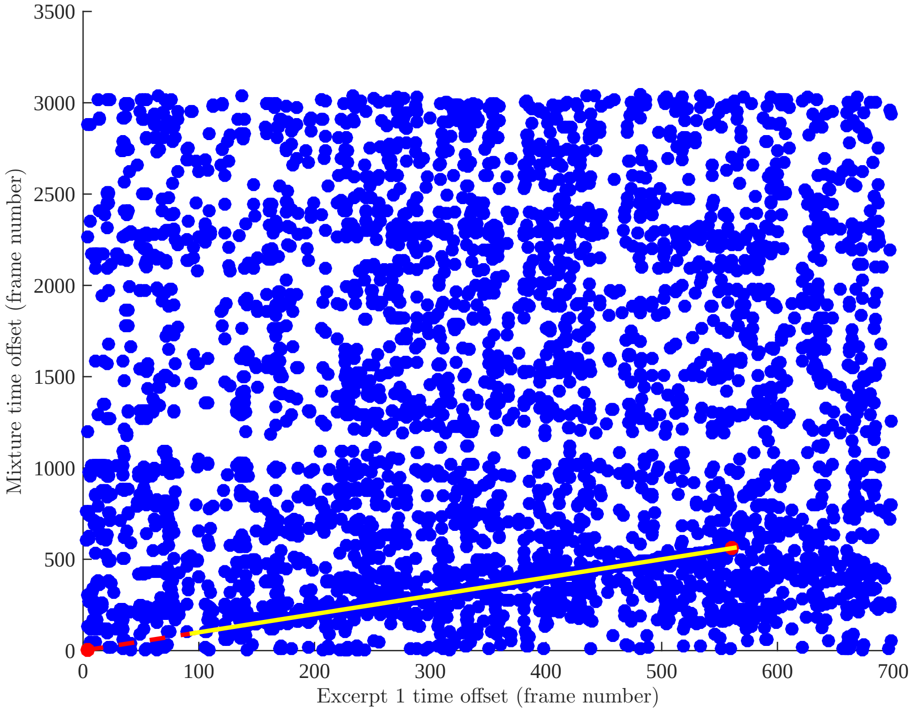
2.3.3 Simulation 03

This simulation has the objective of verifying if the quick-search method will be able to detect the musical excerpts with correct length when using musical segments with higher durations as reference for the algorithm. The same excerpts with 10 seconds of duration are now searched using 15-and 30-second long musical segments of the same reference music and containing all the samples from the original excerpts in themselves. The time variable gain on Figure 2.5a was once again applied. In Figures 2.6 and 2.7 it is possible to check the results.

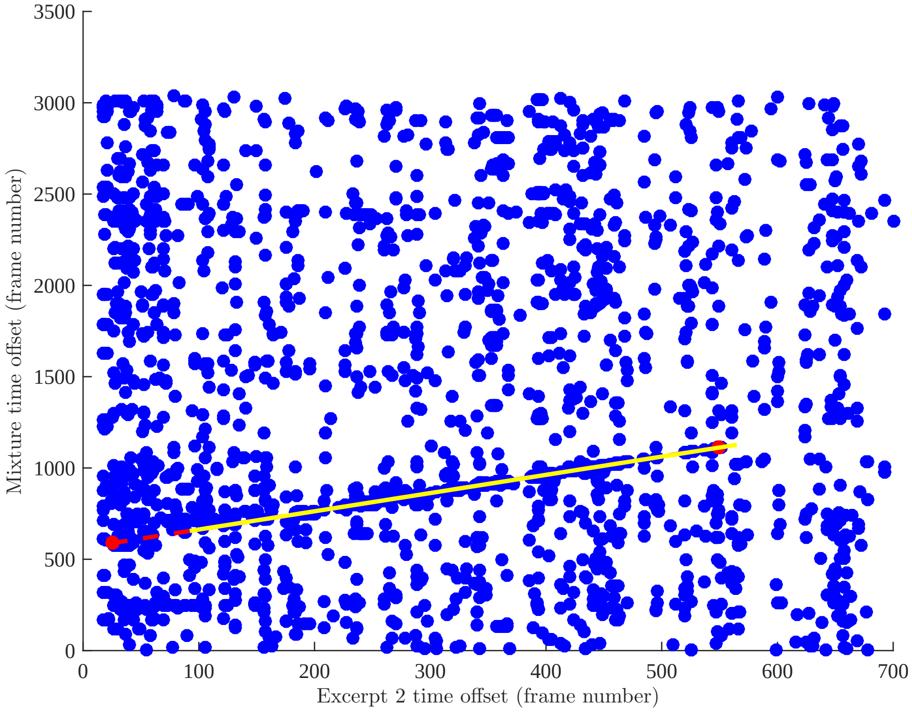
Observe the presence of a red dashed line with two big circles at each edge. This line represents the part of the musical segment that the algorithm considered present in the mixture, whereas the yellow line represents the correct length and location of the utilised musical excerpts. It can be concluded that the algorithm was able to find the correct 45°-inclination line, which means that the method could effectively compute the relative time offset (linear coefficient of the line) between the signals. However, the implemented algorithm detected many 'false positives' in the mixture, specially when using larger segments.

The reason for that is the way the algorithm was implemented. It starts by correctly detecting the line with inclination of 45° that passes by the highest number of points in the scatter plot. Later, it gets the first and last points on the line and uses them to define the end-points of the segment that is effectively present in the mixture and forms the red line. A problem occurs when such end-points are originated from landmarks coming from other random points of the spectrograms of the signals that unfortunately had this same particular relative time offset. The correct approach would be to get the points that are not only on top of the line, but also close enough to the cluster of points that are actually forming the line themselves.

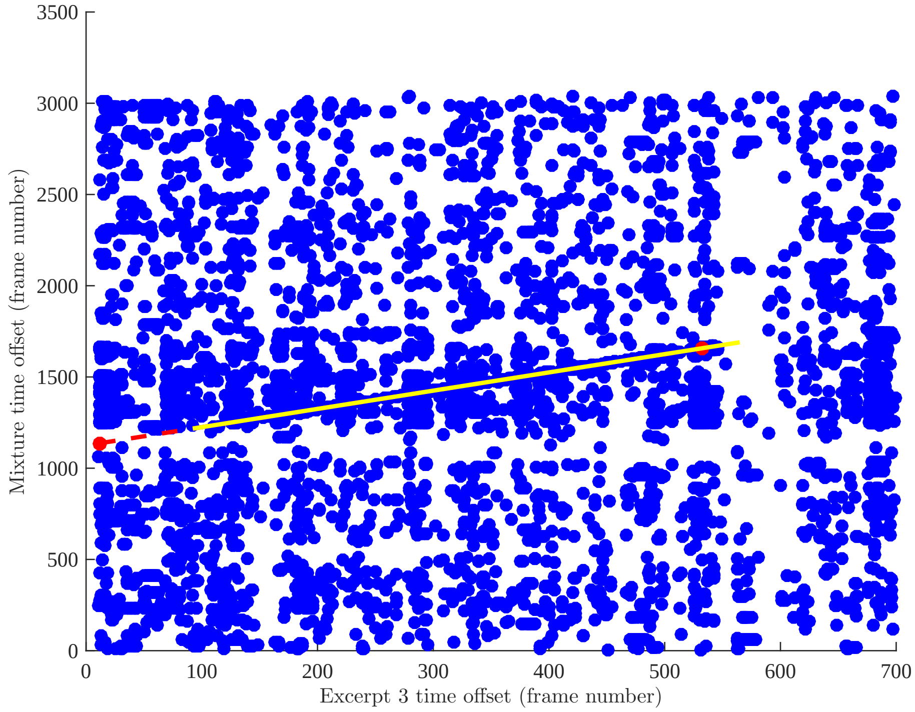
Nonetheless, this type of error regarding the length of the segment is not too alarming. Since the next step will be the estimation of the gain that was applied to the excerpt, is it possible to fix its incorrect length if we manage to estimate an approximately zero gain for the samples next to its borders.



(a) Excerpt 1 contained in a 15-second audio segment.

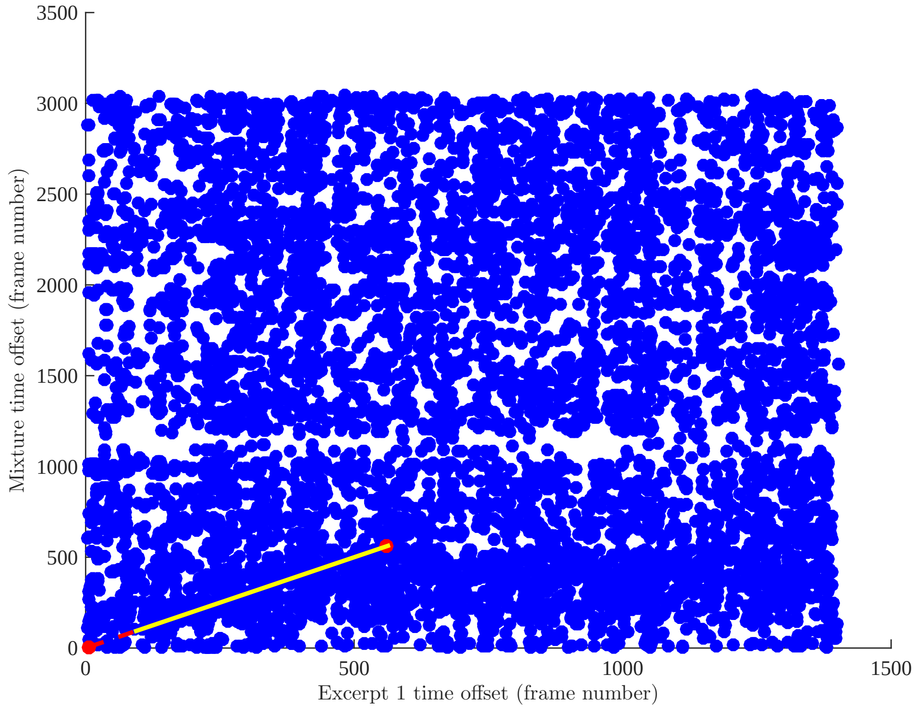


(b) Excerpt 2 contained in a 15-second audio segment.

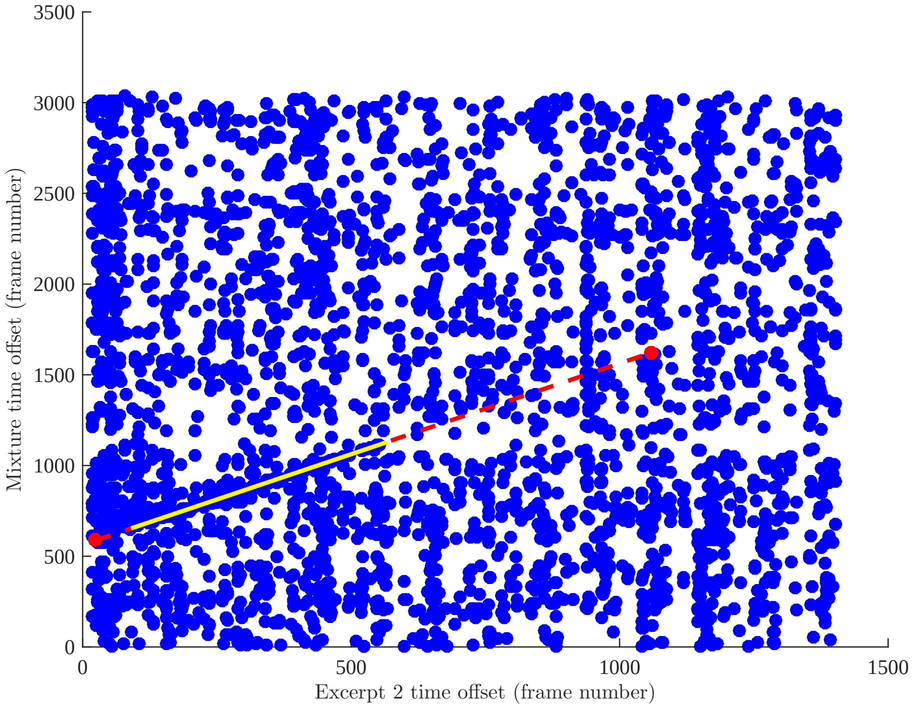


(c) Excerpt 3 contained in a 15-second audio segment.

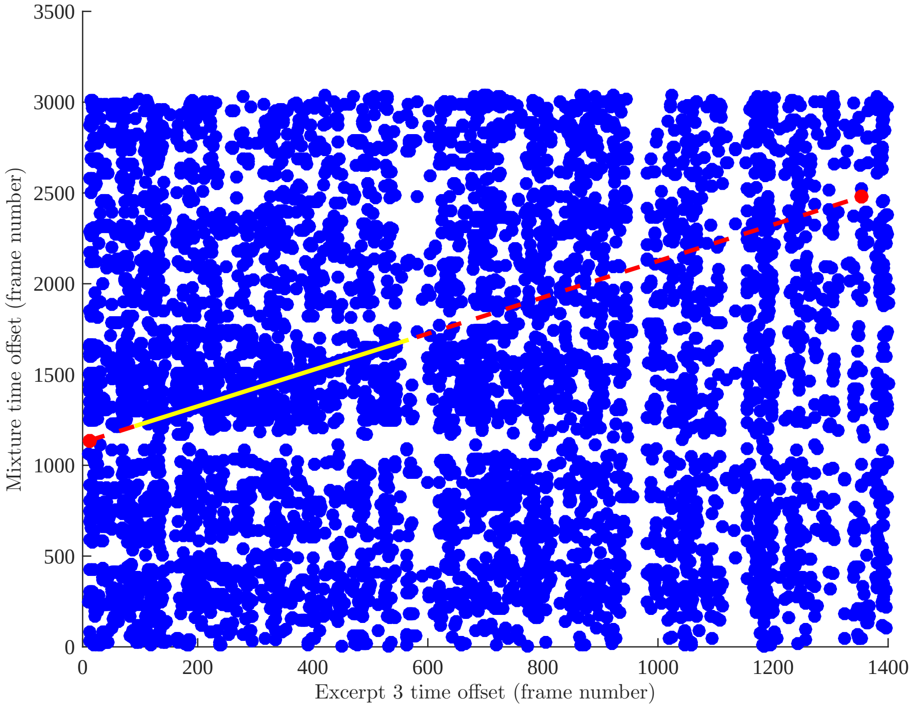
Figure 2.6: Results of synchronisation using excerpts with 15-second duration.



(a) Excerpt 1 contained in a 30-second audio segment.



(b) Excerpt 2 contained in a 30-second audio segment.



(c) Excerpt 3 contained in a 30-second audio segment.

Figure 2.7: Results of synchronisation using excerpts with 30-second duration.

Chapter 3

Time-Variable Gain Estimation

When a musical track is used in a film, TV series or soap opera, for example, it usually appears with variable gain throughout the video scene. For instance, it is common to gradually increase the volume of a music piece (fade-in), or to reduce its volume from a certain level to silence (fade-out). Both types of fade are useful to make the beginning and the end of each song excerpt smoother, without any prominent glitch [38]. Also, they can be used to soften the attack sounds of certain percussion instruments [38] or to lower/raise the music volume depending on whether or not there is the concurrent presence of a character's voice in the scene.

Therefore, it is clear that during a music track extraction procedure we should take into account the possible occurrence of a time-variable gain on the music excerpts. The gain curve estimation procedure is explained in this chapter, including detailed explanations on how to reproduce the algorithm as well as some results of the simulations.

3.1 Fundamentals

The basic idea behind the time-variable gain estimation algorithm is to use a template matching technique. This technique is widely known in the signal processing theory, and it is mostly used in computer vision [39], but can also be used in source separation applications such as long pulse removal from old music recordings for audio restoration [40].

In the context of audio restoration, it is possible to estimate an average shape for the long noise pulses that are present in a degraded recording and use it as a template for the undesired information. With such template available, it is also possible to find the constant amplitude gain that 'matches' the music recording, allowing its removal from the signal.

Correspondingly, from the perspective of this project, we can utilise an excerpt from the undesired music track obtainable from officially published CDs related to the audiovisual work we are working on and use it as the reference template. After that, we can use a similar process to estimate the gain that best matches the soundtrack signal.

In order to consider fade effects and time-variable gains, the implemented algorithm divides the template into short segments (odd-length) with superposition and estimates a gain that best fits it to the soundtrack. Afterwards, each gain is associated with the time sample related to the centre of its respective segment. We can achieve a sample-by-sample variable gain by just applying a linear interpolation to the other time samples.

3.2 Mathematical Analysis

Consider we have access to a clean musical track signal $m(n)$; we can generate a template signal $t(n)$ to be searched in the soundtrack signal $s(n)$ by just cutting a limited T-length signal from $m(n)$, i.e.,

$$t(n) = \begin{cases} m(n-\delta), & \forall n \in \mathbb{Z} : \{0 \leq n < T\}, \\ 0, & \forall n \in \mathbb{Z} : \{n < 0\} \cup \{n \geq T\}, \end{cases} \tag{3.1}$$

where δ is just a symbol to represent that the first sample of the template does not have to be the first sample of the music-track, it might have been originated from any sample of $m(n)$.

The problem we are dealing with is trying to get a good estimation for an also T-length time-variable gain $\alpha(n)$ that satisfies the following equation:

$$s(n) = d(n) + \alpha(n-\Delta)t(n-\Delta), \tag{3.2}$$

where $d(n)$ is the dialogue-sfx signal and Δ represents the time sample where the full template begins in the mixture. If we compare Equation (3.2) to Equation (1.1), it is possible to conclude that we are analysing a segment of the soundtrack signal where $\mu(n)$ is a modified version (with a time-variable gain) of a segment of the reference track $m(n)$ we have available.

It is valuable to note that neither δ nor Δ will have an important role in the present analysis. As we are modeling just the standalone version of the time-variable gain estimator, we can assume that the signals in Equation (3.2) have already been synchronised. In other words, we can consider that δ and Δ are determined a priori either by estimation procedures like those discussed on Chapter 2 or by any other method. Thus, rewriting the problem using the synchronised signals, we get

$$s_T(n) = d_T(n) + \alpha(n)t(n), \tag{3.3}$$

where $s_T(n)$ and $d_T(n)$ are the T-length trimmed soundtrack and dialogue-sfx signals after having been properly aligned to the template and $n \in \{0, 1, 2, \ldots, T-1\}$.

Now, using small L-length rectangular windows on $t(n)$ (with overlap), we can form shorter signals $t_{L_i}(n)$, where i is just an index for representing each segment (frame) of $t(n)$ created and $n \in \{0, 1, 2, \ldots, L-1\}$. Adopting this procedure, we can consider the gain will remain constant for the duration of each segment if we use $L \ll T$.

After the segmentation of every signal involved in Equation (3.3) and their updated realignment, we can then observe that, for the analysis of a generic L-length template segment $t_L(n)$, we have

$$s_L(n) = d_L(n) + \alpha t_L(n), \tag{3.4}$$

where $s_L(n)$ and $d_L(n)$ are the realigned L-length versions of the soundtrack and dialogue-sfx signals, α is the constant gain used in this particular frame of the template and $n \in \{0, 1, 2, \ldots, L-1\}$.

⋆ Note that we have omitted the index i from Equation (3.4) for the sake of notation simplicity. However, this value is important for the resynchronisation procedure. Each segment $t_{L_i}(n)$ should be aligned with its respective $s_{L_i}(n)$ and $d_{L_i}(n)$. Since we already have a previously estimated value for Δ, the realignment can be easily done by just adding different offsets to it, depending on the position of the rectangular window that generated $t_{L_i}(n)$.

At this point, we just need a good way to estimate a value for α that acceptably fits to our data. This can be achieved by using a simple template matching technique as the one described in [40, 41]. Despite this procedure have been primarily applied to the removal of long noise pulses from degraded audio signals, it can be tailored to our case.

The first step is to compute the statistical cross-correlation [11] between $s_L(n)$ and $t_L(n)$ at zero time lag:

$$\begin{aligned} R_{s_L t_L}(n,n) &= \mathbb{E}\Big\{ \, [d_L(n) + \alpha t_L(n)] \, [t_L(n)] \, \Big\} = \mathbb{E}\Big\{ d_L(n) t_L(n) \Big\} + \alpha \mathbb{E}\Big\{ t_L(n) t_L(n) \Big\} \\ &= R_{d_L t_L}(n,n) + \alpha \; R_{t_L t_L}(n,n), \end{aligned} \tag{3.5}$$

where $\mathbb{E}\{\bullet\}$ represents the statistical expected value [11] of the argument, $R_{d_L t_L}(n,n)$ is the statistical cross-correlation of $d_L(n)$ and $t_L(n)$ at zero time lag and $R_{t_L t_L}(n,n)$

is the autocorrelation of $t_L(n)$ also at zero time lag. Note that the dialogue-sfx signal, which is essentially a speech signal, does not relate to the music-track signal. We may assume that $d_L(n)$ is statistically uncorrelated to $t_L(n)$. Consequently, it is possible to rewrite $R_{d_L t_L}(n,n)$ as

$$R_{d_L t_L}(n,n) = \mathbb{E}\Big\{d_L(n)t_L(n)\Big\} = \mathbb{E}\Big\{d_L(n)\Big\}\mathbb{E}\Big\{t_L(n)\Big\} = \overline{d_L} \cdot \overline{t_L}, \tag{3.6}$$

where $\overline{d_L}$ is the expected value of $d_L(n)$ and $\overline{t_L}$ is the expected value of $t_L(n)$.

Using the results from Equation (3.6), we are able to simplify Equation (3.5) and find

$$R_{s_L t_L}(n,n) = \overline{d_L} \cdot \overline{t_L} + \alpha\ R_{t_L t_L}(n,n). \tag{3.7}$$

An important fact to realise is that we only have access to a single sample of each stochastic sequence involved in Equation (3.7). Then, applying the principle of the ergodicity [42], let us assume we can approximate each statistical correlation $R_{\bullet\bullet}(n,n)$ by their temporal counterparts $\mathcal{R}_{\bullet\bullet}(0)$ and each expected value $\mathbb{E}\{\bullet\}$ by the respective average time-domain value $\mathcal{A}\{\bullet\}$.

Therefore, it is possible to approximate the terms of Equation (3.7) as

$$\mathcal{R}_{\mathbf{s}_L \mathbf{t}_L}(0) = \mathcal{R}_{\mathbf{d}_L \mathbf{t}_L}(0) + \alpha \mathcal{R}_{\mathbf{t}_L \mathbf{t}_L}(0) = \mathcal{A}\Big\{\mathbf{d}_L\Big\}\mathcal{A}\Big\{\mathbf{t}_L\Big\} + \alpha \mathcal{R}_{\mathbf{t}_L \mathbf{t}_L}(0), \tag{3.8}$$

$$\frac{\mathbf{t}_L^\top \mathbf{s}_L}{L} = \frac{\texttt{sum}(\mathbf{d}_L)\texttt{sum}(\mathbf{t}_L)}{L^2} + \alpha \frac{\mathbf{t}_L^\top \mathbf{t}_L}{L}, \tag{3.9}$$

where $\mathbf{s}_L = [s_L(0)\ ,\ s_L(1)\ ,\ \dots\ ,\ s_L(L-1)]^\top$, $\mathbf{d}_L = [d_L(0)\ ,\ d_L(1)\ ,\ \dots\ ,\ d_L(L-1)]^\top$, $\mathbf{t}_L = [t_L(0)\ ,\ t_L(1)\ ,\ \dots\ ,\ t_L(L-1)]^\top$ and the function `sum` computes the sum of the elements in its arguments.

Despite being uncorrelated with each other, the signals $d_L(n)$ and $t_L(n)$ are not necessarily either statistically or temporally orthogonal. Not only each expected value might be different from zero, but also we have no guarantee that at least one average time-domain value will be zero. However, let us take a closer look at their temporal cross-correlation:

$$\mathcal{R}_{\mathbf{d}_L \mathbf{t}_L}(0) = \frac{\texttt{sum}(\mathbf{d}_L) \cdot \texttt{sum}(\mathbf{t}_L)}{L^2}. \tag{3.10}$$

Considering that both signals have an oscillatory nature and that they have been originated from an excerpt of a `.wav` file whose sample values vary from -1 to 1, it is possible to assume that the value of $\texttt{sum}(\mathbf{d}_L) \cdot \texttt{sum}(\mathbf{t}_L)$ (numerator of Equation (3.10)) will be much smaller than the value of L^2 (denominator of Equation (3.10)), which has an order of magnitude of 10^4. This assumption is even more acceptable if we use higher values for the window length L.

Summarizing, we can consider that

$$\mathcal{R}_{\mathbf{d}_L\mathbf{t}_L}(0) \approx 0. \tag{3.11}$$

On the other hand, if the value of L becomes too large, we might come into another problem. The approximation we did before, i.e., the supposition that the variable gain $\alpha(n)$ can be considered constant during the whole duration L of a frame may become unrealistic.

It is therefore clear that there is a trade-off between increasing and decreasing the value of L. In practice, we will be using segments with tens to hundreds of millisecond of duration, say from 20 ms to 200 ms, which would give a value of L around 960 to 9600 in the simulations of Section 3.3, as high-quality signals with sampling frequency of 48000 Hz have been used.

Finally, it is possible to conclude that an estimate $\hat{\alpha}$ for α could be obtained by computing

$$\hat{\alpha} = \frac{\mathcal{R}_{\mathbf{s}_L\mathbf{t}_L}(0)}{\mathcal{R}_{\mathbf{t}_L\mathbf{t}_L}(0)} = \frac{\mathbf{t}_L^\top \mathbf{s}_L}{\mathbf{t}_L^\top \mathbf{t}_L}. \tag{3.12}$$

3.3 Gain Estimation Simulations and Results

In order to check the performance of the algorithm for estimation of time-variable gain, some Matlab simulations were performed using different values for L and for the overlap between each analysed frame.

Some artificial soundtracks signals were created by mixing a generic dialogue signal with an excerpt of a high-quality musical recording. The sampling rate is 48000 Hz and pre-defined time-variable gain curves were used. Also, it is worth noting that the dialogue signal had a total duration of approximately 31 seconds, whereas the musical excerpt had only 17 seconds of duration, starting at 16 s. Thus, the final mixture will always have a total of $16 + 17 = 33$ seconds and its last 2 seconds will have only the single presence of the music-track, with no dialogue signal. Therefore, we can effectively compare the performance of the gain estimation algorithm in the presence and in the absence of dialogue.

The figure of merit utilised to measure the accuracy of the estimation algorithm is the Mean Absolute Percentage Error (MAPE). The closer to 0, which means that 100 % of the gain samples have been perfectly estimated, the better.

3.3.1 Simulation 01

In this simulation, the curve of the reference gain applied to the music-track excerpt can be divided into three parts:

- A linear gain from 0.2 to 0.8 (Fade-in), during the first 9 seconds;
- Constant $\alpha(n) = 0.8$ during the next 4 seconds;
- Linear gain from 0.8 to 0.05 (Fade-out), during the last 4 seconds.

This simulation is intended to give an insight on how well the algorithm performs when dealing with simple linear fades and constant gains in the soundtrack signals. The results using smaller values of L (25 ms and 50 ms) are shown in Figure 3.1, while the results using larger values (100 ms and 200 ms) for the window length are illustrated in Figure 3.2.

We may conclude that the values of 25 ms and 50 ms for L are not recommended for the algorithm. Using those values, the final estimated gain curve had many undesired high frequency components. Even during the constant part, there were many errors in the gain estimation. This fact is reflected by higher values for MAPE in those cases.

By using higher values for L it was possible to obtain much better results, with a really small value of MAPE of 3.05 % using $L = 200$ ms and an overlap of 25 % against at best 7.89 % using $L = 50$ ms with an overlap of 25 %.

It is important to note that, as expected, in the last 2 seconds of the mixture, where there was only the presence of the music-track, the results were much better; not matter the size of the window length neither the overlap value, the estimated gain curve was able to follow the reference signal.

Regarding the performance versus overlap size, we can notice that in this simulation there were no fast variations on the time-variable gain. Only small linear changes and constant values were used. This fact did not cause the necessity of using larger overlaps to 'predict' possible fast variations. Remember that a small overlap also permits the algorithm to make linear interpolations with more points between two estimated gain values. As the largest part of the reference gain consisted of linear functions of time, the usage of smaller overlap sizes was able to generate better results if compared to larger values.

3.3.2 Simulation 02

In this simulation, the reference gain curve was a linear chirp, applied along the whole duration of the music-track excerpt. The point here is to analyse the performance of the algorithm when there are oscillations in the volume of the music-track present in the mixture. Despite not being easy to find a gain curve with many oscillations in a real world soundtrack signal, a simulation like this is important to give a better idea on how well a constant window length and overlap value behave under slow and fast variations of a time-variable gain. The results are spread between two figures, with Figure 3.3 showing the behaviour of the system using small values (25 ms and 50 ms) of L and with Figure 3.4 illustrating the results for large values (100 ms and 200 ms) of L.

The result is very similar to that of the simulation in Subsection 3.3.1. The utilisation of smaller window lengths generated values of MAPE greater than 5.97 %, while much better results could be obtained using $L = 100$ ms or $L = 200$ ms instead. For instance, a MAPE = 4.22 % can be achieved using $L = 100$ and overlap = 25 %.

Overall, the estimated gain curve was able to reproduce the oscillations from the reference chirp without any significant problem.

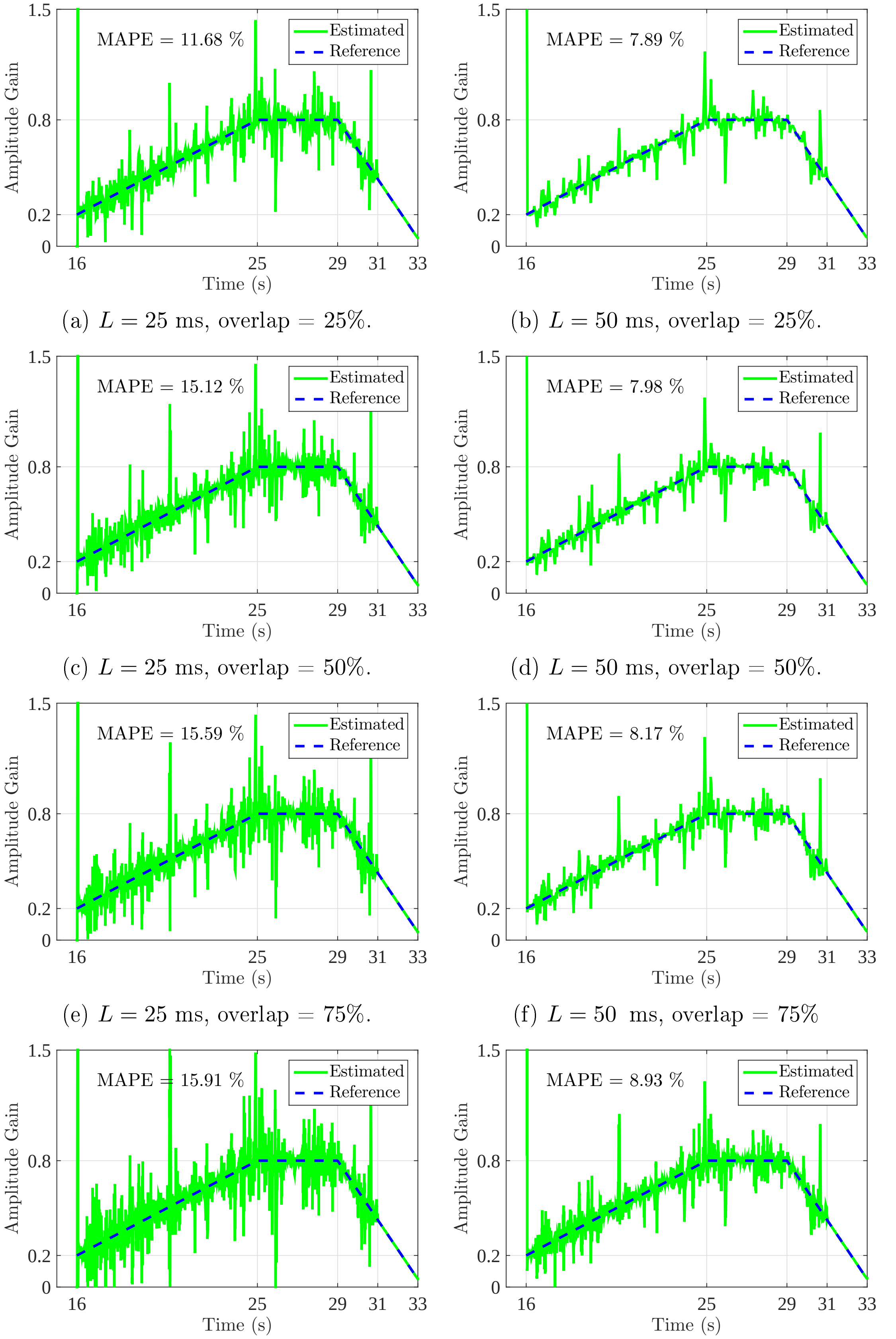
3.3.3 Simulation 03

This simulation had the objective of testing the performance of the algorithm under large discontinuities on the time-variable gain curve. The results are shown in Figures 3.5 and 3.6.

Note there was a period of 3 seconds where the gain applied to the music-track was null. This is an important test to check if the estimator would manage to estimate this null value. When processing non-artificial soundtracks, for example, we cannot ensure that the searched template is fully present in the mixture: the soundtrack may include only part of it. Hence, the algorithm is expected to attribute zero gain value to the missing parts.

As we can see in both figures, the results are in line with the past simulations. Higher values of window length achieved better MAPE values. The sudden steps in the time-variable gain were not a problem for the algorithm, which kept its performance even during the higher oscillation periods. Another interesting fact is that we can conclude that a zero gain can also be estimated if part of the template is missing in the mixture.

The best results this time occured for a larger value for the overlap if compared to the past simulations (50 % overlap against 25 % overlap in the others). However, the difference in the results was not too significant to consider the new value a better choice for this parameter in the general case. The reason for this result is probably that with a larger overlap between windows, we have more points to correctly estimate the discontinuities on the time-variable gain.



(a) $L = 25$ ms, overlap = 25%.

(b) $L = 50$ ms, overlap = 25%.

(c) $L = 25$ ms, overlap = 50%.

(d) $L = 50$ ms, overlap = 50%.

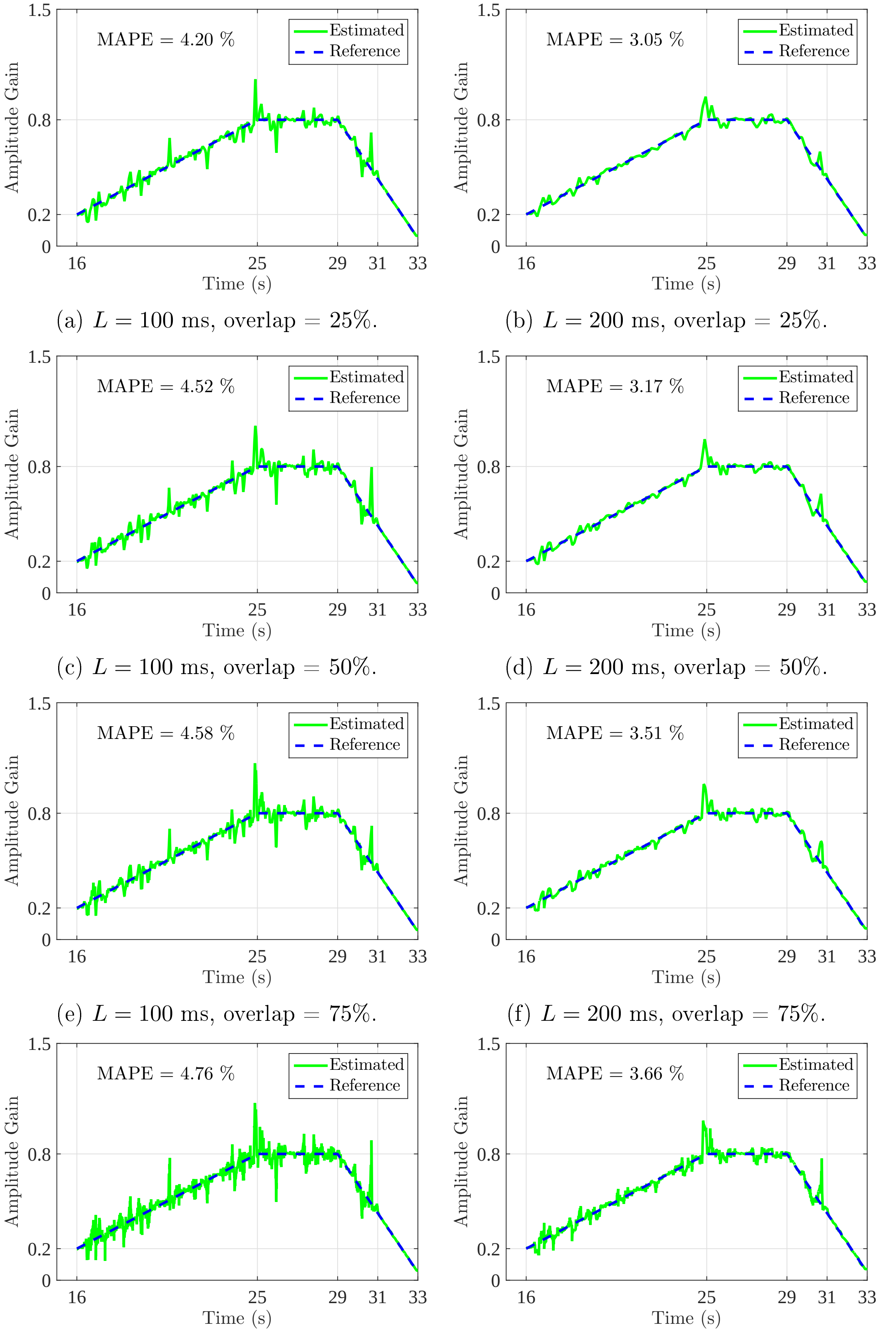
(e) $L = 25$ ms, overlap = 75%.

(f) $L = 50$ ms, overlap = 75%

(g) $L = 25$ ms, overlap = $L - 1$.

(h) $L = 50$ ms, overlap = $L - 1$.

Figure 3.1: Results of the time-variable gain estimation for $L = 25$ ms and $L = 50$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts.



(a) $L = 100$ ms, overlap $= 25\%$.

(b) $L = 200$ ms, overlap $= 25\%$.

(c) $L = 100$ ms, overlap $= 50\%$.

(d) $L = 200$ ms, overlap $= 50\%$.

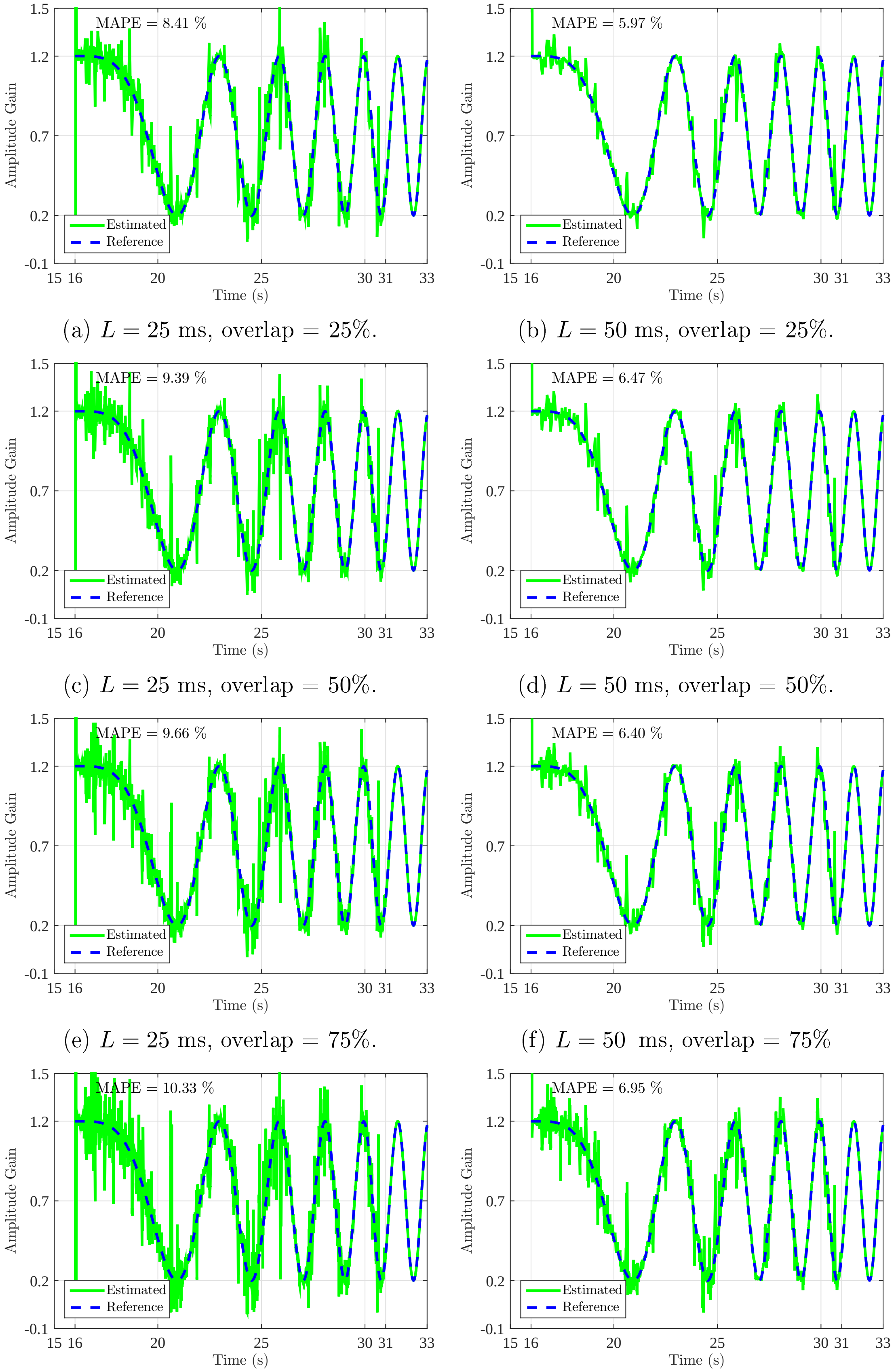
(e) $L = 100$ ms, overlap $= 75\%$.

(f) $L = 200$ ms, overlap $= 75\%$.

(g) $L = 100$ ms, overlap $= L - 1$.

(h) $L = 200$ ms, overlap $= L - 1$.

Figure 3.2: Results of the time-variable gain estimation for $L = 100$ ms and $L = 200$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts.



(a) $L = 25$ ms, overlap = 25%.

(b) $L = 50$ ms, overlap = 25%.

(c) $L = 25$ ms, overlap = 50%.

(d) $L = 50$ ms, overlap = 50%.

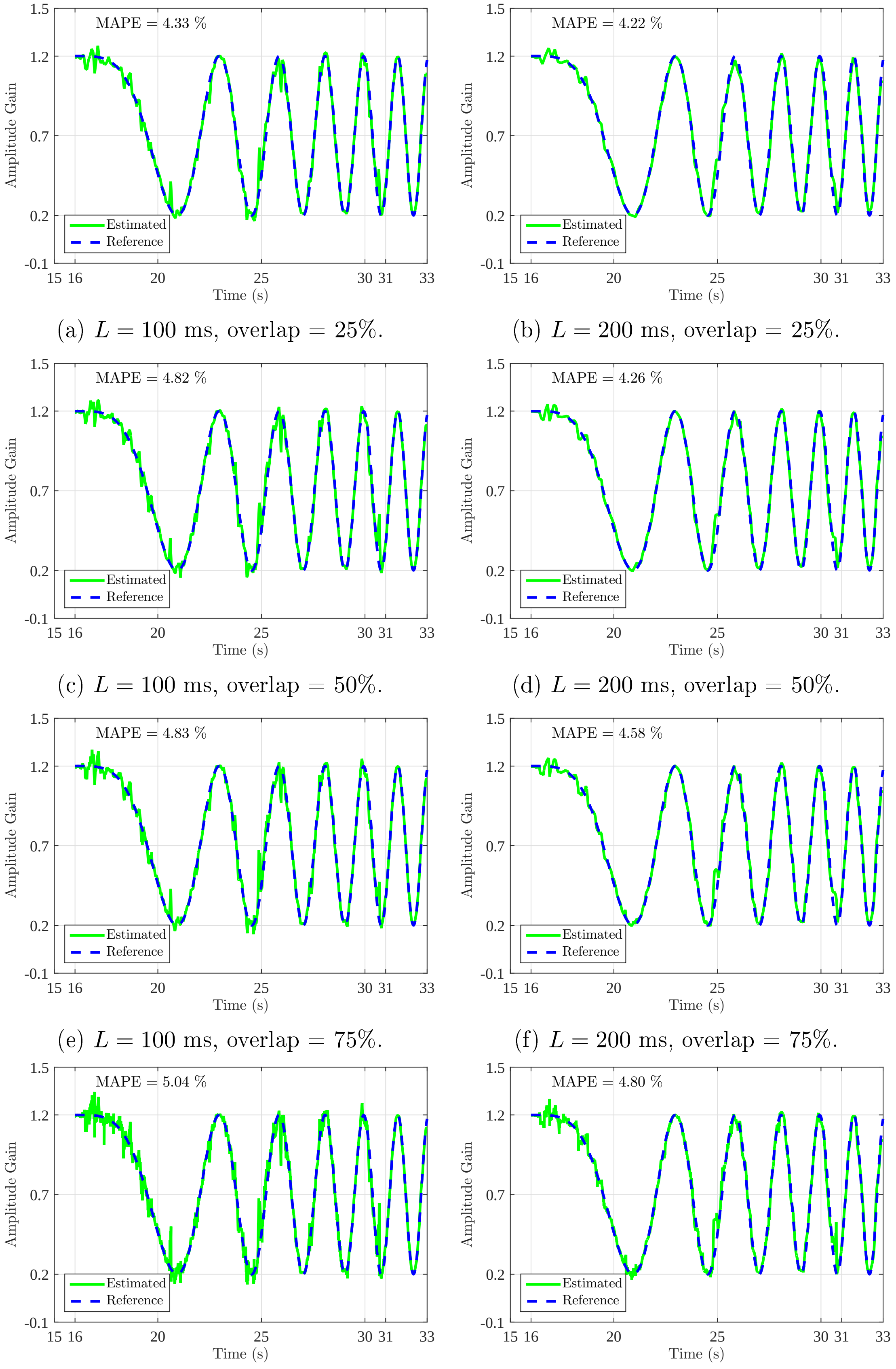
(e) $L = 25$ ms, overlap = 75%.

(f) $L = 50$ ms, overlap = 75%

(g) $L = 25$ ms, overlap = $L - 1$.

(h) $L = 50$ ms, overlap = $L - 1$.

Figure 3.3: Results of the time-variable gain estimation for $L = 25$ ms and $L = 50$ ms using a chirp as the reference gain curve.



(a) $L = 100$ ms, overlap $=$ 25%.

(b) $L = 200$ ms, overlap $=$ 25%.

(c) $L = 100$ ms, overlap $=$ 50%.

(d) $L = 200$ ms, overlap $=$ 50%.

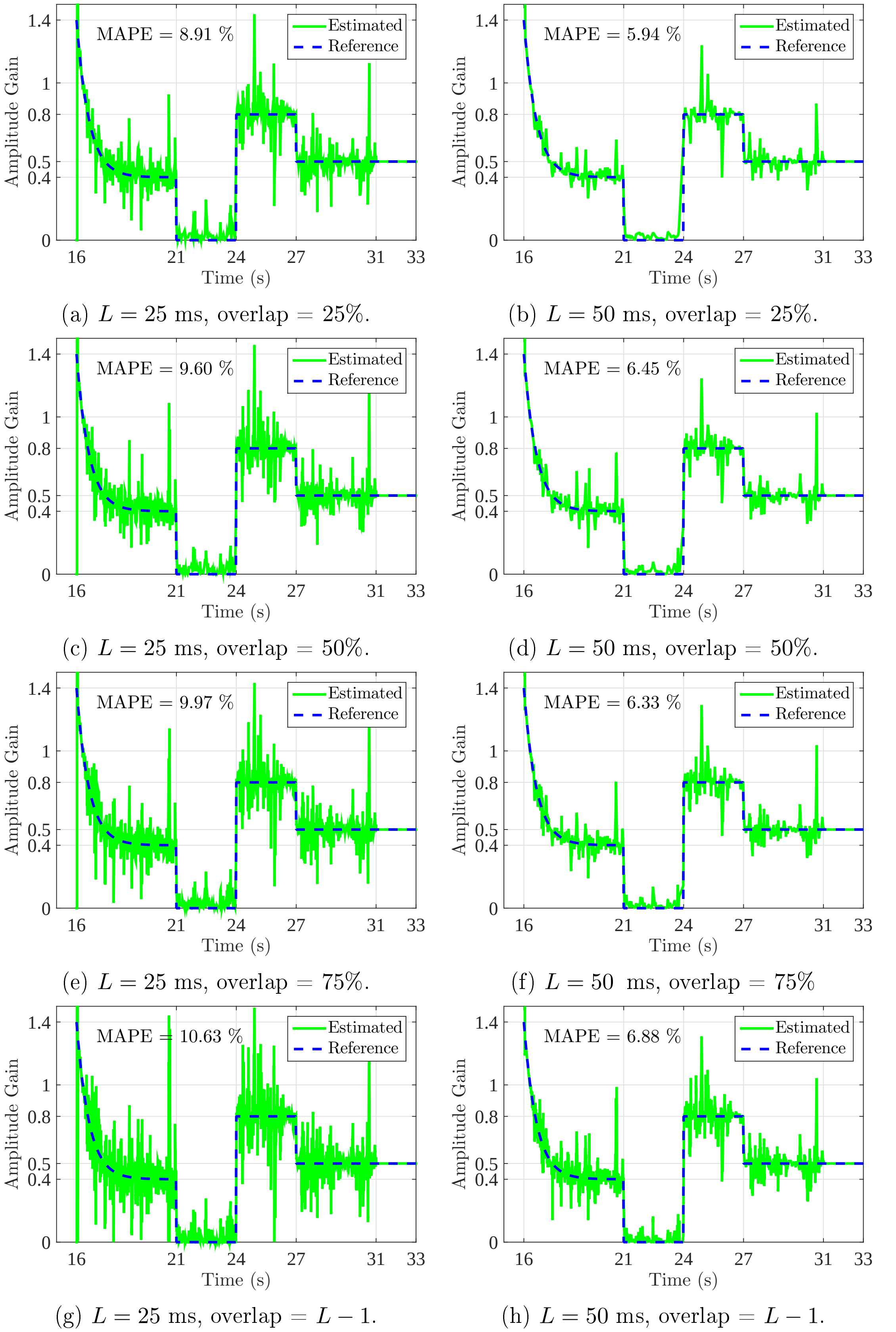
(e) $L = 100$ ms, overlap $=$ 75%.

(f) $L = 200$ ms, overlap $=$ 75%.

(g) $L = 100$ ms, overlap $= L - 1$.

(h) $L = 200$ ms, overlap $= L - 1$.

Figure 3.4: Results of the time-variable gain estimation for $L = 100$ ms and $L = 200$ ms using a chirp as the reference gain curve.



(a) $L = 25$ ms, overlap = 25%.

(b) $L = 50$ ms, overlap = 25%.

(c) $L = 25$ ms, overlap = 50%.

(d) $L = 50$ ms, overlap = 50%.

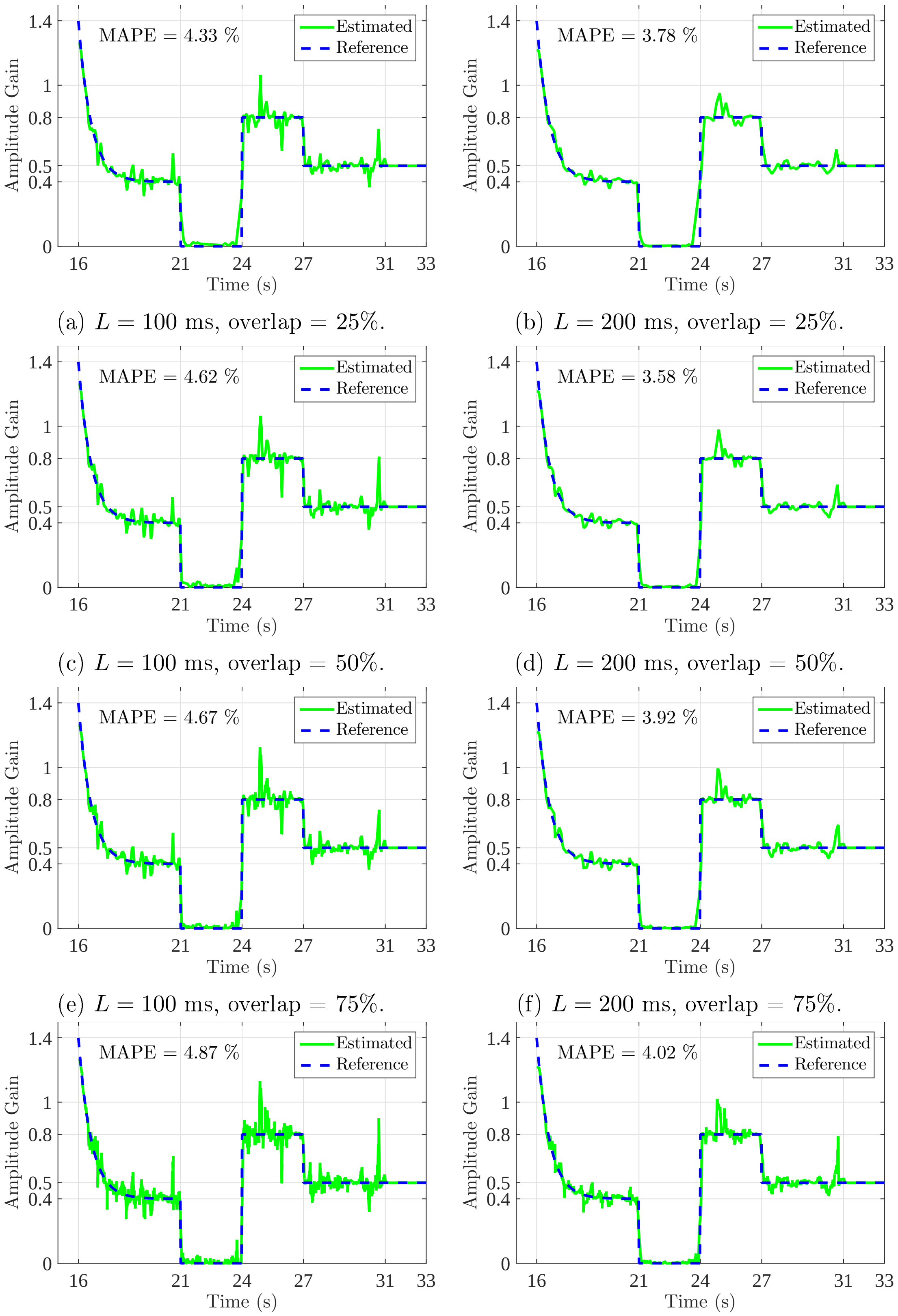
(e) $L = 25$ ms, overlap = 75%.

(f) $L = 50$ ms, overlap = 75%

(g) $L = 25$ ms, overlap = $L - 1$.

(h) $L = 50$ ms, overlap = $L - 1$.

Figure 3.5: Results of the time-variable gain estimation for $L = 25$ ms and $L = 50$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts.



(a) $L = 100$ ms, overlap = 25%.

(b) $L = 200$ ms, overlap = 25%.

(c) $L = 100$ ms, overlap = 50%.

(d) $L = 200$ ms, overlap = 50%.

(e) $L = 100$ ms, overlap = 75%.

(f) $L = 200$ ms, overlap = 75%.

(g) $L = 100$ ms, overlap = $L - 1$.

(h) $L = 200$ ms, overlap = $L - 1$.

Figure 3.6: Results of the time-variable gain estimation for $L = 100$ ms and $L = 200$ ms using a gain curve with a linear fade-in, a constant and a linear fade-out parts.

Chapter 4

Estimation of the Filter Coefficients

Even though we have access to the original audio recording that has been used as a music-track in a film, soap opera or any other audiovisual piece, we are not able to conclude, a priori, whether the film-makers have applied a filter to the musical signal before putting it on their work. Hence, it is important to have a step where we compare the reference signal (original recording), which we have at hand, with the potentially filtered music signal, which actually appears in the final version of the soundtrack of an audiovisual work mixed with the dialogue signal.

This chapter explains the procedure utilised in order to estimate the filter coefficients that should be used to filter the music-track signal before applying a variable gain on it and creating the mixture (soundtrack). Section 4.1 gives a detailed explanation of how the algorithm works and Section 4.2 shows the results of the simulations made in order to test the performance of the algorithm.

Before continuing, it is important to emphasise that the whole mathematical analysis of this part as well as every simulation assumes the signals have already been synchronised. This means the chapter will consider the algorithm for estimation of the filter coefficients as a standalone version, not taking into account any delay detection or gain estimation procedures that have also to be done to effectively remove a musical track from the soundtrack signal. It has been used only artificially created mixtures where it is possible to make sure the delay and the gains are known.

4.1 Wiener Filtering Algorithm

A generic signal enhancement [43] task is illustrated in Figure 4.1. In this type of task, a signal $x(n)$ is corrupted by an uncorrelated noise $\eta_1(n)$. Another noise signal $\eta_2(n)$, different from $\eta_1(n)$, but correlated with it, is also available. If $\eta_2(n)$ is used as an input to a filter $h(n)$ while $x(n)$ is seen as a reference (desired) signal, it is possible to estimate the coefficients of the filter that makes $y(n)$ maximally close to $\eta_1(n)$. Such task can be analysed as a minimisation of a cost function related to the error signal $\epsilon(n)$, which will become an enhanced version of the corrupted signal and consequently a good estimation for the target signal $x(n)$.

There are many ways to tackle this minimisation problem depending on the cost function to be minimised [43]; however, since we have access to the whole music-track and the whole soundtrack before the analysis, the algorithm does not need to be implemented in real time. In this case it is possible to use the expected value of the squared error signal, which is the cost function for the ideal signal enhancement problem. This algorithm is known as Wiener Filtering [42] Algorithm, and the final estimated coefficients for the filter are called Wiener Filter Coefficients.

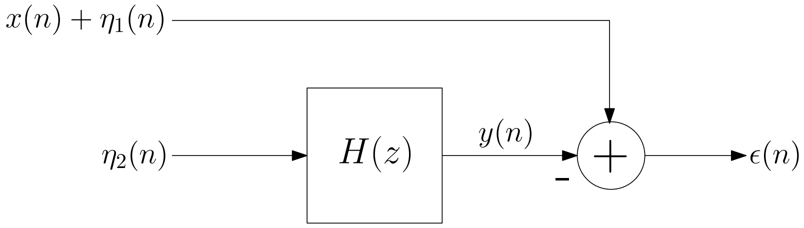


Figure 4.1: Generic signal enhancement framework.

Putting the signal enhancement problem under the perspective of the dissertation, it is possible to say that there is a segment of a reference music signal $m(n)$ which has been filtered by an unknown filter, generating the signal $m'(n)$. After the filtering process, $m'(n)$ was added to the dialogue-sfx signal $d(n)$, generating the soundtrack signal

$$s(n) = d(n) + m'(n). \tag{4.1}$$

Observe that in Equation (4.1) we are analysing a segment of the soundtrack signal where an excerpt of $m(n)$ is the music-track, this is the reason why we can use $m(n)$ instead of $\mu(n)$ in this case. Therefore, if we consider $m(n)$ (the signal we have access to) a different, but correlated signal with $m'(n)$, we can use the framework illustrated in Figure 4.2 and conclude the problem can also be modelled as a signal enhancement task.

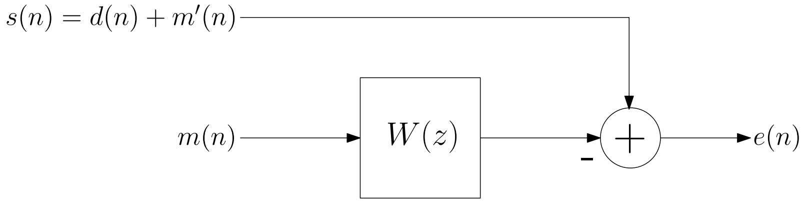


Figure 4.2: Dialogue-sfx enhancement framework.

Thus, we need to find the coefficients of a filter $W(z)$ that minimise the expected value of the squared error signal $e(n)$. Supposing the filter is an Finite-length Impulse Response (FIR) [37] filter with size M and we analyse the problem in a particular time sample $n^\star$, consider the following definitions:

- Vector $\mathbf{w}(n^\star)$ is the vector with the coefficients of the filter in this particular time sample, i.e.,

$$\mathbf{w}(n^\star) = [w_0(n^\star) \;\; w_1(n^\star) \;\; \ldots \;\; w_{M-1}(n^\star)]^\top ; \tag{4.2}$$

- Vector $\mathbf{m}(n^\star)$ is comprised of M samples of the original music track signal starting from $n^\star$ until $n^\star + M - 1$, i.e.,

$$\mathbf{m}(n^\star) = [m(n^\star) \;\; m(n^\star+1) \;\; \ldots \;\; m(n^\star+M-1)]^\top . \tag{4.3}$$

Using those definitions, it is easy to observe that the output of the filter can be calculated by a simple inner product,

$$m_w(n^\star) = \mathbf{w}^\top(n^\star)\mathbf{m}(n^\star) = \mathbf{m}^\top(n^\star)\mathbf{w}(n^\star). \tag{4.4}$$

Now, we can use Equation (4.4) and compute the expected value of the squared error signal, which is the cost function of the minimisation problem we should solve.

$$\begin{aligned}\mathbb{E}\Big\{e^2(n^\star)\Big\} &= \mathbb{E}\Big\{\,[s(n^\star) - \mathbf{w}^\top(n^\star)\mathbf{m}(n^\star)]\,[s(n^\star) - \mathbf{w}^\top(n^\star)\mathbf{m}(n^\star)]\,\Big\}\\ &= \mathbb{E}\Big\{s(n^\star)s(n^\star)\Big\} - 2\,\mathbf{w}^\top(n^\star)\,\mathbb{E}\Big\{s(n^\star)\mathbf{m}(n^\star)\Big\} + \\ &+ \mathbf{w}^\top(n^\star)\mathbb{E}\Big\{\mathbf{m}(n^\star)\mathbf{m}^\top(n^\star)\Big\}\mathbf{w}(n^\star).\end{aligned} \tag{4.5}$$

Note that the cost function on Equation (4.5) is a second-order positive definite function of $\mathbf{w}(n^\star)$ [43]; so, it is possible to minimise it by equating its derivative with respect to the filter coefficients to zero. Mathematically,

$$\frac{\partial\mathbb{E}\Big\{e^2(n^\star)\Big\}}{\partial\mathbf{w}(n^\star)} = -2\,\mathbb{E}\Big\{s(n^\star)\mathbf{m}(n^\star)\Big\} + 2\,\mathbb{E}\Big\{\mathbf{m}(n^\star)\mathbf{m}^\top(n^\star)\Big\}\mathbf{w}(n^\star), \tag{4.6}$$

$$\frac{\partial\mathbb{E}\Big\{e^2(n^\star)\Big\}}{\partial\mathbf{w}(n^\star)} = \mathbf{0} \quad\Rightarrow\quad \mathbb{E}\Big\{\mathbf{m}(n^\star)\mathbf{m}^\top(n^\star)\Big\}\mathbf{w}(n^\star) = \mathbb{E}\Big\{s(n^\star)\mathbf{m}(n^\star)\Big\}. \tag{4.7}$$

In order to solve Equation (4.7), it is important to observe that $\mathbb{E}\Big\{\mathbf{m}(n^\star)\mathbf{m}^\top(n^\star)\Big\}$ is the $M \times M$ autocorrelation matrix [11] $\mathbf{R}_{mm}(n^\star)$ of the reference signal $m(n^\star)$, i.e.,

$$\mathbb{E}\Big\{\mathbf{m}(n^\star)\mathbf{m}^\top(n^\star)\Big\} = \mathbf{R}_{mm}(n^\star), \tag{4.8}$$

which can be rewritten as being the matrix:

$$\mathbb{E}\left\{\begin{bmatrix} m(n^\star)m(n^\star) & m(n^\star)m(n^\star{+}1) & \dots & m(n^\star)m(n^\star{+}M{-}1) \\ m(n^\star{+}1)m(n^\star) & m(n^\star{+}1)m(n^\star{+}1) & \dots & m(n^\star{+}1)m(n^\star{+}M{-}1) \\ \vdots & \vdots & \ddots & \vdots \\ m(n^\star{+}M{-}1)m(n^\star) & m(n^\star{+}M{-}1)m(n^\star{+}1) & \dots & m(n^\star{+}M{-}1)m(n^\star{+}M{-}1) \end{bmatrix}\right\}.$$

In addition,

$$\mathbb{E}\Big\{s(n^\star)\mathbf{m}(n^\star)\Big\} = \begin{bmatrix} \mathbb{E}\Big\{s(n^\star)m(n^\star)\Big\} \\ \mathbb{E}\Big\{s(n^\star)m(n^\star{+}1)\Big\} \\ \vdots \\ \mathbb{E}\Big\{s(n^\star)m(n^\star{+}M{-}1)\Big\} \end{bmatrix} = \begin{bmatrix} r_{sm}(n^\star, n^\star) \\ r_{sm}(n^\star, n^\star{+}1) \\ \vdots \\ r_{sm}(n^\star, n^\star{+}M{-}1) \end{bmatrix} = \mathbf{r}_{s\mathbf{m}}(n^\star).$$

- ⋆ Note that in the music-track removal case, it is possible to estimate not only the autocorrelation values that appear in matrix $\mathbf{R}_{mm}(n^\star)$, but also the cross-correlation values that appear in $\mathbf{r}_{s\mathbf{m}}(n^\star)$ because the whole signal $m(n)$ is available before-hand. Such values are estimated using the ergodicity property [11] by replacing the expected values with their corresponding averaged time-domain values.

Therefore, Equation (4.7) can be rewritten as

$$\mathbf{R}_{mm}(n^\star)\mathbf{w}(n^\star) = \mathbf{r}_{s\mathbf{m}}(n^\star), \tag{4.9}$$

which is a linear system whose solution gives the Wiener filter coefficients

$$\mathbf{w}(n^\star) = \mathbf{R}_{mm}^{-1}(n^\star)\mathbf{r}_{s\mathbf{m}}(n^\star). \tag{4.10}$$

Observe that the final Wiener filter coefficients will directly depend on the choice of the analysis sample $n^\star$. Such dependence is verified by Equation (4.10), which shows that the final coefficients will be a function of $n^\star$. Hence, given we have a considerable time interval to work with,

- Which value of $n^\star$ is the best value to be used in the algorithm?

This is a tricky question to receive an exact answer. Let us say we have found a musical segment in $\mu(n)$ that have been properly synchronised to the reference music template available. During the whole duration the excerpt, even though the programme makers might have applied a time-variable gain, it is not usual to use different filters in the process. On the other hand, if we use this approach to the estimation, the final Wiener filter coefficients will be different depending on the sample $n^\star$ we choose to do the analysis. The project decision was to use a few different values of $n^\star$ to process the signals and combine the different results in order to get the final estimation for the filter coefficients. The values of $n^\star$ were chosen spaced out by a pre-determined interval of I samples along the music segment duration. Such parameter is defined by the user.

Furthermore, it is important to realise that we should not use a very large interval around $n^\star$ to estimate the necessary correlations that appear in Equation (4.10). The reason is that, due to the possibility of existence of a time-variable gain in the music-track segment, a very large window around $n^\star$ would lead to a bias in the correlation approximation for that specific point. Since in Chapter 3 the project considered small windows where the time-variable gain could be assumed constant, in we are going to use the same size for the correlation windows: 200 ms, as shown in the results from the simulations in Section 3.3.

4.1.1 Estimation of the Final Coefficients

Considering we have estimated a total of N different Wiener filters $\mathbf{w}(n_1^\star), \mathbf{w}(n_2^\star), \ldots, \mathbf{w}(n_N^\star)$, each using a different analysis sample $n_1^\star, n_2^\star, \ldots, n_N^\star$, it is possible to define their frequency response as

$$W_i(\mathrm{e}^{\mathrm{j}\Omega}) = \mathcal{F}\left\{\mathbf{w}(n_i^\star)\right\}, \qquad \text{with } i \in \{1, 2, \ldots, N\}, \tag{4.11}$$

where $\mathcal{F}\{\bullet\}$ represents the Discrete-Time Fourier Transform (DTFT) [37] of the argument.

The algorithm to combine the coefficient values to form the final filter $\mathbf{w}(n)$ consisted of a simple average value of the filter coefficients or a median value procedure, which could be applied to time-domain or to frequency-domain representation. Therefore, there were 3 different options to create the final Wiener filter tested by the author:

1 – Time-Domain Average Value (TDAV):

$$\mathbf{w}(n) = \frac{\mathbf{w}(n_1^\star) + \mathbf{w}(n_2^\star) + \cdots + \mathbf{w}(n_N^\star)}{N}; \tag{4.12}$$

2 – Time-Domain Median Value (TDMV):

$$\mathbf{w}(n) = \begin{bmatrix} \texttt{median}(w_0(n_1^\star), w_0(n_2^\star), \ldots, w_0(n_N^\star)) \\ \texttt{median}(w_1(n_1^\star), w_1(n_2^\star), \ldots, w_1(n_N^\star)) \\ \vdots \\ \texttt{median}(w_{M-1}(n_1^\star), w_{M-1}(n_2^\star), \ldots, w_{M-1}(n_N^\star)) \end{bmatrix}; \tag{4.13}$$

3 – Frequency-Domain Median Value (FDMV):

$$W(\mathrm{e}^{\mathrm{j}\Omega}) = \begin{array}{l} \texttt{median}\left(\mathrm{Re}\left\{W_1(\mathrm{e}^{\mathrm{j}\Omega})\right\}, \ldots, \mathrm{Re}\left\{W_N(\mathrm{e}^{\mathrm{j}\Omega})\right\}\right) \quad + \\ \texttt{median}\left(\mathrm{Im}\left\{W_1(\mathrm{e}^{\mathrm{j}\Omega})\right\}, \ldots, \mathrm{Im}\left\{W_N(\mathrm{e}^{\mathrm{j}\Omega})\right\}\right) \mathrm{j} \end{array} \tag{4.14}$$

Section 4.2 shows the results for low-pass, band-pass and high-pass filters being estimated by the aforementioned algorithm.

4.2 Wiener Filtering Simulations

All simulations in this section have been executed using only artificial signals in order to be possible to compare the effectively used filter with the final Wiener filter estimated by the algorithm. It is expected that the results will be different depending on the frequency band stimulated by the music $m(n)$; in order to highlight this issue, 2 different musical excerpts of 17 seconds have been taken from 2 commercial song signals: the first had no spectral component above 15 kHz, while the second exhibited much more information at higher frequencies. It is possible to compare the difference in their frequency span by comparing their spectrograms, shown in Figures 4.3 and 4.4. The sampling frequency was 48 kHz and the order of the estimated Wiener filters was 255. The filters utilised in the simulations were artificially created trying to emulate a usual equalisation filter often used by producers in commercial musical recordings.

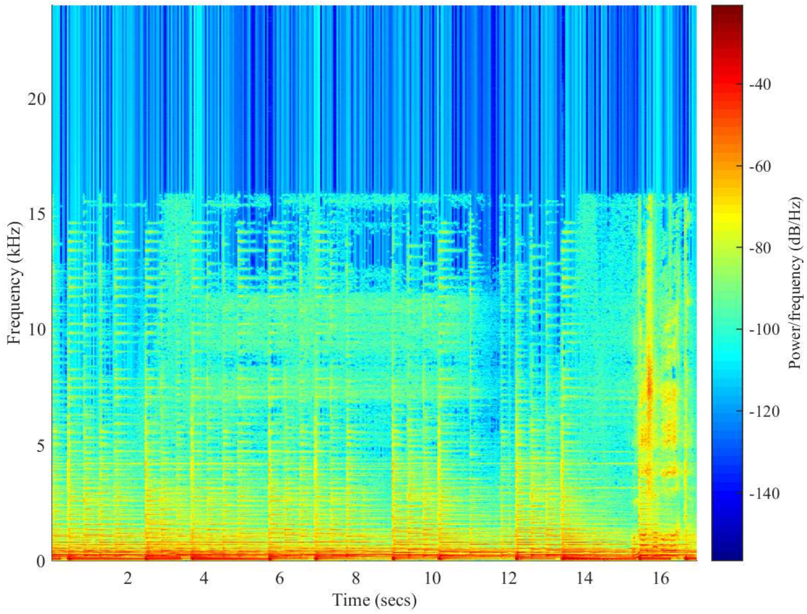


Figure 4.3: Spectrogram of the first excerpt signal.

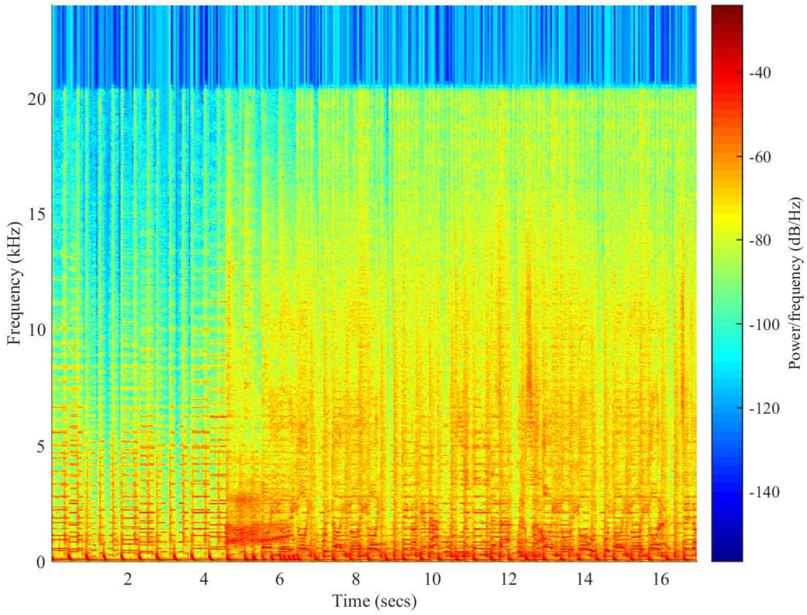


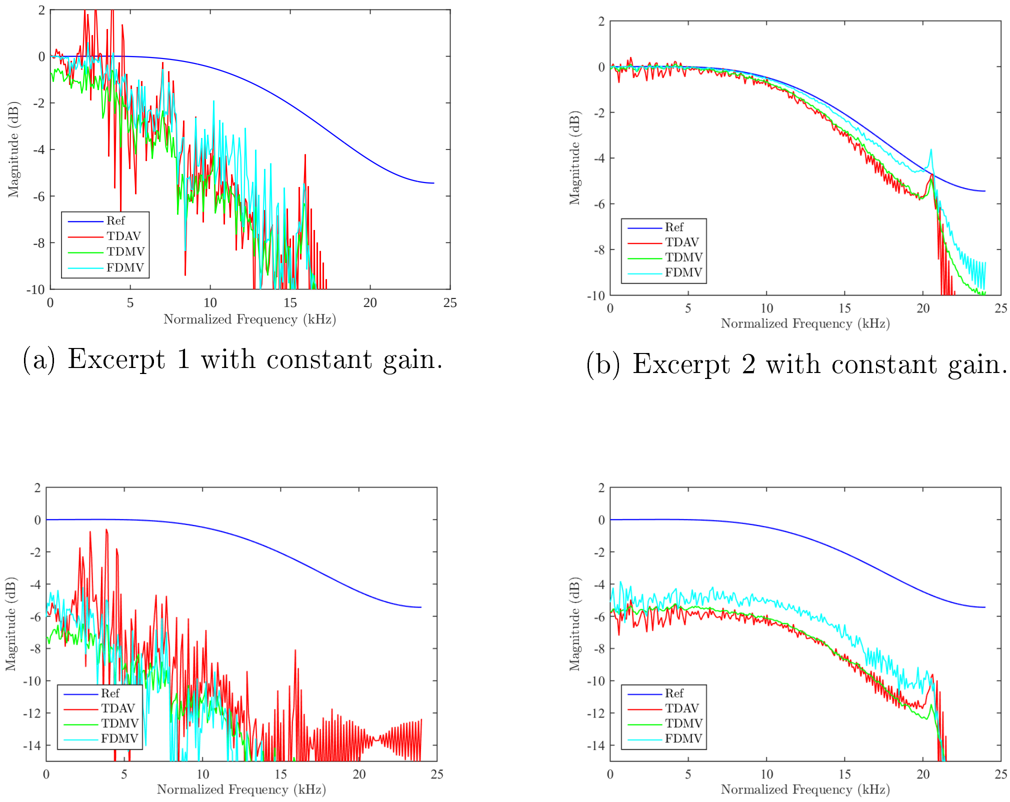
Figure 4.4: Spectrogram of the second excerpt signal.

4.2.1 Low-pass Filtering Simulation

In this simulation, the low-pass filter with the coefficients

$$\begin{bmatrix} -0.03214 & 0.11627 & 0.83115 & 0.11627 & -0.03214 \end{bmatrix}^\top$$

was applied to both excerpts. Then they were added to a dialogue signal using a constant gain of 1 and a variable gain such as the one shown in Figures 3.1 and 3.2. The results are shown in Figure 4.5.



(a) Excerpt 1 with constant gain.

(b) Excerpt 2 with constant gain.

(c) Excerpt 1 with variable gain.

(d) Excerpt 2 with variable gain.

Figure 4.5: Final low-pass Wiener filter estimation.

4.2.2 Band-pass Filtering Simulation

In this simulation, it the band-pass filter with the coefficients

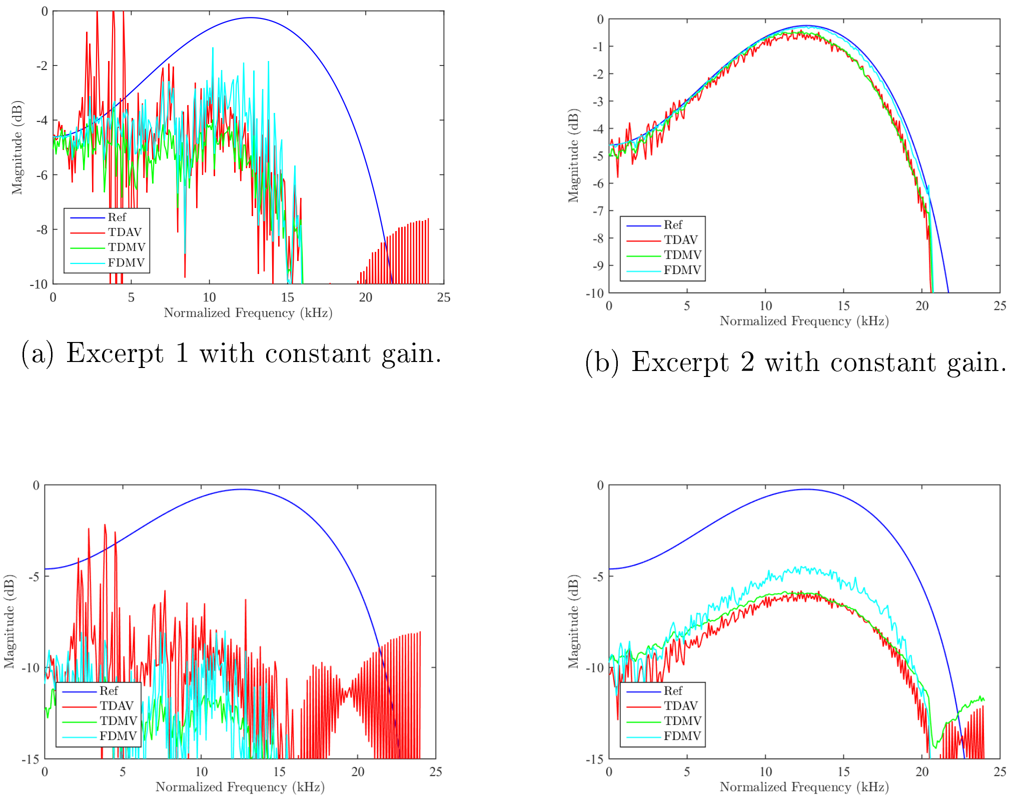
$$\begin{bmatrix} -0.19550 & 0.48972 & 0.48972 & -0.19550 \end{bmatrix}^\top$$

was applied to both excerpts. Then they were added to a dialogue signal using a constant gain of 1 and a variable gain such as the one shown in Figures 3.1 and 3.2. The results are shown in Figure 4.6.

4.2.3 High-pass Filtering Simulation

In this simulation, the high-pass filter with the coefficients

$$\begin{bmatrix} -0.09068 & -0.07476 & 0.91754 & -0.07476 & -0.09068 \end{bmatrix}^\top$$



(a) Excerpt 1 with constant gain.

(b) Excerpt 2 with constant gain.

(c) Excerpt 1 with variable gain.

(d) Excerpt 2 with variable gain.

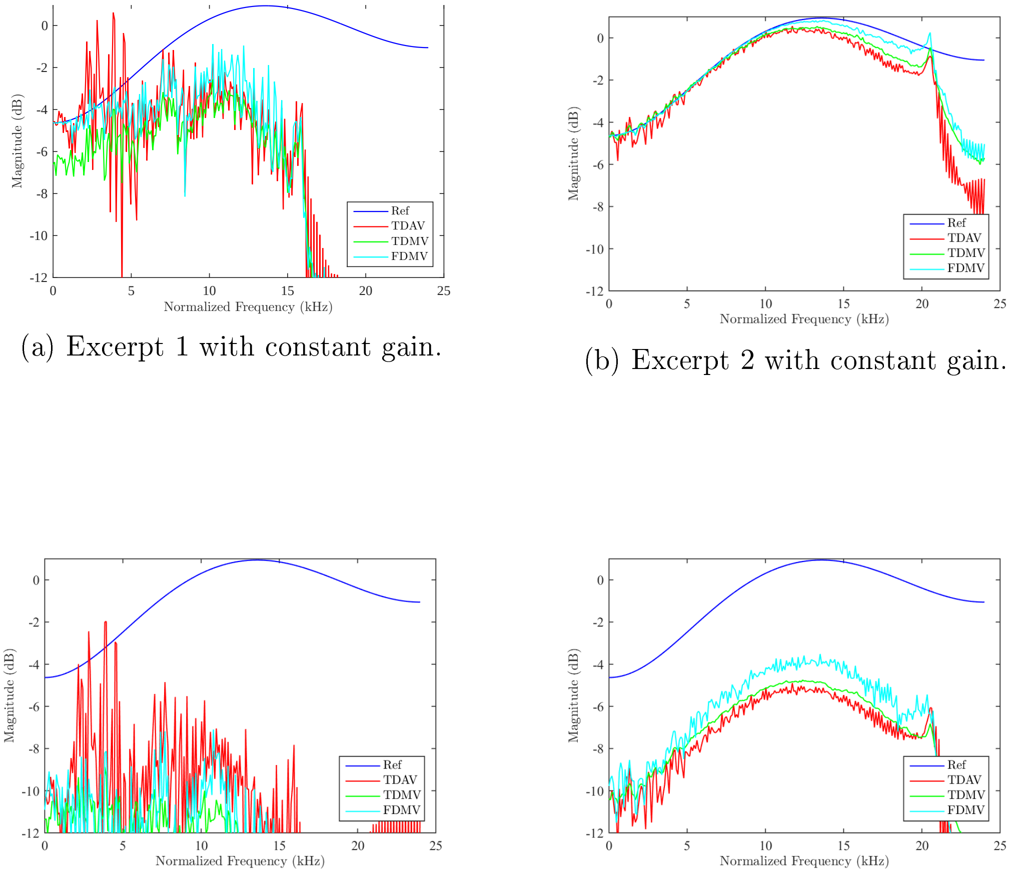
Figure 4.6: Final band-pass Wiener filter estimation.

was applied to both excerpts. Then they were added to a dialogue signal using a constant gain of 1 and a variable gain such as the one shown in Figures 3.1 and 3.2. The results are shown in Figure 4.7.

4.2.4 Final Remarks

In all simulations the results followed the same logic. While the algorithm works well when dealing with excerpts of songs that have a high span of frequencies, its behaviour when processing poorer (referring to the quantities of simultaneous frequencies in the spectrogram of the music segment) songs was not satisfying. This is in line with what was expected. Since we are trying to estimate the filter coefficients analysing only a single sample (segment of $m(n)$) of a random process, it is mandatory for such sample to have sufficient information about the process, which in this case are the frequency components.

It can also be concluded that the best way to combine the filters is to use the process FDMV, where we use as the final Wiener filter response the median value of the real and imaginary parts of the frequency response of each filter. This was the chosen combination procedure to be used in the final music track removal technique.



(a) Excerpt 1 with constant gain.

(b) Excerpt 2 with constant gain.

(c) Excerpt 1 with variable gain.

(d) Excerpt 2 with variable gain.

Figure 4.7: Final high-pass Wiener filter estimation.

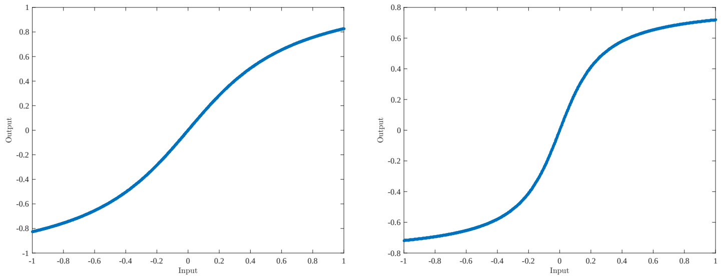
Chapter 5

Non-linear Distortions

This chapter introduces a new component into the separation problem: the presence of non-linear distortions. Such distortions are commonly applied in TV programmes and influence on how each part of the separation algorithm behaves in non-artificially created soundtracks. Some tests have been made applying a soft-clipping function on the soundtracks and the results are reported. It is important to point out that the project did not focus on trying to implement new techniques for estimating and removing the distortions. This single problem involves an enormous amount of challenges that are out of the scope of this project, but it definitely deserves future studies.

5.1 Dynamic Range Compression

Probably the most common source of non-linear distortions in commercial television programmes is the presence of dynamic range compression [44]. Such non-linearity can be modelled as memoryless soft-clipping functions that are applied to a single or various signals during the soundtrack creation. To give a better understanding, an illustration can be seen in Figure 5.1. It can be noted that the term ‘compression’ comes from the fact that the high valued samples are mapped into smaller values and vice-versa.



(a) Soft Compression.

(b) Hard Compression.

Figure 5.1: Example of soft-clipping/dynamic range compression functions.

In many cases, the audio signal of a television programme suffers from dynamic range compression. This is often intentionally applied by the broadcasting industry to reduce the volume of loud sounds and to amplify the volume of quiet sounds [45]. Therefore, it is important to check how the system will behave when such distortions are present and consider the application of audio declipping techniques such as [46] if necessary.

This type of non-linear distortion is the only type the author has considered for study, but there are other types that could also be present under the project perspective. Some examples are distortions due to codification or quantisation procedures, and compressing functions with memory. Both types of distortions may potentially be present in real-life TV programmes.

In the specialised literature, there are some works that try to estimate the inverse function of a particular curve of compression with the objective of decompressing the data [46–48]. Such methods are usually known as 'declipping' methods. What they basically do is to model a generic soft-clipping function as a particular shaped form and resolve an optimisation problem using the compressed data as the observations of the process.

Before starting with the simulations, it is important to note that depending on the signal that is being compressed, the solution to the problem will become more challenging or less challenging. For instance, imagine the music-track is added to the dialogue-sfx signal and they both are compressed together. If our algorithm do not handle the separation problem by itself, we can use one of the declipping methods cited above on the soundtrack to estimate the compressing function applied and utilise it to decompress the data for a later separation. However, note that if the musical segment is compressed before being added to the mixture, applying a decompressing technique on the soundtrack will not solve the problem, and will keep distortions present in the music-track.

5.2 Simulations

Two simulations were performed to test the performance of the algorithm. The first applied a variable gain to the excerpt and executed the compression of the mixture, whereas the other executed the compression of the excerpt before applying the variable gain and creating the mixture. The compression was perfomed in a 2-step procedure:

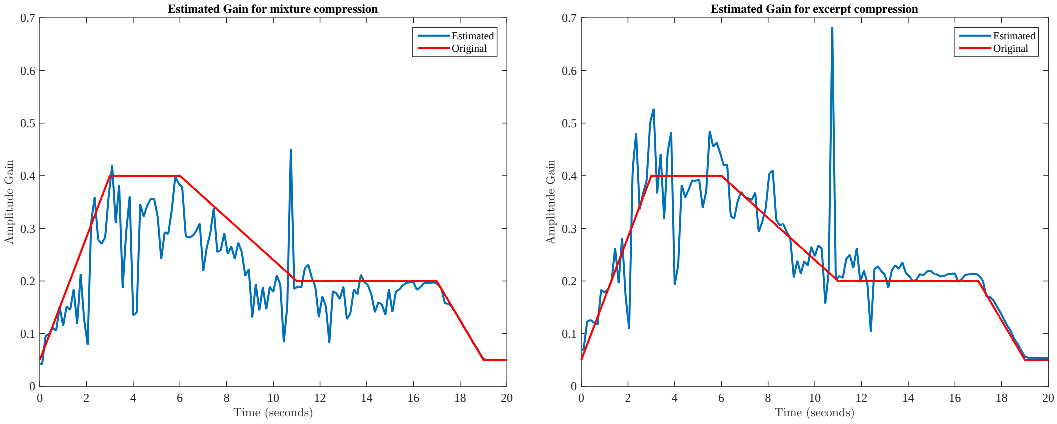
1 – Application of an arctangent function to the original signal ('input'),

$$\text{output} \quad \Leftarrow \quad \frac{\texttt{arctan}(2 \cdot \text{input})}{2}$$

2 – Normalisation of the standard deviation of the compressed signal ('output'),

$$\text{output} \quad \Leftarrow \quad \frac{\texttt{std}(\text{input})}{\texttt{std}(\text{output})} \cdot \text{output}$$

The excerpt is 20-second long and the same size was used for the segments searched in the mixture. After the removal procedure was performed, the results for the estimated gain in each case are illustrated in Figure 5.2, while the results for the separated signals can be seen in Table 5.1.



(a) Estimated gain for the first case. (b) Estimated gain for the second case.

Figure 5.2: Results for the time-variable gain estimation procedure applied on compressed versions of soundtrack.

Table 5.1: Separation results for both simulations in dB.

Compression on	SDR	SIR	SAR
mixture	14.87	38.31	14.88
excerpt	18.23	34.30	18.45

Analysing the results it is possible to say the time-variable gain estimator was pretty robust when estimating the gains for both compressed cases. It seems the algorithm was capable of dealing with dynamic range compression by itself, and therefore, there was no need to consider the application of declipping methods to improve the results.

An important fact to note is that, in the estimated gain-curves, there are many peaks and random oscillations that were not present in the originally applied gain functions. This happened because the application of a compression function to the signals was modelled by the separation algorithm as being variations on the time-variable gain. In order words, those peaks appeared in the estimated versions of the gain-curves as an attempt of the gain-estimation step to adapt for the non-linearities imposed by the compression.

This simulation has been done using simple artificially created excerpts and a single compression function, thus nothing can be stated for the general compressed case; the algorithm must be further tested over compressed signals to yield a reliable and more precise conclusion.

Chapter 6

Removal of Specific Segments of the Music-Track

This chapter describes how to implement the complete removal procedure, putting together the previously explained algorithms. It explains how each part can be combined and shows the results of a case study using artificially created signals and considerations regarding the application of the algorithm on real soundtrack signals.

Before starting the detailed explanation of the complete removal algorithm, it is important to point out that all algorithms related to the project are publicly available in [49].

6.1 Complete Removal Algorithm

In Section 2.1, it was mentioned that the quick-search algorithm was implemented using the landmarks of the mixture signal as the basis to construct the reference hash-token matrix. Such decision makes the removal procedure become easier to implement as an iterative remove-reconstruct database algorithm. Its basic principle of operation is to start computing the landmarks of the soundtrack signal to construct the reference hash-token matrix. After this, the original music-track we have available is divided into smaller segments without overlap that are searched in the soundtrack using the quick-search algorithm explained in Section 2.1. After the presence of some of them has been confirmed, the synchronisation procedure of Section 2.2 is applied.

Then, the musical segments can be used as templates for the template matching technique explained in Chapter 3 to estimate the time-variable gain applied to them. Once we have the different gains at hand, it is possible to remove the segments from the soundtrack just by subtracting scaled versions of the template in the time-domain. After all this process is finished, the stored hash-token matrix is deleted from the database and a new one is constructed, but, this time, using the processed version of the soundtrack instead. Since we are constantly subtracting scaled versions of the original musical segments from the mixture, it is expected that in the next iteration there will be less landmark matches between the musical segments and the 'new' soundtrack. The whole search-syncronise-scale-remove-reconstruct procedure is repeated until a stop condition is reached. Such condition can be defined as a maximum number of iterations, or as a small value for the maximum landmark matches between each segment and the iteratively reconstructed database.

The Wiener filtering algorithm explained in Chapter 4 is applied only once during the whole process, if we desire to estimate a potential equalisation filter. The author decided to use it as a pre-processing step, where larger and overlapping segments of the reference music signal are used as potential templates to be searched in and aligned with the mixture. The idea is that once the 'best' segment (the segment that has the higher number of landmark matches with the same relative time offset with the database) is found and synchronised, it can be used to estimate the Wiener filter coefficients, which are then used to filter the whole music-track signal at once. By performing the filtering process this way it is possible to avoid problems that may arise when considering the synchronisation procedure and filter delay. It is easier if we now work with an already filtered version of the music-track to be removed from the soundtrack instead. The filtered music signal is then used as the actual reference signal for the musical segments search and removal.

It is important to note that it is possible to perform the Wiener filtering algorithm in the aforementioned way if we consider that we are looking for segments of the reference signal in a soundtrack where there is only a single filter potentially applied to every segment of the reference music. If, for example, a large soundtrack signal should be processed by the algorithm, where several different filters were used to filter the music segments, it is necessary that each part of the soundtrack signal is processed separately.

During the implementation and test phases of the algorithm, it has been observed an unexpected interesting fact. Considering we obtained synchronised versions of reference segment and of the soundtrack signal, there will not necessarily be a lower amount of landmark matches in the next iteration. This happens because when we estimate a time-variable gain using the template matching technique, the gain usually have many low-amplitude, but peaky oscillations around a smooth gain function. When we apply it to the excerpt and perform its removal from the soundtrack signal, the peaky oscillations create an oscillating function around zero DC value that mixes with the dialogue-sfx in the residual signal, thus creating new peaks in the mixture spectrogram in the next iteration of the removal process and, consequently, yielding more landmarks that end up matching with a musical segment.

And this problem cannot be solved by the template matching algorithm, which is not able to generate a new set of gains, since there is not actually a musical segment perfectly synchronised with its mirrored version in the mixture.

This problem affected the stopping conditions of the algorithm and could lead it into an infinite loop. Therefore, a new logic of implementation or a procedure for fixing this problem must be properly addressed in future work.

6.2 Artificial Signal Case Study

Four different simulations have been done to test the performance of the separation method. The same 17-second duration excerpt signal utilised in the simulations of Chapter 3 has been also used in this chapter as an artificial signal for case study. The simulations have been performed dividing the whole reference music signal into small segments. The idea is to compare how a specific choice for the size of the segments affects the performance of the algorithm. The Wiener filter procedure, when applied, was always performed by synchronising the best segment of 15 second duration of the music signal with the soundtrack before applying the algorithm explained in Chapter 4.

6.2.1 Simulation 01 – Constant Gain and no Filter

In this simulation, a constant gain of 0.45 was applied to the musical excerpt before adding it to a generic dialogue-sfx signal. No filter was used in the process. Table 6.1 shows the results of the segment removal technique skipping the Wiener filtering method.

Table 6.1: Separation results for Simulation 01 in dB.

Excerpt Duration	SDR	SIR	SAR
3 s	11.44	20.78	12.01
5 s	15.56	27.13	15.88
7 s	16.63	27.72	16.99
10 s	15.37	31.28	15.48
15 s	19.08	35.38	19.19
20 s	16.57	34.75	16.64
25 s	23.67	34.78	24.02
30 s	15.57	33.40	15.64

It can be noted that for a constant gain and no equalisation filter, the algorithm performs better when dealing with segments of the music-track with size approximately equal to the 17 seconds of the excerpt applied to the signal. If the size of each segment is too small, there will be several undetected borders between contiguous segments detected inside the original 17-second signal, which will decrease the separation quality. However, for too large sizes we can say the performance may also worsen due to the fact explained in Subsection 2.3.3. The algorithm may end up finding a segment larger than 17 seconds in the mixture, and even though the time-variable gain is able to approximate the silent part of the segment to zero, it still impacts negatively the separation results.

6.2.2 Simulation 02 – Variable Gain and No Filter

In this simulation, a variable gain such as the blue line in Figure 3.1 was applied to the 17-second excerpt signal. This time there was also no equalisation filter, but the Wiener procedure was tested to see how the algorithm would behave. The quality assessment of the results of the separation using different segment sizes for the quick-search method is shown in Table 6.2. The frequency response of the final estimated Wiener filter is shown in Figure 6.2.

Table 6.2: Separation results for Simulation 02 in dB.

—	No Wiener Filtering			With Wiener Filtering		
Excerpt Duration	SDR	SIR	SAR	SDR	SIR	SAR
3 s	16.44	34.16	16.52	15.22	23.62	15.92
5 s	19.33	38.79	19.38	15.82	23.09	16.74
7 s	19.69	38.65	19.74	16.25	25.63	16.80
10 s	19.23	33.70	19.39	15.69	23.68	16.45
15 s	22.22	38.96	22.32	18.30	23.19	20.01
20 s	22.75	37.30	22.91	20.17	23.18	23.21
25 s	29.06	39.68	29.45	20.07	23.09	23.09
30 s	22.21	37.17	22.35	18.84	23.32	20.77

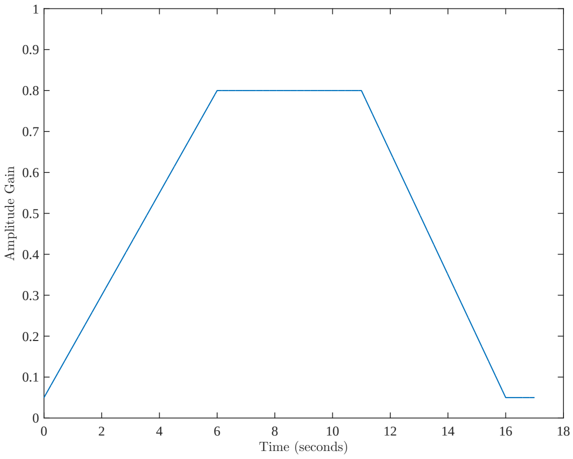


Figure 6.1: Time-variable gain curve applied to the excerpts during Simulations 02 and 04.

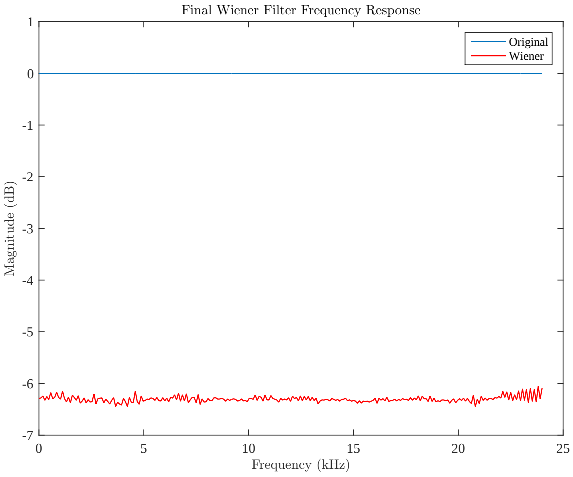


Figure 6.2: Frequency response of the original filter used in Simulation 02 and the respective response of the estimated Wiener filter.

Overall, the results of the algorithm applied to a time-variable gain mixture are in line with those of the previous simulation. However, we can see that the Wiener filter had many small oscillations around an approximately constant gain. Such oscillations leave residuals on the separated signals, hence lowering the performance of the algorithm.

6.2.3 Simulation 03 – Constant Gain and Filter

In this simulation, a low-pass filter was applied to the excerpt signal before adding it to the dialogue-sfx signal. The constant gain is the same value applied in Simulation 01. We use the Wiener filtering to estimate the filter coefficients followed by the template matching technique to find the gains. Table 6.3 shows the results. Furthermore, the removal procedure without the use of any estimated filter is also shown for comparison.

Table 6.3: Separation results for Simulation 03 in dB.

—	No Wiener Filtering			With Wiener Filtering		
Excerpt Duration	SDR	SIR	SAR	SDR	SIR	SAR
3 s	13.36	25.08	13.68	13.07	23.52	13.50
5 s	15.47	29.27	15.66	15.14	27.98	15.38
7 s	16.44	29.38	16.67	16.09	29.12	16.32
10 s	16.01	29.41	16.22	18.48	30.44	18.77
15 s	17.04	28.56	17.36	15.90	27.30	16.23
20 s	20.29	27.42	21.23	20.26	29.14	20.86
25 s	20.29	27.42	21.23	20.19	29.37	20.75
30 s	19.55	26.56	20.52	19.49	29.11	20.00

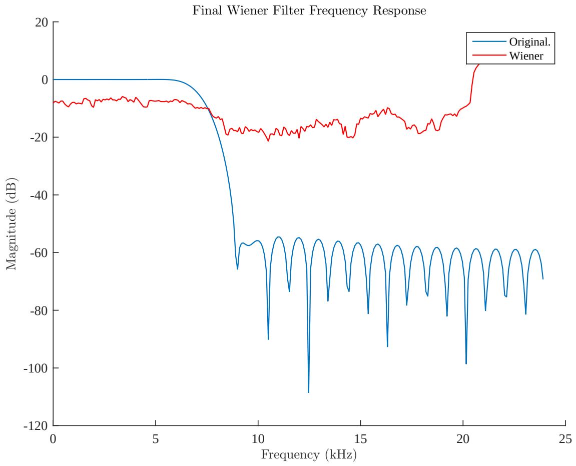


Figure 6.3: Frequency response of the original filter used in Simulation 03 and the respective response of the estimated Wiener filter.

It is expected that the Wiener filtering technique would improve the results of the final separation algorithm. However, by analysing the results, it is possible to note that it did not improve the results at all. In the higher frequencies, the Wiener filter applies incorrect high-values for the gain. Also, there was no equalisation filter applied to the excerpt signal before putting it into the mixture, thus, using the Wiener filtering technique can end up adding more errors in the separation process.

6.2.4 Simulation 04 – Variable Gain and Filter

In this simulation, the same low-pass filter was applied to the excerpt signal before adding it to the dialogue-sfx signal. However, this time, the variable gain used in Simulation 02 was also applied to the excerpts. The results of the removal procedure using the Wiener Filtering technique and without using it appear in Table 6.4.

Table 6.4: Separation results for Simulation 04 in dB.

—	No Wiener Filtering			With Wiener Filtering		
Excerpt Duration	SDR	SIR	SAR	SDR	SIR	SAR
3 s	13.45	25.43	13.75	13.22	23.92	13.62
5 s	14.39	26.47	14.68	14.42	25.08	14.83
7 s	16.81	25.45	17.46	16.70	24.76	17.46
10 s	17.25	26.55	17.80	16.62	25.17	17.28
15 s	17.66	23.17	19.12	18.21	25.44	19.14
20 s	20.32	22.67	24.12	20.76	24.96	22.84
25 s	21.07	23.09	25.38	21.69	25.02	24.42
30 s	19.98	22.58	23.47	20.72	24.98	22.78

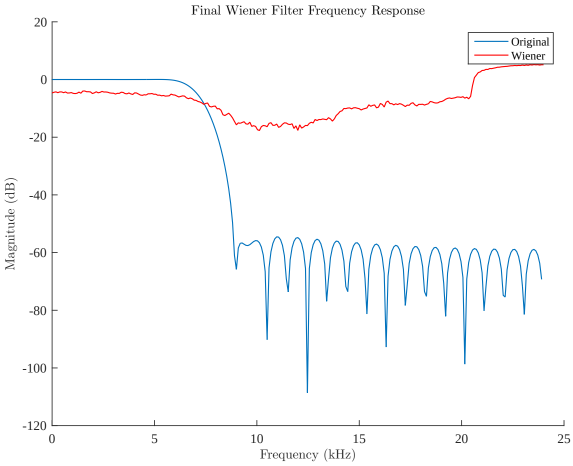


Figure 6.4: Frequency response of the original filter used in Simulation 04 and the respective response of the estimated Wiener filter.

Once again, the Wiener filter incorrectly boosted higher frequencies of the signal. This is mainly because the music does not have much information in higher frequencies, thus the filter is not able to properly find the gains. Overall, those results were expected.

6.3 Real-life TV Programmes

Soundtracks and references music signals from a Brazilian soap opera [34] were obtained from officially published material and they were used to test the performance of the algorithm when applied to non-artificially created soundtrack signals.

Unfortunately, the removal procedure was unable to perform the separation in real-life scenarios, even though it has shown a high level of robustness in the artificially simulated cases. It is hard to indicate a single reason for the incorrect behaviour of the method, but there are some directions that the author considers relevant.

When dealing with soundtracks from real-life TV programmes, the separation system should address many other problems that were not considered in the dissertation until now. First of all, we cannot ensure that the music signals we are employing as references for some parts of the programme's music-track are exactly the same versions that were effectively used in the programme's music-track, even though they might have been obtained from officially published CDs associated with the audiovisual work. And this is not only considering they can be different recordings of the same music with unique characteristics, but also realising that the way the stereo channels are mixed together in the final soundtrack signal might not be directly related to the way they are mixed in its CD counterpart. What guarantees that the producers did not put only one of the channels in their work? Or maybe used the right channel in the left side and vice-versa? We might have made an inadequate simplification when averaging the signals to transform them in mono versions.

Furthermore, it is important to note that the removal procedure is done in the time domain. This makes the algorithm really sensitive to errors in the synchronisation step. If the musical segment is synchronised a few samples forward or behind the exact sample, the whole gain estimation algorithm will not work properly and wrong values for the gains are going to be estimated for the excerpt signal. It is not difficult to imagine a case where the quick-search method would have problems to synchronise; in stereo signals, for example, it is common to have the channels mixed with lagged versions of the same signal. Hence, when processing such type of signals, it is easy to note the search method will have difficulties to find a proper synchronisation. If this happens, the algorithm may continuously find a segment around a sample in time, but will never get the correct time variable gains to properly remove it from the soundtrack.

Another fact worthing pointing out is that, even though the algorithm works with soft-clipping functions as demonstrated in Chapter 5, there may exist other non-linear distortions that could affect the performance of the separation algorithm. We are not able to confirm if it was the same zero-loss version that we have available in our `.wav` file that was added to the soundtrack. The creators of the audiovisual material might have used encoded file types produced by a lossy data compression technique such as `.mp3` files.

Chapter 7

Conclusion and Future Works

This chapter has the final set of conclusions that can be drawn with the research project. It also gives the author's final considerations with respect to the problem of automatic removing of musical segments from the soundtrack of audiovisual materials. It finishes with new ideas for future works.

7.1 Final Remarks

This work proposed the creation of a system to automatically detect and remove musical excerpts from audiovisual material. Overall, considering the whole set of simulations presented in the previous chapters of the dissertation, it is possible to conclude that the algorithm performed satisfactorily in artificially created scenarios where all the variables and parameters present in the mixture process were considered.

The system can effectively check which musical segments of a particular music is present in a soundtrack and is also able to properly synchronise them to the instant they appear in the mixture. The detection algorithm is robust to noise (other sources in the mixture such as characters' dialogues and sound effects) and have a lower complexity if compared to regular searching algorithms based on cross-correlation functions. The results showed that the audio-fingerprinting technique can be used to quick-search a musical segment in the soundtrack signal with efficiency. However, when dealing with echoes and reverberations, the synchronisation of the musical excerpts becomes more difficult because, when executing the proposed method under those circumstances, multiple relative time-offsets emerge from the matching landmarks, yielding several clusters that form different lines in the scatter-plot of Input Offset × Output Offset.

The time-variable gain estimator could also correctly predict values for gains, even in cases where discontinuities and rapid varying oscillations were present. However, a major drawback of the system is that it is really sensitive to errors in the synchronisation step, as mentioned in Section 6.3.

The Wiener filtering procedure is probably the step that showed most fragile among them all. It is really dependent on the frequency information of the music-track we would like to remove, hence there is definitely much room for improvements in this step.

The application of the proposed separation algorithm in soundtracks from real TV programmes was unfortunately unsuccessful. Such real-case scenarios need a more detailed investigation to handle the new set of unconsidered variables in the dissertation such as how to properly tackle a stereo mixed signal or how to detect the presence of other non-linear distortions.

7.2 Research Directions

As future works, it is first and foremost proposed to effectively investigate what is the major issue that is blocking the algorithm to perform the separation on real soundtrack signals. The principal options are the presence of several non-linearities, the existence of particular stereo remixing, and the presence of lagged versions of the excerpts in left-right channels. Another possibility is the use of different encoded versions that prevent the template matching technique to properly estimate the correct gains for the removal.

The major drawbacks of the algorithm have already been stated throughout the previous chapters and there is still a lot of room for improvements, specially when considering the Wiener filter algorithm, which is the step of the removal procedure with the worse results. Some ideas for improvements in the estimation of the coefficients are to instead of using a simple Wiener filter procedure one could try a more complex algorithm such as Kalman filter to properly handle non-stationary signals, or use an adaptive filtering technique.

Another direction for future works could be improving the results using the partially fixed non-negative matrix factorisation in a similar way it is done in [19] when detecting samples of musics in other songs.

The dissertation discussed each subject related to the implemented algorithm in detail as well as its particular drawbacks, thus the reader can easily make their own adaptation of the system or use the implemented codes [49] as a starting point for new projects. Furthermore, it tried to present the theme with as many explanations as possible to give a better understanding of this new and challenging task.

Appendix A

Objective Quality Assessment of Sound Source Separation Results

Consider the following model for one of the resultant signals of a generic sound sources separation process [32]:

$$\hat{s} = s_{\text{source}} + e_{\text{interf}} + e_{\text{artef}} + e_{\text{noise}}, \tag{A.1}$$

where $\hat{s}$ is an estimate of the original signal s_{source} emitted by one of the sources in the mixture, e_{interf} is the interference caused by the other sources in the separated signal, and e_{artef} are the artefacts or defects possibly inserted in the result due to the separation procedure. The term e_{noise} should be added in the signal modelling when there is a necessity for representing a noise in the mixture. In our case, this value will be zero.

Note that it is necessary to have a prior knowledge not only of the original signal of the target source we want to evaluate, which, in this case, is a segment of the dialogue-sfx signal, but also of the other original sources in the mixture, which, in this case, is the segment of the music-track signal.

Now, it is possible to define 3 important measures for the audio quality:

Signal-to-Distortion Ratio (SDR)

In the project, the SDR is the main objective measure to evaluate the separation quality. It gives us an insight on how much information from $d(n)$ we effectively have on the result compared with the information related to the undesired signal $m(n)$. It is an estimate for the general quality of the separation process. The SDR can be defined as:

$$\text{SDR} = 10 \log_{10} \frac{\|s_{\text{source}}\|^2}{\|e_{\text{interf}} + e_{\text{artef}} + e_{\text{noise}}\|^2}. \tag{A.2}$$

Signal-to-Interference Ratio (SIR)

The SIR evaluates the quality of the separation measuring how much of the other sources have been inserted in the target source. It ignores the noise and the artefacts inserted by the separation method. It is possible to define the SIR as:

$$\mathrm{SIR} = 10\log_{10}\frac{||s_{\text{source}}||^2}{||e_{\text{interf}}||^2}. \tag{A.3}$$

Sources-to-Artefacts Ratio (SAR)

The SAR gives us an idea of the amount of defects have been inserted during the separation process. It is an estimate of how much stronger are the information of the different signals compared to the amount of included artefacts. It is defined as:

$$\mathrm{SAR} = 10\log_{10}\frac{||s_{\text{source}} + e_{\text{interf}} + e_{\text{noise}}||^2}{||e_{\text{artef}}||^2}. \tag{A.4}$$

All those metrics have been developed in [32] and a package with MATLAB implementations is publicly available in [33].

Bibliography

[1] STANKOVIĆ, S., OROVIĆ, I., SEJDIĆ, E. *Multimedia Signals and Systems.* New York, USA, Springer US, 2012.

[2] BYRD, D., CRAWFORD, T. “Problems of Music Information Retrieval in the Real World”, *Information Processing and Management: An International Journal*, v. 38, n. 2, pp. 249–272, March 2002.

[3] CHRISTENSEN, M., JAKOBSSON, A. *Multi-Pitch Estimation*, v. 5, *Synthesis Lectures on Speech and Audio Processing.* San Rafael, USA, Morgan & Claypool, 2009.

[4] GKIOKAS, A., KATSOUROS, V., CARAYANNIS, G., et al. “Music Tempo Estimation and Beat Tracking by Applying Source Separation and Metrical Relations”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, n. 37, pp. 421–424, Kyoto, Japan, March 2012.

[5] KLAPURI, A., VIRTANEN, T. “Automatic Music Transcription”. In: Havelock, D., Kuwano, S., Vorländer, M. (Eds.), *Handbook of Signal Processing in Acoustics*, cap. 20, pp. 277–303, New York, USA, Springer, 2008.

[6] HAYKIN, S. *Unsupervised Adaptive Filtering: Blind source separation*, v. 1. New York, USA, Wiley, 2000.

[7] UBC. “Guia Música em Audiovisual”. `http://www.ubc.org.br/Anexos/Publicacoes/ubc-guia-musica-audiovisual.pdf`. (in Portuguese), accessed in August 2018.

[8] LIUTKUS, A., LEVEAU, P. “Separation of Music+Effects Sound Track from Several International Versions of the Same Movie”. In: *AES Convention*, n. 128, London, UK, May 2010.

[9] SCHENKER, H. *Harmony.* Chicago, USA, University of Chicago Press, 1954.

[10] RAFII, A., PARDO, B. "REpeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 21, n. 1, pp. 71–82, January 2013.

[11] PEEBLES, P. Z. *Probability, Random Variables, and Random Signal Principles.* McGraw-Hill Series in Electrical and Computer Engineering. 4 ed. New York, USA, McGraw-Hill, 2000.

[12] DINIZ, P. S. R., DA SILVA, E. A. B., NETTO, S. L. *Digital Signal Processing: System Analysis and Design.* 2nd ed. Cambridge, UK, Cambridge University Press, 2010.

[13] LIUTKUS, A. E. A. "Adaptive Filtering for Music/Voice Separation exploiting the Repeating Musical Structure". In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, n. 37, pp. 53–56, Kyoto, Japan, March 2012.

[14] SOFIANOS, S. E. A. "H-Semantics: A Hybrid Approach to Singing Voice Separation", *Journal of the Audio Engineering Society*, v. 60, n. 10, pp. 831–841, October 2012.

[15] HYVARINEN, A., KARHUNEN, J., OJA, E. *Independent Component Analysis.* New York, USA, Wiley, 2001.

[16] RAFII, Z., DUAN, A., PARDO, B. "Combining Rhythm-Based and Pitch-Based Methods for Background and Melody Separation", *IEEE/ACM Transactions on Audio, Speech and Language Processing*, v. 22, n. 12, pp. 1884–1893, December 2014.

[17] SIMPSON, A. J. R., ROMA, G., PLUMBLEY, M. D. "Deep Karaoke: Extracting Vocals from Musical Mixtures Using a Convolutional Deep Neural Network". In: *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation*, n. 12, pp. 429–436, Liberec, Czech Republic, August 2015.

[18] JANSSON, A., HUMPHREY, E., MONTECCHIO, N., et al. "Singing Voice Separation With Deep U-Net Convolutional Networks". In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, n. 18, Suzhou, China, October 2017.

[19] GURURANI, S., LERCH, A. "Automatic Sample Detection In Polyphonic Music". In: *Proceedings of the International Society for Music Information*

Retrieval Conference (ISMIR), n. 18, pp. 264–271, Suzhou, China, October 2017.

[20] WHITNEY, J. L. *Automatic Recognition of Samples in Hip-Hop Music Through Non-Negative Matrix Factorization.* Master's thesis, University of Miami, Coral Gables, Florida, USA, 2013.

[21] VAN BALEN, J., HARO, M., SERRÀ, J. "Automatic Identification of Samples in Hip-Hop Music". In: *Proceedings of the International Symposium on Computer Music Modeling and Retrieval (CMMR)*, n. 9, pp. 544–551, London, UK, June 2012.

[22] VAN BALEN, J. *Automatic Recognition of Samples in Musical Audio.* Master's thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2011.

[23] BALUJA, S., COVELL, M. "Audio Fingerprinting: Combining Computer Vision & Data Stream Processing". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, n. 2, pp. 213–216, Honolulu, HI, USA, May 2007.

[24] HAITSMA, J., KALKER, T. "A Highly Robust Audio Fingerprinting System". In: *Proceedings of the International Society for Music Information Retrieval ISMIR*, n. 3, pp. 107–115, Paris, France, October 2002.

[25] WANG, A. "An Industrial-Strength Audio Search Algorithm". In: *Proceedings of the International Conference on Music Information Retrieval*, n. 4, pp. 7–13, Baltimore, MD, USA, October 2003.

[26] ZHU, B., LI, W., WANG, Z., et al. "A Novel Audio Fingerprinting Method Robust to Time Scale Modification and Pitch Shifting". In: *Proceedings of the ACM International Conference on Multimedia*, n. 18, pp. 987–990, Firenze, Italy, October 2010.

[27] VIRTANEN, T. "Unsupervised Learning Methods for Source Separation in Monaural Music Signals". In: Klapuri, A., Davy, M. (Eds.), *Signal Processing Methods for Music Transcription*, Springer, cap. 9, pp. 267–296, New York, USA, 2006.

[28] CANO, P., BATLLE, E., KALKER, T., et al. "A Review of Audio Fingerprinting", *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, v. 41, n. 3, pp. 271–284, 2005.

[29] BECH, S., ZACHAROV, N. *Perceptual Audio Evaluation - Theory, Method and Application.* Chichester, UK, Wiley, 2006.

[30] ITU-T RECOMMENDATION 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”. February 2001.

[31] ITU-R RECOMMENDATION BS.1387. “Method for Objective Measurements of Perceived Audio”. December 1998.

[32] VINCENT, E., GRIBONVAL, R., FÉVOTTE, C. “Performance Measurement in Blind Audio Source Separation”, *IEEE Transactions on Audio, Speech and Language Processing*, v. 14, n. 4, pp. 1462–1469, 2006.

[33] FÉVOTTE, C., GRIBONVAL, R., E., V. “BSS Eval: A Toolbox for Performance Measurement in (Blind) Source Separation”. `http://bass-db.gforge.inria.fr/bss_eval`. (in Portuguese), acessed in August 2018.

[34] TV GLOBO INTERNACIONAL. “Brazil Avenue”. `https://web.archive.org/web/20131029191020/http://www.globotvinternational.com/prodDet.asp?prodId=195&catId=1&random=1370297840562&random=1370298019796`. (Original title: Avenida Brasil), accessed in August 2018.

[35] WIKIPEDIA CONTRIBUTORS. “Avenida Brasil (TV series) — Soundtrack”. `https://en.wikipedia.org/w/index.php?title=Plagiarism&oldid=5139350`. Accessed in August 2018.

[36] ELLIS, D. “Robust Landmark-Based Audio Fingerprinting”. `http://www.ee.columbia.edu/ln/rosa/matlab/fingerprint/`. Acessed in August 2018.

[37] HAYKIN, S., VAN VEEN, B. *Signals and Systems*. 2nd ed. New York, USA, Wiley, 2002.

[38] LANGFORD, S. *Digital Audio Editing*. Burlington, USA, Focal Press, 2014.

[39] BRUNELLI, R. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.

[40] ESQUEF, P. A. A., BISCAINHO, L. W. P. “DSP Techniques for Enhancement of Sound Recordings”. In: Ahmad, A. M. A., Khalil, I. (Eds.), *Multimedia Transcoding and Wireless Networks*, Hershey: Idea Group, cap. 16, pp. 307 – 338, 2009.

[41] VASEGHI, S. V. *Algorithms for Restoration of Archived Gramophone Recordings*. Phd dissertation, Engineering Department, Cambridge University, Cambridge, England, 1988.

[42] HAYES, M. H. *Statistical Digital Signal Processing and Modeling.* New York, USA, John Wiley & Sons, 1996.

[43] DINIZ, P. S. R. *Adaptive Filtering: Algorithms and Practical Implementation.* 4 ed. New York, USA, Springer, 2013.

[44] REESE, D., GROSS, L., GROSS, B. *Audio Production Worktext: Concepts, Techniques and Equipment.* Focal Press, 2009.

[45] FOLLANSBEE, J. *Hands-On Guide to Streaming Media: An Introduction to Delivering On-Demand Media.* Focal Press, 2006.

[46] ÁVILA. F. R., TCHEOU, M. P., BISCAINHO, L. W. P. "Audio Soft Declipping Based on Constrained Weighted Least Squares", *IEEE Signal Processing Letters*, v. 24, n. 9, September 2017.

[47] AVILA, F., BISCAINHO, L. W. P. "Audio Soft Declipping Based On Weighted L1-norm". In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 299–303, New Paltz, NY, October 2017.

[48] DUARTE, L. T., SUYAMA, R., ATTUX, R., et al. "A Sparsity-Based Method for Blind Compensation of a Memoryless Nonlinear Distortion: Application to Ion-Selective Electrodes", *IEEE Sensors Journal*, v. 15, n. 5.

[49] LORDELO, C. "Link with the Implemented Algorithms for Automatic Removal of Music Tracks from TV Programmes". `http://www.smt.ufrj.br/~carlos.lordelo/Masters/`. Accessed in September 2018.