Complex NMF for Multichannel Source Separation

by

Tung Nguyen, **B.Sc.**

A thesis submitted to the
Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
December 6, 2016

Abstract

Recently, a new tool known as Nonnegative Matrix Factorization (NMF) has presented itself as a formidable and useful tool for providing a parts based representation of matrix data. It has been applied with success in audio signal processing for topics such as blind source separation (BSS), music transcription and for representing musical and/or speech mixtures as additively occurring nonnegative representations of audio components. In the STFT domain, this is due to the fact that Fourier coefficients can be stored and processed as either matrices or tensors, and the additive mixtures of sounds can be further parametrized and factorized in some way to output a parts based time-frequency representation of the sound mixtures. In this thesis, we consider both speech mixtures and musical mixtures, as valid types of mixtures to be separated by the proposed algorithm. This thesis presents research and a proposed algorithm that addresses the problem of underdetermined multichannel frequency domain BSS, and investigates spatial covariance matrix (SCM) based NMF and single channel CNMF algorithms as applied to complex (as opposed to nonnegative) STFT coefficients. The research also investigates K-means clustering applied to interchannel frequency dependent phase differences in order to achieve source separation using SCM NMF based techniques.

Keywords: Nonnegative Matrix Factorization, Complex Matrix Factorization, Spatial Covariance Matrices, Underdetermined multi-channel source separation, Blind Source Separation, multi-channel STFT

Dedicated to the artists, musicians and mathematicians of the world, and all other greats minds that came before us.

Acknowledgments

I would like to extend my greatest thanks and gratitude first and foremost to my parents, former teachers, colleagues, friends and family, and thesis supervisor, Dr. Richard Dansereau.

I would like to thank Dr. Dansereau for his patience, confidence and willingness to meet and challenge my ideas for the betterment of the quality of my work. I would like to thank my parents for a long list of things, but I can only highlight for now some of the important ones. I thank you for bringing me into the world, teaching me the qualities that we value, for raising me into an adult, and for influencing me with the motivation and confidence to believe that I can achieve nearly any task that I put my focus towards. Lastly, thank you for your love, and your unwavering support for my academic and career pursuits. To my sister Nhung, thank you for being who you are, for encouraging me to journey down this road, and thank you in large part for influencing me to a large degree into who I am today.

To everyone I have mentioned, I thank you for the support and encouragement, the opportunity to converse and confide in each other through the odd and sometimes seemingly rough times that were my entire academic career to date. To those who helped me set goals for myself in life, and instilled the attitude in me to strive to achieve them, I thank you. To those who sparked the curiosity and showed me the tools that could be used to carve out solutions to the problems that we encounter in our field of work, I also thank you.

To former professors, senior, junior and department colleagues, those who I admire, and strive to learn from each and every day, thank you for your willingness to converse and share your knowledge.

To my close friends and any friend that I have come across in my life, I thank you for your kind will, your effort towards creating or maintaining a friendship, for treating me kindly, and for being the reflection that mirrors me in such a way that makes believe that I am someone who is worthy of friendship.

To the Ottawa-Carleton Institute for Electrical and Computer Engineering I thank you for providing everything that you have provided that has allowed me to concentrate on my research in a professional, helpful and friendly and exciting work environment.

To anyone currently reading this, I thank you and hope to meet with you again some day.

Table of Contents

List of Tables

List of Figures

Nomenclature

Table 0.1: Some Mathematical Symbols and Operators

$\mathbb{R}$	Real Number
$\mathbb{C}$	Complex Number
$\text{Re}\{z\}$	Real part of complex number $z \in \mathbb{C}$
$\text{Im}\{z\}$	Imaginary part of complex number $z \in \mathbb{C}$
z^*	Complex conjugate of scalar complex number $z, z^* \in \mathbb{C}$
$\arg\{z\}$	Complex argument of complex number $z \in \mathbb{C}$, where $-\pi \leq \arg\{z\} \leq \pi$
$\mathbb{R}^n$	n-dimensional real vector space
$\mathbb{C}^n$	n-dimensional complex vector space
$\underset{\theta}{\text{argmin}} J(\theta)$	denotes that we seek to optimize $J(\theta)$ for the value of the parameter θ that minimizes it
$D(\mathbf{A}\|\|\mathbf{B})$	divergence measure between two matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbb{E}[\cdot]$	Expectation operator
$\exp\{\cdot\}$	exponential function
$\log\{\cdot\}, \ln\{\cdot\}$	natural logarithm function
$\text{sign}(x)$	signum function
$\text{tr}(\mathbf{A})$	trace of the matrix $\mathbf{A}$
$[\cdot]^T$	transpose of a matrix or vector quantity
$[\cdot]^H$	Hermitian transpose of a complex matrix or vector quantity
$\circledast$ or $.*$	Hadamard product (elementwise multiplication)
$\mathbf{a} \circ \mathbf{b}$	Outer product of real column vectors $\mathbf{a} \in \mathbb{R}^N$ $\mathbf{b} \in \mathbb{R}^M$ resulting in matrix of size $N \times M$
$\oslash$ or $./$	Elementwise division
$\otimes$	Kronecker product ($\mathbf{A} \otimes \mathbf{B} = [a_{ij}\mathbf{B}]$, See Appendix), alternatively $\text{kron}(\mathbf{A}, \mathbf{B})$
δ_{ij}	Kronecker delta
$\|\|\mathbf{x}\|\|_p$	p-norm of $\mathbf{x}$

Table 0.2: Some Abbreviations

BSS	Blind Source Separation
CMF	Complex Matrix Factorization
DFT	Discrete Fourier Transform
DOA	Direction of Arrival
DSP	Digital Signal Processing
EG	Exponentiated Gradient Update Rule
EM	Expectation Maximization
FD	Frequency Domain
FD-BSS	Frequency Domain Blind Source Separation
ICA	Independent Component Analysis
IS	Itakura-Saito Divergence
KL	Kullback-Leibler Divergence
LS	Least Squares
ML	Maximum Likelihood
MM	Majorization Minimization
MU	Multiplicative Update Rule
NMF	Nonnegative Matrix Factorization
NTF	Nonnegative Tensor Factorization
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
SAR	Signal to Artifact Ratio
SCM	Spatial Covariance Matrix
SDR	Signal to Distortion Ratio
SIR	Signal to Interference Ratio
SNR	Signal to Noise Ratio
STFT	Short-Time Fourier Transform
TDOA	Time-Difference of Arrival
UBSS	Underdetermined Blind Source Separation

Chapter 1

Introduction

Nonnegative matrix factorization (NMF) [1] can be considered a sibling to the more statistically intensive Expectation Maximization (EM) algorithm [2–5]. The original NMF (Lee and Seung) provided a generic optimization framework which was described as being solely dependent on modelling an observed matrix as being the result of the multiplication of two basis dictionary (or factor) matrices. The usefulness of this general optimization framework was evidenced by a considerable number of algorithms that spawned shortly after in its likeness [6, 7] which demonstrated that NMF could be applied as a tool in a diverse number of different possible applications.

We propose in this thesis, that NMF algorithms at this point in time are well developed enough that they should be attractive methods to use in attempting to solve ongoing research in blind source separation and blind signal processing [8, 9] since contemporary signal processing techniques are already becoming more interrelated with advanced mathematical optimization techniques [10–13] such as machine learning [14] and pattern classification [4, 15]. Specifically we propose that it should be applied in particular to the area of blind source separation with respect to multichannel mixtures of audio signals. In the work by Sawada [16], spatial covariance matrix (SCM) based NMF was used where the SCM matrix per time frequency bin was modelled by an interchannel frequency dependent mixing covariance matrix multiplied by a source component variance term in the likeness of the multichannel Wiener filter [17].

With regards to currently available algorithms, there seem to be a specific class of algorithms that do best at resolving undetermined mixtures of audio spectrogram data in the short-time Fourier transform (STFT) domain, and this class of algorithms is known by the name of *spatial covariance* based source separation algorithms [18–23]. In this thesis we focus on studying in particular two fairly recent spatial covariance-based algorithms [16, 24] that are also formulated upon the key principle of NMF, which are derived based upon a

Euclidean distance based formulation of the observed multichannel time-frequency spectrogram. There in fact exists a third relevant algorithm, detailed in [25], that corresponds to a single channel NMF algorithm, by the name of *complex* nonnegative matrix factorization (CNMF), that has also to a large degree inspired the content of this thesis.

Typically time frequency audio representations apply the basic NMF model for modelling audio spectrograms, in such a way that two nonnegative factor matrices are used to model the source component variance term. Respectively, these included a dictionary consisting of harmonic (frequency dependent) basis column vectors as well as a dictionary of row vectors consisting of time activations (for STFT frames) for each NMF component.

Upon building the foundations of the NMF principle, the thesis will also connect various useful and relevant results having to do with multichannel source separation, such as *spatial* filtering, sound propagation of audio components with respect to multiple sources and multiple microphones, and even audio beamforming [26, 27]. For instance, in the subject of beamforming, having multiple microphones and knowing the *geometrical* arrangement of these microphones is highly beneficial to algorithmically localizing and determining the most likely location of sources in the surrounding acoustic environment. It will be a focus of the thesis to show that such an approach can be used to inspire the development of algorithms in audio source separation with NMF by treating the observed data as being the result of multiplicative and additive operations in the *complex* STFT domain. A focused effort will be made to *classify* various parameters within the proposed algorithm that model the multichannel and complex-valued STFT spectra of estimated source parameters and to reconstruct estimated source signals from their parametrization, corresponding as closely to their actual spatial locations in the surrounding acoustic environment as possible. Furthermore, matters of how to optimally organize sound components of competing and interfering output classes according to a multichannel filtering and spatial propagation scenario will be addressed.

Due to the flexibility of NMF in defining the model in terms of a matrix parameter set that characterizes the representation that it provides with respect to the source separation algorithm, it will also be a focus to demonstrate that *data clustering* can be used in a flexible way to determine an enhanced spatial estimate of the correct configuration of source positions in the surrounding acoustic environment aligned and optimally classified across all output classes and STFT time frequency bins.

1.1 Hypothesis

In this thesis an algorithm for determining a *complex* NMF (CNMF) representation of multi-channel audio spectrogram data in the STFT domain is proposed and the hypothesis that time difference of arrival and learning techniques such as data clustering can be used to augment the proposed CNMF audio source separation algorithm is to be tested.

1.2 Problem Statement and Objectives

To set clearly the requirement of what must be returned by the algorithm, given what is to be provided as input to the algorithm; the CNMF representation must return a set of signals (indexed by their class index), where the total number of output signals and classes is to be chosen freely (but reasonably) by the user of the algorithm. The number of classes should be preferably chosen to be equal to the *true* number of sound sources spatially present in the multi-channel mixture in order to avoid aliasing (i.e. overlapping) of STFT spectra in the output signals computed by the algorithm. If the target number of output classes is chosen to be greater than the number of microphone recordings, then the problem to be solved is to be labelled as an *underdetermined source separation* problem. To be demonstrated is that provided with each source separation algorithm is a corresponding source separation (reconstruction, or re-synthesis step) that recovers what is intended to be a close estimate of the true (unmixed) source signals, and/or their corresponding (filtered) spatial images, that are to be reconstructed on the basis of a parts based parametrization according to the appropriate cluster/class label per source. In other words, the parametrization can be interpreted to also provide a decomposition of audio components occurring in the time-frequency domain that are to some degree also parametrized spatially according to their class membership.

Also to be demonstrated is that the way to best parametrize and reveal the common spatial characteristics, in a multi-channel sense, is having to do with the estimated microphone *time differences*, which are frequency dependent (when considering how the phase part of the observed microphone STFT coefficients were generated) and have a frequency domain interpretation related to the physical geometry and dimensions of a microphone array [24], if one assumes that the true spatial locations of the sources remain stationary and that the far-field model for sound propagation is a valid approximation of the observed and recorded multi-channel sound mixture. When considering interchannel microphone mixtures, the time differences (also known as delays) can be converted between the time domain and frequency domain via the time shifting property of the Fourier transform.

Using the far-field model for sound propagation [17, 28] it is fairly easy to express time difference of arrival quantities in terms of a frequency-domain representation, and how this is true is intended to be shown in a detailed manner, at the beginning of the next chapter. Throughout the thesis and in presenting the proposed algorithm we will seek to demonstrate how it can be applied to developing spatially intelligible source separation algorithms, as was suggested by the development of an SCM CNMF algorithm as proposed in [24], which introduced an NMF parameter whose purpose was to parametrize *spatial weights* associated with *directions of arrival*, relative to the microphone array, thus hinting at the most *plausible* locations of active sources in an acoustic environment with multiple sources. We shall emphasize that this is one of the key concepts that will allow the source estimation computed by proposed algorithm as well as the algorithm in [24] to perform better than some existing algorithms that do not adequately utilize a spatial parametrization of the source separation problem in deriving an NMF source separation algorithm.

The STFT and especially the multichannel STFT computed on a spatially diverse set of microphone signals captured from a microphone array containing multiple microphones, will be shown as a sufficient tool for analyzing both the spatial and spectral time frequency behaviour of source signals within an acoustic environment for localizing and separating source signals within a spatially motivated source separation algorithm. We will exploit methods for providing *spatial directivity* of the microphone array by considering principles that fundamentally describe the geometrical parametrization of source and microphone positions (i.e. source and sink positions) within an acoustic environment.

We also propose that a key algorithmic *difference* between existing multichannel STFT modelling algorithms, such as the ones presented in [16, 24], and the proposed algorithm shall have mostly to do with an added requirement of parametrizing the *initial phase* of the source estimates. We focus on *augmenting* the typical requirement of audio NMF models, which is to parametrize additive audio components using mainly a two-factor interpretation of the magnitude spectra, characterized for all sources across all time-frequency bins, to a two-factor plus phase *dictionary*, as first suggested could be done in the research by Kameoka et al. [25] that termed the phrase 'complex NMF' (CNMF) to algorithms that were able to accurately model phase spectra of audio signals represented in terms of their complex STFT spectra.

Therefore, it can now be stated explicitly, that a "sub-hypothesis" of the thesis is that the proposed algorithm also seeks to show that proper estimation of the "initial phases" of learned sound components should also lead to improved performance gains and sound intelligibility in the output of the multichannel sound separation algorithm. Recent, but separate

research as conducted in [29] emphasizes (and enforces its own set of constraints) that in order to produce perceptually pleasant sounding separated output signals, the accurate STFT *phase recovery* of the signals, classified according to their output class, must be taken into consideration as part of the source separation and reconstruction step.

It will be the sole point of Chapter 5 to emphasize and to demonstrate how the ideas presented in [16,24,25] can be unified with the objective of providing both a spatially intelligible multichannel algorithm, as well as a source-phase accurate algorithm capable of parametrizing all the appropriate spectral and spatial characteristics that should be associated with sources in order to reconstruct the best possible estimates of the true (but unknown and unobserved) sources. In Chapter 6, we will verify and present the development of the proposed algorithm, in order to determine whether or not the objective seems to have been achieved.

The short and long term goal of the research carried out in this thesis is to be able to demonstrate the flexibility, power, and usefulness of NMF as a tool for signal processing in audio.

1.3 Background

1.3.1 Broad Classification of Topics in Digital Signal Processing and Mathematical Optimization

Figure 1.1 provides an overview of prominent techniques used in signal processing algorithms [30]. At the top level of the tree, we classify algorithms as being either transform based, or model based. The illustration also allows for the possibility of a particular algorithm to borrow certain aspects from various different points within the tree.

Ordinarily, for audio signal processing, using a transform based method such as the Fourier transform provides adequate insight into the short time behaviour and frequency dependent periodicity of a collection of digital audio samples obtained by windowing the audio signal using a windowing function. As the short time characteristics of the signal inevitably vary as a function of time, the problems and solutions associated to explaining the the time variations present in audio captures converted to the STFT domain typically become much more potentially complex and challenging. This will especially be the case when a desired signal component or a desired signal parametrization (i.e. representation) is in fact unobserved and must be perhaps *inferred* by some algorithmic method. In order to approach the challenges of explaining spectral behaviour of signal variation across time and frequency, within adverse conditions such as noise and interference, one might begin to draw

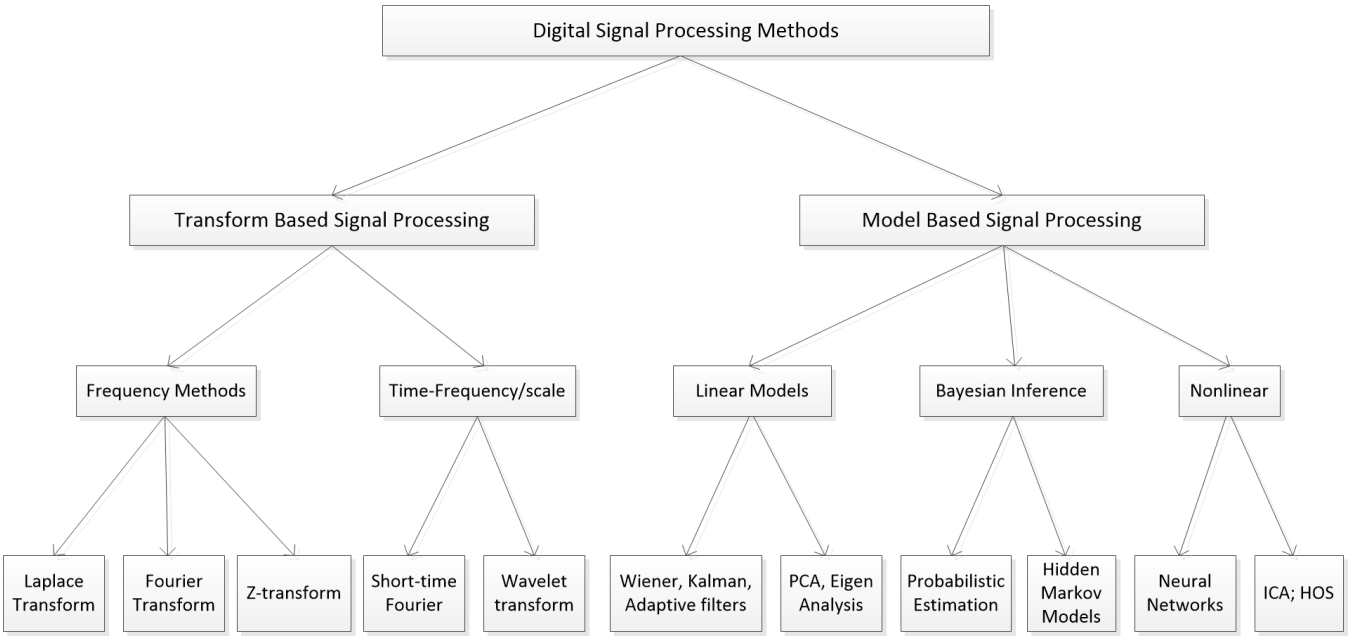

Figure 1.1: Classification of Digital Signal Processing Methods

upon tools from mathematical optimization such as multivariable calculus, linear algebra, and numerical optimization [31–34] which focus on the study of functions, derivatives, and the development of elegant solutions and/or algorithms for computing the unknown quantities surrounding a difficult problem such as the mixing of complex-valued STFT spectra. A treatment of complex-valued vector spaces, functions, and derivatives defined over them is provided within [35], and can be used for further guidance in developing useful algorithms under the notion of complex-valued data.

The approach towards Maximum Likelihood (ML), EM (a MM based procedure towards solving ML problems), and K-means provided within the text by Bishop on pattern recognition [4] are concepts that are useful towards understanding the original NMF formulation and its extensions. In a similar vein, convex optimization [13] algorithms focus on defining a calculus over convex sets and convex functions, in order to exploit and apply the usage of functions having the special property of being convex, such as quadratic and logarithmic functions, encountered commonly in (but not only limited to) various signal processing subjects. NMF based derivations typically the study and careful consideration of so called distances and/or divergence measures defined over matrices and in some cases, tensors, in order to develop algorithm variants capable of optimally explaining an observed matrix and to algorithmically force a corresponding *parametrization* of the observed matrix to converge towards the observed matrix, element-wise, in some optimal sense. Therefore, NMF algorithms, and a justification for their usefulness (e.g. convergence behaviour) can be shown

to resemble that of certain EM, pattern classification, and convex optimization algorithms, in that nearly all of these classes of algorithms exploit the convexity of their optimization criterion (cost functions).

Further considering Figure 1.1, we highlight some of the subjects that are most relevant to the research to be presented within this thesis. The proposed algorithm, by virtue of operating in the short-time Fourier transform domain, is designed to manipulate complex-valued Fourier transform coefficients, and we intend to derive a model for the multichannel STFT that is in some part complex and in some part nonnegative, in terms of the matrix parameters that characterize the algorithm's model. For the reconstruction of parametrized source signals from observed data, we examine in principle the Wiener filter (notably its frequency-domain formulation) and in particular the multichannel Wiener filter, within the context of how it is used for optimally filtering and extracting desired (i.e. true) signals and/or signal components given a set of multichannel STFT array recordings. Since the multichannel Wiener filter does not necessarily prescribe to simultaneously exploit SCM NMF and or CNMF methods, the proposed algorithm, which can be viewed as a hybrid between SCM NMF and CNMF based methods, will be compared against the state of the art algorithms that exploit the SCM NMF based method combined with the multichannel Wiener filter based reconstruction step.

1.3.2 Blind Source Separation and Nonnegative Matrix Factorization

Different types of matrix factorizations and their associated algorithms have been prevalent in various mathematical, signal processing and linear algebra literatures for several decades [12, 32]. Among these factorizations which are most useful and likely to be the best understood include the Singular Value Decomposition (SVD), Principle Component Analysis (PCA), and the LU, QR and Cholesky decompositions [32]. These algorithms are commonly used as baseline algorithms for comparing to NMF and for making analogies to, with respect to NMF.

Recently, Nonnegative Matrix Factorization (NMF) has become a new and important subject in the area of signal processing and in the development of unsupervised learning algorithms for explaining mixtures of audio data in the time-frequency domain. NMF as well as some classes of algorithms related to it such as blind source separation (BSS) and various matrix component analysis algorithms (e.g. ICA, PCA, or GCA - generalized component analysis) must develop a set of learning rules for the purpose of achieving this end goal of being able to explain the observed data mixtures. It is for this reason that NMF, BSS and

various classes of algorithms similar to them are often classified as machine learning (ML) or small-scale artificial intelligence (AI) algorithms.

In the text by Cichocki et al. [7] a unifying treatment of these classes of algorithms is provided with a set of examples that demonstrate some currently existing applications for the algorithms.

With the focus of the thesis being centered on audio source separation in the frequency domain with the added challenge of achieving a source separation with less available microphone signals than the number of source signals (i.e. underdetermined source separation) we focus on studying techniques that exploit the spatial diversity (e.g. time difference of arrival information) in a blind source separation context, best suited to providing adequate underdetermined separation of either speech or music source signals.

While state of the art techniques in blind source separation (BSS) [36, 37] do offer relevant techniques for the modelling and extraction of key source signal features in the time-frequency domain that are compatible with a TDoA model, these techniques do sometimes rely upon ICA based methods which have been known to have some limitations in jointly uniting estimated TDoA's of parametrized source signals in the frequency domain across all frequencies of the multichannel mixture model [24]. Such is the case and is evident for any algorithm that requires a 'permutation alignment' scheme such as the methods presented in [38] and [39].

While state of the art NMF and Blind source separation algorithms typically find elucidation in the problem if we consider the most key issue, which is that we must absolutely find a transformation or coding of the observed data which has true physical meaning or interpretation.

Furthermore, as mentioned as one of the possible goals within the list, if we are able to exploit or infer a priori information about the mixture, then this can eventually greatly benefit us, in aiding towards the development of the problem solution.

In this thesis we will leverage a set of highlighted concepts from existing state of the art source separation algorithms in order to propose an allegedly enhanced algorithm, derived upon the principle of *majorization minimization* [40], which is an optimization principle that is explained as being relevant to developing to most EM and NMF-based optimization frameworks.

1.3.2.1 Basic NMF Model

We now introduce the Basic NMF as first researched early on by Paatero and Tapper [41] and later popularized by Lee and Seung [1, 42] once it was shown that it could be applied to factorizing image data.

We specify the problem formulation of the basic NMF model, which is illustrated within Figure 1.2.

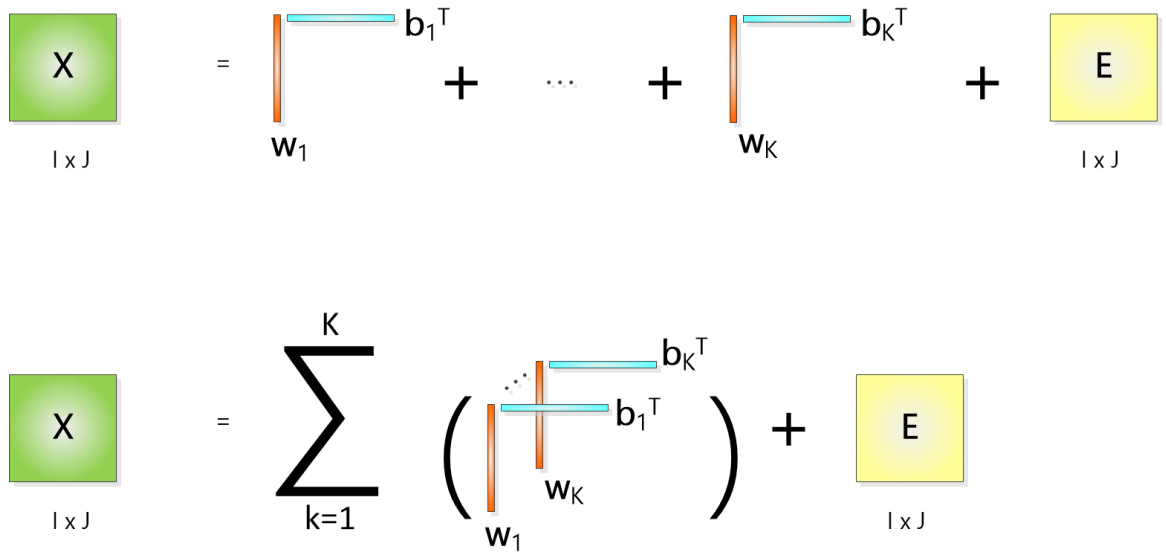

Figure 1.2: Basic NMF Model Illustration

Given the observed nonnegative data matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$, find two nonnegative factor matrix parameters

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K] \in \mathbb{R}^{I \times K}$$

and

$$\mathbf{H} = \mathbf{B}^T = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_K]^T \in \mathbb{R}^{K \times J}$$

that factorizes the observed matrix $\mathbf{X}$ as well as possible such that

$$\mathbf{X} = \mathbf{WH} + \mathbf{E} = \mathbf{WB}^T + \mathbf{E}. \tag{1.1}$$

The interpretations of the factors $\mathbf{W}$ and $\mathbf{H}$ have different meanings depending on the application or the type of data that the model represents. For instance, in a BSS problem, $\mathbf{W}$ represents the mixing matrix where as $\mathbf{H}$ represents the source signals.

In clustering problems $\mathbf{W}$ is the basis matrix and $\mathbf{H}$ denotes the weighting matrix.

In time-frequency analysis of the magnitude spectra of audio signals $\mathbf{W}$ represents the

matrix of spectrally redundant basis patterns, while elements of row vectors in $\mathbf{H}$ (that is, the row vectors $\mathbf{b}_k^T$) represent time activation bins at which the spectral sound patterns are activated.

An alternative representation of the model can be expressed as given by

$$\mathbf{X} = \sum_{k=1}^{K} \mathbf{w}_k \circ \mathbf{b}_k + \mathbf{E} = \sum_{k=1}^{K} \mathbf{w}_k \mathbf{b}_k^T + \mathbf{E}. \tag{1.2}$$

The usefulness of the basic NMF as described by equations 1.1 and 1.2 is found in terms of developing optimization rules that can explain the components of an observed nonnegative data matrix $\mathbf{X}$ in terms of a decomposition of rank-one nonnegative matrices $\mathbf{w}_k \mathbf{b}_k^T$. Furthermore, proceeding under the working assumption of the observed data and the parametrized matrix decomposition being entirely nonnegative is useful in terms of providing straightforward physical interpretations conveying important patterns and possible meaning behind the data that the NMF algorithm is able to reveal. Furthermore, depending on the divergence measure and optimization technique used to derive the update rules, we typically are able to develop easy to implement learning rules for iteratively updating the matrix parameters, that are well-defined in terms of typical linear algebra and matrix computations.

1.3.3 Example Applications of Nonnegative Matrix Factorization

1.3.3.1 Music Transcription

NMF algorithm variants have been successfully applied to the algorithmic decomposition problem of music transcription [6], score informed music processing [43–46], harmonic modelling of musical audio components [47], and parametric modelling of music signals [48]. Although we don't focus on these models in the thesis, we acknowledge that NMF has the flexibility to be extended for the purpose of describing time-frequency parametrizations of musical signals in terms of score and the musical scale as evidenced by these particular algorithms. Furthermore, the evidence of NMF being applied to this problem and to this area of research illustrates the concept of NMF naturally being a compatible tool for time-frequency analysis since it is demonstrated to be able to provide parametrizations of spectrally redundant basis patterns as contained within a dictionary matrix of harmonic comb vectors that are intended to approximately model the spectral envelope of musical notes (basis vectors), while also providing a parametrization of the occurrence of time activation sequences conveyed within a so-called time activation matrix.

1.3.3.2 Data Clustering and Text Mining

Document clustering and text mining represent another class of applications in which the usage of NMF algorithms can be applied [7, 49].

Existing methods that have been applied to the subject of document clustering include Latent Semantic Indexing (LSI) but recently NMF methods have been applied [7] in order to achieve the common objective of providing a *clustered* representation of an observed matrix $\mathbf{Y}$ in terms of a basis dictionary matrix $\mathbf{A}$ and an indicator matrix $\mathbf{X}$.

Given a set of documents, where the total number of documents is equal to a fixed number T, we intend to classify (i.e. cluster) the documents based upon similarity of their content. To do so suppose we invent a list of relevant terms where the total number of terms is equal to a fixed number I.

We then set up the matrix $\mathbf{Y}$ such that $\mathbf{Y} \in \mathbb{R}^{I \times T}$ where the nonnegative elements of the matrix $\mathbf{Y}$ represent relative frequencies of the terms (distinct words, or short sequences of words) labelled with index i occurring within the tth document and are populated as given by

$$y_{it} = F_{it} \log\left(\frac{T}{T_i}\right) \tag{1.3}$$

where F_{it} represents the *number of times* the ith term appears in document t, T_i represents the number of documents that contain the ith term, and T represents the total number of documents and the total number of columns of $\mathbf{Y}$. We note that in general y_{it} is typically nonnegative and has two possible ways for the i-tth element to be zero. The first is if its relative frequency F_{it} is zero and the second is if the ith term appears in *all* the documents, that is $T_i = T$ and thus $\log\left(\frac{T}{T_i}\right) = 0$, which in fact sets the entire ith row to zero.

Populating the matrix $\mathbf{Y}$ in this manner can be understood then to generate patterns within the columns of the matrix $\mathbf{Y}$, where we should recall that the total number of columns is equal to the total number of documents T and that we intend to *cluster* the columns, if possible, in such a way that associates (identifies) similarities between groups of documents. In order to achieve this, let us specify a maximum number of *topics* K, and consider that the basis matrix $\mathbf{A}$ should have K total columns and that the indicator matrix $\mathbf{X}$ should have K total rows, which subsequently leads to the factorization

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \tag{1.4}$$

which is a factorization of the matrix of *term-dependent* relative frequency patterns conveyed in the column vectors of $\mathbf{Y}$ per document t represented in terms of the matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{X} \in \mathbb{R}^{K \times T}$. Here, the basis matrix $\mathbf{A}$ attempts to model an appropriate

pattern of relative frequencies corresponding to the kth 'topic', per column of $\mathbf{A}$, and the cluster indicator matrix $\mathbf{X}$ from this set of basis patterns is able to convey a classification of the documents, in terms of the dictionary of topic vectors. Typically, if $\mathbf{X}$ has the property of being orthogonal, that is, $\mathbf{X}\mathbf{X}^T \approx I_K$, where I_K is the $K \times K$ identity matrix, then the maximum-valued element per tth column of $\mathbf{X}$ will convey with more probability (certainty) that a certain document t should belong to the cluster k at which the maximum occurs. Typically, the factorization is also more meaningful if any of the matrices $\mathbf{Y}$, $\mathbf{A}$, and $\mathbf{X}$ have the property of containing some or many zeros (i.e. have the property of being *sparse*). Further topics in applying NMF to the topic of text mining were described in [49].

Furthermore, there exist various *constraints* that can be applied to NMF models that allow for more meaningful interpretations of the NMF factors such as *sparsity* [50] and *orthogonality* [51], which correspond to only two.

In focusing on developing multiplicative update algorithms applied to the time-frequency domain representations of audio data, it will be in fact important to be comfortable with the task of sub-indexing the various matrix and tensor parameter quantities that in some way form the output time-frequency domain NMF model. It is for this reason that within the appendix we include various nonnegative *tensor* factorization at section A.7, for the purpose of introducing this concept. These NTF (nonnegative tensor) decompositions were provided in the text by [7] which details some of their actual corresponding applications.

Since tensor parameters do occur in the proposed research we suggest that the reader be at least familiar with this concept. The reason for this is that sub-indexing the matrix and/or tensor parameters in practice and in principle typically corresponds to implementing multiplicative updates on in terms of software, and in terms of matrix computing libraries such as Armadillo C++ [52] and MATLAB.

1.4 Multi-Sensor and Multi-Source Problem Description

In order to fully elucidate the problem description of the *multichannel* (i.e. multi-sensor) source separation problem with regards to time domain signals analyzed according to their STFT we will consider the following convention of labelling sources signals $s_l[n]$, spatial image signals $y_{ml}[n]$, and observed microphone signals $x_m[n]$ in the development of the thesis. We emphasize that the following convention is a fairly typical and commonly encountered convention for the labelling and description of the sensor and source signals. The description is provided for clarity and explicitness, and for the purpose of unifying or distinguishing the

details a particular algorithm from other algorithms, in how certain characteristics of the signals to be separated (estimated) are in obtained in practice.

We first consider a set of unavailable source signals

$$\{s_1[n], s_2[n], ..., s_L[n]\} \qquad \text{for } l = 1, .., L \tag{1.5}$$

for which we would develop an algorithm to algorithmically determine a set of *estimated* source signals

$$\{\hat{s}_1[n], \hat{s}_2[n], ..., \hat{s}_L[n]\} \qquad \text{for } l = 1, .., L \tag{1.6}$$

that accurately model the equivalent effect of the *true* source signals $\{s_l[n]\}$ in generating the observed signals $x_m[n]$, and by specifying only to the algorithm the total number of source signals L that we think are present within the room environment.

We propose that the *spatial filtering* of the true source signals occurs according a discrete-time convolution is given by

$$\begin{aligned} y_{ml}[n] &= s_l[n] * h_{ml}[n] \qquad \text{for } m = 1, ..., M,\ l = 1, ..., L \\ &= \sum_{p=-\infty}^{\infty} s_l[p] h_{ml}[n-p] \end{aligned} \tag{1.7}$$

where the signal $y_{ml}[n]$ corresponds to the spatial image, seen as the output of the convolution between the lth unknown source signal $s_l[n]$ and its acoustic impulse response to microphone m, $h_{ml}[n]$. Depending on the main peak within $h_{ml}[n]$ there is an associated and quantifiable delay that can be associate with each $h_{ml}[n]$.

Additively combining all spatial images for the total number of unknown sources, per microphone, each microphone signal can be seen as the sum of all of its spatial images $y_{ml}[n]$ for $l = 1, ..., L$ as given by

$$x_m[n] = \sum_{l=1}^{L} y_{ml}[n] \qquad \text{for } m = 1, ..., M. \tag{1.8}$$

where since equation 1.8 represents the sum of L signals, each of the y_{ml} may not be recoverable and/or distinguishable to a human listener. The underdetermined source separation problem then in fact, requires applying an appropriate algorithm to recover the unobserved spatial images y_{ml} solely given the *available signals* $x_m[n]$. As will be seen within constructing synthetic mixtures of the $x_m[n]$ using room simulation software, we will exploit the fact

that if the spatial images y_{ml} are not too overlapped in time and frequency and occur from distinct *spatial locations* (i.e. *angles of arrival*) then the task of separating them may be in fact eased immensely. We now consider the frequency domain interpretation of the spatial filtering parameters as relevant to providing a time-frequency domain interpretation of the addition of spatial images, resulting in observed signals $x_m[n]$.

Considering again equation 1.7 we note that the DFT of the spatial filters representing impulse responses $h_{ml}[n]$ can be expressed in the frequency domain as

$$h_{ml}(k) \qquad k = 0, ..., N-1 \tag{1.9}$$

and can be considered as a simple linear-time invariant (LTI) filter that is only frequency dependent (k are the discrete frequencies of the DFT) and hence not time-varying (and thus not dependent on the STFT frame index r), since we specified that the algorithm requires that source positions of each source remain spatially stationary. Thus, in short, equation 1.9 specifies the acoustic transfer function between the unknown but stationary source l and microphone m.

The mth microphone signal's STFT shall be computed as

$$X_m[r,k] \xleftarrow[STFT]{} x_m[n] \tag{1.10}$$

We emphasize here, that equation 1.10, as described in this manner corresponds to the notion of a *multichannel STFT* where $X_m[r,k]$ denotes the multichannel STFT at channel m and the variables k and r respectively index the STFT's frequency and frame bins (i.e. time frequency bins). In the next section we will demonstrate the STFT analysis of real audio signals.

If an NMF, CNMF, or SCM NMF algorithm is capable of algorithmically resolving, in the STFT domain, the spatial effect of equation 1.9, then we propose that the source reconstruction of estimated source signals, according to equation 1.6, becomes simple to achieve. Therefore, in Chapter 2, a great deal of effort will be spent towards thoroughly studying the principle of spatial filtering applied to multi-sensor microphone arrays. It is suggested that, following having considered Chapter 2, the reader also consider Appendix B for an appropriate illustration overview of how *complex* STFT coefficients are manipulated computationally by a multichannel source separation algorithm. Apart from this illustration, located specifically at section B.2.1, the rest of this appendix includes a standard description of the DFT, STFT and single channel Wiener filter.

Since the concepts of spatial filtering are embedded into the concept of the multichannel

Wiener filter, it is in fact quite natural to utilize the multichannel Wiener filter for reconstruction purposes as a baseline algorithm, for which other source separation algorithms should be compared to. The multichannel Wiener filter can be shown to be MMSE optimal in a multi sensor and multi source sense. An advanced question, and one that we would like to consider, is how does the multichannel Wiener filtering principle, upon which most SCM NMF algorithms are based, perform when the source signals are greatly overlapped in either time and/or frequency. Thus the proposed algorithm, will deviate from exactly a multichannel Wiener filtering inspired parametrization of the SCM NMF algorithms encountered in the literature, and will be shown to be parametrized differently than how the multichannel Wiener filter exactly prescribes.

In Chapter 5, we will consider state of the art algorithms that directly exploit the principle of spatial filtering with multi-sensor microphone arrays by applying the far-field sound propagation model to the SCM NMF model. This state of the art algorithm will hereby be referred to as the DOA SCM NMF algorithm [24] and is credited to the research of Nikunen and Virtanen.

1.5 Example of Multichannel STFT Analysis with Real Speech and Music signals

Figure 1.3 demonstrates the STFT equation applied to 10 seconds of audio corresponding to male speech. We note that the magnitude spectrogram has a highly dynamic and information rich behaviour as the speech signal analyzed across each STFT frame *varies* and evolves spectrally across each successive STFT frame. The reader is pointed to the appendices at sections B.1 and B.3 for an overview of the DFT (Discrete Fourier Transform) and STFT (Short Time Fourier Transform), respectively.

It can be considered for the STFT of the male speech signal that, at any particular frame (i.e. analyzing the signal locally) the signal can approximately be viewed as quasi periodic. The short time and local behaviour, however, we shall attempt to illustrate typically depends on how the male speaker is currently (at any given time instant) applying the usage of his articulators (mouth, jaw and lips) to form certain words and/or sentences.

Having considered male speech as a type of relevant signal, we can then consider signals occurring from musical instruments as shown according to Figure 1.4 as another type of real audio signal that we may be interested in recovering and being able to *parametrize*.

Figure 1.4 shows an additive mixture of a musical signal (a sequence of 6 musical notes) in the STFT domain. Again we can note that the musical signal when observed in the STFT

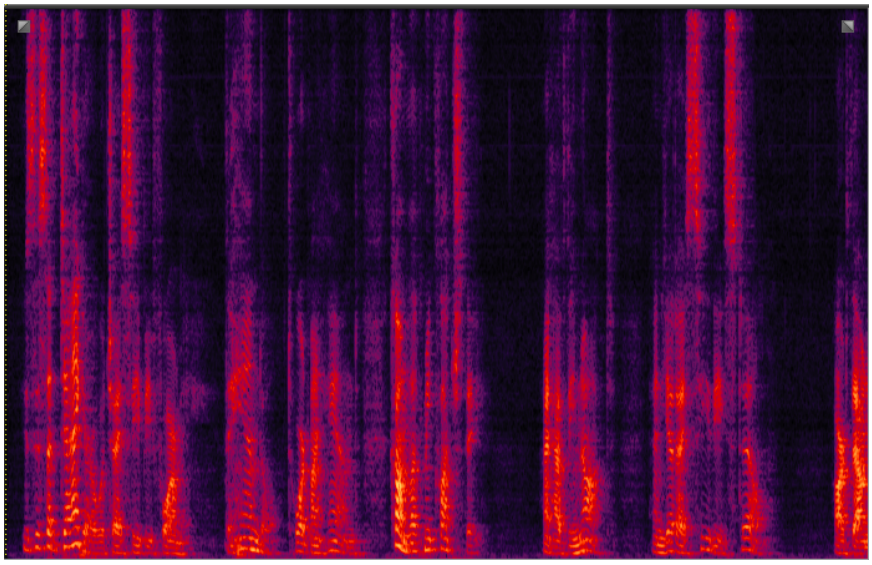

Figure 1.3: Example of STFT analysis applied to male speech signal

domain is highly dynamic and information-rich. Also we note that both representations of the observed signal contain the same information, but represented differently according to DFT, STFT and Fourier theory. In the time domain the amplitude information is conveyed on the vertical axis, where as in the STFT domain the amplitude information is conveyed as bright colors, where the lack of a color represents the absence of amplitude information.

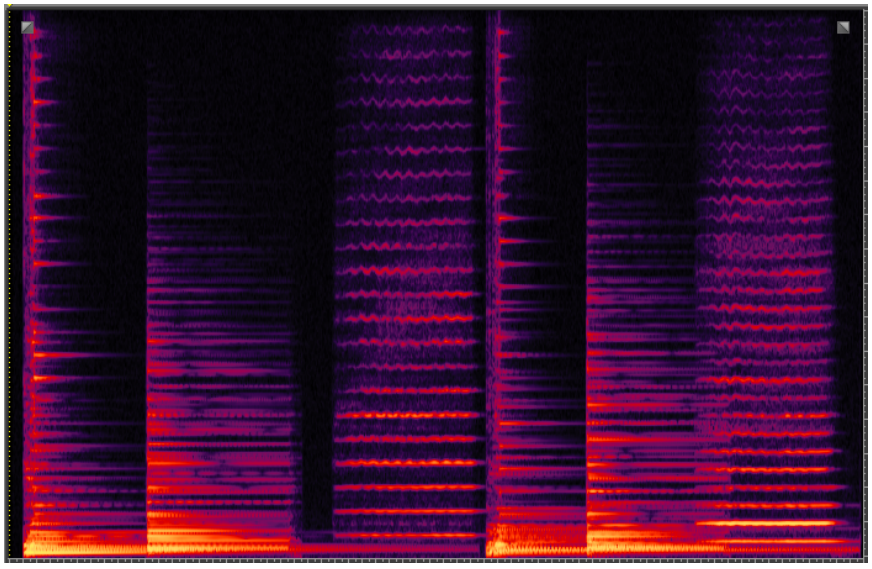

Figure 1.4: Example of STFT analysis applied to combined spatial images of musical sources

We note that according to Figure 1.4 we may be more adequately be able to deduce

what type of musical and harmonic signatures each burst of signal in fact corresponds to. Sequentially, and in either figure, the sequence of *musical notes* corresponds to a guitar note, a piano note, a violin note, then again a guitar note, piano note, and violin note. Unlike in Figure 1.3 we do not imply that the observed signal here has *not been* manipulated in any way to demonstrate what we are in fact observing. It has in fact been manipulated, additively, from six independent sources of information. Using acoustic room simulation software we have essentially simulated a hypothetical but appropriate scenario in which three musicians located at spatially distinct positions within a room environment have each played two musical notes at non overlapping time segments on their respective instruments. We intend to illustrate and describe with what purpose this has been done, with regards to whether or not source separation can be achieved under this scenario. It will be an objective of the thesis to provide an algorithm capable of algorithmically determining the configuration that generated this additive musical mixture, such as at what relative positions (or relative angles) relative to a reference position (the center of a microphone array) was each musician that played a particular note actually located. This scenario will serve as a useful instance or test case in the thesis for considering how signals (in this case musical signals) can be represented using algorithms such as NMF or SCM based NMF for representing audio spectrograms. These classes of algorithms will attempt to directly *parametrize* the STFT representation illustrated in Figure 1.4 in terms of a nonnegative and/or complex matrix parameter set. For instance SCM based NMF is a multichannel extension of the two factor NMF model first introduced in section 1.3.2.1.

In summary, in order to *simulate* and/or *construct* the observed signal for this test case, the signal segments corresponding to each musical instrument (guitar, piano, violin; 6 total) were spatially *filtered* using room simulation software (such as MCRoomSim [53]), and then added together digitally as time domain so-called *spatial images*. Thus the observed microphone signal corresponds to the additive superposition of the spatial information as seen by one of *two* simulated microphones. Thus in the room simulation that generated this particular microphone signal, another spatially distinct microphone signal was simulated that would be available to the source separation algorithm. It was specified to the room simulation software that the microphones be spaced approximately 10cm apart.

The purpose of demonstrating this STFT representation of musical notes is also to show that individual signal *components* can be superimposed in the time domain or time-frequency domain, and depending on how we appropriately parametrize this superposition we may then in fact be able to either partially or fully describe *separable* effects (additive, subtractive or multiplicative) corresponding to how audio events within a real acoustic and/or room environment truly occurred. Furthermore, if we have more than once sensor, which we have

stated shall be a requirement on the algorithm input, and we compute phase differences between sensors (microphones) as information that can be used to classify signals at each time frequency bin, we can begin to analyze and provide a classified representation(discriminate or distinguish, so to speak) that is both frequency dependent and time (activation) dependent. We intend that such a fully classified representation will provide a justifiable and appropriate source separation that extracts output signals on the basis of spatially similarity between signal components belonging to the same algorithmically learned class.

In this thesis we are interested in knowing if there exists a way to *classify* the source signals given not *one* but in fact *two* spatially related microphone recordings (by virtue of having been simulated under the same room simulation scenario.

Specifically, we intend not only to classify but to *separate* signal components (audio components) in such a way that reflect the *true*, but unknown given only the observed signals, *configuration* of the room simulation software and furthermore *parametrize, describe* and *explain* the observed mixtures in such a way that in some manner also is able to explain the set of operations that we have just verbally described and that were in fact used to essentially *construct* the representations of the signals shown in Figure 1.4 where we again point out that in order to construct the musical signals shown, it was required to manipulate separately the musical notes from each effective *source* signal (where we intend for each source signal to effectively correspond to one of three possible musical instruments; the guitar, the piano or the violin).

It will be shown that doing so purely on the basis of a two-channel stereo set of observed signals still presents a major challenge of audio signal processing and there exist perhaps only an exclusive set of algorithms that are well equipped enough to tackle source separation problems of this form (multichannel NMF algorithms such as SCM NMF). Therefore, we lastly emphasize that in this section we have not shown multiple microphone STFT repesentations, but that we assume that in practice, at least *two* channels of spatially related microphone channels will be available to us in order to consider how to *optimally* begin to process them. Thus we will often refer to the thesis's proposed algorithm and algorithms that are similar to it (since we consider them to have the same basic starting requirement of at least *two* spatially related and available microphone recordings) as *multichannel-STFT* based algorithms. Where this can be adequately understood and evaluated to mean, more simply, two-channel STFT's, but we have throughout the thesis still referred to them in this way, in order to provide more generality.

Chapter 2

Overview of Model Based and Transform Based Signal Processing Techniques

In the current chapter we intend to introduce four key topics, namely, spatial filtering models, the frequency domain single channel Wiener filter, the MVDR beamformer, and the multichannel Wiener filter. Firstly, we intend to describe the concept of spatial filtering with microphone arrays since it will be seen that at any frequency bin (or time frequency bin, if considering the STFT) that microphone dependent and frequency dependent amplitude and phase information can be encoded into acoustic transfer functions that specify a direct acoustic path between source and microphone positions. Furthermore, the acoustic (in fact, spatial) transfer function between any source and any microphone is dependent on both the position of sources and microphones as well as the relative difference between microphone positions, relative to the microphone array center, when considering one acoustic source at a time. Namely, this approximation is applied when the far field model of sound propagation is assuming, which approximate the near field model, but when the sources are located appropriately far away from the microphone array center.

2.1 Spatial Filtering with Microphone Arrays

2.1.1 Spatial Sampling of Sound Fields and Near-Field vs Far-Field Sound Propagation

In Appendix section B.2.1 we provide a discussion of arbitrary signals whose short time characteristics could be described using a Fourier series representation, and we described the additivity of spatial STFT coefficients by analogy in stating that a *decomposition* of the addition of parametrizable STFT coefficients must satisfy a *vector interpretation* of complex components in the complex STFT domain. The reader is pointed to and suggested to consider

the discussion in that section either prior to or after considering the current chapter.

We will now attempt to provide a thorough and useful insight into the somewhat challenging task of fully characterizing time differences of arrival and directions of arrival between sound waves occurring spatially at source positions and their effect as seen by a single microphone or multiple microphones within an acoustic environment. Here, and throughout the thesis this concept will be thereby referred to as the concept of *spatial filtering*. In order to illustrate the concept of spatial filtering it can be assumed momentarily and for illustration purposes that the source positions and microphone positions are exactly and geometrically known. This can be assumed since by applying room simulation software we effectively construct observed microphones signals that have occurred precisely due to the choice of spatial positions of the microphones and sources within the room environment.

In the following section it is demonstrated how to utilize exact knowledge of *multiple* microphone positions with respect to one another, in order to deduce the appropriate and corresponding *direction of arrival* of a *single* source signal impinging on the microphone array from a *fixed* spatial position within an acoustic environment.

Consequently, we then propose that the microphone time differences depend entirely upon the angle of arrival of the signal and thus assuming angle of arrival information we have is correct it can be used to *uniquely* determine the time difference of arrival (say, between any pair of microphones) from a particular *angle* of arrival (also typically known according to another term as *look direction*) or vice versa. That is, they correspond to equivalent information assuming that the source position is fixed and is known exactly and can be specified in terms of either a spatial vector pointing to the source or an angle (direction) of arrival pointing *approximately* to the actual spatial location of the source signal.

In the following sections we focus on learning the spatial filtering relationships between the source signal and the multiple microphone signals assuming we do indeed know the source position and consequently the direction of arrival of the source signal. In practice, for an actual blind source separation algorithm it may only be reasonable to assume that the microphone positions are spatially known, but not the spatial positions of the sources. This may be obvious but is in fact beneficial to now emphasize explicitly. This was illustrated conceptually according to equation 1.8.

Thus eventually we will seek to parametrize and to explain the occurrence of the observed microphone signals in terms of the most *plausible* and *significant* angles of arrival of an acoustic environment, since studying the concept of spatial filtering will in fact enable us to be able to develop the related concept of *source localization*. The importance of the concept of source localization will become more evident once we assume that if we are dealing with multiple source spectra at a particular time frequency bin, then it should be our motivation

to parametrize and to deal with the multiple sources separately.

For now, we at the moment still only seek to provide a fundamental introduction to the topic of *spatial filtering*, with regards to a *single* source and not multiple. Therefore, the signal captured per microphone, which shall generically be labelled $x_m(t)$, conveys possibly attenuated and time shifted signal information from the target source signal, and is partially redundant (e.g. in terms of its time-frequency spectra) when compared to observed signal capture at another microphone. Therefore in the time domain the signals $x_m(t)$ in fact will exhibit temporal correlation and in the frequency domain the spectra of the signals might also be similar in terms of their respective Fourier transforms, subject to only differences of amplitude and phase.

An objective of the proposed research is to show that CNMF algorithms, to some degree, will be able to separate the spectral part of the observed signal (corresponding to the effect of the source signal) from the frequency dependent spatial filtering signature (corresponding to each microphone being located at a slightly different spatial location).

2.1.1.1 Far-Field Model

We now present the far field model [17,28] of sound propagation as described with [28] which in contrast to the near field model, assumes that the spatial location of the target source is situated sufficiently far relative to a set of closely spaced microphones specified by the vectors $\mathbf{n}_m$.

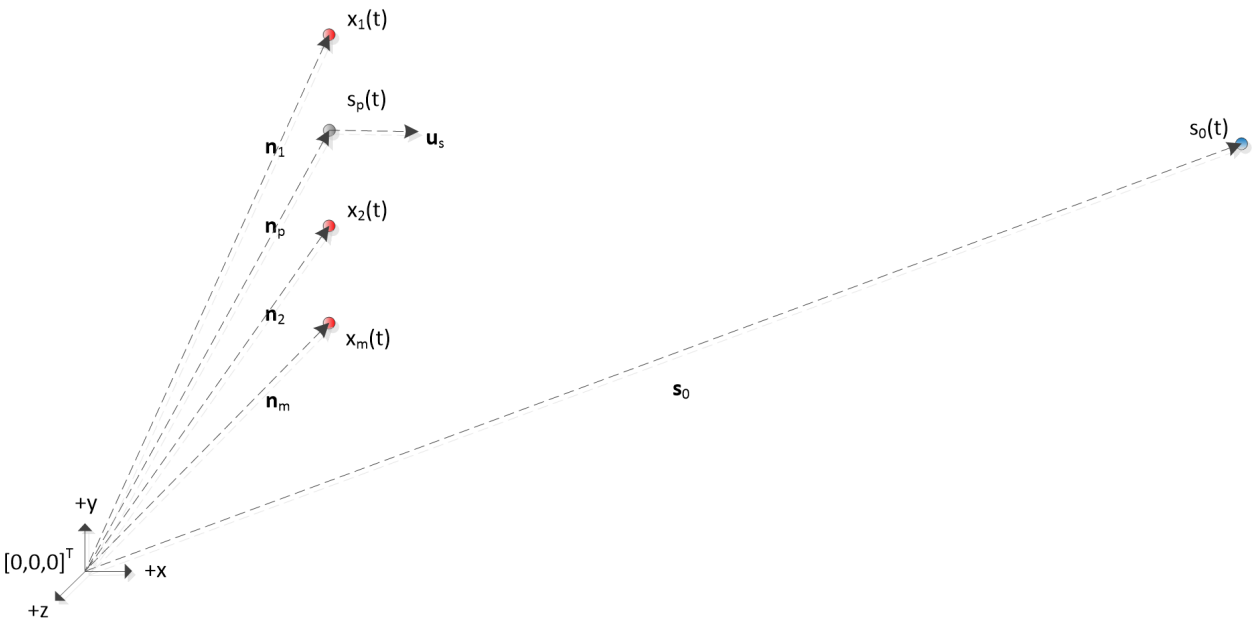

Figure 2.1: Example of Source and Mic Configuration for Far Field Model

The far-field model proposes that the microphone signal $x_m(t)$ can now be approximated as given by

$$\begin{aligned} x_m(t) \approx s(t + \triangle\tau_m) &\approx s\Big(t + \frac{\langle \mathbf{r}_m, \mathbf{u}_s \rangle}{c}\Big) \\ &\approx s\Big(t + \frac{\mathbf{r}_m^T \mathbf{u}_s}{c}\Big) \end{aligned} \tag{2.1}$$

where $\mathbf{r}_m = \mathbf{n}_m - \mathbf{n}_p$ is a vector pointing from the reference point $\mathbf{n}_p$ to the mth microphone $\mathbf{n}_m$. In the far-field model we assume that the reference point should be chosen to be placed physically close to the microphones, and collectively, the microphone array is to be located far away from the sound source. Under these assumptions, the sound waves can be approximately modelled as plane waves, and the time delays per microphone $\triangle\tau_m$ can now be computed using the inner product relationship between the vectors $\mathbf{u}_s$ and $\mathbf{r}_m$. Continuing from equation 2.1 this is equivalent to

$$x_m(t) \approx s\Big(t + \frac{[\mathbf{n}_m - \mathbf{n}_p]^T \mathbf{u}_s}{c}\Big) \tag{2.2}$$

where $\mathbf{u}_s$ is a unit vector pointing from the reference point to the sound source and can be computed as given by

$$\mathbf{u}_s = \frac{\mathbf{s}_0 - \mathbf{n}_p}{||\mathbf{s}_0 - \mathbf{n}_p||}. \tag{2.3}$$

The frequency domain representation of equation 2.2 is then to be interpreted as

$$X_m(f, \mathbf{n}_m) = S(f) \exp(j2\pi f \triangle\tau_m) \tag{2.4}$$

or equivalently, if the mth time difference $\triangle\tau_m$ is expressed as a negative time difference $\triangle\tau_m^{(-)} = -\frac{[\mathbf{n}_m - \mathbf{n}_p]^T \mathbf{u}_s}{c}$.

Then equation 2.4 consequently becomes

$$X_m(f, \mathbf{n}_m) = S(f) \exp(-j2\pi f \triangle\tau_m^{(-)}) \tag{2.5}$$

which is the convention taken in [24], which we may perhaps then use as a way to interpret the time differences as "delays" as having to do with the causality property of the Fourier Transform, which sound waves must conform to. Lastly, we can note that if we take $\mathbf{n}_p$, the reference point to be the origin $[0, 0, 0]^T$ of the coordinate system, then the time difference $\triangle\tau_m$ reduces to

$$\begin{aligned}\triangle\tau_m &= \frac{[\mathbf{n}_m - \mathbf{n}_p]^T \mathbf{u}_s}{c} \\ &= \frac{\mathbf{n}_m^T \mathbf{u}_s}{c}.\end{aligned} \tag{2.6}$$

To this point we have analyzed the multichannel spatial sound source propagation model carefully, focusing in particular on spatial filtering in the frequency domain, and having considered both the near field and far field models. The far field model represents an approximation of the near field model.

Equipped with the far field model described according to equation 2.4, we intend then to emphasize that properly understanding the discussion of a two-microphone linear array is sufficient to understanding microphone arrays with three or more sensors. When considering frequency and microphone dependent information provided by a multichannel STFT, the spatial covariance matrix that can be easily computed per frequency bin provides a relevant multichannel matrix quantity that must satisfy the principle of superposition (sum of individual covariance matrices due to individual sources). When we have the simple, but not obsolete task, of resolving the interchannel phase difference (direction of arrival) of a sound source for which we measure the effect of using two sensors, we simply need to consider the off-diagonal element of multichannel and 2×2 spatial covariance matrices computed at each frequency in order to obtain the entire acoustic transfer function between the reference point (array center) and the source whose direction of arrival we seek. This method was in fact exploited in the research by Virtanen and Nikunen [24].

2.1.1.2 Two-Microphone Linear Array

In the following section we seek to illustrate that a look direction (direction of arrival) and frequency dependent signature to be associated with any particular look direction can be calculated when considering any two microphones at a time. This will be illustrated by example according to the description provided for a uniform linear array according to [28]. Thus we emphasize that in analyzing directions of arrival we should first consider the microphones, in pairs, since we have seen a model that simply requires that the relative *locations* of the microphones be known with respect to a reference point near the center of the microphone array, in order to calculate microphone specific time differences of arrival and interchannel time differences of arrival with respect to a sound source with fixed position. Under the spatial parametrization of the geometrical look direction model, where a look direction points to the angle of arrival of the impinging sound source, we recall that the

far field model explains its associate time delay and amplitude scaling approximately and sufficiently as a function of the spatial geometry between microphones and sources relative to the center of the microphone array.

We can begin by specifying the spatial coordinates of two microphones located at equidistant distances from the origin and along the z-axis as

$$\mathbf{r}_1 = \frac{d}{2}\mathbf{e}_z \text{ and } \mathbf{r}_2 = -\frac{d}{2}\mathbf{e}_z \tag{2.7}$$

where $\mathbf{r}_1$ and $\mathbf{r}_1$ are vectors pointing from the origin (treated as a point of reference) to the first and second microphones, respectively, in a two-microphone uniform linear array. Also $\mathbf{d}$ specifies the physical distance between the two microphones and $\mathbf{e}_z$ and $-\mathbf{e}_z$ are unit vectors pointing pointing along the positive and negative z-axes, respectively. If we define an angle θ_s as the angle of arrival within the z-y plane, referenced to the positive z-axis, we can parametrize all possible angles of arrival within the z-y plane, but for which there exists some mirror (redundant) angles of arrival that cannot be distinguished from each other since they happen to have identical *frequency dependent spatial signatures*.

If we consider a harmonic plane wave (sound wave) occurring from an angle of arrival specified by θ, the phase of the harmonic wave that occurs at the first and second microphones respectively will be described as

$$\begin{aligned}\phi_1 &= \phi + \tilde{\beta}\langle \mathbf{r}_1, \mathbf{u}_s \rangle \\ &= \phi + \frac{\tilde{\beta} d}{2}\langle \mathbf{e}_z, \mathbf{u}_s \rangle\end{aligned} \tag{2.8}$$

and

$$\begin{aligned}\phi_2 &= \phi + \tilde{\beta}\langle \mathbf{r}_2, \mathbf{u}_s \rangle \\ &= \phi - \frac{\tilde{\beta} d}{2}\langle \mathbf{e}_z, \mathbf{u}_s \rangle\end{aligned} \tag{2.9}$$

where ϕ represents the *initial* or *absolute* phase of the sound occurring at the fixed position of the sound origin, and the terms $\tilde{\beta}\langle \mathbf{r}_1, \mathbf{u}_s \rangle$ and $\tilde{\beta}\langle \mathbf{r}_2, \mathbf{u}_s \rangle$ represent the *microphone-specific* frequency dependent time differences as are explained *spatially* by the far-field propagation model and we will seek to utilize as key information to parametrizing the proposed CNMF algorithm. Furthermore $\tilde{\beta}$ is simply the wavenumber as given by

$$\tilde{\beta} = \frac{2\pi f}{v} \tag{2.10}$$

where v is the speed of sound. In this thesis we shall greatly focus on how this model can be extended to suitably work well with CNMF algorithms as an extension to the algorithm, in a source separation problem context, by taking into consideration both the *intial phase* as well as the interchannel phase *difference* when considering any two pairs of microphones at a time. Given ϕ_1 and ϕ_2 we can then express the quantity

$$\Delta\phi = \phi_1 - \phi_2 = \tilde{\beta} d \langle \mathbf{e}_z, \mathbf{u}_s \rangle = \tilde{\beta} d \cos(\theta_s) \tag{2.11}$$

where $\Delta\phi$ represents the frequency-dependent and angle of arrival dependent *unwrapped* interchannel phase difference between the pair of microphones corresponding to $m = 1$ and $m = 2$.

We further note that in equations 2.8 and 2.8 the microphone dependent phases are both dependent on the *initial* phase ϕ of the targeted source signal however that when considering the interchannel time *difference* of the pair of microphones, as described by equation 2.11, we cannot observe its effect in any way, due to the subtraction that occurs.

While some CNMF models use this notion as evidence to argue the side of the argument that the initial phase of the target source signal needn't be parametrized as part of the CNMF model, the proposed research shall attempt to *initial* phase estimates as allegedly beneficial information to the CNMF model, that should be parametrized, and perhaps obtained by some other means of learning the appropriate data representation of the observed time frequency data per multichannel STFT bin.

Furthermore, it will be seen that source separation algorithms that happen to follow this particular argument, consequently utilize a multichannel Wiener filtering-like source reconstruction step, in order to *recover* the initial phase of source estimates per time-frequency bin, of reconstructed source signals. This method has been shown to be fairly effective and tends to works well, especially when there is only a small degree of time-frequency overlap between distinctly competing sources, however we will explore the possibility that directly modelling the initial phases of estimate sources as being directly considered as a key aspect of the CNMF model can perhaps improve source estimations within a source separation algorithm.

Figure 2.2 demonstrates the result of analyzing equation 2.11 as a function of frequency and the angle of arrival of the source signal as specified by θ_s. Plotting $\Delta\phi$ directly and element-wise corresponds to plotting the non-phase wrapped interchannel phase difference as a function of azimuth θ and frequency f (however, is not shown here). Using $\Delta\phi$, we have

computed instead $\arg(e^{j\Delta\phi})$ in order to generate the *phase wrapped* frequency dependent pattern at each row corresponding to a particular value of θ_s, shown in Figure 2.2. From this point forth in the thesis, we focus entirely on considering only the *phase wrapped* interchannel phase difference of a two-microphone linear array since we assume that we will be working with a conventionally computed STFT (non-heterodyned), which would result in all signals being analyzed by the STFT being also phase wrapped.

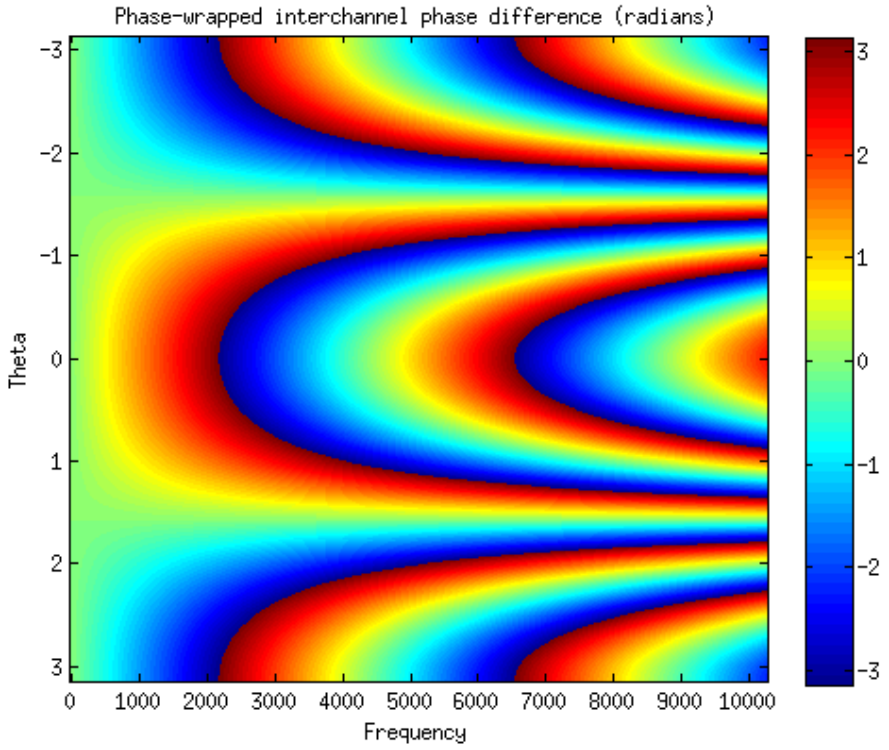

Figure 2.2: Phase-wrapped interchannel phase difference

2.2 Frequency Domain Formulation of Single Channel Wiener Filter

In Appendix section B.4.1.1 we consider the time-domain formulation and solution to the single channel Wiener solution. The FIR Wiener filter provides a statistical interpretation that is MMSE optimal according to the Wiener criterion (objective function). Here we consider its frequency-domain formulation (spectral) interpretation that we propose will eventually be useful for frequency domain source separation. We will detail this interpretation in the current section, following the description provided within [30]. Later it will be shown that in

combining the single channel frequency domain Wiener filter with the concept of the MVDR multichannel adaptive beamformer, it is possible to use the solution of the multichannel Wiener filter to inspire source separation algorithms with reconstruction of signals whose parameters (model) are appropriately chosen.

Proceeding with the derivation and taking the Fourier Transform of both sides of equation B.17, we obtain the following interpretation of the Wiener filter in the frequency-domain as given by

$$\hat{X}(f) = W(f)Y(f). \tag{2.12}$$

The error function $E(f)$ is analogous to the error function described in equation B.18 is now given by

$$\begin{aligned} E(f) &= X(f) - \hat{X}(f) \\ &= X(f) - W(f)Y(f). \end{aligned} \tag{2.13}$$

The frequency domain mean-square error criterion is now formulated for all frequencies f and can be considered as given by

$$\mathbb{E}\Big[|E(f)|^2\Big] = \mathbb{E}\big[(X(f) - W(f)Y(f))^*(X(f) - W(f)Y(f))\big]. \tag{2.14}$$

The optimal frequency domain Wiener filter $W(f)$ is then computed by computing the gradient of the MSE criterion with respect to $W(f)$ as given by

$$\frac{\partial \mathbb{E}\Big[|E(f)|^2\Big]}{\partial W(f)} = 2W(f)P_{YY}(f) - 2P_{XY}(f) = 0 \tag{2.15}$$

where, once again, we set the gradient equal to zero and solve for the optimal filter $W(f)$.

The quantities

$$P_{YY}(f) = \mathbb{E}\big[Y(f)Y^*(f)\big] \text{ , and } P_{XY}(f) = \mathbb{E}\big[X(f)Y^*(f)\big]$$

represent frequency domain statistical estimates of autocorrelated and cross-correlated signals. More precisely, $P_{YY}(f)$ and $P_{XY}(f)$ correspond to power spectral densities (PSD's) and are computed in practice as the Fourier transforms of the autocorrelation and cross-correlation sequences defined in equations B.23 and B.24, respectively.

From 2.15, the least mean square error frequency domain Wiener filter is

$$W(f) = \frac{P_{XY}(f)}{P_{YY}(f)} \tag{2.16}$$

For the particular case of a signal observed in additive random noise, this result can be illustrated by considering the following model of the observed signal $y(m)$ parametrized as being the result of a desired signal component $x(m)$ plus an additive noise component $n(m)$ as given by

$$y(m) = x(m) + n(m). \tag{2.17}$$

The observed, noisy signal in the frequency domain is given by

$$Y(f) = X(f) + N(f) \tag{2.18}$$

and the optimal frequency-domain Wiener filter is given by

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \tag{2.19}$$

In the objective of recovering an optimal minimum mean square error (MMSE) estimate of the clean signal $X(f)$ equation 2.19 we substitute $W(f)$ as specified by equation 2.19 into equation 2.12, such that we obtain the frequency domain estimate $\hat{X}(f)$ as given by

$$\hat{X}(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} Y(f) \tag{2.20}$$

and the usefulness of this frequency domain interpretation will be considered within the thesis in one of many possible ways that are analogous to the result obtained here in equation 2.20. For instance, a multichannel variant of the Wiener filter that utilizes concepts from adaptive beamforming [17,54], and is a concept that will be addressed later within the thesis.

By having considered the derivation of the Wiener filter in both the time domain and the frequency domain we have defined a set of principles that will allow us to consider the statistical and spectral interpretations of observed (i.e. available) and unobserved (i.e. unavailable signals) upon which the models for the observed signals typically follow the form of equation 2.18.

We note here, that if we consider the signal $X(f)$ to be an *observed* signal, as well as the signal $Y(f)$ then we suggest that the problem of inferring the noise term $N(f)$ becomes entirely trivial, and thus the usefulness of the Wiener filter would not be adequately described.

We would rather emphasize that the usefulness of the FIR and frequency domain Wiener

filter becomes more evident in problems where the $X(f)$ is unobserved and it is only the $Y(f)$ that we consider should be modelled as the sum of additive combinations of desired and possibly interfering components. The Wiener filter conditions should in both cases be able to describe the statistical property of the signals and spectral decomposition of the signals appropriately, but the details constituting the problem are highly dependent on the specific application under consideration (e.g. system identification, linear prediction, multichannel Wiener filtering).

2.3 MVDR Beamformer

In this section we introduce the MVDR beamformer [17,27], as an important signal processing tool and technology in the subject of optimum signal enhancement and microphone array processing. The MVDR beamformer, was first derived by Capon in [55] and later modified to operate adaptively by Frost in [56], who was the first to propose a constrained least-mean-square (LMS) type adaptation of the algorithm [27]. The optimal beamformer provides a set of optimal weights, $\mathbf{W}_0^{opt}(f)$, which can be derived within a frequency domain interpretation of the microphone signals as was done in [17] (and [57]), which the treatment of the MVDR beamformer presented in this section shall most closely follow.

If we now augment equation 2.4 as a vector equation, we have the following multichannel description of the vector $\mathbf{X}(f)$ per frequency bin

$$\mathbf{X}(f) = \mathbf{D}_0(f)S_0(f) + \mathbf{N}(f). \tag{2.21}$$

where $\mathbf{D}_0(f) \in \mathbb{C}^{M\times 1}$ encodes the microphone specific and frequency dependent phase differences relative to the array center (according to the far field model) of the M channel microphone array, looking towards a particular angle of arrival. If $\mathbf{D}_0(f) \in \mathbb{C}^{M\times 1}$ is configured appropriately, then the microphone array can be understood to possess spatial directivity looking towards the source signal $S_0(f)$ located at position $\mathbf{s}_0$. We can refer to $\mathbf{D}_0(f)S_0(f) \in \mathbb{C}^{M\times 1}$ as the spatial image vector due to the source signal $S_0(f)$. Now $\mathbf{X}(f)$ represents the observed multichannel (i.e. spatially diverse) vector of frequency domain coefficients at frequency bin f (note that the STFT frame bin index n can be omitted for the purpose of this discussion, but can be restored if needed, within another discussion that considers the windowing of signals into multiple STFT frames).

Given this frequency domain model of how the multichannel vector $\mathbf{X}(f)$ was generated, in order to reconstruct an estimate of the signal $S_0(f)$ we will compute the beamformer output as the vector product

$$Y(f) = \mathbf{W}_0(f)\mathbf{X}(f). \tag{2.22}$$

Here, $Y(f)$ represents the beamformer output, which we intend will, under the correct circumstances, provide us with an estimate of the desired signal $S_0(f)$ corresponding to spatial position $\mathbf{s}_0$. We then seek to successfully derive an optimal solution for the vector of optimal weights $\mathbf{W}_0(f) \in \mathbb{C}^{1\times M}$, preferably in terms of any of the frequency-dependent (and in some cases multi-channel) quantities shown within equations 2.21 and 2.22 that are available to us.

If we consider now the MVDR criterion $Q = \mathbb{E}[|Y_n|^2]$ defined as the variance of the noise term $Y_n = \mathbf{W}_0(f)\mathbf{N}(f)$ obtained by substituting equation 2.21 into 2.22 and minimizing Q with respect to $\mathbf{W}_0(f)$ it leads to the optimal MVDR beamformer $\mathbf{W}_0^{opt}(f)$ as given in summary (according to [17]) given by

$$\begin{aligned}\mathbf{W}_0^{opt}(f) &= -\lambda(f)\mathbf{D}_0^H(f)\boldsymbol{\Sigma}_{NN}^{-1}(f) \\ &= \frac{\mathbf{D}_0^H(f)\boldsymbol{\Sigma}_{NN}^{-1}(f)}{\mathbf{D}_0^H(f)\boldsymbol{\Sigma}_{NN}^{-1}(f)\mathbf{D}_0(f)}\end{aligned} \tag{2.23}$$

where $\lambda(f) = -\Big(\mathbf{D}_0^H(f)\boldsymbol{\Sigma}_{NN}^{-1}(f)\mathbf{D}_0(f)\Big)^{-1}$.

Substituting 2.23 into 2.22 we then obtain the MVDR beamformer's filtered output $Y(f)$ as given by

$$Y(f) = \frac{\mathbf{D}_0^H(f)\boldsymbol{\Sigma}_{NN}^{-1}(f)}{\mathbf{D}_0^H(f)\boldsymbol{\Sigma}_{NN}^{-1}(f)\mathbf{D}_0(f)}\mathbf{X}(f). \tag{2.24}$$

An perhaps now evident but important fact about the MVDR beamformer solution is that it connects the concept of multichannel spatial filtering to the concept of the multichannel Wiener filter, a topic that is referred to frequently in the literature on SCM NMF based methods for source separation. In the next section we will precisely consider how the MVDR beamformer can be combined with the single channel Wiener filter in order to formulate the so called *multichannel* Wiener filter.

2.4 The Multichnanel Wiener Filter

We now connect the results of the MVDR beamformer and Wiener filter, covered in earlier sections, resulting in the multi-channel Wiener filter. The multichannel Wiener filter provides

a MMSE estimate of the desired source signal based upon a frequency domain filtering operation of a multichannel DFT or STFT vector in a particular time-frequency bin.

The multichannel Wiener filter suggests we can provide a statistically optimal MMSE estimate of the desired source signal, $\hat{s}_{nf}$, given that we have access to various statistical and spatial properties about the observed and unobserved signal parameters, namely that we carefully consider the *covariance matrix* of the *parameterized* model for the observed signal $\mathbf{x}_{nf}$, as well as a vector $\mathbf{d}_f$ that specifies the appropriate frequency-dependent scaling and phase shifts as seen by the multichannel microphone array in observing the *spatially* filtered effect of the unobserved source signal s_{nf}, per microphone m. In this case, since we assume that we are only interested in estimating the effect of a single coefficient s_{nf} in a particular time frequency bin, that the basic multichannel Wiener filter described here corresponds to an *overdetermined* source separation problem, since for this illustration we have access to *more* microphone observations than the number of target source signals (here, just one).

In this section the notion of source index is dropped since we consider that the multichannel Wiener Filter [54] will be able to focus and enhance in a particular look direction defined by the configuration of the steering vector, $\mathbf{d}_f$, where

$$\mathbf{d}_f \propto \begin{bmatrix} 1 \\ g_2 e^{-j2\pi f \tau_2} \\ \vdots \\ g_M e^{-j2\pi f \tau_M} \end{bmatrix} \in \mathbb{C}^{M \times 1}$$

and where the reference position is taken to be equal to the position of microphone 1, s_{nf}. Also, here $\mathbf{b}_{nf}$ represents a multichannel noise vector of a particular (time frequency) TF bin.

We must first *suggest* how the vector $\mathbf{x}_{nf}$ was generated parametrically as given by

$$\mathbf{x}_{nf} = s_{nf}\mathbf{d}_{0,f} + \mathbf{b}_{nf} \tag{2.25}$$

which the multichannel Wiener filter model proposes is an appropriate parametrization of how the observed coefficients for the microphone array at a particular time frequency bin were generated. According to [17] the noise covariance matrix $\Sigma_{nf}^{\mathbf{b}'}$ can be expressed as

$$\Sigma_{nf}^{\mathbf{b}'} = \Sigma_{nf}^{\mathbf{b}} + (\phi_{nf}^{b,0})\mathbf{I} \tag{2.26}$$

a sum of components due to the uncorrelated noise $(\phi_{nf}^{b,0})\mathbf{I}$ with variance $\phi_{nf}^{b,0}$ plus a component due to a correlated (spatial) noise as given by $\Sigma_{nf}^{\mathbf{b}}$. The covariance matrix of the input vector $\mathbf{x}_{nf}$ can then be expressed as given by

$$\Sigma_{nf}^{\mathbf{x}} = \phi_{nf}^{s}\mathbf{d}_{0,f}\mathbf{d}_{0,f}^{H} + \Sigma_{nf}^{\mathbf{b}'}. \tag{2.27}$$

Since s_{nf} represents the unobserved part of the model, that must be estimated, let us then formulate that there exists an unknown matrix processor $\mathbf{g}(f)$, such that the optimal estimate of s_{nf}, will be given by the optimal estimate of $\mathbf{g}(f)$, given by

$$\hat{s}_{nf} = \mathbf{g}(f)\mathbf{x}_{nf} \tag{2.28}$$

where since we expect $\hat{s}_{nf}$ to be a complex scalar, and since $\mathbf{x}_{nf}$ we set up to be an $M \times 1$ complex column vector, we deduce that the matrix processor $\mathbf{g}(f)$ here must be a $1 \times M$ complex row vector. According to this problem formulation and its provided description of how $\mathbf{x}_{nf}$ was generated, we define the probabilistic cost function ϵ as the mean-squared error between the true but unobserved source signal s_{nf} and the matrix-processed estimate of the source signal $\hat{s}_{nf}$, as described by equation 2.28 such that

$$\begin{aligned}\epsilon &= \mathbb{E}\Big[|s_{nf} - \mathbf{g}(f)\mathbf{x}_{nf}|^2\Big] \\ &= \mathbb{E}\Big[(s_{nf} - \mathbf{g}(f)\mathbf{x}_{nf})(s_{nf}^{*} - \mathbf{x}_{nf}^{H}\mathbf{g}^{H}(f))\Big]\end{aligned} \tag{2.29}$$

and by taking the complex gradient of ϵ with respect to $\mathbf{g}^{H}(f)$ and setting the result equal to zero we obtain that

$$\mathbb{E}\big[s_{nf}\mathbf{x}_{nf}^{H}\big] - \mathbf{g}(f)\mathbb{E}\big[\mathbf{x}_{nf}^{H}\mathbf{x}_{nf}\big] = 0 \tag{2.30}$$

where subsequently

$$\begin{aligned}\phi_{nf}^{s}\mathbf{d}_{0,f}^{H} &= \mathbf{g}_{opt}(f)\Big(\Sigma_{nf}^{\mathbf{x}}\Big) \\ \mathbf{g}_{opt}(f) &= \phi_{nf}^{s}\mathbf{d}_{0,f}^{H}\Big(\Sigma_{nf}^{\mathbf{x}}\Big)^{-1}\end{aligned} \tag{2.31}$$

and $\mathbf{g}_{opt}(f)$ compactly denotes the optimal multichannel matrix processor (i.e. multichannel Wiener filter).

Under these assumptions, the optimal MMSE estimator of the signal s_{nf} is the multichannel Wiener Filter as given by

$$\hat{s}_{nf} = \phi_{nf}^{s}\mathbf{d}_{0,f}^{H}\Big(\Sigma_{nf}^{\mathbf{x}}\Big)^{-1}\mathbf{x}_{nf} \tag{2.32}$$

where the vector $\mathbf{x}_{nf}$ here is not the parametrized vector as specified by equation 2.25, but is in fact the actual (observed) multichannel STFT vector at time-frequency bin n-f. Thus, from equation 2.32 we obtain a practical and MMSE estimate of how to algorithmically extract the desired source signal $\hat{s}_{nf}$ based upon the *parametrization* of the observed source signals as specified by the vector $\mathbf{x}_{nf}$, when we formulate the filtering (extraction) problem such that there exists at most only one single source component s_{nf} that is to be extracted from the multichannel microphone array analyzed in terms of its multichannel STFT.

We would also like to show that the multichannel Wiener filter can be described as being composed of a MVDR beamformer followed by a Wiener post-filter

$$\mathbf{g}_{opt}(f) = h_{WF}(f)h_{MVDR}(f). \tag{2.33}$$

By invoking the Woodbury matrix inversion formula [17] the quantity $\left(\Sigma_{nf}^{\mathbf{x}}\right)^{-1}$ can be expressed as given by

$$\left(\Sigma_{nf}^{\mathbf{x}}\right)^{-1} = \left(\Sigma_{nf}^{\mathbf{b}'}\right)^{-1} - \left(\Sigma_{nf}^{\mathbf{b}'}\right)^{-1}\phi_{nf}^{s}\mathbf{d}_{0,f}\left(1 + \mathbf{d}_{0,f}^{H}\left(\Sigma_{nf}^{\mathbf{b}'}\right)^{-1}\phi_{nf}^{s}\mathbf{d}_{0,f}\right)^{-1}\mathbf{d}_{0,f}^{H}\left(\Sigma_{nf}^{\mathbf{b}'}\right)^{-1}. \tag{2.34}$$

and substituted into 2.32 to express the multichannel Wiener filter in terms of the noise covariance matrix $\left(\Sigma_{nf}^{\mathbf{b}}\right)$. Considering then that the MVDR beamformer has the form of

$$U_{nf} = \Lambda(f)\mathbf{d}_{0,f}^{H}\left(\Sigma_{nf}^{\mathbf{b}}\right)^{-1}\mathbf{x}_{nf} \tag{2.35}$$

which corresponds to the equation specified by 2.24

where

$$\Lambda^{-1}(f) = \mathbf{d}_{0,f}^{H}\left(\Sigma_{nf}^{\mathbf{b}'}\right)^{-1}\mathbf{d}_{0,f}$$

then the signal $\hat{s}_{nf}$ can be expressed as the output of the multichannel filtering of the vector $\mathbf{x}_{fn}$

$$\begin{aligned}\hat{s}_{nf} &= \frac{\phi_{nf}^{s}}{\phi_{nf}^{y}}U_{nf} \\ &= \frac{\phi_{nf}^{s}}{\phi_{nf}^{s} + \Lambda(f)}U_{nf} \\ &= \left(\frac{\phi_{nf}^{s}}{\phi_{nf}^{s} + \Lambda(f)}\right)\Lambda(f)\mathbf{d}_{0,f}^{H}\left(\Sigma_{nf}^{\mathbf{b}}\right)^{-1}\mathbf{x}_{nf}\end{aligned} \tag{2.36}$$

where the output of the MVDR beamformer followed by the Wiener post-filter $\frac{\phi^s_{nf}}{\phi^y_{nf}}$ is demonstrated to represent an alternative representation for equation 2.32.

Therefore, we have illustrated in this section that the Wiener filter concept applied in a multichannel sense, and due to its derivation upon the MVDR beamformer, has the capability, if properly configured according to the steering vector $\mathbf{d}_f$ of some degree of spatial directivity towards the significant look direction that characterizes the target source signal. This will be of special importance to deriving more *spatially* separable representations of audio signals per time-frequency bin. We then consider the multichannel Gaussian model for observed spatial images, which will further apply the multichannel Wiener filtering equation of 2.32, but now in a context where we assume that *more than one* source component s_{nf} can be present and problem of statistically estimating the *desired* signal component s_{nf} is now changed to the problem of statistically estimating a spatial image $\mathbf{y}_{jnf}$ due to the contribution of the jth source at time frequency bin n-f. Thus we assume that each time frequency bin corresponds to an additive mixture of one of J possible spatial images where the spatial images $\mathbf{y}_{jnf}$ can possibly *overlap* to result in the observed vector $\mathbf{x}_{nf}$.

This concept was in fact first introduced in section B.3 surrounding the explanation for 1.7, and where *spatial images* can simply be considered as the Fourier transform of equation 1.7.

2.5 Multichannel Gaussian Model

We now consider the multichannel Gaussian model [54] which attempts to model the likelihood (probability) of observing spatial image $\mathbf{y}_{jnf}$ and where the complete data likelihood is defined for the model as given by

$$P(\mathbf{y}_{jnf}|\mathbf{\Sigma}_{jnf}) = \frac{1}{\det\left(\pi\mathbf{\Sigma}_{jnf}\right)} e^{-\mathbf{y}^H_{jnf}\mathbf{\Sigma}^{-1}_{jnf}\mathbf{y}_{jnf}} \tag{2.37}$$

where $\mathbf{y}_{jnf}$ represents the jth spatial image parametrizing the contribution of source j to the observed multichannel mixture vector $\mathbf{x}_{nf}$. $\mathbf{y}_{jnf}$ represents the jth spatial image described probabilistically by a zero mean Gaussian distribution where

$$\mathbf{\Sigma}_{jnf} = V_{jnf}\mathbf{R}_{jf} \tag{2.38}$$

represents a source dependent covariance matrix at time frequency bin n-f and due to the contribution of source j and V_{jnf} represents the source variance of the jth source at the same time frequency bin.

The observed data likelihood function is defined for the model as given by

$$P(\mathbf{x}_{nf}|V_{jnf},\mathbf{R}_{jf}) = \frac{1}{\det\left(\pi\sum_{j=1}^{J}V_{jnf}\mathbf{R}_{jf}\right)}e^{-\mathbf{x}_{nf}^{H}\left(\sum_{j=1}^{J}V_{jnf}\mathbf{R}_{jf}\right)^{-1}\mathbf{x}_{nf}^{H}} \tag{2.39}$$

The observed multichannel STFT vector is explained probabilistically by a zero mean Gaussian distribution whose covariance is the sum of the source dependent covariance matrices as given by

$$\Sigma_{nf}^{\mathbf{x}} = \sum_{j=1}^{J}\mathbf{\Sigma}_{jnf} \tag{2.40}$$

and estimates of the spatial images $\hat{\mathbf{y}}_{jnf}$ per output source j and per time frequency bin n-f can thus be *reconstructed* as given by

$$\hat{\mathbf{y}}_{jnf} = V_{jnf}\mathbf{R}_{jf}\left(\Sigma_{nf}^{\mathbf{x}}\right)^{-1}\mathbf{x}_{nf}. \tag{2.41}$$

Therefore the covariance matrix $\Sigma_{nf}^{\mathbf{x}}$ can be seen as significantly important to constructing adequate estimates of the spatial images, as given by $\hat{\mathbf{y}}_{jnf}$, where equation 2.41 is necessarily the most interesting result of considering the multichannel Gaussian model.

When the number of sources J exceeds the number of microphone observations M per time frequency bin it is proposed that the multichannel Gaussian model is still an appropriate means to obtaining source estimates according to spatial image vectors $\hat{\mathbf{y}}_{jnf}$ for $j = 1, ..., J$ according to equation 2.41.

In [54] it is suggested that a source separation solution to the multichannel Gaussian model can be achieved and the parameters of the multichannel Gaussian model inferred in one of two possible approaches:

- A maximum a posteriori (MAP) based estimation of all model parameters using the EM algorithm.
- MAP based estimation of the source STFT coefficients according to equation 2.41

In this thesis we will not attempt to adhere directly to this suggestion but instead will follow the approach of the two highlight SCM NMF based algorithms [16, 24], which provide a source reconstruction procedure similar to that of equation 2.41 that is defined entirely based upon is own parametrization within an SCM NMF based processing approach.

To give more context as to the usage of multichannel MMSE filtering, the multichannel Gaussian model and spatial covariance matrix parametrization using NMF; the concept was also applied and developed within algorithms such as [19]. Fairly recently further methods

have popularized the technique as evidenced by a class of fairly recently occurring spatial SCM based algorithms [18–23].

2.6 Review of Chapter

The intent of this chapter has been to directly demonstrate the development of topics in audio signal processing that have directly inspired SCM-NMF based algorithms. The state of the art algorithms, [16, 24] upon which the proposed algorithms are inspired, can therefore be inherently related in some capacity to the majority of the topics introduced in this chapter. Whether a particular source separation algorithm eventually outperforms or is sub-optimal in comparison to another algorithm, may in fact be related to how well, in principle it is able to resolve and/or satisfy some or all of the considerations covered within this chapter.

To this point, it has not yet been covered how to exploit the NMF principle as well as other data optimization principles such as K-means [4], however, these topics shall all eventually converge when considering more thoroughly the state of the art algorithms [16, 24] to be covered in Chapter 5.

Discussion within Appendix B provides further justification of how to apply the tools covered within this chapter as frequency domain transformations of audio signals. We must rigorously consider them to this level of detail since early research [6, 7, 16] of NMF applied to time-frequency representations encapsulated this information by suggesting to apply NMF processing to solely the magnitude-spectrogram (absent of its complex phase) of audio signals, as opposed to the complex STFT spectrogram of multichannel audio signals in their entirety.

In this chapter of the thesis, in fact, a significant number of immediately relevant and practically useful concepts have been introduced. Subsequently, combining the concepts introduced within this chapter with the primary topics to be covered within the next two chapters, the reader should be equipped with the sufficient tools required to understand an introduction level explanation of the state of the art algorithms to be introduced within Chapter 5. Subsequently Chapter 6, builds upon the three primary algorithms covered within Chapter 5.

In particular, the far field model for sound propagation detailed in section 2.1 of the current chapter will provide a frequency domain (STFT domain) view of the multichannel audio processing of capture microphone signals that are to be provided to the as input signals proposed algorithm that is instrumental to source separation via the concept of K-means clustering. Since, with the exception of section 2.5, the concepts detailed in this chapter illustrated a scenario where we could assume that only *one* source signal was to

be processed with regards to a set of *two or more* sensors, the challenge in developing an appropriate algorithm will be having to do with understanding how to apply sections 2.1 to 2.4 to a *multiple source* and *underdetermined* (more sources than sensors) source separation problem formulation. Explicitly, the reader should understand that we proposed that only algorithms such as [16, 24] to be covered in Chapter 5 are capable of achieving satisfactory underdetermined source separation. The subsequent chapters will continue to build towards providing all the tools necessary to understand these two particular algorithms.

Chapter 3

Mathematical Optimization Concepts Applicable to Frequency Domain Source Separation

In the current chapter we focus on providing the tools required to analyze the derivation and development of EM and NMF algorithms, with the intent of using them to solve frequency domain source separation algorithms. Such issues are relevant to ensuring that any parametrization of a particular algorithm has desirable convergence properties.

What we hope to again upon fully detailing the methods to be covered within this chapter is at the very least an adequate (and rather, *disciplined*) sense of how observed data quantities should be treated separately from parametrized data quantities, and of equal importance, how to appropriately define *measures of fit* for the *clustering* (i.e. labelling of patterns) of data (practically speaking, vectors) that are relevant for achieving pattern classification, which is in fact of great interest for developing spatially intelligible audio source algorithms and techniques.

In this chapter we will focus on demonstrating the use of various statistical optimization methods that are typically applicable to large scale problems regarding matrix data. In formulating EM algorithms, it is first relevant to consider the concept of Maximum Likelihood. In considering the K-means algorithm it is first relevant to consider the EM algorithm. In order to appreciate the use of divergence measures used in building appropriate cost functions for either NMF or EM based algorithms it is of importance to consider the principle of Jensen's inequality, which can typically be seen as a necessary step to be invoked in either EM algorithms or NMF algorithms as a means of constructing upper or lower bounds on (surrogate functions) to a primary objective or cost function that *globally* characterizes the behaviour of the optimization problem as a function of the model parameters.

In this particular chapter, not all of the algorithms, need be necessarily applied specifically

to audio signals, but rather, in the current chapter and some parts of subsequent chapters, the methods that are described are generalized to consider any time of matrix or vectorized data that is the result of a *generative* or *statistically* motivated model that explains how the data was generated.

NMF, effectively, happens to fall within this description, and NMF as applied to time frequency representations of audio signals, happens to be only one particular application of the class of algorithms corresponding to NMF.

Multiplicative updates (the update rules of interest to adapting the algorithm parameters) of NMF algorithms are justified in terms of their auxiliary derivations. The original NMF by Lee and Seung [1] is first credited to demonstrating how this could be done by exploiting the so-called auxiliary function (majorization minimization) method. From a high level, since both an auxiliary function and its associated objective function can both be explained in terms of matrix parameters, an auxiliary function proof is typically meant to exploit an inequality (sometimes but not necessarily Jensen's Inequality, for instance) defined with respect to common or similar terms found within either the auxiliary function or its associated objective function.

Algorithms such as [16, 24, 25] applied the auxiliary function approach (derived and provided their own multiplicative update rules) while extending the two-factor basic NMF model applied to either single channel or multichannel microphone array audio processing in the complex STFT time-frequency domain. These algorithms will be introduced in later chapters, however, we shall first introduce the key concepts that inspired the derivation of the basic NMF algorithm, and will do so subsequently within the sections that follow.

3.0.1 Jensen's Inequality and Convexity

A common prerequisite to most probabilistic estimation optimization problems, Jensen's inequality, which can be stated as [13]

$$\varphi(\mathbb{E}[X]) \leq \mathbb{E}[\varphi(X)] \tag{3.1}$$

where $\mathbb{E}[\cdot]$ represents the mathematical expectation (averaging) operator and X is some random variable upon which a probabilistic distribution (probabilistic model) is proposed and $\varphi(\cdot)$ represents a convex function. A graphical interpretation of Jensen's inequality in terms of Gaussian like distributions is provide in Figure 3.1.

For a real convex function φ, numbers $x_1, x_2, \ldots, x_n$ in its domain, and positive weights a_i, the inequality as given by

$$\varphi\Big(\frac{\sum a_i x_i}{\sum a_i}\Big) \leq \frac{\sum a_i \varphi(x_i)}{\sum a_i} \tag{3.2}$$

represents Jensen's inequality. The inequality is reversed if φ is concave

$$\varphi\Big(\frac{\sum a_i x_i}{\sum a_i}\Big) \geq \frac{\sum a_i \varphi(x_i)}{\sum a_i} \tag{3.3}$$

and equality occurs if and only if $x_1 = x_2 = \ldots = x_n$ or if φ is affine.

When $n = 2$ only, Jensen's inequality can be illustrated to signify that the secant line that occurs between the function $\varphi(\cdot)$ evaluated at points x_1 and x_2 simply lies above the convex function itself, between the set of all real numbers between the numbers x_1 and x_2. The $n > 2$ this concept is extended but can still be shown to remain true. One way to prove this is by considering Karamata's inequality [58] which can be considered a generalization of Jensen's inequality.

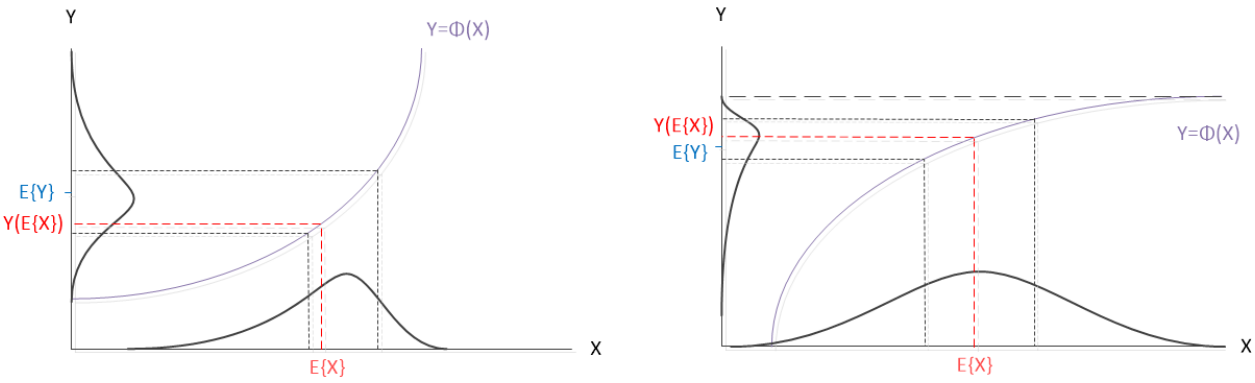

Figure 3.1: Jensen's Inequality for Convex and Concave functions defined over Random Variables

3.0.2 Majorization Minimization

A reader largely unfamiliar with NMF will quickly come to the realization when consulting NMF literature that most NMF algorithms are derived from what is explicitly known as an auxiliary function. An auxiliary function can be considered a surrogate to an objective function defined upon a specifically chosen divergence or distance measure. At first this concept may seem rather challenging, complicated, and perhaps unintuitive, however, an aim of this thesis is to begin to build an intuition of why auxiliary functions are needed and allow the optimization to perform better. Loosely speaking, the reasons for this in

large part are having to do with convexity of constituent functions upon which the primary objective function is defined. If an objective function is composed of solely convex (or concave) constituent functions, then an auxiliary function may not in any way be entirely necessary to finding a suitable solution. However, if the problem is slightly nonconvex but is still believed to be tractable, then using an auxiliary function to derive a solution, is in practice more stable and leads to more tractable and convergently stable solutions, which for NMF, occur in the form of *multiplicative* update rules. Such problems classified as either slightly nonconvex or entirely nonconvex are typically classified as difficult, and therefore it is sometimes implied that if an auxiliary function is needed then it means that the problem to be solved is understandably difficult.

Next we introduce a class of algorithms used for optimization known as majorization-minimization (MM) algorithms, of which EM can be considered directly related to. Here the term majorization is used solely with regards to functions, for the purpose of illustrating how to construct surrogate (auxiliary) functions and is not to be confused with the denser topic of majorization, which goes by the same name and is covered by Marshall and Olkin in [59].

We focus again simply on providing a set of notations and nomenclature (on functions) that illustrate the philosophy of the optimization procedure. As emphasized in [40], both EM and MM provide a suggested or prescribed description for creating algorithms and can be moreso considered to be classes of algorithms, rather than being exact algorithms, per se. MM seeks to simplify the original problem, by carefully inspecting the loglikelihood, or some other objective function to be minimized, by paying close attention to convexity and any possible inequalities than can be invoked in order solve the problem.

We start by considering again the problem of minimizing a real valued function or surface, which we will denote $f(\theta)$, given a multivariate parameter set θ. If we let $\theta^{(m)}$ represent a fixed value in the parameter space of all possible θ for the current iteration m, and denote $g(\theta|\theta^{(m)})$ a function whose form depends on $\theta^{(m)}$, then we seek to say that the function $g(\theta|\theta^{(m)})$ *majorizes* the function $f(\theta)$ provided the following two conditions are true:

$$\begin{aligned} g(\theta|\theta^{(m)}) &\geq f(\theta) \text{ for all } \theta \\ g(\theta^{(m)}|\theta^{(m)}) &= f(\theta^{(m)}) \end{aligned} \tag{3.4}$$

Stated differently, and in correspondence with an intuitive interpretation, the function $g(\theta|\theta^{(m)})$, viewed as a surface, lies above the surface $f(\theta)$ and is tangent to it at the point $\theta = \theta^{(m)}$. As important as this visual interpretation is, it is equaly important to pay attention

to the arguments of the function, as doing so will ensure that the any such of inequalities of the same form that might correspond to subsequent iterations of the MM procedure can be easily verified to be true or false. Therefore, we quickly review the respective roles of first and second arguments of the majorizing function $g(\theta|\theta^{(m)})$. The first argument should be thought of as an independent variable of the majorizing function (varying it can guarantee that the majorizing function evaluated at the chosen value of θ is at least *greater* than the function $f(\theta)$ evaluated at the same θ, assuming θ is not chosen as the tangency point $\theta^{(m)}$) while the second argument should be thought of as the key parameter of the function; its point of tangercy $\theta^{(m)}$ specifies and reminds us where $g(\theta|\theta^{(m)})$ is tangent to the original function $f(\theta)$. The crux of the MM algorithm is then to actually minimize the majorizing function $g(\theta|\theta^{(m)})$ instead of the original function $f(\theta)$ in such a way that (in the convergence of the algorithm) iteratively solves the original problem formulation - of minimizing the original function $f(\theta)$. So, if $\theta^{(m+1)}$ denotes a minimizing value of the current majorizing function $g(\theta|\theta^{(m)})$, and the quantity $\theta^{(m-1)} - \theta^{(m)}$ represents a proposed "step" (difference, or update) from the mth iteration to the $(m+1)$th iteration, then it can be shown that the MM algorithm forces the function $f(\theta)$ downhill as a result of this step.

As mentioned, we first clearly establish that $\theta^{(m+1)}$ represents the output of the following minimization problem

$$\theta^{(m+1)} = \underset{\theta}{\operatorname{argmin}}\ g(\theta, \theta^{(m)}) \tag{3.5}$$

and can subsequently show the following inequality to be true

$$\begin{aligned} f(\theta^{(m+1)}) &= g(\theta^{(m+1)}|\theta^{(m)}) + f(\theta^{(m+1)}) - g(\theta^{m+1}|\theta^{(m)}) \\ &\leq g(\theta^{(m)}|\theta^{(m)}) + f(\theta^{(m)}) - g(\theta^{(m)}|\theta^{(m)}) \\ &= f(\theta^{(m)}) \end{aligned} \tag{3.6}$$

as well, which [40] states follows from the fact that $g(\theta^{(m+1)}|\theta^{(m)}) \leq g(\theta^{(m)}|\theta^{(m)})$.

Invoking this, and cancelling some terms within the equation 3.6, it can be re-written more compactly and perhaps more clearly in a form similar to that presented in [60].

If we again use 3.5 to denote the update rule from the mth to $(m+1)$th iteration for choosing the minimizer $\theta^{(m+1)}$ of the majorizing function $g(\theta, \theta^{(m)})$, then the following sequence of inequalities can be shown to be true

$$f(\theta^{(m+1)}) \leq g(\theta^{m+1}|\theta^{(m)}) \leq g(\theta^{m}|\theta^{(m)}) = f(\theta^{(m)}) \tag{3.7}$$

where reading this sequence of inequalities from right to left, one can conclude that the MM procedure indeed iteratively forces the function $f(\theta)$ (for $\theta = \theta^{(m)}, \theta^{(m+1)}...$) downhill.

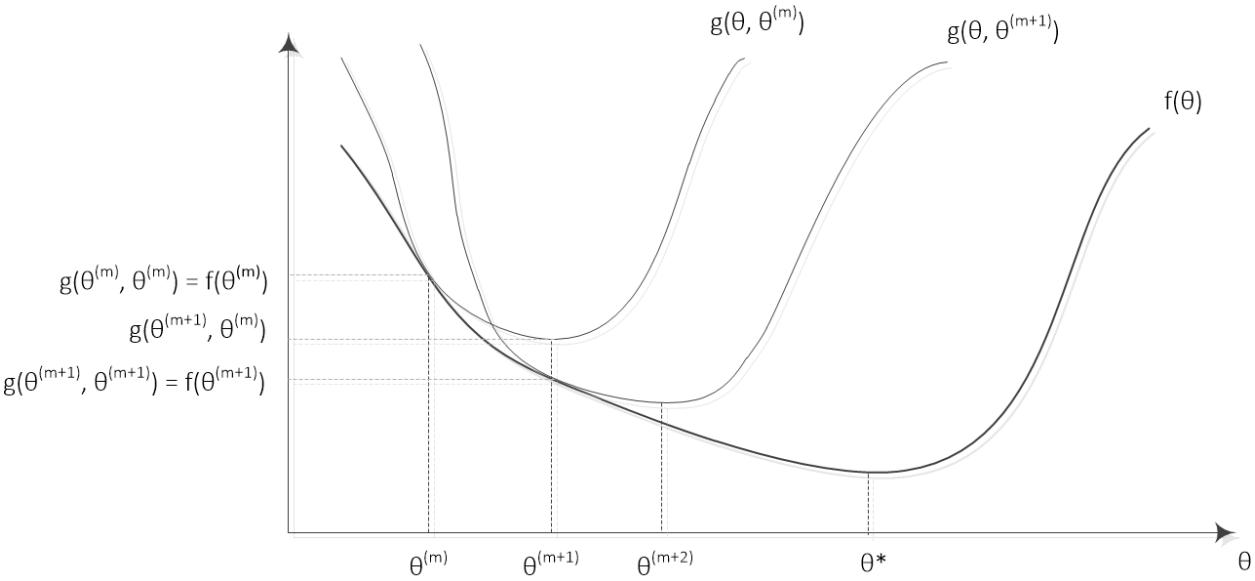

Figure 3.2: Principle of the MM Procedure

Classes of MM algorithms achieve the desired property of numerical stability, with the added cost of being iterative, which in some cases may be less direct than other optimization techniques.

3.0.2.1 K-Means Clustering

We now consider the K-means approach provided by [4] which is to utilized in later parts of the thesis in order to discover and identify important patterns in data that various algorithms attempt to model and provide adequate representations of. It will be shown that successful source separation and/or pattern classification will be achieved on the basis of understanding both *similarities* and *dissimilarities* between various groups and/or subsets vectors within the observed set of vectors $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$.

The premise of it is to identify patterns within a set of data (vectors) to which a set of K mean vectors of the same dimension can be applied in order to represent the original set of vectors.

K-means begins by an objective function to be minimized, also referred to as a *distortion measure*, as given by

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mathbf{u}_k||^2. \tag{3.8}$$

The first requirement of applying K-means is that of a data set that we would like to represent $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ with N total D-dimensional observation vectors in $\mathbb{R}^{Dx1}$. One of the main goals of K-means is to compute an assignment or partition, of the data set, into a target number K of clusters, with each cluster being characterized by its cluster center or mean vector $\mathbf{u}_k$ also in $\mathbb{R}^{Dx1}$. Since $N > K$ (K is chosen to be smaller than N), K-means typically works best if some "groups" of vectors within the data set are highly similar, since the vectors in that group could be fairly well represented as belonging to a single cluster represented by the cluster's mean vector $\mathbf{u}_k$. The similarity measure that the K-means algorithm is based upon is the Euclidean Distance. To account for the loose expectation that the data set itself forms groups or clusters of similar vectors that can, in some configuration, be either identified or learned, the cluster indicator matrix r_{nk} is introduced as part of the model. This indicator matrix r_{nk} provides a set of variables that precisely label the amount of similarity between vectors in the data set and vectors in the mean set (the set of K mean vectors). The cluster indicator variables can be chosen as hard or soft, that is to say, binary-valued (0 or 1) or "fuzzy-valued" (taking on values in the interval $\{0, 1\}$).

To minimize the Distortion measure J and to iteratively obtain the converged indicator and mean sequences r_{nk} and $\mathbf{u}_k$ we update them according to the following steps. Note that the steps provided here and throughout the thesis are for soft (fuzzy-valued) indicators (For hard (binary-valued) K-means, [4] can be consulted):

1. Randomly initialize the set of K mean vectors $\{\mathbf{u}_1, ..., \mathbf{u}_K\}$.

2. Compute assignment of soft indicators, r_{nk} with the variable $m = 2$:

$$r_{nk} = \frac{1}{\sum_{r=1}^{K} \left(\frac{||\mathbf{x}_n - \mathbf{u}_k||}{||\mathbf{x}_n - \mathbf{u}_r||} \right)^{\frac{2}{m-1}}} \tag{3.9}$$

3. Compute assignment of means $\mathbf{u}_k$, with the variable $m = 2$:

$$\mathbf{u}_k = \frac{\sum_{n=1}^{N} {r_{nk}}^m \mathbf{x}_n}{\sum_{n=1}^{N} {r_{nk}}^m} \tag{3.10}$$

4. Evaluate the distorted measure J: Stop, if it has been minimized to a satisfactory degree, if not continue by repeating steps 2 and 3.

We suggest then what to take away from having considered the K-means algorithm as

it pertains to considering how it should be applied. In order to make it easier to recognize problems in which K-means can be applied let us summarize the K-means algorithm compactly by introducing notation as given by

$$\{\mathbf{u}_k, r_{nk}\} \leftarrow \{\mathbf{x}_n\} \tag{3.11}$$

where the mean vectors $\mathbf{u}_k$ must be of the same dimension as the observed vectors $\mathbf{x}_n$, and K is some number that must be chosen as less than the total number of observed vectors $\{\mathbf{x}_n\}$ which is equal to N.

Thus from this compact interpretation, we can see what the name in fact suggests, quickly, if we use this type of reasoning applied to areas in which we suspect that the algorithm may be useful to enhance or augment a currently obtained result.

We must then simply think of how an arbitrary set of data should be mapped into the set of vectors $\{\mathbf{x}_n\}$ and then we simply define the distortion measure J, apply the algorithm, and then consider what results occur within the set $\{\mathbf{u}_k, r_{nk}\}$.

3.1 Cost Functions

The study of matrices, and pattern classification are not too far removed from the study of these types of functions however, often we are highly interested in the probabilistic properties and convergence properties of two-argument functions specifically referred to as *divergences* and/or *distances*.

Algorithm development must carefully consider how to utilize these distances and divergence as a starting point to adequately *constructing* more useful functions in a pattern classification, data, or matrix optimization problem. Often we must determine how to become practically motivated to utilize these *measures of fit* in some particular way that describes how to most optimally develop appropriate methods for defining cost functions that are fully applicable to large scale data fitting problems and that can be fully described in terms of a useful set of chosen data or matrix parameters, where the parameters can be optimized to allow the cost functions to typically converge to a fixed point. Only if the algorithm in fact converges to a fixed point consistently (given an appropriately fair data set) that represents a *global minimum* or maximum, do we then consider that the learning rules of the developed parametrization might be useful in practice. For further discussion of these topics in the context of NMF based algorithms, the reader is suggested to consult the appendices at sections A.1 and A.3, for instance.

3.2 Review of Chapter

In summary, chapter 3 has provided an introduction into the topics of statistical methods used for maximum likelihood, pattern recognition and matrix representation principles for modelling a set of vectors, typically treated as arbitrary observed data that are to be indexed and explained element-wise on the basis of the most plausible statistical or numerical model that can be provided that most likely explains how the data was generated.

The key take away subjects that we can consider have been covered within this chapter include

- A simplified illustration of the auxiliary function approach (explained in terms of majorization minimization) as well as Jensen's inequality which were demonstrated as being applicable with respect to EM based algorithms, but will also be shown to be applicable to most NMF based algorithm derivations that we will in fact encounter.
- The K-means algorithm focusing on pattern classification of an observed data set that returns a set of mean vectors and indicator values that can be used to approximately represent the observed data.

We have also provided a convergence analysis of the basic NMF algorithm as provided within section A.3. A related treatment of commonly applied divergences measures applied to the two-factor, single chanel audio spectrogram decomposition model will be provided in the next chapter, focusing primarily on the concept of basic NMF applied to audio spectrogram matrix representation and conventional NMF based processing techniques under this scenario (single channel scenario). Later, it will be seen that when K-means is applied to a multichannel NMF based audio source separation scenario, the K-means algorithm proves to be immediately useful, and in fact instrumental to achieving significant results. A challenge will be how to apply the K-means algorithm to CNMF and SCM NMF based processing, where we extend the notion of nonnegatively processed STFT coefficients to complex valued STFT coefficients. The reader is encouraged to consult sections of the appendices B.2, A.5, and A.6 for a discussion pertaining to complex valued matrix optimization in the complex STFT domain.

Firstly however, before considering such issues, both the convergence and derivation of the multiplicative update rules corresponding to the original NMF algorithm by Lee and Seung will be considered at the beginning of the next chapter.

It will also have to be addressed however, that the *nonnegative* spectogram two factor model, without applying the spatial and multichannel signal processing concepts covered in Chapter 2 to it, is not a truly adequate representation for providing recognizable *spatial*

audio features within the NMF decomposition/parametrization. The topic of multichannel Wiener filtering, as it pertains to SCM NMF based algorithms will therefore be re-considered carefully within subsequent chapters as a concept that is capable of representing and providing appropriate spatial audio features within a source separation model, algorithm, and parametrization. Multichannel Wiener filtering based methods will seek to exploit the available spatial diversity property of having access to multiple spatially diverse microphone recordings, assuming a microphone array is used to capture audio signals. It will be shown that when combined with NMF and K-means based processing, the MM principle as well as the proper use of Jensen's inequality can be used to provide auxiliary functions to source separation problem formulations that may seem difficult to solve tractabily.

We then propose that the SCM NMF based algorithms to be introduced will be able to provide both system identification (of spatial filters) as well as signal identification capability that occur due to the presence of the two factor NMF model contained as part of the SCM NMF models.

Chapter 4

Modelling Time Frequency Domain Audio Features with NMF and other Topics

4.1 Nonnegative Matrix Factorization

In this chapter we illustrate the first auxiliary function approach (i.e. the MM approach detailed in section 3.0.2 but now applied to nonnegative matrices) that was most notably implemented and resulted in multiplicative update rules capable of iteratively providing a parts based representation of nonnegative factor matrices (i.e. the NMF parameter set). The convergence analysis of classes of NMF algorithms can be demonstrated as having the property of being stable and guaranteed in terms of monotonic convergence, to some degree, subject a set of conditions, as discussed within section A.3 and to be further shown within this chapter.

The derivation of any particular NMF algorithm is typically a highly detail-oriented task. As with EM algorithms, it requires some level of ingenuity in justifying the construction of a particular objective function and subsequent auxiliary function and further proving mathematically that the auxiliary function is in fact valid. Beyond creativity, it requires some level of willingness to experiment, especially in deriving extensions to more basic or currently existing algorithms.

It is for this reason that we now take the opportunity to consider for the first time a more thorough analysis of the original NMF algorithm by Lee and Seung. To this point in the thesis we have often referred to the original NMF algorithm as a two factor (i.e. two parameter, two matrix) NMF algorithm. It will now be demonstrated why this naming convention is appropriate. In CNMF algorithms and SCM NMF algorithms, it will be important to be able to recognize the role of the original two factor NMF algorithm, in order to be able to interpret CNMF and SCM NMF as extensions that were in fact inspired by the original two factor NMF algorithm. Another naming convention for the original NMF algorithm is to simply

refer to it as *the auxiliary function approach*. This is because the principle of majorization minimization when applied to solving problems formulated as NMF are known to lead to multiplicative update rules for the matrix parameters, sometimes referred to as *dictionary matrices*. The multiplicative update rules themselves are also often easy to implement in terms of linear algebra and matrix subroutines, proposed ideas can easily be tested and brought to an implementable state of development with a fair amount of ease.

4.1.1 Auxiliary Function Derivation and Convergence Analysis

Lee and Seung, in [1] are credited for formulating the original version of a new class of algorithms known as Nonnegative Matrix Factorization algorithms. With a starting point similar to that of Principle Components Analysis and Vector Quantization, that is, to consider how to factorize a data matrix subject to different constraints (such as orthogonality constraints), it is emphasized that the constraint which characterizes NMF is that of non-negativity, which can be shown to be useful when analyzing a matrix dataset that may have been generated from a sparse and/or parts-based representation.

They provide two alternative formulations for NMF as optimization problems which form the starting point for deriving the auxiliary functions and update rules associated with them.

Before stating them, we first consider that the model to be conceived consists of nonnegative factors W and H whose multiplication approximates an observed nonnegative data matrix V such that

$$V \approx WH \tag{4.1}$$

where this model exactly describes the basic NMF model as first introduced in section 1.3.2.1 and according to Figure 1.2, but where the matrix V in this case replaces the matrix X (labelling of the observed matrix is simply applied differently, within the current section). If we then recall respectively the Euclidean distance (A.2) and Kullback-Leibler Divergence (A.4) cost functions introduced in Chapter 1, we can begin to explicitly define the two alternative formulations of Nonnegative Matrix Factorization as according to

> Problem 1: Minimize the objective function $||V - WH||^2$ subject to the nonnegativity constraints $W, H \geq 0$.
>
> Problem 2: Minimize the objective function $D_{KL}(V||WH)$ subject to the nonnegativity constraints $W, H \geq 0$.

A key observation that must be stated for both cost functions is that although the functions $||V - WH||^2$ and $D_{KL}(V||WH)$ are convex with respect to each of the matrices

W and H when considered individually, they are not convex when considering both W and H together [1]. With the aim of deriving an iterative algorithm that can output an optimal solution corresponding to global minima of the cost functions, it is for this reason that it is unreasonable to expect that such an objective can be easily achieved. Numerical optimization provides techniques that are capable of finding local minima. An auxiliary function method is presented in the sections that follow for both cost functions that offer a fair compromise between optimality, speed, and ease of implementation, according to Lee and Seung.

4.1.2 Multiplicative Update Rules

The derivation of multiplicative update rules, as proposed by [1] is provided in the Appendix at section A.6.1.

The multiplicative update rules for the Lee and Seung algorithm can then be compactly provided in summary as given by

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}, \quad W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}} \tag{4.2}$$

for the square Euclidean distance derivation, and by

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}}, \quad W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_v H_{av}} \tag{4.3}$$

for the KL Divergence based derivation.

Example applications for the multiplicative update rules shown here were discussed in section 1.3.3. By considering section 4.1.3 and the introductory section of [16] it was shown that they can immediately be applied towards analyzing single channel spectrograms and for representing *nonnegative* decompositions of redundant spectral patterns within observed audio spectrograms in a useful manner. Furthermore there exist various extensions to the basic NMF update rules as discussed within [7] that can be extended and applied within a broad and diverse amount of applications where meaningful matrix decompositions and factorizations are required. One such type of extension or variant is known as the nonnegative tensor factorization (NTF) model, where the notion of nonnegative matrices are extended to the notion of nonnegative tensors (sometimes referred to as multi-way tensors). Three such models are illustrated in section A.7 of the appendix.

4.1.2.1 Enforcing Nonnegativity

In order to enforce nonnegativity, it is typically suggested only persist the positive values of any matrix or tensor quantity that represents a parameter of the NMF algorithm. Computationally, this is meant to signify that negative values of a nonnegative matrix should be clipped (and set to zero), at the output of an update rule for any NMF parameter and once per iteration. In practice, in the proposed algorithm's implementation and various other implementations, this has been evidenced to work well in practice.

4.1.3 Basic NMF concept applied to Time Frequency Representations of Audio Data

Equipped now with the definition solution (multiplicative update rules) of the NMF problem formulation, we will consider how to apply the concept of NMF when the observed matrix to be factorized is in fact a *nonnegative* audio STFT spectrogram whose nonnegatively processed coefficients in each time frequency bin must be explained as an observed time frequency matrix. We shall then consider how the squared Euclidean distance, KL divergence, and IS divergence corresponding to equations A.4 to A.5 can be applied as simple two argument functions that can be applied to evaluating two matrices $\mathbf{X}$ and $\hat{\mathbf{X}}$ in a *element-wise* manner. Specifically, it is implied that

$$\hat{\mathbf{X}} \triangleq \mathbf{TV} \tag{4.4}$$

and that

$$\mathbf{X} = \hat{\mathbf{X}} + \mathbf{E} \tag{4.5}$$

where $\mathbf{X}, \hat{\mathbf{X}} \in \mathbb{R}^{F \times N}$, $\mathbf{T} \in \mathbb{R}^{F \times K}$, $\mathbf{V} \in \mathbb{R}^{K \times N}$, and $\mathbf{E} \in \mathbb{R}^{F \times N}$ is an approximation error matrix, that can be used to determine how well the NMF model has converged according to the global minimization of the divergence $D_*(\mathbf{X}, \hat{\mathbf{X}})$. Sometimes equation 4.5 is more conveniently expressed without including the term $\mathbf{E}$ as given by

$$\mathbf{X} \approx \hat{\mathbf{X}} \tag{4.6}$$

in order to signify that if the factors $\mathbf{T}$ and $\mathbf{V}$ have converged via multiplicative updates to a reasonably stable solution then the approximation matrix $\mathbf{E}$ should become almost negligible in terms of its amplitudes, considered element-wise. Thus we should always recall that $\mathbf{X}$ is meant to signify an observed matrix, while $\hat{\mathbf{X}}$ is meant to signify the matrix that

occurs due to the multiplication of NMF parameter matrices $\mathbf{T}$ and $\mathbf{V}$ and that $\hat{\mathbf{X}}$ is only required to represent the observed matrix $\mathbf{X}$, *approximately*.

In order to apply the notion of an element-wise matrix divergence between the observed and parametrized matrices $\hat{\mathbf{X}}$ and $\mathbf{X}$, we apply the concept of a *general* distance/divergence as first introduced according to equation A.1 such that

$$D_*(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d_*(x_{fn}, \hat{x}_{fn}) \tag{4.7}$$

represents matrix divergence that takes an element-wise divergence (defined over all elements of the pair of matrices) and then computes the sum of all the element-wise divergences, $d_*(x_{fn}, \hat{x}_{fn})$. We imply that matrix distances and divergences are useful when applied to the two factor NMF parametrization of an observed matrix $\mathbf{X}$ given a factorization in terms of matrices $\mathbf{T}$, and $\mathbf{V}$ where we are interested in knowing if the optimization problem

$$\mathbf{X} \approx \hat{\mathbf{X}} \triangleq \mathbf{TV} \qquad \text{subject to } \mathbf{T} \geq 0 \ , \ \mathbf{V} \geq 0 \tag{4.8}$$

can be solved and more specifically whether or not there indeed exists a *factorization* of the matrices $\mathbf{T}$ and $\mathbf{V}$ that *evaluated* in terms of the element-wise distance divergence between the matrices $\mathbf{X}$ $\hat{\mathbf{X}}$ that happens to be essentially very small but also meaningful in terms of conveying appropriate patterns of interest or redundancies within the matrices $\mathbf{T}$ and $\mathbf{V}$ (depending on the application). Explicitly stated equation 4.8 represents the typical NMF decomposition model for representing an audio signal's nonnegatively pre-processed STFT [16].

In order to enforce the nonnegativity of the observed spectrogram, a nonnegative pre-processing of the (naturally complex-valued) observed STFT coefficent $\tilde{x}_{fn}$ should be processed on the basis of making all of the observed matrix elements strictly nonnegative or alternatively computing by the nonnegative square root of the matrix elements as given by

$$x_{fn} = \begin{cases} |\tilde{x}_{fn}| \\ |\tilde{x}_{fn}|^2 = \tilde{x}_{fn}\tilde{x}^*_{fn} \end{cases} \tag{4.9}$$

$$\hat{x}_{fn} = \sum_{k=1}^{K} t_{fk} v_{kn} \tag{4.10}$$

This type of processing that was demonstrated within [16] proved to be useful and a corresponding type of processing of the observed signals exists when the spectogram is extended to becoming a multichannel spectrogram. In the single channel context, the parameters of

the basic two factor NMF model were described in terms of factor matrix $\mathbf{T}$ and $\mathbf{V}$, where $\mathbf{T} \in \mathbb{R}^{F \times K}$ is the basis of *spectral* basis vectors and $\mathbf{V} \in \mathbb{R}^{K \times N}$ is the matrix of activations, whose rows correspond to *time activation sequences*.

In order to *sustain* the conditions of equation 4.8, namely that both $\mathbf{T}$ and $\mathbf{V}$ provide a representation that is nonnegative while approximately modelling the observed matrix $\mathbf{X}$, an algorithm development typically seeks what are known as *multiplicative update rules* such as the ones shown in equations 4.2 and 4.3

We now would like to further explain how equations 4.4 to 4.10 can be used to fully specify an approach to processing and analyzing the physical meaning of matrix representations of audio spectrograms (nonnegative STFT coefficients). At time frequency bin f-n we consider that there exist K total *component bins* that the quantity $\mathbf{T}$ can be sub-indexed at, where

$$\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_k] \in \mathbb{R}^{F \times K}, \qquad \text{where } \mathbf{t}_k = \begin{bmatrix} t_{1,k} \\ t_{2,k} \\ \vdots \\ t_{F,k} \end{bmatrix} \in \mathbb{R}^{F \times 1}$$
$$\text{and } [\mathbf{T}]_{f,k} = t_{fk} = [\mathbf{t}_k]_f \tag{4.11}$$

corresponds to the matrix $\mathbf{T}$ sub-indexed at frequency bin f and component bin k. Secondly we consider that

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_k \end{bmatrix} \in \mathbb{R}^{K \times N}, \qquad \text{where } \mathbf{v}_k = [v_{k,1}, v_{k,2}, ..., v_{k,N}] \in \mathbb{R}^{1 \times N}$$
$$\text{and } [\mathbf{V}]_{k,n} = v_{kn} = [\mathbf{v}_k]_n \tag{4.12}$$

corresponds to the matrix $\mathbf{V}$ sub-indexed at activation bin n and component bin k.

And thus we here have the basic NMF model as given by $\mathbf{X}$ where we populate $\mathbf{X}$ as suggested according to equation 4.9 as given by

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{f1} & x_{f2} & \dots & x_{fn} \end{bmatrix} \approx \mathbf{TV} \in \mathbb{R}^{F \times N}$$

$$\begin{aligned} \mathbf{X} &\approx \mathbf{TV} \\ \mathbf{X} &\approx \sum_{k=1}^{K} \mathbf{t}_k \circ \mathbf{v}_k^T \\ \mathbf{X} &\approx \sum_{k=1}^{K} \mathbf{t}_k \mathbf{v}_k \\ \mathbf{X} &\approx \sum_{k=1}^{K} \begin{bmatrix} t_{1,k} \\ t_{2,k} \\ \vdots \\ t_{F,k} \end{bmatrix} [v_{k,1}, v_{k,2}, ..., v_{k,N}] \end{aligned} \tag{4.13}$$

The sum of all the K components at a time frequency bin then meant model the observed signal's coefficient x_{fn}. We can also consider the reasons why a particular value x_{fn} in the output spectrogram may be either zero or nonnegative when the matrix $\mathbf{F} \in \mathbb{R}^{F \times N}$ represents an audio spectogram (single channel) of nonnegative STFT coefficients:

- First we might note that in a single channel context we might not immediately be concerned with *source separation* but in fact simply *approximate* sound representation of typical or common sounds (i.e. noises, human speakers, background music). Thus we might intend to re-phrase the objective of source separation to simply providing an appropriate sound representation in terms of rank-one nonnegative matrices of various *audio components* describing common sounds in according to only an *approximate* nonnegative audio spectrogram. For example constant amplitude and constant frequency sinusoids that might be modelled within musical mixtures would be an appropriate type of signal for rank-one nonnegative component matrices to represent.

- Within such a nonnegative audio spectrogram analysis x_{fn} being zero would signify that the *coefficient* of the processed single channel spectrogram at time x_{fn}, has an amplitude that must be appropriately modelled as a *silent* time-frequency bin corresponding to the *absence* of a spectra component within the current time frequency bin. Therefore, the representation, according to the sum $\sum_{k=1}^{K} t_{fk} v_{kn}$ across all component bins, must somehow and eventually (by the algorithm) determine an appropriate

parametrization that will reflect this (i.e. the absence of a spectral coefficient at the current time frequency bin). If the time frequency mixture corresponding to $\mathbf{X}$ does not contain too many active sources (i.e. audio components), then it can be said that there exists a certain *plausibility* due to the *harmonic* structure of music and speech, that the neighbouring time activation bins, that is the $(n-1)$th and the $(n+1)$th time-frequency bins, also contain the presence of a zero (i.e. at either $x_{f,n-1}$ or at $x_{f,n-1}$). This would be especially the case in music, where the constant variation of musical activations can be exploited. This was in fact exploited by the research within [61] where a *temporal smoothness* weighting factor was applied to the activation matrix parameters so that the v_{kn} and $v_{k,n-1}$ neighbouring activation bins were more likely to have similar amplitudes (i.e. to enforce that each activation row vector have the property of being *smooth*).

- On the other hand, within such a nonnegative audio spectrogram analysis x_{fn} being non-zero would signify that *coefficient* of the processed single channel spectrogram at time x_{fn}, has an amplitude that must be appropriately modelled as a *non-silent* time-frequency bin. Therefore the sum of the components in this time frequency bin, that is $\sum_{k=1}^{K} t_{fk}v_{kn}$ is required to sum to something nonnegative, and not in fact zero, this time. We can attempt to explain the scenario with inequalities, if we suppose that the processed and *observed* STFT coefficient x_{fn} at time frequency bin f-n has some particular value c such that $x_{fn} = c$, but currently the sum $\sum_{k=1}^{K} t_{fk}v_{kn} > c$, then within one iteration we would ideally expect that the *multiplicative update rules* for $\mathbf{T}$ and $\mathbf{V}$ update the NMF representation such that the sum $\sum_{k=1}^{K} t_{fk}v_{kn}$ as computed within the time frequency bin f-n *converges* towards the value of c, but in practice we cannot guarantee this (since the columns of $\mathbf{T}$ combined with the rows $\mathbf{V}$ provide a set of K rank-one approximations upon which other time-frequency bins are dependent) for any particular set of update rules. We will show however that for the basic NMF model, convergence according to the described scenario is typically good, assuming that an appropriately good choice of the number of components K needed to well-represent the mixture X is made.

With this overview of the two factor $\mathbf{T}$ and $\mathbf{V}$ model, we suggest for the reader to further consult the referenced work of [16] in order to consider how the various divergences introduced within sections A.1.0.1 to A.1.0.3 were applied to the two factor model for each particular distance/divergence and for obtaining in each case a different maximum likelihood and probabilistic interpretation and corresponding set of multiplicative update rules for

updating the factors $\mathbf{T}$ and $\mathbf{V}$.

Indended as a helpful illustration, Figure 4.1 graphically demonstrates the general concept in a scenario where we suppose that we will attempt to apply the two-factor basic NMF model to the processed STFT of a signal consisting of a sequence of non-overlapping musical notes.

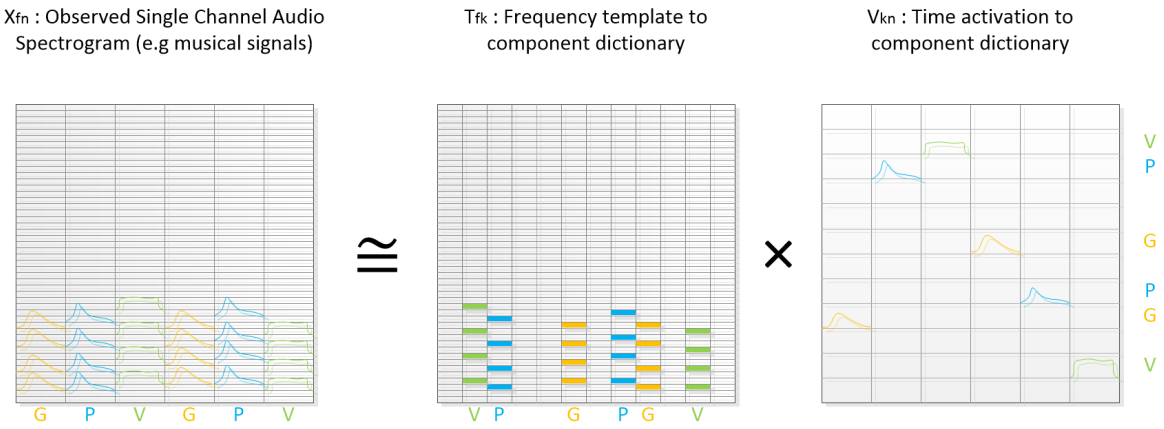

Figure 4.1: Two Factor NMF Illustration applied to additively combined spectrogram of musical notes.

The figure illustrates a converged representation that could occur by iteratively applying the Lee and Seung multiplicative update rules [1, 16] when the number of NMF basis (rank-one nonnegative matrix components) has been chosen as $K = 12$. If the each of the notes in the matrix $\mathbf{X}$ are well represented as a rank-one nonnegative NMF component, then in fact choosing the number of NMF bases as $K = 6$ may be sufficient, since we have knowledge from having indeed constructed the observed matrix $\mathbf{X}$ that there should only be in fact 6 notes that occur sequentially.

If the $\mathbf{T}$ and $\mathbf{V}$ illustrated within Figure 4.1 is to be used for source separation purposes, we should take this opportunity to point out two important facts that can be easily at this point observed.

- The NMF bases for $k = 1, ..., 12$ occur in an unordered fashion. That is, each rank one matrix component conveys the appropriate frequency and temporal information about the musical note, however, in order to do processing and extract meaning from the set of NMF bases, we have to proceed under the assumption that we are unable to extract anything meaningful from the order in which the NMF bases for $k = 1, ..., 12$ are outputted by the NMF algorithm.

- Secondly, although for each instrument (i.e. Guitar, Piano, Violin) we might be able to reconstruct and listen to notes individually, and recognize in fact that there exists

two notes from each instrument, without considering additional techniques applied to the two-factor **T** and **V** model, NMF alone suggests no way to achieve the grouping of notes common to each instrument.

4.2 Review of Chapter

In Chapter 4 we introduced for the first time in detail nonnegative matrix factorization (NMF) whose derivations play a key role in inspiring and deriving algorithms relevant to STFT domain source separation such as complex nonnegative matrix factorization algorithms (CNMF) and spatial covariance matrix (SCM) NMF algorithms, that are to be introduced in the next chapter. The basic NMF model was first introduced in section 1.3.2.1 and showed a complete derivation of update rules, corresponding to the obtain rules specified within [1] by Lee and Seung.

A sufficient understanding of the topics covered to this point in the thesis, as well as an understanding of STFT methods applied to audio signals allows us to propose that although NMF provides a strong starting point for algorithmically achieving source separation of audio spectrogram representations, the single channel two-factor NMF model leaves something more to be desired in terms of becoming applicable to multichannel STFT based source separation algorithms. Under the working assumption that was specified in section 1.5, that the proposed algorithm shall utilize two or more closely spaced microphone sensors, the question remains, upon what principle should we classify the additively combined spatial images of source signals to be separated, as explained in section 1.4.

We also must note however, that for considering multichannel STFT representations, the update rules specified in section 4.1.2 unfortunately are known to be not entirely adequate to achieving source separation of sources whose spectra necessarily might overlap in the time-frequency domain of audio spectrograms. Thus we will seek to know how the two factor model (harmonic template dictionary and time activation dictionary) can be applied as a signal processing tool in a *multichannel* audio context and if it can be shown to be capable of providing not just *audio representations* but further *separable* audio representations. Furthermore we would not only like to consider the representation of nonnegatively processed spectograms but processing techniques that appropriately persist the complex phase part of the observed multichannel STFT spectrogram. In the next chapter we demonstrate that there exist techniques for processing the observed data that satisfies this requirement, as well as NMF based models that are capable of being modelled with corresponding multiplicative update rules that in the end goal (with the addition of source clustering and source reconstruction procedures) do in fact achieve source separation of signals in a BSS

and underdetermined scenario.

4.3 Considerations for extending Single Channel NMF to Multichannel NMF

Beyond this point we begin to focus exclusively on multichannel STFT source separation based algorithms since the proposed algorithm is also meant to be applied to STFT data occuring from a multichannel microphone array. However, since the emphasis will become how to formulate and understand models that parametrize a complex observed spectrogram matrix (as opposed to a nonnegative one) we will consider one last single algorithm known as CNMF [25] as detailed by Kameoka et al.

Subsequently, after having considered this model, we will re-exercise our knowledge of multichannel Wiener filtering based techniques in order to consider how to process by a similar analogy, the observed complex-valued data occuring from the complex STFT spectogram in terms of the so called covariance matrices that occur at each time frequency bin.

It should become evident by reconsidering section 2.5 and the corresponding topics covered in Chapter 2 that a very simple way to consider the uniting of NMF based models and the multichannel Wiener filter used for multichannel source separation is to focus on the concept of *parametrizing* the source variance term of the multichannel Wiener filter (i.e. V_{jnf}) as described within equation 2.38 using the two-factor (T and V) NMF model introduced in Chapter 3, and for instance according to equation 4.13. In principle, the conceptual connection here is appropriate since V_{jnf} specifies a jth source index (i.e. label) and by applying a component based representation such as NMF, a linking of the jth source to the kth NMF component label can be achieved. This will be as illustrated within the next chapter but can be already noticed as early as at this point in the thesis.

Thus, although perhaps the two factor NMF model alone is not entirely appropriate for analyzing scenarios of multichannel source separation, it becomes strikingly useful when combined in this manner with the multichannel Wiener filter model of section 2.5. This is best demonstrated, in the context of SCM NMF based algorithms, within the [16] and [24] algorithms. Essentially, an important component of the thesis will be to focus mainly on these two algorithms as illustrative examples into obtaining meaningful insight into the highly interrelated concepts of spatial covariance matrix (SCM) modelling per TF bin, clustering (linking) of sounding components across TF bins, and source reconstruction that are collectively applied in combination.

Chapter 5

Single Channel and Multi-Channel Frequency Domain Blind Source Separation Algorithms

5.1 Overview of current Chapter

In the previous chapter we focused on developing the principles required to grasp the algorithms contained within the current chapter as well as the following chapters.

The algorithms to be seen in the following chapter that best illustrate the two key concepts of *clustering* and *source reconstruction* are the ones developed in regards to SCM-based processing of time-frequency dependent covariance matrices in an NMF based context [16, 24]. Where as clustering takes as input the learned spatial audio features and outputs indicators that specify optimal correspondences between output classes and NMF bases, source reconstruction utilizes both the output of the clustering and the NMF bases to provide time-frequency estimates of parametrized source signals.

The joint (i.e. coupled) estimation framework of these algorithms alleviates the issue of permutation and frequency dependent scaling and ambiguity alignment issues that are sometimes present within other frequency domain blind and undetermined source separation algorithms. That is, there exist other frequency domain source separation algorithms that are not based in principle on NMF, but do also attempt to exploit frequency dependent phase difference (time difference of arrival between microphones) information in order to achieve underdetermined BSS, such as the algorithm described in [38]. This model for instance, although somewhat relevant, is less relevant in principle to the proposed research than SCM NMF algorithms, which will appropriately be the focus of this chapter. Update rules and meaningful interpretations for SCM NMF algorithms will be demonstrated in such a way that one need only consider how to interpret the converged NMF parameter set and to further consider if spatial audio features are fully reflected within certain converged NMF parameters, treated as the output of the iterative SCM NMF algorithm and the input to

clustering and source reconstruction.

Another key point of emphasis of the current chapter will be to make evident the importance of the multichannel Wiener filter (equation 2.41) as having provided SCM NMF models with:

- A procedure to estimate spatial image vectors according to equation 2.41. Reconstruction of spatial images per time frequency bin corresponds to achieving source separation in the multichannel STFT domain.
- The notion of a sum of product representation of the multichannel covariance matrix according to equations 2.40 and 2.38.
- As explained in section 4.3, a connection to NMF that allows the basic two factor (harmonic template dictionary and activation dictionary) to be applied for the purpose of spectrally representing, per output source signal, audio components described in a rank-one nonnegative sense, that can be further manipulated within the SCM NMF algorithm, depending on what is considered the optimal way to cluster and/or reconstruct source estimates.

Before introducing SCM NMF based models though, the first topic that will be considered, we first shall propose is of nearly equal importance to consider carefully. In [25], an algorithm that extends the basic NMF model was detailed and it was proven that the basic two factor NMF model could be extended in such a way so as to associate phase spectra, characterized by a *phase dictionary* that was demonstrated could provide an enhanced decomposition of source STFT spectra. During the time at which this was introduced, few NMF models had been developed in audio signal processing literature that could reasonably derive a parametrization and appropriate learning rules for such a so-called phase dictionary. For this reason, the algorithm is included within the chapter and will be considered in detail prior to considering the multichannel SCM NMF algorithms. We first introduce this algorithm in the next section.

5.2 Complex NMF

With some renaming of the parameters and indices the following is an overview of the single channel CNMF algorithm by Kameoka et al. [25]. In addition to [25], single channel CNMF was also investigated by Brian King in [62].

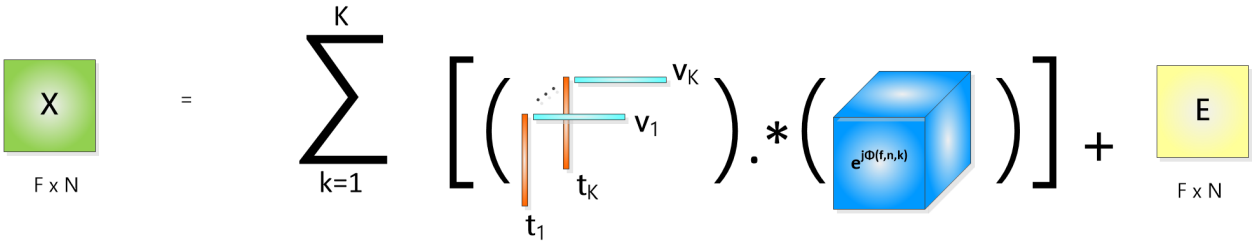

Figure 5.1: Kameoka et al. CNMF Model Illustration

Figure 5.1 details a pictorial representation of the single channel STFT algorithm and its parametrization, which is an extension of the basic NMF model as detailed in section 1.3.2.1.

Presented below is the modelling of a single channel of STFT data according to

$$F_{f,n} = \sum_{k} T_{f,k} V_{k,n} \exp\left(j\Phi_{k,f,n}\right) \tag{5.1}$$

and

$$Y_{f,n} = F_{f,n} + \epsilon_{f,n}. \tag{5.2}$$

The above model suggests that a single channel of STFT data can be modelled with the use of a harmonic frequency template dictionary $T_{f,k}$, and an activation matrix parameter $V_{n,k}$ and additionally, and differently from other conventional NMF methods that attempt to model the magnitude spectrogram of the STFT, it introduces a complex phase quantity $\exp\left(j\Phi_{k,f,n}\right)$, targeted at quantifying the initial phase shifts associated with the single channel STFT's components. Similar to some of the models that follow, the component index is integrated out per STFT time-frequency bin, so that the output quantity $F_{f,n}$ has no notion of component; only the frequency and time activation indices f and n persist. In this type of model, it can be taken to be true that a closer approximation of the data can be achieved in the model by choosing a higher value for the number of components K.

It will be of importance to show that an update rule for $\exp\left(j\Phi_{k,f,n}\right)$, which although clearly does not represent a nonnegative quantity, can still be derived in a sensible way and "from inspection" of the definition of the proposed auxiliary function, so to speak, such that the update rule is able to minimize the objective function associated with the auxiliary function upon which the other update rules are also derived.

The likelihood and consequent log-likelihood functions are now listed here below

$$P(Y|\theta) = \prod_{f,n} \mathcal{N}_c(Y_{f,n}|F_{f,n}, \sigma^2)$$
$$\propto \prod_{f,n} \frac{1}{\pi\sigma^2} \exp\Big(- \frac{|Y_{f,n} - F_{f,n}|^2}{\sigma^2} \Big) \tag{5.3}$$

Where $\theta = \{T, V, \Phi\}$ represents the single channel CNMF parameter set.

We then seek to minimize the negative log-likelihood function plus a sparsity term, defined to be

$$f(\theta) = \sum_{f,n} |Y_{f,n} - F_{f,n}|^2 + 2\lambda \sum_{k,n} |V_{k,n}|^p \tag{5.4}$$

$$\text{subject to} \sum_f T_{f,k} = 1$$

Before even attempting to understand the update rule derivations, we can notice some properties of the minimization problem to be solved.

The objective function introduces a second term intended to introduce sparsity into $V_{n,k}$ by setting up the built-in condition that the $|\cdot|^p$ p-norm of $V_{n,k}$ be minimized as well.

Lastly, this particular algorithm sets up the constraint that summing over the frequency bins in any particular component bin should add to unity. Hopefully it is clear that this constraint has the purpose of normalizing the frequency-dependent columns of the $T_{f,k}$ parameter, such that no particular column or "basis vector" is able to significantly overpower any other column of the matrix parameter, for any particular component index k.

The following two inequalities are used to construct an auxiliary function.

$$|Y_{f,n} - F_{f,n}|^2 \leq \sum_k \frac{|\bar{Y}_{k,f,n} - T_{f,k}V_{k,n} \exp(j\Phi_{k,f,n})|^2}{\beta_{k,f,n}} \tag{5.5}$$

$$|V_{k,n}|^p \leq \frac{p|\bar{V}_{k,n}|^{p-2}}{2} {V_{k,n}}^2 + |\bar{V}_{k,n}|^p - \frac{p|\bar{V}_{k,n}|^p}{2} \tag{5.6}$$

Using both the defintion of the model 5.1 and the fact that the algorithm defines the condition that the latent components sum to the f-nth measurement that is, $\sum_k \bar{Y}_{k,f,n} = Y_{f,n}$,

5.5 can be explicitly written in a way that shows that it follows more noticeably from Jensen's Inequality:

$$|\sum_{k} \bar{Y}_{k,f,n} - T_{f,k}V_{k,n}\exp(j\Phi_{k,f,n})|^2 \leq \sum_{k} \frac{|\bar{Y}_{k,f,n} - T_{f,k}V_{k,n}\exp(j\Phi_{k,f,n})|^2}{\beta_{k,f,n}} \tag{5.7}$$

And it can be verified for both inequalities 5.5 and 5.6, that the terms on the right hand side bound the terms on the left hand side from above, and a tangency (equality) occurs when the auxiliary variables $\bar{Y}_{k,f,n}$ and $\bar{V}_{k,n}$ each take on a particular value. These are:

$$\bar{Y}_{k,f,n} = T_{f,k}V_{k,n}\exp(j\Phi_{k,f,n}) + \beta_{k,f,n}(Y_{f,n} - F_{f,n}) \tag{5.8}$$

and

$$\bar{V}_{k,n} = V_{k,n} \tag{5.9}$$

Altogether, these properties prove that the function

$$f^{+}(\theta,\bar{\theta}) = \sum_{k,f,n} \frac{|\bar{Y}_{k,f,n} - T_{f,k}V_{k,n}\exp(j\Phi_{k,f,n})|^2}{\beta_{k,f,n}} + \lambda \sum_{k,n} \left(p|\bar{V}_{k,n}|^{p-2}{V_{k,n}}^2 + 2|\bar{V}_{k,n}|^p - p|\bar{V}_{k,n}|^p \right) \tag{5.10}$$

is an auxiliary function for $f(\theta)$ and λ is a scalar parameter used for controlling the amount of sparsity to introduce into $V_{k,n}$. Update rules are derived by iteratively minimizing $f(\theta,\bar{\theta})$, locally. Since $f(\theta,\bar{\theta})$ bounds $f(\theta)$ from above, and both functions are composed of constituent convex functions, minimization the auxiliary function $f(\theta,\bar{\theta})$ should lead to minimization of the objective function $f(\theta)$, in an EM-like fashion, but for convex functions as opposed to concave ones.

A key significance of this complex NMF algorithm is the inclusion of the initial phase parameter as part of the NMF model. Since the complex NMF model in this particular case models only a single channel STFT as opposed to a multichannel STFT, the spatial covariance method cannot be exploited in order to determine interchannel properties of sound sources for localizing sounds in an acoustic environment in which sounds components are characterized by their interchannel time differences or equivalently their direction of arrival with respect to the multichannel microphone array.

5.2.1 Update Rules

Optimizing $f^{+}(\theta,\bar{\theta})$ with respect to $T_{f,k}$ and $V_{k,n}$ results in the following update rules

$$T_{f,k} \leftarrow \frac{\sum_n \frac{V_{k,n}}{\beta_{k,f,n}} \mathrm{Re}\{\bar{Y}^*_{k,f,n} \exp(j\Phi_{k,f,n})\}}{\sum_n \frac{V^2_{k,n}}{\beta_{k,f,n}}} \tag{5.11}$$

$$V_{k,n} \leftarrow \frac{\sum_f \frac{T_{f,k}}{\beta_{k,f,n}} \mathrm{Re}\{\bar{Y}^*_{k,f,n} \exp(j\Phi_{k,f,n})\}}{\sum_f \frac{T^2_{f,k}}{\beta_{k,f,n}} + \lambda p \left|\bar{V}_{k,n}\right|^{p-2}} \tag{5.12}$$

$$\exp(j\Phi_{k,f,n}) \leftarrow \frac{\bar{Y}_{k,f,n}}{\left|\bar{Y}_{k,f,n}\right|} \tag{5.13}$$

In closing, the derivation of the phase update rule (5.13) will be explained.

First, the algorithms suggests to express 5.10 as a function of terms that do and do not depend on $\Phi_{k,f,n}$. By using c to denote those that do not, $f^+(\theta, \bar{\theta})$ can be written as follows

$$f^+(\theta, \bar{\theta}) = c - 2\sum_{k,f,n} |A_{k,f,n}| \cos(\Phi_{k,f,n} - C_{k,f,n}) \tag{5.14}$$

where

$$C_{k,f,n} = \arg(\bar{Y}_{k,f,n}) \text{ , and } |A_{k,f,n}| = \frac{\bar{Y}_{k,f,n} T_{f,k} V_{k,n}}{\beta_{k,f,n}}$$

By inspection of 5.14, it should be evident that the negative term of interest, following c is maximum (ie: the function is minimized, since the negative term then becomes "as negative as possible") when the term inside of it, $\cos(\Phi_{k,f,n} - C_{k,f,n}) = 1$.

For the cosine of a difference to be unity, the two terms must in fact be equal to each other. And therefore, we should then derive the update rule such that we set the quantity $\Phi_{k,f,n}$ to have the same value as $C_{k,f,n}$. Hence, we come upon the update rule defined in 5.13, where $\exp(j\Phi_{k,f,n})$ is taken to have the same phase as the auxiliary variable $\bar{Y}_{k,f,n}$, with the additional desired property of also having unit magnitude.

5.3 Spatial Covariance NMF Models

5.3.1 Signal Representation and Spatial Covariance Matrix (SCM) Processing

We present and suggest here the procedure for computing spatial covariance matrices from multi-channel STFT data, as described in detail in both [24] and [16]. This method extends

the methods described by equations 4.9 and 4.10 to the pre-processing of *multichannel* STFT observations. We start by arranging each STFT bin into an $\mathbb{C}^{M\times 1}$ column vector of complex-valued STFT coefficients.

$$\mathbf{x}_{fn} = \begin{bmatrix} x_{fn1} \\ x_{fn2} \\ \vdots \\ x_{fnM} \end{bmatrix} \in \mathbb{C}^{M\times 1} \tag{5.15}$$

Here, we use $\hat{\mathbf{x}}_{fn}$ to denote a corresponding processed version of the vector which is consequently the square-rooted version of the above f-n'th STFT bin column vector

$$\hat{\mathbf{x}}_{fn} = \begin{bmatrix} |x_{fn1}|^{1/2}\,\mathrm{sign}(x_{fn1}) \\ |x_{fn2}|^{1/2}\,\mathrm{sign}(x_{fn2}) \\ \vdots \\ |x_{fnM}|^{1/2}\,\mathrm{sign}(x_{fnM}) \end{bmatrix} \in \mathbb{C}^{M\times 1} \tag{5.16}$$

where $\mathrm{sign}(z) = \dfrac{z}{|z|}$ represents the signum function for complex numbers. It divides out the original complex number z by it's magnitude and only persists the phase part. The resulting output quantity is thus a complex number with unit magnitude, but with the same phase as z had, originally.

To compute the spatial covariance matrix quantity that we seek, we simply consider computing an outer-product like operation on the vector on the processed quantity $\hat{\mathbf{x}}_{fn}$:

$$\mathbf{X}_{fn} = \hat{\mathbf{x}}_{fn}\hat{\mathbf{x}}_{fn}^{H} \tag{5.17}$$

Where the $\hat{\mathbf{x}}_{fn}^{H}$ is a row vector that represents the Hermitian transpose of $\hat{\mathbf{x}}_{fn}$. In the same fashion as the models presented in [24] or [16], the observed spatial covariance matrix can be modelled as a function of the NMF algorithm parameters as

$$\mathbf{X}_{fn} \approx \sum_{k=1}^{K} \mathbf{H}_{fk}\hat{s}_{fnk} \tag{5.18}$$

where the term here $\hat{s}_{fnk}$ refers to *source component* of the sound mixture at time frequency bin n-f. In an SCM NMF algorithm $\hat{s}_{fnk}$ is intended to represent the variance of the kth NMF sound component at time-frequency bin n-f where the meaning of $\hat{\mathbf{s}}_{fnk}$ here is almost analogous to the meaning of the source variance term V_{jnf} within equation 2.38. But when the observed vector $\hat{\mathbf{x}}_{fn}$ has first been processed by taking its elementwise square root according to equation 5.16 then it can in fact be interpreted simply as the nonnegative amplitude or magnitude of the source component at time frequency bin n-f. It will be seen that an SCM NMF model seeks to extend the two factor basic NMF model to *decompose*

the 3 way tensor $\hat{s}_{fnk}$ into a set of K rank-one nonnegative matrices that each describe the kth NMF component at time-frequency bin n-f.

To model the effect of spatial filtering between the kth NMF component and the mth microphone the term $\mathbf{H}_{fk}$ represents an interchannel covariance matrix which can be modelled as outer product

$$\mathbf{H}_{fk} = \mathbf{h}_{fk}\mathbf{h}_{fk}^{H} + \epsilon_{fk}\mathbf{I} \tag{5.19}$$

Where $\mathbf{h} = [h_1, ..., h_M]^T \in \mathbb{C}^M$ represents the Fourier transform of the impulse response from the k-th source component to the m-th microphone [16]. We note here that the main diagonal elements, which under the computed spatial covariance matrix method are necessarily real valued, represent the power gain of the k-th basis sound component at frequency bin f to each microphone m. We need only emphasize the previous point as well as the fact that the off diagonal elements are in fact complex-valued and provide a frequency-domain phase difference between microphones. If the spatial geometry of the microphone is known, and considered in advance, as in [24], then the frequency-domain phase difference interpretation can be easily converted to a time domain quantity which corresponds to the time difference of arrival between microphones.

Thus it can be seen that the *sum of product* form of equation 5.18 descriptively explains how the *observed* SCM matrix $\mathbf{X}_{fn}$ occurred as a result of a structured parametrization of the model parameters at each time frequency bin n-f. Namely, there exists the spatial consideration, as specified by the matrix $\mathbf{H}_{fk}$, as well as the temporal and frequency-dependent consideration to be explained by the learning of *magnitude spectra* of sources, to be learned within the development of the algorithm, as specified by $\hat{s}_{fnk}$.

Given the properties of this interpretation, it is also worth mentioning that in treating the spatial covariance matrices per multichannel STFT time-frequency bin as the observed data, makes it somewhat ambiguous or unknown in terms of how to extract information about the *initial* phase of sound sources as first introduced in section 2.1.1.2. As spatial covariance matrix processing on the observed data has as a strong point the capability of revealing interchannel phase difference quantities, in the off diagonal elements, but due to the squaring of the k-th source component's complex STFT coefficient per STFT bin, information about the initial phase would be lost, as an unavoidable and undesirable consequence of the spatial covariance matrix processing, and precisely, the squaring of the observed vector observed vector $\hat{\mathbf{x}}_{fn}$ by its Hermitian transpose $\hat{\mathbf{x}}_{fn}^{H}$. If a method was available to achieve it, correct estimation the initial phase of NMF sound components, as was shown could be achieved in a single channel CNMF model, we hypothesize to be able to significantly enhance the perceptual sound quality of the source separated output signals that we seek in a multichannel

CNMF model.

The method that we eventually suggest, and that we believe has not been as explored as often as for SCM NMF models, is to seek to *directly* explain the observed vector $\hat{\mathbf{x}}_{fn}$ per time frequency bin in terms of an appropriate parametrization, as opposed to focusing on explaining its spatial covariance matrix $\mathbf{X}_{fn}$ per time frequency bin.

If we once again consider any of the M elements at a particular value m, we focus on the fact that the initial phase is extractable a precise value can be used to consider its value by computing the quantity

$$\arg(x_{fnm}) \tag{5.20}$$

where $\arg(\cdot)$ is the complex argument operator that returns the phase of a complex number and x_{fnm} is the mth observation at time frequency bin f-n.

We can note that in 5.15, the absolute (initial) phase term is observable per microphone STFT coefficient observed at microphone bin m. We propose according to equation 5.16 that the observed initial phase is explainable in terms of source initial phases so long as we are able to separately the microphone dependent phase shift associated microphone m. This property is important later in the thesis and within the next chapter, but it is in fact not typically of great significance towards the modelling of sources in an SCM NMF based algorithm.

The difference in philosophy corresponds to providing the CNMF algorithm with a different *target* to converge to. Ideally, we propose the philosophy that a desirable parametrization should be able to explain *both* the processed covariance matrix $\mathbf{X}_{fn}$ *as well* as the observed vector $\mathbf{x}_{fn}$. However, we will see that this is actually difficult to achieve in practice, that is, to monotonically ensure that a *initial phase learning* rule as well as an effective *interchannel phase learning* rule can both be prioritized simultaneously in order to provide an optimal representation and/or decomposition of both $\mathbf{x}_{fn}$ and $\mathbf{X}_{fn}$. We will now more thoroughly consider SCM NMF models in order to provide an analysis of their merits for being able to explain $\mathbf{X}_{fn}$.

5.3.2 EU-NMF (Squared Euclidean distance) Spatial Covariance NMF

The following algorithm presents a strong case for why multichannel algorithms should definitely consider the spatial covariance matrix (SCM) interpretation of nonnegative matrix factorization with complex-valued data in order to successfully achieve meaningful source

separation of sounds with potentially spatially redundant (i.e. similar) as well as dissimilar spatial properties. In the following work by Sawada et al. [16] two key NMF variants based upon the commonly used Euclidean distance and Itakura-Saito divergence measures were applied to solving the source separation problem in the multichannel and multi-source (possibly overlapping) time frequency domain (STFT domain).

As such, *four* sets of multiplicative update rules were included in the presentation of the work. Two per multichannel divergence measure, and two corresponding to a clusterable, and non-clusterable set of multiplicative update rules. In the following section we focus on analyzing the clusterable Euclidean distance based set of multiplicative update rules, and the derivation surrounding it, however, the modelling and parametrization of the SCM model for the other sets of multiplicative update rules are fairly similar and are based upon the same basic SCM model. In the previous section the basic SCM model was generically described as a *sum-of-product* form for additively representing the observed multichannel SCM, $\mathbf{X}_{fn}$, per time-frequency bin as previously described by equation 5.18.

In order to re-parametrize equation 5.18 in such a way that resembles a two-factor NMF representation, the term $\hat{s}_{fnk}$ is replaced with the matrices t_{fk} and v_{kn}, which together intend to model the effect of the magnitude spectra of audio components, which can be considered separately according to the component bin index variable k, and which provide a nonnegative time-frequency representation of the kth audio component.

$$\begin{aligned}[\hat{\mathbf{X}}]_{fn} &= \sum_{k}^{K} [\mathbf{H}]_{fk} t_{fk} v_{kn} \\ &= \sum_{k}^{K} \left(\sum_{l}^{L} [\mathbf{H}]_{fl} z_{lk} \right) t_{fk} v_{kn} \end{aligned} \tag{5.21}$$

$$[\mathbf{H}]_{fk} = \sum_{l}^{L} [\mathbf{H}]_{fl} z_{lk} \tag{5.22}$$

Here, the matrix $[\mathbf{H}]_{fk}$ is again specified for the purpose of modelling the spatial property of the k-th NMF basis at frequency bin f. As mentioned, it can be considered that $\mathbf{H}_{fk}$ represents an interchannel covariance matrix, which is now augmented as previously described in equation 5.18 with the NMF parameter z_{lk}, and now a class dependent interchannel covariance matrix $\mathbf{H}_{fl}$. An update rule can be derived such that z_{lk} converges to a plausible and partitioned state such that the class variable l specifies a correspondence between lth class and the kth NMF basis. Another interpretation of z_{lk} is that it associates in each of its elements, a *probability* that the kth NMF basis component should be associated with the

learned spatial features of class l and the spatially parametrized matrix $\mathbf{H}_{fl}$. Therefore, z_{lk} here is referred to as cluster indicator latent variable. In [16] it is described as a key parameter whose configuration is learned via iteratively applying its multiplicative update rule and later using it as a spatial audio feature which can be used as the input to a clustering method.

Proceeding with an optimization procedure that iteratively updates the parametrized covariance matrix $\hat{\mathbf{X}}_{fn}$, based upon the condition that it must converge appropriately towards the actual observed covariance matrix $\mathbf{X}_{fn}$ (computed as an outer product from the observed multichannel STFT vector at time frequency bin f-n) a *likelihood function* corresponding to a multichannel Euclidean distance based cost function is given by

$$\begin{aligned} P(\mathbf{X}|\mathbf{T},\mathbf{V},\mathbf{H}) &= \prod_{f=1}^{F}\prod_{n=1}^{N}\prod_{m=1}^{M}\prod_{n=1}^{M}\mathcal{N}_c\Big([[\mathbf{X}]_{f,n}]_{mn}\Big|[[\hat{\mathbf{X}}]_{f,n}]_{mn},1\Big) \\ &\propto \exp(-||\mathbf{X}_{fn}-\hat{\mathbf{X}}_{fn}||_F^2) \end{aligned} \tag{5.23}$$

Taking the $\log(\cdot)$ of equation 5.23 results in the cost function that we aim towards minimizing. We then formulate the optimization problem as the task of minimizing the negative log-likelihood as given by

$$D_{Eu}(\mathbf{X},\hat{\mathbf{X}}) = \sum_{f=1}^{F}\sum_{n=1}^{N}||[\mathbf{X}]_{fn}-[\hat{\mathbf{X}}]_{fn}||_F^2$$

$$D_{Eu}(\mathbf{X},\{\mathbf{T},\mathbf{V},\mathbf{H},\mathbf{Z}\}) = \sum_{f=1}^{F}\sum_{n=1}^{N}||[\mathbf{X}]_{fn}-\sum_{k}^{K}\Big(\sum_{l}^{L}[\mathbf{H}]_{fl}z_{lk}\Big)t_{fk}v_{kn}||_F^2 \tag{5.24}$$

and re-writing it by substituting 5.21 into 5.23 results in the objective function f as given by

$$\begin{aligned} f(\mathbf{T},\mathbf{V},\mathbf{H}) = &\sum_{f,n}\mathrm{tr}\Big[\Big(\sum_{k}[\mathbf{H}]_{fk}t_{fk}v_{kn}\Big)\Big(\sum_{k}[\mathbf{H}]_{fk}t_{fk}v_{kn}\Big)^H\Big] \\ &-\sum_{f,n,k}t_{fk}v_{nk}\,\mathrm{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fk}^H) - \sum_{f,n,k}t_{fk}v_{nk}\,\mathrm{tr}([\mathbf{H}]_{fk}[[\hat{\mathbf{X}}]_{fn}^H \end{aligned} \tag{5.25}$$

where we apply the definition of the Frobenius norm in order to expand the terms. It is suggested in [16] that multiplicative update rules can be derived by considering an auxiliary function corresponding to the objective function of 5.25 as given by

$$f^{+}(\mathbf{T},\mathbf{V},\mathbf{H},\mathbf{R}) = \sum_{f,n,k} t_{fk}^{2} v_{nk}^{2} \operatorname{tr}([\mathbf{H}]_{fk}\mathbf{R}_{fnk}^{-1}[\mathbf{H}]_{fk}^{H}) - \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fk}^{H}) - \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}(\mathbf{H}]_{fk}[\hat{\mathbf{X}}]_{fn}^{H}) \tag{5.26}$$

and then optimizing the *auxiliary function* with respect to each of the NMF parameters. In equation 5.26, the $\mathbf{R}_{fnk} \in \mathbb{C}^{M\times M}$ are auxiliary matrix variables that are hermitian positive definite. Viewed as a set of matrix components in a particular time-frequency bin, the constraint that they must sum to the identity matrix is set up. That is, $\sum_{k} \mathbf{R}_{fnk} = \mathbf{I}$

The following properties with regards to f, f^{+} and the various NMF parameters can be verified:

1. $f(\mathbf{T},\mathbf{V},\mathbf{H}) \leq f^{+}(\mathbf{T},\mathbf{V},\mathbf{H},\mathbf{R})$.

2. $f(\mathbf{T},\mathbf{V},\mathbf{H}) = \min_{\mathbf{R}} f^{+}(\mathbf{T},\mathbf{V},\mathbf{H},\mathbf{R})$. Specifically, this can be achieved by setting up a Lagrangian function from f^{+} and minimizing it with respect to $\mathbf{R}_{fnk}$, and invoking the constraint $\sum_{k} \mathbf{R}_{fnk} = \mathbf{I}$ in order to find a minimizing expression for $\mathbf{R}_{fnk}$ in terms of the parameters. This update rule for $\mathbf{R}_{fnk}$ can be shown to be ([16], Appendix):

$$\mathbf{R}_{fnk} = [\hat{\mathbf{X}}]_{fn}^{-1}[\mathbf{H}]_{fk} t_{fk} v_{nk} \tag{5.27}$$

3. It can easily be shown that subsituting 5.27 back into 5.26 achieves the original objective function f:

$$\mathbf{R}_{fnk}^{-1} = \frac{[\mathbf{H}]_{fk}^{-1}[\hat{\mathbf{X}}]_{fn}}{t_{fk}v_{nk}}$$

$$
\begin{aligned}
f^{+}(\mathbf{T},\mathbf{V},\mathbf{H},\mathbf{R}) &= \sum_{f,n,k} t_{fk}^2 v_{nk}^2 \operatorname{tr}([\mathbf{H}]_{fk}\mathbf{R}_{fnk}^{-1}[\mathbf{H}]_{fk}^{H}) \\
&- \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fk}^{H}) - \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}(\mathbf{H}]_{fk}[\hat{\mathbf{X}}]_{fn}^{H}) \\
&= \sum_{f,n,k} t_{fk}^2 v_{nk}^2 \operatorname{tr}([\mathbf{H}]_{fk}\Big(\frac{[\mathbf{H}]_{fk}^{-1}[\hat{\mathbf{X}}]_{fn}}{t_{fk}v_{nk}}\Big)[\mathbf{H}]_{fk}^{H}) \\
&- \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fk}^{H}) - \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}(\mathbf{H}]_{fk}[\hat{\mathbf{X}}]_{fn}^{H}) \\
&= \sum_{f,n} \operatorname{tr}([\hat{\mathbf{X}}]_{fn} \sum_{k} [\mathbf{H}]_{fk}^{H} t_{fk}v_{nk}) \\
&- \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fk}^{H}) - \sum_{f,n,k} t_{fk}v_{nk} \operatorname{tr}(\mathbf{H}]_{fk}[\hat{\mathbf{X}}]_{fn}^{H}) \\
&= f(\mathbf{T},\mathbf{V},\mathbf{H})
\end{aligned}
$$

4. Further minimizing f^{+} with respect to its parameters $\mathbf{T},\mathbf{V},\mathbf{H}$, which by virtue of f^{+} being the auxiliary function for f, minimizes f as well, and leads to the update rules shown in the next section.

5.3.3 Update Rules

The multiplicative update rules obtained by optimizing the auxiliary function f^{+} in each of the NMF parameters are given as

$$
t_{fk} \leftarrow t_{fk} \frac{\sum_{l} z_{lk} \sum_{n} v_{kn} \operatorname{tr}([\mathbf{X}]_{fn}[\mathbf{H}]_{fl})}{\sum_{l} z_{lk} \sum_{n} v_{kn} \operatorname{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fl})} \tag{5.28}
$$

$$
v_{kn} \leftarrow v_{kn} \frac{\sum_{l} z_{lk} \sum_{f} t_{fk} \operatorname{tr}([\mathbf{X}]_{fn}[\mathbf{H}]_{fl})}{\sum_{l} z_{lk} \sum_{f} t_{fk} \operatorname{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fl})} \tag{5.29}
$$

$$
z_{lk} \leftarrow z_{lk} \frac{\sum_{f,n} t_{fk}v_{kn} \operatorname{tr}([\mathbf{X}]_{fn}[\mathbf{H}]_{fl})}{\sum_{f,n} t_{fk}v_{kn} \operatorname{tr}([\hat{\mathbf{X}}]_{fn}[\mathbf{H}]_{fl})} \tag{5.30}
$$

$$[\mathbf{H}]_{fl} \leftarrow [\mathbf{H}]_{fl} .^{*} \left(\sum_{k} z_{lk} t_{fk} \sum_{n} v_{kn} [\hat{\mathbf{X}}]_{fn} \right)^{-1} \times \left(\sum_{k} z_{lk} t_{fk} \sum_{n} v_{kn} [\mathbf{X}]_{fn} \right) \tag{5.31}$$

At this point it is suggested that the reader consult Appendix section C.1, which explains how the EU SCM NMF algorithm applies the clustering procedure and source reconstruction procedure in order to classify and reconstruct output signals on the basis of common spatial similarity of audio components (i.e. NMF basis components). In the next section we consider a variant of EU SCM NMF known as DOA SCM NMF, which builds upon SCM model, multiplicative update rules, clustering, and source reconstruction in a manner that focuses on encoding frequency dependent direction of arrival matrices (DOA kernel matrices) into the quantity $\mathbf{H}_{fk}$, as described according to equations 5.19 and 5.22.

5.3.4 DOA (Direction of Arrival) Spatial Covariance NMF

What can be considered another state of the art algorithm that builds upon the EU-NMF SCM model of the previous section, is the DOA (Direction of Arrival) Spatial Covariance NMF algorithm proposed by Virtanen and Nikunen [24].At each time-frequency bin, the SCM provides a spatial parametrization of the sources by magnitude and phase differences between the recorded channels. The novelty of the algorithm lies in proposing that each SCM matrix be linked to a set of DoA kernels that geometrically sample the spatial space uniformly around a microphone array by a set of vectors $\{\mathbf{k}_o\}$ analogous to the vector $\mathbf{u}_s$ of equation 2.1.

The proposed CNMF algorithm is consequently able to provide estimates of the source parameters (magnitude spectra and DoA kernel direction weights) upon the clustering of the direction weights (component and look direction dependent) treated as feature vectors over which to be clustered.

The modelling of SCM matrices in terms of the spectral and spatial parameters per source and per time-frequency bin in the surrounding acoustic sound field can thus be summarized in terms of its merits concisely as follows:

- Magnitude spectra are modelled by the multiplication of two separate NMF factor matrices similarly as to in a single channel CNMF sense. One of which is a frequency dependent harmonic dictionary matrix and the other which is a time activation dictionary matrix. The motivation of such a model for the magnitude spectra is for finding spectrally redundant and/or a parts based representation of the source signals, which NMF and/or CNMF is typically known to be capable of providing. Column vectors and

row vectors in each respective matrix are coupled and uniquely specified according to their component index k, which for each pair of vectors, we visualize the pair of vectors as providing a low rank approximation of the variation of the sound across time and frequency over all STFT time frequency bins. We then imply that this parametrization fully corresponds to a spectral characterization of that sound, but we further require that the spatial information pertaining to the magnitude spectra of the sound component be linked to k and correspond to its actual spatial location within the acoustic sound field.

- In order to achieve this, a new NMF parameter, to be known as 'spatial direction weights', is to be introduced. It is to be defined as being dependent on look direction (associated with the far field model and time differences of arrival) and source component index k, over which clustering of the source component index k into target output class q can be achieved. It is proposed that we should derive and apply its CNMF update rule iteratively with respect to the spatial direction weights parameter and other CNMF parameters, and then to cluster over the spatial direction weights (within a post-processing step) to reveal the optimal linking that should be associated with sound components. In common with the magnitude spectra parameters, the spatial direction weights are also restricted to be nonnegative.

- Direction of arrival (DoA) kernel matrices are to be computed (in part statically and in part adaptively having to do respectively with their complex-valued phase and positive-valued magnitudes) and are to provide at each time frequency bin a complex hermitian positive semidefinite representation of spatial covariance matrices by implying that each SCM at each time frequency bin can be represented as a linear combination of the 'significant' DoA kernel matrices (look direction and frequency dependent). For example in a particular time frequency bin, the significant look directions for that time frequency bin, would be learned on the basis of a combination of the CNMF parameters having converged in that time-frequency bin, and on the basis of having clustered that time-frequency bin in the context of all other time-frequency bins, and thus, in a global sense. In fact, this implies that spatially related and observed multichannel STFT coefficients that are perhaps located at non-adjacent time frequency bins can be linked to each other and jointly optimized according to the CNMF algorithm and its corresponding clustering recipe, in order to provide a parametrization which provides an accurate, plausible, and separable representation of the sound mixture in the multichannel STFT, and complex SCM domains.

- We populate DoA kernel matrices $\mathbf{W}_{fo}$ on the basis of a finite set of user configurable

spatial vectors $\{\mathbf{k}_o\}$ that are linked to $\mathbf{W}_{fo}$ and encode look direction (angle of arrival) information on the basis of the indexing variable o. Furthermore, the vectors $\mathbf{k}_o$ should originate from the array center as illustrated within Figure 5.2.

- The appropriate weighting of DoA kernel matrices according to nonnegative magnitude spectra and spatial direction weights consequently results in a parametrized source component dependent interchannel mixing matrix that corresponds to the one described in equation 5.19.

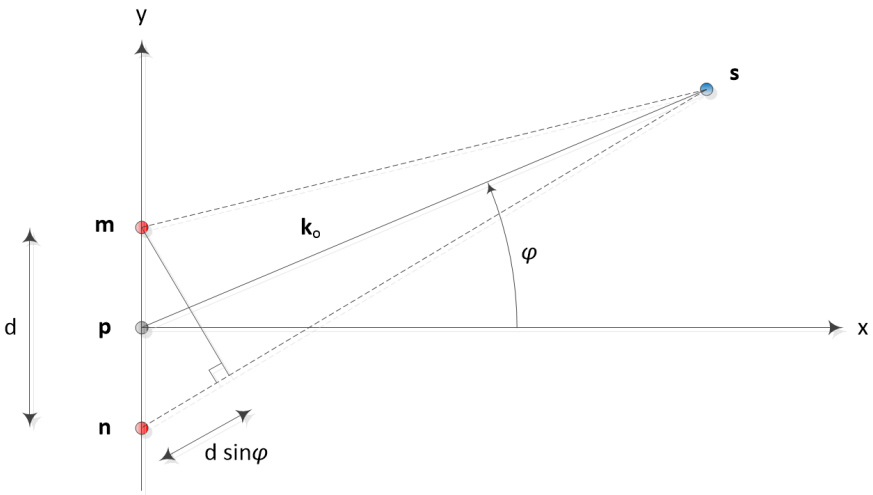

Figure 5.2: Array geometry illustration consisting of microphones m and n as seen from bird's eye view, azimuth angle represented by ϕ

We will now consider the algorithm more in detail. First, we connect the far field model of sound propagation covered in section 2.1.1.1 to illustrate the value of analyzing frequency domain phase shifts as a function of arbitrary "look directions".

DoA kernel matrices will be shown to be crucial in connecting time difference of arrivals which obviously characterize spatial propagation and interchannel characteristics of estimated sound components that should be grouped to a common source, selected from a set of possible sources assumed to be positioned (in a stationary fashion) in the far field, distant from the microphone array. DoA kernel matrices can be populated as follows:

$$[\mathbf{W}_{fo}] = \exp(j2\pi(\mathrm{f})\tau_{nm}(\mathbf{k}_o)), \quad \mathrm{f} = \frac{(f-1)F_s}{N_{STFT}} \tag{5.32}$$

Where:

1. f is to be considered as a frequency in Hz, and f denotes the discrete frequency variable.

2. N_{STFT} is the fft length of the STFT.

3. F_s is the sampling frequency, taken in simulations to be 16000Hz.

4. the quantity $2\pi f \tau_{nm}(\mathbf{k}_o)$ is an interchannel time difference that indexes the off-diagonal elements of the matrix $\mathbf{W}_{fo}$ as described in [24].

In this CNMF spatial covariance variant, the spatial covariance mixing matrix is modelled as a sum of all look direction kernel matrices $\mathbf{W}_{fo}$, weighted by spatial weights z_{ko}.

The sum of product form is also utilized again for the purpose of representing the matrix $\mathbf{H}_{fk}$ as given by

$$\mathbf{H}_{fk} = \sum_{o}^{O} \mathbf{W}_{fo} z_{ko} \tag{5.33}$$

but in this case a set of frequency-dependent and look-direction dependent DoA kernels $\mathbf{W}_{fo}$ are introduced into the spatial parametrization as well as a set of nonnegative spatial weights z_{ko} that are look direction and component-bin dependent.

The covariance matrix per time frequency bin is then modelled as dependent on the matrix $\mathbf{H}_{fk}$ scaled by magnitude spectra as given by

$$\begin{aligned} \hat{\mathbf{X}}_{fn} &= \sum_{k=1}^{K} \mathbf{H}_{fk} t_{fk} v_{kn} \\ &= \sum_{k}^{K} \sum_{o}^{O} \mathbf{W}_{fo} z_{ko} t_{fk} v_{kn} \end{aligned} \tag{5.34}$$

where t_{fk} and v_{kn} model the time frequency characteristics of NMF basis components, of which we consider each time-frequency bin to be the result of up to a maximum of K additive audio components with different possible spatial features as described according to equation 5.34. The likelihood function is:

$$\begin{aligned} P(\mathbf{X}|\mathbf{T},\mathbf{V},\mathbf{Z},\mathbf{W}) &= \prod_{f=1}^{F} \prod_{n=1}^{N} \prod_{m=1}^{M} \prod_{n=1}^{M} \mathcal{N}_c\big([\mathbf{X}_{f,n}]_{mn} \big| [\hat{\mathbf{X}}_{f,n}]_{mn}, 1\big) \\ &\propto \exp(-||\mathbf{X}_{fn} - \hat{\mathbf{X}}_{fn}||_F^2) \end{aligned} \tag{5.35}$$

Minimize the negative log-likelihood function: $\mathcal{L}(\theta)$ with $\theta = \{\mathbf{W}, \mathbf{Z}, \mathbf{T}, \mathbf{V}\}$

$$\begin{aligned}\mathcal{L}(\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V}) &= \sum_{f=1}^{F}\sum_{n=1}^{N}||\mathbf{X}_{fn}-\hat{\mathbf{X}}_{fn}||_F^2 \\ &= \sum_{f=1}^{F}\sum_{n=1}^{N}||\mathbf{X}_{fn}-\sum_{k}^{K}\sum_{o}^{O}\mathbf{W}_{fo}z_{ko}t_{fk}v_{kn}||_F^2 \end{aligned} \tag{5.36}$$

Where $\mathbf{E}_{fn} = \mathbf{X}_{fn} - \sum_{k}^{K}\sum_{o}^{O}\mathbf{W}_{fo}z_{ko}t_{fk}v_{kn}$ and $\hat{x}_{fn} = \sum_{k,o} z_{ko}t_{fk}v_{kn}$

Where, since we assume direct optimization of 5.36 to be difficult, the majorizing function $\mathcal{L}^+$ is utilized:

$$\mathcal{L}^+(\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V}) = \sum_{f=1}^{F}\sum_{n=1}^{N}\sum_{k}^{K}\sum_{o}^{O}||\mathbf{C}_{fnko}-\mathbf{W}_{fo}z_{ko}t_{fk}v_{kn}||_F^2 \tag{5.37}$$

In a MM or EM fashion, the majorizing function $\mathcal{L}^+(\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V},\mathbf{C})$ is introduced, for which the two crucial properties can be verified:

1. $\mathcal{L}(\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V}) \leq \mathcal{L}^+(\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V},\mathbf{C})$
2. $\mathcal{L}(\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V}) = \min_{\mathbf{C}} \mathcal{L}^+(\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V},\mathbf{C})$

Where the first condition signifies that the objective function $\mathcal{L}$ can be verified to upper bounded by the majorizing function $\mathcal{L}^+$ for any choice of the parameter set $\theta = \{\mathbf{W},\mathbf{Z},\mathbf{T},\mathbf{V}\}$. The second condition further signifies that a choice can be made for the auxiliary matrix variable $\mathbf{C}$ that achieves the necessary tangency condition for the majorizing function, if chosen as follows:

$$\mathbf{C}_{fnko} = \mathbf{W}_{fo}z_{ko}t_{fk}v_{kn} + r_{fnko}\Big(\hat{\mathbf{X}}_{fn} - \sum_{k,o}\mathbf{W}_{fo}z_{ko}t_{fk}v_{kn}\Big) \tag{5.38}$$

In a similar fashion as in [63] and [25], the latent components can be related to the observed spatial covariance matrix $\mathbf{X}_{fn}$ in the f-nth time-frequency STFT bin. We require then another set of conditions to be true:

1. $\sum_{k,o}\mathbf{C}_{fnko} = \mathbf{X}_{fn}$
2. $r_{fnko} = \dfrac{z_{ko}t_{fk}v_{kn}}{\hat{x}_{fn}}$
3. $\hat{x}_{fn} = \sum_{ko} z_{ko}t_{fk}v_{kn}$

4. $\sum_{k,o} r_{fnko} = 1, r_{fnko} > 0$

Now, comparing 5.36 to 5.36, utilizing weights r_{fnko} in combination with Jensen's inequality, we can move the inner double summation to the outside of the convex Frobenius norm operator. With all properties of the majorizing function $\mathcal{L}^+$ now well defined, the update rules for the algorithm can be derived by minimizing it with respect to the parameter set $\theta = \{\mathbf{W}, \mathbf{Z}, \mathbf{T}, \mathbf{V}\}$, and are presented in summary in the subsequent section.

5.3.5 Update Rules

By optimizing $\mathcal{L}^+$ with respect to the CNMF parameter set, we obtain multiplicative update rules for the CNMF parameters set, as given by

$$z_{ko} \leftarrow z_{ko}\left[1 + \frac{\sum_{f,n} t_{fk} v_{kn} \operatorname{tr}(\mathbf{E}_{fn}\mathbf{W}_{fo})}{\sum_{f,n} t_{fk} v_{kn} \hat{x}_{fn}}\right] \tag{5.39}$$

$$t_{fk} \leftarrow t_{fk}\left[1 + \frac{\sum_{n,o} z_{ko} v_{kn} \operatorname{tr}(\mathbf{E}_{fn}\mathbf{W}_{fo})}{\sum_{n,o} z_{ko} v_{kn} \hat{x}_{fn}}\right] \tag{5.40}$$

$$v_{kn} \leftarrow v_{kn}\left[1 + \frac{\sum_{f,o} z_{ko} t_{fk} \operatorname{tr}(\mathbf{E}_{fn}\mathbf{W}_{fo})}{\sum_{f,o} z_{ko} t_{fk} \hat{x}_{fn}}\right] \tag{5.41}$$

$$\mathbf{W}_{fo} \leftarrow \mathbf{W}_{fo} \,.^{*}\left[\sum_{n,k} z_{ko} t_{fk} v_{kn} \hat{x}_{fn} + \sum_{n,k} z_{ko} t_{fk} v_{kn} \mathbf{E}_{fn}\right]. \tag{5.42}$$

According to [24] there exists a specific and suggested way to update the CNMF parameter $\mathbf{W}_{fo}$ according to equation 5.42 that involves its eigenvalue decomposition. Essentially it is suggested to update only its nonnegative magnitudes, element-wise, and to persist its element-wise phase information, which encodes information about DOA kernels.

At this point it is suggested that the reader consult Appendix section C.2, which explains the clustering procedure and source reconstruction procedure of the DOA SCM NMF algorithm. In that section, the K-means post processing procedure suggested by [24] is described explicitly. A K-means procedure applied to the spatial weights parameter z_{ko} is detailed, which outputs the indicator variable b_{qk} that is of key importance to the source reconstruction procedure of the DOA SCM NMF algorithm.

Also, at Appendix section D.2, we have provided the result of simulating an implementation of the DOA SCM NMF algorithm for the musical signals described in section 1.5. The configuration of the source separation to be applied as input to the algorithm is undetermined with three sources (guitar, piano, violin) and two microphones (a symmetric

two-microphone linear microphone array). After introducing the thesis proposed algorithm, we intend to demonstrate how the thesis's proposed algorithm performs against the DOA SCM NMF algorithm, in terms of performance, under the same source separation scenario.

5.4 Other Approaches, Algorithms and Topics pertaining to Multichannel Source Separation

The algorithms presented thus far have utilized NMF based auxiliary functions for proceeding to develop optimization frameworks that allow derivation of multiplicative update rules to be obtained in order for each respective algorithm to learn a meaningful parametrization and in order to obtain a parts based representation of audio components in the time frequency domain. The merits of the algorithms considered in this chapter include:

- Being able to associate spatial directivity of sound components as seen and observed by the microphone array.
- Being able to obtain spatially unclassified vectors (features) in some of the NMF algorithm parameters, that provide a matrix data space of vectors over which two cluster, that is to group, classify, or assign labels to subsets of vectors that should be linked on the basis of similar spatial (i.e. source) properties. We emphasize that the linking of these features, within the thesis, is to be generally termed as associating or assigning a *class* label, to any of the unclassified features, which were learned by applying a multiplicative update rule to a particular NMF parameter (for instance a spatial weights matrix parameter, or a cluster indicator matrix parameter).
- The parametrization according to matrices provides meaningful interpretations of spectral and spatial audio features that can be manipulated in the TF domain.
- Being able to obtain a two matrix NMF factor representation of magnitude spectra of source components, by extending the basic two factor NMF model, that is compatible with the spatial parametrization merits of the algorithms as previously described.
- By iterating NMF according to multiplicative update rules, randomly initialized matrix parameters are forcibly caused to eventually converge towards plausible representations of learned audio features and/or learned audio components.
- The spatial directivity of a multiple microphone array is possible by exploiting the far field model of sound propagation. For a single channel CNMF model, a significant issue

of importance to the CNMF parametrization of the single channel STFT spectrogram is with regards to the *phase dictionary* of audio components. Thus far the concept of including a *phase dictionary* as part of a CNMF optimization framework has only been explored in a single channel CNMF context to the best of our knowledge.

- The modelling of *interchannel phases* has however, been considered as part of the CNMF optimization framework as is evidenced by considering the SCM matrix per time frequency bin, and its off-diagonal elements, that occur due to SCM NMF based processing. It was noted that, in a small microphone array, interchannel phases differences described by considering the SCM model are typically more visible than gain differences for each microphone [16], as described by how converged SCM matrices are able to approximate processed SCM matrices, per time frequency bin.

5.5 Key Conclusions for the Chapter

In this chapter the objective was to unite the concepts covered in the chapters leading up to it, with the objective of applying all of the considerations necessary to achieve source separation and source reconstruction in the context of a scenario where the observed data is processed as closely to a complex STFT representation as possible. Thus, it should be very clear at this point that treating the observed data as complex STFT coefficients as opposed to nonnegative STFT coefficents, has been demonstrated to be more challenging (but perhaps also more rewarding) than the methods illustrated in section 4.1.3, for instance.

First, we began by considering the ideas presented in [25], which demonstrated that a CNMF problem formulation could be derived and applied to the original NMF of Lee and Seung [1] covered in section 4.1, on the basis that STFT spectra for analyzing audio signals are inherently complex-valued (nonnegativity, in previous research was taken as a simplification of the analysis). A set of update equations were derived on the basis of this new CNMF formulation, and aside from modeling the nonnegative factor matrices according to equations 5.11 and 5.12, for the purpose modeling the spectral amplitudes of sound components across all time-frequency-component bins, the CNMF model derived a novel phase update rule as given by 5.13 (which resulted in the appropriate naming of the algorithm as a CNMF algorithm). In conclusion, [25] also showed as a final point that the novel CNMF algorithm proved to be convergently as stable as was the case in the original NMF by Lee and Seung. It furthermore provided a special condition under which the update rules for 5.11 and 5.12 could be reduced to be exactly equivalent to the Euclidean distance update rules derived by Lee and Seung A.6.1.

A possible interpretation of the connection of [25] to the proposed algorithm would simply be that the proposed algorithm represents an extension of the original CNMF algorithm but having to do with a spatially diverse and multichannel configuration of the observed STFT, since the algorithm by Kameoka et al. assumes the observed data to be only a single channel of observed STFT coefficients. One possible issue with this interpretation however is that it does not by itself adequately demonstrate the important perspective of considering the spatial covariance properties (introduced in section 5.3) associated with multichannel spatial covariance matrix (SCM) NMF models [16, 24].

The usage of the EU SCM NMF algorithm [16] was shown to be such that the update rules (section 5.3.3) are first meant to be applied iteratively to a point of stablility (i.e. satisfactory convergence, or when the matrix parameters can be shown to no longer change significantly between iterations). Assuming all of the model parameters, according to the clusterable sum of product (equation 5.21) have converged and source estimates have been approprately clustered, the model and then utilizes the parameter z_{lk} as features over which to cluster the output signals audio components that have similar spatial properties. Output STFT's per output class are reconstructed on the basis of the optimal clustering configuration and reconstruction step specified in section C.2.2.

Building upon the EU SCM NMF algorithm and following a similar overall source separation procedure, another important SCM NMF algorithm was introduced, which was the one specified in [24] by Virtanen and Nikunen, the DOA SCM NMF algorithm. While still being based upon a sum of product SCM parametrization of the observed multichannel covariance matrix, according to equation 5.34, the novel concept introduced within this algorithm was to define the notion of direction of arrival (DoA) kernel matrices, which encode pre-calculated frequency dependent and look direction dependent information into the parameter $\mathbf{W}_{fo}$, defined according to equation 5.32 and to which the authors of [24] propose the update rule should only be applied in order to update the nonnegative magnitude of the DOA kernel matrix, but the phase (pre-computed on the basis of discrete STFT frequencies and discrete look directions) of the Hermitian positive definite matrix, should be persisted throughout applying the algorithm. In order to learn the optimal configuration a set of spatial weights, z_{ko} were introduced into the model. In terms of clustering, analogously to the EU SCM NMF algorithm, a K-means approach could be applied following allowing the NMF multiplicative update rules to converge towards an appropriate solution. Details regarding the clustering and source reconstruction for the algorithm were provided in appendix section C.2. In the next chapter, we will consider the proposed algorithm, which can be viewed both an extension of the single channel CNMF algorithm and the DOA SCM NMF algorithm.

Chapter 6

Proposed Algorithm

The end goal of the proposed algorithm is again to obtain multiplicative update rules and to determine if in some configuration the proposed algorithm can be shown to exceed the performance of the state of the art DOA SCM NMF algorithm, detailed in the previous chapter. The algorithm parameters, are chosen differently than in the DOA SCM NMF algorithm however, and are even chosen differently than according to the sum of product SCM NMF general formulation, described according to equation 5.18. Processing and pre-encoding of microphone and frequency dependent phase difference information (with respect to the center of a microphone array containing densely space microphones) will still be a priority as according to equation 5.32, and is again essentially based upon the far field model as detailed in section 2.1.1.1. It will be shown however, that it may be appropriate to consider directly modelling the multichannel vector quantity $\mathbf{h}_{fk}$ as it occurs in 5.19 as opposed to the quantity $\mathbf{H}_{fk}$ as is typically done in SCM NMF based algorithms.

As in both cases, for the EU SCM NMF and DOA SCM NMF, the clustering procedure of each respective algorithm is dependent on how the parametrization of $\mathbf{H}_{fk}$ occurs and how the multiplicative update rule cause its associate parameters to parametrize the class to component correspondence. In the proposed algorithm, the clustering procedure depends correspondingly on $\mathbf{h}_{fk}$ and a similar principle as well. However, it will also be a newly proposed concept to attempt to propose how the clustering may directly incorporate and associate class and time activation corresponding within the representation for the parameter $\mathbf{h}_{fk}$. That is, $\mathbf{h}_{fk}$ in the proposed algorithm will be linked simultaneously to class dependent and time activation dependent information. Furthermore, $\mathbf{h}_{fk}$ will still be meant to encode spatial information and information pertaining to DOAs (look directions) and assign an optimal clustering between classes and components as in the DOA SCM NMF reference algorithm. How the proposed algorithm achieves this in practice is based upon introducing various matrix and/or tensor (multi-way tensor) parameters that are in some cases not present in the SCM NMF algorithms of the previous chapter. In the next section we provide

an introduction to the proposed algorithm by revisiting the scenario presented in section 1.4.

6.1 Intro to Problem Formulation and Illustration of Model

Figure 6.1 details a high level explanation of the proposed CNMF algorithm.

We take the opportunity at this point to explain the generation of the observed microphone signals from filtering of source signals by impulse response filters corresponding to acoustic transfer functions obtained from room simulation software, where the *acoustic paths* between each pair of microphone and source was chosen in some configuration geometrically, and then specified exactly to the room simulation software, MCRoomSim, in order to simulate spatial filtering of source signals subsequently providing us with a representation of their resulting spatial images.

The spatial images are represented as signals with the following labelling convention, as similarly illustrated in the diagram, as given by

$$\begin{aligned} y_{ml}[n] &= s_l[n] * h_{ml}[n] \qquad \text{for } m = 1, ..., M,\ l = 1, ..., L \\ &= \sum_{p=-\infty}^{\infty} s_l[p] h_{ml}[n-p] \end{aligned} \tag{6.1}$$

where $*$ is the standard discrete time *convolution* operator, and the impulse responses $h_{ml}[n]$ used to filter the lth source signal $s_l[n]$ were a finite duration in terms of a certain number of samples. The dummy variable p here is used for computing the convolution sum (inner product) at each output sample n of the output *spatial image* signal $y_{ml}[n]$.

In order to generate the mth observed microphone signal, each set of spatial images per microphone, were summed, thus, additively overlapping the spatial images in both time and frequency, and thus creating the multi-signal (but spatially diverse) mixture for the source separation algorithm to be tested against.

$$x_m[n] = \sum_{l=1}^{L} y_{ml}[n] \qquad \text{for } m = 1, ..., M \tag{6.2}$$

For test case 1, since the three source signals chosen were set up as musical source signals, $L = 3$ was chosen to be the total number of sources.

Since MCRoomSim was configured with no room reverberation properties (anechoic conditions), the *tail end* of impulse responses $h_{ml}[n]$ was noted as negligible as compared to

a simulation that could have been intended to test the reverberation modelling of a source separation algorithm more intensively.

The output of equation 6.2 per microphone index m corresponds to the observed set of signals $x_m[n]$. The right hand side of equation 6.2 corresponds to information that is in no way specified to the proposed algorithm, and thus can be interpreted to have been effectively *thrown away*. Thus the proposed algorithm should proceed to parametrize source estimates and must do so in a way that is both *blind* and *underdetermined*, where the number of available microphones is *less* that the number of target source signals that should be provided at the algorithm output.

For testing the proposed algorithm, we have thus far only experimented with $M = 2$ as the total number of observed microphone signals, and $L = 3$ as the total number of source signals to be recovered.

Therefore, the current chapter intends to thoroughly detail the technical implementation details of the illustration provided by figure 6.1. The end goal of the proposed algorithm can therefore clearly be stated as requiring that the algorithm provide an estimated and parametrized set of source signals

$$\hat{s}_1[n], \hat{s}_2[n], ..., \hat{s}_L[n] \tag{6.3}$$

that closely approximate the effect of the true but *unobserved* source signals

$$s_1[n], s_2[n], ..., s_L[n] \tag{6.4}$$

based only upon available knowledge of the observed microphone signals

$$x_1[n], ..., x_M[n] \tag{6.5}$$

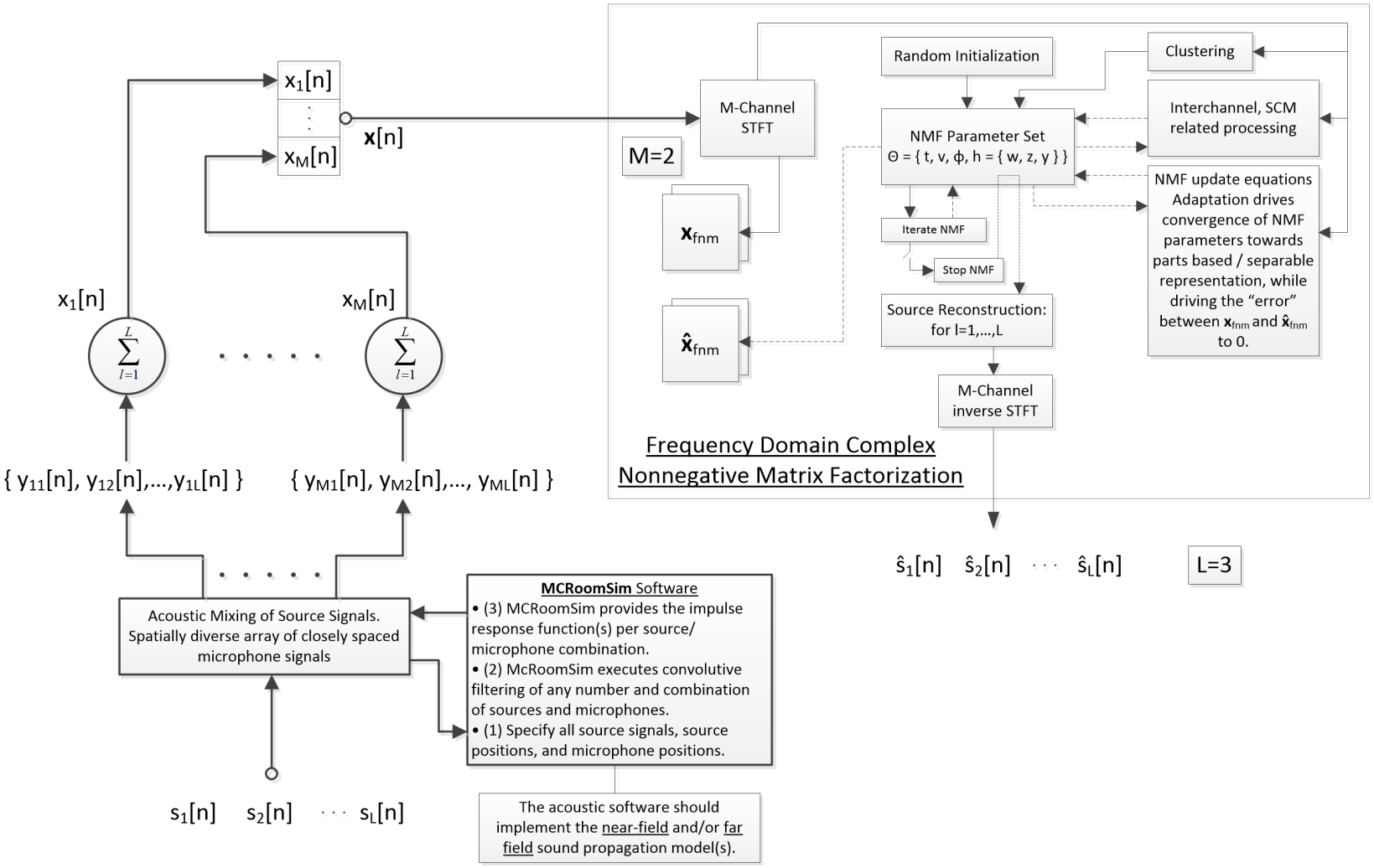

Figure 6.1: Proposed CNMF Algorithm High level Explanation

6.1.0.1 MCRoomSim

In the following section we describe how the observed microphone signals were generated using the freely available room simulation software MCRoomSim [53].

As mentioned in [24], an important novelty and characteristic of the SCM CNMF algorithm proposed in that paper is to define a set of look direction vectors $\mathbf{k}_o$ that spatially sample the surface of a unit sphere around the geometrical center of the array.

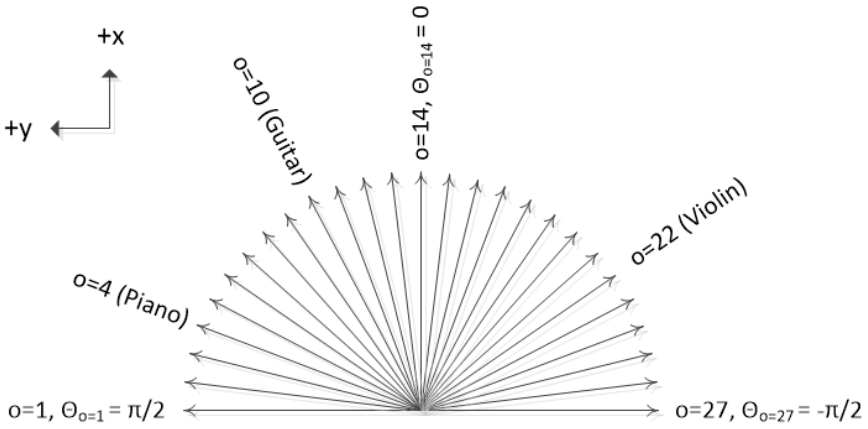

Figure 6.2: Look Directions surrounding microphone array center for musical source signals

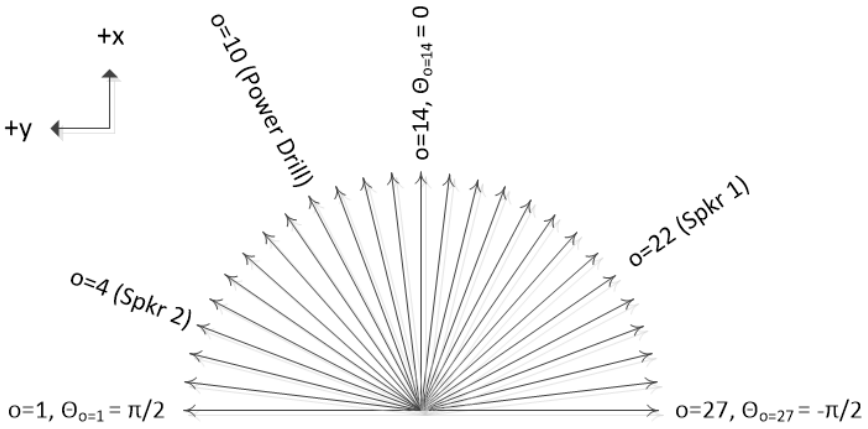

Figure 6.3: Look Directions surrounding microphone array center for speech source signals

In total three test cases were constructed. Each test case assumed a three source, two microphone mixture scenario. For the primary test case the source signals corresponded to

musical instruments, and their positions relative to the microphone array center is illustrated in figure 6.2. For the secondary test cases corresponding to speech the source signals and their positions relative to the microphone array center is illustrated in figure 6.3.

Again we emphasize that it is the primary test case with musical signals that we will focus on in this chapter, but a similar procedure was certainly used to generate the microphone mixtures corresponding to the secondary test cases. The choice of source positions were restricted to lie within a half circle, spatially surrounding a two microphone array, spatially sampled with 27 total look directions surrounding the microphone array. The choice of using a half circle as opposed to a full circle was in order to avoid ambiguity regarding "mirror" look directions associated with the limitation of a stereo microphone array only being able to uniquely identify time differences of arrival, for a half circle configuration. Simply put, had we chose a three microphone array, then the ability to uniquely distinguish look directions would have been increased.

The distance between the left and right microphones of the microphone array was configured to be 10cm.

The following equations were used to specify the exact source positions in a Cartesian coordinate system

$$\text{position(Piano)} = A_c + [R_P \cos(\theta_P), R_P \sin(\theta_P), 1] \tag{6.6}$$

$$\text{position(Guitar)} = A_c + [R_G \cos(\theta_G), R_G \sin(\theta_G), 1] \tag{6.7}$$

$$\text{position(Violin)} = A_c + [R_V \cos(\theta_G), R_V \sin(\theta_V), 1] \tag{6.8}$$

where the variables R and θ were used to specify an active and stationary source at a particular look direction, whose approximate position and DoA relative to the array center A_c as specified by the far field model was calculated dependent on the choice of the two variables R and θ.

Table 6.1: Table Summarizing Calculation of Source Positions

Source Signal	Look Direction (o)	θ_o (radians)	Radius (R)	Position (Cartesian Coordinates)
Piano	4	1.2083	1.8	$[2.7683, 3.1330, 1]$
Guitar	10	0.4833	1.5	$[3.4582, 2.1471, 1]$
Violin	22	-1.0875	1.2	$[2.6877, 0.3875, 1]$

Thus, the array center was specified and configured by the vector A_c as given by

$$A_c = [2.13, 1.45, 1.00] \tag{6.9}$$

Where each of the distances is in meters and therefore a distance of 0.1 meters corresponds to the microphone spacing of 10cm.

The left and right microphone were labelled as $m = 1$ and $m = 2$ respectively, and were positioned equidistant from the array center specified by A_c at positions $[2.13, 1.50, 1.00]$ and $[2.13, 1.40, 1.00]$, respectively.

Having populated table 6.1 in the suggested manner, we then specified for MCRoomSim to apply the appropriate spatial filtering based upon the specified *source positions* as well as *microphone positions*. We propose that the Cartesian coordinates specified in the fifth column of the table, also correspond to the configuration illustrated pictorially in Figure 6.2. A direct conversion between a particular look direction o and its corresponding *azimuth* angle (in radians) referenced to the position $\theta_{o=14} = 0$ can be computed according to

$$\begin{aligned} \theta_o &= \frac{\pi}{2} - \frac{\pi}{O-1}(o-1) \\ \theta_o &= \frac{\pi}{2} - \frac{\pi}{27-1}(o-1) \end{aligned} \tag{6.10}$$

which for test case 1 specifies a half circle equidistantly sampled by 27 total look direction vectors in terms of the azimuthal angle θ_o. If $z = 1$ were not chosen to be fixed, we could vary z to geometrically sample a unit sphere as was described in [24]. Within table 6.1 and according to equations 6.6 to 6.8 the values for the radius R per source signal were chosen arbitrarily such that $R > 1$ (since we need consider that the *far field* model will be implied towards justifying the separation of the source signals).

Following running the MCRoomSim software according to the values specified within table 6.1, the *impulse response sequence* vectors that were returned by MCRoomSim were therefore used to *spatially filter* via convolution the spatial image signals to be combined additively per microphone as described by equations 6.1 and 6.2 in the previous section.

The interested reader can be pointed to the appendix in section D.4 in order to observe the impulse response sequences returned by MCRoomSim for each *source* and *microphone* combination for the 3 sources and 2-microphone array.

6.1.0.2 Illustration of modelling observed multichannel STFT vector

The following section intends to provide an explanation of an STFT coefficient mixture scenario at an arbitrary time frequency bin of an arbitrary number of spatial images occurring from an arbitrary number of look directions.

As the title more precisely but generally suggests, we focus on the modelling of the observed multichannel STFT vector in terms of a source vector $\mathbf{s}_{fn}$ that causes due to each of its elements an unobserved additive combination of spatial image vectors of size $M \times 1$ to be seen in some configuration within the observed multichannel STFT vector $\mathbf{x}_{fn}$. We will now consider an appropriate model capable of modelling this effect.

We shall attempt to emphasize that the key to inferring the correct parametrization of the spatial images, is similar to approaches taken in previous chapters and especially within the most recent chapter of the thesis in which multichannel methods were considered.

As per usual we will assume that the observed data occurred within a multichannel STFT bin in which the number of microphones is two, that is $M = 2$ the number of sources to be extracted is *at most* three, that is $L = 3$. Therefore, a *transformation of the vector* $\mathbf{s}_{fn}$ that is applied to generate an observed STFT vector $\mathbf{x}_{fn}$ at an arbitrary time frequency bin can be described as given by

$$\mathbf{x}_{fn} = \mathbf{H}_f \mathbf{s}_{fn} \tag{6.11}$$

where $\mathbf{x}_{fn} \in \mathbb{C}^{M \times 1}$ is the observed STFT vector, $\mathbf{H}_f \in \mathbb{C}^{M \times L}$ is the spatial filtering matrix, and $\mathbf{s}_{fn} \in \mathbb{C}^{L \times 1}$ is the vector of true source STFT coefficients at the arbitrary time frequency bin.

We will begin by first considering a few things that we would like to emphasize about equation 6.11. Also we will intend to augment equation 6.11 differently as we proceed further.

The first is that we to consider the case where we sub-index the matrix $\mathbf{H}_f$ according to its class index l which represents some particular *output class* that we are interested in.

Let us assume with regards to the current time frequency bin that $\mathbf{s}_{fn}$ represents the effect of the *true* output source components, but in particular that of a *dominant* class l such that $\mathbf{s}_{fn}$ represents an active source component at only *one* particular source bin and the contribution from the other true sources is somehow known to be zero.

This could be then in fact represented as corresponding to

$$\mathbf{s}_{fn} = \begin{bmatrix} 0 \\ 0 \\ 2e^{j\phi_3} \end{bmatrix} \in \mathbb{C}^{L\times 1} \tag{6.12}$$

where the amplitude and phase of the source coefficient at $l = 3$ are 2 and $e^{j\phi_3}$, respectively, and are chosen here arbitrarily. Therefore, we propose that somehow (for illustration purposes) we have resulted upon the true source coefficient, $2e^{j\phi_3}$, but we shall intend to treat it as an *estimated* or *parametrized* source coefficient where it could be such it represents the result of the output of a reconstruction step and we would like to verify its *plausibility* given the *observed* STFT vector in this particular time-frequency bin, $\mathbf{x}_{fn}$.

Since the source coefficients for $l = 1$ and $l = 2$ are zero, the explanation is in fact very simplified and can be summarized as given by

$$\mathbf{x}_{fn} = [\mathbf{H}_f]_{l=3} \begin{bmatrix} 0 \\ 0 \\ 2e^{j\phi_3} \end{bmatrix} \tag{6.13}$$

where we intentionally sub-index the matrix $\mathbf{H}_f$ at $l = 3$ to extract the lth column vector only. Here we refer to the term

$$2e^{j\phi_3}[\mathbf{H}_f]_{l=3} \in \mathbb{C}^{M\times 1} \tag{6.14}$$

as the *spatial image* of source $l = 3$ at time frequency bin f-n, where the spatial images due to the contributions of sources $l = 1$ and $l = 2$ are conveniently zero within this particular illustration. Thus, in general, $\mathbf{x}_{fn}$ occurs due to the superposition (i.e. contribution) of all spatial images summed across $l = 1, ..., L$ which is fully generalized according to equation 6.11 for at most $L = 3$ sources but is conveniently illustrated according to the the scenario described by equations 6.13 and 6.14. We note that the spatial image should also be interpreted as a *multichannel* vector since it has exactly the same size as the vector $\mathbf{x}_{fn}$. Thus, the multichannel source separation problem in the STFT coefficient domain can be appropriately viewed as equivalent to fully specifying and *labelling* the occurrence of spatial images due to a certain number of sources L across all time frequency bins f-n, given M observations, per time-frequency bin.

Further describing equation 6.13 we have that it is only the vector $[\mathbf{H}_f]_{l=3} \in \mathbb{C}^{M\times 1}$ sub-indexed at $l = 3$ carries the information of the source vector $\mathbf{s}_{fn}$ sub-indexed at $l = 3$ into the output vector $\mathbf{x}_{fn}$, in the illustrated scenario. Therefore, whether or not in fact the proposed representation of $\mathbf{s}_{fn}$ according to equation 6.13 is plausible or not, depends on in fact what the representation for $[\mathbf{H}_f]_{l=3}$ should be, which we also propose might be dependent on the *spatial configuration* that specifies the spatial filtering parametrization that should be

described between source $l = 3$ and microphones $m = 1$ and $m = 2$.

Therefore, we thus take the opportunity within the current illustration to emphasize the concept that we intend for the vector $[\mathbf{H}_f]_{l=3} \in \mathbb{C}^{M\times 1}$ to apply a *microphone-dependent* phase shift of the source STFT coefficient at $l = 3$ which corresponds to the third element of the vector $\mathbf{s}_{fn}$. Thus here, the observed output vector $\mathbf{x}_{fn}$ has occurred solely due to the contribution of the output coefficient at source (class) index $l = 3$ and not due to the classes at $l = 1$ and $l = 2$. Therefore, the information contained within $\mathbf{H}_f$ sub-indexed at $l = 1$ and $l = 2$ is somewhat irrelevant in this case, for explaining $\mathbf{x}_{fn}$ and the overall problem is fairly easy, since this scenario analogously explains a scenario corresponding to *non-overlapping* spatial spectra of source STFT coefficients. In other words, it is only the spatial image at $l = 3$ that dominates the current time frequency bin and this is conveyed by considering the interpretations of equations 6.13 and 6.14.

The discussion to this point has been fairly simple but is to some degree adequate, as a starting point for understanding the proposed algorithm, since for a $M = 2$ and $L = 3$ mixing system it actually describes what could be observed in a significant number of multichannel STFT time frequency bins, in the converged result of the algorithm, when the test case is set up such that a high degree of *non-overlap* (e.g. temporal overlap) has been configured between the true source signals.

It also emphasizes that since *globally* the algorithm may do a significant amount of computations in terms of multiplications, additions, and sub-indexing across one of many possible bins to be sub-indexed, upon reconstructing the algorithm we effectively require that the estimated and parametrized signals have been reconstructed *locally* (i.e. at a particular time-frequency bin) in such a way that the parametrization corresponds to the described sort of spatial filtering between any source l and its spatially filtered spectra as seen by the microphone for $m = 1, .., M$. We propose that it is by considering evidence *local* evidence that it can determined whether or not the algorithm is doing something *globally* that is in fact useful or not.

Before moving on, a further point to mention about the scenario described by equation 6.13 is that it allows for the consideration of what happens in principle when the parametrization provided by the right hand side is transformed into the SCM matrix domain. If we consider instead the matrix product $\mathbf{x}_{fn}\mathbf{x}_{fn}^H$ then we propose that $[\mathbf{H}_f]_{l=3}$ becomes a spatial interchannel covariance matrix like quantity $[\mathbf{H}_f]_{l=3}[\mathbf{H}_f]_{l=3}^H$, where we now consider $M \times M$ Hermitian symmetric matrix per time frequency bin, and per source l instead of an $M \times 1$ complex vector. We can first emphasize that the concept of having a priori knowledge that the classes corresponding to $l = 1$ and $l = 2$ having negligible contribution from the respective sources could be applied in a similar way, but more precisely it correspond to a SCM

based processing context as opposed to a multichannel STFT processing one. context. We also point out that the contribution of the scaling that occurs due to the source coefficient $2e^{j\phi_3}$ would become focused a matter of estimating the variance, that is $2^2 = 4$ as opposed to estimating the amplitude and the initial phase, that is 2 and $e^{j\phi_3}$, respectively. This is due to the fact that the reconstruction of output sources requires only the variance and subsequently infers the phase from the observed STFT vector $\mathbf{x}_{fn}$ within a multichannel Wiener filtering-like reconstruction step. Therefore, we emphasize that in each time-frequency bin the source separation problem is transformed into considering the superposition of source covariance matrices whose spatial contribution to the observed (but processed) SCM quantity $\mathbf{x}_{fn}\mathbf{x}_{fn}^H$ in the illustrated scenario could be appropriately modelled as

$$(4)[\mathbf{H}_f]_{l=3}[\mathbf{H}_f]_{l=3}^H \in \mathbb{C}^{M\times M} \tag{6.15}$$

as opposed to the superposition of spatial image vectors whose contribution in the illustrated scenario was described as in equation 6.14.

Thus, the *appropriate distinction* between considering the parametrization of the observed data a *spatial covariance* representation, in contrast to a *spatial image* representation, is also very important to consider.

Regrettably, we will have to attempt to explain the proposed algorithm in a more global sense, and begin to expand the illustration provided at a certain time frequency bin as initially described within equation 6.11.

We then introduce Figure 6.4 which pictorially illustrates the notion of how the L spatial image vectors contribute to the M microphones to combine additively in terms of STFT coefficients.

According to this illustration we suggest that the spatial positions of sources are *parametrizable* by a set of unit vectors $\{\mathbf{k}_o\}$ where the unit vector is identified by the label o where each look direction unit vector $\mathbf{k}_o$ is a spatial vector that physically and geometrically samples the physical space surrounding a microphone array. The physical direction that $\mathbf{k}_o$ specifies can be characterized in terms of azimuth and elevation or can alternatively be considered in terms of a 3D coordinate system, as illustrated within [24, 28]. Therefore, the proposed illustration and proposed algorithm are based upon the far field model described in section 2.1.1.1 and is most similar to the DoA SCM NMF algorithm detailed in section 5.3.4.

We then appropriately define the look direction o as specifying per microphone m that is given by

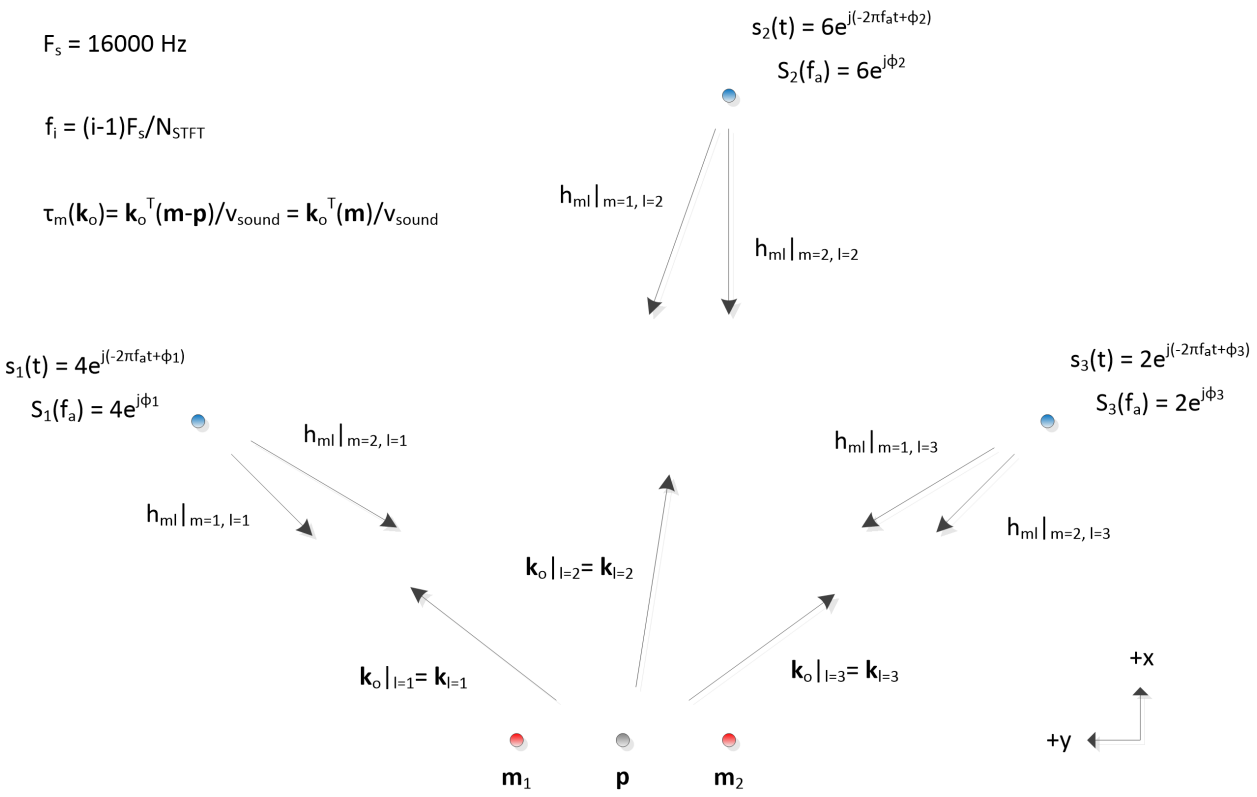

Figure 6.4: Multichannel Frequency Domain Filtering Illustration

$$\tau_m(\mathbf{k}_o) = \mathbf{k}_o^T(\mathbf{m}_m - \mathbf{p})/v_{sound} \tag{6.16}$$

In order to illustrate and specify a *clustered correspondence* between a particular look direction o and a parametrized class source component $S_l(f_a)$ we have in some cases pre-labelled the correspondence encoded within the time difference quantity and labelled it accordingly such that

$$\tau_m(\mathbf{k}_l) \leftarrow \tau_m(\mathbf{k}_o) \tag{6.17}$$

in some instances. As illustrated within Figure 6.2 the intention of the full set of look direction vectors $\{\mathbf{k}_o\}$ for $o = 1, ..., O$ to entirely parametrize the spatial region of interest surrounding the multi microphone array system. The intention of the algorithm however, in short, is to adaptive learn an appropriate spatial parametrization of the pre-defined surrounding area and to *indicate* within the set of look directions, which of them are the most prominent or dominant, considered across all time-frequency bins.

Within the modified illustration, and within the proposed algorithm, we focus on identifying the most *significant* source contributions which considered per microphone m, we

determined to be at most *three* if L is chosen such that $L = 3$. This is appropriately conveyed in Figure 6.4. This will be achieved on the basis of a set of nonnegative *indicator* values equivalently referred to as spatial weights that associated an optimal class (i.e. source signal) to look direction correspondence, and is similar to the notion of spatial weights first introduced within [24]. Given the parametrization of the spatial weights it will be possible to associate a zero (null) nonnegative value with certain look directions, thus in clustering and in iteratively obtaining an NMF parametrization, it will be a motivation to only persist the look directions that correspond to *true* sources, and to recover their STFT coefficient parametrization, at each time frequency bin, subject to the added stipulation of the spatial filtering of the sources that is to be modelled.

Therefore, what we imply within Figure 6.4 is that a desirable converged result between the look direction parametrization according to the set of vectors $\{\mathbf{k}_o\}$ for $o = 1, ..., O$ and the lth target source signal $S_l(f_a)$ can be obtained, that *indicates* from which look direction o did the lth labelled class component most likely occur.

Hence we propose that this can be achieved at any particular time frequency bin, and presumably we would like for the algorithm to be able to indicate this information *across all time frequency bins* of the multichannel STFT. This will be difficult to achieve in practice, however, we intend to show that it may in fact be both possible and meaningful to do so, thus leading to a plausible and acceptable source separation output of multichannel STFT mixtures where the source signals are music and speech, which is later to be demonstrated in the evaluation of the proposed algorithm. In the current chapter and for the test case labelled as test case 1, the source signals are constructed as musical notes. The observed multichannel STFT is thus constructed as the superposition of multiple notes from multiple instruments, played at non-overlapping time instances. Under this test condition, it will be more straightforward to test the *spatial directivity* of the microphone array and proposed CNMF algorithm, as is intended to be shown.

To further describe the scenario illustrated pictorially in Figure 6.4 we need only augment the original description provided by equation 6.11 as follows by introducing the system of equations

$$X(m_1, f_a) = S_1(f_a)e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=1})} + S_2(f_a)e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=2})} + S_3(f_a)e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=3})} \tag{6.18}$$

$$X(m_2, f_a) = S_1(f_a)e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=1})} + S_2(f_a)e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=2})} + S_3(f_a)e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=3})} \tag{6.19}$$

where $X(m_1, f_a)$ and $X(m_2, f_a)$ respectively represent the observed STFT coefficients of the multi-source mixture, and represent the individual elements of the multichannel STFT vector $\mathbf{x}_{fn}$ as in equation 6.11. We note that, as compared to equation 6.11 the subscript for the STFT frame index n has been dropped in order to focus on illustrating the scenario within a *single* STFT analysis frame.

Thus we also denote that the choice of STFT frequency bin f is also arbitrary and thus we use the label f_a in Figure 6.4 and in equations 6.18 and 6.19. Therefore, both the STFT frame bin n and frequency bin f can be chosen *arbitrarily*, within the illustration we focus more on the STFT bin *local* and on describing its spatial components as described by the additive superposition of spatial images, which equations 6.18 and 6.19 convey in a manner that is equivalent to the general scenario described by equation 6.11.

Considering the observed coefficient $X(m_1, f_a)$, equation 6.18 conveys that the output coefficient is entirely additively determined by the sequence

$$\{S_1(f_a), S_2(f_a), S_3(f_a)\} \tag{6.20}$$

phase shifted by the sequence

$$\{e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=1})}, e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=2})}, e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=3})}\} \tag{6.21}$$

A somewhat challenging issue that will have to be faced is to parametrize entirely the effect both the magnitude and phase spectra of the complex valued sequence of variables $S_l(f_a)$ at a class bin l for which we allocate a location within the algorithm that we intend to convey the effect of the STFT coefficient of the frequency domain source signal $S_l(f_a)$ at a particular time frequency bin.

We also intend that by applying the appropriate phase shift $e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=1})}$ that the appropriate spatial contribution at microphone m due to source l will be appropriately modelled.

In considering more carefully the elements of the sequence 6.21, we can specify that the spatial effect of the phase shift parameter, essentially controls the look direction with respect to microphones $m = 1$ and $m = 2$ and is specifically controlled by the value of the parameter $\tau_m(\mathbf{k}_l)$. Recalling the far-field model as described in section 2.1.1.1, we had originally considered that a *reference point*, typically chosen as the approximate *center* could be spatially and geometrically defined. Therefore, in the time domain and according to the far field model the time domain signal $x_1(t)$ can be approximately modelled as

$$x_{m=1}(t) \approx s_{l=1}(t + \triangle\tau_{m=1,l=1}) + s_{l=2}(t + \triangle\tau_{m=1,l=2}) + s_{l=3}(t + \triangle\tau_{m=1,l=3}) \tag{6.22}$$

and a similar representation can be written for the signal $x_{m=2}(t)$. Assuming the far field model can approximately model the effect of multiple source signals in a multichannel context, we will also attempt to test the notion that initial phases of the source signals $S_l(f_a)$ can be adequately parametrized by modelling the observed signal at microphone m in this manner, which will be a perhaps novel but still uncertain concept to be evaluated.

In principle we have stated and propose that the time difference $\tau_m(k_o)$ is *microphone-specific* and corresponds to a time difference of arrival of the m microphone referenced to the reference point that was chosen as the array center. $\tau_m(k_o)$ is only *look direction* dependent as is the premise of the far field model and there for labelling it in this way, we intended to appropriately convey this.

It has been evidenced within the research of [24] that modelling the observed signal in this manner by spatially and geometrically sampling the area surrounding the microphone array with unit vectors $\{\mathbf{k}_o\}$ could be used to enhance the performance an SCM NMF parametrization of multichannel STFT signals and thus we will utilize this as a basis for exploring the overall concept further within the proposed research.

Thus, in focusing back on the interpretations of the source separation problem within the frequency domain we consider the expression of the system of equations first described within equations 6.18 and 6.19 now the form of a more matrix equation as given by

$$\begin{bmatrix} X(m_1, f_a) \\ X(m_2, f_a) \end{bmatrix} = \begin{bmatrix} e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=1})} & e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=2})} & e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=3})} \\ e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=1})} & e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=2})} & e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=3})} \end{bmatrix} \begin{bmatrix} S_1(f_a) \\ S_2(f_a) \\ S_3(f_a) \end{bmatrix} \tag{6.23}$$

which again emphasizes and makes it most evident here that the task of parametrizing the *spatial filtering* that occurs from sources to microphones assuming the far field model of signal propagation, is effectively a two part frequency problem: a signal parametrization problem and a (spatial) filter parametrization problem.

Since equation 6.23 lacks the parametrization of source to microphone *gain differences*, we can augment the parametrization once again by modifying it as given by

$$\begin{bmatrix} X(m_1, f_a) \\ X(m_2, f_a) \end{bmatrix} = \begin{bmatrix} |h_{1,1}|e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=1})} & |h_{1,2}|e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=2})} & |h_{1,3}|e^{j2\pi f_a \tau_{m_1}(\mathbf{k}_{l=3})} \\ |h_{2,1}|e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=1})} & |h_{2,2}|e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=2})} & |h_{2,3}|e^{j2\pi f_a \tau_{m_2}(\mathbf{k}_{l=3})} \end{bmatrix} \begin{bmatrix} S_1(f_a) \\ S_2(f_a) \\ S_3(f_a) \end{bmatrix} \tag{6.24}$$

where the terms $h_{m,l}$ are put in place to model an attenuation (scaling) between the lth source and mth microphone. For the superposition of at most L possible sources per time frequency bin, the attenuation factors $h_{m,l}$ will to some degree model a difference in the amplitudes or observed power of the terms $X(m_1, f_a)$ and $X(m_1, f_a)$. If we consider the point as mentioned within [16] for SCM based methods, it is said that typically for small microphone arrays, interchannel phase differences may be more visible than interchannel gain differences as conveyed by spatial covariance matrix processing, and therefore, we propose one interpretation of this point that suggests that grouping classes and linking sound components from different classes on the basis of similarity of their interchannel phase differences may be more of an effective point of focus than to focus on mth observed gains (observed variances) for modelling multichannel and multi source mixtures of sound components. Therefore, we propose that equation 6.23 represents a useful and appropriate simplification to the problem that conveys and emphasizes this point well with regards to suggesting how the additivity of sources should be parametrized.

We therefore, state now that we have addressed the issue of parametrizing possible superpositions of spatial source mixtures, having considered adequately both a spatial image interpretation of the observed multichannel STFT vector $\mathbf{x}_{fn}$ as well as a spatial covariance interpretation of its corresponding spatial covariance matrix $\mathbf{X}_{fn}$. It will now be required to consider how to extend the two-factor NMF model in terms of NMF parameter matrices $\mathbf{T}$ and $\mathbf{V}$ we suggest to the reader that we will inevitably have to rely upon the CNMF algorithm's multiplicative update rules in these two particular parameters in order to appropriately learn the amplitude (magnitude spectra) within the vectors corresponding to parametrized source estimates in each time frequency bin $\mathbf{s}_{fn}$.

Thus it will be proposed that it will be necessary for the reader to be capable of understanding both the global requirements of the CNMF algorithm as well as the local requirements of the CNMF algorithm, treating the observed data $\mathbf{x}_{fn}$ as data to be modelled appropriately across *all* time frequency bins, $f = 1, ..., F$ and $n = 1, ..., N$ while considering as a secondary consequence (and objective to be) the modelling of the spatial covariance matrix $\mathbf{X}_{fn}$. How this will be done is entirely dependent on the choice of NMF factors that

will extend the basic two factor NMF representation provided by $\mathbf{T}$ and $\mathbf{V}$ that serves as a starting point for the algorithm to be developed and built-upon.

Moving on from the illustration, we shall consequently propose to develop a set of NMF based CNMF update rules that will iteratively and adaptively populate the parameters of the model 6.23 across all time frequency bins.

We will derive an auxiliary function algorithm that is a multichannel but non covariance matrix extension similar to of the algorithm presented in section 5.2 and similar in principle to the algorithm presented in sections 5.3.2 and 5.3.4. Many methods within the previous chapter suggest appropriate and relevant interpretations , and provide techniques and suggestions as to how equation 6.24 can be optimized by introducing cost functions and mathematical optimization techniques defined upon the model and associated cost functions.

First, since we have considered the sources in the room environment to be stationary throughout the time interval to be analyzed by NMF, we propose that spatial characteristics of sound sources (pertaining to a frequency domain partitioned interpretation of their interchannel microphone properties) can be clustered and labelled according to a class index, and thus we use the term class in this context, synonymously with separated output signal. Again, we re-emphasize that the sound components will be combined additively, and eventually, in the proposed algorithms reconstruction step, but that the partitioning of components according to which class they have either been assigned to or to which the algorithm has learned that they should be assigned to will be dependent on the component to class correspondence, according to the partitioning or labelling scheme that has been defined according to the (clustering) indicator matrices.

In light of the multichannel algorithms covered in Chapter 4 [16,24,64], we fully acknowledge that the physical interpretation of the covariance matrix $\mathbf{X}_{fn}$ of the multichannel STFT vector per time frequency bin plays an important role in the respective reconstruction and clustering steps of these multichannel sound source separation algorithms, especially with regards to parts of the algorithms concerned with either clustering or partitioning sound components into their respective output classes, based on frequency domain spatial information, or equivalently interchannel time domain time difference of arrival information with regards to the microphone array. Again, we remind the reader of what the off diagonal elements of either the upper or lower triangular, as was first introduced in section 5.3.

Later in this chapter, we first introduce a key parameter $e^{j\Phi_S(f,n,k)}$, which can adequately be referred to as a *phase dictionary* for the purpose of attempting to model phase spectra of parametrized source estimates. Furthermore, its important will be applied for the purpose for accurate reconstruction of perceptually robust output sources within the algorithm's reconstruction step, which does not depending on a multichannel Wiener filter such as the

other algorithms considered thus far.

It will be demonstrated, how these CNMF parameters have been selected by design for characterizing the various parameters of the CNMF model and each play a particular role in characterizing the complex spectral and temporal signatures of output sound sources. Aside from this set of CNMF parameters, the proposed algorithm also introduces a set of clustering matrix parameters, $\mathbf{Z}$ and $\mathbf{Y}$, chosen for the purpose of assigning correspondence between target output classes and the set of all possible look directions and complex spectra in the CNMF component bins. Clustering and the proposed CNMF algorithm in tandem will represent the necessary tools for eliminating the look directions of disinterest to the target output classes, for example, and for converging to the correct configuration of complex spectra needed to additively represent the observed complex multichannel STFT data.

Figure 6.5 shows an illustration of properly clustered output classes, represented in the STFT domain, where assuming a high degree of temporal non overlap between competing sound sources, a simple yet physically meaningful separation can be achieved.

The corresponded between non (temporally) neighbouring sound components has been achieved on the basis of common interchannel (frequency dependent) spatial properties between sound components belonging to the same class.

A correspondence between each class and its characterizing optimal look direction vector, provides an optimal fit and explanation for sound components (with subtle, but important and quantifiable interchannel time differences) occurring as seen by the stereo array of microphone sensors.

The illustration provided by Figure 6.5 is based upon the notion of *minimal overlap* between spectra of competing source signals. This was discussed as an intentional consequence of the input signal design as described to some degree within section 1.5 and corresponding to the primary test case of musical mixtures of non-overlapping source signals where the musical signals correspond to notes from different instruments, played only when the other instruments happen to be silent with respect to one another.

The proposed algorithm's capacity for obtaining such a decomposition will be a point of emphasis to illustrate more clearly in the sections that follow.

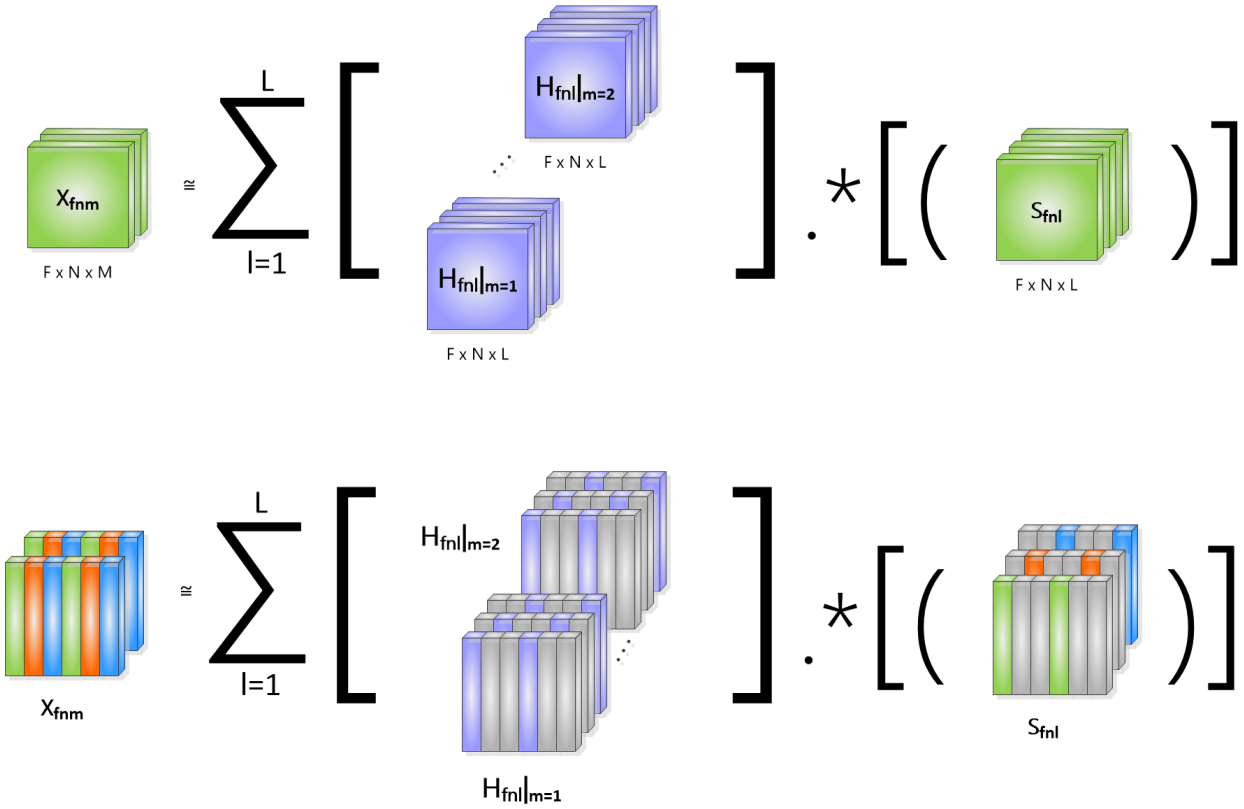

Figure 6.5: Class and Time Dependent Mixing Model Illustration

6.1.1 Connection to Spatial Covariance Matrix Processing

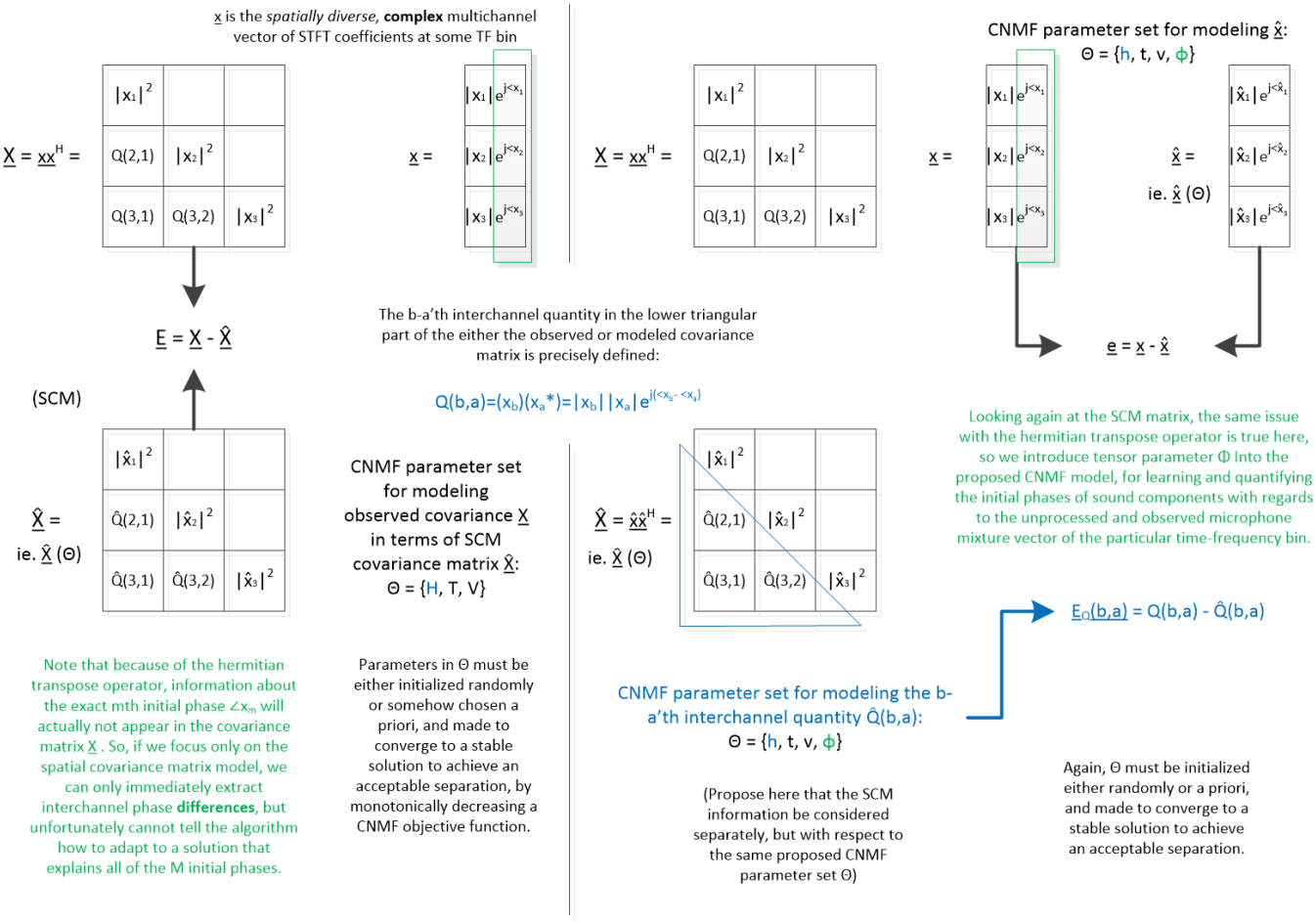

Figure 6.6: Side by side comparison of typical SCM CNMF Processing (left) vs. the proposed CNMF algorithm (right) at any particular time-frequency bin

Figure 6.6 depicts the importance of considering the spatial covariance matrix (SCM) per time frequency bin, which can be computed as covered in section 5.3.1, according to equation 5.18. We thus seek either an adequate parametrization (i.e. factorization) of either $\mathbf{x}_{fn}$ or its covariance matrix $\mathbf{x}_{fn} = \mathbf{x}_{fn}\mathbf{x}_{fn}^H$ and shall denote the factorized (parametrized) matrices per time frequency bin as $\hat{\mathbf{x}}_{fn}$ and $\hat{\mathbf{X}}_{fn}$, respectively.

Considering the spatial covariance matrices of time frequency bins provides a view of the observed data that allows us to begin considering how to tell the algorithm to adapt to a plausible representation, in each of the CNMF model parameters.

To enforce nonnegativity in a multichannel sense it should be understood that any computation of the observed vector $\mathbf{x}_{fn}$ to produce $\mathbf{X}_{fn}$ should result in $\hat{\mathbf{X}}_{fn}$ corresponding to a Hermitian positive semidefinite covariance matrix. Aside from the standard way of verifying

that a matrix is Hermitian positive semi-definite, by inspecting $\mathbf{X}_{fn}$ it can be quickly be understood that elements along the main diagonal must be positive valued.

Furthermore, if we largely agree that the multichannel STFT vector $\mathbf{x}_{fn}$ resulted from an additive multi-source mixture of possibly overlapping (ie. competing) sound components characterized in the same time frequency bin that $\mathbf{x}_{fn}$ corresponds to by related STFT coefficents $\mathbf{s}_{fn}$ then we should consider whether or not the processed observed covariance matrix $\mathbf{X}_{fn}$ immediately provides any hints as to how $\mathbf{s}_{fn}$ should be populated as a vector of amplitudes (or magnitudes) with a set of respective complex initial phases. The short answer that the proposed algorithm emphasizes is that the SCM modeling of STFT data in a multichannel sense provides negligible information, at least with regards to estimating the initial phases. Once again, it was pointed out in [24] that "the *absolute* phase of the sources is not significant from the parameter estimation point of view" in a CNMF SCM algorithm, and the rest of this particular algorithm was derived under this assumption which we see to be valid by considering the illustration in the left half of figure 6.6. This is only to say that if we restrict ourselves to deriving SCM CNMF based models, then in the reconstruction step of the output sources, the assignment of absolute phase (i.e. initial phase) complex spectra must be obtained (either elsewhere, or) likely via some means within the source reconstruction step. This can be seen in (C.10), where $\mathbf{y}_{fn}^{(q)}$ (here, q specifies the output class) is reconstructed as a function of all purely nonnegative valued parameters, except for the observed multichannel STFT vector $\mathbf{x}_{fn}$, which provides the absolute phase, per time frequency bin to be assigned to the separated output signal $\mathbf{y}_{fn}^{(q)}$.

To further develop an understanding of the hermitian positive semi-definite matrix $\mathbf{X}_{fn}$ we point out that from the computation of $\mathbf{x}_{fn}\mathbf{x}_{fn}^{H}$, the upper triangular part of the covariance matrix (not including the main diagonal) provides redundant information as that provided in the lower triangular part, which is why in Figure 6.6 it is only the lower triangular part of the SCM that is ever shown.

What the right half of Figure 6.6 in fact suggests, that is supplemental to the left half of the figure, is to update (i.e. to derive a set of update rules over the) the CNMF parameter set in order to directly resolve the initial phase per time frequency bin of the multichannel STFT vector $\mathbf{x}_{fn}$, as opposed to postponing resolution of source phases to the source reconstruction step as was done in [24], and similarly in [16].

In the next section we formally introduce the entire CNMF parameter set for the proposed algorithm, as we intend to build upon the ideas covered thus far. Our objectives going forth will be to demonstrate more in detail how the two factor basic NMF model in terms of $\mathbf{T}$ and $\mathbf{V}$ will be *extended* in order to

- Associate with $\mathbf{T}$ and $\mathbf{V}$ a *partitioning* according to a partition indicator matrix $\mathbf{Y}_{lk}$

that maps component indices $k = 1, ..., K$ to class indices $l = 1, ..., L$

- Associate with the classes $l = 1, ..., L$ the possibility of being characterized by its most significant and plausible look direction $o, ..., L$ where the look directions specify the *direction of arrival* and appropriate microphone-dependent phase shift to be associated with each spatial image occurring at each microphone m.

- Demonstrate how DoA kernel matrices first introduced in section 5.3.4 will be applied within the context of the proposed CNMF algorithm and model.

- Demonstrate how a phase dictionary $e^{j\Phi_S(f,n,k)}$ that was inspired according to the algorithm described in 5.2 will be applied within the context of the proposed CNMF algorithm and model.

- Describe how K-means clustering described within a multichannel context will be applied in a manner that was inspired by the SCM NMF algorithms detailed in the previous chapter in sections C.1.2 and C.2.1, and where shown in those algorithms to be closely interrelated with robust source reconstruction and appropriate spatial parametrization of the CNMF parameters related to modelling the effect of spatial filtering between NMF components specified by k and the m microphone observation. The same principle will apply within the proposed algorithm, and we must demonstrate this to be the case, in order to illustrate the reason why source reconstruction can be achieved successfully.

6.2 Algorithm Parameter Set

Table 6.2: Parameters List

	Parameter Significance
$[\tilde{\mathbf{x}}_{fn}]_m$	STFT of the recorded microphone data.
$[\hat{\mathbf{x}}_{fn}]_m$	Modelled STFT.
$[\mathbf{h}_{fk}]_m$	mth element of the Channel mixing vector as a function of the K total components.
$[\mathbf{h}_{fl}]_m$	mth element of the Channel mixing vector as a function of only the L most significant classes.
t_{fk}	Frequency template parameter as a function of the kth component.
v_{kn}	Activation parameter as a function of the kth component.
z_{ol}	Look direction to class membership indicator parameter.
y_{lk}	Component to class membership indicator parameter.
w_{fom}	Frequency x Look direction, channel mixing parameter
$e^{j\Phi_S(f,n,k)}$	Frequency x Activation x Component phase parameters to be estimated.
$\exp(j\Phi_W(f,o,m))$	Frequency x Look direction x channel allowed (pre-defined) set of phases based on time difference of arrival.
$\exp(j\Phi_U(f,l,n,m))$	Frequency x Class x Time activation bin x channel parameter to be configured based on thresholding ofthe learned Indicator values from the Pre-Clustering step.
$\boldsymbol{\mu}_{(f,l,n)}$	Alternative representation and naming of the 4-way tensor $\exp(j\Phi_U(f,l,n,m))$

That is, $[\boldsymbol{\mu}_{(f,l,n)}]_m = \exp(j\Phi_U(f,l,n,m))$

and for some f,l,n, $\boldsymbol{\mu}_{(f,l,n)} \in \mathbb{C}^{M\times 1}$ and for some f,l,n,m, $\exp(j\Phi_U(f,l,n,m)) \in \mathbb{C}$

Table 6.3: Indices List

	Index Significance
m	Microphone channel index
M	Total number of microphone channels
f	STFT frequency index
F	Total number of STFT frequency bins
n	STFT time activation index
N	Total number of STFT time bins
k	component bin index
K	Total number of allocated component bins for the model
l	class index
L	Total number of allocated class bins
o	look direction index
O	Total number of look directions

In the test configuration(s) that were used, the sizes that were chosen can be provided in summary as follows

- $M = 2$, must be chosen equal to the number of available micropphones.
- $L = 3$, should be chosen as the number of available sound sources to be separated.
- $K = 24$, suggested to be chosen as a number divisible by L in order to partition the matrix $\mathbf{Y}$, appropriately.
- $O = 27$, equal to the number of equally spaced look direction unit vectors.
- $F = 1025$, FFT size of STFT was chosen to be 1024.
- $N = 390$, number of STFT frames is dependent on sample rate and length of signal which were 16000 and roughly 20 seconds approximately.

Figure 6.7: Proposed Model

Figure 6.7 depicts a graphical representation of the proposed CNMF model that will be used to obtain a separable representation of the observed multichannel STFT data (i.e. $\hat{\mathbf{x}}_{fn}$ index at time-frequency bin f-n) containing complex spectra of spatially filtered sound sources. Although we intend to show how multiplicative update rules can be developed, we first must demonstrate a so-called *unfolding* of the CNMF matrix factorization decomposed in terms of its parameters set.

- Magnitude spectra modelled by the t_{fk} and v_{kn} parameters. Traversing component index $k = 1 : K$
- Complex phase part of the magnitude spectra are modelled by the parameter $e^{j\Phi_S(f,n,k)}$
- Frequency and look direction dependent DoA kernel vectors $\mathbf{w}_{fo}$ for the purpose of explaining the multichannel STFT vectors per time-frequency bin as opposed to the multichannel SCM's per time-frequency bin.
- We introduce a frequency dependent spatial filtering parameter that is also a function of activation index n and is given by $\mathbf{h}_{fnk} \in \mathbf{M} \times \mathbf{1}$. We propose that a parameter $\exp(j\Phi_U(f,l,n,m))$ should be applied in such a way that $\mathbf{h}_{fnk} \in \mathbf{M} \times \mathbf{1}$ is appropriately modified depending on how $\exp(j\Phi_U(f,l,n,m))$ should be modified.
- We propose that $\exp(j\Phi_U(f,l,n,m))$ should be populated within a pre-clustering step once an optimal spatial weights matrix z_{ol} has first been populated. Taking the appropriate information from z_{ol}, the clustering output, and a pre-configured class to component indicator matrix y_{lk} we propose that $\exp(j\Phi_U(f,l,n,m))$ can then be configured in an optimal manner and across activation (STFT frame) indices $n = 1, ..., N$. By introducing the parameter $\exp(j\Phi_U(f,l,n,m))$ we intend for the algorithm to have the capability of linking class dependent information to time activation dependent information as occurs in Figure 6.5. For instance, spatially we might want to separate and localize sound sources originating from three distinct locations within the acoustic environment. However, we may not guarantee (any conditions upon the observed input data) that the STFT spectra of the true sources be *adjacent* at all times in terms of either time or frequency. In other words, we would like the algorithm to be capable of both learning and distinguishing between sound sources corresponding to different and/or unique spatial locations. Furthermore, at some time frequency bins and at some spatial locations, we might somehow be able to determine or perhaps *infer* that the STFT spectra corresponding to a time-frequency-*spatial* bin be essentially null, and we would like for the algorithm to be able to adapt itself appropriately in terms of being

able to provide a meaningful parametrization as it pertains to the source separation problem under such a scenario.

- Given this so-called optimal configuration, it is in fact only the parameters t_{fk}, v_{kn}, and $e^{j\Phi_S(f,n,k)}$ whose update rules should be applied iteratively to further minimize the CNMF objective function (i.e. its multichannel and time-frequency optimization criterion).

6.3 Algorithm Development, Approach, and Input Signal Design

6.3.1 Design of Input Data Signals

In the following section the design of the input signals to be used as the primary test case for exercising the algorithm is shown. The expected difficulty in terms of solving the intended primary test case is by no means of the highest calibre of difficulty, as for example, the designed input signals are purposefully chosen to be orthogonal in time. However, we hope that venturing out with the objective of solving it will provide a fundamental proof of concept for the beginnings of developing new and potentially more challenging test cases.

It will still be of great importance however, to benchmark the output separation quality of the primary test case against what is assumed to be the current state of the art algorithm's [24] separation quality in terms of solving the separation problem against the same set of input signals.

6.3.1.1 Test Case 1

Figures 6.8 to 6.14 demonstrate a test case scenario where each spatially diverse microphone signal represents a mixture of six difference notes (chords, in fact) from three different musical instruments, specifically, a violin, a guitar and a piano. In summary, each instrument (ie. sound source) plays two musical notes in total, with exaggerated pauses in between notes. Thus, intentionally, in modelling the sound mixture signals that will exercise the separation capability of the CNMF algorithm, the MCRoomSim mixture of musical notes per microphone signal was constructed such that notes had little to no overlap between temporally activated regions of sound. Therefore, the proposed algorithm was envisioned to be able to

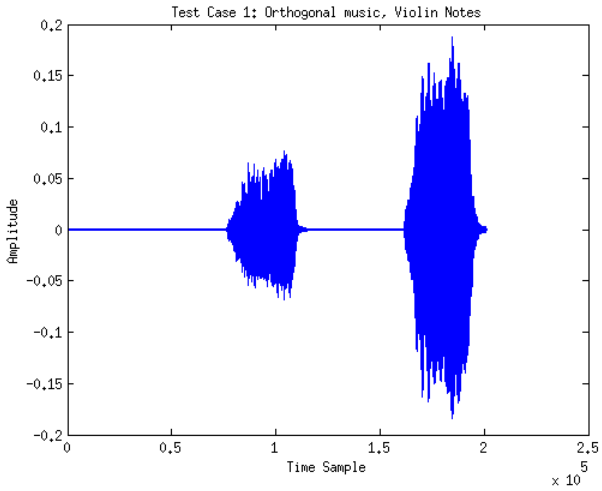

Figure 6.8: A pair of Violin Notes, Time Domain

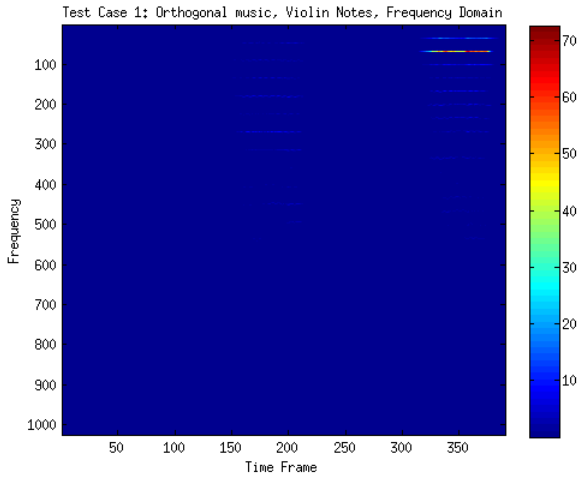

Figure 6.9: A pair of Violin Notes, Frequency Domain

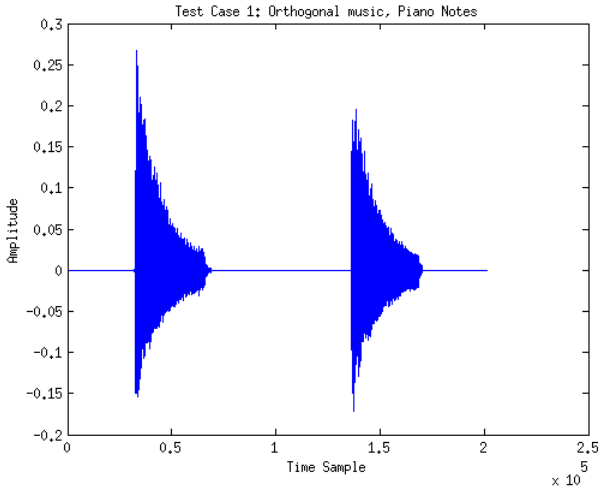

Figure 6.10: A pair of Piano Notes, Time Domain

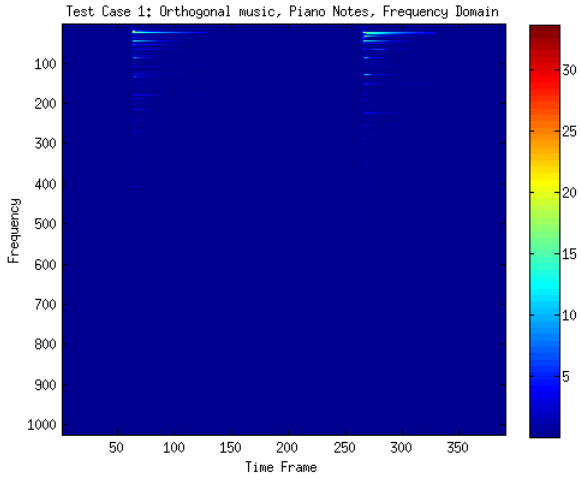

Figure 6.11: A pair of Piano Notes, Frequency Domain

suppress, per output class, interfering musical notes, in the output CNMF representation. This particular test case, aside from being the primary debug test case, was also the motivation for being able to derive the proposed algorithm's interchannel clustering scheme as outline in section 6.5.1. Prior to settling upon that scheme, a consideration of the possible physical interpretation of the CNMF model parameters that would achieve an acceptable separation is illustrated in 6.5

The STFT parameters for analyzing the source and observed microphone signals in the STFT domain were hinted at in section 6.2.

The sample rate was chosen as 16000Hz applied to signals of length in the order of

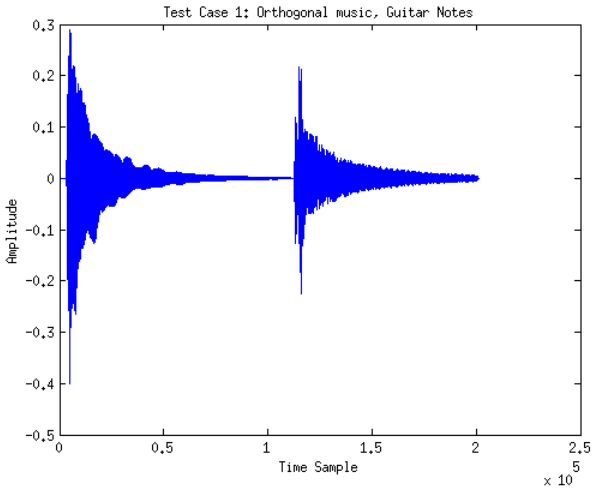

Figure 6.12: A pair of Guitar Notes, Time Domain

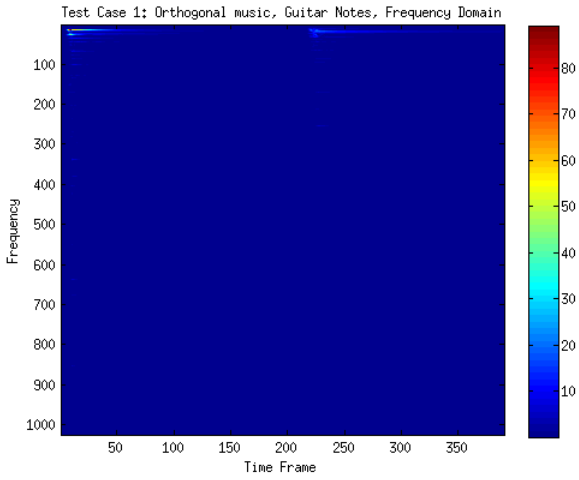

Figure 6.13: A pair of Guitar Notes, Frequency Domain

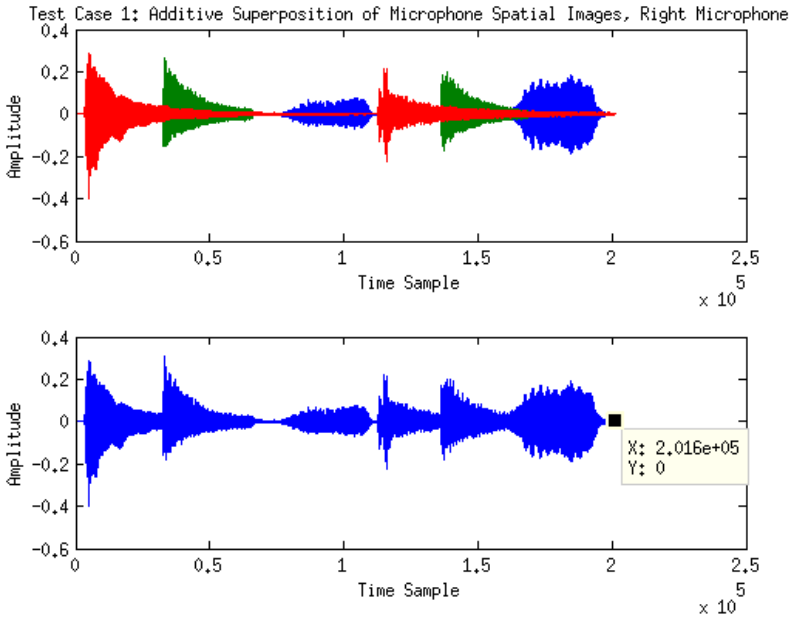

Figure 6.14: Additive Superposition of the three source signal as seen by the right microphone

approximately 20 seconds in duration. The FFT size for the STFT was chosen as 1024, this resulted in a total number of STFT frames coresponding to the signal duration of 390. This is reflected in the set of figures on the right hand side of each pair of figures.

We propose that the described set of input signals represent a relatively easy scenario to initially challenge the algorithm with, however, more challenging test scenarios were applied with speech signals replacing the musical signals as the source signals.

The corresponding set of figures corresponding to two supplementary test cases designed with speech are included in the Appendix and the reader is pointed to section D.5 to consider them if they so choose. The simulation results (treated as output data to be evaluated) for the supplementary results were included within the next chapter when benchmarking the proposed algorithm against a reference algorithm, however, no further references will be made to them for now, in describing the proposed algorithm within the current chapter.

6.3.1.2 Microphone Array Geometry and Pre-Defined Directions of Arrival configuration

We first describe the method for defining the spatial sampling of the area surrounding the microphone array and suggest that the parameter $\mathbf{w}_{fo}$ will be used to encode this information as frequency domain phase differences according to

$$\mathbf{w}_{fo} = \begin{bmatrix} |[\mathbf{w}_{fo}]_1| \exp j2\pi \frac{(f-1)F_s}{N_{STFT}} \tau_1(\mathbf{k}_o) \\ |[\mathbf{w}_{fo}]_2| \exp j2\pi \frac{(f-1)F_s}{N_{STFT}} \tau_2(\mathbf{k}_o) \\ \vdots \\ |[\mathbf{w}_{fo}]_m| \exp j2\pi \frac{(f-1)F_s}{N_{STFT}} \tau_M(\mathbf{k}_o) \end{bmatrix} \in \mathbb{C}^{M \times 1} \tag{6.25}$$

as was suggested could be achieved in the context of DoA kernel matrices as within [24]. As a first requirement, a set of look direction unit vectors $\{\mathbf{k}_o\}$ must be defined for a certain number of look directions indexed by the variable $o = 1, ..., O$ and how this was configured for the proposed algorithm, for $O = 27$ sampling a half circle in the x-y plane was described according to Figure 6.2 and within section 6.1.0.1.

This was based upon the far field model of sound propagation, for which [17, 28] were adequate for describing and a further treatment was provided within 2.1.1.1. The DoA kernel matrices were described within section 5.3.4 and within [24].

Thus time differences

$$\tau_m(\mathbf{k}_o) = -\frac{\mathbf{k}_o^T(\mathbf{n}_m - \mathbf{n}_p)}{v} = -\frac{\mathbf{k}_o^T \mathbf{n}_m}{v} \tag{6.26}$$

represent *microphone-dependent* time differences, referenced with respect to the geometrical reference point $\mathbf{n}_p$ which corresponds to the array center and is typically set to 0 such that $\mathbf{n}_p = \mathbf{0}$.

For testing the proposed algorithm $\mathbf{n}_{m=1}$ and $\mathbf{n}_{m=2}$ were placed along the positive and negative y-axis at equally spaced distance from the array center within MCRoomSim according to equation 6.9. The microphone spacing was chosen to be 10cm.

It should be noted however, that populating equations 6.26 and 6.25 requires no knowledge of the configuration of MCRoomSim so long as the observed signals and microphone

array geometry achieved within MCRoomSim returns impulse response vectors that were properly used to filter the spatial image signals and additively combine them in the correct way as according to equation 6.2.

Assuming that $M = 2$ and we have a stereo microphone array, we can generate a predictable frequency dependent phase pattern per look direction o as was done by considering the equations presented within 2.1.1.2

Figures 6.15 and 6.16 can be obtained by populating the following tensor for $M = 2$ show the result of this by computing the complex argument $\arg(\cdot)$ of the time difference defined between channel $m = b = 2$ and $m = a = 1$ where we can define

$$\tau_{ba}(\mathbf{k}_o) = \tau_b(\mathbf{k}_o) - \tau_a(\mathbf{k}_o) = \tau_2(\mathbf{k}_o) - \tau_1(\mathbf{k}_o) \tag{6.27}$$

and then consider the look direction and frequency dependent phase pattern generated by considering the complex argument $\arg(\cdot)$ of the *frequency-domain interchannel* phases as already defined within 6.25. Thus to consider the *interchannel phase difference* as a real number between $-\pi$ and π we consider the interchannel quantity defined between channel $b = 2$ and channel $a = 1$ as given by

$$\arg\left[\exp\left(j2\pi\frac{(f-1)F_s}{N_{STFT}}(\tau_2(\mathbf{k}_o) - \tau_1(\mathbf{k}_o))\right)\right] \tag{6.28}$$

which results in the subsequent figures if we vary all frequencies $f = 1, .., F$ across all look directions $o = 1, ..., O$. Figure 6.16 corresponds however to $|\mathbf{k}_o| = r = 1\text{m}$ where as Figure 6.15 corresponds however to $|\mathbf{k}_o| = r = \frac{1}{3.20}\text{m}$.

Figures 6.15 and 6.16 are generated from considering all possible look directions (with the total number of look directions chosen to be 27) spatially sampling a half circle in a 2D-plane surrounding a stereo microphone array.

By considering each spatially sampled look direction, one at a time, it can evidently be seen from these two figures that associated with each possible look direction is a frequency dependent signature. Also what can be seen from these figures is that these frequency dependent signatures of neighbouring look direction bins can be noticeably similar.

Using the complex tensor described by equation 6.34, which is inspired by equation 5.32 (which was defined in a spatial covariance matrix processing context) and neglecting its magnitude, and considering only the phase, an interchannel phase pattern as a function of frequency and look direction can be computed for any pair of microphones b and a. In this instance, the configuration chosen is $b = 2$ and $a = 1$, simply due to the fact that we consider the microphone array to be stereo.

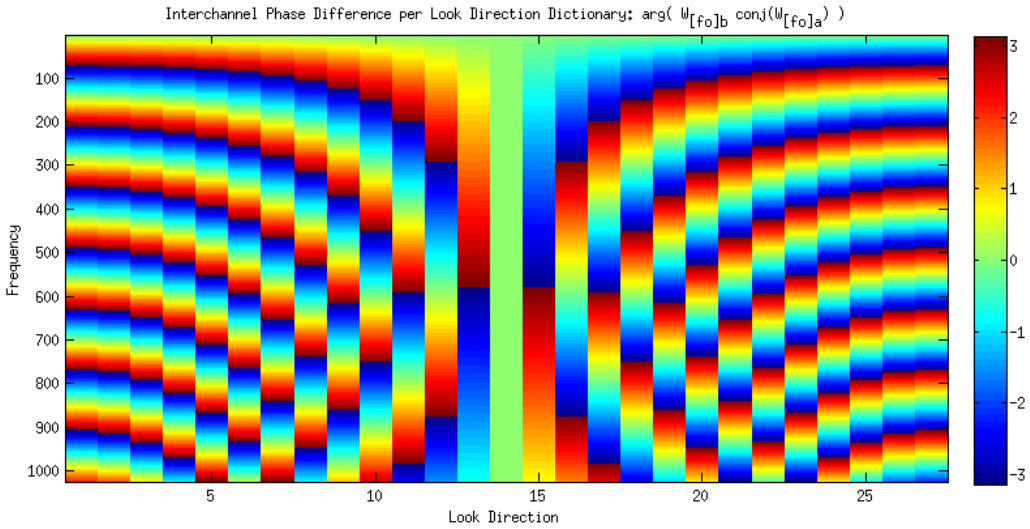

Figure 6.15: Interchannel phase difference between channel 'b' and 'a', computed per look direction for spatial radius of $\frac{1}{3.20}$ meters

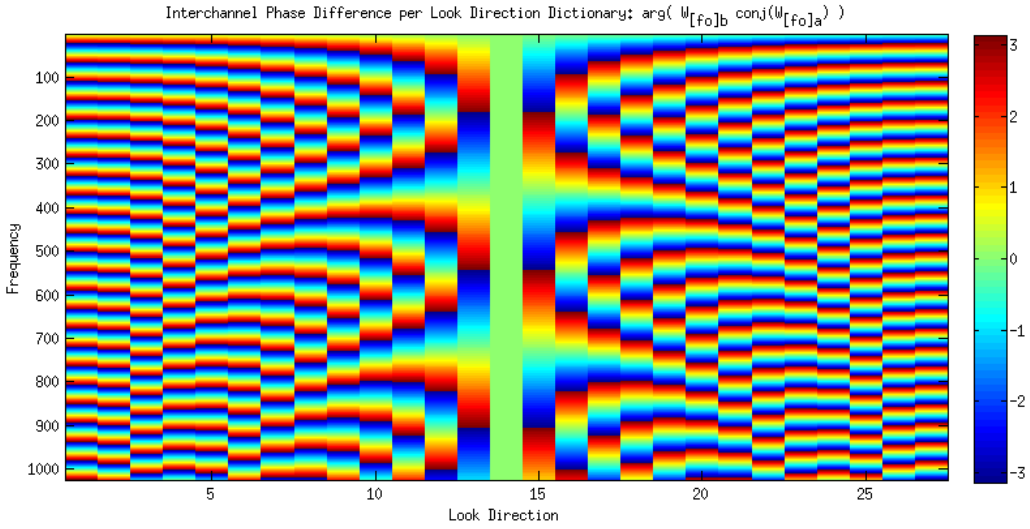

Figure 6.16: Interchannel phase difference between channel 'b' and 'a', computed per look direction for spatial radius of 1m

As can be seen from the figures, a characterizing property of spatially sampling each and any of the look directions as a function of frequency is that the phase difference between microphones varies across frequency as a periodic sawtooth waveform. For look directions 1-13, as the frequency increases the phase difference considered as the quantity $\arg(\frac{w_{fo,b}}{|w_{fo,b}|}\frac{w^*_{fo,a}}{|w_{fo,a}|})$ has a positive slope, where as for look directions 15-27 the interchannel phase difference has a negative slope. For all of the look directins except for look direction 14, the sawtooth has an amplitude (peak value) of + or - π.

6.4 Algorithm

In order to obtain similar updates as in the algorithms [16,24,25] we shall begin by modelling the parametrized multichannel spectrogram as

$$[\hat{\mathbf{x}}_{fn}]_m = \sum_{k=1}^{K} [\mathbf{h}_{fnk}]_m t_{fk} v_{kn} e^{j\Phi_S(f,n,k)} \tag{6.29}$$

in each time, frequency, and microphone bin, where

$$\hat{\mathbf{x}}_{fn} = \begin{bmatrix} [\hat{\mathbf{x}}_{fn}]_1 \\ [\hat{\mathbf{x}}_{fn}]_2 \\ \vdots \\ [\hat{\mathbf{x}}_{fn}]_M \end{bmatrix} \in \mathbb{C}^{M\times 1} \text{ , and } \mathbf{h}_{fnk} = \begin{bmatrix} [\mathbf{h}_{fnk}]_1 \\ [\mathbf{h}_{fnk}]_2 \\ \vdots \\ [\mathbf{h}_{fnk}]_M \end{bmatrix} \in \mathbb{C}^{M\times 1}$$

and propose that the observed multichannel spectrogram can be approximated as

$$[\tilde{\mathbf{x}}_{fn}]_m \approx \sum_{k=1}^{K} [\mathbf{h}_{fnk}]_m t_{fk} v_{kn} e^{j\Phi_S(f,n,k)} + E_{fnm} \tag{6.30}$$

where

$$\tilde{\mathbf{x}}_{fn} = \begin{bmatrix} [\tilde{\mathbf{x}}_{fn}]_1 \\ [\tilde{\mathbf{x}}_{fn}]_2 \\ \vdots \\ [\tilde{\mathbf{x}}_{fn}]_M \end{bmatrix} \in \mathbb{C}^{M\times 1}$$

is the observed multichannel STFT vector in each time frequency bin.

The various matrix and tensor quantities on the right hand side of the equation describing the proposed model, 6.29 can be considered to be (in a maximum likelihood sense) mainly either parameters of the model or in some cases missing data. Equation 6.29 seeks to model the effect of *additive* spatial source spectra, as seen by the mth microphone, and as explained by analogy according to the description of equation B.13. Since the *true* source spectra are unknown, we will seek to parametrize them utilizing the proposed matrix and tensor parameter set $\theta = \{W, T, V, Y, Z, \Phi_W, \Phi_S, \Phi_U\}$ within a CNMF optimization framework. We can recall that the description of the parameter set θ was provided in a brief overview and within section 6.2.

We also imply that equation 6.29 focuses on *directly* modelling the multichannel STFT vector $\tilde{\mathbf{x}}_{fn}$ as opposed to focusing on modelling its *processed* spatial covariance matrix, per time-frequency bin, as done in [16,24] and as described in section 5.3.1. We intend for the parameters t_{fk} and v_{kn} to parametrize the magnitude spectra of source estimates and for the parameter $e^{j\Phi_S(f,n,k)}$ to *directly* model the complex phase part of estimates of source spectra. Therefore, reconstructed source estimates will not require a Wiener filter based

reconstruction step as utilized in [16, 24]. The phase to be associated with reconstructed source estimates in the time-frequency domain will be dependent on how phases from the *phase dictionary* $e^{j\Phi_S(f,n,k)}$ are utilized.

We may need to introduce additional processing (such as clustering) upon some of the matrix parameters in order to achieve an enhanced and more meaningful of the parametrization of the decomposition (e.g. with respect to time-frequency domain spatial filtering between acoustic sources and microphones) of the observed multichannel spectrogram.

We propose that an appropriate selection of CNMF matrix parameters will benefit us in terms of finding a desirable decomposition of the source spectra that can be properly labelled and will allow appropriate reconstruction of estimated sources, that closely approximates the effect of the true, but unknown source signals. If this can be shown to be the case, in the convergence of the iterative learning algorithm, then the observed multichannel spectrogram $\tilde{\mathbf{x}}_{fn}$ as described in equation 6.30 should become very close to the parametrized multichannel spectrogram $\tilde{\mathbf{x}}_{fn}$ and the algorithm should drive the approximation error term E_{fnm} towards having a mean squared value, averaged across all time frequency and channel bins, that approaches zero as the algorithm updates itself iteratively, per CNMF iteration.

Furthermore, in order to cluster over the NMF bases as in section IV of Sawada 2013, let us represent $\mathbf{h}_{fnk}$ in two possible alternative and equivalent ways :

$$\mathbf{h}_{fnk} = \sum_{l=1}^{L} (\mathbf{h}_{fl} \circledast \boldsymbol{\mu}_{(f,l,n)}) y_{lk} \tag{6.31}$$

$$[\mathbf{h}_{fnk}]_m = \sum_{l=1}^{L} [\mathbf{h}_{fl}]_m y_{lk} e^{j\Phi_U(f,l,n,m)} \tag{6.32}$$

Where $l = 1 : L$ specifies a cluster or "class" index, and $\mathbf{h}_{fl} \in \mathbb{C}^{M\times 1}$, y_{lk} is introduced as a cluster indicator latent variable that associates a particular class l with NMF component k . Note $L < K$ throughout the algorithm and we should choose $L_{inititial} = K$ to initialize the algorithm. The variable l is used to parametrically index spectral and spatial membership to estimated source l classified terms of an intermediate representation that we intend will eventually converge towards an accurate representation corresponding to one of the *true* source signals. The parameter $\mathbf{h}_{fnk}$ is a 4-way tensor that encodes the *frequency-domain spatial parametrization* of the unknown source spatial images, per observed microphone channel, per time-frequency bin, and per component bin. Here, the parameters $\mathbf{h}_{fnk}$ and $\mathbf{h}_{fl}$ are analogous to the parameters $[\mathbf{H}]_{fk}$ and $[\mathbf{H}]_{fl}$ of equation 5.22, but are intended to model the effect of spatial filtering in the non-SCM time-frequency domain. As was the case in [16], for the role of the parameter $[\mathbf{H}]_{fl}$, we intend for the parameter $\mathbf{h}_{fl}$ to model and to parametrize

the spatial filtering of the acoustic transfer function between the mth microphone and the lth source. In the proposed algorithm, we further encode $\mathbf{h}_{fl}$ with pre-determined spatial and geometrical information describing the multichannel microphone array, which was a key concept as described within [24].

Therefore, applying the main idea from Virtanen and Nikunen's algorithm, let us also remodel $\mathbf{h}_{fl}$, by defining look direction unit vectors $\{\mathbf{k}_o\}$ for a set of look directions $o = 1, ..., O$ that are to spatially sample a predetermined set of directions of arrival surrounding the microphone array, such that

$$\mathbf{h}_{fl} = \sum_{o=1}^{O} \mathbf{w}_{fo} z_{ol} \tag{6.33}$$

where

$$\mathbf{w}_{fo} = \begin{bmatrix} |[\mathbf{w}_{fo}]_1| \exp j2\pi \frac{(f-1)F_s}{N_{STFT}} \tau_1(\mathbf{k}_o) \\ |[\mathbf{w}_{fo}]_2| \exp j2\pi \frac{(f-1)F_s}{N_{STFT}} \tau_2(\mathbf{k}_o) \\ \vdots \\ |[\mathbf{w}_{fo}]_m| \exp j2\pi \frac{(f-1)F_s}{N_{STFT}} \tau_M(\mathbf{k}_o) \end{bmatrix} \in \mathbb{C}^{M \times 1} \tag{6.34}$$

and where $\tau_m(\mathbf{k}_o)$ is the time difference of arrival at microphone m associated with look direction $\{\mathbf{k}_o\}$. Furthermore, we will propose to use spatial weights z_{ol}, sub-indexed across classes, $[Z]_{:,1}, [Z]_{:,2}, ..., [Z]_{:,L} \in \mathbb{C}^{O \times 1}$ as the "feature vectors" over which we will base our cluster processing, as suggested could be done by Nikunen and Virtanen (but in their algorithm, to be done at the output of a post-processing clustering). Equations 6.33 and 6.34 fully describe the spatial sampling of adjacent look directions surrounding the microphone array according to a set of spatial unit vectors $\{\mathbf{k}_o\}$, from which we can describe an approximate far-field model interpretation of the spatial filtering between the mth microphone and the lth unknown class and/or the oth look direction. The DoA kernel matrices as described within [24] that were both frequency and look-direction dependent were applied within an M microphone spatial covariance matrix based algorithm in order to pre-encode the algorithm with geometrical information pertaining to directions of arrival of source components impinging upon the microphone array. The frequency domain *interchannel* time differences could be *read-off* from the DoA kernel matrix by considering the off-diagonal elements of the DoA kernel matrix at frequency f and look direction o. The DoA kernel matrix was described in the previous chapter according to equation 5.32. For the proposed algorithm equation 6.34 describes an $M \times 1$ complex vector and not an $M \times M$ DoA kernel like matrix quantity. Thus we interpret equation 6.34 as specifying the direct frequency domain time difference

between the *direct* path between the kth parametrized source component's (unknown) spatial position and the mth microphone's (known) spatial position, as according to the far-field model as fully described in section 2.1.1.1. If needed, we could easily generate a DoA kernel matrix like quantity by computing the quantity $\mathbf{w}_{fo}\mathbf{w}_{fo}^H$. We derive an interchannel-based objective function as an algorithm extension in the next section following the current section. We propose that, in the proposed algorithm as well as the DoA SCM algorithm, that the effectiveness of the interchannel modelling, per time frequency bin, in a two microphone source separation configuration, is revealed in how well the algorithm's learned $\hat{\mathbf{x}}\hat{\mathbf{x}}^H$ (i.e. the *parametrized* SCM matrix per time frequency bin) is able to approximately model the *observed* spatial covariance matrix per time frequency bin $\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H$. We emphasize that we are not the first to propose this idea, since the DoA algorithm also acknowledges this requirement, however, it does so in focusing on an SCM optimization framework, and furthermore, without parametrizing the initial phases of source spectra. Where as both algorithms utilize the notion of DoA kernel matrices computing in the form of $\mathbf{w}_{fo}\mathbf{w}_{fo}^H$, the proposed algorithm will be shown to accurately model the off diagonal element of the observed covariance matrix $\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H$ very closely, which could signify that initial phase estimation combined with the DoA kernel matrix concept, leads to improved parametrization of the SCM model, with the processed quantity $\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H$ acting as the target data quantity to be modelled per time-frequency bin.

In how spatial vectors $\{\mathbf{k}_o\}$ were populated in order to test the proposed algorithm, for test case 1, 27 total vectors $\{\mathbf{k}_o\}$ were utilized to spatially sample the area surrounding a two-microphone microphone array corresponding to a *half-circle*. As was described in [24], it would be straightforward to populate a set of vectors spatially sampling a unit *sphere* surrounding the microphone array according to 6.34, but doing so would be more computationally expensive since it would likely require a much larger set of vectors, and the algorithm complexity notably increases as a function of the total number of allocated look directions.

Continuing with the description of the current model, we define respectively the likelihood and negative log-likelihood, as follows:

$$P(\tilde{\mathbf{x}}|W,T,V,Y,Z,\Phi_W,\Phi_S,\Phi_U) = \prod_{f=1}^{F}\prod_{n=1}^{N}\prod_{m=1}^{M}\mathcal{N}_c([\tilde{\mathbf{x}}_{fn}]_m|[\hat{\mathbf{x}}_{fn}]_m,\sigma^2=1) \tag{6.35}$$

$$\begin{aligned}\log P(\tilde{\mathbf{x}}|W,T,V,Y,Z,\Phi_W,\Phi_S,\Phi_U) &= \sum_{f=1}^{F}\sum_{n=1}^{N}\sum_{m=1}^{M}\log\mathcal{N}_c([\tilde{\mathbf{x}}_{fn}]_m|[\hat{\mathbf{x}}_{fn}]_m,\sigma^2=1)\\ &= \sum_{f=1}^{F}\sum_{n=1}^{N}\sum_{m=1}^{M}\log\left(\frac{1}{\sqrt{2(1)\pi}}e^{(-\frac{|[\tilde{\mathbf{x}}_{fn}]_m-[\hat{\mathbf{x}}_{fn}]_m|^2}{2(1)})}\right)\\ &= \sum_{f=1}^{F}\sum_{n=1}^{N}\sum_{m=1}^{M}\log\left(\frac{1}{\sqrt{2\pi}}\right)-\frac{|[\tilde{\mathbf{x}}_{fn}]_m-[\hat{\mathbf{x}}_{fn}]_m|^2}{2}\end{aligned} \tag{6.36}$$

Here, we apply the same simplifying assumption as that made in [16] and [24], of the variance term of the univariate complex normal distribution being unity, and only seeking to explain the mean value of the complex normal distribution. Interpreted probabilistically, and under this assumption, the proposed model can be understood as requiring that the expected value of the model (mean value, seen as a parameter) in each time frequency and channel bin approximates the observed value of the data. Noting of course that the optimization problem at hand involves complex-valued observed data, and a corresponding complex-valued mean to be estimated in each time frequency and channel bin, the model is therefore designed to have the intended capacity to learn the optimal additive representation required for a successful separation, given complex additivity of STFT interfering and competing sound sources, in a multichannel mixing and linear time-invariant sense.

As a result of not specifying the covariance matrix in a complex multivariate sense, we avoid the consequence of having to invert an $M \times M$ matrix as a key step of the algorithm. In an effort to obtain improved results, a second auxiliary function is derived, aiming to satisfy that the off diagonal elements of such a covariance matrix-like quantity are well modelled by the parameters, also in a CNMF-like fashion, as a function of the same model parameters.

But to focus our efforts once again towards equation 6.36, requires us to consider minimizing the negative log-likelihood (allowing ourselves to ignore constant terms) and to then define the objective function to be minimized via auxiliary function derivation more compactly as follows:

$$\mathcal{L}(W,T,V,Y,Z,\Phi_W,\Phi_S,\Phi_U) = \sum_{m=1}^{M}\sum_{f=1}^{F}\sum_{n=1}^{N}|[\tilde{\mathbf{x}}_{fn}]_m-[\hat{\mathbf{x}}_{fn}]_m|^2 \tag{6.37}$$

with the intent of minimizing the squared error between the observed and modelled spectrogram, where we name

$$E_{fnm} = [\tilde{\mathbf{x}}_{fn}]_m - [\hat{\mathbf{x}}_{fn}]_m \tag{6.38}$$

the "error" of the model. Given the definitions presented thus far we can re-write the model more explicitly as:

$$[\hat{\mathbf{x}}_{fn}]_m = \sum_{k=1}^{K} \Big(\sum_{l=1}^{L} \Big(\sum_{o=1}^{O} [\mathbf{w}_{fo}]_m z_{ol} \Big) y_{lk} e^{j\Phi_U(f,l,n,m)} \Big) t_{fk} v_{kn} e^{j\Phi_S(f,n,k)}$$

and simply use a different notation to express

$$[\mathbf{w}_{fo}]_m = w_{fom} e^{j\Phi_W(f,o,m)} \tag{6.39}$$

in order to convey that this parameter can be represented by its absolute value and its phase. The effect of the w_{fom} parameter is to apply a channel and frequency depending scaling to a particular look direction. This corresponds to the fact that impulse responses between source positions and microphone positions when considered in the frequency domain (i.e. transfer functions) are frequency dependent and will consequently spectrally shape the source signal's frequency response. Assuming that a correspondence between a *significant* look direction o and class l can be learned by an appropriate clustering method, the w_{fom} parameter will thus represent the magnitude response of the transfer function between the mth microphone and the lth class that the significant look direction corresponds to. As mentionned, we intend for this optimal correspondence to be specified by the spatial weights parameter z_{ol}, in the likeness of the spatial weights parameter that was utilized in the CNMF DoA SCM based algorithm of [24]. Furthermore, simply applying the definition 6.39, we have that:

$$[\hat{\mathbf{x}}_{fn}]_m = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{o=1}^{O} w_{fom} e^{j\Phi_W(f,o,m)} e^{j\Phi_U(f,l,n,m)} z_{ol} y_{lk} t_{fk} v_{kn} e^{j\Phi_S(f,n,k)} \tag{6.40}$$

And now define an auxiliary variable C_{fnmklo} (can also be interpreteted as *latent components*) and the condition that they must satisfy as given by

$$\sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{o=1}^{O} C_{fnmklo} = [\tilde{\mathbf{x}}_{fn}]_m \tag{6.41}$$

$$\text{Where } \tilde{\mathbf{x}}_{fn} = \begin{bmatrix} [\tilde{\mathbf{x}}_{fn}]_1 \\ [\tilde{\mathbf{x}}_{fn}]_2 \\ \vdots \\ [\tilde{\mathbf{x}}_{fn}]_M \end{bmatrix} \in \mathbb{C}^{M \times 1}$$

is a M-channel microphone (observation) time-freq bin.

Also define a set of weights β_{fnmklo} such that

$$\sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{o=1}^{O}\beta_{fnmklo} = [\mathbf{1}]_{fnm}$$

Claim that the auxiliary function is minimized with respect its auxiliary variable C_{fnmklo} and achieves tangency with the main objective function when it is chosen as follows:

$$\begin{aligned} C_{fnmklo} &= w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}e^{j\Phi_W(f,o,m)}e^{j\Phi_U(f,l,n,m)}e^{j\Phi_S(f,n,k)} + \beta_{fnmklo}([\tilde{\mathbf{x}}_{fn}]_m - [\hat{\mathbf{x}}_{fn}]) \\ &= w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} + \beta_{fnmklo}(E_{fnm}) \end{aligned}$$

And define,

$$\beta_{fnmklo} = \frac{w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}}{\hat{x}_{fnm}} \tag{6.42}$$

$$\hat{x}_{fnm} = \sum_{c}^{K}\sum_{d}^{L}\sum_{e}^{O} w_{fem}z_{ed}y_{dc}t_{fc}v_{cn} \tag{6.43}$$

and now try the auxiliary function:

$$\mathcal{L}^{+}(W,T,V,Y,Z,\Phi_W,\Phi_S,\Phi_U) = \sum_{m,f,n,k,l,o}\frac{\left|C_{fnmklo} - w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\right|^2}{\beta_{fnmklo}}$$

$$\begin{aligned} &= \sum_{m,f,n,k,l,o}\frac{1}{\beta_{fnmklo}}\{C_{fnmklo}C^{*}_{fnmklo} - C_{fnmklo}w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}e^{-j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} \\ &\quad - C^{*}_{fnmklo}w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} + w^2_{fom}z^2_{ol}y^2_{lk}t^2_{fk}v^2_{kn}\} \end{aligned} \tag{6.44}$$

Take the partial derivatives of $\mathcal{L}^{+}$ with respect to each of its parameter matrices and tensors, also treating C_{fnmklo} as a parameter. We will not sub the expression for C_{fnmklo} in until after we have computed each partial derivative. We do this in order to obtain multiplicative updates. Therefore note: $C_{fnmklo}C^{*}_{fnmklo}$ is a term of the auxiliary function that can be ignored when computing each partial derivative.

In summary, we propose that

$$\mathcal{L}^{+}(W,T,V,Y,Z,\Phi_W,\Phi_S,\Phi_U) \geq \mathcal{L}(W,T,V,Y,Z,\Phi_W,\Phi_S,\Phi_U) \tag{6.45}$$

by the application of Jensen's inequality. This will be shown, by substituting equation 6.41 into equation 6.37, subsequently resulting in the following alternative way of writing the objective function

$$\mathcal{L}(W,T,V,Y,Z,\Phi_W,\Phi_S,\Phi_U) = \sum_{m=1}^{M}\sum_{f=1}^{F}\sum_{n=1}^{N}\left|\sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{o=1}^{O} C_{fnmklo} - [\hat{\mathbf{x}}_{fn}]_m\right|^2 \tag{6.46}$$

where the definition of $[\hat{\mathbf{x}}_{fn}]_m$ can be specified as according to equation 6.40 and we will now seek to apply Jensen's inequality as first described within equation 3.2.

In order to do so we first must acknowledge that the operator $|\cdot|^2$ is a convex function. In comparing $\mathcal{L}^+$ as defined in 6.44 to $\mathcal{L}$ as written according to equation 6.46 we note that the summation, $\sum_{klo}(\cdot)$, that occurs within equation 6.46 (i.e. within both of the equations 6.41 and 6.40) is restricted to being *within* the operator $|\cdot|^2$ (i.e. inside of the convex operator). In other words, the operator wraps the summation.

We then note that in equation 6.44, the summation has been moved *outside* of the convex operator by introducing a set of appropriate *weights* β_{fnmklo} as defined within equation 6.42. Subsequently the convex operator $|\cdot|^2$ appears now on the inside of the summation $\sum_{klo}(\cdot)$ within the function defined as $\mathcal{L}^+$ according to equation 6.44. Since this follows the concept of Jensen's inequality $\mathcal{L}^+$ is shown to be an upper bounding auxiliary function to the objective function $\mathcal{L}$.

Therefore we have that $\mathcal{L}^+$ can be verified to be an auxiliary function of $\mathcal{L}$, and have shown equation 6.45 to be true, by having considered Jensen's inequality. We further recall that a similar approach was used, applying Jensen's inequality to show that the auxiliary functions as specified by equations 5.37 and 5.10 were also auxiliary functions for their respective objective funtions.

It then can be shown how to optimize the proposed CNMF parameter w_{fom} with respect to $\mathcal{L}^+$ by computing the partial derivative $\frac{\partial \mathcal{L}^+}{\partial w_{fom}}$ as given by

$$\begin{aligned}
\frac{\partial \mathcal{L}^{+}}{\partial w_{fom}} =& \sum_{n,k,l} \frac{1}{\beta_{fnmklo}} [-C_{fnmklo} z_{ol} y_{lk} t_{fk} v_{kn} e^{-j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} \\
& - C^{*}_{fnmklo} z_{ol} y_{lk} t_{fk} v_{kn} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} + 2 w_{fom} z_{ol}^2 y_{lk}^2 t_{fk}^2 v_{kn}^2] \\
&= -\sum_{n,k,l} \frac{2 z_{ol} y_{lk} t_{fk} v_{kn}}{\beta_{fnmklo}} \operatorname{Re}\{C^{*}_{fnmklo} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\} \\
&+ \sum_{n,k,l} \frac{2 w_{fom} z_{ol}^2 y_{lk}^2 t_{fk}^2 v_{kn}^2}{\beta_{fnmklo}} \\
&= -\sum_{n,k,l} \frac{2 \hat{x}_{fnm}}{w_{fom}} \operatorname{Re}\{C^{*}_{fnmklo} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\} \\
&+ \sum_{n,k,l} 2 z_{ol} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm} \\
&= -\frac{1}{w_{fom}} \sum_{n,k,l} 2 \hat{x}_{fnm} \operatorname{Re}\{w_{fom} z_{ol} y_{lk} t_{fk} v_{kn} \\
&+ w_{fom} z_{ol} y_{lk} t_{fk} v_{kn} \frac{E^{*}_{fnm}}{\hat{x}_{fnm}} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\} + \sum_{n,k,l} 2 z_{ol} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm} \\
&= -\frac{1}{w_{fom}} \sum_{n,k,l} 2 \hat{x}_{fnm} w_{fom} z_{ol} y_{lk} t_{fk} v_{kn} \operatorname{Re}\{1 + \frac{E^{*}_{fnm}}{\hat{x}_{fnm}} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\} \\
&+ \sum_{n,k,l} 2 z_{ol} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm}
\end{aligned}$$

Set $\frac{\partial \mathcal{L}^{+}}{\partial w_{fom}} = 0$, and re-arrange in order to obtain the multiplicative update:

$$w_{fom} \leftarrow w_{fom} \frac{\sum_{n,k,l} z_{ol} y_{lk} t_{fk} v_{kn} \operatorname{Re}\{\hat{x}_{fnm} + E^{*}_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{n,k,l} z_{ol} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm}} \tag{6.47}$$

The update derivations for $z_{ol}, y_{lk}, t_{fk}, v_{kn}$ are all obtained in a similar manner and provided in summary for convenience:

$$z_{ol} \leftarrow z_{ol} \frac{\sum_{m,f,n,k} w_{fom} y_{lk} t_{fk} v_{kn} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,f,n,k} w_{fom} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm}} \tag{6.48}$$

$$y_{lk} \leftarrow y_{lk} \frac{\sum_{m,f,n,o} w_{fom} z_{ol} t_{fk} v_{kn} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,f,n,o} w_{fom} z_{ol} t_{fk} v_{kn} \hat{x}_{fnm}} \tag{6.49}$$

$$t_{fk} \leftarrow t_{fk} \frac{\sum_{m,n,l,o} w_{fom} z_{ol} y_{lk} v_{kn} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,n,l,o} w_{fom} z_{ol} y_{lk} v_{kn} \hat{x}_{fnm}} \tag{6.50}$$

$$v_{kn} \leftarrow v_{kn} \frac{\sum_{m,f,l,o} w_{fom} z_{ol} y_{lk} t_{fk} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,f,l,o} w_{fom} z_{ol} y_{lk} t_{fk} \hat{x}_{fnm}} \tag{6.51}$$

In order to obtain the update for the variable phase parameter, $\Phi_S(f,n,k)$, we have the option compute the partial derivative $\dfrac{\partial \mathcal{L}^+}{\partial \Phi_S(f,n,k)}$, massage the result and set it to equal 0 and solve for $\Phi_S(f,n,k)$, or equivalently we can define $\mathcal{G}^+_{\mathcal{S}}$ from $\mathcal{L}^+$ by removing all terms that do not depend on $\Phi_S(f,n,k)$. And thus construct $\mathcal{G}^+_{\mathcal{S}}$ as follows:

$$\mathcal{G}^+_{\mathcal{S}} = 2 \operatorname{Re}\{(e^{j\Phi_S(f,n,k)}) \sum_{m,l,o} \frac{C^*_{fnmklo} w_{fom} z_{ol} y_{lk} t_{fk} v_{kn} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)]}}{\beta_{fnmklo}}\}$$

Inspect, and choose :

$$\begin{aligned}\Phi_S(f,n,k) &= -phase\{\sum_{m,l,o}\frac{C^*_{fnmklo}w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)]}}{\beta_{fnmklo}}\}\\ &= -phase\{\sum_{m,l,o}\hat{x}_{fnm}C^*_{fnmklo}e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)]}\}\\ &= -phase\{\sum_{m,l,o}\hat{x}_{fnm}(w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}e^{-j\Phi_S(f,n,k)}\\ &+ \frac{w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}}{\hat{x}_{fnm}}E^*_{fnm}e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)]})\}\\ &= -phase\{\sum_{m,l,o}w_{fom}z_{ol}y_{lk}t_{fk}v_{kn}(\hat{x}_{fnm}e^{-j\Phi_S(f,n,k)}+E^*_{fnm}e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)]})\}\end{aligned} \tag{6.52}$$

The respective update rules for $w_{fom}, z_{ol}, y_{lk}, t_{fk}, v_{kn}$, and $\exp(j(\Phi_S(f,n,k))$ can be shown to monotonically decrease the auxiliary function as specified by equation 6.44 however, we will have to consider a second auxiliary function analysis for obtaining plausible spatial covariance matrices $\mathbf{X}_{fn}$ across all time frequency bins which will be detailed in the following section. By selecting a certain configuration from the two sets of updates of what update rules to enable and disable an optimal configuration was determined that typically provided the best source separation performance with regards to unknown speech or music signals.

We will consider in the next chapter how the optimal configuration was demonstrated to monotonically decrease booth of the cost functions that we have considered within this chapter.

Furthermore, in Appendix section A.8 we describe how the numerator and denominator quantities of multiplicative update rules corresponding to CNMF parameters can be computed more efficiently by considering the decomposition of each number and denominator per update rule. Implementation of the multiplicative update rules was challenging in the sense of considering how to parallelize certain computations to be computed as quickly and efficiently as possible on a modern desktop machine. For the purpose of demonstrating the algorithmic properties of the update rules in the next chapter we demonstrate that we propose the implementation used to test the algorithm were clean and free of errors, since we obtained monotonic convergence graphs that verifying the update rules to decrease the cost functions by a certain amount at each iteration. This was the desired and expected consequence that we sought to achieve.

At this point, the reader is pointed to appendices A.10 and A.9.1 for the rest of the procedure pertaining to the proposed algorithm. Presented in section A.9.1 is an extension to

the proposed algorithm that causes the parametrization of equation 6.29 to become optimal in a spatial covariance matrix sense. The objective function and auxiliary function for the extension are respectively described according to equations A.88 and A.93. Multiplicative update rules are obtained by optimizing the auxiliary function with respect to the parameter set $\theta = \{W, T, V, Y, Z, \Phi_W, \Phi_S, \Phi_U\}$, however we suggest that some parameters needn't be updated algorithmically, but instead on the basis of clustering. Section A.10 details this.

6.5 Algorithm Illustration Considering Musical Notes

If we reconsider the algorithm test case referred to as 'test case 1' and originally described in section 6.3 we can describe an illustration to the problem, conceptually in terms of what we may expect the separation to provide.

The following *hypothetical* illustration shown in figure 6.17 for the proposed algorithm demonstrates a targeted (desired representation of) sound source separation that can be achieved assuming each musical note (here we consider 6 notes in total, two per instrument) can be represented spectrally and across the time duration of the note (and in a low rank sense) by two respective basis vectors each belonging to the frequency template and time activation NMF dictionary matrix factors $\mathbf{T}$ and $\mathbf{V}$, respectively.

We also consider in this example illustration that in the desired $\mathbf{V}$ matrix, that is, the time activation factor matrix, that the outputted notes considered per component bin k, contain minimal overlap with regards to other notes corresponding to other component bins.

Therefore, the outputted CNMF representation shown here most closely corresponds to a representation which could be outputted for Test Case 1, in which musical notes from a mixture piano, violin, and guitar instruments was constructed with the help of the MCRoomSim tool and encoded into a stereo mixture of microphone channels.

It was indeed originally hypothesized that the proposed algorithm would do well at separating the complex multichannel STFT dataset in a CNMF sense, by separating each musical note (with all its spectral and temporal properties, conveyed in the STFT domain by its fourier coefficients) into a single rank 1 approximation of the note, corresponding to only a single k component bin, with minimal or negligible contribution from any other k component bins.

This idea was later dropped in favour of taking the approach of using the indicator matrix $\mathbf{Y}$ to partition the components as groups of frequency domain signals, which as a group necessarily belonged to a single class, per class bin l.

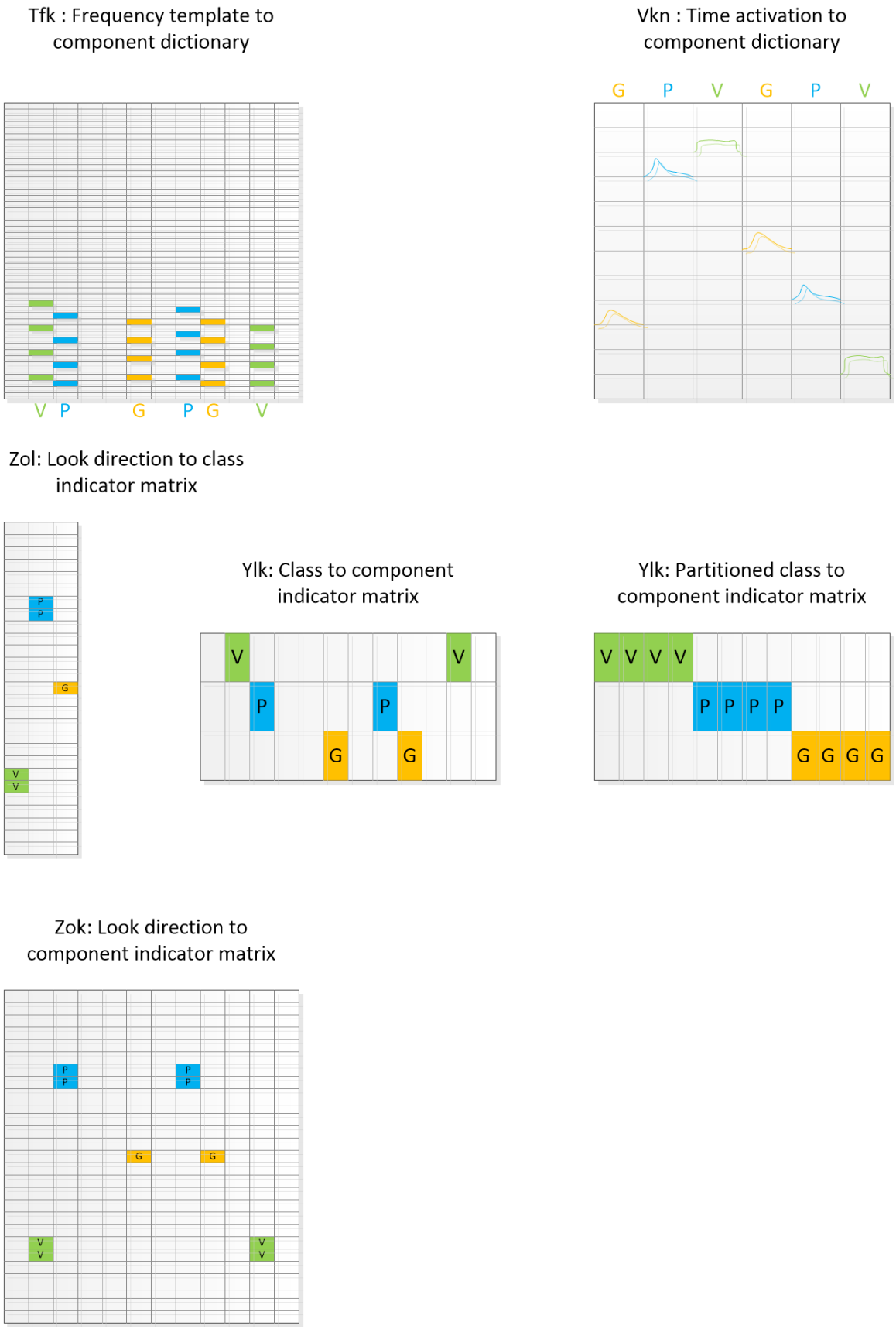

Figure 6.17: Illustration Considering Six Musical Notes from a Guitar, Piano and Violin

The difference between the two philosophies is illustrated in Figure 6.17 by considering the partitioned and non-partitioned versions of $\mathbf{Y}$.

Note that the illustration does not coincide with the algorithm configuration used in the next chapter and within generating the source separation that is to be demonstrated within the next chapter. The illustration is hypothetical and meant to illustrate the development of the algorithm in terms of assuming what a plausible output corresponding to test case 1 might might look like graphically and conceptually.

Assuming the partition indicator matrix $\mathbf{Y}$ as will be done in the next chapter, we obtain the result of not having any empty columns in the factor matrices $\mathbf{T}$ and $\mathbf{V}$ which was determined to be problematic for obtaining monotonic convergence and in fact introduced unwanted artefacts in various matrix parameters not only corresponding to $\mathbf{T}$ and $\mathbf{V}$.

In appendix section 6.2 we also demonstrate an *orthogonality* cost function that we attempted to derive using a gradient descent based technique, however this was also dropped and disabled in favour of better demonstrating performance configuring it to be disabled.

Thus we emphasize that according to the configuration that we suggest according to section section A.10 that will be detailed further when discussing simulated results contained within the next chapter that we demonstrate that the algorithm is capable of achieving a consistent performance with respect to source separate of unobserved source signals in the STFT domain.

Since musical signals corresponding to test case 1 were initially used as the primary test scenario for the proposed algorithm it allowed the justification and experimentation of the algorithm configuration proposed within this chapter should to be chosen as the optimal configuration of the proposed algorithm that we believe should be capable of separating most source mixtures, music or speech.

This was justified since we later applied the same algorithm with similar configuration to inputs that we constructed from speech without having to change the configuration in most cases.

6.5.1 K-means Pre-Clustering Step

The following scenario corresponds to what we have denoted as 'test case 1' and thus the primary test case whose signals were demonstrated as within section 6.3.1.1.

Therefore, the time-frequency phase representation according to Figure 6.18 was generated from computing the *complex argument* $\arg(\cdot)$ per time frequency bin of the spatial covariance matrix $\mathbf{X}_{fn} = \mathbf{x}_{fn}\mathbf{x}_{fn}^{H} \in \mathbb{C}^{M\times M}$ at the off-diagonal element of the SCM matrix which if we denote channel $m = 2 = b$ and $m = 1 = a$ then we can generalize the $b - a$th

off diagonal element in the *lower triangular* part of the covariance matrix $\mathbf{X}_{fn}$ for scenarios when $M > 2$.

In the following section we outline a clustering step that we suggest should be computed for a certain number of iterations prior to computing the NMF iterations, according to a stopping condition defined upon the K-means distortion measure J.

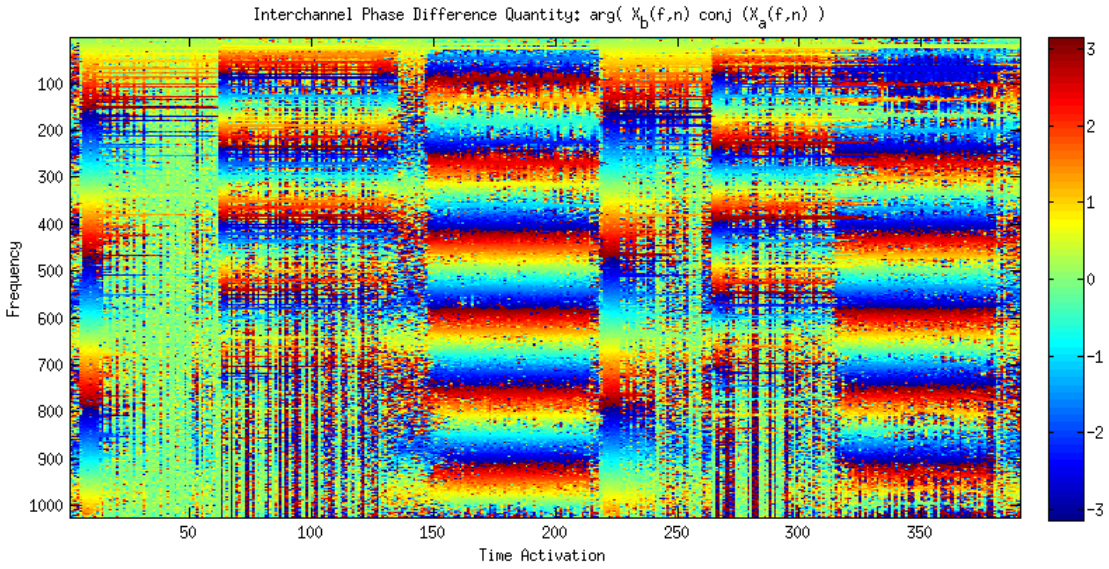

Figure 6.18: Interchannel Phase Difference Quantity between channel 'b' and 'a'

Do K-means clustering over the quantity

$$\tilde{G}_{fn}(b,a) = \arg(\tilde{Q}_{fn}(b,a)) \triangleq \arg(\tilde{X}_b(f,n)\tilde{X}_a^*(f,n)) \tag{6.53}$$

Define $[g\_v]_n \triangleq \tilde{G}_{fn}(b,a).col(n) \in \mathbb{R}^{F\times 1}$

$$J = \sum_{n=1}^{N}\sum_{q=1}^{L} p_{qn}||[g\_v]_n - \mathbf{c}_q||^2 \tag{6.54}$$

1. Randomly initialize the set of L mean row vectors $\{\mathbf{c}_1, ..., \mathbf{c}_q, ..., \mathbf{c}_L\}$ for $q = 1, ..., L$. $\mathbf{c}_q \in \mathbb{R}^{F\times 1}$

2. Compute assignment of soft indicators, p_{qn} with the variable $m = 2$:

$$p_{qn} = \frac{1}{\sum_{r=1}^{L}\left(\frac{||[g\_v]_n - \mathbf{c}_q||}{||[g\_v]_n - \mathbf{c}_r||}\right)^{\frac{2}{m-1}}} \tag{6.55}$$

3. Compute assignment of means $\mathbf{c}_q$, with the variable $m = 2$:

$$\mathbf{c}_q = \frac{\sum_{n=1}^{N} {p_{qn}}^m [g\_v]_n}{\sum_{n=1}^{N} {p_{qn}}^m} \tag{6.56}$$

4. Evaluate the distorted measure J: Stop, if it has been minimized to a satisfactory degree, if not continue by repeating steps 2 and 3.

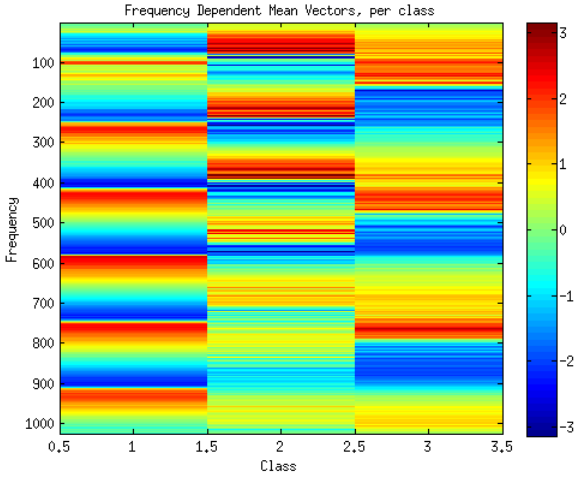

Figure 6.19: Frequency Dependent Mean Vectors, per class

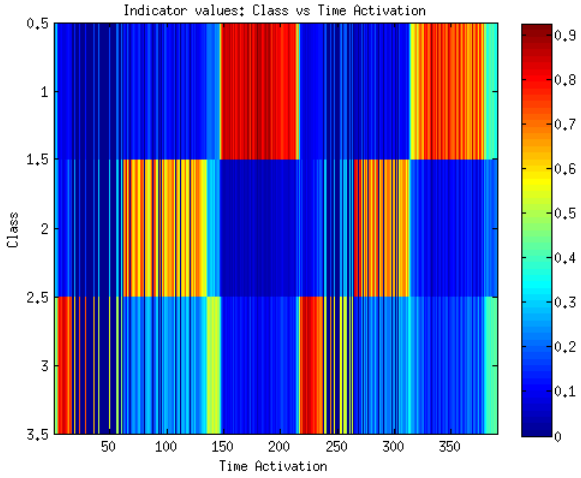

Figure 6.20: Indicator values: Class vs Time Activation

The physical interpretation of Figure 6.20 is that the indicators can be used as a class dependent partitioning of information related to which sound sources are temporally consistent (from an interchannel time difference perspective) across time activation bins. When the degree of overlap between competing sound sources is minimal, the indicator values have a much more straightforward clustering interpretation than when there could be a heavy degree of overlap.

The presence of zeros in any time activation bins for a particular class signifies that the k-means clustering pre-step has determined that (for the current pair of microphones being considered) there exists no significant detectable phase difference pattern between sound sources for the sound source (or multiple sound sources) that correspond to a particular look direction within the acoustic environment. This (the presence of zeros in the indicator bin) could either mean that there exists silence or that even perhaps the sound source originates

from a null look direction (one geometrically equidistant to the two microphones) such as look direction 14.

As Figure 6.18 represents the interchannel phase difference data generated from observed interchannel data, not modelled interchannel data, it will later be shown how the aforementioned notion causes some difficulty in adequately modelling the observed interchannel data. This is evident for the guitar note present at the beginning of the test case, during the decay portion of the guitar note signal.

Figure 6.19 demonstrates the mean vectors obtained at the output of the K-means clustering. They represent frequency dependent phases and can be considered to serve as basis vectors, per output class, corresponding to the *look directions of most significance*.

By computing a search within the so-called look direction dictionary as shown within figure 6.15, an *optimal correspondence* between look directions and each of the L total mean vectors can be computed. The class label l per cluster then indicates to which look direction should the lth class be associated with. This is highly meaningful since we will initialize the CNMF algorithm with a component to class partition defined according to the configuration of the matrix $\mathbf{Y}$. Therefore, per class l, whatever component are assigned to its correspond partition, can be interpreted in advanced to be necessarily associated with the optimal look direction o that is to be associated with the lth class.

Since the result shown corresponds to non-overlapping signals, the significance of the result is meaningful to us in explaining what we might expect to obtain, in having construct the musical signals in non-overlapping manner as intended. A possible extension and interpretation of this to the case of more overlapping but still slightly non-overlapping test scenario is then proposed

1. We note that the activation matrix parameter $\mathbf{V}$ indexed at the k-nth element specifies a sequence of activations in each row corresponding to a component k.

2. We note that the K-means indicators p_{qn} provide a *class to activation* correspondence on the basis of analyzing the spatial properties of matrices $\mathbf{x}_{fn}$ in each time frequency bin f-n.

3. A side note is that here there in fact exist L mean vectors and not 'K' and therefore the name K-means could be converted to be thought of as an 'L-means' algorithm for $q = 1, ..., L$.

4. We consider the possibility (i.e. option) of thus *partitioning* the matrix $\mathbf{Y}$ as to where the two possible options (i.e. philosophies) to the partitioning of $\mathbf{Y}$ were illustrated in Figure 6.17. According to this figure, and having compared the two possibilities, we

suggest to partition $\mathbf{Y}$ in such a way that $K = 24$ and $L = 3$, 8 CNMF components are partitioned per CNMF output class l. Therefore, a sparse representation of $\mathbf{Y}$ is achieved immediately that allows for simpler and more meaningful interpretation of the parametrization of source signals for $l = 1, ..., L$.

5. We also note that Figure 6.20 which shows the learned indicator values, seemingly corresponds to a physical interpretation that we have proposed that we might expect for the activation matrix parameter $\mathbf{V}$ as described within section 6.17 for this particular test case of musical mixtures of notes from different instruments.

6. According to our suggested partitioning, we might assign in fact up to 8 possible CNMF components per class l, where as the figure somewhat suggests that *two* physical notes be assigned at least two distinct component bins within the outputted representation. If we choose the partitioned representation of $\mathbf{Y}$ as being more useful, then we propose that up to 8 CNMF components could be applied to determining a representation of the two musical notes per output class (musical) instrument (i.e. per source).

Therefore, we intend to apply this possible interpretation to suggest how the proposed algorithm should be configured, given the output of the K-means algorithm, and in particular the indicators p_{qn} which can be utilized in such a way to populate the activations v_{kn} in a way that takes into account what was described.

Furthermore, through consideration of this interpretation, it was found necessary to conceive a pre-processing clustering step (to be introduced in section 6.5.1), along with a new phase conjugation (as a function of time activation index n) parameter $e^{(j\Phi_U(f,l,n,m))}$ which was originally thought to be needed as an interleaved processing or post-processing step, but was later found to be just as effective, if not more effective, as a pre-processing step, with the results saved, and later used as another view of reference quantities for the proposed CNMF algorithm to check against.

The value of the K-means preprocessing step was found to be that it could provide an early estimation and coupling of the most significant classes and look directions. And the mean and indicator matrices of the K-means would both reveal highly important information to be parsed and used valuably by the CNMF proposed. The role of the conceived $e^{(j\Phi_U(f,l,n,m))}$ parameter would be to in part ensure that the correct conjugation be applied to *classes of disinterest* in the correct time activation bins.

6.6 Key Observations and Conclusions for this Chapter

To this point we have covered an in-depth description of concepts surrounding the proposed algorithm. We now review some of these concepts before moving onto the next chapter. Firstly we propose that we have derived a spatially intelligible and configurable CNMF algorithm that estimates as part of source parameters absolute phases (i.e. $e^{j\Phi_S(f,n,k)}$) to be used for constructing complex-valued spectral sound components when combined in tandem with the matrix parameters $\mathbf{T}$ and $\mathbf{V}$. This part of the model was inspired by single channel CNMF as detailed in [25].

To further parametrize the multichannel CNMF model in terms of spatial signatures to be associated with learned complex spectra of sound components per time-frequency bin, the number of relevant spatial signatures to be identified need be equivalent to the number of output classes, since each output source should associate a frequency dependent signature (multichannel frequency domain filtering operation) on top of its sound components (e.g. musical notes) that can be partially identified by considering interchannel microphone time differences as part of the model and/or a priori (pre-clustered) information.

We emphasize that clustering occurs in the frequency domain and that correctly resolving a reliable configuration depends on a DoA kernel-like quantity, namely the pre-coding of the $e^{j\Phi_W(f,o,m)}$ parameter, which encodes all possible look directions when traversing the $o = 1 : O$ index, in terms of parametrizable interchannel microphone phase differences and the spatial geometry of the microphone array, by considering the far field model of spatial sound propagation.

The key information that we extract from applying the K-means like pre-clustering step as outlined in section 6.5.1 are:

- We obtain a very good hint of the look directions of significance parametrized by the CNMF parameter $\mathbf{Z}$. This will be confirmed in the next chapter when reviewing the results of the converged CNMF algorithm. We thus are able to make the simple and effective assignment of one optimal look direction o per class bin l and assign a '1' to associate the linking of a look direction o to a class l and a '0' to dissociate the linking of a look direction to a class. Thus, for example, assuming we allocate $L = 3$ and $O = 27$, that is, 3 total classes and 27 possible look directions, we only assign three '1's in total within the matrix $\mathbf{Z}$ (i.e. a single '1' per column of the matrix $\mathbf{Z}$). The optimally configured and significant look directions encoded within the z_{ol} (effectively learned from doing K-means) can be seen to directly correspond to the look directions

used when originally configuring MCRoomSim to generate the multichannel mixture signals corresponding to precise spatial filtering of source signals (musical signals for test case 1). Therefore, we equip the algorithm beforehand with a configuration that we hope will lead to a plausible source separation.

- We obtain a very good hint as to how to populate the phase conjugation parameter $e^{(j\Phi_U(f,l,n,m))}$, a 4-way tensor that we proposed should be included in the model, and that would be related to improving both the local and global behaviour of covariance matrices in all time frequency bins across quantities internal to the CNMF algorithm (namely to associate the correct class bin to time activation bin correspondence). We conceptually propose that, once we know with a fair amount of confidence at what time frequency bins a certain class of signals components should and should not be activated, then we need to "turn off" the phase shift associated with signal components *not* belonging to a class. Without introducing $e^{(j\Phi_U(f,l,n,m))}$, then the parameter $e^{j\Phi_W(f,o,m)}$ might harshly apply the wrong interchannel phase shifts to the wrong set of CNMF complex valued spectral components as characterized entirely by $\mathbf{T}$, $\mathbf{V}$, and $e^{j\Phi_S(f,n,k)}$. We again point out that the suggested way of populating $e^{(j\Phi_U(f,l,n,m))}$ is dependent on the p_{qn} K-means indicator matrix, as given by equation 6.55 and as shown in Figure 6.20. The parameter p_{qn} can be thought of as a priori information to the CNMF algorithm and provides a suggested way to consider what sound components at different activation bins should likely be grouped into the same class.

We further propose the following key points to be considered about the derived proposed CNMF algorithm:

1. We have developed multiplicative update rules corresponding to real or complex valued CNMF parameter matrices and tensors. The update rules corresponding to the CNMF parameters t_{fk}, v_{kn}, z_{ol} can y_{lk} can be verified to maintain the nonnegative property that we impose.

2. We have derived a Euclidean Based NMF optimization procedure that is formulated in a non-SCM manner and treats the observed STFT vector per time frequency bin as the target quantity to be modelled as opposed to the processed SCM matrix per time frequency bin as the quantity to be modelled.

3. The Majorization Minimization technique that was applied in order to obtain the multiplicative update rules was most similar to those developed within [25] and [24].

4. Clustering can be viewed the alignment of time differences (i.e. class and frequency dependent phase difference) between microphones per output source indexed by output class. In other words it can be viewed as having the objective of resolving spatial impulse responses between each microphone and output source pair.

5. Due to the pre-coding of spatial geometry, spatial weights, and the proposed clustering technique, the proposed algorithm possesses a spatial directivity and source localization-like capability of linking source STFT components in non-adjacent STFT bins. The spectral characteristics of the source estimates of true sources can be learned iteratively and adaptively by the CNMF algorithm via updating the NMF factors corresponding to t_{fk} and v_{kn} as well as the time frequency phase spectra by updating $e^{j\Phi_S(f,n,k)}$. Plausible time frequency representations of sources can be obtained by reconstructing the source estimates by sub-indexing CNMF parameters that depend on the class index l for $l = 1, ..., L$.

6. The class to component correspondence matrix y_{lk} was also configured in such a way as to pre-determine and assign subsets of source components indexed by k to necessarily be associated with a particular class l. For instance, and within the next chapter it will be demonstrated that $K = 24$ total source components were allocated and 8 per class l were used to construct source estimates. It was found that this was an sufficient assignment for the purpose of adequate source separation performance.

7. We proved that the auxiliary function and multiplicative updates were derived on the basis of the majorization-minimization algorithm, as well as a Jensen's inequality based justification for the auxiliary function being an upper bound to the primary cost function. The formulation of the auxiliary function for the algorithm most closely follow that which was detailed in [24, 25] for Euclidean distance based derivations.

8. By sequentially apply CNMF multiplicative updates iteratively *after* having done a K-means clustering analysis we believe that we have proposed a novel technique for multichannel source separation in the STFT domain that we believe should be benchmarked in terms of performance against existing algorithms.

Therefore, based upon the material developed within the current chapter and the preceding chapters, we will outline the method to be used for testing the proposed algorithm's performance. We will furthermore provide graphical results and analysis surrounding them that in fact best illustrate what the algorithm is doing in practice, and not just conceptually. We will the benchmark the proposed algorithm against the state of the art DoA algorithm as detailed in [24].

Chapter 7

Algorithm Performance and Benchmarks

In this chapter we will intend to study the performance and time-frequency domain behaviour of the proposed CNMF algorithm iterated for a total of 100 CNMF iterations.

Once we have demonstrated the performance of the proposed algorithm, we then would like to revisit the reference algorithm, chosen to be DoA SCM algorithm, as outlined in section 5.3.4 and within [24]. We will intend to show the performance of each respective algorithm within separate sections, but applied to the same test case of spatially occurring mixtures of non-overlapping (in either time, nor spatially in terms of look directions) musical sources corresponding to mixtures of three distinct instruments (piano, violin, guitar) playing two notes in total over a duration of roughly 20 seconds.

In order to see if the DoA SCM NMF algorithm is comparable, we will then also study the behaviour of the DoA SCM NMF algorithm also iterated for 100 CNMF iterations and thus we will have two sets of converged CNMF parameters to compare to each other, although both algorithm's CNMF decomposition are not identical since the proposed algorithm as described in the previous chapter primarily on describing a parametrization of $\mathbf{x}_{fn}$ adequately where as the DoA SCM NMF algorithm focuses on describing a parametrization of $\mathbf{X}_{fn}$ adequately. Appendix section D.2 demonstrates and explains the result of implementing the DOA SCM NMF with test case 1 (musical sources) and the reader is advised to consider it at some point.

Another difference was that the DoA SCM NMF algorithm obtained source estimates in the form of spatial image vectors $\mathbf{y}_{fn}^{(l)} \in \mathbb{C}^{M\times 1}$ per time frequency bin (and for $l = 1, ..., L$ on the basis of a Wiener filtering like reconstruction step, where as the proposed algorithm attempted to model the spatial image vectors $\mathbf{y}_{fn}^{(l)} \in \mathbb{C}^{M\times 1}$ by embedding a full representation of all their possible characteristics within the CNMF model itself.

And therefore, no Wiener filter reconstruction is needed for the proposed algorithm, and either source estimates or spatial image estimates can be obtained by directly sub-indexing the model as specified by equation 6.40 for $l = 1, ..., L$ and using the various indicator

matrices $\mathbf{Y}$ and $\mathbf{Z}$ for summing over component and look direction bins corresponding to $k = 1, ..., K$ and $o = 1, ..., O$. This is how the spatial images and/or source estimates for the proposed algorithm applied to test case 1 were obtained within the current chapter's results, in fact.

As discussed in the previous chapter it was alluded to, the difference in Wiener filtering post processing reconstruction vs not Wiener filtering, is that this somewhat presents implications and perhaps differences between the two algorithms terms of how to interpret initial (i.e. absolute) phase characteristics of the parametrized quantities $\hat{\mathbf{x}}_{fn}$ and $\hat{\mathbf{X}}_{fn}$ which attempt to model the effect of $\mathbf{x}_{fn}$ and $\mathbf{X}_{fn}$ in each time-frequency bin. By considering the results of this chapter, this should be better understood as compared to what could only be explained at the time in terms of a description or illustration, regarding these possible differences.

Reconsidering and focusing on the task of evaluating source separation performance, we should note that in considering that only one test case (the musical signals test cases) applied to both algorithms may not be enough to significantly and adequately test the capabilities of both algorithms, two more test cases were applied to both algorithms, namely:

- Another non-overlapping mixture of three sound sources (temporally non-overlapping), that consisted of two distinct males speakers as two out of the three sources signals, and the third source signal applied as a chainsaw signal (i.e. particularly loud and noisy signal).

- An *overlapping* mixture (temporally overlapping) of the same three sound sources, thus exhibiting a high degree of temporal and spectral contention in the time-frequency, multichannel STFT domain.

Interestingly enough, both the proposed and DoA SCM NMF faired comparably in all three test cases, with the DoA SCM NMF algorithm doing slightly better in most cases, unfortunately. According to the outputted scores for certain metrics, there could be reasons that signify that the proposed algorithms has a unique set of merits that are perhaps not present within the DoA algorithm. Since in total we included the results for 7 metrics tested within three 3 total test cases, it will be demonstrated that there exists enough data to begin to draw conclusions as to which algorithm performs better, both overall and with regards to specific areas.

7.1 Simulation Results for the Proposed Algorithm

7.1.1 Test Case 1: Non Overlapping Musical Notes

Here, we observe the converged result of 100 NMF iterations of the proposed algorithm. Figure 7.2 shows the converged result and Figure 7.3 shows the progression of the convergence. Plotted here are the absolute value of the multichannel spectrograms. Figure 7.2 corresponds to the absolute value of the the complex multichannel STFT model as described by equation 6.29. On the left, 7.1 shows the STFT of the observed multichannel spectrogram, and upon comparing it with the converged result, on the right, the reader may conclude that the model seems to have sufficiently converged towards its target. There do exist subtle differences where the magnitude spectrogram is not perfectly modelled, which may not be easily visible at the resolution at which the figures are shown.

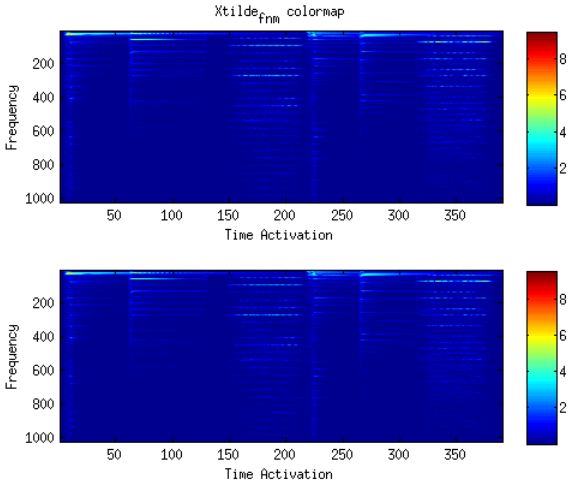

Figure 7.1: Magnitude of Target Stereo STFT

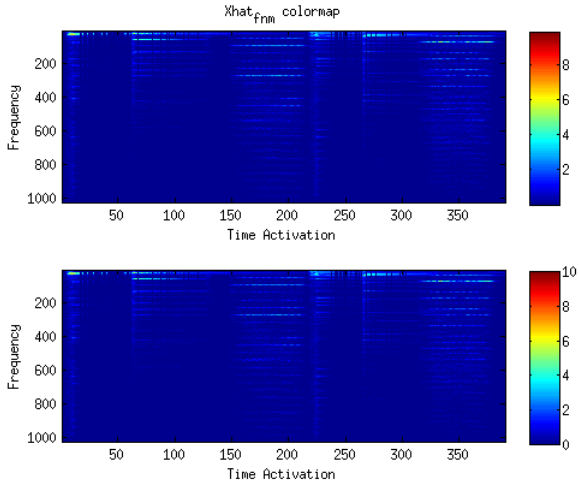

Figure 7.2: Magnitude of Modelled Stereo STFT

Considering then Figure 7.3 which corresponds to the model error, plotted over 100 CNMF iterations. Explained briefly, how this figure was generated can be understood as by wrapping a set of operations, per iteration, and then plotting how the cost function values (i.e. the error values) changed as a consequence of running the entire set of operations that were wrapped (i.e. after running all CNMF updates in each and all of the CNMF parameters within a single CNMF iteration). The reader is encouraged to review the set of CNMF operations to be computed per CNMF iteration, as described in Appendix section A.10. Understood then from a high level, and by inspecting the two figures it can be largely

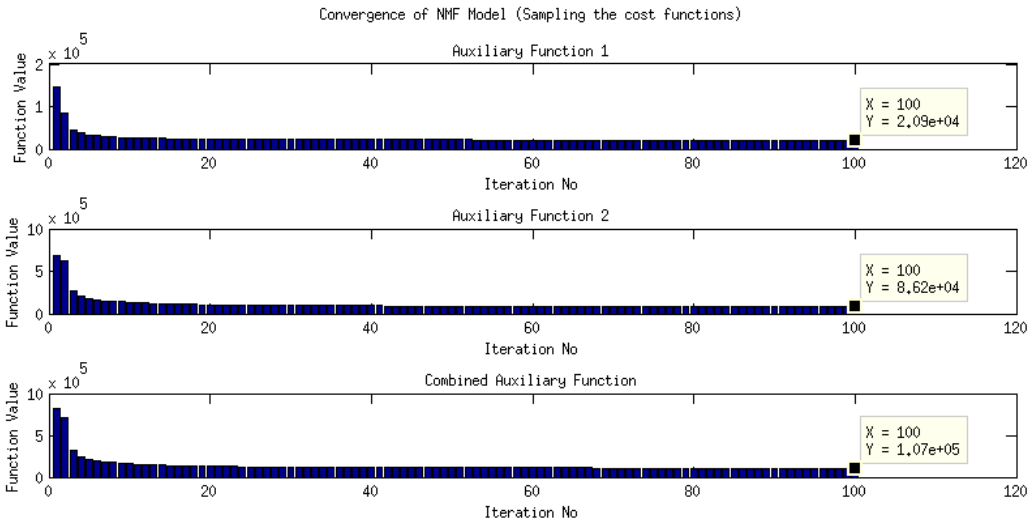

Figure 7.3: Minimization of the Two Objective Functions separately and combined

understood at this point that the proposed CNMF algorithm is behaving reasonably well as a result of being tested under test case 1.

We now present a set of figures showing the parametrization of converged CNMF parameters having iterated the proposed algorithm for 100 iterations.

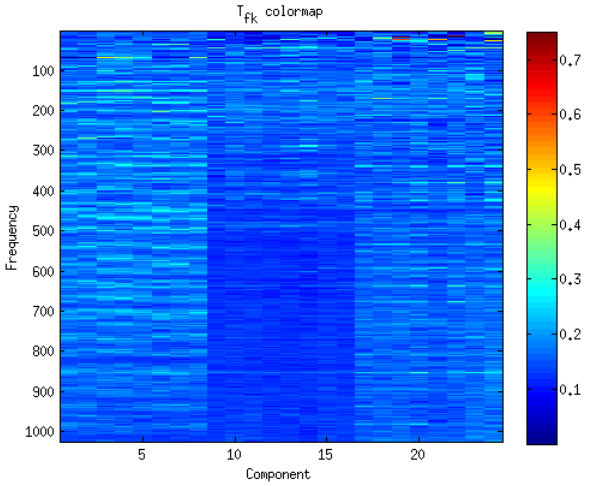

Figure 7.4: Frequency Template Dictionary Matrix t_{fk}

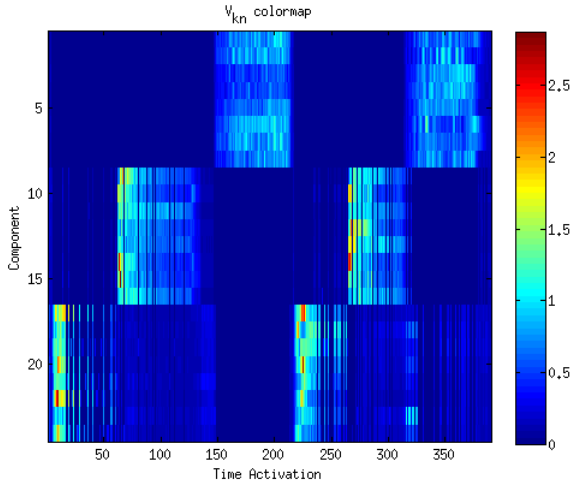

Figure 7.5: Time Activation Matrix v_{kn}

Figure 7.4 shows the converged CNMF parameter t_{fk}.

For test case 1 (i.e. the current test case) the best result was obtained by enabling only the update rule specified by (6.50).

In terms of other test cases (different input signal data) we typically obtained good behaviour by enabling (6.50) and sometimes optionally enabling (A.100).

Figure 7.5 shows the converged CNMF parameter v_{kn}.

For test case 1 the best result was obtained by enabling only the update rule specified by (6.51).

In terms of other test cases we typically obtained good behaviour by enabling (6.51) and sometimes optionally enabling (A.101).

According to the demonstrated result, shown in Figure 7.5, it is shown that the actual activation matrix v_{kn} has found sparsity in the time activation regions of *disinterest* as correspondingly similar to the indicator matrix (class to activations indicator matrix) obtained in Figure 6.20.

As specified earlier, in this section we focus solely on explaining test case 1, which was also utilized as the primary test case that was focused on in explaining the development of the proposed algorithm in the previous chapter.

Next we observe the parametrization of the various dictionary NMF factor matrices as well as indicator factor matrices and we show that the interchannel phase and initial phases quantities that were hypothesized in a multichannel sense have sufficiently converged to provide a plausible source separation output. Looking back at the time-frequency NMF factors T and V, we can see for that for the activation matrix V, the activation vectors have converged to a result that is continuous in adjacent time-frequency bins.

For the harmonic dictionary matrix T we can note that the partitioning of the components into their output classes has provided a set of possible basis vectors per source that are used to additively combine as STFT spectra in order to construct the output sources per class.

Figure 7.6 shows the converged CNMF parameter w_{fom}.

Figures 7.7 and 7.8 show the converged CNMF parameters z_{ol} and y_{lk}, respectively. z_{ol} (obtained from the output of pre-clustering) depicts the optimal correspondence between the most significant look direction per output class. y_{lk} depicts the partitioning of each set of components to each particular output class.

Again we emphasize that we can note from the $\mathbf{Z}$ matrix that an optimal correspondence has been obtained via the pre-clustering step that was outline in section 6.5.1 . This correspondence extends to the partition that was created and associates an optimal look direction not only to a class but to all components within that class.

This is best seen by considering the connection between $\mathbf{Z}$, $\mathbf{Y}$, and $\mathbf{W}$, in combination. A conceptual explanation was provided in section 6.5 however in that section we omitted the inclusion of the W parameter for simplicity, since the $\mathbf{W}$ parameter represents a tensor

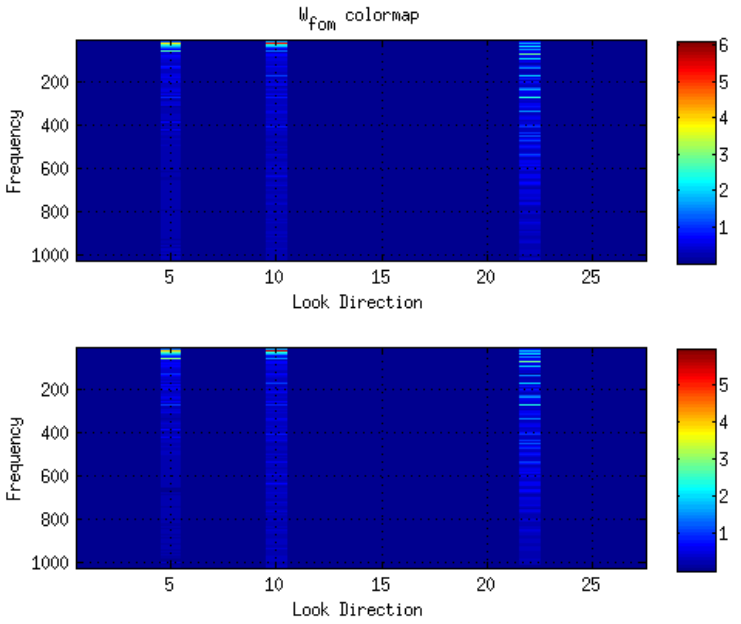

Figure 7.6: Amplitude of Channel Mixing Tensor Parameter w_{fom}

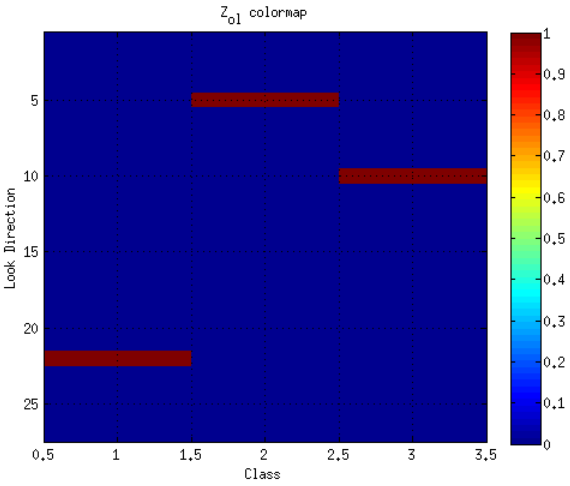

Figure 7.7: Look Direction to Class Indicator Matrix z_{ol}

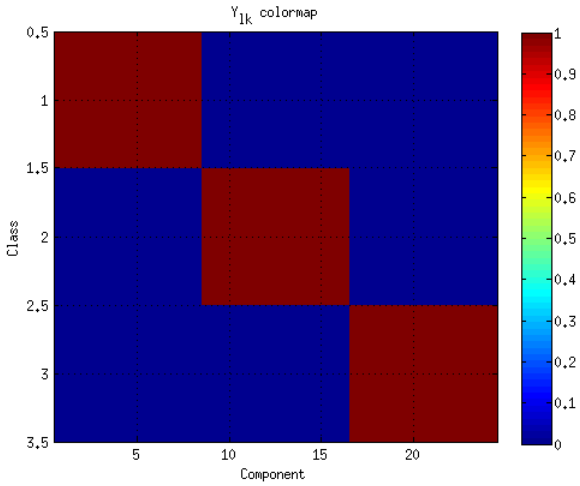

Figure 7.8: Component to Class Indicator/Partition Matrix y_{lk}

unlike the nonnegative parameters $\mathbf{Z}$, $\mathbf{Y}$.

Furthermore, we omit in this section the plotted result for the $e^{j\Phi_S(f,n,k)}$ parameter, however we propose that it has converged successfully on the basis of its monotonic convergence graph and the fact that the interchannel pattern (interchannel phase between b and a'th channel) has converged successfully towards its target. We noted that when the $e^{j\Phi_S(f,n,k)}$ parameter and its update had not yet been adequately iterated, that the interchannel phase quantity differed fairly noticeably from its target, and thus we conclude that there exists

an inter-dependence of parameters between resolving the interchannel phase simultaneously with the $e^{j\Phi_S(f,n,k)}$ parameter.

Instead we focus on comparing Figure 7.9 which is occurs due to the convergence of the proposed CNMF model to the reference quantity shown in Figure 6.18. In thesimulation, it can be shown that whenever the parameter $e^{j\Phi_S(f,n,k)}$ does not converge appropriately, then we do not in fact obtain the nice result for the modelled interchannel phase difference between microphones channels 1 and 2 as shown in Figure 7.9.

The most evident issue with the converged result that can be observed by comparing the two figures (7.9 and 6.18) is such that the converged result, Figure 7.9, suffers from being unable to model the region corresponding to look direction 14 (which is according to Figure 6.2 physically located closest to the guitar source), where the first guitar note and sound source is present at the beginning and begins to decay. This would correspond to the fact that for look direction 14 the corresponding sound source is spatially located in such a way that is it equidistant to the two microphones and thus corresponds to equal time differences (an interchannel phase difference quantity of 0). After the guitar signal begins to decay, corresponding to time activation index of approximately 20, the interchannel phase difference is much less prominent, and the proposed algorithm encounters difficulty in modelling this, as can be seen in figure 7.9 between time activation indices 20 and 50 approximately.

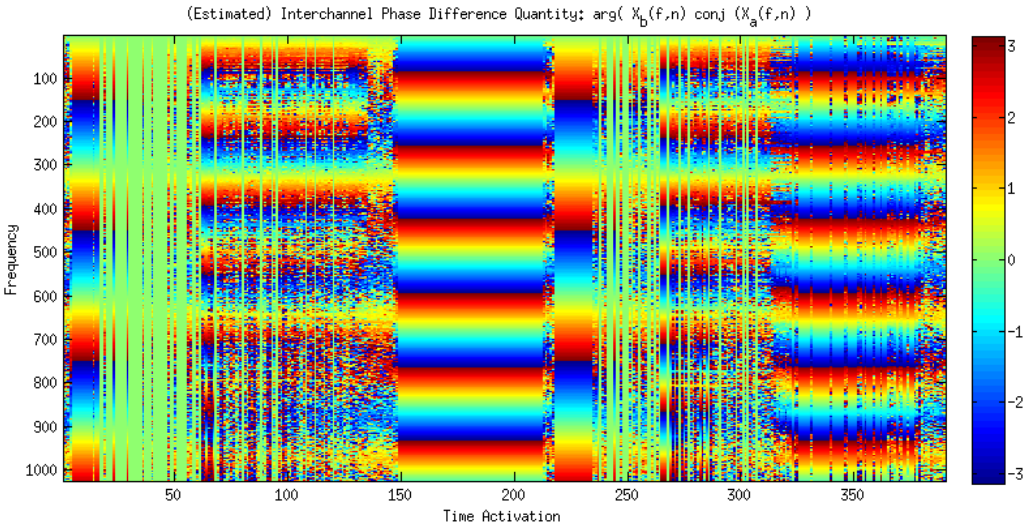

Figure 7.9: Interchannel Phase Difference Quantity (based on converged CNMF parameters)

CNMF parameter $e^{j\Phi_S(f,n,k)}$ which represents the phase dictionary and which we can point out is a parameter that is not in common with the DoA SCM reference algorithm, but is in fact in common with the single channel CNMF algorithm outlined in section 5.2.

We propose (it can be shown experimentally) that the update rule for the phase dictionary as specified according 6.52 in fact minimizes both cost functions, both individually and combined (additively) and thus represents a novel iterative learning rule for parametrizing phase spectra of source estimates in a multichannel CNMF sense.

Figures 7.10 to 7.15 demonstrate the parametrized source estimates in both the time domain and the short time Fourier transform domain.

Metrics will be applied in the sections that follow in order to computationally measure the similarity of the estimated and parametrized source signals against the true source signals that the CNMF algorithms have not had access to.

The results will be interesting to consider.

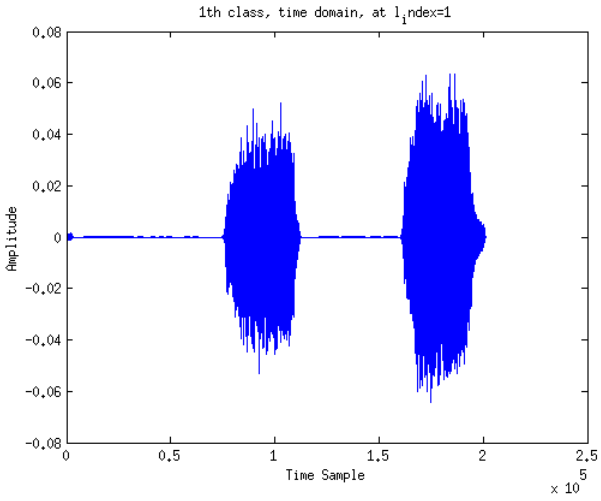

Figure 7.10: Separated Output Class: Violin Signal, Time Domain

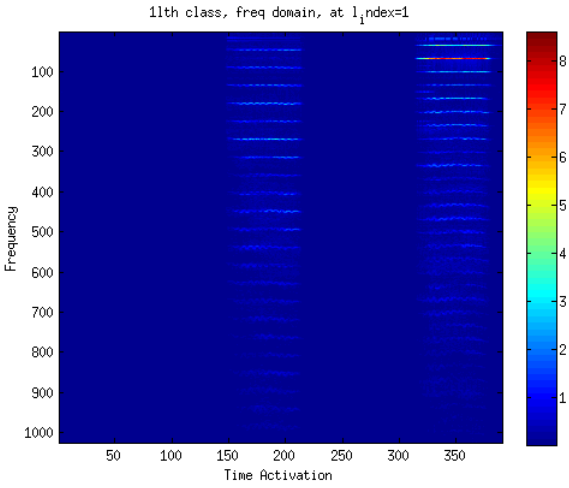

Figure 7.11: Separated Output Class: Violin Signal, Frequency Domain

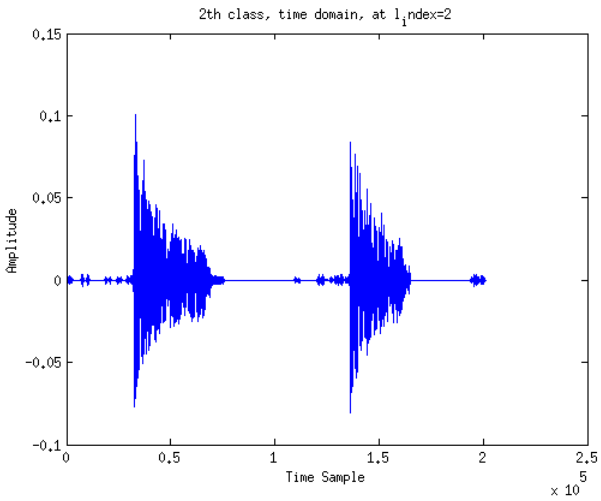

Figure 7.12: Separated Output Class: Piano Signal, Time Domain

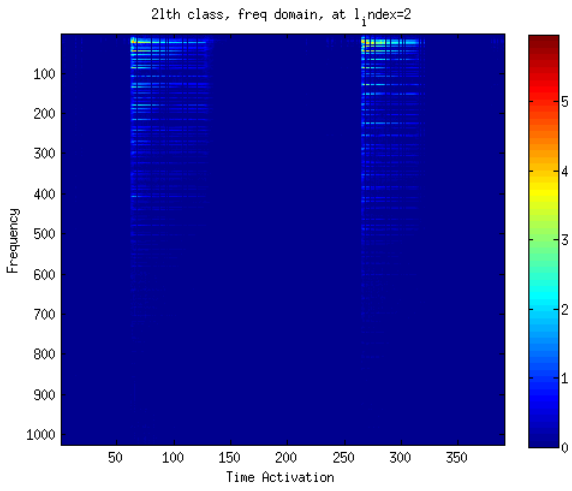

Figure 7.13: Separated Output Class: Piano Signal, Frequency Domain

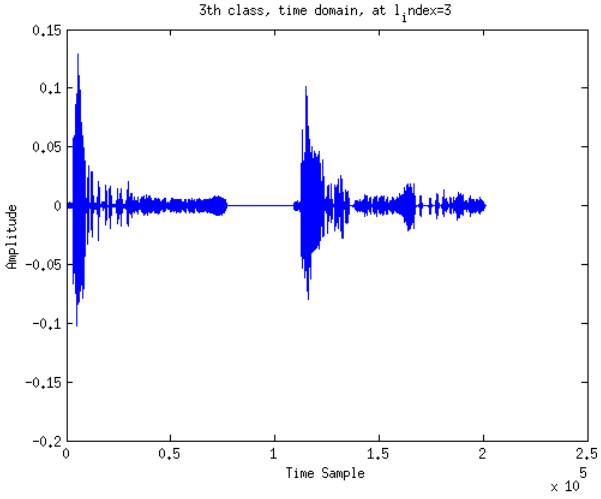

Figure 7.14: Separated Output Class: Guitar Signal, Time Domain

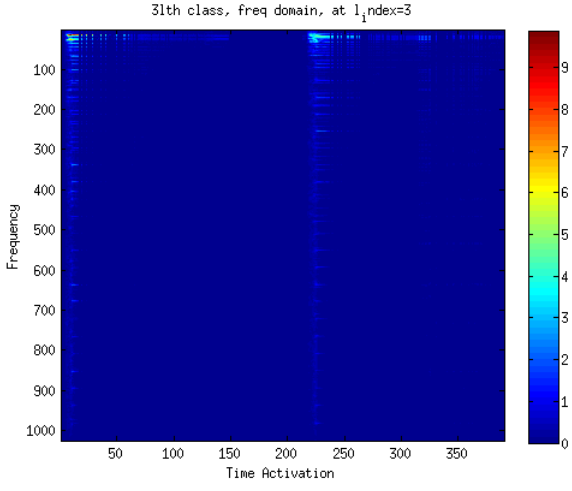

Figure 7.15: Separated Output Class: Guitar Signal, Frequency Domain

7.2 Benchmarking of Separation Quality and Separation Metrics

Figure 7.16 depicts an executive summary of the separation metrics discussed in the previous section utilized in comparing the proposed CNMF algorithm to the reference CNMF algorithm. Each listed value per separation metric corresponds to an average of three total test cases, thus comparing at a high level the overall performance of the proposed and reference CNMF algorithms, in terms of being able to reconstruct parametrized source signal estimates. Metrics were introduced within Appendix D.1. Appendices D.1 to D.3 further

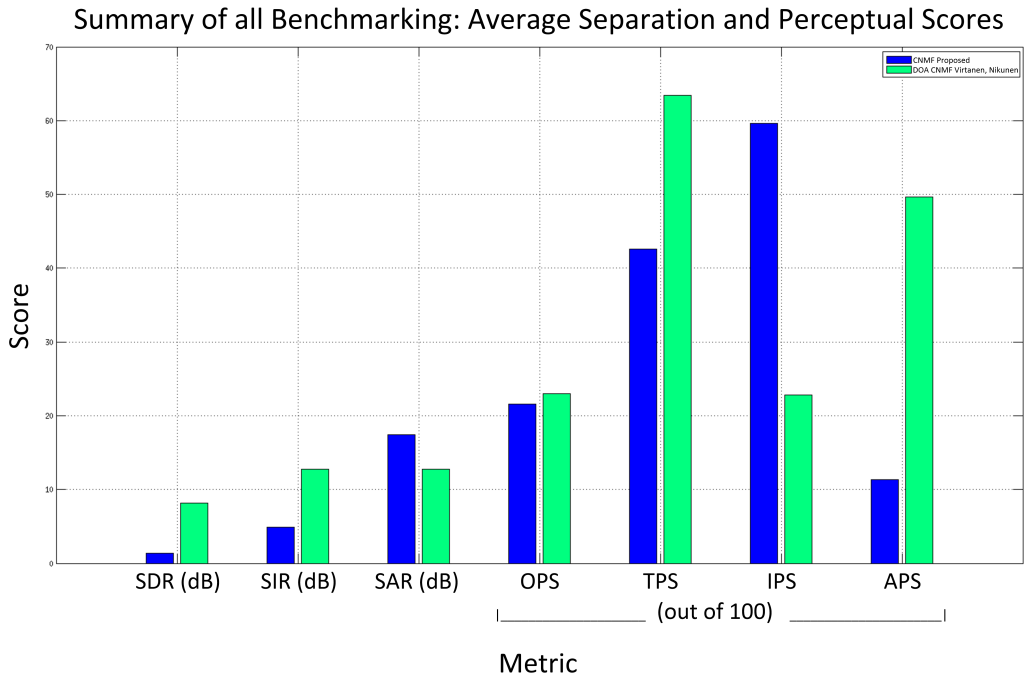

Figure 7.16: Executive Summary of Average Scores for all Metrics

detail how Figure 7.16 was computed as an average of three total test cases (9 total scores, a mixture of three signals was separated per test case) per metric and per algorithm. In the blue, the scores per metric for proposed algorithm are shown. In green, the scores for the DOA SCM NMF algorithm are shown. Each score is obtained by applying to the metric computation software a reference signal (i.e. a clean/true reference source signal) and a signal corresponding to the output of the algorithm under test (i.e. the estimated source signal). The SDR (Signal to Distortion) ratio for the proposed algorithm is particularly low, which is unfortunate, it outperforms the DOA SCM NMF algorithm in terms of SAR (Signal to Artifact) ratio as well as IPS (Interference-related perceptual score).

Chapter 8

Conclusions and Closing Remarks

In closing we review and highlight the development that the thesis has provided in each chapter. In Chapter 1 we emphasized that STFT matrices and their nonnegative and/complex STFT spectra (coefficients) could be considered with a high attention to detail in each time frequency bin of analysis, to provide a description into an amplitude dependent, phase dependent, frame dependent, and frequency dependent analysis of audio signals whose short-time behaviour is well modelled spectrally entirely via a single channel or multichannel complex observed STFT matrix dataset. We also introduced the basic NMF model as well as provided some introduction to its application outside of the area of audio STFT analysis. We discussed the requirements that NMF models typically have, in contrast to other existing matrix factorization methods, and whose presence within source separation and signal processing research may currently be more well recognized than NMF currently is. In chapter 2 we considered four useful subjects in both spatial and multichannel audio signal processing; spatial filtering according to the far field model, the frequency domain single channel Wiener filter, the MDVD beamformer, and the multichannel Wiener filter. In order to explain how NMF based parametrizations of multichannel time frequency audio algorithms are developed in practice we hinted that typically an advanced model based *decomposition* can be applied that must allow and furthermore, consistently ensure, that a certain data or matrix decomposition can adequately describe the observed data in the way that its parameters are used to compute an output matrix, and that the output matrix can be algorithmically *controlled* so that its representation *converges* towards that of a particular target, the observed data matrix. This concept was introduced in Chapter 3, where we introduced Jensen's inequality as an applicable tool to NMF based optimization problems. We then introduced the principle of majorization minimization (i.e. the auxiliary function approach) as an interpretation of iterative solutions to problem formulations requiring an NMF based solution. In chapter 4 we adequately considered the original NMF formulation and derivation by Lee and Seung.

We then connected it to treatment of the multichannel Wiener filter, covered in earlier chapters, and subsequently re-considered the significance of the multichannel Gaussian model, in order to consider how the multichannel Wiener filter model was extended to the concept of underdetermined source separation techniques that provide a point of emphasis to classes of algorithms known as spatial covariance matrix algorithms that occur in the complex and multichannel STFT domain. The effectiveness of this point of view can be confirmed by a set of existing algorithms [18–23] that seem to be motivated by acknowledging the covariance matrix method and the multichannel Wiener filtering model. The the next chapter we intended to focus on both the spatial covariance matrix analysis as well as another set of key concepts, namely complex NMF and time difference of arrival clustering, in addition to multichannel SCM based methods. Since within an existing SCM based processing method it was shown that TDoA (and DoA) information could be embedded within an SCM based NMF model [24], we intended to consider if the method could be improved by further studying it even more carefully. We provided an overview of its derivation as well as the derivation of an algorithm upon which it was based, [16] which was also subsequently an SCM based model and we noted that they were both derived upon the basis of a Euclidean distance based matrix similarity measure applied to covariance matrices in each time-frequency bin. Therefore we considered that it could be possible, if we focused on developing a similar but slightly modified parametrization of these CNMF SCM based methods that could extend the ideas developed by these two particular algorithms for applying clustering on the output of *spatially learned NMF features*. In the DoA algorithm, these were referred to as spatial weights. In the original Euclidean SCM algorithm, this was a subtle point, but was essentially that we should consider the complex argument (i.e. phase) of each SCM matrix in its off-diagonal element its an $M \times M$ SCM matrix as being similar on the basis of the its spatial properties with regards to other classes (i.e. other sources). These concepts in addition to clustering were addressed further in detail in Appendix C. Where as the Euclidean SCM algorithm introduced a set of cluster indicator latent variables to make itself flexible on the basis of re-ordering and re-labelling *NMF components* with similar spatial properties, the DoA SCM algorithm intended to simplify the complexity of the cluster processing, by introducing a set of pre-configured *DoA kernel matrices*. These DoA kernel matrices, in tandem with a set of newly parametrized nonnegative spatial weight parameters, were used to achieve a similar end goal of clustering learned NMF *basis components*, representing spectral audio features across *unlinked* groups of time-frequency regions, in such a way that optimally described an adequate overall *spatial* representation of source parametrizations, with the priority of output sources being *linked* on the basis of having similar spatial signatures. We thus carried these ideas into the development of the proposed algorithm in Chapter 6,

but started off with an approach somewhat outside of the typical SCM-based processing approach, in order to consider the effect of a *phase dictionary* introduced in order to model complex spectra of parametrized source estimates which we proposed was not a commonly considered part of SCM based parametrizations since it does not exactly fit the structure of the well known multichannel Wiener filter. Still, due to the existance of the complex NMF algorithm that was applied in a single channel scenario [25] we thought that this might be a worthwhile avenue to explore if an appropriate consideration of both SCM-based and CNMF based techniques were applied simultaneously. The CNMF representation of our proposed algorithm was demonstrated pictorially in Figure 6.7 but had some internal details that needed to be addressed, and therefore these details were addressed throughout Chapter 6, with the aid of various illustrations and explanations. Since Figure 6.7 strives to first explain the spatial image vectors per time frequency bin as opposed to the spatial covariance *matrices* per time frequency bin, we propose that this parametrization is the multichannel extension of the single channel CNMF algorithm, that was first derived in [25]. If we indeed grasp this concept well, then applying the SCM-processing perspective could be applied in order to further enhance the multichannel behaviour of the proposed multichannel CNMF algorithm, by considered such details as clustering, and SCM analysis of the observed and parametrized SCM matrix per time-frequency bin, which was what we intended to do, in tuning the proposed algorithm in such a way as to provide optimal performance.

8.1 Executive Summary of Proposed Novelty

Considering the relevant parts of chapters 5 and 6 that describe the implementation of the proposed algorithm vs. the reference algorithm, we can then highlight and contrast the algorithmic differences between the proposed algorithm and the reference (DOA SCM NMF) algorithm [24].

DOA SCM NMF algorithm

- As compared to the DOA SCM NMF algorithm, within the proposed algorithm we avoid normalizations of the CNMF parameters as described was to be necessary as within [24]. (Merit)
- As compared to the DOA SCM NMF algorithm, we avoid computing an eigenvalue decomposition on the CNMF parameter $\mathbf{W}_{fo}$. (Merit)
- Update equations can be considered more difficult implement for the DOA SCM algorithm, in having to deal with SCM matrices at each time frequency bin, computing

their eigenvalue decomposition, to ensure Hermitian positive semi-definiteness. The size of SCM matrices grows according to the square of the value of M. Merit.

- The component to look direction indicator matrix z_{ko} is updated as an NMF parameter. The quality of the source separation is limited to how well this parameter converges. And for instance the parameter z_{ko} may converge less well when the sources are more heavily overlapped in temporally and/or spectrally. Evidence of the limitation seems to be more apparent for test cases where the source signals are more heavily overlapped in time (e.g. heavily overlapping speech, test case 3). Source separation metrics aside, one noticeable issue in listening to outputted signals is that listening to separated signals for the DOA algorithm, there is evidence that shows that the DOA algorithm is working sub optimally (e.g. male speaker from unwanted source occurring more prominently in a separated output signal than compared with the proposed algorithm). For the same test case, and with the proposed algorithm, the unwanted signal component is more appropriately muted, when it should be, however, the desired (wanted) signal component comes through as sounding somewhat more distorted and or noisy (which could explain the overall lower score, as seen by the various different metrics). (Merit)

Proposed algorithm

- As compared to the DOA SCM NMF algorithm, we introduce the extra CNMF phase parameter $e^{j\Phi_S(f,n,k)}$, which increases the size of the overall CNMF algorithm parameter set. (Detriment)

- As compared to the DOA SCM NMF algorithm, it was found the better performance and stability occurred when it was chosen to not update indicator matrices z_{ol} and y_{lk}. (Merit)

- As compared to the DOA SCM NMF algorithm, an $M \times 1$ multichannel STFT *vector* per TF bin is parametrized, as opposed to an SCM matrix per TF bin. This difference can be seen to decrease the size of the overall CNMF algorithm parameter set size. The higher the value of M, the larger the difference will be, due to the fact that SCM matrices are of size $M \times M$. (Merit)

- As compared to the DOA SCM NMF algorithm, which implements equations 5.39 to 5.42, corresponding to only a single cost function, the proposed algorithm suggests two sets of update equations, corresponding to equations 6.47 to 6.52 and equations A.96 to A.101. Since we propose however not to update the parameters z_{ol} and y_{lk}, though for instances as well as to omit updating various other parameters in some instances,

this reduces the overall computational responsibility of the proposed algorithm. (Detriment)

- As compared to the DOA SCM NMF algorithm, when considering the interchannel phase difference, discussed surrounding Figures D.10 and 7.9; the proposed algorithm appears to surpass the capability of the DOA algorithm in its ability to model the interchannel phase difference between microphones 1 and 2. We emphasize that this in fact achieves what we intended for the proposed model to achieve. (Merit)

8.2 Executive Summary of Performance

Benchmarked against the DoA algorithm [24] treated as a baseline algorithm, we consider that although it performed adequately, it did not perform as exceedingly well as we had originally hoped it might. However, it performed adequately well enough, exceeding the performance of the DoA algorithm in *some* of the chosen benchmarking metrics, that further research into its possibly merits over the benchmark algorithm, could possibly be considered. The DoA algorithm algorithm performed significantly better in terms of the well known SDR metric, which most algorithms use as a starting point for considering the overall separation capability of a source separation algorithm. The test scenarios were set up in an underdetermined (more sources than microphone observations) configurations, and stereo mixtures of at most three simultaneous sources were applied as the two channel signals to be applied as input to each of the algorithms to be compared. Supplementary simulation results were included in section D when the input mixtures that were applied consisted of two male speakers and a chainsaw, which were signals that were provided by the authors of [24] in testing their proposed algorithm. The signals were thus downloaded and applied to the MCRoomSim software, in order to construct the mixture signals that were used and to be applied to our proposed algorithm in a test scenario.

A final set of highlights for our proposed algorithm vs the DoA benchmark algorithm can be considered in summary:

1. The proposed algorithm performed well in terms of the SAR metric as shown by considering Figure D.22.

2. The proposed algorithm performed well in terms of the IPS metric as shown by considering Figure D.28.

3. The proposed algorithm seems to technically exceed the performance of the DoA algorithm in principle when considering per time frequency bin the modelled output of the

quantity computed by taking the (unwrapped) complex argument of the off diagonal element of the parametrized spatial covariance matrix $\mathbf{X}_{fn}$. This can be viewed by first considering how well Figure 7.9 approximates Figure 6.18. It was our intention to achieve this output in such a way that also monotonically decreased the CNMF algorithm in all its parameters, a property of the proposed algorithm that was also demonstrated. If we then consider the result of the DoA algorithm as in Figures D.10 and D.9 respectively, it is made evident that the quantity is not seemingly modelled as well as for the proposed algorithm. We hypothesize that we have obtained this difference in adequately parametrizing our model by having introduced the update rule and parametrization of the phase dictionary parameter $e^{j\Phi_S(f,n,k)}$. Secondly we hypothesize that it could be due to our suggestion of K-means pre-processing as described in section 6.5.1.

4. We note that when speech sources were used the same differences in SDR, SAR, and IPS between the two algorithms were observed, and therefore these could signify key differences in algorithmic tendencies of each particular algorithm (i.e. the scores for test cases 2 and 3 were consistent with the scores for test case 1 when considering SDR, SAR, and IPS). This result can be observed by considering Figures D.18, D.22, and D.28 as evidence.

Therefore we hope that we have adequately featured the merits and limitations of the proposed algorithm and have adequately compared it against the existing state of the art DoA SCM NMF algorithm, in terms of both procedure and performance.

The short and long term goal of this research has been to make the case for NMF, CNMF, and SCM NMF algorithms as a novel and potentially unexplored, yet highly useful and meaningful set of tools for research in the area of audio signal processing.

Part of this was to be able to demonstrate its potential for solving seemingly difficult and partially intractable multivariate optimization problems in the STFT domain, for which the variables were contained within matrix and tensor quantities, and in some cases were restricted to be nonnegative and in some cases were allowed to be complex-valued. With a similar design philosophy set forth as that described in [24], in the early stages of conceiving and beginning to develop the algorithm, no requirement was necessarily set or defined to prioritize minimizing its computational cost as a crucial design parameter, but instead a continued and ongoing goal was always to consider improvements that would lead to achieving the highest possible correctness in terms of quality of separation, as perceived by a human listener, or end-user of the algorithm. One could even argue that given current hardware and software capabilities of today's computers, developing such algorithms is too complex and

requires too much effort for too little in return. Of course, it goes without saying that the purpose of this research was to argue the other side of the coin, and while still being developmental and a work in progress, per se, one would hope that individuals and researchers in the audio community as well as as well as the signal processing community will have something beneficial to take away from having considered it. We believe that the current performance of the proposed algorithm measures fairly well against some of the algorithms with comparable problem objectives and capabilities, and as demonstrated, although it clearly does not exceedingly surpass the state of the art reference algorithm [24], in every considered metric.

Where as the CNMF proposed by Virtanen and Nikunen is admittedly computationally expensive due to the number of DoA kernels used, the thesis's proposed algorithm is admittedly expensive due to the size of the phase dictionary matrix $e^{j\Phi_S(f,n,k)}$. Convergence of the algorithm, defined in terms of its primary and secondary cost functions, also seems to be good; as a particular configuration of the algorithm was found to be able to monotonically decrease the cost functions.

Future research directions in this field could include as a point of emphasis, investigating why the proposed algorithm performs poorly as seen by the SDR metric. Also a more streamlined approach to verifying in detail if absolute phases of reconstructed source spectra exactly align with absolute phases of the true source spectra (complex spectra, thus taking into consideration both amplitude and phase) analyzed in the STFT domain.

Lastly, we would like to thank the reader for their interest in having considered the proposed research and hope that it has invoked an increased sense of interest in the area of research that classifies the proposed algorithm as a source separation algorithm that is indeed worth studying.

List of References

[1] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, pp. 556–562, 2001.

[2] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[4] C. Bishop, "Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn," 2007.

[5] D. Liang, "Technical details about the expectation maximization (em) algorithm," 2015.

[6] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pp. 177–180, IEEE, 2003.

[7] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[8] A. Cichocki and S.-i. Amari, *Adaptive blind signal and image processing: learning algorithms and applications*, vol. 1. John Wiley & Sons, 2002.

[9] S. S. Haykin, *Unsupervised adaptive filtering: Blind source separation*, vol. 1. Wiley-Interscience, 2000.

[10] C. L. Byrne, *Signal Processing: a mathematical approach*. CRC Press, 2014.

[11] S. V. Vaseghi, *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[12] T. K. S. Moon and C. Wynn, *Mathematical methods and algorithms for signal processing*. No. 621.39: 51 MON, 2000.

[13] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[14] R. R. Curtin, J. R. Cline, N. P. Slagle, W. B. March, P. Ram, N. A. Mehta, and A. G. Gray, "Mlpack: A scalable c++ machine learning library," *Journal of Machine Learning Research*, vol. 14, no. Mar, pp. 801–805, 2013.

[15] V. Cherkassky and F. M. Mulier, *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.

[16] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Multichannel extensions of non-negative matrix factorization with complex-valued data," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 971–982, 2013.

[17] I. J. Tashev, *Sound capture and processing: practical approaches*. John Wiley & Sons, 2009.

[18] S. Arberet, A. Ozerov, N. Q. Duong, E. Vincent, R. Gribonval, F. Bimbot, and P. Vandergheynst, "Nonnegative matrix factorization and spatial covariance model for under-determined reverberant audio source separation," in *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, pp. 1–4, IEEE, 2010.

[19] N. Q. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010.

[20] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, 2010.

[21] N. Q. Duong, E. Vincent, and R. Gribonval, "Spatial covariance models for under-determined reverberant audio source separation," in *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 129–132, IEEE, 2009.

[22] N. Q. Duong, E. Vincent, and R. Gribonval, "Under-determined convolutive blind source separation using spatial covariance models," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 9–12, IEEE, 2010.

[23] M. Fakhry, P. Svaizer, and M. Omologo, "Reverberant audio source separation using partially pre-trained nonnegative matrix factorization," in *2014 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 273–277, Sept 2014.

[24] J. Nikunen and T. Virtanen, "Direction of arrival based spatial covariance model for blind sound source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 3, pp. 727–739, 2014.

[25] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, "Complex nmf: A new sparse representation for acoustic signals," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3437–3440, IEEE, 2009.

[26] W. Kellermann, "Beamforming for speech and audio signals," in *Handbook of signal processing in acoustics*, pp. 691–702, Springer, 2008.

[27] E. A. P. Habets, J. Benesty, I. Cohen, S. Gannot, and J. Dmochowski, "New insights into the mvdr beamformer in room acoustics," *IEEE transactions on audio, speech, and language processing*, vol. 18, no. 1, p. 158, 2010.

[28] P. Vary and R. Martin, *Digital speech transmission: Enhancement, coding and error concealment*. John Wiley & Sons, 2006.

[29] P. Magron, R. Badeau, and B. David, "Complex nmf under phase constraints based on signal modeling: application to audio source separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 46–50, IEEE, 2016.

[30] S. V. Vaseghi, *Multimedia signal processing: theory and applications in speech, music and communications*. John Wiley & Sons, 2007.

[31] H. Neudecker and J. R. Magnus, "Matrix differential calculus with applications in statistics and econometrics," 1988.

[32] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 2012.

[33] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, vol. 15. SIAM, 1995.

[34] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, pp. 67–68, 1999.

[35] A. Hjørungnes, *Complex-valued matrix derivatives: with applications in signal processing and communications*. Cambridge University Press, 2011.

[36] S. Makino, T.-W. Lee, and H. Sawada, *Blind speech separation*, vol. 615. Springer, 2007.

[37] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[38] S. Makino, H. Sawada, and S. Araki, "Frequency-domain blind source separation," in *Blind Speech Separation*, pp. 47–78, Springer, 2007.

[39] S. Winter, W. Kellermann, H. Sawada, and S. Makino, "Map-based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and l 1-norm minimization," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 81–81, 2007.

[40] D. R. Hunter and K. Lange, "A tutorial on mm algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[41] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[42] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[43] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, 2014.

[44] S. Ewert and M. Müller, "Using score-informed constraints for nmf-based source separation," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 129–132, IEEE, 2012.

[45] R. Hennequin, B. David, and R. Badeau, "Score informed audio source separation using a parametric model of non-negative spectrogram," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[46] S. Ewert, *Signal Processing Methods for Music Synchronization, Audio Matching, and Source Separation*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2012.

[47] R. Hennequin, R. Badeau, and B. David, "Time-dependent parametric and harmonic templates in non-negative matrix factorization," in *Proc. of the 13th International Conference on Digital Audio Effects (DAFx)*, 2010.

[48] S. Ewert, M. Müller, and M. Sandler, "Efficient data adaption for musical source separation methods based on parametric models," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 46–50, IEEE, 2013.

[49] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational statistics & data analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[50] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[51] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 126–135, ACM, 2006.

[52] C. Sanderson, "Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments," 2010.

[53] A. Wabnitz, N. Epain, C. Jin, and A. Van Schaik, "Room acoustics simulation for multichannel microphone arrays," in *Proceedings of the International Symposium on Room Acoustics*, pp. 1–6, 2010.

[54] N. Bertin, "Audio rendering, coding and separation source localization and separation,"

[55] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[56] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.

[57] L. Harry and V. Trees, "Optimum array processing: part iv of detection, estimation, and modulation theory," *New York: John Wiler & Sons*, 2002.

[58] Z. Kadelburg, D. Dukic, M. Lukic, and I. Matic, "Inequalities of karamata, schur and muirhead, and some applications," *The Teaching of Mathematics*, vol. 8, no. 1, pp. 31–45, 2005.

[59] A. W. Marshall and I. Olkin, "Theory of majorization and its applications," *Academic, New York*, vol. 16, pp. 4–93, 1979.

[60] F. Sha, Y. Lin, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming," *Neural Computation*, vol. 19, no. 8, pp. 2004–2031, 2007.

[61] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 538–549, 2010.

[62] B. J. King, *New methods of complex matrix factorization for single-channel source separation and analysis*. PhD thesis, 2013.

[63] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "New formulations and efficient algorithms for multichannel nmf," in *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 153–156, IEEE, 2011.

[64] A. Ozerov and C. Fevotte, "Multichannel nonnegative matrix factorization in convolutive mixtures. with application to blind audio source separation," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3137–3140, IEEE, 2009.

[65] T. P. Minka, "Old and new matrix algebra useful for statistics," *See www. stat. cmu. edu/minka/papers/matrix. html*, 2000.

[66] J. Traa, "Matrix calculus - notes on the derivative of a trace," *http://cal. cs. illinois. edu/ johannes/research/matrix calculus. pdf*.

[67] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[68] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, 2007.

[69] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.

[70] T. Adali and P. J. Schreier, "Optimization and estimation of complex-valued signals: Theory and applications in filtering and blind source separation," *IEEE Signal Processing Magazine*, vol. 5, no. 31, pp. 112–128, 2014.

[71] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab user guide," 2016.

[72] M. Frigo and S. G. Johnson, "The design and implementation of fftw3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[73] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Pearson Higher Education, 2010.

[74] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer handbook of speech processing*. Springer Science & Business Media, 2007.

[75] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.

[76] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–2057, 2011.

[77] E. Vincent, S. Araki, F. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Q. Duong, "The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges," *Signal Processing*, vol. 92, no. 8, pp. 1928–1936, 2012.

Appendix A

Practical and Essential Computational Considerations for Development of the Proposed Algorithm

Development, experimentation and testing of the Proposed Algorithm and other algorithms were primarily carried out in MATLAB code where possible, with computationally heavy bottleneck-like sections interfaced to MATLAB's MEX layer and from there coded with the Armadillo C++ Linear Algebra library.

In both cases, under the hood, the Intel Math Kernel Library, provided by the MATLAB distribution, provided the LAPACK and BLAS subroutines required to carry out computationally fast matrix multiplication and other fundamental linear algebra subroutines.

Interfacing to C and C++ allowed parallelization of parallelizable sections where MATLAB's available parallel computing library and sometimes already optimized or multithreaded subroutines were not of satisfaction in terms of flexibility and efficiency.

It is believed that CPU utilization during computation of bottleneck sections was maximized by a factor of the number of available CPU cores as compared to the single core case, by incorporating the pthread libraries available through the Linux Environment on Linux Mint 17. A build of Intel i7-4770k with 32 GB of RAM was used to run the proposed algorithm. On 20 seconds of 2-channel STFT audio data the memory utilization never exceeded 16GB of RAM. The high RAM utilization as shown by that figure was due in part to perhaps algorithm complexity but could also be bloated by a factor of 8 (the number of CPU cores) to take into consideration multiple copies of memory associated with threaded sections running on separate cores, allocated in order to allow computation to occur in parallel for higher CPU utilization and lower processing run-times.

Mixing pthreads with the Intel MKL layer through MATLAB did not seem to cause any issue, as functionally pthreads was used to assign the work and the MKL library seemed to

carry it out willingly without raising any odd or unsolvable exceptions.

A.1 Matrix Norms, Distances and Divergences

We now move on to consider tools that are more immediately and evidently useful to developing nonnegative matrix factorization algorithms. Namely, we consider various distances and/or diverges that are commonly used and encountered when considering how NMF based methods are typically derived.

We consider the general notion of distance/divergence $D_*(\mathbf{A}, \mathbf{B})$ defined upon *matrices* $\mathbf{A}$ and $\mathbf{B}$ both of size $I \times J$ which can be expressed as a sum of the element-wise distance/divergences as given by

$$D_*(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{I} \sum_{j=1}^{J} d_*(a_{ij}, b_{ij}) \tag{A.1}$$

where $d_*(a_{ij}, b_{ij})$ specifies one of several possible element-wise distance or divergences measures. We will be interested in using matrix distance/divergences, defined in this manner in order to begin to develop cost functions defined upon the matrices $\mathbf{A}$ and $\mathbf{B}$.

A.1.0.1 Squared Euclidean Distance

The squared Euclidean distance defined in terms of the matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{I \times J}$ is given by

$$D_{Eu}(\mathbf{A}, \mathbf{B}) = ||\mathbf{A} - \mathbf{B}||_F^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} |a_{ij} - b_{ij}|^2 \tag{A.2}$$

We note that it can be expressed as the square of the output of the Frobenius norm operator as specified in appendix section A.2.4, where the Frobenius norm represents a norm defined upon matrices. We further note that element-wise the distance is defined as given by

$$d_{Eu}(a_{ij}, b_{ij}) = |a_{ij} - b_{ij}|^2 \tag{A.3}$$

A.1.0.2 Kullback-Leibler Divergence

The Kullback-Leibler divergence defined in terms of the matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{I \times J}$ is given by

$$d_{KL}(a_{ij}, b_{ij}) = \left(a_{ij} \log \frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij} \right) \tag{A.4}$$

A.1.0.3 Itakura-Saito Divergence

The Itakura-Saito Divergence defined in terms of the matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{I \times J}$ is given by

$$d_{IS}(a_{ij}, b_{ij}) = \left(\frac{a_{ij}}{b_{ij}} - \log \frac{a_{ij}}{b_{ij}} - 1 \right) \tag{A.5}$$

A.1.0.4 Interpretation of element-wise distance/divergences with regards to NMF

In common with all three distances/divergences is the fact that if $a_{ij} = b_{ij}$ the distance/divergence achieves its minimum value which is in all three cases zero, under this condition. For being applied to NMF, we assume that both a_{ij} and b_{ij} will typically nonnegative real numbers, and that we will not immediately achieve $a_{ij} = b_{ij}$. Typically the matrix $\mathbf{A}$ will represent an *observed* data matrix and the matrix $\mathbf{B}$ will represent the *factorized matrix* which is a function of NMF parameters (i.e. some NMF *decomposition*. We note that if we suppose the element a_{ij} is fixed to some arbitrary number c along the real number line and compare it with b_{ij} surrounding c. As mentioned, if $b_{ij} = 0$ the all of the distances/divergences will achieve the minimum value of 0, but if $b_{ij} > c$ the distance/divergence's value increases, as is the case if $b_{ij} < c$. Therefore, the distances/divergences can be shown to be *convex* surrounding the point c and depending on the value of c at the i-j row-column of the matrix $\mathbf{A}$ or $\mathbf{B}$, since we treat the *observed* value of a_{ij} as being the *target* value and associate a *cost* with the factorized matrix output value b_{ij} have a large distance (on the real number line) from a_{ij} (we can assume $a_{ij} = c$ arbitarily). The concept here was illustrated visually within [16] and with regards to two factor NMF for describing nonnegative matrix spectrogram representations of audio..

A.2 Properties of Matrix and Tensor Algebra

In order to circumvent the computational overhead of rearranging matrix elements, we should concern ourselves with vectorization, which is a mathematical formalization of how matrix elements are typically stored in computer memory. In principle, this well known implementation convention should be the case in order to provide ease of access to basic operations such as matrix multiplication.

Familiarity with vectorization can be gained by experimenting with the MATLAB function 'reshape()'.

A.2.1 Indexing rows, columns and slices of a matrix or tensor quantity

A.2.1.1 Horizontal, Lateral and Frontal Slices

Although MATLAB provides automatic indexing of horizontal, lateral and frontal slices, we might guess that frontal accesses might naturally not require (the overhead of) rearranging of the array elements. The reason for this guess is that, for example, Armadillo's .slice() function implicitly corresponds to only to the "frontal" slice convention, and has no corresponding functions to return the "horizontal" and "lateral" slices as MATLAB does.

Therefore, computation of an NMF or NTF algorithm may require some initial planning as to choosing how the row, column, and slice indices are laid out, in order to achieve computation of the model and the algorithm updates in a desirable manner.

A.2.2 Outer Product

Outer product of real column vectors $\mathbf{a} \in \mathbb{R}^D$ $\mathbf{b} \in \mathbb{R}^M$

$$\mathbf{a} \circ \mathbf{b} \tag{A.6}$$

results in a rectangular matrix of size $D \times M$.

A.2.3 Kronecker Product

The Kronecker product of the matrices $\mathbf{A}$ and $\mathbf{B}$ is given by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{D1}\mathbf{B} & a_{D2}\mathbf{B} & \dots & a_{Dn}\mathbf{B} \end{bmatrix} \tag{A.7}$$

When $\mathbf{a} \in \mathbb{R}^D$ $\mathbf{b} \in \mathbb{R}^M$ as in section A.6 the Kronecker product of the quantity

$$\mathbf{a} \otimes \mathbf{b}^T = \mathbf{a} \circ \mathbf{b} = \begin{bmatrix} \mathbf{a}_1\mathbf{b}_1 & \mathbf{a}_1\mathbf{b}_2 & \dots & \mathbf{a}_1\mathbf{b}_M \\ \mathbf{a}_2\mathbf{b}_1 & \mathbf{a}_2\mathbf{b}_2 & \dots & \mathbf{a}_2\mathbf{b}_M \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_d\mathbf{b}_1 & \mathbf{a}_D\mathbf{b}_2 & \dots & \mathbf{a}_D\mathbf{b}_M \end{bmatrix} \in \mathbb{R}^{D \times M} \tag{A.8}$$

A.2.4 Frobenius Norm

We define the Frobenius norm of a matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ as given by [32]

$$||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \tag{A.9}$$

and where it can be expressed equivalently in terms of the trace operator as given by

$$||\mathbf{A}||_F = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^H)} \tag{A.10}$$

and where the matrix $\mathbf{A}^H$ represents the Hermitian transpose of the matrix $\mathbf{A}$

A.2.5 Matrices: Algebra, Derivatives, Calculus, and Index Notation

In this section, we come across the notion of the derivative of a matrix, or scalar quantity, with respect to a matrix, vector or scalar quantity, with the understanding that a matrix quantity is just an extension of a vector quantity, and can be seen as a rearrangement of a vector quantity. In many examples of scalar functions of matrix quantities, the $\mathrm{Tr}\,(\cdot)$ operator is involved in returning the scalar valued output.

[65] defines six possible kinds of derivatives summarized in the table below.

	Scalar	Vector	Matrix
Scalar	$\frac{dy}{dx}$	$\frac{d\mathbf{y}}{dx} = \left[\frac{\partial y}{\partial x_j}\right]$	$\frac{d\mathbf{Y}}{dx} = \left[\frac{\partial y_{ij}}{\partial x}\right]$
Vector	$\frac{dy}{d\mathbf{x}} = \left[\frac{\partial y}{\partial x_j}\right]$	$\frac{d\mathbf{y}}{d\mathbf{x}} = \left[\frac{\partial y_i}{\partial x_j}\right]$	
Matrix	$\frac{dy}{d\mathbf{X}} = \left[\frac{\partial y}{\partial x_{ji}}\right]$		

The following section will focus on derivatives of the kind described in the first column, and namely the third entry in the first column - the derivative of a scalar valued function with respect to a matrix quantity.

Note that in general, the functions arranged in the quantity $\mathbf{y}$ and the variables arranged in the quantity $\mathbf{x}$ need necessarily not be arranged in terms of matrices for computation and definition of the quantities $d\mathbf{y}$ and $d\mathbf{x}$, which are related to each other as described here below. Here, $\mathbf{x}$ and $\mathbf{y}$ are considered and allowed to be vectors of different arbitrary lengths:

$$\begin{aligned}\mathbf{y}(\mathbf{x}+d\mathbf{x}) &= \mathbf{y}(\mathbf{x}) + d\mathbf{y}(\mathbf{x}) \\ &= \mathbf{y}(\mathbf{x}) + \mathbf{A}d\mathbf{x}\end{aligned} \tag{A.11}$$

Here, $\mathbf{A}$ is called the derivative, and ubiquitously known in calculus as the Jacobian matrix $\mathbf{J}_{x\to y}$. Here we denote $\mathbf{y}$ a vector whose elements contain the functions whose partial derivatives we require. We also note that the equation specifies that the right hand side is characterized named by three quantities:

1. $\mathbf{x}$, the fixed point of the variable space. If we choose to denote the total number of variables as N, then we mean precisely that $\mathbf{x} \in \mathbb{R}^N$

2. $\mathbf{y}(\mathbf{x})$, denoted in shorthand as $\mathbf{y}$ is an M-dimensional vector $\in \mathbb{R}^M$, which contains the values of the M functions evaluated at the fixed point $\mathbf{x}$. Precisely $\mathbf{y}(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}^M$

3. We would like to know if we were to change $\mathbf{x}$ by a certain amount, by which we denote the difference $d\mathbf{x}$, by what amount will the vector $\mathbf{y}$ change, relative to its original location in its own $\mathbb{R}^M$ vector space. We note then that the second line of equation A.11 suggests that this difference here that we seek is captured exactly as a function of the Jacobian matrix $\mathbf{A}$, ie: the derivative (the multi-variable and in general, multi-function derivative).

[65] defines the the vector $d\mathbf{y}$, containing the differences that we seek to know, as being the part of $\mathbf{y}(\mathbf{x} + d\mathbf{x}) - \mathbf{y}(\mathbf{x})$ (the new value of the output vector less its old value) that is linear in $d\mathbf{x}$.

For the case that the vectors $\mathbf{x}$ and/or $\mathbf{y}$ reduce to either just x and/or y (scalars), then the matrix quantity $\mathbf{A}$ containing the partial derivatives of the variables in $\mathbf{x}$ with respect to the functions contained within $\mathbf{y}$ need not be arranged in terms of a matrix, but in general can be arranged as a vector quantity $\mathbf{a}$ as well (or even a scalar quantity a), as described by the table below:

$dy = adx$	$dy = \mathbf{a}dx$	$d\mathbf{Y} = \mathbf{A}dx$
$dy = \mathbf{a}d\mathbf{x}$	$d\mathbf{y} = \mathbf{A}d\mathbf{x}$	
$dy = tr(\mathbf{A}d\mathbf{X})$		

Canonical forms of differential expressions

For example, since there exists no entry in the table for computing the Jacobian $\mathbf{A}$ when the matrix of variables whose partials need to be computed $\mathbf{X}$ with respect to a vector of functions $\mathbf{y}$, for instance, the problem would then be vectorised by computing $d\mathbf{x} = d\,\mathrm{vec}\,\mathbf{X}$. And then consequently the dimensions of the problem to be solved as well as the Jacobian

matrix $\mathbf{A}$ would in turn be well defined according to the second entry of the second column of the above table.

Lastly, if we reduce the discussion now again to the first column of this table and the previous table, specifically to the third entry of the first column, then we note the use of the trace operator, $\mathrm{Tr}(\cdot)$. Here $\mathbf{X}$ is allowed to be non-square, with the requirement that $\mathbf{A}$ have the dimensions of its transpose. The differential dy, then, is obtained by taking the trace of the product of $\mathbf{A}$ and the matrix differential quantity of $\mathbf{X}$, $d\mathbf{X}$.

Listed below are a set of rules for computing the differentials of matrix expressions, and can be applied also in the case that 'x' is also a scalar or vector quantity x or $\mathbf{x}$, as well.

$$d\mathbf{C} = 0 \text{ (for constant}\mathbf{C}) \tag{A.12}$$
$$d(\alpha\mathbf{X}) = \alpha d\mathbf{X} \tag{A.13}$$
$$d(\mathbf{X} + \mathbf{W}) = d\mathbf{X} + d\mathbf{W} \tag{A.14}$$
$$d(\mathrm{Tr}\,(\mathbf{X})) = \mathrm{Tr}\,(d\mathbf{X}) \tag{A.15}$$
$$d(\mathbf{X}\mathbf{W}) = (d\mathbf{X})\mathbf{W} + \mathbf{X}d\mathbf{W} \tag{A.16}$$
$$d(\mathbf{X} \otimes \mathbf{W}) = (d\mathbf{X}) \otimes \mathbf{W} + \mathbf{X} \otimes d\mathbf{W} \tag{A.17}$$
$$d(\mathbf{X} \circ \mathbf{W}) = (d\mathbf{X}) \circ \mathbf{W} + \mathbf{X} \circ d\mathbf{W} \tag{A.18}$$
$$d(\mathbf{X}^{-1}) = -\mathbf{X}^{-1}(d\mathbf{X})\mathbf{X}^{-1} \tag{A.19}$$
$$d|\mathbf{X}| = |\mathbf{X}|\,\mathrm{Tr}\,(\mathbf{X}^{-1}d\mathbf{X}) \tag{A.20}$$
$$d\log|\mathbf{X}| = \mathrm{Tr}\,(\mathbf{X}^{-1}d\mathbf{X}) \tag{A.21}$$
$$\tag{A.22}$$

This set of rules can iteratively be used (in a chain rule - like fashion) to compute differential expressions, and then in combination with the second table, massaged into a canonical form that exactly specifies the resulting derivative quantity a, $\mathbf{a}$, or $\mathbf{A}$.

Lastly, we note that, in referring back to equation A.11, the transpose of $\mathbf{A}$, refers to the gradient of $\mathbf{y}$, denoted $\nabla\mathbf{y}$, which is often the nearly equivalent and required quantity in optimization problems.

A.2.5.1 More Properties and Some Examples

We return to make a point about the aforementioned trace operator, $\mathrm{Tr}\,(\cdot)$, which is that it is assumed to take as argument only matrices that are square matrices. Which means that for the trace of a multiplication of matrices, we should always check that the number of rows

in the leftmost matrix equals the number of columns in the rightmost matrix of the matrix multiplication.

Also, note two more nice properties of the trace operator, its invariance under cyclic permutations, and the trace of a transposed matrix:

1. $\mathrm{Tr}(\mathbf{ABCD}) = \mathrm{Tr}(\mathbf{BCDA}) = \mathrm{Tr}(\mathbf{CDAB}) = \mathrm{Tr}(\mathbf{DABC})$
2. $\mathrm{Tr}(\mathbf{A}^T) = \mathrm{Tr}(\mathbf{A})$

We now introduce a common and useful notation for indexing matrix elements which is useful when considering functions of the trace of a matrix and functions of other such related matrix operations.

Consider the matrix product $\mathbf{AB}$:

$$[\mathbf{AB}]_{ik} = \sum_j A_{ij} B_{jk}$$

Or the matrix product $\mathbf{ABC}$:

$$[\mathbf{ABC}]_{ii} = \sum_j A_{ij} [\mathbf{BC}]_{ji} = \sum_j \sum_k A_{ij} B_{jk} C_{ki}$$

Note that since the output matrix is square, its trace is well defined.

Example 1. Let us define $f = \mathrm{Tr}(\mathbf{AXB})$

Let us first compute $\frac{\partial f}{\partial \mathbf{X}} = \mathrm{Tr}(\mathbf{AXB})$ using index notation and then afterwards using the aforementioned differentiation rules.

Writing the expression using index notation:

$$f = \sum_i [\mathbf{AXB}]_{ii} = \sum_i \sum_j A_{ij} [\mathbf{XB}]_{ji} = \sum_i \sum_j A_{ij} \sum_k X_{jk} B_{ki} = \sum_i \sum_j \sum_k A_{ij} X_{jk} B_{ki}$$

Now it is easy to see that, by differentiation of the sum with respect to its j, kth element, we obtain:

$$\begin{aligned}\frac{\partial f}{\partial X_{jk}} &= \sum_i A_{ij} B_{ki} \\ &= [\mathbf{BA}]_{kj} \\ &= [\mathbf{A}^T \mathbf{B}^T]_{jk}\end{aligned}$$

Where the result is transposed in the final step so as to have the same dimensions as $\mathbf{X}$ had, originally.

Let us solve the same problem using differentiation rules. Note that as suggested in [65] $\frac{df}{d\mathbf{X}}$ should be laid out according to the transpose of $\mathbf{X}$. Therefore we should check that the final result that we obtain is $\in \mathbb{R}^{K \times J}$.

Computing the differential of the function f, we have:

$$\begin{aligned}d \operatorname{Tr}(\mathbf{AXB}) &= \operatorname{Tr}(\mathbf{A}(d\mathbf{X})\mathbf{B}) \\ &= \operatorname{Tr}(\mathbf{BA}d\mathbf{X}) \\ \frac{d}{d\mathbf{X}} \operatorname{Tr}(\mathbf{AXB}) &= \mathbf{BA}\end{aligned}$$

Which is the transpose of the result that we found earlier, as we expected.

Example 2. Let us define $f = \operatorname{Tr}(\mathbf{AX}^T\mathbf{BXC})$

Rewriting f using index notation, we have:

$$\begin{aligned}f &= \sum_i [\mathbf{AX}^T\mathbf{BXC}]_{ii} \\ &= \sum_i \sum_l \sum_h \sum_j \sum_k A_{il} X_{hl} B_{hj} X_{jk} C_{ki}\end{aligned} \tag{A.23}$$

Compute the partials with respect to both contributions of X, separately:

$$\frac{\partial f}{\partial X_{jk}} = \sum_i \sum_l \sum_h A_{il} X_{hl} B_{hj} C_{ki} = [\mathbf{CAX}^T\mathbf{B}]_{kj} = [(\mathbf{CAX}^T\mathbf{B})^T]_{jk}$$

$$\frac{\partial f}{\partial X_{hl}} = \sum_i \sum_j \sum_k A_{il} B_{hj} X_{jk} C_{ki} = [\mathbf{BXCA}]_{hl}$$

Note that: l was chosen as an alias for k and indexes a dimension of a matrix quantity of the same dimension.
And that likewise h was chosen as an alias for j and indexes a dimension of a matrix quantity of the same dimension.

Therefore we have that

$$\begin{aligned}\frac{\partial}{\partial \mathbf{X}} \operatorname{Tr}(\mathbf{AX}^T\mathbf{BXC}) &= (\mathbf{CAX}^T\mathbf{B})^T + (\mathbf{BXCA}) \\ &= (\mathbf{CAX}^T\mathbf{B})^T + (\mathbf{A}^T\mathbf{C}^T\mathbf{X}^T\mathbf{B}^T)^T\end{aligned}$$

Now we again try solving the problem using differentiation rules.
Computing the differential of the function f, we have:

$$\begin{aligned}d \operatorname{Tr}[\mathbf{AX}^T\mathbf{BXC}] &= \operatorname{Tr}[\mathbf{AX}^T\mathbf{B}(d\mathbf{X})\mathbf{C}] + \operatorname{Tr}[\mathbf{A}(d\mathbf{X})^T\mathbf{BXC}] \\ &= \operatorname{Tr}[\mathbf{AX}^T\mathbf{BC}(d\mathbf{X})] + \operatorname{Tr}[\mathbf{C}^T\mathbf{X}^T\mathbf{B}^T(d\mathbf{X})\mathbf{A}^T] \\ &= \operatorname{Tr}[\mathbf{CAX}^T\mathbf{B}(d\mathbf{X}) + \mathbf{A}^T\mathbf{C}^T\mathbf{X}^T\mathbf{B}^T(d\mathbf{X})] \\ \frac{d}{d\mathbf{X}} \operatorname{Tr}[\mathbf{AX}^T\mathbf{BXC}] &= \mathbf{CAX^TB} + \mathbf{A}^T\mathbf{C}^T\mathbf{X}^T\mathbf{B}^T\end{aligned}$$

Which is what we expected.

Example 3. Squared Frobenius norm:
Define

$$\begin{aligned}f &= ||\mathbf{X} - \mathbf{WH}||_F^2 = \operatorname{Tr}[(\mathbf{X} - \mathbf{WH})(\mathbf{X} - \mathbf{WH})^T] \\ &= \operatorname{Tr}[\mathbf{XX}^T] - \operatorname{Tr}[\mathbf{XH}^T\mathbf{W}^T] - \operatorname{Tr}[\mathbf{WHX}^T] + \operatorname{Tr}[\mathbf{WHH}^T\mathbf{W}^T]\end{aligned}$$

As before, compute the differential of the function with respect to $\mathbf{W}$ and re-arrange to obtain the derivative.

$$\begin{aligned} df &= -\operatorname{Tr}[\mathbf{X}\mathbf{H}^T(d\mathbf{W})^T] - \operatorname{Tr}[(d\mathbf{W})\mathbf{H}\mathbf{X}^T] + \operatorname{Tr}[(d\mathbf{W})\mathbf{H}\mathbf{H}^T\mathbf{W}^T + \mathbf{W}\mathbf{H}\mathbf{H}^T(d\mathbf{W})^T] \\ &= -\operatorname{Tr}[(d\mathbf{W})\mathbf{H}\mathbf{X}^T] - \operatorname{Tr}[(d\mathbf{W})\mathbf{H}\mathbf{X}^T] + \operatorname{Tr}[\mathbf{H}\mathbf{H}^T\mathbf{W}^T(d\mathbf{W}) + (d\mathbf{W})\mathbf{H}\mathbf{H}^T\mathbf{W}^T] \\ &= \operatorname{Tr}[(-\mathbf{H}\mathbf{X}^T - \mathbf{H}\mathbf{X}^T + \mathbf{H}\mathbf{H}^T\mathbf{W}^T + \mathbf{H}\mathbf{H}^T\mathbf{W}^T)(d\mathbf{W})] \end{aligned}$$

And therefore,

$$\frac{df}{d\mathbf{W}} = -2\mathbf{H}\mathbf{X}^T + 2\mathbf{H}\mathbf{H}^T\mathbf{W}^T$$

$$\frac{\partial f}{\partial \mathbf{W}} = \left[\frac{df}{d\mathbf{W}}\right]^T = -2\mathbf{X}\mathbf{H}^T + 2\mathbf{W}\mathbf{H}\mathbf{H}^T$$

We have thus shown only just three examples, however, hopefully these three will provide a motivation as to the reason that we should pay consideration to matrix derivatives. Many more examples with various applications and illustrations can be found in [65] and [66].

A.3 Optimization of NMF algorithms and analysis of convergence behaviour

The convergence properties of the NMF algorithm and its update rules were studied more in depth in [67, 68].

The formulation of the core mathematical optimization problem of the NMF model refers to the minimization of the criteron function as given by

$$\underset{\mathbf{H}}{\operatorname{argmin}}\ C(\mathbf{H}) = D(\mathbf{X}|\mathbf{W}\mathbf{H}) \qquad \text{subject to } \mathbf{H} \geq 0 \tag{A.24}$$

where $\mathbf{W}$ is considered to be fixed for the purpose of minimizing the criterion with respect to $\mathbf{H}$ for the current iteration and $\mathbf{X}$ plays the role of the observed matrix to be fitted. The described optimization problem can be applied to the minimization with respect to criterion $C(\mathbf{H})$ due to the symmetry of the factorization $\mathbf{X} \approx \mathbf{W}\mathbf{H}$ and the fact that $\mathbf{X}^T \approx \mathbf{H}^T\mathbf{W}^T$. Therefore, due to the symmetry, the result of the optimization of $C(\mathbf{H})$ could be applied to a scenario where the roles of $\mathbf{W}$ and $\mathbf{H}$ were interchanged.

Continuing with the original problem, finding a suitable representation for the matrix $\mathbf{H}$ separates into finding a set of vectors $\mathbf{h}$ that at each column index n of $\mathbf{H}$ solve the problem

$$\underset{\mathbf{h}}{\operatorname{argmin}}\ C(\mathbf{h}) = D(\mathbf{x}|\mathbf{W}\mathbf{h}) \qquad \text{subject to } \mathbf{h} \geq 0. \tag{A.25}$$

We note that in either case the optimization must be solved subject to nonnegativity constraints $\mathbf{H} \geq 0$ and $\mathbf{h} \geq 0$.

We also note that the gradient of the criterion, $\nabla_{\mathbf{h}} C(\mathbf{h}_*)$, can be computed as given by

$$\nabla_{\mathbf{h}} C(\mathbf{h}) = \mathbf{W}^T[(\mathbf{W}\mathbf{h})^{\cdot(\beta-2)}(\mathbf{W}\mathbf{h} - \mathbf{v})] \tag{A.26}$$

where the divergence measure used here corresponds to the β-divergence, which is defined as a generalized divergence that is capable of representing the IS, KL divergences and the Euclidean distance when β is set to $0, 1$, and 2, respectively [67].

In considering A.25, a requirement of the problem solution will satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions [34] such that any optimal solution $\mathbf{h}^*$ must satisfy in order to be considered an admissible solution. The conditions are given as

$$\nabla_{\mathbf{h}} C(\mathbf{h}^*).\mathbf{h}^* = 0 \tag{A.27}$$

$$\nabla_{\mathbf{h}} C(\mathbf{h}^*) \geq 0 \tag{A.28}$$

$$\mathbf{h}^* \geq 0 \tag{A.29}$$

which in total signify that the solution must be positive, its gradient must be positive, and the elementwise product between the solution and the gradient of the criterion $\nabla_{\mathbf{h}} C(\mathbf{h}^*)$ must all be zero, elementwise.

In discussing convergence and monotonicity of the associated learning rules associated with the gradient of the criterion described by equation A.26, it can be said that

- An algorithm is monotone if it produces a sequence of iterates $\{\mathbf{h}^{(i)}\}_{i\geq 0}$, such that $C(\mathbf{h}^{(i+1)}) \leq C(\mathbf{h}^{(i)})$ which signifies that the criterion is monotonically decreased at each subsequent iteration for all $i \geq 0$
- An algorithm is convergent if converges to a fixed point $\mathbf{h}^*$ satisfying the KKT conditions.

A.4 Low-Rank Modelling and Singular Value Decomposition

We now consider an important tool in the efficient modelling of representations of matrix data known as the Singular value decomposition (SVD). The SVD decomposition of a rank deficient matrix A is given by

$$A = \sum_{k=1}^{r} \sigma_k u_k \mathbf{v}_k. \tag{A.30}$$

The SVD as described by equation A.30 bears some resemblance to the basic NMF model as introduced in section 1.3.2.1 except for the fact that the SVD vectors u_k and $\mathbf{v}_k$ are here not restricted to being nonnegative as they are in the NMF model. Another important difference is that the NMF model has no notion of singular values σ_k.

$$\hat{A}_p = \sum_{k=1}^{r} \sigma_k u_k \mathbf{v}_k, \quad p < r \tag{A.31}$$

Alternatively we write the SVD of the rectangular matrix A as

$$A = U \Sigma V^H \tag{A.32}$$

where we generalize the elements of U and V to be complex valued as in [69] and Σ has the same size as A with nonnegative and real diagonal entries. The diagonal entries correspond to the singular values σ_k and represent the nonnegative square roots of the eigenvalues of $A^H A$ or AA^H depending on whether the matrix A is fat (has more columns than rows) or skinny (has more rows than columns).

There are various useful applications of decomposing A according to equation A.32.

For instance, by knowing the decomposition of A in terms of U, Σ, and V we can easily compute the pseudo inverse of A, denoted $A^\dagger$. Therefore, the pseudo inverse matrix $A^\dagger$ is given by

$$A^\dagger = V \Sigma^\dagger U^H \tag{A.33}$$

According to [69] the immediate value of $A^\dagger$ is that it can be used to solve a commonly occurring least squares optimization problem where A represents a transformation matrix with respect to an unknown vector x that is a column vector with the same number of elements as the number of columns in A.

The problem can be stated as

$$\underset{\mathbf{x}}{\operatorname{argmin}} ||b - Ax||_F^2 \tag{A.34}$$

where we can assume in general that $b \in \mathbb{C}^N$, $A \in \mathbb{C}^{N \times P}$ are known , and that we seek some vector $x \in \mathbb{C}^P$ that minimizes the vector quantity $b - Ax$ which applied to the Frobenius norm returns a scalar valued nonnegative output.

The behaviour of this problem depends upon the singularity and rank of the matrix $A^H A$.

The well known least squares solution says that the optimal x is given by

$$x = A^\dagger b \tag{A.35}$$

which shows that by knowing the SVD factorization/decomposition of the matrix A, the optimal solution to the problem expressed by A.34 can be computed easily according to equation A.33 and by applying the least squares solution as given by equation A.35.

Therefore we have immediately described the main usefulness of the SVD here.

So long as the vectors u_k and $\mathbf{v}_k$ as described in A.30 do not have the added *requirement* of being nonnegative, then an SVD interpretation of the minimization problem in A.34 (which shows up frequently) can often be considered adequate. However, if the low rank decomposition does in fact require a nonnegative interpretation of the matrix components, then an NMF formulation for a particular problem may be more appropriate.

Since the optimal x is typically defined as the vector x that has the least norm of all vectors x that minimize the function A.34 (we must typically assume that there could be more than one), if we somehow know that there should only be a single x that minimizes A.34 (by considering the rank of $A^H A$) then in practice $A^\dagger$ can be computed more efficiently by

$$A^\dagger = (A^H A)^{-1} A^H \tag{A.36}$$

rather by computing the SVD decomposition of A. It can be noted that this method can be applied for obtaining the optimal Wiener filter $\mathbf{w}$ in the Wiener filtering problem described by equation B.18. This was demonstrated within [30].

A.5 Complex Derivatives and Complex Gradient Updates

In the following section we briefly consider the subject of complex derivatives and a complex gradient descent update rule for real valued functions of complex valued variables.

We provide a treatment of the subject of functions of vectors matrices in section A.2.5 of the Appendix, but we focus here on the scenario where the vectors and matrices are necessarily *complex-valued*.

In real valued optimization problems it was explained that we typically require the gradient of a function, which specifies to us the direction of maximal first order ascent.

We seek an optimization rule that provides insight into how to compute learning rules that correspond to the direction of maximal first order change when the vector of variables is no longer real but complex-valued.

Here we present of a development of the notations necessary to consider the notion of a complex gradient for the purpose of considering optimization in the complex domain. In developing complex nonnegative matrix factorization algorithms, some of the matrix parameters will be, in consequence, complex-valued, and we may be interested in computing derivatives of a cost function with respect to each of the complex-valued variables.

We follow the detailed description of complex derivatives of real valued functions of complex variables as developed within [70, 71], but intend to highlight only the key and useful points for the purpose of utilizing the concepts for developing certain principles that may be of relevance in a CNMF optimization problem.

We will illustrate the use of the *conjugate* and non-conjugate co-gradients which represent partial derivatives that may be of relevance, and can be computed each respectively with respect to a function of complex variables $f(\mathbf{w}) : \mathbb{C}^{N\times 1} \rightarrow \mathbb{R}$ that maps a sequence of variables contained within the vector $\mathbf{w} \in \mathbb{C}^{N\times 1}$ to a real-valued function f. An example of the function f could be for instance the Frobenius norm operator which is capable of mapping complex valued vectors of the form $\mathbf{w} \in \mathbb{C}^{N\times 1}$ to a real valued output value.

We first begin by denoting the complex vector

$$\mathbf{w} \in \mathbb{C}^{N\times 1} \tag{A.37}$$

as the complex valued vector of N complex variables to be optimized. For reasons explained within the Appendix, section A.6, it is necessary to stack the vector $\mathbf{w}$ on top of its conjugate $\mathbf{w}^*$, and in doing so the following complex gradient descent update rule can be applied for optimizing $f(\mathbf{w})$ as a function of either $\mathbf{w}$ or its conjugate $\mathbf{w}^*$

$$\begin{bmatrix} \triangle\mathbf{w} \\ \triangle\mathbf{w}^* \end{bmatrix} = -2\mu \begin{bmatrix} \frac{\partial f}{\partial \mathbf{w}^*} \\ \frac{\partial f}{\partial \mathbf{w}} \end{bmatrix} \rightarrow \triangle\mathbf{w} = -2\mu \frac{\partial f}{\partial \mathbf{w}^*} \tag{A.38}$$

where $\triangle\mathbf{w}$ represents the update rule for the non-stacked vector $\mathbf{w}$. In specifying that

$\triangle \mathbf{w} = -2\mu \dfrac{\partial f}{\partial \mathbf{w}^*}$ represents a gradient update rule it can be alternatively interpreted to signify that the direction of maximal first order *ascent* is dependent on the term $\dfrac{\partial f}{\partial \mathbf{w}^*}$.

Here, we refer to the term $\dfrac{\partial f}{\partial \mathbf{w}^*}$ as the *conjugate co-gradient*, and it represents a partial derivative vector, corresponding to the partial derivative of $f(\mathbf{w})$ with respect to the conjugated (non-stacked) vector $\mathbf{w}^*$.

We can consider one brief example on the usage of complex gradients. If we begin by considering the function $f(\mathbf{w})$ as given by

$$f(\mathbf{w}) = \sin\left((\mathbf{w}^*)^T\mathbf{w} + (\mathbf{w}^* + \mathbf{w})^T\mathbf{a}\right) \tag{A.39}$$

We then have that computing the non-conjugate and conjugate co-gradients results in the terms

$$\frac{\partial f}{\partial \mathbf{w}} = \cos\left((\mathbf{w}^*)^T\mathbf{w} + (\mathbf{w}^* + \mathbf{w})^T\mathbf{a}\right) \cdot (\mathbf{w}^* + \mathbf{a}) \tag{A.40}$$

and

$$\frac{\partial f}{\partial \mathbf{w}^*} = \cos\left((\mathbf{w}^*)^T\mathbf{w} + (\mathbf{w}^* + \mathbf{w})^T\mathbf{a}\right) \cdot (\mathbf{w} + \mathbf{a}) \tag{A.41}$$

respectively. This here demonstrates that the non-conjugate co-gradient and the conjugate co-gradient as specified respectively by equations A.40 and A.41 correspond to complex-conjugates of one another, which happens to be the case for any real-valued function $f(\mathbf{w})$. This simple but useful result then becomes quite handy in practice, since it signifies that we only require the computation of one of the two co-gradients because either one can be computed as the conjugate of the other.

A.6 Complex Gradient (continued)

Continuing from within section A.5 in order to demonstrate details behind the development of a complex gradient update rule, we further consider equation A.37 by stating that the complex vector $\mathbf{w} \in \mathbb{C}^{N\times 1}$ can be expressed as a sum of its real and imaginary components as given by

$$\mathbf{w} = \mathbf{w}_r + j\mathbf{w}_i \tag{A.42}$$

where we then denote its real and imaginary parts respectively by $\mathbf{w}_r \in \mathbb{R}^{N\times 1}$ and $\mathbf{w}_i \in \mathbb{R}^{N\times 1}$ which are also both column vectors of size N but are each real-valued.

We further define two more vectors $\mathbf{w}_R$ and $\mathbf{w}_C$ the former is given by

$$\mathbf{w}_R = [\mathbf{w}_r^T, \mathbf{w}_i^T]^T \in \mathbb{R}^{2N\times 1} \tag{A.43}$$

where $\mathbf{w}_R$ is a $2N$ column vector obtained by stacking $\mathbf{w}_r$ on top of $\mathbf{w}_i$. Subsequently, the latter is given by

$$\mathbf{w}_C = [\mathbf{w}^T, \mathbf{w}^H]^T \in \mathbb{C}^{2N\times 1} \tag{A.44}$$

where $\mathbf{w}_C$ is a $2N$ complex valued column vector obtained by stacking $\mathbf{w}$ on top of its complex conjuate $\mathbf{w}^*$. Here $\mathbf{w}_C$ is termed the complex augmented vector.

Therefore the vector $\mathbf{w}_R$ is a vector containing all the real and imaginary parts of the vector $\mathbf{w}$ and the vector $\mathbf{w}_C$ is a vector containing the non conjugate and conjugate sequence of elements contained within $\mathbf{w}$.

The vector $\mathbf{w}_C$ provides obviously redundant information as that provided by $\mathbf{w}_R$ but its usage is more a matter of convenience for developing insight into complex derivatives of real valued functions.

The complex augmented vector $\mathbf{w}_C$ can in fact be considered to be related to the stacked vector $\mathbf{w}_R$ in one of two possible ways, the first of which is described by

$$\mathbf{w}_C = \mathbf{U}_N \mathbf{w}_R \tag{A.45}$$

and the second of which is given by

$$\mathbf{w}_R = \frac{1}{2}\mathbf{U}_N^H \mathbf{w}_C \tag{A.46}$$

where the matrix $\mathbf{U}_N$ is given by

$$\mathbf{U}_N = \begin{bmatrix} I & jI \\ I & -jI \end{bmatrix} \in \mathbb{C}^{2N\times 2N} \tag{A.47}$$

and is considered *unitary* up to a factor of 2, that is, $\mathbf{U}_N\mathbf{U}_N^H = \mathbf{U}_N^H\mathbf{U}_N = 2I$.

We then consider a function defined upon the complex sequence contained within the vector $\mathbf{w}$ to be denoted $f(\mathbf{w}) : \mathbb{C}^{N\times 1} \to \mathbb{R}$ that is at least first order differentiable. It is then possible to show that the stacked vectors $\mathbf{w}_R$ and $\mathbf{w}_C$ are useful in computing the partial derivatives of the function $f(\mathbf{w})$ to where

$$\frac{\partial f}{\partial \mathbf{w}_{\mathbb{R}}} = \mathbf{U}_N^H \frac{\partial f}{\partial \mathbf{w}_C^*} \tag{A.48}$$

is how the partial derivative of the function with respect $f(\mathbf{w})$ to the vector $\mathbf{w}_R$ can be computed.

A complex gradient descent update rule can then be expressed as now given by

$$\triangle \mathbf{w}_C = \mathbf{U}_N \triangle \mathbf{w}_{\mathbb{R}} = -\mu \mathbf{U}_N \frac{\partial f}{\partial \mathbf{w}_{\mathbb{R}}} = -2\mu \frac{\partial f}{\partial \mathbf{w}_C} \tag{A.49}$$

which can then be further simplified into the form that corresponded to equation A.38.

A.6.1 Auxiliary Function Derivation and Convergence Analysis

Convergence conditions for Auxiliary Functions Considering a similar reasoning as demonstrated in section 3.0.2, we define the following two necessary and sufficient conditions for a function $G(h, h^t)$ to be an auxiliary function of another function to be minimized, $F(h)$, where

1. $G(h, h^t \geq F(h)$
2. $G(h, h^t = F(h)$

and where the variable t specifies the iteration number for the algorithm. Furthermore, we rely again upon the following minimization rule which specifies the approach for choosing the $(t+1)$th value of h, that is, the optimal value for the next iterate, h^{t+1} as given by

$$h^{t+1} = \underset{h}{\operatorname{argmin}}\ G(h, h^t)$$

And it can be proven, that between the tth and $t+1$th iterations we have the following consequence as given by

$$F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h)$$

and thus by iterating subsequently according to the described update rule we obtain a sequence of estimates that, in their convergence, drives the fixed point h^t of the current iterate to a location that situates a local minimum of the objective function $h_{min} = \underset{h}{\operatorname{argmin}}\ F(h)$. This is described as given by

$$F(h_{min})...F(h^{t+1}) \leq F(h^t)... \leq F(h_2) \leq F(h_1) \leq F(h_0).$$

We intend then to derive a set of updates rules can be derived in a multiplicative update format, resulting from these considerations.

Euclidean Distance Auxiliary Function To be more explicit and to keep track of what matrix operations are valid, let us now specify a set of dimensions to the matrix multiplication defined earlier, where we had described

$$V \approx WH$$

and we shall now specify the that

1. $V \in \mathbb{R}^{I \times J}$, $W \in \mathbb{R}^{I \times K}$, $H \in \mathbb{R}^{K \times J}$

2. The variable t will be included as a superscript to h, when it could in fact be dropped. Note that it should not be confused in any way with the transpose operator, $(\cdot)^T$.

3. The variables a and b will be used to index anything having to do with K, so as to remain consistent with [1]. The variables i and μ will be used to index I and J, respectively.

We can then define the objective function to be defined, in terms of seeking an update rule for the quantity h^t, which corresponds to a column in the matrix H for some value of μ, which we implicitly drop; again so as to remain consistent with [1]. To be clear, $h^t \in \mathbb{R}^{K\times 1}$. The objective function to be minimized, then is

$$F(h) = \frac{1}{2}\sum_i (v_i - \sum_a W_{ia}h_a)^2 \tag{A.50}$$

which is simply a scalar valued objective function, and well defined since the indices i and a are summed over for $i = 1, ..., I$ and $a = 1, ..., K$. We can note again here, that, the vector $v \in \mathbb{R}^{I\times 1}$ also corresponds to the μth column in the matrix V, however the μ index is implied again and thus dropped.

Next, we claim that

$$G(h, h^t) = F(h^t) + (h - h^t)^T\nabla F(h^t) + \frac{1}{2}(h - h^t)^T K(h^t)(h - h^t)^T \tag{A.51}$$

is an auxiliary function for $F(h)$, where the constituent terms are to be described. Since it is nearly trivial to show that $G(h, h) = F(h)$, we need only show that $G(h, h^t) \geq F(h)$.

To begin to do this, let us first consider the following way of rewriting $f(h)$ as a quadratic approximation in terms of the function's first and second derivatives, and noting that is highly similar to the proposed auxiliary function $G(h, h^t)$ and is given by

$$F(h) = F(h^t) + (h - h^t)^T\nabla F(h^t) + \frac{1}{2}(h - h^t)^T(W^TW)(h - h^t)^T \tag{A.52}$$

and we then introduce the following diagonal matrix,

$$K_{ab}(h^t) = \delta_{ab}(W^TWh^t)_a/h^t_a = \text{diag}\left[(W^TWh)./h\right] \in \mathbb{R}^{K\times K} \tag{A.53}$$

and by comparing A.52 to A.51, we can see that the difference amounts to being a function of the quantities $K(h^t)$ and W^TW and that $G(h, h^t) \geq F(h)$ therefore depends on the inequality

$$0 \leq (h - h)^T[K(h^t) - W^TW](h - h) \tag{A.54}$$

which corresponds to a row vector, times a matrix, times a column vector multiplication, which outputs a scalar. We would like to know if the right hand side of the inequality is positive for any choice of $h - h^t$. Verifying this to be true would correspond to the statement that $K - W^T W$ is in fact a positive semi-definite matrix.

We follow the treatment of this task as described in [1] , by introducing the matrix

$$\begin{aligned} M_{ab}(h^t) &= h_a^t (K(h^t) - W^T W)_{ab} h_b^t \\ &= (K - W^T W) \circledast [(h^t \otimes (h^t)^T] \end{aligned} \tag{A.55}$$

which is a rescaling of the components of $K - W^T W$, and where we should note that we can use the operators $\circledast$ and $\otimes$ to describe an equivalent computation of the matrix $M \in \mathbb{R}^{K \times K}$, which correspond to the Hadamard and Kronecker product, respectively.

[1] suggests that proving M is a positive definite matrix is a necessary and sufficient condition for proving the positive definiteness of the matrix $K - W^T W$. We use the vector p to test the positivity of the scalar product

$$p^T M p = \sum_{ab} p_a M_{ab} p_b \tag{A.56}$$

$$= \sum_{ab} h_a^t (W^T W)_{ab} h_b^t p_a^2 - p_a h_a^t (W^T W)_{ab} h_b^t p_b \tag{A.57}$$

$$= \sum_{ab} (W^T W)_{ab} h_a^t h_b^t \Big[\frac{1}{2} p_a^2 + \frac{1}{2} p_b^2 - p_a p_b \Big] \tag{A.58}$$

$$= \sum_{ab} (W^T W)_{ab} h_a^t h_b^t (p_a - p_b)^2 \tag{A.59}$$

$$\geq 0 \tag{A.60}$$

where we can further check the equivalence between lines A.56 and A.57, which may be difficult to see if one is largely not familiar with the Kronecker delta function δ_{ab} , by reviewing its definition and then seeing how it is applied to the problem.

The Kronecker delta function we recall is defined as given by

$$\delta_{ab} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases}$$

Denoting R as the symmetric matrix $R = W^T W$, we can start by showing that the left

hand term in A.57 (which we recall is a function of $K(h^t)$) can be more explicitly expressed as given by

$$\begin{aligned} p^T\Big[K(h^t) \circledast [(h^t \otimes (h^t)^T]\Big]p &= \sum_{ab} p_a h_a^t K_{ab}(h^t) h_b^t p_b \\ &= \sum_{ab} \delta_{ab} \frac{(Rh^t)_a}{h_a^t} p_a h_a^t h_b^t p_b \\ &= \sum_{ab} \delta_{ab} (Rh^t)_a p_a h_b^t p_b \\ &= \sum_{ab} \Big[\operatorname{diag}(Rh^t) \circledast [p \otimes (h^t \circledast p)^T] \Big]_{ab} \\ &= \operatorname{Tr}\Big[\operatorname{diag}(Rh^t) \circledast [p \otimes (h^t \circledast p)^T] \Big] \end{aligned} \tag{A.61}$$

We emphasize again that showing these equations is a first step to showing equivalence between A.56 and A.57. Hopefully it is also clear that we focus primarily on the left hand term of A.57, since it depends on the aforementioned $K(h^t)$ matrix and the kronecker delta function δ_{ab}, which the right hand term does not. It can be seen that the left hand term can be expressed as a double summation over all elements of an $\in \mathbb{R}^{K \times K}$ diagonal matrix multiplied element-wise with a Kronecker product of the same dimensions. We note that elements that were subscripted with an a subscript index (as part of the double summation) are laid into the *columns* of the Kronecker product, where as elements subscripted with a b index are laid into the *rows*. And that since the sum of the off diagonal elements of the diagonal matrix are zero, the trace of the output matrix quantity is in this case equivalent to the double summation of the output matrix.

It is then possible to see the equivalance between A.56 and A.57 by comparing A.61 to the following expression

$$\begin{aligned} \sum_{ab} R_{ab} h_a h_b p_a p_a &= \sum_{ab} \Big[R \circledast [(p \circledast p \circledast h_t) \otimes (h_t)^T] \Big]_{ab} \\ &= \sum_{ab} \Big[R \circledast \Big(\frac{1}{2} [(p \circledast p \circledast h_t) \otimes (h_t)^T] + \frac{1}{2} [h_t \otimes (p \circledast p \circledast h_t)^T] \Big) \Big]_{ab} \\ &= \sum_{ab} \frac{1}{2} R_{ab} h_a h_b (p_a)^2 + \sum_{ab} \frac{1}{2} R_{ab} h_a h_b (p_b)^2 \end{aligned} \tag{A.62}$$

where, it happens to be true and can be verified that the right hand sides of A.61 and A.62 are in fact equivalent if we assume R to be symmetric, which we do. Note that A.62 is

not a function of a diagonal matrix as before, yet it is still equivalent to A.61. This can be understood to be possible if we consider summing across the columns of the matrix quantity in A.62 and compare it element-wise to the main diagonal of the matrix quantity in A.61. They can be verified to be the same element-wise.

In conclusion, A.59 can be shown to be true and thus $K - W^T W$ is proved to be a positive semi-definite matrix, as intended.

Finally, we can show the following update rule to be derived as a result of properties proven to be true to this point, namely that $G(h, h^t) \geq F(h, h^t)$, from which the following update rules

$$h_t = h_t - K(h^t)^{-1} \nabla F(h^t) \tag{A.63}$$

$$h_a^{t+1} = h_a^t \frac{(W^T v)_a}{(W^T W h^t)_a} \tag{A.64}$$

can be derived by minimizing the auxiliary function $G(h, h^t)$ with respect to h^t

It can be shown that a similar set of update rules for the quantity $w^t \in \mathbb{R}^{1 \times K}$ (which corresponds to the ith row of W at the current iteration t) can be derived. The proof is omitted and we later present just the multiplicative update rule itself.

Kullback Leibler Auxiliary Function In order to derive multiplicative update rules for the Kullback Leibler divergence, it will be proposed that

$$G(h, h^t) = \sum_i (v_i \log v_i - v_i) + \sum_{ia} W_{ia} h_a - \sum_{ia} v_i \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \left(\log W_{ia} h_a - \log \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \right) \tag{A.65}$$

is an auxiliary function for

$$\begin{aligned} F(h) &= \sum_i v_i \log \frac{v_i}{\sum_a W_{ia} h_a} - v_i + \sum_a W_{ia} h_a \\ &= \sum_i (v_i \log v_i) - \left(v_i \log \sum_a W_{ia} h_a \right) - v_i + \left(\sum_a W_{ia} h_a \right). \end{aligned} \tag{A.66}$$

and we intend to show that $G(h, h^t)$ evaluated at $h = h^t$ results in a tangency with $F(h^t)$, which can be shown by considering

$$
\begin{aligned}
G(h^t, h^t) &= \sum_i (v_i \log v_i - v_i) + \sum_{ia} W_{ia} h_a^t - \sum_{ia} v_i \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \left(\log W_{ia} h_a^t - \log \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \right) \\
&= \sum_i (v_i \log v_i - v_i) + \sum_{ia} W_{ia} h_a^t - \sum_i v_i \frac{\sum_a W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \left(- \log \frac{1}{\sum_b W_{ib} h_b^t} \right) \\
&= \sum_i (v_i \log v_i) - v_i + \sum_{ia} W_{ia} h_a^t + \sum_i v_i \left(\log \frac{1}{\sum_b W_{ib} h_b^t} \right) \\
&= \sum_i \left[(v_i \log v_i) - \left(v_i \log \sum_a W_{ia} h_a^t \right) - v_i \right] + \sum_{ia} W_{ia} h_a^t \\
&= F(h^t). \qquad \text{(A.67)}
\end{aligned}
$$

Let us show then the following inequality that will help us to prove that $G(h, h^t) \geq F(h)$ and and that $G(h, h^t)$ can be confirmed to be a true auxiliary function for the objective function $F(h)$ by considering that

$$
\begin{aligned}
- \log \sum_a W_{ia} h_a &\leq - \sum_a \alpha_a \log \frac{W_{ia} h_a}{\alpha_a} \\
- \log \sum_a W_{ia} h_a &\leq - \sum_a \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \Big(\log W_{ia} h_a - \log \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \Big)
\end{aligned} \qquad \text{(A.68)}
$$

and by applying Jensen's inequality and due to the convexity of the *negative* of the $\log(\cdot)$ function the $\log(\cdot)$ can be moved inside the summation, and furthermore upper bounds the left hand side of the inequality. The α_a are made to be nonnegative and to conform to the condition of summing to unity by choosing them as a function of the current parameter estimate h_t as given by

$$
\alpha_a = \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t}
$$

A.7 NTF Models

In order to gain some familiarity with the concepts presented, let us examine the PARAFAC, NTF1 and NTF2 Models as described in Cichocki et al. [7].

A.7.1 Three way PARAFAC Model

$$y_{itq} = \sum_{j=1}^{J} a_{ij} b_{tj} c_{qj} + e_{itq} \tag{A.69}$$

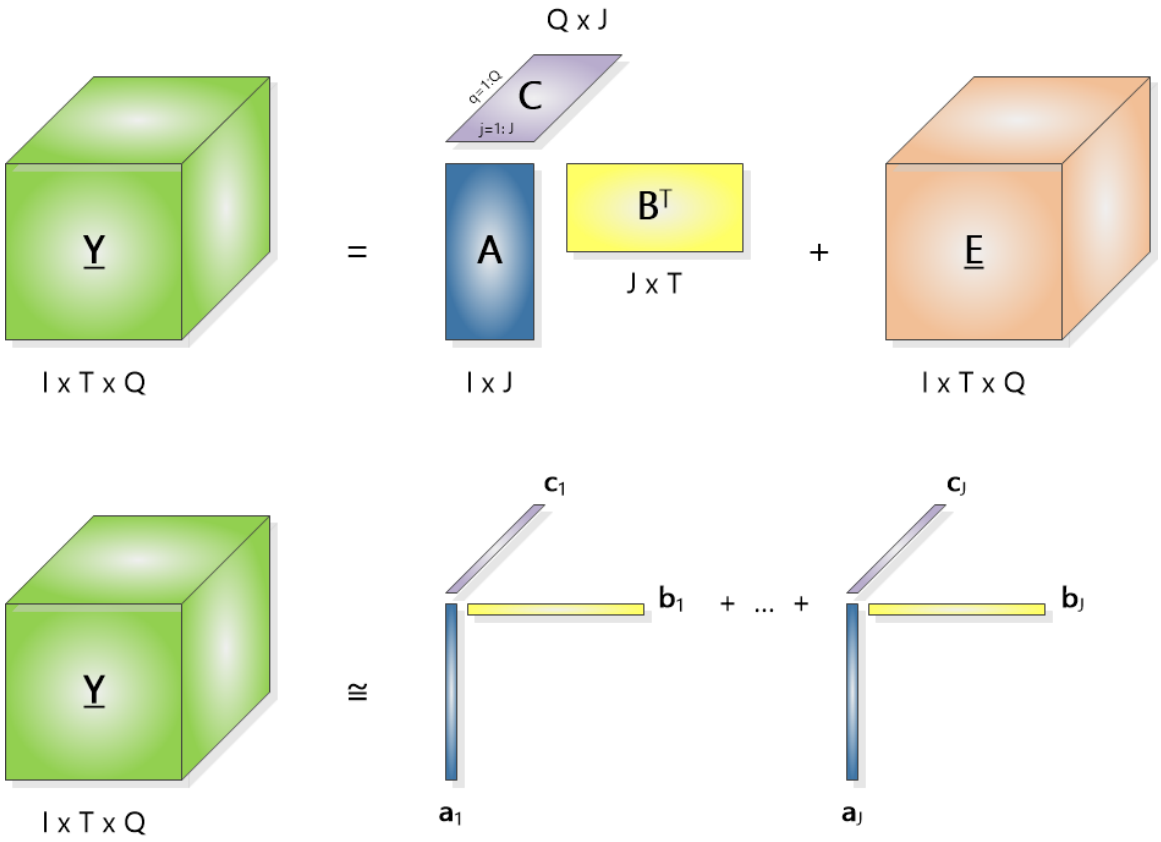

Figure A.1: Three Way PARAFAC Model

A.7.2 NTF1

$$y_{itq} = \sum_{j=1}^{J} a_{ij} b_{tjq} c_{qj} + e_{itq} \tag{A.70}$$

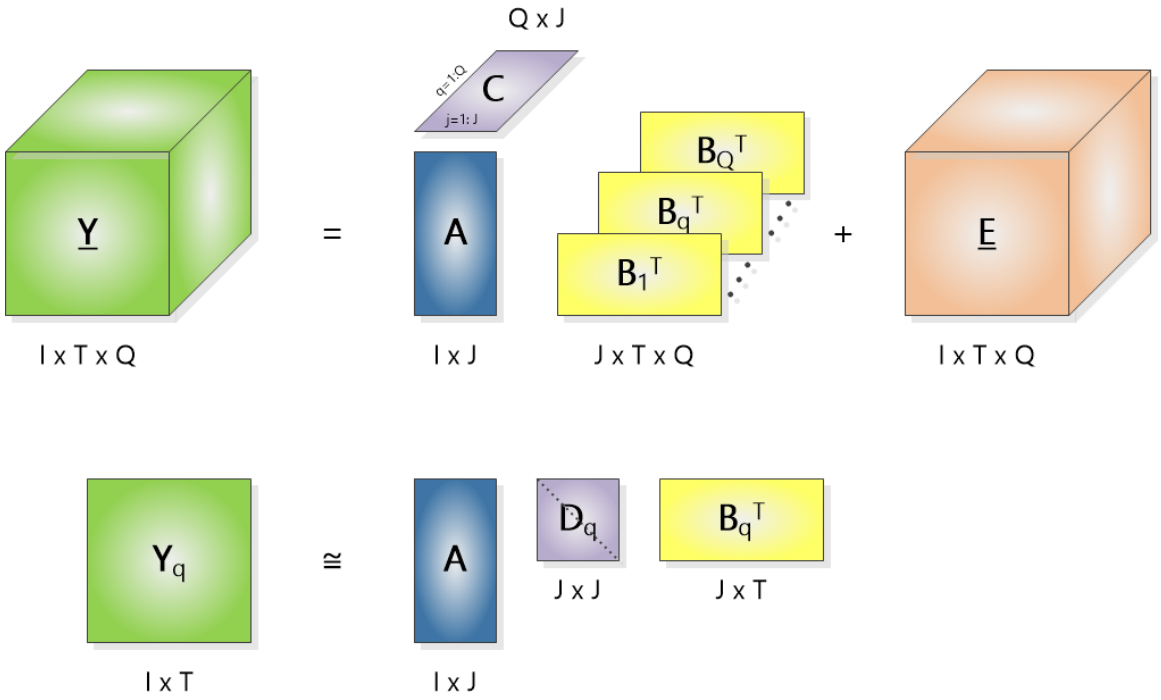

Figure A.2: NTF1 Model

A.7.3 NTF2

$$y_{itq} = \sum_{j=1}^{J} a_{ijq} b_{tjq} c_{qj} + e_{itq} \tag{A.71}$$

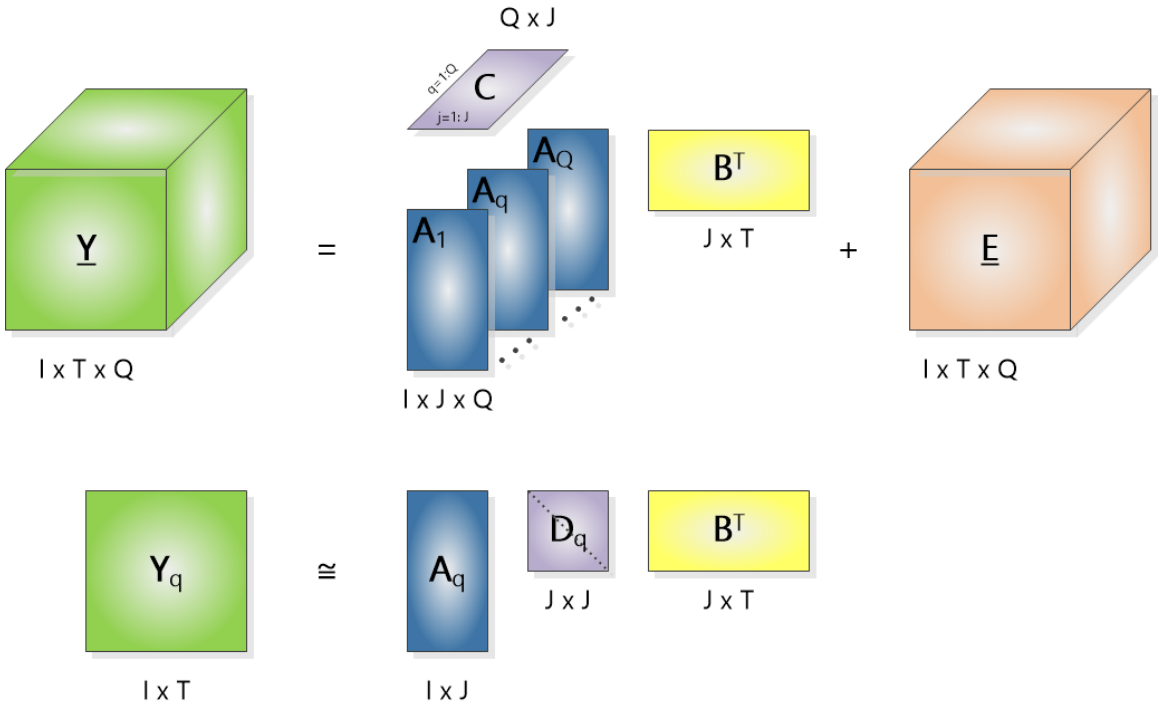

Figure A.3: NTF2 Model

A.8 Reviewing the Update Rules

A.8.0.1 W update

$$w_{fom} \leftarrow w_{fom} \frac{\sum_{n,k,l} z_{ol} y_{lk} t_{fk} v_{kn} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{n,k,l} z_{ol} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm}}$$

Let us rearrange the result into a more computationally friendly form.

$$w_{fom} \leftarrow w_{fom} \frac{num_W}{den_W}$$

Where

$$\begin{aligned} den_W &= \sum_{n,k,l} z_{ol} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm} \\ &= \sum_n \hat{x}_{fnm} \sum_k t_{fk} v_{kn} \sum_l z_{ol} y_{lk} \end{aligned} \tag{A.72}$$

and

$$\begin{aligned} num_W &= \sum_{n,k,l} z_{ol} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm} + \operatorname{Re}\left\{ \sum_{n,k,l} z_{ol} y_{lk} t_{fk} v_{kn} E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} \right\} \\ &= den_W + \operatorname{Re}\left\{ \left\{ \sum_n E^*_{fnm} \sum_k t_{fk} v_{kn} e^{j\Phi_S(f,n,k)} \sum_l z_{ol} y_{lk} e^{j\Phi_U(f,l,n,m)} \right\} e^{j\Phi_W(f,o,m)} \right\} \end{aligned} \tag{A.73}$$

And let us proceed to do the same for the other update rules.

A.8.0.2 Z update

$$z_{ol} \leftarrow z_{ol} \frac{\sum_{m,f,n,k} w_{fom} y_{lk} t_{fk} v_{kn} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,f,n,k} w_{fom} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm}}$$

$$z_{ol} \leftarrow z_{ol} \frac{num_Z}{den_Z}$$

$$\begin{aligned} den_Z &= \sum_{m,f,n,k} w_{fom} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm} \\ &= \sum_m \sum_f w_{fom} \sum_n \hat{x}_{fnm} \sum_k t_{fk} v_{kn} y_{lk} \end{aligned} \tag{A.74}$$

$$\begin{aligned} num_Z &= \sum_{m,f,n,k} w_{fom} y_{lk} t_{fk} v_{kn} \hat{x}_{fnm} + \mathrm{Re}\left\{ \sum_{m,f,n,k} w_{fom} y_{lk} t_{fk} v_{kn} E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} \right\} \\ &= den_Z + \mathrm{Re}\left\{ \sum_m \sum_f w_{fom} e^{j\Phi_W(f,o,m)} \sum_n E^*_{fnm} e^{j\Phi_U(f,l,n,m)} \sum_k t_{fk} v_{kn} y_{lk} e^{j\Phi_S(f,n,k)} \right\} \end{aligned} \tag{A.75}$$

A.8.0.3 Y update

$$y_{lk} \leftarrow y_{lk} \frac{\sum_{m,f,n,o} w_{fom} z_{ol} t_{fk} v_{kn} \mathrm{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,f,n,o} w_{fom} z_{ol} t_{fk} v_{kn} \hat{x}_{fnm}}$$

$$y_{lk} \leftarrow y_{lk} \frac{num_Y}{den_Y}$$

$$\begin{aligned} den_Y &= \sum_{m,f,n,o} w_{fom} z_{ol} t_{fk} v_{kn} \hat{x}_{fnm} \\ &= \sum_m \sum_f t_{fk} \sum_n \hat{x}_{fnm} v_{kn} \sum_o w_{fom} z_{ol} \end{aligned} \tag{A.76}$$

$$\begin{aligned} num_Y &= \sum_{m,f,n,o} w_{fom} z_{ol} t_{fk} v_{kn} \hat{x}_{fnm} + \mathrm{Re}\left\{ \sum_{m,f,n,o} w_{fom} z_{ol} t_{fk} v_{kn} E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} \right\} \\ &= den_Y + \mathrm{Re}\left\{ \sum_m \sum_f t_{fk} \sum_n E^*_{fnm} v_{kn} e^{j\Phi_S(f,n,k)} e^{j\Phi_U(f,l,n,m)} \sum_o w_{fom} z_{ol} e^{j\Phi_W(f,o,m)} \right\} \end{aligned} \tag{A.77}$$

A.8.0.4 T update

$$t_{fk} \leftarrow t_{fk} \frac{\sum_{m,n,l,o} w_{fom} z_{ol} y_{lk} v_{kn} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,n,l,o} w_{fom} z_{ol} y_{lk} v_{kn} \hat{x}_{fnm}}$$

$$t_{fk} \leftarrow t_{fk} \frac{num_T}{den_T}$$

$$\begin{aligned} den_T &= \sum_{m,n,l,o} w_{fom} z_{ol} y_{lk} v_{kn} \hat{x}_{fnm} \\ &= \sum_m \sum_n \hat{x}_{fnm} v_{kn} \sum_l y_{lk} \sum_o w_{fom} z_{ol} \end{aligned} \tag{A.78}$$

$$\begin{aligned} num_T &= \sum_{m,n,l,o} w_{fom} z_{ol} y_{lk} v_{kn} \hat{x}_{fnm} + \operatorname{Re}\left\{ \sum_{m,n,l,o} w_{fom} z_{ol} y_{lk} v_{kn} E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} \right\} \\ &= den_T + \operatorname{Re}\left\{ \sum_m \sum_n E^*_{fnm} v_{kn} e^{j\Phi_S(f,n,k)} \sum_l y_{lk} e^{j\Phi_U(f,l,n,m)} \sum_o w_{fom} z_{ol} e^{j\Phi_W(f,o,m)} \right\} \end{aligned} \tag{A.79}$$

A.8.0.5 V update

$$v_{kn} \leftarrow v_{kn} \frac{\sum_{m,f,l,o} w_{fom} z_{ol} y_{lk} t_{fk} \operatorname{Re}\{\hat{x}_{fnm} + E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]}\}}{\sum_{m,f,l,o} w_{fom} z_{ol} y_{lk} t_{fk} \hat{x}_{fnm}}$$

$$v_{kn} \leftarrow v_{kn} \frac{num_V}{den_V}$$

$$\begin{aligned} den_V &= \sum_{m,f,l,o} w_{fom} z_{ol} y_{lk} t_{fk} \hat{x}_{fnm} \\ &= \sum_m \sum_f \hat{x}_{fnm} t_{fk} \sum_l y_{lk} \sum_o w_{fom} z_{ol} \end{aligned} \tag{A.80}$$

$$num_V = \sum_{m,f,l,o} w_{fom} z_{ol} y_{lk} t_{fk} \hat{x}_{fnm} + \mathrm{Re}\left\{ \sum_{m,f,l,o} w_{fom} z_{ol} y_{lk} t_{fk} E^*_{fnm} e^{j[\Phi_W(f,o,m)+\Phi_U(f,l,n,m)+\Phi_S(f,n,k)]} \right\}$$
$$= den_V + \mathrm{Re}\left\{ \sum_m \sum_f E^*_{fnm} t_{fk} e^{j\Phi_S(f,n,k)} \sum_l y_{lk} e^{j\Phi_U(f,l,n,m)} \sum_o w_{fom} z_{ol} e^{j\Phi_W(f,o,m)} \right\} \tag{A.81}$$

A.8.1 Interchannel Auxiliary Function: Z Update Derivation

We seek to compute

$$\frac{\partial}{\partial z_{ol}} \mathcal{C}^+_{\phi_d} = \frac{\partial}{\partial z_{ol}} \mathcal{C}^+_{\phi_d} + \frac{\partial}{\partial z_{pq}} \mathcal{C}^+_{\phi_d}$$

$$
\begin{aligned}
\frac{\partial}{\partial z_{ol}}\mathcal{C}^{+}_{\phi_d} = &\sum_{f,n}\sum_{k,g}\sum_{q,p}\frac{1}{\beta_{fnkglqop}}\Big\{ \\
&- R_{fnkglqop}[\mathbf{w}_{fo}]^{*}_{b}[\mathbf{w}_{fp}]_{a} z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg}e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)+\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]} \\
&- R^{*}_{fnkglqop}[\mathbf{w}_{fo}]_{b}[\mathbf{w}_{fp}]^{*}_{a} z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg}e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]} \\
&+ 2w^{2}_{fob}w^{2}_{fpa}z_{ol}z^{2}_{pq}y^{2}_{lk}y^{2}_{qg}t^{2}_{fk}t^{2}_{fg}v^{2}_{kn}v^{2}_{kg}\Big\} \\
&= \sum_{f,n}\sum_{k,g}\sum_{q,p}\Big\{ - [\mathbf{w}_{fo}]^{*}_{b}[\mathbf{w}_{fp}]_{a} z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg}\Big(E_{fn}e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)+\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]} \\
&+ \hat{x}_{fn}e^{j[\Phi_W(f,o,b)-\Phi_W(f,o,a)]}\Big) \\
&- [\mathbf{w}_{fo}]_{b}[\mathbf{w}_{fp}]^{*}_{a} z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg}\Big(E^{*}_{fn}e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]} \\
&+ \hat{x}_{fn}e^{j[-\Phi_W(f,o,b)+\Phi_W(f,o,a)]}\Big) \\
&+ 2\frac{\hat{x}_{fn}z_{ol}w^{2}_{fob}w^{2}_{fpa}z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg}}{z_{ol}w_{fob}w_{fpa}}\Big\} \\
&= \sum_{f,n}\sum_{k,g}\sum_{q,p} -\Big(E_{fn}e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)+\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]}\Big)[\mathbf{w}_{fo}]^{*}_{b}[\mathbf{w}_{fp}]_{a} z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg} \\
&- \Big(E^{*}_{fn}e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]}\Big)[\mathbf{w}_{fo}]_{b}[\mathbf{w}_{fp}]^{*}_{a} z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg} \\
&+ \sum_{f,n}\sum_{k,g}\sum_{q,p}(-2+2)\hat{x}_{fn}w_{fob}w_{fpa}z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg} \\
&= \sum_{f,n,k} -\big(E_{fn}e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)]}\big)\Big(\sum_{g,q,p}[\mathbf{w}_{fp}]_{a} z_{pq}y_{qg}t_{fg}v_{kg}e^{j[\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]}\Big)[\mathbf{w}_{fo}]^{*}_{b}y_{lk}t_{fk}v_{kn} \\
&- \big(E^{*}_{fn}e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)]}\big)([\mathbf{w}_{fo}]_{b}y_{lk}t_{fk}v_{kn})\Big(\sum_{g,q,p}[\mathbf{w}_{fp}]^{*}_{a} z_{pq}y_{qg}t_{fg}v_{kg}e^{j[-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]}\Big) \\
&+ \sum_{f,n,k}(-2+2)\hat{x}_{fn}w_{fob}y_{lk}t_{fk}v_{kn}\sum_{g,q,p}w_{fpa}z_{pq}y_{qg}t_{fg}v_{kg} \\
&= \sum_{f,n,k} -E_{fn}[\hat{\mathbf{x}}_{fn}]_{a}[\mathbf{w}_{fo}]^{*}_{b}y_{lk}t_{fk}v_{kn}e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)]} \\
&- E^{*}_{fn}[\hat{\mathbf{x}}_{fn}]^{*}_{a}[\mathbf{w}_{fo}]_{b}y_{lk}t_{fk}v_{kn}e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)]} + \sum_{f,n,k}(-2+2)\hat{x}_{fn}w_{fob}y_{lk}t_{fk}v_{kn}\hat{x}_{fn,a} \\
&= \sum_{f,n,k} -2\,\mathrm{Re}\left\{E_{fn}[\hat{\mathbf{x}}_{fn}]_{a}[\mathbf{w}_{fo}]^{*}_{b}y_{lk}t_{fk}v_{kn}e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)]}\right\} + (-2+2)\hat{x}_{fn}w_{fob}y_{lk}t_{fk}v_{kn}\hat{x}_{fn,a}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial}{\partial z_{pq}}\mathcal{C}^{+}_{\phi_d} = &\sum_{f,n}\sum_{k,g}\sum_{q,p}\frac{1}{\beta_{fnkglqop}}\Big\{ \\
&- R_{fnkglqop}[\mathbf{w}_{fo}]^{*}_{b}[\mathbf{w}_{fp}]_{a} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)+\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]} \\
&- R^{*}_{fnkglqop}[\mathbf{w}_{fo}]_{b}[\mathbf{w}_{fp}]^{*}_{a} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]} \\
&+ 2 w^2_{fob} w^2_{fpa} z_{pq} z^2_{ol} y^2_{lk} y^2_{qg} t^2_{fk} t^2_{fg} v^2_{kn} v^2_{kg} \Big\} \\
&= \sum_{f,n}\sum_{k,g}\sum_{q,p}\Big\{ - [\mathbf{w}_{fo}]^{*}_{b}[\mathbf{w}_{fp}]_{a} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} \Big(E_{fn} e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)+\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]} \\
&+ \hat{x}_{fn} e^{j[\Phi_W(f,o,b)-\Phi_W(f,o,a)]} \Big) \\
&- [\mathbf{w}_{fo}]_{b}[\mathbf{w}_{fp}]^{*}_{a} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} \Big(E^{*}_{fn} e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]} \\
&+ \hat{x}_{fn} e^{j[-\Phi_W(f,o,b)+\Phi_W(f,o,a)]} \Big) \\
&+ 2\frac{\hat{x}_{fn} z_{pq} w^2_{fob} w^2_{fpa} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg}}{z_{pq} w_{fob} w_{fpa}} \Big\} \\
&= \sum_{f,n}\sum_{k,g}\sum_{q,p} - \Big(E_{fn} e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)+\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]} \Big) [\mathbf{w}_{fo}]^{*}_{b}[\mathbf{w}_{fp}]_{a} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} \\
&- \Big(E^{*}_{fn} e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]} \Big) [\mathbf{w}_{fo}]_{b}[\mathbf{w}_{fp}]^{*}_{a} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} \\
&+ \sum_{f,n}\sum_{k,g}\sum_{q,p}(-2+2)\hat{x}_{fn} w_{fob} w_{fpa} z_{ol} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} \\
&= \sum_{f,n,g} - \big(E_{fn} e^{j[\Phi_{S(f,n,g)}+\Phi_U(f,n,q,a)]} \big) \Big(\sum_{k,l,o} [\mathbf{w}_{fo}]^{*}_{b} z_{ol} y_{lk} t_{fk} v_{kn} e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)]} \Big) [\mathbf{w}_{fp}]_{a} y_{qg} t_{fg} v_{kg} \\
&- \big(E^{*}_{fn} e^{j[-\Phi_{S(f,n,g)}-\Phi_U(f,n,q,a)]} \big) \big([\mathbf{w}_{fp}]^{*}_{a} y_{qg} t_{fg} v_{kg} \big) \Big(\sum_{k,l,o} [\mathbf{w}_{fo}]_{b} z_{ol} y_{lk} t_{fk} v_{kn} e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)]} \Big) \\
&+ \sum_{f,n,g}(-2+2)\hat{x}_{fn} w_{fpa} y_{qg} t_{fg} v_{kg} \sum_{k,l,o} z_{ol} w_{fob} y_{lk} t_{fk} v_{kn} \\
&= \sum_{f,n,g} - E_{fn} [\hat{\mathbf{x}}_{fn}]^{*}_{b} [\mathbf{w}_{fp}]_{a} y_{qg} t_{fg} v_{gn} e^{j[\Phi_S(f,n,g)+\Phi_U(f,n,q,a)]} \\
&- E^{*}_{fn} [\hat{\mathbf{x}}_{fn}]_{b} [\mathbf{w}_{fp}]^{*}_{a} y_{qg} t_{fg} v_{gn} e^{j[-\Phi_S(f,n,g)-\Phi_U(f,n,q,a)]} + \sum_{f,n,g}(-2+2)\hat{x}_{fn} w_{fpa} y_{qg} t_{fg} v_{gn} \hat{x}_{fn,b} \\
&= \sum_{f,n,g} -2\,\mathrm{Re}\left\{ E_{fn} [\hat{\mathbf{x}}_{fn}]^{*}_{b} [\mathbf{w}_{fp}]_{a} y_{qg} t_{fg} v_{gn} e^{j[\Phi_S(f,n,g)+\Phi_U(f,n,q,a)]} \right\} + (-2+2)\hat{x}_{fn} w_{fpa} y_{qg} t_{fg} v_{gn} \hat{x}_{fn,b}
\end{aligned}
$$

$$\frac{\partial}{\partial z_{pq}}|_{g\to k,q\to l,p\to o}$$
$$= \sum_{f,n,k} -2\,\mathrm{Re}\left\{E_{fn}[\hat{\mathbf{x}}_{fn}]_b^*[\mathbf{w}_{fo}]_a y_{lk} t_{fk} v_{kn} e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,a)]}\right\} + (-2+2)\hat{x}_{fn} w_{foa} y_{lk} t_{fk} v_{kn} \hat{x}_{fn,b}$$

A.8.2 Orthogonality and Clustering Extensions for V, Z, and Y

As mentioned within the thesis, the proposed algorithm did not do well when corresponding rows or columns within any of the matrices $\mathbf{V}$ or $\mathbf{Y}$ for instance were forced to be entirely zero. To some degree it could be implemented, however, in most cases it was eventually shown to be problematic. The cost functions here and multiplicative update rules demonstrate how an NMF gradient update can be applied in order to enforce the constraint of an orthogonal representation of the matrices $\mathbf{V}$, $\mathbf{Z}$, or $\mathbf{Y}$, however none of these update rules were enabled in practice since a superior overall configuration of the proposed algorithm was eventually discovered.

To *illustrate*, however, a derived set of orthogonality constraints, let us introduce two new cost functions each of which is intended to impose either row orthogonality or column orthogonality on the rows and columns of our NMF parameters.

A.8.2.1 Row Orthogonality

Let us focus on deriving a way into introduce more orthogonality into the $k = 1 : K$ rows of the activation parameter v_{kn}. To do this we will rely upon shaping the quantity $VV^T \in \mathbb{R}^{K\times K}$ with the help of a mask that corresponds to a KxK banded matrix so as to force the quantity to resemble more of a diagonal matrix with zeros on the off diagonal regions and only significant energy located on the main diagonal and neighbouring diagonals.

Let us denote the masked quantity to be used as a reference quantity R_v and derive the row orthogonality update cost function as follows:

$$\begin{aligned}\Psi_V &= ||R_v - VV^T||_F^2 \\ &= \mathrm{tr}[(R_v - VV^T)(R_v - VV^T)^T] \\ &= \mathrm{tr}[R_v R_v^T - R_v VV^T - VV^T R_v^T + VV^T VV^T] \end{aligned} \tag{A.82}$$

where $V \in \mathbb{R}^{K\times N}$ is the activation matrix with k, nth element $[V]_{kn} = v_{kn}$ and where $R_v \in \mathbb{R}^{K\times K}$ is the reference quantity computed for the purpose of introducing orthogonality

into the rows of V. We continue by converting between matrix notation and "index notation" with respect to the trace operator:

$$\begin{aligned}\Psi_V &= \sum_a^K [R_v R_v^T]_{aa} - \sum_c^K [R_v V V^T]_{cc} - \sum_f^K [V V^T R_v^T]_{ff} + \sum_i^K [V V^T V V^T]_{ii} \\ &= \sum_a^K \sum_b^K r_{ab} r_{ab} - \sum_c^K \sum_d^K \sum_e^N r_{cd} v_{de} v_{ce} - \sum_f^K \sum_g^N \sum_h^K v_{fg} v_{hg} r_{fh} + \sum_i^K \sum_j^N \sum_k^K \sum_l^N v_{ij} v_{kj} v_{kl} v_{il}\end{aligned}$$

Compute $\dfrac{\partial \Psi_V}{\partial v_{kn}}$

$$\begin{aligned}\frac{\partial \Psi_V}{\partial v_{kn}} &= \frac{\partial \Psi_V}{\partial v_{de}} + \frac{\partial \Psi_V}{\partial v_{ce}} + \frac{\partial \Psi_V}{\partial v_{fg}} + \frac{\partial \Psi_V}{\partial v_{hg}} + \frac{\partial \Psi_V}{\partial v_{ij}} + \frac{\partial \Psi_V}{\partial v_{kj}} + \frac{\partial \Psi_V}{\partial v_{kl}} + \frac{\partial \Psi_V}{\partial v_{il}} \\ &= -\sum_c^K r_{cd} v_{ce} - \sum_d^K r_{cd} v_{de} - \sum_h^K v_{hg} r_{fh} - \sum_f^K v_{fg} r_{fh} + \sum_k^K \sum_l^N v_{kj} v_{kl} v_{il} + \sum_i^K \sum_l^N v_{ij} v_{kl} v_{il} \\ &+ \sum_i^K \sum_j^N v_{ij} v_{kj} v_{il} + \sum_j^N \sum_k^K v_{ij} v_{kj} v_{kl} \\ &= -[V^T R_V]_{ed} - [R_V V]_{ce} - [V^T R_V^T]_{gf} - [R_V^T V]_{hg} + [V^T V V^T]_{ji} + [V V^T V]_{kj} + [V^T V V^T]_{lk} + [V V^T V]_{il} \\ &= -[R_V^T V]_{de} - [R_V V]_{ce} - [R_V V]_{fg} - [R_V^T V]_{hg} + [V V^T V]_{ij} + [V V^T V]_{kj} + [V V^T V]_{kl} + [V V^T V]_{il} \\ &= -2[R_V^T V]_{kn} - 2[R_V V]_{kn} + 4[V V^T V]_{kn} \end{aligned} \tag{A.83}$$

A.8.2.2 Column Orthogonality

Column orthogonality was achieved in a similar manner but by considering a different cost function as given by

$$\begin{aligned}\Psi_Z &= ||R_z - Z^T Z||_F^2 \\ &= \mathrm{tr}[(R_z - Z^T Z)(R_z - Z^T Z)^T] \\ &= \mathrm{tr}[R_z R_z^T - R_z Z^T Z - Z^T Z R_z^T + Z^T Z Z^T Z] \end{aligned} \tag{A.84}$$

$$\Psi_Z = \sum_a^L [R_z R_z^T]_{aa} - \sum_c^L [R_z Z^T Z]_{cc} - \sum_f^L [Z^T Z R_z^T]_{ff} + \sum_i^L [Z^T Z Z^T Z]_{ii}$$
$$= \sum_a^L \sum_b^L r_{ab} r_{ab} - \sum_c^L \sum_d^L \sum_e^O r_{cd} z_{ed} z_{ec} - \sum_f^L \sum_g^O \sum_h^L z_{gf} z_{gh} r_{fh} + \sum_i^L \sum_j^O \sum_k^L \sum_m^O z_{ji} z_{jk} z_{mk} z_{mi}$$

Compute $\dfrac{\partial \Psi_Z}{\partial z_{ol}}$

$$\frac{\partial \Psi_Z}{\partial z_{ol}} = \frac{\partial \Psi_Z}{\partial z_{ed}} + \frac{\partial \Psi_Z}{\partial z_{ec}} + \frac{\partial \Psi_Z}{\partial z_{gf}} + \frac{\partial \Psi_Z}{\partial z_{gh}} + \frac{\partial \Psi_Z}{\partial z_{ji}} + \frac{\partial \Psi_Z}{\partial z_{jk}} + \frac{\partial \Psi_Z}{\partial z_{mk}} + \frac{\partial \Psi_Z}{\partial z_{mi}}$$
$$= -\sum_c^L r_{cd} z_{ec} - \sum_d^L r_{cd} z_{ed} - \sum_h^L z_{gh} r_{fh} - \sum_f^L z_{gf} r_{fh} + \sum_k^L \sum_m^O z_{jk} z_{mk} z_{mi} + \sum_i^L \sum_m^O z_{ji} z_{mk} z_{mi}$$
$$+ \sum_i^L \sum_j^O z_{ji} z_{jk} z_{mi} + \sum_j^O \sum_k^L z_{ji} z_{mk} z_{mk}$$
$$= -[Z R_Z]_{ed} - [R_Z Z^T]_{ce} - [Z R_Z^T]_{gf} - [R_Z^T Z^T]_{hg} + [Z Z^T Z]_{ji} + [Z^T Z Z^T]_{kj} + [Z Z^T Z]_{mk} + [Z^T Z Z^T]_{im}$$
$$= -[Z R_Z]_{ed} - [Z R_Z^T]_{ec} - [Z R_Z^T]_{gf} - [Z R_Z]_{gh} + [Z Z^T Z]_{ji} + [Z Z^T Z]_{jk} + [Z Z^T Z]_{mk} + [Z Z^T Z]_{mi}$$
$$= -2[Z R_Z]_{ol} - 2[Z R_Z^T]_{ol} + 4[Z Z^T Z]_{ol} \tag{A.85}$$

A.9 Proposed Algorithm Extensions

A.9.1 Interchannel Auxiliary Function Extension

Define an auxiliary variable $R_{fnkglqop}$ such that

$$\sum_{k,g} \sum_{l,q} \sum_{o,p} R_{fnkglqop} = \tilde{Q}_{fn}(b,a) = [\tilde{\mathbf{x}}_{fn}]_b [\tilde{\mathbf{x}}_{fn}]_a^* \tag{A.86}$$

Model the interchannel quantity between channel b and a as follows:

$$\begin{aligned}\hat{Q}_{fn}(b,a) &= [\hat{\mathbf{x}}_{fn}]_b[\hat{\mathbf{x}}_{fn}]_a^* \\ &= \sum_k h_{fnk,b}t_{fk}v_{kn}e^{j\Phi_S(f,n,k)}\sum_g h^*_{fng,a}t_{fg}v_{gn}e^{-j\Phi_{S(f,n,g)}} \\ &= \sum_{k,g}\sum_{l,q}\sum_{o,p}[\mathbf{w}_{fo}]_b[\mathbf{w}_{fp}]^*_a z_{ol}z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{gn}e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,l,a)]}\end{aligned} \tag{A.87}$$

Define the parameter set $\theta = \{W, T, V, Y, Z, \Phi_W, \Phi_S, \Phi_U\}$ and the objective function to be minimized as follows:

$$\mathcal{C}_{\phi_d}(\theta) = \sum_{f=1}^{F}\sum_{n=1}^{N}\left|\tilde{Q}_{fn}(b,a) - \hat{Q}_{fn}(b,a)\right|^2 \tag{A.88}$$

with the intent of minimizing the squared error between the observed and modelled spectrogram, where we name

$$E_{fn} = \tilde{Q}_{fn}(b,a) - \hat{Q}_{fn}(b,a) \tag{A.89}$$

the "error" of the model.

Also define a set of weights $\beta_{fnkglqop}(b,a)$ such that

$$\sum_{k,g}\sum_{l,q}\sum_{o,p}\beta_{fnkglqop}(b,a) = [\mathbf{1}]_{fn}$$

Claim that the auxiliary function is minimized when:

$$\begin{aligned}R_{fnkglqop}(b,a) &= [\mathbf{w}_{fo}]_b[\mathbf{w}_{fp}]^*_a z_{ol}z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg}e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,l,a)]} \\ &\quad + (\beta_{fnkglqop}(b,a))E_{fn}\end{aligned}$$

And define,

$$\beta_{fnkglqop}(b,a) = \frac{w_{fob}w_{fpa}z_{ol}z_{pq}y_{lk}y_{qg}t_{fk}t_{fg}v_{kn}v_{kg}}{\hat{x}_{fn}(b,a)} \tag{A.90}$$

And the following magnitude model:

$$\hat{x}_{fn}(b,a) = \sum_{k}\sum_{l}\sum_{o} w_{fob} z_{ol} y_{lk} t_{fk} v_{kn} \sum_{g}\sum_{q}\sum_{p} w_{fpa} z_{pq} y_{qg} t_{fg} v_{gn} \tag{A.91}$$

Where an interpretation of this magnitude model can be defined in terms of the b and a'th microphone channels of the previous magnitude model as follows:

$$\hat{x}_{fn}(b,a) = \hat{x}_{fn,b}\hat{x}_{fn,a} \tag{A.92}$$

To this point we have denoted each of the variables related to the new auxiliary function explicitly as a function of b and a. From this point forward we may implicitly drop this labelling convention for a more compact notation.

Now the auxiliary function to be minimized is presented:

$$\begin{aligned}\mathcal{C}^{+}_{\phi_d}(\theta) = &\sum_{f,n}\sum_{k,g}\sum_{l,q}\sum_{o,p}\frac{1}{\beta_{fnkglqop}}\Big\{\big|R_{fnkglqop} \\ &- [\mathbf{w}_{fo}]_b[\mathbf{w}_{fp}]^*_a z_{ol} z_{pq} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,l,a)]}\big|^2\Big\}\end{aligned}$$

$$\begin{aligned}=&\sum_{f,n}\sum_{k,g}\sum_{l,q}\sum_{o,p}\frac{1}{\beta_{fnkglqop}}\Big\{R_{fnkglqop}R^*_{fnkglqop} \\ &- R_{fnkglqop}[\mathbf{w}_{fo}]^*_b[\mathbf{w}_{fp}]_a z_{ol} z_{pq} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} e^{j[-\Phi_S(f,n,k)-\Phi_U(f,n,l,b)+\Phi_{S(f,n,g)}+\Phi_U(f,n,l,a)]} \\ &- R^*_{fnkglqop}[\mathbf{w}_{fo}]_b[\mathbf{w}_{fp}]^*_a z_{ol} z_{pq} y_{lk} y_{qg} t_{fk} t_{fg} v_{kn} v_{kg} e^{j[\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_{S(f,n,g)}-\Phi_U(f,n,l,a)]} \\ &+ w^2_{fob} w^2_{fpa} z^2_{ol} z^2_{pq} y^2_{lk} y^2_{qg} t^2_{fk} t^2_{fg} v^2_{kn} v^2_{kg}\Big\}\end{aligned} \tag{A.93}$$

We can now also note that:

$$\frac{R_{fnkglqop}}{\beta_{fnkglqop}} = \hat{x}_{fn} e^{j[\Phi_W(f,o,b)+\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_W(f,o,a)-\Phi_{S(f,n,g)}-\Phi_U(f,n,l,a)]} + E_{fn} \tag{A.94}$$

$$\frac{R^*_{fnkglqop}}{\beta_{fnkglqop}} = \hat{x}_{fn} e^{-j[\Phi_W(f,o,b)+\Phi_S(f,n,k)+\Phi_U(f,n,l,b)-\Phi_W(f,o,a)-\Phi_{S(f,n,g)}-\Phi_U(f,n,l,a)]} + E^*_{fn} \tag{A.95}$$

The update derivation of the parameter z_{ol} is included in the thesis but the reader will be pointed to the Appendix at section A.8.1 for its derivation.

Take the partial derivatives of $\mathcal{C}^{+}$ with respect to each of its parameter matrices and

tensors, also treating $R_{fnkglqop}$ as a parameter. We will not sub the expression for $R_{fnkglqop}$ in until after we have computed each partial derivative. We do this in order to obtain multiplicative updates. Therefore note: $R_{fnkglqop}R^*_{fnkglqop}$ is a term of the auxiliary function that can be ignored when computing each partial derivative.

Set $\dfrac{\partial \mathcal{C}^+}{\partial w_{fom}} = 0$, and re-arrange in order to obtain the multiplicative update:

$$w_{fob} \leftarrow w_{fob} \frac{\sum\limits_{n,k,l} 2\Big[\mathrm{Re}\{E_{fn}([\hat{\mathbf{x}}_{fn}]_a e^{-j[\Phi_W(f,o,b)+\Phi_U(f,n,l,b)+\Phi_S(f,n,k)]}\} + \hat{x}_{fn}\hat{x}_{fn,a}\Big] z_{ol}y_{lk}t_{fk}v_{kn}}{\sum\limits_{n,k,l} 2\hat{x}_{fn}\hat{x}_{fn,a}z_{ol}y_{lk}t_{fk}v_{kn}} \tag{A.96}$$

$$w_{foa} \leftarrow w_{foa} \frac{\sum\limits_{n,k,l} 2\Big[\mathrm{Re}\{E_{fn}([\hat{\mathbf{x}}_{fn}]^*_b e^{j[\Phi_W(f,o,a)+\Phi_U(f,n,l,a)+\Phi_S(f,n,k)]}\} + \hat{x}_{fn}\hat{x}_{fn,b}\Big] z_{ol}y_{lk}t_{fk}v_{kn}}{\sum\limits_{n,k,l} 2\hat{x}_{fn}\hat{x}_{fn,b}z_{ol}y_{lk}t_{fk}v_{kn}} \tag{A.97}$$

The update derivations for $z_{ol}, y_{lk}, t_{fk}, v_{kn}$ are all obtained in a similar manner and provided in summary for convenience:

$$z_{ol} \leftarrow z_{ol} \frac{\sum\limits_{f,n,k} 2\Big[\mathrm{Re}\{E_{fn}([\hat{\mathbf{x}}_{fn}]_a[\mathbf{w}_{fo}]^*_b e^{-j[\Phi_U(f,n,l,b)+\Phi_S(f,n,k)]} + [\hat{\mathbf{x}}_{fn}]^*_b[\mathbf{w}_{fp}]_a e^{j[\Phi_U(f,n,l,a)+\Phi_S(f,n,k)]})\}\Big] y_{lk}t_{fk}v_{kn} + den_Z}{den_Z} \tag{A.98}$$

$$den_Z = \sum_{f,n,k} 2\hat{x}_{fn}(w_{fob}\hat{x}_{fn,a} + w_{foa}\hat{x}_{fn,b})y_{lk}t_{fk}v_{kn}$$

$$y_{lk} \leftarrow y_{lk} \frac{\sum\limits_{f,n,o} 2\Big[\mathrm{Re}\{E_{fn}([\hat{\mathbf{x}}_{fn}]_a[\mathbf{w}_{fo}]^*_b e^{-j[\Phi_U(f,n,l,b)+\Phi_S(f,n,k)]} + [\hat{\mathbf{x}}_{fn}]^*_b[\mathbf{w}_{fp}]_a e^{j[\Phi_U(f,n,l,a)+\Phi_S(f,n,k)]})\}\Big] z_{ol}t_{fk}v_{kn} + den_Y}{den_Y} \tag{A.99}$$

$$den_Y = \sum_{f,n,o} 2\hat{x}_{fn}(w_{fob}\hat{x}_{fn,a} + w_{foa}\hat{x}_{fn,b})z_{ol}t_{fk}v_{kn}$$

$$t_{fk} \leftarrow t_{fk} \frac{\sum\limits_{n,l,o} 2\Big[\mathrm{Re}\{E_{fn}([\hat{\mathbf{x}}_{fn}]_a[\mathbf{w}_{fo}]^*_b e^{-j[\Phi_U(f,n,l,b)+\Phi_S(f,n,k)]} + [\hat{\mathbf{x}}_{fn}]^*_b[\mathbf{w}_{fp}]_a e^{j[\Phi_U(f,n,l,a)+\Phi_S(f,n,k)]})\}\Big] z_{ol}y_{lk}v_{kn} + den_T}{den_T} \tag{A.100}$$

$$den_T = \sum_{n,l,o} 2\hat{x}_{fn}(w_{fob}\hat{x}_{fn,a} + w_{foa}\hat{x}_{fn,b})z_{ol}y_{lk}v_{kn}$$

$$v_{kn} \leftarrow v_{kn} \frac{\sum\limits_{f,l,o} 2\Big[\mathrm{Re}\{E_{fn}([\hat{\mathbf{x}}_{fn}]_a[\mathbf{w}_{fo}]^*_b e^{-j[\Phi_U(f,n,l,b)+\Phi_S(f,n,k)]} + [\hat{\mathbf{x}}_{fn}]^*_b[\mathbf{w}_{fp}]_a e^{j[\Phi_U(f,n,l,a)+\Phi_S(f,n,k)]})\}\Big] z_{ol}y_{lk}t_{fk} + den_V}{den_V} \tag{A.101}$$

$$den_V = \sum_{f,l,o} 2\hat{x}_{fn}(w_{fob}\hat{x}_{fn,a} + w_{foa}\hat{x}_{fn,b})z_{ol}y_{lk}t_{fk}$$

$$\frac{\partial \mathcal{C}^+}{\partial \Phi_S(f,n,k)} = \frac{\partial \mathcal{C}^+}{\partial \Phi_S(f,n,k)} + \frac{\partial \mathcal{C}^+}{\partial \Phi_S(f,n,g)}$$

$$\frac{\partial \mathcal{C}^{+}}{\partial \Phi_S(f,n,k)} = \mathrm{Re}\left\{ -je^{-j\Phi_S(f,n,k)} \sum_{l,o} -2\Big[E_{fn}[\hat{\mathbf{x}}_{fn}]_a [\mathbf{w}_{fo}]_b^* + \hat{x}_{fn}\hat{x}_{fn,a} w_{fob} e^{j\Phi_S(f,n,k)}\Big] z_{ol} y_{lk} t_{fk} v_{kn} \right\}$$
$$= \mathrm{Re}\left\{ je^{j\Phi_S(f,n,k)} \sum_{l,o} -2\Big[E_{fn}^{*}[\hat{\mathbf{x}}_{fn}]_a^* [\mathbf{w}_{fo}]_b + \hat{x}_{fn}\hat{x}_{fn,a} w_{fob} e^{-j\Phi_S(f,n,k)}\Big] z_{ol} y_{lk} t_{fk} v_{kn} \right\} \tag{A.102}$$

$$\frac{\partial \mathcal{C}^{+}}{\partial \Phi_S(f,n,g)} = \mathrm{Re}\left\{ je^{j\Phi_S(f,n,k)} \sum_{l,o} -2\Big[E_{fn}[\hat{\mathbf{x}}_{fn}]_b^* [\mathbf{w}_{fo}]_a + \hat{x}_{fn}\hat{x}_{fn,b} w_{foa} e^{-j\Phi_S(f,n,k)}\Big] z_{ol} y_{lk} t_{fk} v_{kn} \right\} \tag{A.103}$$

Although we have included update rules for all CNMF parameters corresponding to optimization of equation A.93, we in fact suggest that it is not necessary to iteratively apply the learning rules for some of the various parameters derived in this section, notably in all cases we suggest not to apply the learning rule for $e^{-j\Phi_S(f,n,k)}$ specified by equations A.102 and A.103.

It was found in practice that monotonic convergence of both cost functions was achieved by enabling only the phase dictionary update rule as described in equation 6.52.

Equations A.102 and A.103 were derived by considering equation A.93 according to the material provided in section A.5. Oddly though, the optimization did not require that even enable both of these since better performance was found in practice by enabling only 6.52.

A.10 Proposed Algorithm Implementation

The computational steps of the proposed algorithm can be described in summary as follows:

1. Compute the quantity (6.53). Do K-means. Save the indicator matrix p_{qn} and the mean vectors $\mathbf{c}_q \in \mathbb{R}^{F\times 1}$.

2. Compute an interchannel quantity based upon the phase part of the look directions tensor $e^{j\Phi_W(f,o,m)}$ (6.39).

$$\tilde{G}_{fo}(b,a) = arg(e^{j\Phi_W(f,o,b)} e^{j\Phi_W(f,o,a)}) \tag{A.104}$$

   for a=1, b=2. Write a code that associates an optimal match (correspondence) between the look directions (o=1,..., O) and the K-means output classes (q=1,...,L) by comparing columns of $[\tilde{G}_{fo}(b,a)].col(o) \in \mathbf{R}^{F\times 1}$ with the computed mean vectors $\mathbf{c}_q \in \mathbf{R}^{F\times 1}$.

3. Initialize the matrix z_{ol} with zeros. Populate z_{ol} with a single 1 per column, according to the computed correspondence. We no longer need to consider updating z_{ol} as a

parameter of the NMF since we propose that using the parameter z_{ol}, populated in this way, the algorithm should already have prior knowledge of the optimal look direction to class correspondence.

4. Populate y_{lk} into a binary partition with a certain number of active components per class, dependent on the total number of classes and components. Example: (7.8).

5. Randomly initialize t_{fk}, v_{kn}, w_{fom}, and $e^{(j\Phi_S(f,n,k))}$ and initialize the iteration counter. Decide upon a target number of desired iterations.

6. Compute the model and its associated error 6.38.

7. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1.

8. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2.

9. Compute the update rule for t_{fk}

10. For all possible interchannel combinations: Compute the interchannel target quantity and the associated error A.89.

11. Compute the update rule for t_{fk} as a function of the interchannel auxiliary function.

12. Compute the model and its associated error 6.38.

13. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1. Compare with the value obtained in step 7 to determine whether the update for t_{fk} increased or decreased the value for the current iteration.

14. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2. Compare with the value obtained in step 8 to determine whether the update for t_{fk} increased or decreased the value for the current iteration.

15. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1.

16. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2.

17. Compute the update rule for v_{nk}

18. For all possible interchannel combinations: Compute the interchannel target quantity and the associated error A.89.

19. Compute the update rule for v_{nk} as a function of the interchannel auxiliary function.

20. Compute the model and its associated error 6.38.

21. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1. Compare with the value obtained in step 15 to determine whether the update for v_{kn} increased or decreased the value for the current iteration.

22. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2. Compare with the value obtained in step 16 to determine whether the update for v_{kn} increased or decreased the value for the current iteration.

23. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1.

24. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2.

25. Compute the update rule for $e^{(j\Phi_S(f,n,k))}$

26. Compute the model and its associated error 6.38.

27. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1. Compare with the value obtained in step 23 to determine whether the update for $e^{(j\Phi_S(f,n,k))}$ increased or decreased the value for the current iteration.

28. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2. Compare with the value obtained in step 24 to determine whether the update for $e^{(j\Phi_S(f,n,k))}$ increased or decreased the value for the current iteration.

29. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1.

30. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2.

31. Compute the update rule for w_{fom}

32. For all possible interchannel combinations: Compute the interchannel target quantity and the associated error A.89.

33. For all possible interchannel combinations: Compute the update rules for w_{fob}, w_{foa} as described.

34. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 1. Compare with the value obtained in step 29 to determine whether the update for w_{fom} increased or decreased the value for the current iteration.

35. Sample (Save) the current value of the sum of squared differences between the observed and modelled data for objective function 2. Compare with the value obtained in step 30 to determine whether the update for w_{fom} increased or decreased the value for the current iteration.

36. Increment the iteration counter and return to step (6), if the target number of desired iterations has not been reached. If it has, stop and exit.

Figure A.4 illustrates graphically the set of steps suggest by the proposed algorithm in applying multiplicative updates. We note that the K-means clustering module is intended to be applied directly on the output of a time-frequency domain processing of the multichannel STFT data. This will be detailed in the sections that follow.

Primarily we suggest only to in fact update the CNMF parameters corresponding to $\mathbf{T}$, $\mathbf{V}$, $e^{j\Phi_S(f,n,k)}$, and the absolute value of $\mathbf{W}$ since the complex part of $\mathbf{W}$ encodes DoA kernel like spatial information. Therefore, the list of steps, although computationally complex, can be summarized compactly within Figure A.4. It will be necessary however, to consider carefully the convergence behaviour of an appropriately configured algorithm configuration. Obviously, mis-configuring a particular module or implementing a particular module in a way that does not exactly correspond to a particular update rule or something having to do with clustering can lead to results that can cause the algorithm to diverge from a desirable source separation result.

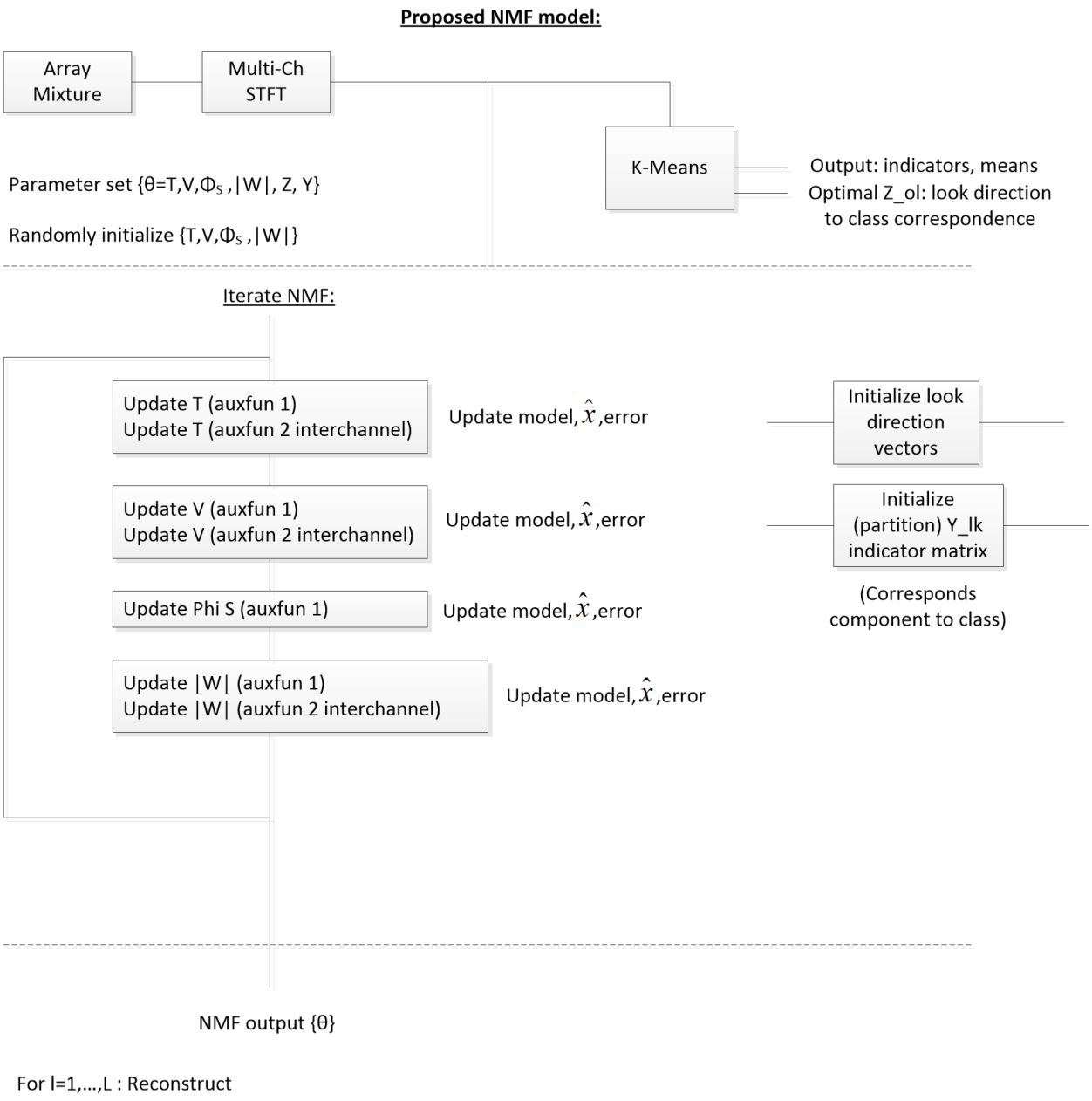

Figure A.4: Proposed algorithm

Appendix B

Further Supplementary Topics

B.1 Discrete Fourier Transform

We here provide a working description of the Discrete Fourier transform (DFT) which computes a linear system transformation upon a discrete time domain signal $x(m)$ of length N and returns a complex valued vector of coefficients $X(k)$ of the same length N that can be used to represent and interpret the signal in the transform domain. This brief description has a corresponding illustration that is depicted in Figure B.1, where the time samples of $x(m)$ are laid out on the left and the frequency samples of the transformed vector quantity $X(k)$ are laid out on the right. We follow the development of the DFT as provided in [30].

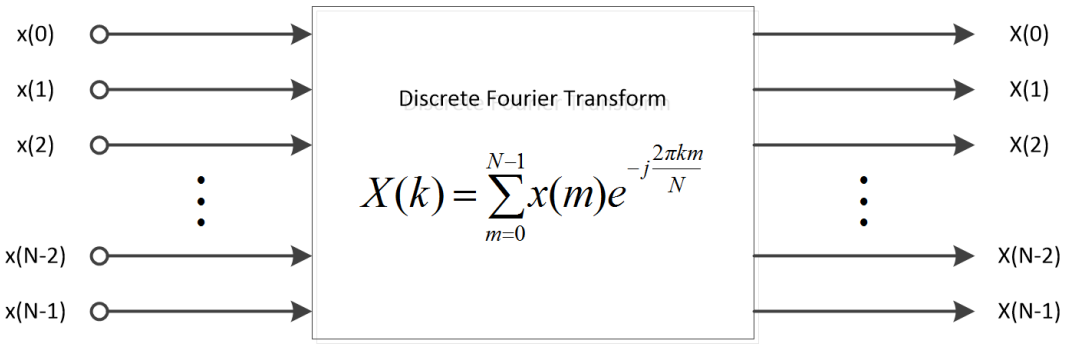

Figure B.1: Discrete Fourier Transform

In the proposed research of the thesis we are interested in computing the DFT on discretized audio signals in order to obtain discretized complex-valued frequency spectra as specified by $X(k)$ where the $X(k)$ are typically laid out as column vectors with the lowest discrete frequency index $k = 0$ being labelled at the top of the column vector and the highest discrete frequency index $k = N - 1$ being labelled at the bottom. These discrete

frequency indices corresponding to different values of k are what are commonly referred to in the thesis as 'frequency bins', signifying that we intend to modify or parametrize a vector in terms of its complex-valued frequency transform domain representation. The DFT coefficients $X(k) \in \mathbb{C}$ for $k = 0 : N-1$ are scalar valued complex quantities encoding amplitude and phase information about the periodicity of the time domain signal $x(m)$ at each discrete frequency k.

To emphasize a so-called matrix interpretation of the DFT we will focus on equations B.3, B.4, and B.5, however we will address equations B.1 and B.2 pertaining to the discrete sampling of continuous time and frequency signals, as required.

In order to convert a DFT representation in terms of DFT coefficients $X(k)$ back into a discrete time series representation corresponding to signal $x(m)$ we 'resynthesize' (reconstruct) the signal $x(m)$ as a according the inverse transformation equation as specified by (B.4).

$$X(f) = \int_{-\infty}^{\infty} x(t) \sum_{m=0}^{N-1} \delta(t - mT_s) e^{-j2\pi ft} dt = \sum_{m=0}^{N-1} x(mT_s) e^{-j2\pi fmT_s} \tag{B.1}$$

$$X(k) = X(f)\delta\Big(f - \frac{k}{N}F_s\Big) \tag{B.2}$$

In principle, the discrete Fourier transform coefficients $X(k)$ is obtained by sampling the continuous Fourier transform $X(k)$ at discrete frequencies $f = kF_s/N$ where k is an integer and F_s is the sampling rate used to discretely sample the continuous time signal $x(t)$

$$X(k) = \sum_{m=0}^{N-1} x(m) e^{-j\frac{2\pi}{N}mk} \qquad k = 0, ..., N-1 \tag{B.3}$$

$$x(m) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}mk} \qquad m = 0, ..., N-1 \tag{B.4}$$

The signal $x(m)$ is reconstructed from coefficients $X(k)$ according to the inverse DFT equation as specified in equation B.4.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-2) \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{4\pi}{N}} & \cdots & e^{-j\frac{2(N-2)\pi}{N}} & e^{-j\frac{2(N-1)\pi}{N}} \\ 1 & e^{-j\frac{4\pi}{N}} & e^{-j\frac{8\pi}{N}} & \cdots & e^{-j\frac{4(N-2)\pi}{N}} & e^{-j\frac{4(N-1)\pi}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & e^{-j\frac{2(N-2)\pi}{N}} & e^{-j\frac{4(N-2)\pi}{N}} & \cdots & e^{-j\frac{2(N-2)(N-2)\pi}{N}} & e^{-j\frac{2(N-2)(N-1)\pi}{N}} \\ 1 & e^{-j\frac{2(N-1)\pi}{N}} & e^{-j\frac{4(N-1)\pi}{N}} & \cdots & e^{-j\frac{2(N-1)(N-2)\pi}{N}} & e^{-j\frac{2(N-1)(N-1)\pi}{N}} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-2) \\ x(N-1) \end{bmatrix} \tag{B.5}$$

The technical details of the DFT and its implementation are based therefore primarily upon these equations. In principle a fast Fourier transform (FFT) will replace the DFT in real implementations, however the FFT is technically supposed to exactly implement the same result as the DFT in outputting the vector of discrete frequencies $X(k)$ for $k = 1, ...N-1$. The primary implementation of FFT software commonly used in most applications is based upon the implementation by Frigo and Johnson as described in [72] and is known by the name of the FFTW3 library.

Therefore, STFT implementations typically rely in fact upon first simply windowing a particular segment corresponding to the signal $x(m)$ and then calling the FFTW3 software to compute the output DFT coefficients. Therefore, the choice of STFT parameters is typically the main question to consider when considering how the DFT (i.e. FFT) was actually applied to generate coefficients $X(k)$.

B.2 Accurate Representation and Modelling of Single Channel and Multichannel Sound Mixtures

B.2.1 Fourier Series: Representation of Periodic Signals

In digital signal processing, the analysis and synthesis of audio signals is greatly interconnected with the use of Fourier series and Fourier transform based methods for analyzing and representing a decomposible audio signal as a sum of periodic components each characterized by their fundamental frequency.

We shall first note the following types of signals

$$x_1(t) = \cos w_0(t) \tag{B.6}$$

$$x_2(t) = \sin w_0(t) \tag{B.7}$$

$$x_3(t) = \cos w_0(t) + j \sin w_0(t) = e^{jw_0(t)} \tag{B.8}$$

can be considered the basis functions of the Fourier transform [30], and any audio signal that can be approximated by a linear combination of these functions, is in theory approximately representable by computing its Fourier transform. This forms the basic but nonetheless immediately useful starting point of digital signal processing, where basic principle of the conversion of continuous time signals, modelling physical phenomena such as audio signals is a key principle to developing DSP algorithms [73] in terms of discrete-time signals and systems. In section B.1 of the appendix we provide a treatment of the DFT and in the next section we introduce the discrete time STFT for analyzing and encoding time-frequency representations of digital signals.

For the current section we assume that the reader has a basic familiarity with at least the concepts contained within a reference such as [73] in order to benefit from and grasp the following illustrations. Therefore we take the opportunity to continue to discuss equations B.6, B.7,B.8 focusing on how analysis of Fourier series and Fourier transform based methods that encode periodic, phase and amplitude information may be applied towards the notion of matrix algorithms and in particular source separation algorithms.

We thus then also consider the case where the functions $x_1(t)$, $x_2(t)$, $x_3(t)$ correspond to possibly scaled and delayed versions of the periodic basis functions, for instance such that

$$x_1(t) = A\cos(\omega_0 t + \phi_d) \tag{B.9}$$

$$x_2(t) = A\sin(\omega_0 t + \phi_d) \tag{B.10}$$

$$\begin{aligned} x_3(t) &= Ae^{j(\omega_0 t + \phi_d)} \\ &= Ae^{j\phi_d}e^{j\omega_0 t} \end{aligned} \tag{B.11}$$

where the resulting scaled and delayed functions are still considered to be periodic after applying arbitrary scaling and delays. We note that for these types of functions the Fourier transform is also capable of encoding the scaling factor A and the delay factor ϕ_d (in radians) associated with each function. We propose that the decomposition of a real audio

signal provided by the analysis of the signal via the DFT will result in discretized frequency components most resembling the type specified by equation B.11, where the term outside the complex exponential function $Ae^{j\phi_d}$ is to be considered as the *coefficient* of the frequency dependent complex exponential basis function $e^{j\omega_0 t}$.

We emphasize that the additivity of complex coefficients of the form

$$A_1 e^{j\phi_{d,1}} + A_2 e^{j\phi_{d,2}} = \text{real}\left\{A_1 e^{j\phi_{d,1}} + A_2 e^{j\phi_{d,2}}\right\} + j\,\text{imag}\left\{A_1 e^{j\phi_{d,1}} + A_2 e^{j\phi_{d,2}}\right\} \tag{B.12}$$

correspond to complex additions that can be interpreted using vector representations in the complex number 'plane' that are separate in terms of the 'real' and 'imaginary' components of the complex numbers to be added. Precisely, equation B.12 corresponds to the addition of two complex scalars and the result is quite simply a complex scalar. This implies however that the addition of two signals characterized by *the same fundamental frequency* but treated as *separate* periodic components (i.e. each with a particular amplitude factor and a particular phase factor), has a corresponding simple but important numerical interpretation in terms of the addition of their *complex coefficients* in the complex plane.

In studying the STFT representation of a real audio signal we will obtain a *matrix* representation specifying the complex coefficients that describe the signal across all possible time and frequency 'bins' of the signal which demonstrates that an audio signal in the STFT domain corresponds to a generated dataset that is extremely information-rich in nature. In considering a real audio signal (one that is the result of a recording from a real acoustic environment) we will begin to consider concepts such as how physical distances, filtering of source signals across physical distances, and relative positions of both microphones and sources are factors that may have contributed to the resulting observed signal, and whether or not the factors are quantities that can be to some degree be parametrized and used to explain the STFT matrix (coefficient) representation of an observed audio signal.

We now engage in a description of how additivity of complex-valued coefficients (viewed as mixtures corresponding to parametrized and additively occurring sound components) will be relevant to formulating and understanding algorithms for applying source separation methods with regards to *spatially* occurring sound mixtures. In this thesis, we propose that we will be able to apply and analyze all the information in terms of a mixture of these coefficients per time frequency bin, per microphone channel, and per source. The source information, unfortunately, will not be available to us, and will be something that we will have to parametrize and can only check that we have estimated reasonably well at the completion of all NMF iterations of the proposed algorithm. We will have multiple

channels of (spatially non-redundant) observed microphone signals available to us (that may be to some degree, *spectrally* redundant) that can be readily analyzed in terms of time and frequency (via the STFT) but no structured information pertaining to *exactly how* the observed mixtures were generated from the true source signals will yet be available to us. Between iterations, however, our parametrization of the time, frequency, and interchannel parameters must continuously begin to 'fit' the *observed data* (the multichannel STFT) to our *parametrization* which we intend will gradually converge towards a representation that approximates the observed data more closely at each iteration of the algorithm, than at the previous iteration. Hence, abstractly, we would ideally accept a parametrization that both fits the observed data well and *simultaneously* parametrizes the true (but unknown) sources well, however we would have to reject:

- A representation that poorly fits the observed data.
- A representation that fits the observed data well but poorly explains the appropriate spatial configuration that generated the observed data.

We will parametrize each discrete-time observed microphone signal via its STFT representation in the time frequency domain as an acoustic mixture of spatially superimposed sound sources. And we have hinted here that there exists a spatial mixing (filtering, or *multiplicative*) aspect of the model that necessarily governs the contribution of so called 'spatial components' that occur *additively* at each time frequency bin depending on how active each source is at the particular time frequency bin, that will be captured and quantified at the same time-frequency bin, as seen by each microphone of the microphone array, and the microphone's observed STFT coefficient at that time-frequency bin.

In other words, for each microphone and source pair, there exists a spatial image (generated from spatially filtering the desired but unknown source signal) that contributes additively to each microphone, thus impacting the observed coefficients at that particular microphone. Depending on the amount of spectral information present in the source signal at a particular time-frequency region, the observed microphone coefficients should certainly convey corresponding (but potentially 'filtered') spectral information at the same corresponding time-frequency region, but in a multi-source mixture scenario this is subject to the possibility that interfering sources may also contribute interfering spectral information additively at the same time frequency region. Thus, the possibility of *competing* source signals whose corresponding spatial images contribute additively to resulting observed microphone STFT coefficients will be a major challenge for the proposed algorithm to deal with.

In summary, we re-emphasize that we have proposed that each observed microphone's STFT can be explained as an additive spatial mixture of a time and frequency dependent

matrix of coefficients, summed over all possible sources. And we propose that by adaptively modifying our parametrization of the mixture at each iteration of the algorithm, we will obtain a parametrization that both fits the observed data and simultaneously parametrizes the true (but unknown) sources well.

We intend to develop and to extend existing audio NMF based time-frequency and source separation learning techniques in order to exploit and parametrize our model of the observed mixture signals in terms of the principle of additivity of complex (amplitude and phase) STFT spectral characteristics between sources at each microphone, whose short-time spectral and temporal behaviour we have proposed can be sufficiently explained in terms of the additivity of STFT coefficients that correspond to basic periodic functions that have only three key parameters (amplitude factor, phase factor, and discrete frequency index) that are truly needed to describe the corresponding periodic (i.e. frequency dependent) representation of the observed and/or and possibly parametrized signals accurately.

If we reconsider equation B.12 describing complex additivity of only *two* separate coefficients and re-write the equation in the same way

$$A_1 e^{j\phi_{d,1}} + A_2 e^{j\phi_{d,2}} = A_0 e^{j\phi_{d,0}} \tag{B.13}$$

but assign the labels A_0 and $\phi_{d,0}$ to the resulting amplitude and scaling factor of the *output* coefficient, $A_0 e^{j\phi_{d,0}}$, then we can emphasize another possible view of the frequency domain source separation problem that we now would like to illustrate. We have written equation B.13 in this way specifically to illustrate, and to propose (by analogy) that we will have to *infer* the quantities A_1, A_2, $\phi_{d,1}$, and $\phi_{d,2}$ based on only the *observed* coefficient $A_0 e^{j\phi_{d,0}}$ *and possibly* any other info that we can gather about how to adequately *parametrize* the listed quantities. In the discussion that follows we also re-emphasize that the problem of 'fitting' the left hand side of equation B.13 to its right hand side has an equivalent vector addition interpretation as illusrated by the way it is written in equation B.12 and if we say that $A_0 e^{j\phi_{d,0}}$ corresponds to the sum of the two vectors then explaining the observed quantity $A_0 e^{j\phi_{d,0}}$ can be visualized as the fitting of two *vectors* in terms of their real and imaginary parts, in the complex number plane. The resulting sum of the two *parametrizable* vectors that we must necessarily in some way *parametrize* shall then be fitted (i.e. added together) in such away that corresponds to the output (observed) quantity $A_0 e^{j\phi_{d,0}}$.

If we imagine that the terms $A_1 e^{j\phi_{d,1}} + A_2 e^{j\phi_{d,2}}$ correspond to STFT coefficients of unknown *frequeny-dependent spatial components* that must be *analogously parametrized* by our algorithm *at a particular time-frequency bin*, then we emphasize that the algorithm will be charged with the optimization task of 'fitting' the spatial component parametrization to the actual observed quantity $A_0 e^{j\phi_{d,0}}$, where the observed quantity could be considered as by the

analogy to be an observed microphone STFT coefficient at the *same* time-frequency bin. In practice, and within the fully developed algorithm, this might happen to be achieved utilizing a combination of various techniques, but primarily the adequate parametrization of the spatial components shall depend upon both iterative learning of the model parameters via NMF (and CNMF update rules) as well as clustering techniques (pertaining to information about the spatial parametrization) defined upon the model parameters.

Effectively, then to summarize the use of the analogy described by equation B.13 in a source separation context only once more, we *must* treat the right hand side as an *observed* quantity that we must consistently strive to adequately explain, and the left hand side of the equation as *parameters* whose correct configuration we must adequately *infer*. In other words, we will have to *infer* the appropriate *decomposition* that we believe caused the *observed* coefficient(s) in order to achieve a plausible source separation result, described in terms of the source STFT coefficients, and their time and frequency dependent spatially occurring STFT parametrization, describing them as additive audio components, that we will seek to fully parametrize.

B.3 STFT Overview

A key and central focus of the thesis will be to study the application of STFT based signal processing methods for analyzing spatially occurring speech and music signals within a typical acoustic room environment.

In studying the short time behaviour of speech and musical signals, we will expect the STFT to be able to provide us with a frequency dependent parametrization of the signals. It will be shown that there exist adequate models for speech and music that focus on modelling for instance, the *spectral envelope* of speech and music signals within a short-time window of analysis (i.e. an STFT frame). Furthermore, we expect it to encode key temporal-dependent and frequency-dependent amplitude and phase information at each time-frequency bin of the STFT analysis.

In principle, and within this thesis when we specifically refer to *spatial filtering* we are referring to the geometry of a *room environment*, and the acoustic path whose frequency domain interpretation could be described in terms of a *transfer function* that we propose shall directly impact, how signals are combined additively (if multiple signals are present within the room environment) as seen correspondingly, by a multiple microphone array. A room environment in audio signal processing can often be described in terms of a room impulse response, where the direct path and reverberant path are combined in order to fully specify the room impulse response. In this thesis we shall mainly consider the effect of the

direct path, in formulating and effectively constructing data sets that we allegedly propose that the proposed algorithm shall be capable of solving.

Therefore, a high level description of the source separation that we aim to solve shall refer to classes of signals that correspond to either musical or human *speakers* that would be physically located within a typical room environment, and whose spatial positions do not geometrically change for the duration of the signals do be analyzed, since we propose that the algorithm will be able to in fact parametrize, on a source by source basis, the appropriate *room impulse response* that should be associated with a particular source.

In order to further simplify the problem of developing spatial filtering parametrization techniques we will furthermore require that the constructed signals correspond to signals for which the room has the property of being *anechoic*, that is we do not attempt to model the reverberant path of the impulse responses, only the direct paths.

With this high level description of the source separation problem in place, we will then state the definition of the well known STFT, that effectively applies a DFT analysis in each frame (i.e. column) of the time-frequency matrix, which we intend to apply to real (but spatially filtered) audio signals.

The discrete STFT of a discrete-time signal $x[n]$ is given by

$$X[rT, k] = \sum_{n=0}^{N-1} x[rT + n]w[n]e^{-2i\pi \frac{kn}{N}} \tag{B.14}$$

where T is a hop size defined in terms of a certain number of samples, r is the frame index value that indexes the hop size parameter T in order to shift the window of analysis (i.e. the starting sample of the frame as determined by the value of rT), and n is a variable used to convert the representation of the time-domain signal $x[n]$ from a discrete time sequence to a frequency domain set of coefficients via a discrete Fourier transform (DFT) operation. The variable k specifies the resulting discrete frequency index obtained per time frame r by computing the DFT per analysis window. The reader can be pointed to the appendix to consider a separate treatment of the DFT B.1

Note that equation B.14 can sometimes be referred to as

$$X[r, k] = \sum_{n=0}^{N-1} x[rT + n]w[n]e^{-2i\pi \frac{kn}{N}}. \tag{B.15}$$

where the previous description still applies exactly, however we drop the notion of externally specifying the hop size T, and keep it only internally as a technical detail. We then focus only on considering the STFT matrix $X[r, k]$ in terms of the variables r and k, the frame and discrete frequency index, respectively.

Conceptually, to generate in the forward direction the STFT coefficient matrix ($X[r,k] \in \mathbb{C}^{RxK}$) we partition the time-domain input signal $x[n]$, window each partition using a window function $w[n]$, and perform elementwise multiplication and accumulation of the windowed time-domain input signal and the discrete-time complex exponential signal $e^{-2i\pi\frac{kn}{N}}$ at a certain frequency corresponding to the frequency index variable k. We also note that the number of discrete frequency points that we obtain will correspond to the number chosen as the DFT size, which is here chosen as N.

Once the information rich signal is converted to the STFT domain the understanding is that signal information (within the time window corresponding to each time-bin) is now encoded by a complex vector of amplitudes and phases, each vector element being characterized by its corresponding frequency bin k and time (frame) bin r.

The discrete short-time Fourier transform (STFT) can also be specified in terms of a 1 sample hop size as

$$X[l,k] = \sum_{n=0}^{N-1} x[l+n]w[n]e^{-2i\pi\frac{kn}{N}}. \tag{B.16}$$

where rT is replaced by l and l indexes the samples of the time-domain signal $x[n]$. The 1 sample hop size STFT as specified by B.16 is usually provided in signal processing literatures [73,74] explaining the STFT alongside the conventional STFT equation B.14, but is typically not as useful in practice as the conventional equation.

The time domain signal $x[n]$ can be reconstructed from its STFT $X[rT,k]$ via the inverse STFT equation which is given by

$$x[n] = \sum_{r}\sum_{k=0}^{N-1} X[rT,k]w'[n-rT]e^{2i\pi\frac{kn}{N}}$$

with the reconstruction condition that helps to define the inverse window function $w'[n]$ as given by

$$\sum_{r} w'[n-rT]w[n-rT] = 1$$

.

B.4 Optimal Filtering and Signal Separation Algorithms

We now consider the development of the Wiener filter and some of its usages in both signal extraction and signal modelling applications. Signal extraction and signal modelling are both of practical interest in blind source separation problems since, in principle, they both require the parametrization of an arbitrary signal or signal component in terms of unknown quantities that typically must be *estimated*. The Wiener filter is known to be able to provide a minimum mean square error (MMSE) estimate of a *desired* signal typically based solely on the modelling of an observed signal that is proposed to have been generated from the output of FIR (finite impulse response) filtering with respect to an unknown set of filter coefficients that must be obtained algorithmically.

B.4.1 Wiener Filter

B.4.1.1 FIR Formulation

The Wiener filter effectively represents one of the current greatest and most powerful innovations in being able to statistically and optimally separate a target signal from noise and/or interferences and is used as a nearly ubiquitous tool in the audio signal processing field and other signal processing fields of research and applications, especially in the study of adaptive filtering [75].

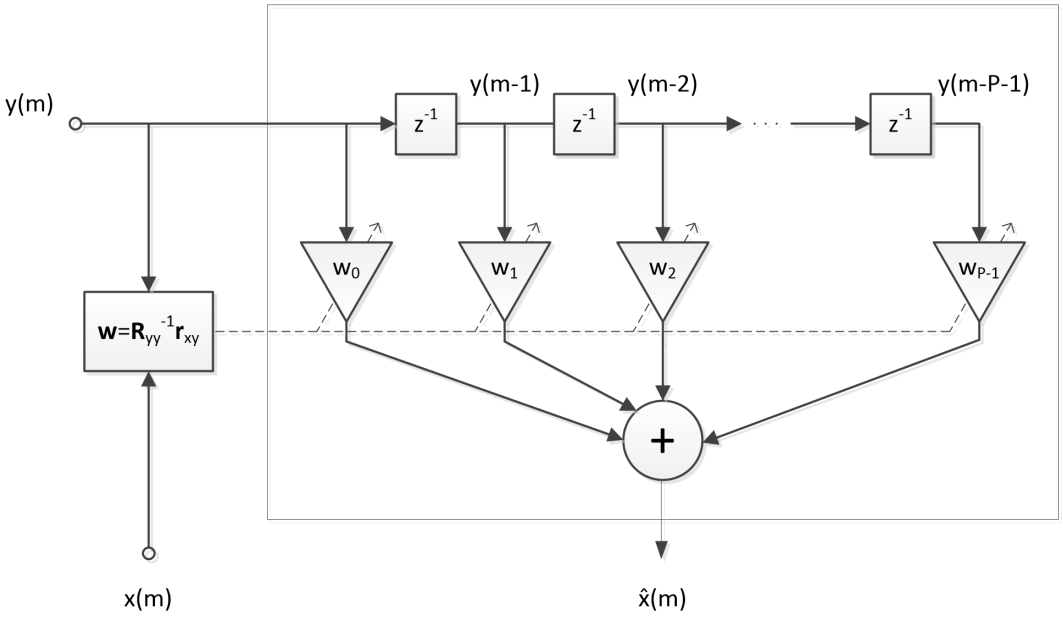

Figure B.2: FIR Wiener Filter

We introduce the Wiener filter according to its derivation as described within [30] and we begin with its mathematical description, by presenting its input-output relation as given by

$$\begin{aligned}\hat{x}(m) &= \sum_{k=0}^{P-1} w_k y(m-k) \\ &= \mathbf{w}^T \mathbf{y} \end{aligned} \tag{B.17}$$

which simply represents an FIR filtering scenario; where the quality of the estimated output signal is parametrized by the quality of the estimate of the optimal filter $\mathbf{w}$. Figure B.2 pictorially demonstrates this input-output relation, in terms of a transversal FIR filtering of the observed signal $y(m)$, by a set of weights $w_0, w_1, ..., w_{P-1}$.

A few things to interpret about the transversal filter are as follows:

1. The first is that we readily and typically have two time domain data quantities available to us in testing the Wiener model, $x(m)$ and $y(m)$. Hopefully, it is evident that for each output sample $x(m)$ we have at least P samples of $y(m)$ (i.e.: P "past" or "delayed" samples, relative to the current mth sample) to carry out the filtering operation, and that the filter length associated with $\mathbf{w}$ is also equal to P. In practice, we often would like to model more than just one sample of the output sequence at a time, and therefore we can assume that $x(m)$ will be represented not just for some arbitrary m but for a

sequence of samples $m = 1, ..., N$ where N is some total number of target samples to be modelled.

2. As mentioned, $\mathbf{w}$ is not readily available to us, and represents a parameter of the model that must be estimated for each sample at index m for which we would like to compute an estimated output sample $\hat{x}(m)$. Therefore an "update rule" for $\mathbf{w}$ (so to speak) must be derived in order to estimate the optimal sequence of filter samples, to be stored within the vector $\mathbf{w}$. This is what the derivation of the Wiener filter attempts to provide and what the Wiener filter, in its convergence, seeks to do as a primary goal or objective.

3. One could argue that the statistical grounds upon which the Wiener filter stands is fairly robust and well-developed compared to NMF. This is pointed out before moving on by taking a step back and looking at the fact that we have two available quantities $x(m)$ and $y(m)$, where the $x(m)$ can be arranged for $m = 1, ..., N$ as a vector quantity, $\mathbf{x}$, and the $y(m)$ can be arranged as a matrix quantity, $\mathbf{Y}$, taking into the account the possible FIR delay shifts as a function of $p = 1, .., P$. Due to its FIR structure, the matrix $\mathbf{Y}$ contains partially redundant information in traversing its rows, as will be seen. We can consider that it is in fact this redundancy of the matrix $\mathbf{Y}$, that should make a plausible solution for the unknown set of filter coefficients in practice more tractable and in principle also more meaningful as well.

For a sequence of observed samples $y(m)$ for $m = 0, .., N - 1$ to be filtered, the Wiener filter error signal $e(m)$ is defined as the difference between the desired signal $\mathbf{x}$ and the filtered estimate signal $\hat{\mathbf{x}}(m)$ as given by

$$\mathbf{e} = \mathbf{x} - \mathbf{Y}\mathbf{w} \tag{B.18}$$

where $\mathbf{e} \in \mathbb{R}^{N\times 1}$, $\mathbf{x} \in \mathbb{R}^{N\times 1}$, $\mathbf{Y} \in \mathbb{R}^{N\times P}$, $\mathbf{w} \in \mathbb{R}^{P\times 1}$ and we note that equation B.18 can be written in a more explicit and detailed manner as given by

$$\begin{bmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{bmatrix} - \begin{bmatrix} y(0) & y(-1) & y(-2) & \cdots & y(1-P) \\ y(1) & y(0) & y(-1) & \cdots & y(2-P) \\ y(2) & y(1) & y(0) & \cdots & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & y(N-3) & \cdots & y(N-P) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{bmatrix} \tag{B.19}$$

We can further note that as we traverse the rows of the matrix equation as given by B.19, the current mth row of the matrix $\mathbf{Y}$ is augmented with m new samples, and the prior/past samples of the observed signal $y(m)$ for $m = 0, ..., 1-P$ are shifted to the right. Because of this it can be understood that the matrix $\mathbf{Y}$ therefore compactly implements FIR filtering on a *row* by row basis based on the available data contained within the matrix $\mathbf{Y}$ and the data to be modelled, arranged within the vector $\mathbf{x}$.

We now consider the statistical derivation of the optimal Wiener filter solution for obtaining the vector of filter coefficients $\mathbf{w}$ as outlined in [30]. In order to define a *single* criterion that ensures that the observed error index at each leading sample m for $m = 1, ..., N-1$ at each row is to be *minimized* the Wiener filter implies that the statistical average of the error signal, be defined as the mean squared average of the error signal $e(m)$ and correspond to the Wiener criterion as given by

$$\begin{aligned} \mathbb{E}[e^2(m)] &= \mathbb{E}[(x(m) - \mathbf{w}^T\mathbf{y})^2] \\ &= \mathbb{E}[x^2(m)] - 2\mathbf{w}^T\mathbb{E}[\mathbf{y}x(m)] + \mathbf{w}^T\mathbb{E}[\mathbf{y}\mathbf{y}]^T]\mathbf{w} \\ &= r_{xx}(0) - 2\mathbf{w}^T\mathbf{r}_{\mathbf{yx}} + \mathbf{w}^T\mathbf{R}_{\mathbf{yy}}\mathbf{w} \end{aligned} \tag{B.20}$$

where here, $\mathbf{y}$ corresponds to the mth row of the matrix $\mathbf{Y}$ and $\mathbb{E}[\cdot]$ corresponds to the *mathematical expectation* operator.

From equation B.20 it can be said that since the mean square error criterion $\mathbb{E}[e^2(m)]$ is a quadratic function of the FIR filter $\mathbf{w}$, then an optimal set of filter coefficients can be found that corresponds to minimizing the criterion at a single minimum point.

Thus it can be shown that the *gradient* with respect to the criterion corresponds to the following quantity as given by

$$\frac{\partial}{\partial \mathbf{w}}\mathbb{E}[e^2(m)] = -2\mathbf{r_{yx}} + 2\mathbf{w}^T\mathbf{R_{yy}} \tag{B.21}$$

and where we provide an appropriate treatment of scalar valued functions of vector valued parameters in the appendix in section A.2.5. By setting equation B.21 equal to zero, it can be shown that the MMSE estimate filter occurs when the vector $\mathbf{w}$ is given by

$$\mathbf{w} = \mathbf{R}_{\mathbf{yy}}^{-1}\mathbf{r_{yx}} \tag{B.22}$$

where $\mathbf{R_{yy}} \in \mathbb{R}^{P\times P}$ is a symmetric Toeplitz matrix representing the autocorrelation matrix of the observed signal $\mathbf{y}$ and where where the correlation values $r_{yy}(k)$ are represented by the elements along the first row or first column of the matrix and can be computed according to

$$r_{yy}(k) = \frac{1}{N}\sum_{m=0}^{N-1} y(m)y(m+k) \qquad \text{for } k = 0, ..., P-1. \tag{B.23}$$

The cross correlation vector $\mathbf{r_{yx}}$ in B.22 corresponds to a set of cross correlation values computed at lag k between the observed signal $\mathbf{y}$ and the desired signal $\mathbf{x}$. Its elements arranged within a *column vector* and whose values can be computed as given by

$$r_{yx}(k) = \frac{1}{N}\sum_{m=0}^{N-1} y(m)x(m+k) \qquad \text{for } p = 0, ..., P-1 \tag{B.24}$$

which hints that if we intend to apply the Wiener filter in such a way that requires the cross correlation vector $\mathbf{r_{yx}}$ to be computed, then we must arrange the problem in such a way that the desired signal $\mathbf{x}$ corresponds to a signal that is also physically available, just as we assume $\mathbf{y}$ is.

Reconsidering now equations B.18 and B.19 we should be able to conclude that the Wiener filter has a useful structure that can in fact exploited and demonstrated as a useful tool in many different signal processing applications, depending on what the observed signal $\mathbf{y}$ and desired signal $\mathbf{x}$ are chosen to be, respectively. It plays a central role in key signal processing technologies such as system identification, echo cancellation, signal restoration, channel equalization, radar, and linear prediction modelling of digital signals [30]. In the current chapter the Wiener filter will be revisited briefly in order to show its use in the form of linear prediction models for modelling the short time behaviour of speech and also music signals.

One last interesting thing to note about the Wiener filter is that depending on the chosen dimensions of equation B.18, (i.e. the chosen filter length P and the number of observed

samples to be modelled N) there also exists a corresponding *geometrical interpretation* of the error signal $e(n)$ in terms of an N-dimensional vector space.

A pictorial demonstration of the graphical interpretation is provided in Figure B.3 of Appendix section B.5.

According to the interpretation, the estimated signal vector $\hat{\mathbf{x}}$, which lies in an N-dimensional vector space, as do the vectors $\mathbf{e}$, $\mathbf{x}$, and the columns of $\mathbf{Y}$, $\hat{\mathbf{x}}$ is the orthogonal projection of the clean signal vector $\mathbf{x}$ onto the subspace spanned by the vectors

$$\{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{P-1}\} \tag{B.25}$$

where we consider $\mathbf{y}_p$ to be the column vector obtained by sub-indexing the matrix $\mathbf{Y}$ at column p.

B.5 Wiener Filter Projection Interpretation

Figure B.3 details an orthogonal projection interpretation of the MMSE signal $\hat{x}_m$ as demonstrated within section B.4.1.1 and as depicted within [30].

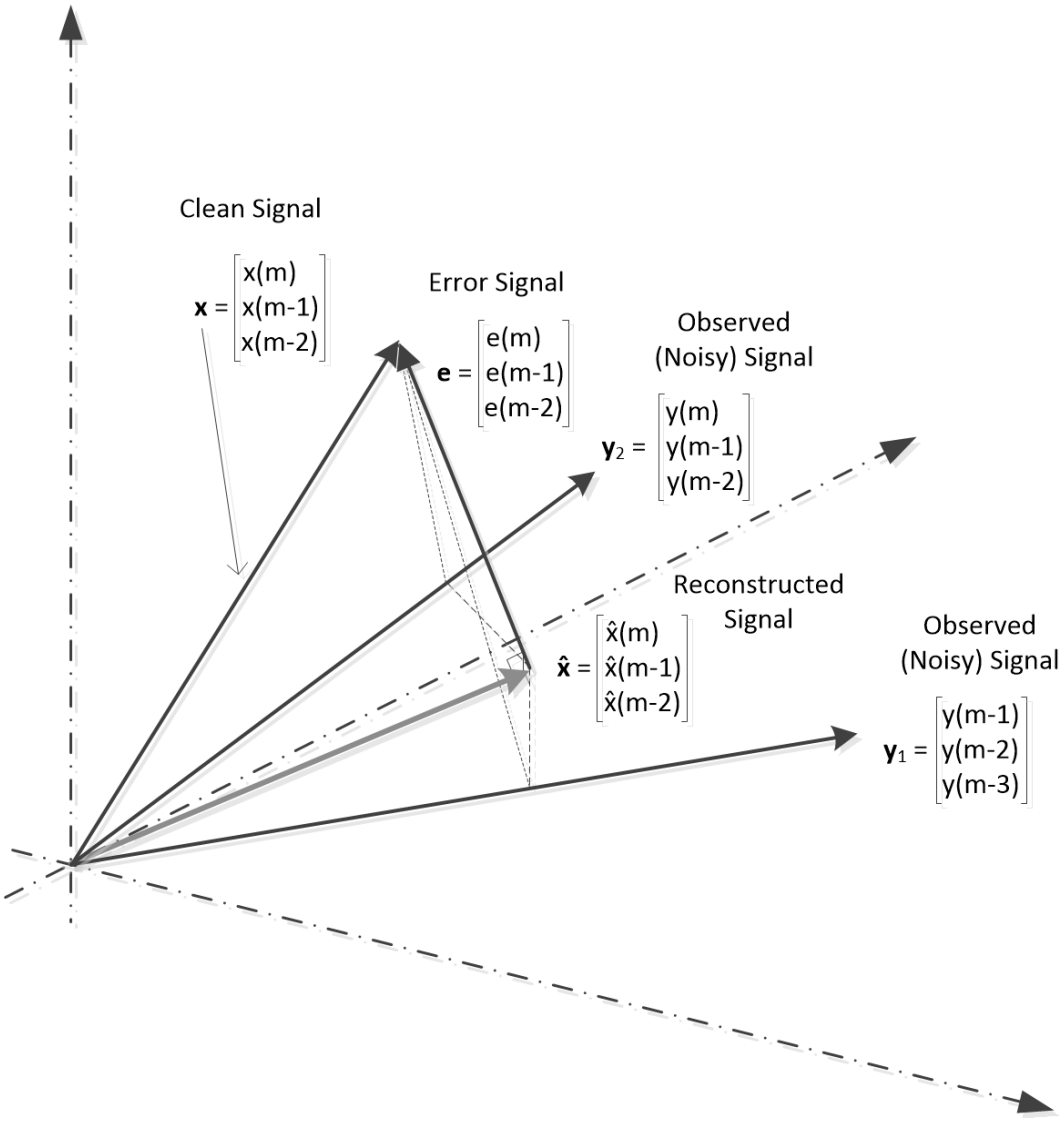

Figure B.3: Wiener Filter Orthogonal Projection

Appendix C

SCM NMF Supplementary Modules

C.1 EU-SCM NMF cont'd

C.1.1 Bottom-Up Clustering Procedure by Merging of Classes

The following similarity criterion between matrices $\mathbf{H}$ sub-indexed at different frequencies f and classes l was suggested should be applied within a type of *bottom up* clustering module that still was intended to iteratively apply learning rules 5.28 to 5.31, but following having let the CNMF algorithm converge to an *intermediate* overall parametrization of the model.

A similarity criterion $d_H(l_1, l_2)$ between matrices of different classes was defined as given by

$$d_H(l_1, l_2) = \sum_{f=1}^{F} ||[\mathbf{H}_{fl_1}] - \mathbf{H}_{fl_2}||_F \tag{C.1}$$

where in [16] it was suggested that an *interleaved* clustering should be applied that enforced the CNMF parameters $\mathbf{Z}$ and $\mathbf{H}$ to have more of a desirable configuration for the purpose of source reconstruction of spatial images corresponding to source estimates. Minimizing the criterion $d_H(l_1, l_2)$ at all frequencies $f = 1, ..., F$ could then be considered as a K-means distortion-like criterion in the context of matrices $\mathbf{H} \in \mathbb{C}^{M \times M}$ with cluster indicators specified by the variable z_{lk}.

The details of the *bottom up* procedure are more precisely as within [16], in summary it can be understood that any such modification of the indicator matrix $\mathbf{Z}$ must in some way appropriately modify the corresponding matrix $\mathbf{H}$ since the matrix $\mathbf{H}$ may be sensitive to modification of $\mathbf{Z}$ and vice versa. In [24] a set of *DoA kernel matrices* were used to encode the matrix $\mathbf{H}$ with more spatial (i.e. frequency dependent phase) information, and essentially encoded both TDoAs (time difference of arrivals) as well as angle of arrivals into the representation of matrices corresponding to $\mathbf{H}$, and thus clustering could be performed within a

K-means like manner instead of the 'bottom up' manner described here, and therefore, no such bottom up clustering module was at all in fact required. Clustering could be performed entirely following applying iterative CNMF learning rules. Thus the DoA algorithm was considered a direct extension of the current algorithm which explains why its structure is consequently similar to the algorithm discussed here.

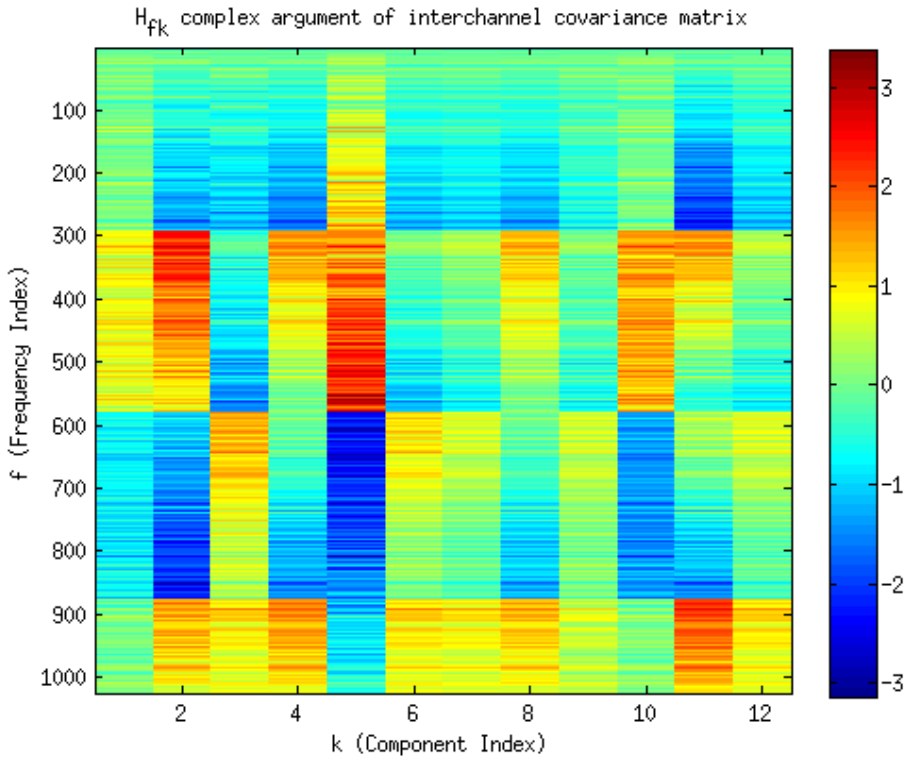

Figure C.1: Example of learned spatial properties represented as $\arg([\mathbf{H}_{fk}]_{12})$ corresponding to the phase difference between microphones for each frequency f and NMF basis k

Figures C.1 to C.3 illustrate a K-means interpretation of the clustering that the EU-NMF SCM algorithm incorporates and proposes that the bottom-up clustering procedure should output. The K-means (with L clusters) interpretation can be summarized according to the notation

$$\{\arg([\mathbf{H}_{fl}]_{12}, z_{lk}\} \leftarrow \{\arg([\mathbf{H}_{fk}]_{12})\} \tag{C.2}$$

where the clustering procedure according to the K-means algorithm (and a convention to describe it in summary) was discussed according to equation 3.11. These figures, however, were not the result of an actual implementation of the EU-NMF SCM algorithm, but are

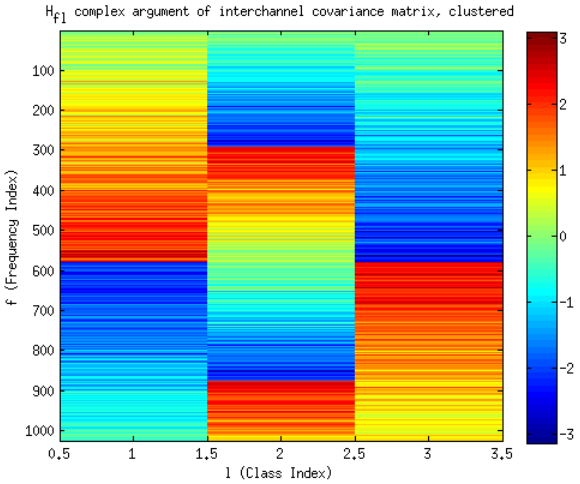

Figure C.2: Clustered phase of SCM NMF Parameter $\mathbf{H}_{fl}$ represented as $\arg([\mathbf{H}_{fl}]_{12})$

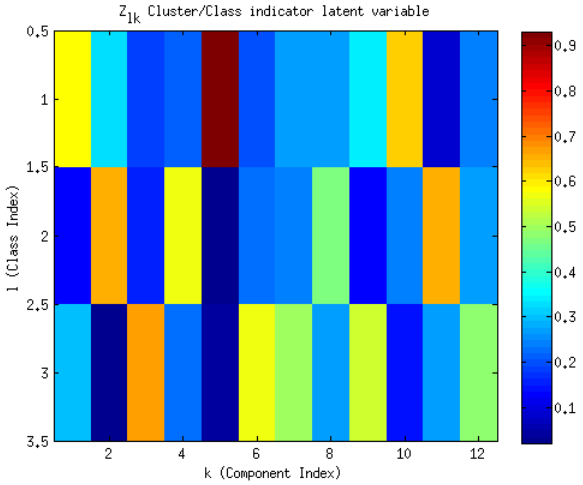

Figure C.3: Cluster indicator latent variable and NMF parameter z_{lk}

simply corresponding to an illustration that was constructed to convey the concept, similar to the illustration that was provided in [16].

C.1.2 Source Reconstruction

$$[\tilde{\mathbf{y}}_{fn}^{(l)}]_m = \frac{[[\mathbf{H}]_{fl}]_{mm} \sum_{k=1}^{K} z_{lk} t_{fk} v_{kn}}{\sum_{l=1}^{L} [[\mathbf{H}]_{fl}]_{mm} \sum_{k=1}^{K} z_{lk} t_{fk} v_{kn}} [\tilde{\mathbf{x}}_{fn}]_m \tag{C.3}$$

$$\tilde{\mathbf{y}}_{fn}^{(l)} = \left(\sum_{k=1}^{K} z_{lk} t_{fk} v_{kn} \right) [\mathbf{H}]_{fl} [\hat{\mathbf{X}}]_{fn}^{-1} \tilde{\mathbf{x}}_{fn} \tag{C.4}$$

The intepretation of the source reconstruction step as detailed by equations C.3 and C.4 can be considered as follows:

- Equations C.3 and C.4 correspond to the reconstruction steps for the Euclidean distance based algorithm and Itakuro-Saito based algorithm, respectively. These algorithms were separately derived using the respective EU and IS cost functions, and provided separate multiplicative update rules in the SCM-CNMF parameters. The EU-NMF update rules correspond to equations 5.28 to 5.31.

- The reconstruction of equation C.4 corresponds to a multichannel Wiener filter source reconstruction technique similar to that described according to equation 2.41 which was

described as being able to recover spatial images of source estimates in multichannel and multi-source context.

- The observed multichannel STFT vector $\mathbf{x}_{fn}$ carries initial phase information into the lth estimated spatial image vector $\tilde{\mathbf{y}}_{fn}^{(l)}$ at time-frequency bin f-n.

- Equation C.3 details the extraction of spectral amplitude (square rooted power) information about the lth estimated spatial image at time frequency and channel bins specified by f, n and m. The extraction of the lth spatial image is dependent primarily upon on the microphone dependent power gain specified by the term $[[\mathbf{H}]_{fl}]_{mm}$, while the time-frequency dependent information is extracted by the mapping of components to classes as specified primarily by the weighting of provided by the term z_{lk} which appears in the numerator of C.3.

- In the denominator of C.3 all the square rooted powers are summed so that the relative amplitudes of extracted sources per class l are normalized to the overall parametrization.

- The nonnegative parameters t_{fk} and v_{kn} characterize the magnitude spectra of audio source components and are an extension of the basic NMF model. Considering the sub-indexing of the kth column of $\mathbf{T}$ and the kth row of $\mathbf{V}$ we construct rank-one nonnegative estimates of the magnitude spectra of source components. The rank-one nonnegative matrices are given as

$$[\mathbf{T}]_k \circ ([\mathbf{V}]_k)^T \in \mathbb{R}^{F \times N} \tag{C.5}$$

  where $[\mathbf{T}]_k \in \mathbb{R}^{F \times 1}$ and $[\mathbf{V}]_k \in \mathbb{R}^{1 \times N}$. This parametrization can be seen to be a direct extension to the basic NMF model first introduced in section 1.2 and is used here for the purpose of specifying source amplitude information (magnitude spectra) of sources at component index k.

  The *partitioning* of these source components as described per component k by equation C.5 into *clusters* is achieved via the *cluster indicator* latent variable $\mathbf{Z} \in \mathbb{R}^{L \times K}$.

- Therefore, in summary, where as equation 5.18 could be considered the non-clustered SCM sum-of-product equation, by introducing cluster indicator latent variable z_{lk} as a nonnegative NMF parameter, equation 5.21 could be viewed as the clusterable sum-of-product equation.

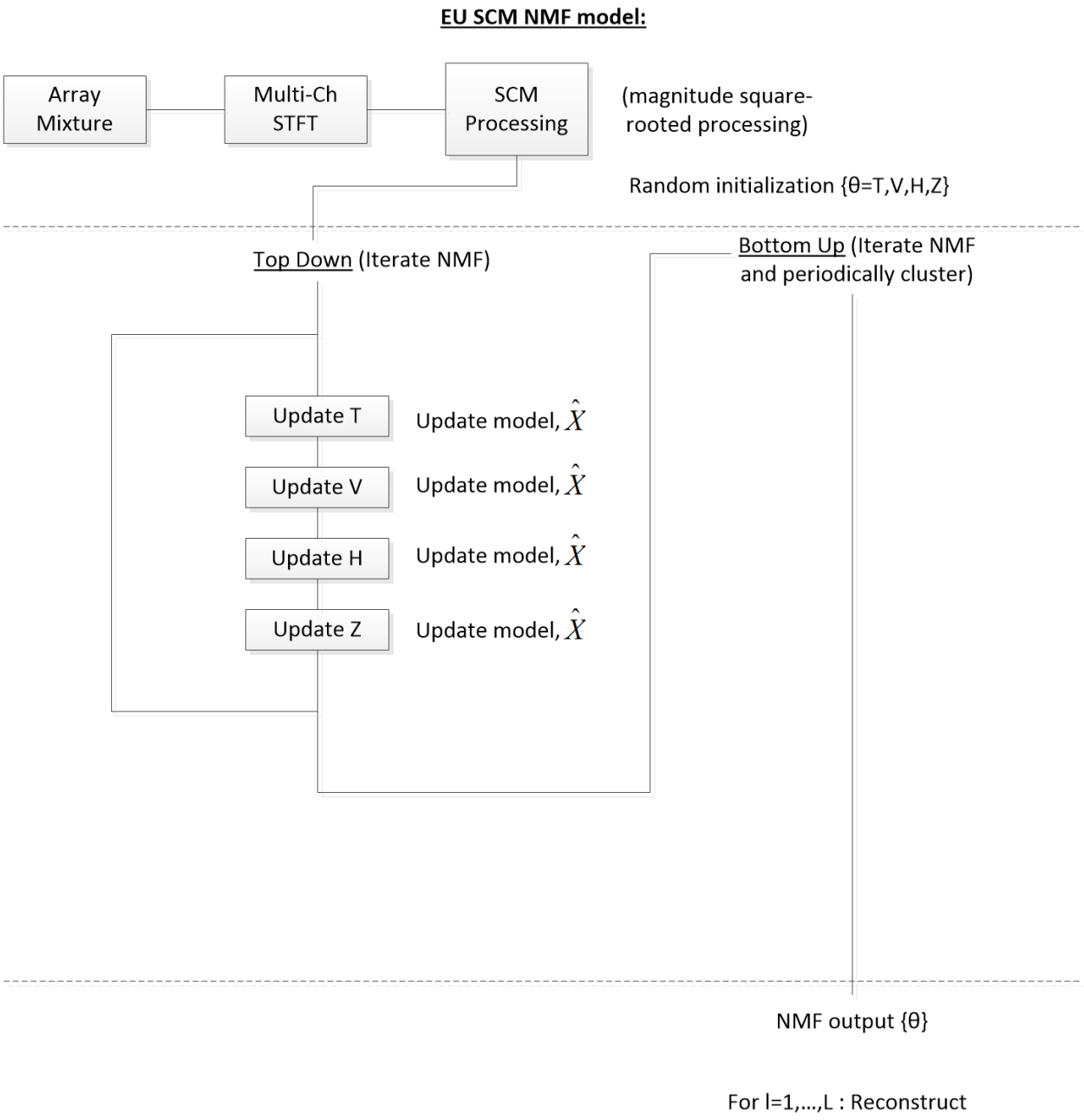

Figure C.4: EU-NMF SCM algorithm

Figure C.4 shows the high level sequence in which the CNMF updates are to be executed iteratively. Equations C.3 and C.4 are applied to the CNMF parameter set $\{\mathbf{T}, \mathbf{V}, \mathbf{H}, \mathbf{Z}\}$ and source estimates $[\tilde{\mathbf{y}}_{fn}^{(l)}]_m$ are reconstructed on the basis sub-indexing the parameters at the indices of output classes $l = 1, ..., L$ to construct time-frequency representations $[\tilde{\mathbf{y}}_{fn}^{(l)}]_m$. Inverse STFT's can the be applied to $[\tilde{\mathbf{y}}_{fn}^{(l)}]_m$ to reconstruct the time-domain signals.

C.2 DOA SCM NMF cont'd

C.2.1 K means Clustering

Before computing the reconstruction step, we seek to cluster the z_{ko} matrix parameter into Q total output clusters, treating the K row vectors of the matrix parameter as the observation vectors, or feature vectors of the data set to be clustered. Thus we seek indicators b_{qk} and means $\mathbf{u}_q$ where Q must be chosen to be of smaller size than K. However we can note in the next section that the reconstruction step is dependent on the indicators b_{qk} only, as well as z_{ko}, but not the means $\mathbf{u}_q$.

1. Randomly initialize the set of Q mean row vectors $\{\mathbf{u}_1, ..., \mathbf{u}_Q\}$. $\mathbf{u}_q \in \mathbb{R}^{1xO}$

2. Compute assignment of soft indicators, b_{qk} with the variable $m = 2$:

$$b_{qk} = \frac{1}{\sum_{r=1}^{Q} \left(\frac{||(z_{ko}).row(k) - \mathbf{u}_q||}{||(z_{ko}).row(k) - \mathbf{u}_r||} \right)^{\frac{2}{m-1}}} \tag{C.6}$$

3. Compute assignment of means $\mathbf{u}_q$, with the variable $m = 2$:

$$\mathbf{u}_q = \frac{\sum_{k=1}^{K} {b_{qk}}^m z_{ko}.row(k)}{\sum_{k=1}^{K} {b_{qk}}^m} \tag{C.7}$$

4. Evaluate the distorted measure J: Stop, if it has been minimized to a satisfactory degree, if not continue by repeating steps 2 and 3.

Figures C.5 to C.7 illustrate a K-means interpretation of the clustering that the DOA SCM NMF algorithm proposes should be applied to the SCM NMF parameter z_{ko}. The K-means (with $q = 1, ..., L$ clusters) interpretation can be summarized according to the notation

$$\{z_{qo}, b_{qk}\} \leftarrow \{z_{ko}\} \tag{C.8}$$

where the clustering procedure according to the K-means algorithm (and a convention to describe it in summary) was discussed according to equation 3.11. The figures presented here are once again demonstrated for illustration purposes however in Chapter 7 we demonstrate

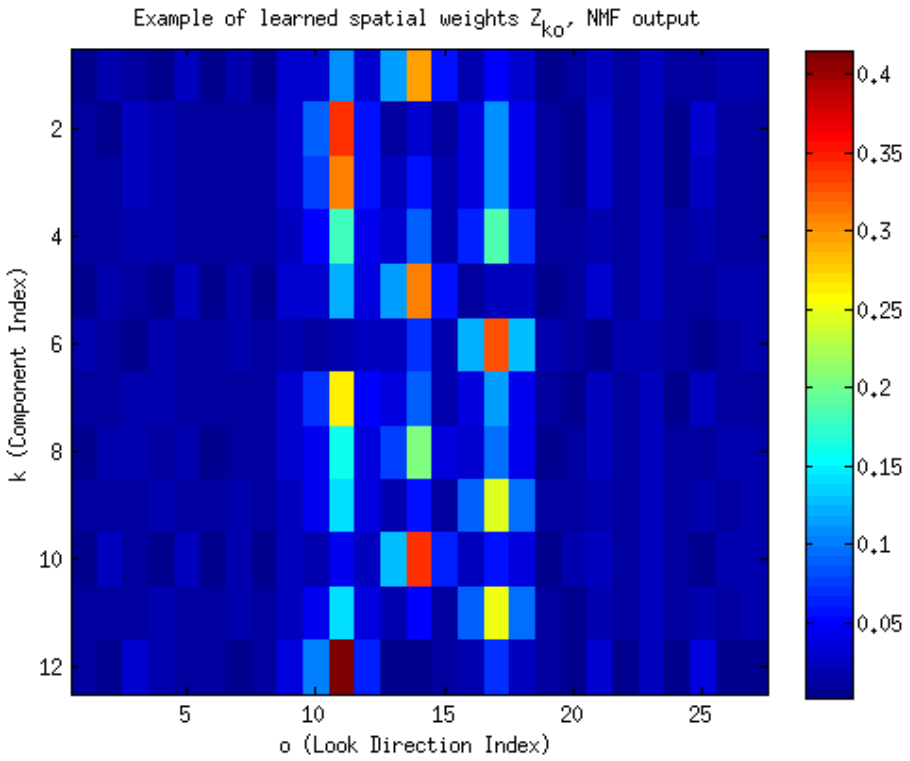

Figure C.5: Learned spatial weights for NMF parameter z_{ko}

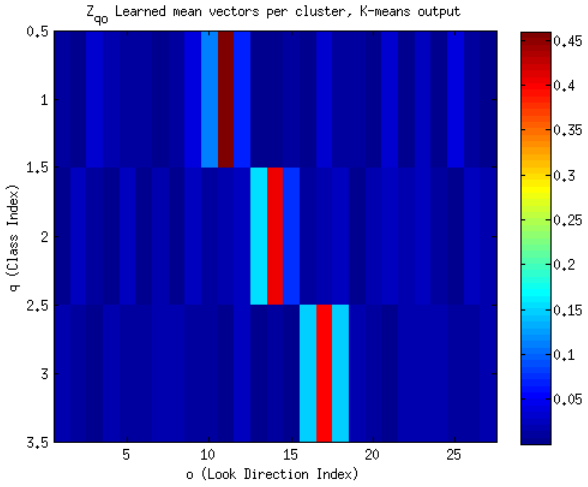

Figure C.6: Learned mean vectors z_{qo} (Look direction, Class) from output of K-means

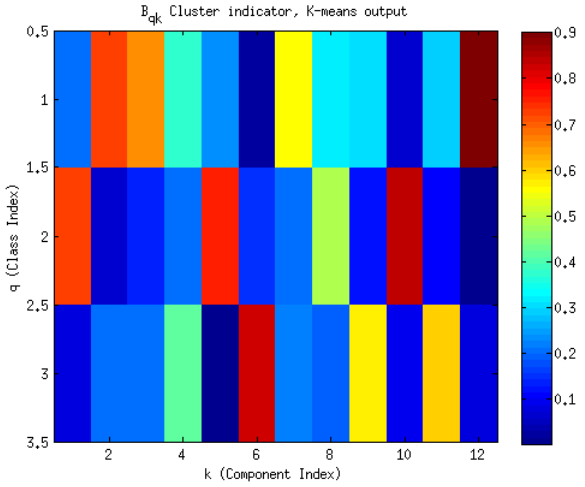

Figure C.7: Learned indicator variable b_{qk} from output of K-means

a similar set of corresponding figures that were in fact the direct result of an implementation of the DOA SCM NMF algorithm that was used to compared the DOA SCM NMF algorithm to the proposed algorithm.

C.2.2 Source Reconstruction

$$s_{fn}^{(q)} = \sum_{k,o} b_{qk} z_{ko} t_{fk} v_{kn} \tag{C.9}$$

$$\mathbf{y}_{fn}^{(q)} = \frac{\sum_{k,o} b_{qk} z_{ko} t_{fk} v_{kn}}{\sum_{q,k,o} b_{qk} z_{ko} t_{fk} v_{kn}} \mathbf{x}_{fn} \tag{C.10}$$

The intepretation of the source reconstruction step as detailed by equation C.10 can be considered as follows:

- We can choose to reconstruct spatial image estimates $\mathbf{y}_{fn}^{(q)}$ or simply just source estimates $s_{fn}^{(q)}$, which are purely nonnegative.
- The reconstruction as specified by C.10 corresponds to a multichannel Wiener filter source reconstruction similar to that described according to equation C.3.
- The observed multichannel STFT vector $\mathbf{x}_{fn}$ carries initial phase information into the qth estimated spatial image vector $\tilde{\mathbf{y}}_{fn}^{(q)}$ at time-frequency bin f-n.
- The mean vectors $\mathbf{u}_q$ outputted by K-means are not necessary to the source reconstruction, since we utilize instead the spatial weights z_{ko} appropriately weighted by indicator values b_{qk}
- Neither of the source reconstruction equations depend on the DoA kernel matrix parameter $\mathbf{W}_{fo}$

Figure C.8 shows the high level sequence in which the CNMF updates are to be executed iteratively. The clustering algorithm occurs solely after the CNMF algorithm has converged thus there is no *interleaved* mechanism for possibly clustering of CNMF parameters *while* the CNMF algorithm is being applied.

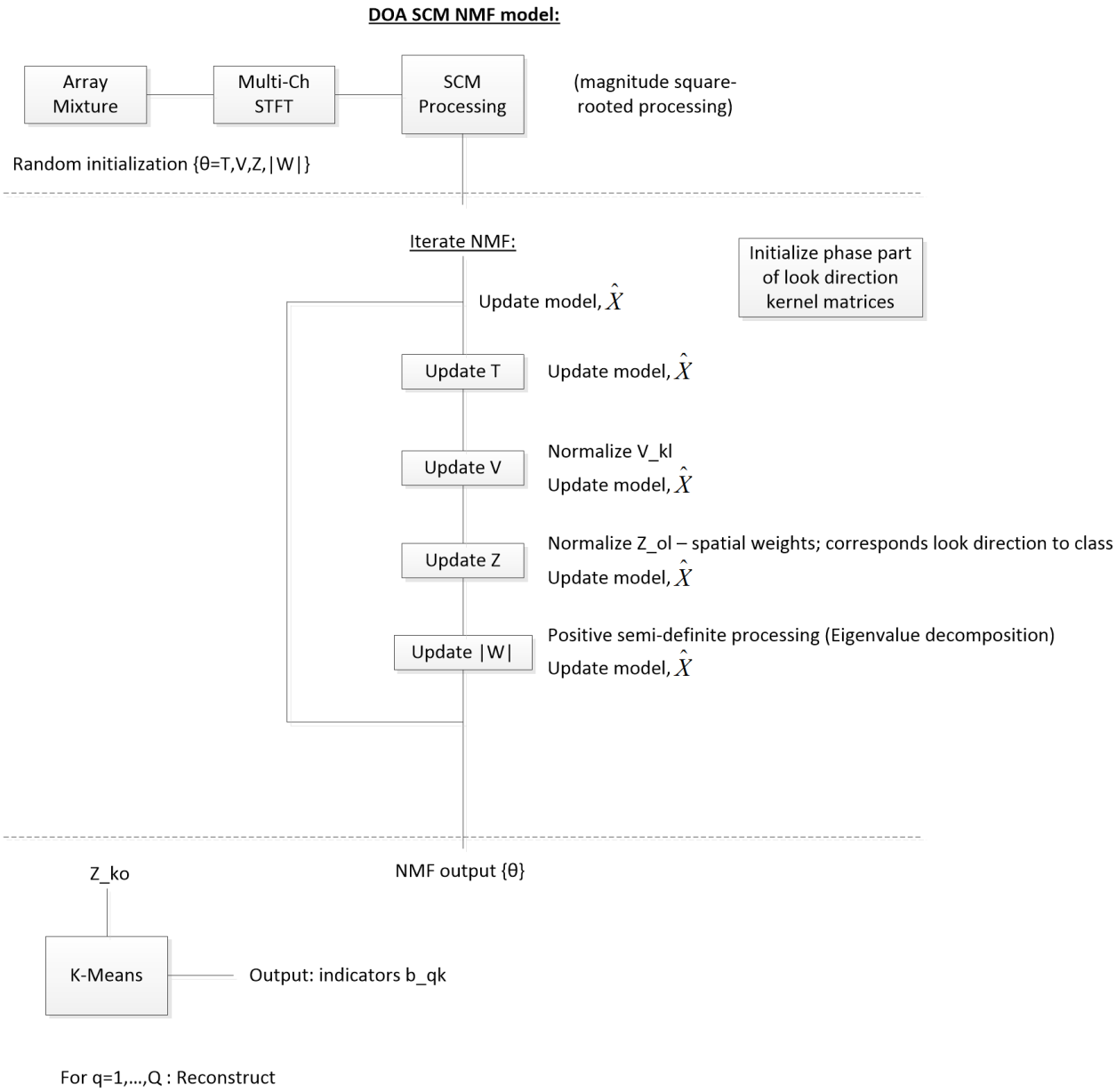

Figure C.8: DOA SCM NMF algorithm

Appendix D

Benchmark, Test Results, and Supplementary Test Cases

D.1 Overview of Source Separation Metrics

We again preface the discussion of these metrics by restating that although ideally we seek objective measures of audio quality that correlate perfectly with human subjective opinion of intelligibility of the separated output source channels, such endeavours are also an iterative process that is subject to the possibility of error, and it is still necessary to verify that the objective measures do indeed do their due diligence. We note that in [76], which provides a careful approach to this problem, two possible issues are pointed to that suggest such metrics are still subject to sometimes introducing the types of errors that they are designed to avoid. Namely that:

1. A computed distortion component does not correspond to what a human would perceive as distortion. For example, if a human listens to a distortion component returned by the separation metric and in fact hears the original desired sources within the audio, this would signify that the separation metric in this instance has done the opposite of its task, that is, it has erroneously identified what should not be included as a distortion as a distortion. These instances point to the fact that separation metrics, like all other audio algorithms, are subject to unwanted errors, signal leakages, and imperfections. If we intend to agree upon the frequent use of metrics, then we must be wary of the fact that the presence of such erroneous information tends to be encapsulated away from the user when they do indeed occur since the metrics are computed in a SNR like fashion, relying upon sum of squares like computations to determine the output values. This possibility was noted to be sometimes present in listening experiments conducted in [76].

2. The explanation provided for the previous issue is that true distortions may be time varying in nature, however, the metrics themselves which rely upon least-squares projections assume time invariance of the mixing model (modelling a so-called distortion LTI filter of a certain duration, typically 32ms as described [77]), and as such perhaps the metric itself is unable to extract the target distortion, as intended in these cases.

3. Another possible issue as noted in [76] is that the metrics can sometimes return computed distortion components that are non-zero even when the estimated target signal has no distortion. As one could easily see, this would also skew the interpretation of the values outputted by the metrics, if such an error were detectable and later found to be present.

Aside from these issues, we must still come to the conclusion that due to the resolution, dimensionality, and volume of information present in audio signal content, it would be very difficult to proceed to analyze audio quality without the use of objective measures. And therefore we opt to include within the thesis, the popular metrics as outlined in [77] and whose implementations are open source, as well as those provided in [76]. Following the description of the metrics as covered in [77], we present the decomposition of the separation quality metrics here as follows:

$$\text{ISR}_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t s_{ij}^{img}(t)^2}{\sum_{i=1}^{I}\sum_t e_{ij}^{spat}(t)^2} \tag{D.1}$$

$$\text{SIR}_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t s_{ij}^{img}(t)^2 + e_{ij}^{spat}(t)^2}{\sum_{i=1}^{I}\sum_t e_{ij}^{interf}(t)^2} \tag{D.2}$$

$$\text{SAR}_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t s_{ij}^{img}(t)^2 + e_{ij}^{spat}(t)^2 + e_{ij}^{spat}(t)^2}{\sum_{i=1}^{I}\sum_t e_{ij}^{artif}(t)^2} \tag{D.3}$$

$$\text{SDR}_j = 10\log_{10}\frac{\sum_{i=1}^{I}\sum_t s_{ij}^{img}(t)^2}{\sum_{i=1}^{I}\sum_t e_{ij}^{spat}(t)^2 + e_{ij}^{interf}(t)^2 + e_{ij}^{artif}(t)^2} \tag{D.4}$$

Where it is hypothesized that the true source image $s_{ij}^{img}(t)$ can be decomposed as the estimated as the difference between the source image $\hat{s}_{ij}^{img}(t)$ and the sum of unwanted components $e_{ij}^{spat}(t), e_{ij}^{interf}(t), e_{ij}^{artif}(t)$, which each be computed with the aid of two projection operators $P(\cdot)_j^L$ and $P(\cdot)_{all}^L$, which both take as argument the true and estimated spatial images $s_{ij}^{img}(t)$ and $\hat{s}_{ij}^{img}(t)$, respectively, in their computation.

The operators respectively are defined as:

1. $P(\cdot)_j^L$ is a least-squares projection onto the subspace spanned by $s_{kj}^{img}(t)$ over all microphone channels and possible delays taken as integer multiples of a delay τ bounded by the distortion filter length L: $s_{ij}^{img}(t-\tau)$, $1 \leq k \leq M$. Where M is the number of microphone channels and L is the distortion filter length and defines $0 \leq \tau \leq L-1$.

2. $P(\cdot)_{all}^L$ is a least-squares projection onto the subspace spanned by $s_{kl}^{img}(t)$, for $1 \leq k \leq M$, $1 \leq l \leq M$, and $0 \leq \tau \leq L-1$.

Thus by applying $P(\cdot)_j^L$ and $P(\cdot)_{all}^L$ equations D.1 to D.4 can be generated. For further details the reader is pointed to the description within [77].

In order to consider the implementation of the OPS, TPS, IPS and APS metrics the reader is pointed to the description provided within [76].

D.1.1 Applying the Metrics to CNMF Reconstructed Source Signals

In applying the software as a basic tool for measuring performance measured in terms of signal similarity, we pass as arguments two separate signals to the software each time we ask it to return an output score for a particular separation quality metric. An 'estimated source' and 'true source' signal are passed as a *pair of signals* for which an objective score needs to be computed, as somewhat of an abstracted query to the performance evaluation software. In principle, for any metric, if the pair of signals are indeed similar or adequately correlated, the software should return a higher score than if the two signals were highly dissimilar. The reason for using separate types of metrics is that it is hypothesized that the signal components may be further decomposed (by the software) and shown exhibit distinct and separate degrees of similarity, if accordingly considered separately (as discussed earlier in section D.1).

In our usage of the software, the reconstructed (CNMF parametrized source) signal corresponds to the 'estimated signal' and the 'true source signal' corresponds to any one of the originals signals that would have been saved and that were used by 'MCRoomSim' to generate the multichannel set of microphone signals (each microphone signal corresponding to a spatially unique but spectrally similar mixture of all the source signals, specifically chosen to be three) that we know in different test cases were purposefully constructed as being either overlapping or non overlapping in time and frequency, in order to provide a variety of different signals to exercise both CNMF algorithms. In organizing the reconstructed CNMF source signals (to be regarded as the *output* of either CNMF algorithm and as *input* to the performance evaluation software) we of course had to manually keep track of such things as

labelling and naming of pairs of true and estimated source signals, to be passed as input(s) to the software.

Thus, in summary, the software has the simple to understand but non trivial task of comparing pairs of signals and to objectively measure how similar they are, in such a way that is ideally as fair and as meaningful as possible. Therefore we emphasize that the software that computes any particular output score corresponding to any particular metric, has absolutely no specific knowledge about the CNMF implementation that generated the pair of signals that it is being asked to compare, and we believe that pointing this out as a distinguishing fact, is highly beneficial. In principle, issues concerning the 'fairness' and associated behaviour should not be identified soley on the basis of a particular outcome (e.g. SDR score); rather the definition of the (SDR) metric should be evaluated in terms of what it (as a metric) provides in terms of its meaningfulness towards *as broad* a class of signals as possible (CNMF signals being just one instance of type of signal). Bringing this up is only to remind the reader that some of these issues were already introduced in section D.1.

In any case, even if a particular metric were indeed unfair, by this point we have settled upon the following metrics to be presented as the ones that are going to be used, and we now proceed to consider the resulting scores in order to determine how the proposed and reference CNMF algorithms compare to each other, assuming for simplicity that all metrics are *adequately* fair.

D.2 DOA Spatial Covariance NMF Reference Algorithm Implementation

D.2.1 Test Case 1: Non Overlapping Musical Notes

In this section we re-visit the algorithm presented within [24], and attempt to demonstrate the various similarities and dissimilarities between the proposed algorithm and this algorithm, treated as a reference algorithm. The CNMF algorithm was iterated for a total of 100 CNMF iterations, as was the case for the proposed algorithm. The output of the CNMF algorithm was applied to the K-means clustering module as proposed within section C.2.1.

Reconstructed sources were obtained via the Wiener filtering like computation of spatial images $\mathbf{y}_{fn}^{(q)}$ for $q = 1, ..., L$ as specified according to C.2.2 and the resulting STFT domain and time-domain reconstructed and separated signals will be shown at the end of this section after first considering each converged CNMF parameter individually.

We recall that the DoA SCM algorithm was an entirely spatial covariance matrix (SCM)

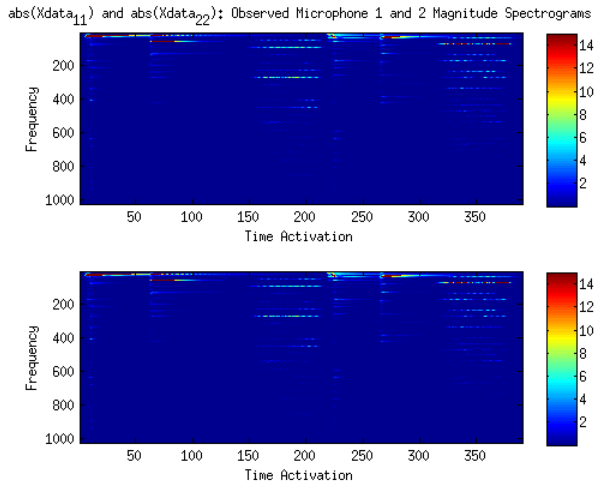

Figure D.1: Magnitude of Target Stereo STFT

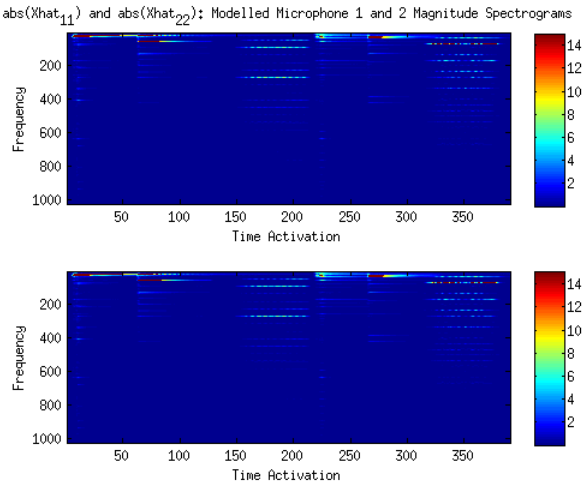

Figure D.2: Magnitude of Modelled Stereo STFT

based approach where the observed spatial covariance matrix is processed from the observed multichannel vector according to the operation $\mathbf{X}_{fn} = \mathbf{x}_{fn}\mathbf{x}_{fn}^{H}$ but where $\mathbf{x}_{fn}$ must also be preprocessed so that its amplitudes correspond to square rooted values.

Therefore the *parametrized* spatial covariance matrix per time frequency bin that the algorithm proposed could be denoted $\hat{\mathbf{X}}_{fn}$ and was shown according to the DoA SCM model to be a function of the matrix parameter set $\theta = \{\mathbf{W}, \mathbf{Z}, \mathbf{T}, \mathbf{V}\}$

where

1. The matrices $\mathbf{T}$ and $\mathbf{V}$ were applied as part of the parameter set to have the similar task as within the context of a two factor basic NMF time frequency representation of magnitude spectra of unknown and also unlabelled source components to first be learned by NMF and to later be linked by clustering.

2. The parameter $\mathbf{W}$ specified DoA kernel matrices that encoded a spatial configuration per look direction and per microphone of frequency domain phase differences that were populated on the basis of the spatial geometry of the microphones with respect to the microphone array center.

3. The parameter $\mathbf{Z} \in \mathbb{R}^{K \times O}$ specified spatial weights z_{ko} that could be used as features vectors at the output of iterating the CNMF algorithm, and applied to a K-means post clustering step as specified within section C.2.1.

Figure D.1 shows the values of elements along the main diagonal of *observed spatial* covariance matrices $\mathbf{X}_{fn}$ per time-frequency bin.

Figure D.2 shows the values of elements along the main diagonal of *parametrized* spatial covariance matrices $\hat{\mathbf{X}}_{fn}$ per time-frequency bin as specified by the DoA SCM model and according to (5.34).

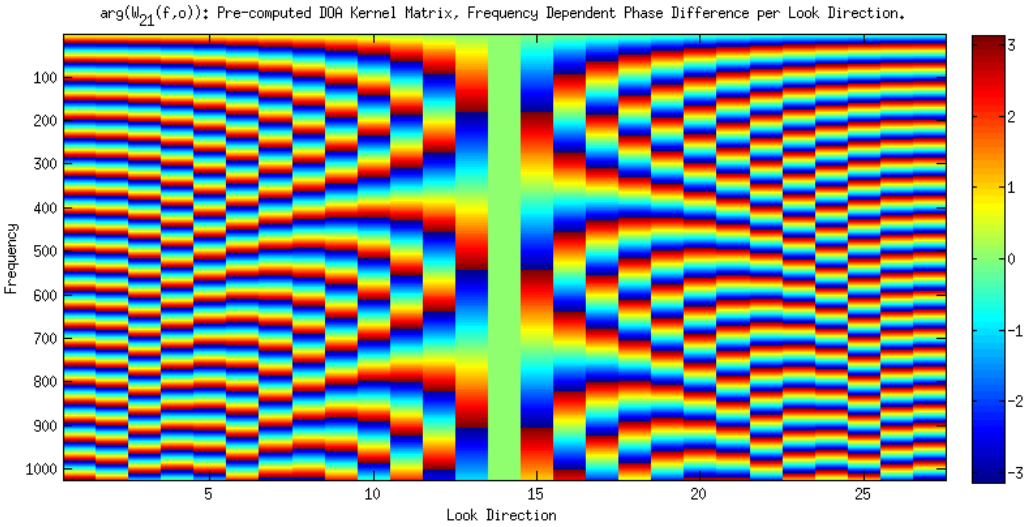

Figure D.3: Interchannel Phase Difference Quantity (modelled, estimated)

Figure D.3 shows the frequency domain phase differences patterns defined according to interchannel time differences between microphones $m = 1$ and $m = 2$ and can thus be considered as reflected within the *off-diagonal* elements of DoA kernel matrices $\mathbf{W}_{fo}$. It was specified that the off diagonal elements of DoA kernel matrices could be specified according to equation 5.32.

An alternative interpretation for Figure D.3 is that it represents a look direction dependent dictionary of the spatial filtering signature that should occur as a function of the time difference between microphones $m = 1$ and $m = 2$ interpreted as frequency domain phase differences. Here, once again, since we apply the *phase wrapped* complex argument operator we note that the real number that specifies the phase in each bin is limited to the interval $-\pi < \arg([\mathbf{W}_{fo}]_{2,1}) < \pi$.

Figure D.4 shows the frequency template dictionary matrix t_{fk} corresponding to frequency dependent column vectors, specified by their component index k, that provide spectrally common frequency patterns and act as basis vectors for the magnitude spectra of the sound sources to be parametrized.

Figure D.5 shows parameter t_{kn} whose rows correspond to time activation vectors. Traversing across the activation index n for any particular row vector at component bin

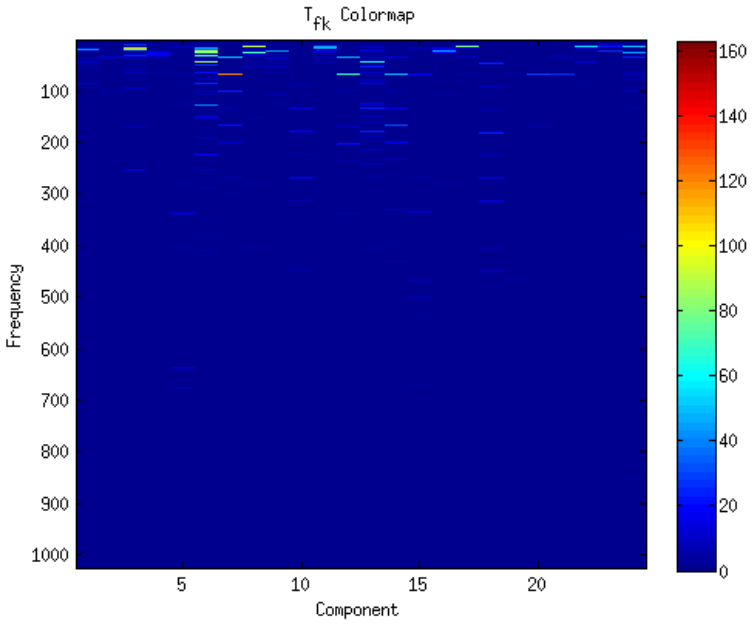

Figure D.4: Frequency Template Dictionary Matrix t_{fk}

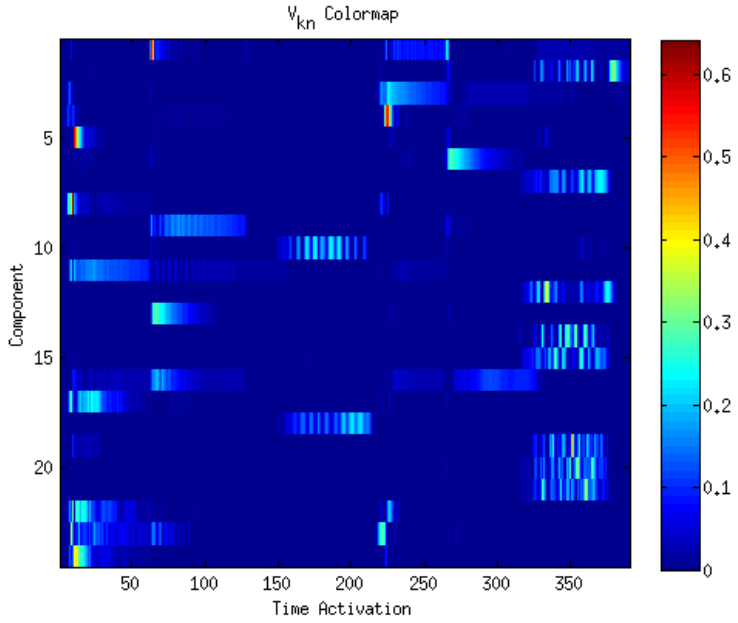

Figure D.5: Activation Dictionary Matrix V_{kn}

k, the row vector models the time variation behaviour of the kth sound component. We note that the learned activation patterns in the converged activation matrix are noticeably sparse and have null activity in a majority of the activation bins. We physically interpret this in fact as a desired behaviour since the current test case being considered was purposefully

constructed in such a way that the source signals were mixed into the multichannel STFT not heavily overlapped in time.

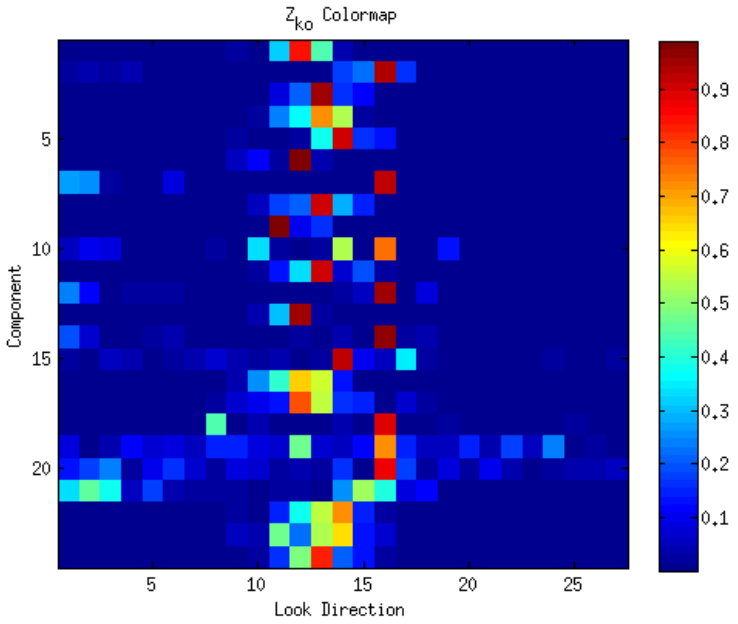

Figure D.6: Feature Vectors Dictionary Matrix z_{ko} (Unclustered)

Figure D.6 shows the matrix of learned spatial weights z_{ko}, for all possible component and look direction bins, that specify how frequency and look direction DoA kernel matrices should be spatially associated to magnitude spectra at different possible component and look direction bins. Evidently, it seems that the CNMF algorithm has learned that spatial weights corresponding to look directions at the extremity locations of half plane sampled in terms of azimuth by microphone array should be weighted (suppressed) by being assigned zero-values. This signifies that the look directions that are significant lie within the spatial look directions geometrically pointing towards the middle of the 2-D half plane that the stereo microphone array spatially samples.

If we assume that the converged result is adequate, we then utilize the converged spatial weights as feature vectors (i.e. observed data with respect to clustering) upon which to apply the method of K-means in order to obtain further parametrization of the spatial weights. That is, we expect the clustering representation provided by the output of K-means to provide us with a correspondence between how the sources that are to be reconstructed (index by the class variable q) in terms of their STFT spectra should be associated to the correct set of CNMF components (according to the component variable k) thus being able to link learned magnitude spectra within the $\mathbf{T}$ and $\mathbf{V}$ on the basis of common spatial "signatures" as to

be extracted the converged matrix z_{ko}.

We assert that $z_{ko} \approx \sum_{q=1}^{L} b_{qk} z_{qo}$ and thus we intend for the output of K-means to provide a representation equivalent to the matrix z_{ko}, based on iterative minimization of the K-means distortion measure.

Thus by doing K-means we are not changing the actual representation but in fact we are only finding a *less redundant* representation of z_{ko}, in terms of the matrix of indicators b_{qk} and the matrix of mean vectors z_{qo}, where the number of clusters as specified by $q = 1, ..., L$ is to be chosen as equivalent to the number of desired output class (source) signals.

Figure D.7 shows the indicator matrix b_{qk} that is applied to the source reconstruction step as outline in section C.2.2.

Figure D.8 shows the mean vectors z_{qo} for $q = 1, ..., L$ whose maximum values per row specify to which look direction the K-means algorithm has determined that the qth class is best associated with. Since we already have the indicator values specified by b_{qk} as well as the spatial weight indicators z_{ko}, and by considering the reconstruction step as specified by C.2.2, we can note that we don't actually require the mean vectors in order to compute the reconstruction of source signals.

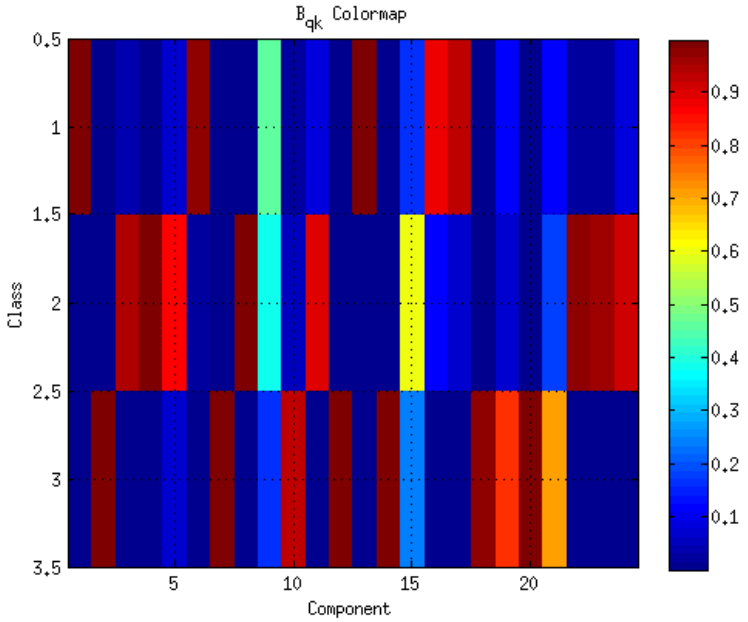

Figure D.7: Component to class indicator matrix b_{qk} (Clustered)

Lastly, we consider how observed and parametrized SCM's (i.e. $\mathbf{X}_{fn}$ and $\hat{\mathbf{X}}_{fn}$) compare at each time frequency bin and if the matrix $\hat{\mathbf{X}}_{fn}$ in fact models the interchannel time differences

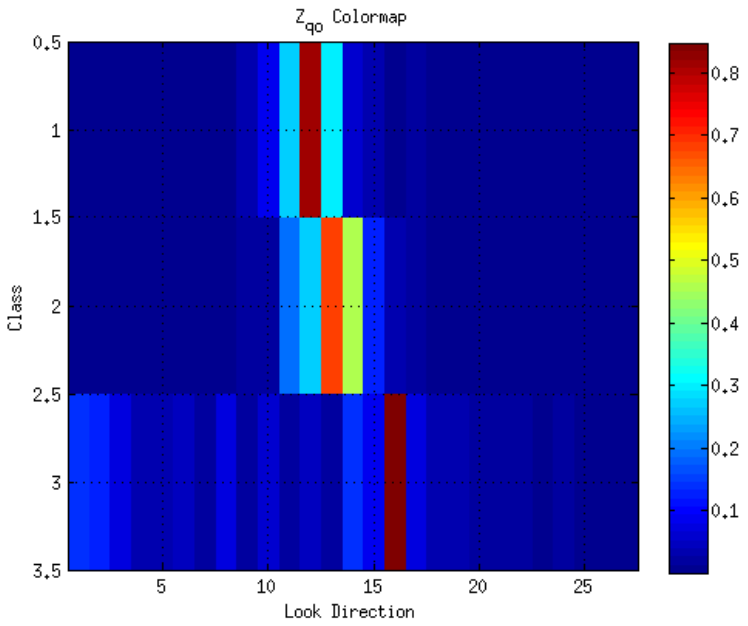

Figure D.8: Look direction feature vectors per class z_{qo} (Clustered)

between overlapping spectra as determined by the multichannel STFT algorithm's learned parametrization according its parameter set $\theta = \{\mathbf{W}, \mathbf{Z}, \mathbf{T}, \mathbf{V}\}$.

In order to do this we shall consider and compared Figures D.9 and D.10 against one another.

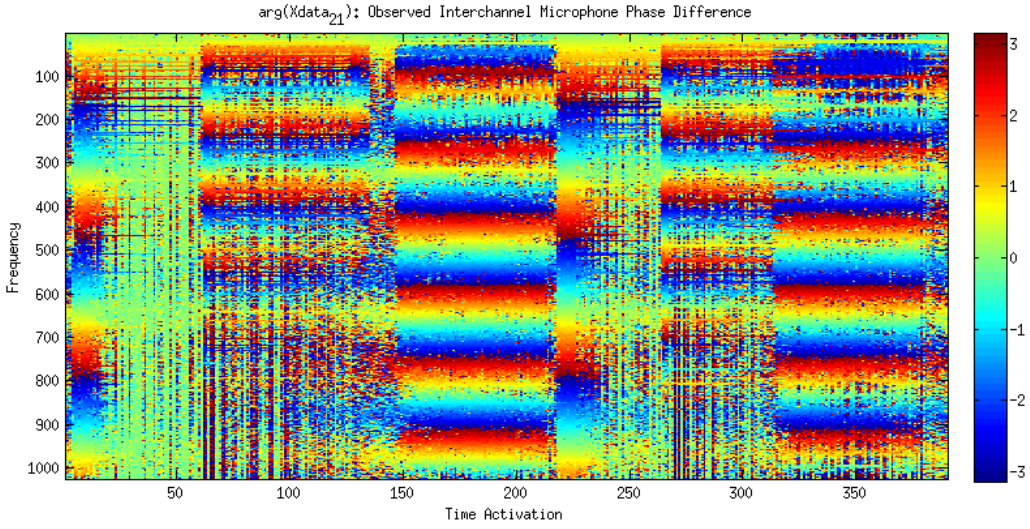

Figure D.9: Observed Interchannel Phase Difference Quantity

Immediately it can be observed that Figure *D*.10 does not entirely model the patterns

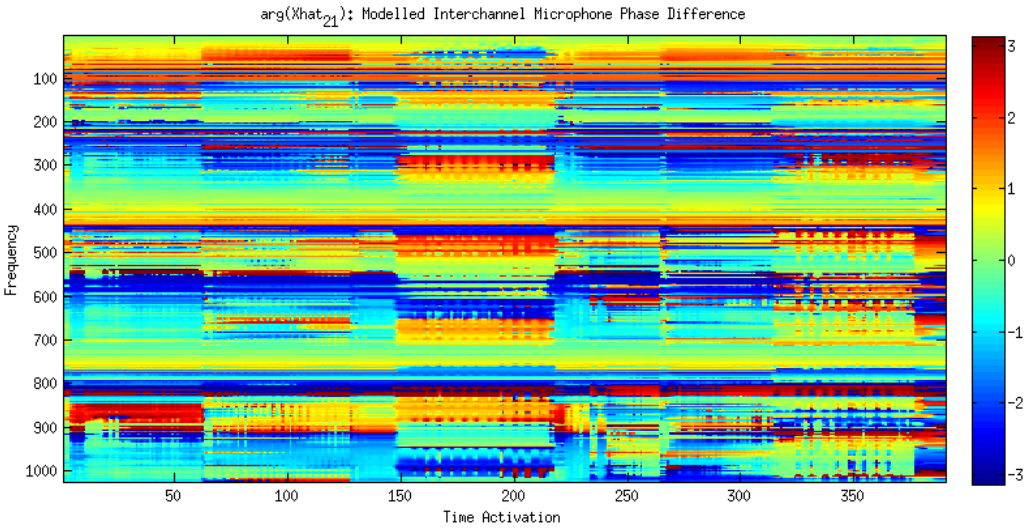

Figure D.10: Interchannel Phase Difference Quantity (based on converged CNMF parameters)

that are observed within Figure D.9. In comparing the DoA algorithm to the proposed algorithm covered earlier, the proposed algorithm (see Figure 7.9) seems to in fact do better at modelling the interchannel phase differences (frequency dependent patterns) at each STFT frame bin $n = 1, ..., N$. This could be in part due to the notion that the proposed algorithm has taken into account to some degree more so the apparent time activation dependence of the observed (target) phase differences as modelled by the target quantity shown in Figure D.9 and has done a better job at modelled this in terms of its clustering algorithm than the DoA algorithm's clustering algorithm.

As will be seen by considering the next few figures, the signals reconstructed by the DoA algorithm have temporal envelopes that are fairly good in terms of *class to component* correspondence

The interchannel phase shifts as modelled according to Figure D.10 may not be all that significant to providing perceptual benefits as compared to the actual absolute (i.e. initial phases) obtained from the Wiener filtering-like reconstruction step (equation C.2.2) that is applied (and that the proposed algorithm in fact does not use).

Therefore this last point could be in fact one of the most key and primary issues with respect to considering why the *perceptual quality* of the DoA algorithm still in most cases performs well as compared to the *proposed algorithm* even though the proposed algorithm shows some signs of *directly* modelling source estimates perhaps better in terms of its CNMF parameter set $\theta = \{W, T, V, Y, Z, \Phi_W, \Phi_S, \Phi_U\}$ as compared to the DoA algorithm and its parameter set $\theta = \{\mathbf{W}, \mathbf{Z}, \mathbf{T}, \mathbf{V}\}$.

As no further experiments have been done, we must further rely upon the metrics that were introduced in order to consider which algorithm seems to do better at separating sources, although, as mentioned the metrics themselves are not perfect indicators of perceptual robustness and integrity of separated audio signals since they have been shown, for instance, to not always correlate perfectly with human assisted (subjective) measures of audio qualities between different types of signals.

Therefore, this matter could be highlighted within the thesis as an ongoing topic that might require further and more meaningful analysis.

Figures D.11 to D.16 demonstrate the parametrized source estimates of the DoA algorithm's separated signals in both the time domain and the short time Fourier transform domain.

As mentioned, metrics will be applied within the next section in order to compare the separated source estimates of the two algorithms (per each common test case) in order to quantify each algorithm's respective performance.

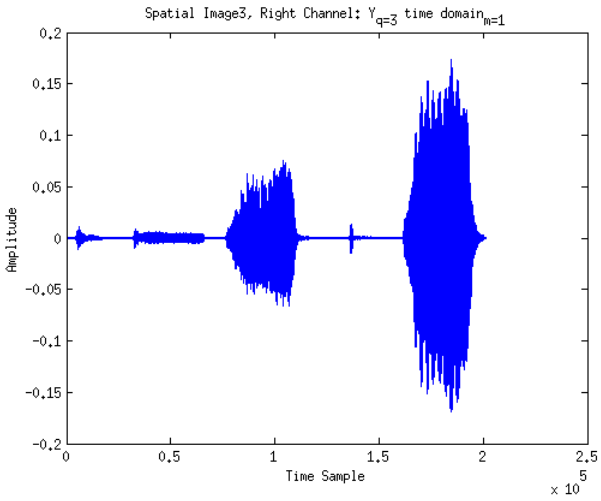

Figure D.11: Separated Output Class: Violin Signal, Time Domain

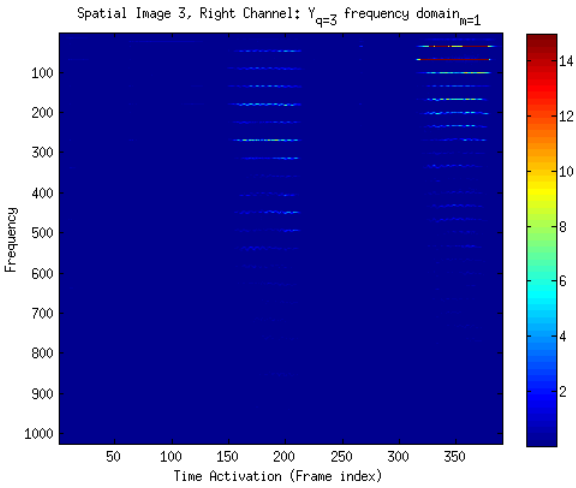

Figure D.12: Separated Output Class: Violin Signal, Frequency Domain

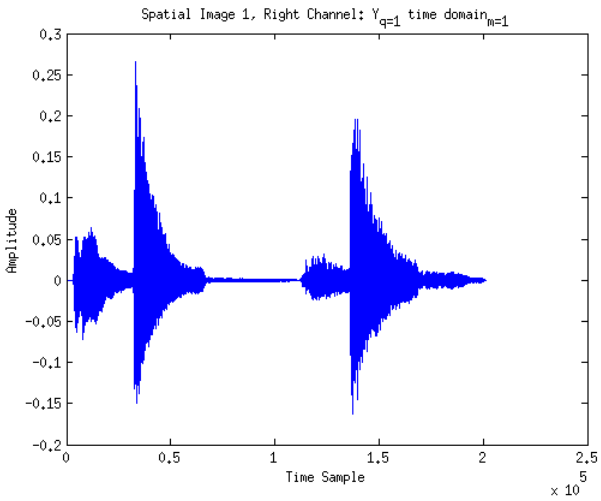

Figure D.13: Separated Output Class: Violin Signal, Time Domain

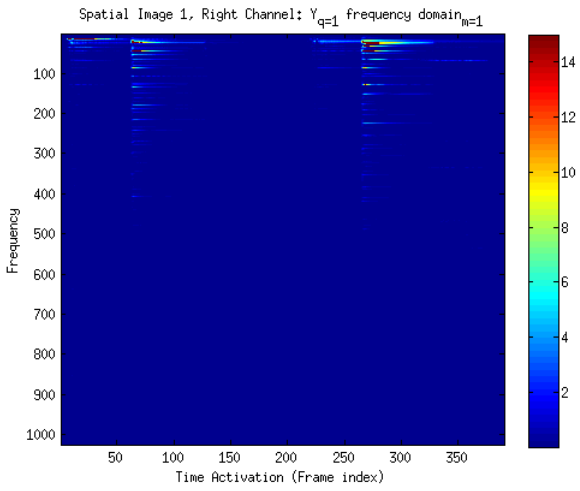

Figure D.14: Separated Output Class: Violin Signal, Frequency Domain

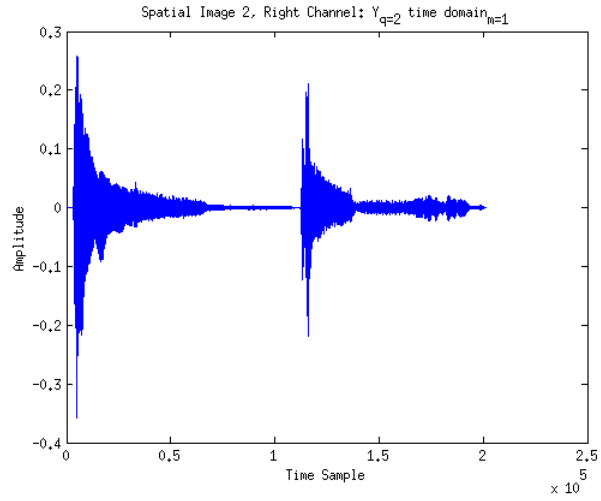

Figure D.15: Separated Output Class: Violin Signal, Time Domain

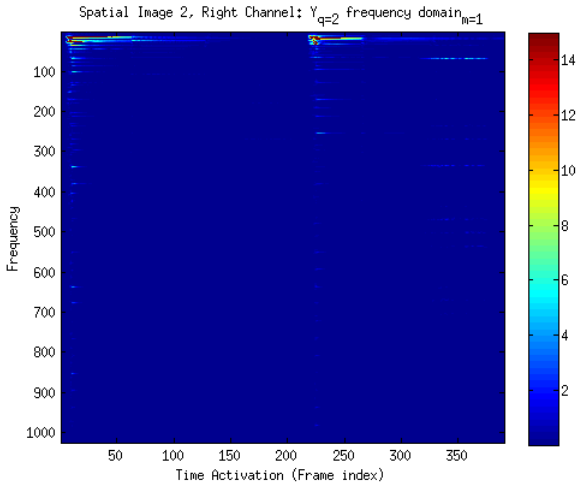

Figure D.16: Separated Output Class: Violin Signal, Frequency Domain

D.3 Benchmarking of Separation Quality and Separation Metrics (Continued)

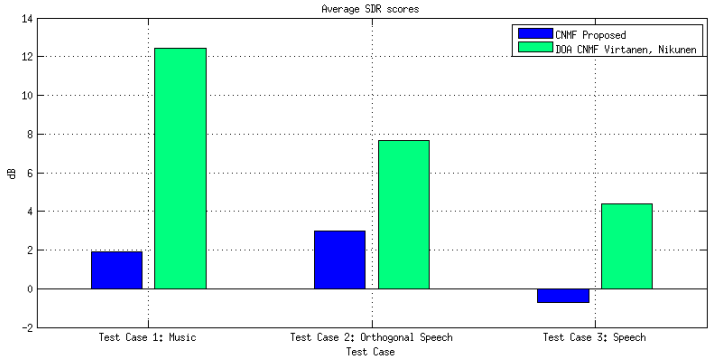

Figure D.17: Average SDR per Test case

Figure D.17 depicts average SDR scores computed for all three test cases, now considered individually. We note by considering this particular figure that the proposed algorithm performs relatively poorly in terms of the SDR metric in all three test cases in total as compared with the reference algorithm. For test case 3, the output SDR score even goes below 0dB, which signifies poor SDR performance when continuous (overlapping) speech is applied to the proposed CNMF algorithm, with three simultaneous speakers.

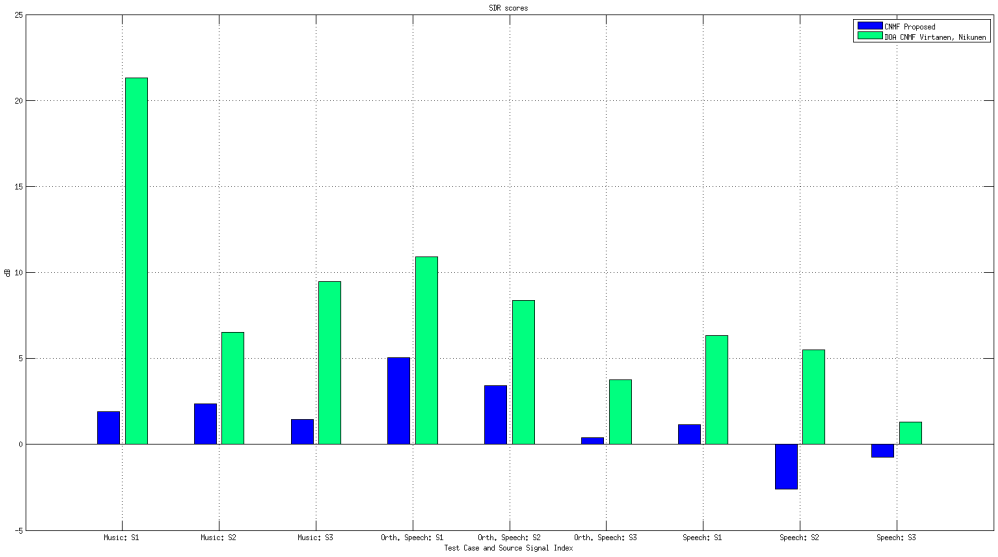

Figure D.18: Measured SDR per Source signal and per Test case

Figure D.18 demonstrates a more detailed view of the non-averaged results. Focusing again on the third test case (continuous speech) we can more precisely identify that the source signal corresponding to speaker 2 had the worst SDR of a source signal for all possible test cases.

By contrast, the maximum SDR computed for any source signal and any test case was computed by the reference algorithm in test case 1 for source 1.

Moving on to subsequent figures corresponding to subsequent metrics the reader should note than an effort was made to preserve the order of the source signals *labels* (e.g. S1, S2, S3) from one figure to the next in order to maintain correctness in terms of analyzing the results.

Next, figure D.19 demonstrates the average SIR per test case and makes it evident that the reference algorithm outperforms the proposed algorithm once again.

Interestingly the overlapping speech test scenario has approached similar performance where as the musical signal test scenario remains such that the reference algorithm does better.

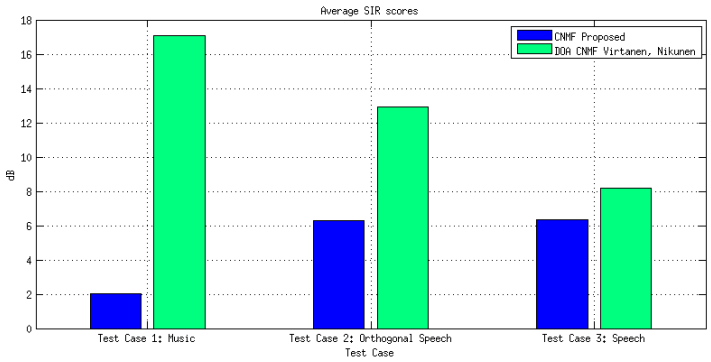

Figure D.19: Average SIR per Test case

Figure D.20 shows the individual scores per source signal and per test case, as expected the scores correspond to non averaged data points resembling the conclusions that can be drawn from the previous figure but in more detail.

Figure D.21 shows the first instance in which the proposed algorithm largely does better than the reference algorithm. Here we consider the SAR value of the separated signals.

Looking at individual test cases according to Figure D.22, we see that the proposed algorithm offers better SAR performance when music signals are used as the mixture signals. For the data points corresponding to test case 2 (non overlapping speech) this is also true.

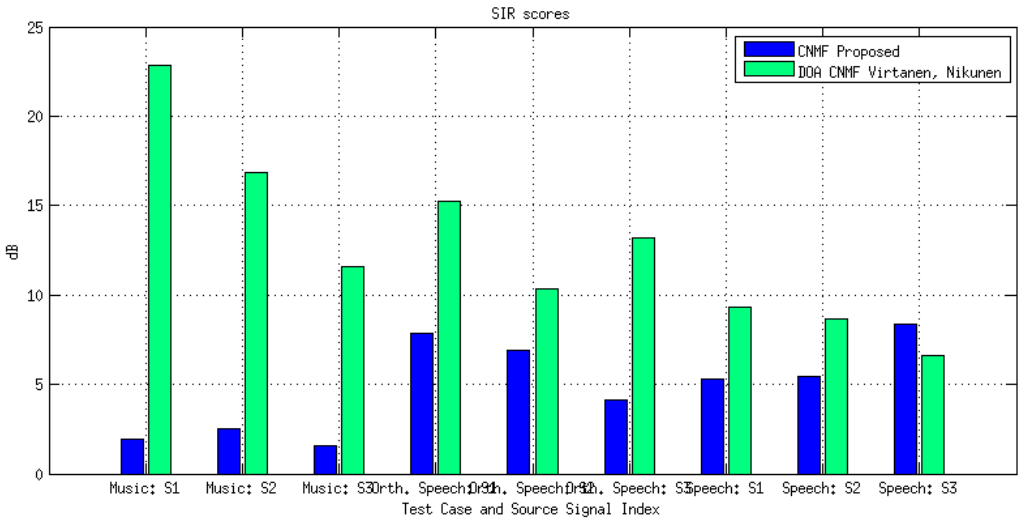

Figure D.20: Measured SIR per Source signal and per Test case

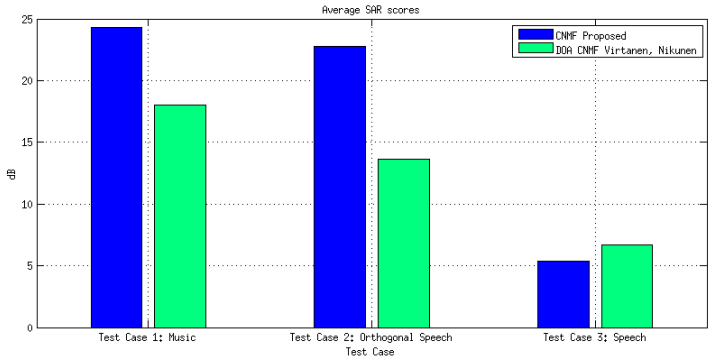

Figure D.21: Average SAR per Test case

For the third test case neither algorithm does exceedingly well but the reference algorithm does edge out the proposed algorithm by a small margin.

We now move on to consider a set of separation metrics that take a different approach to how each of the objective measures is computed. The four metrics to be considered are the OPS, TPS, IPS, and APS metrics.

First we consider the OPS metric, the results of which can be seen in figure D.23 and once again by considering the data points occurring from averaging values of source signals within each test case. Looking at the averages we may conclude that for the OPS metric the proposed algorithm is at least comparable to the reference algorithm. Looking at individual scores in Figure D.24, and in particular at test case 2 we note that the reference algorithm does exceedingly well at reconstructing the speech signal corresponding to label S1.

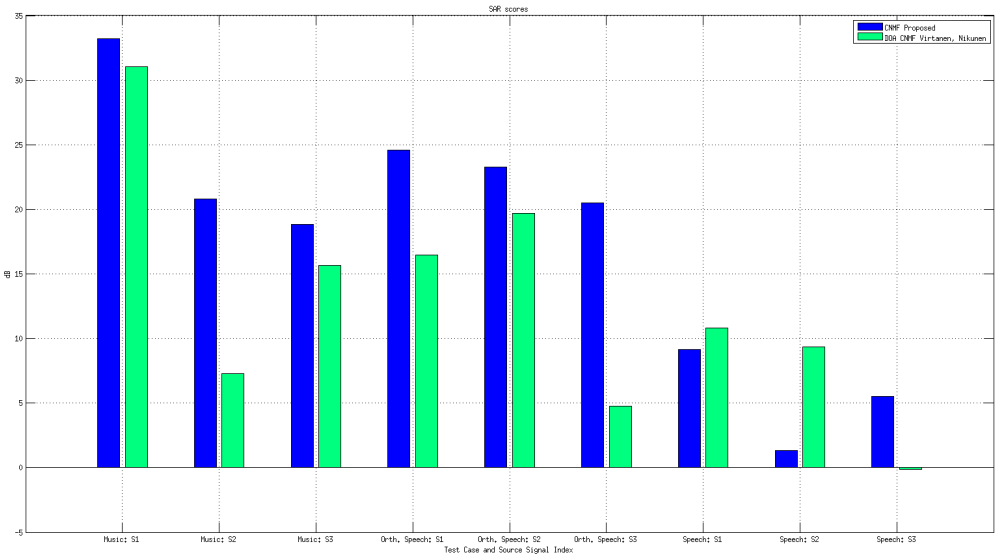

Figure D.22: Measured SAR per Source signal and per Test case

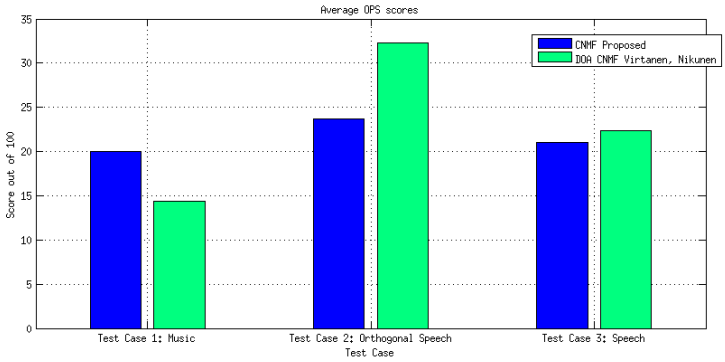

Figure D.23: Average OPS per Test case

But we can also note within the same set of data points that the reference algorithm does poorly at reconstructing the source signal with label S3, according to the OPS metric.

By computing the average of all source signal cores, taken as data points, the reference algorithm does fairly better overall, as seen by considering the average OPS score for test case 2.

Briefly reviewing both Figures D.25 and 7.49 D.26, the separation quality of the reference algorithm largely exceeds that of the proposed and especially with respect to test case 1, but is fairly comparable in terms of test cases 2 and 3.

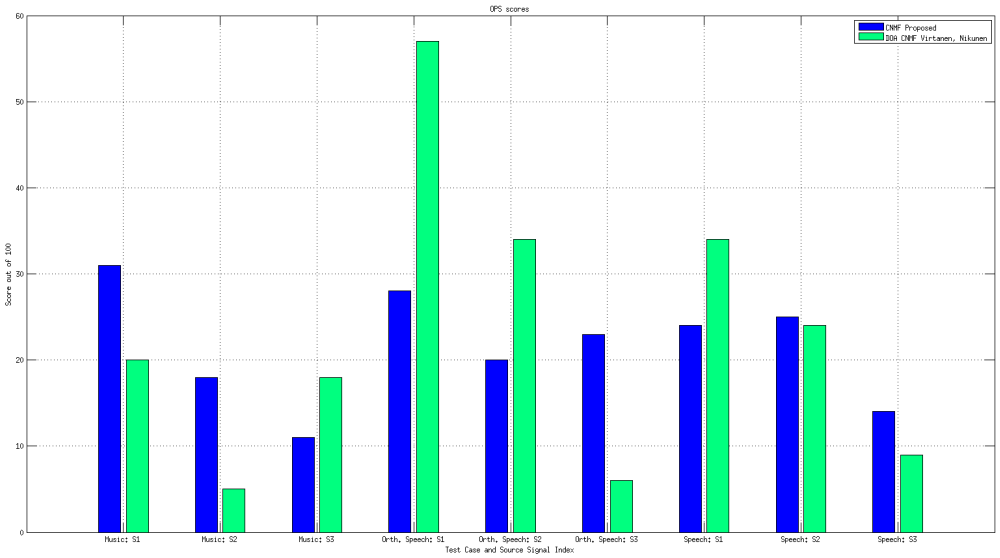

Figure D.24: Measured OPS per Source signal and per Test case

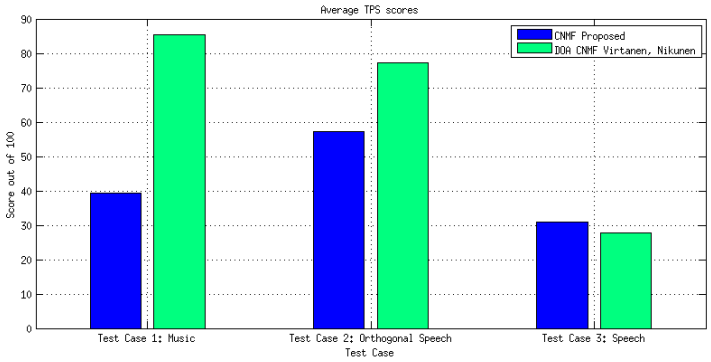

Figure D.25: Average TPS per Test case

For the final two metrics the disparity between scores for the two algorithms is significantly greater than for other metrics that have been considered thus far.

For the IPS metric, as demonstrated within Figure D.27, the proposed algorithm here outperforms the reference algorithm when we consider average scores, regardless if the input signals are music or speech. Figure D.28 shows individual scores for all source signals within all test cases.

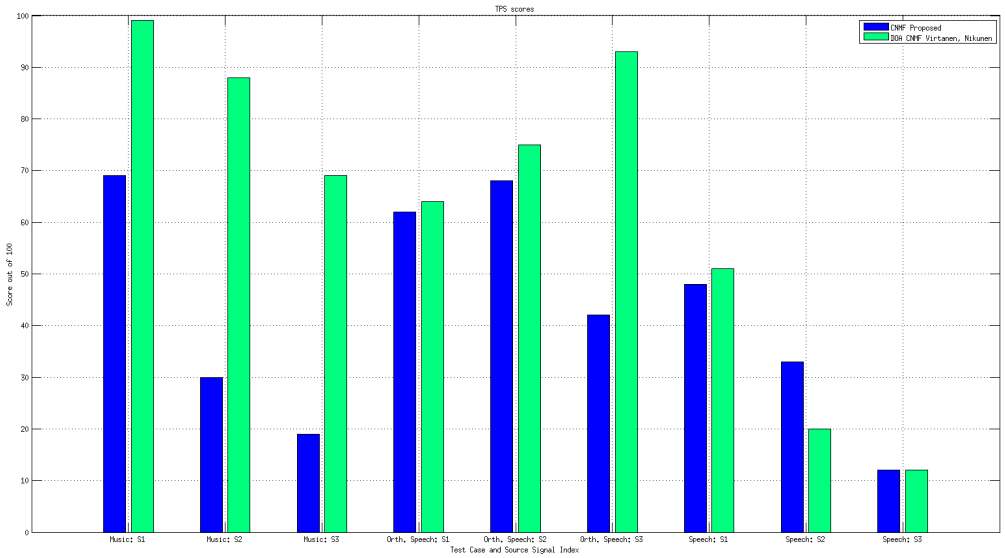

Figure D.26: Measured TPS per Source signal and per Test case

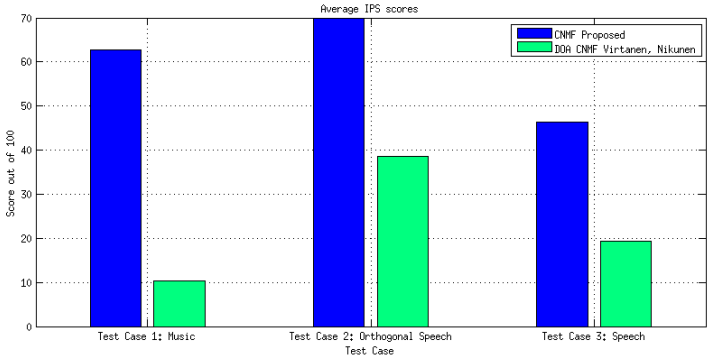

Figure D.27: Average IPS per Test case

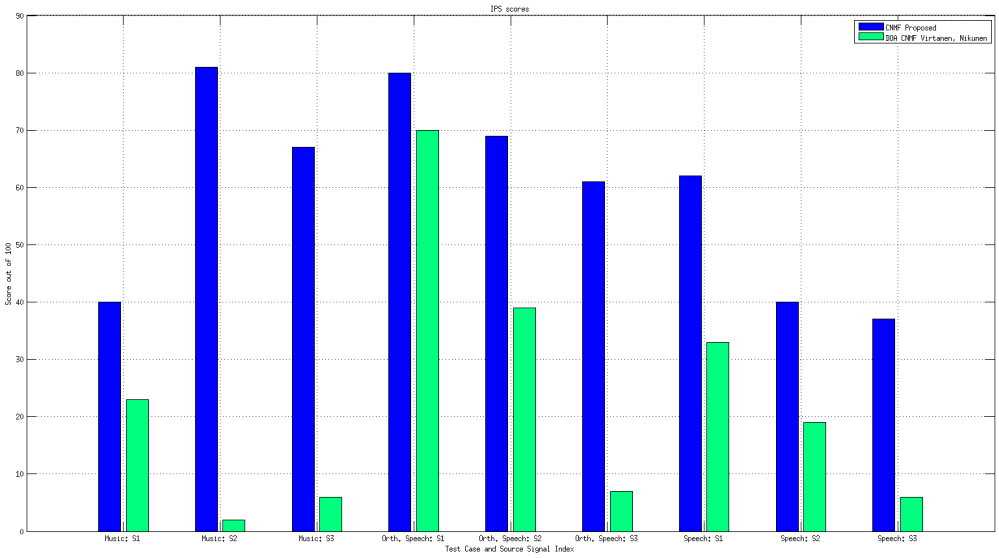

Figure D.28: Measured IPS per Source signal and per Test case

Finally, for the 7th and final metric, the APS metric, the proposed algorithm performs significantly worse than the reference DoA SCM NMF algorithm, which can be seen by considering either Figure D.29 or Figure D.29.

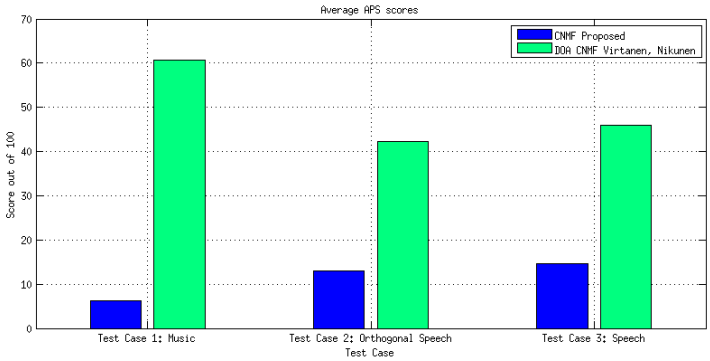

Figure D.29: Average APS per Test case

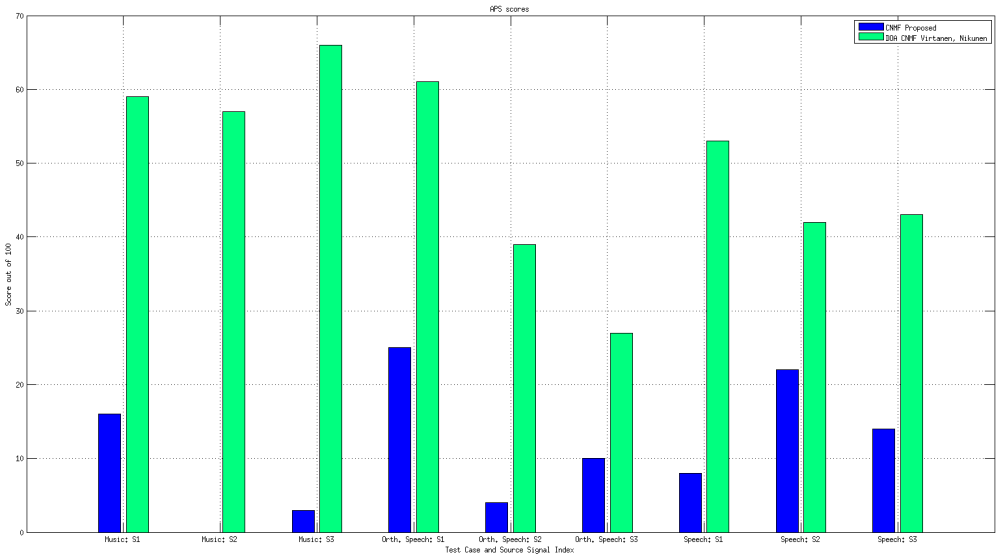

Figure D.30: Measured APS per Source signal and per Test case

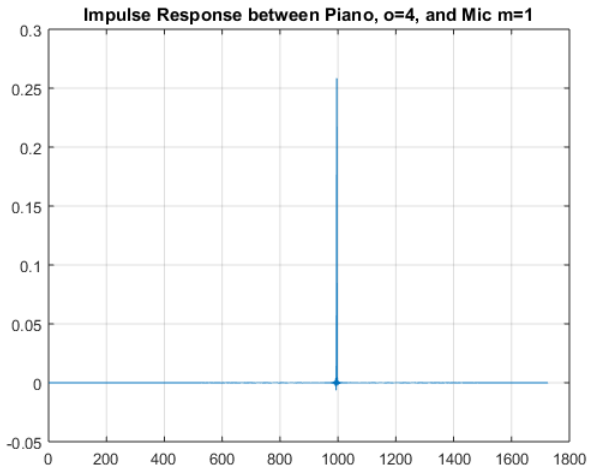

Figure D.31: Piano source to left microphone spatial impulse response sequence

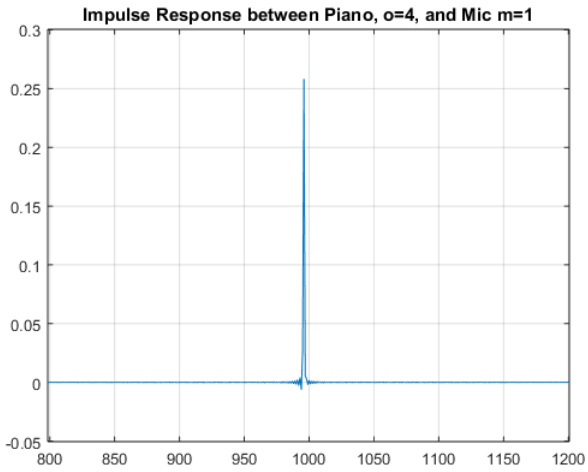

Figure D.32: Piano source to left microphone spatial impulse response sequence, zoomed

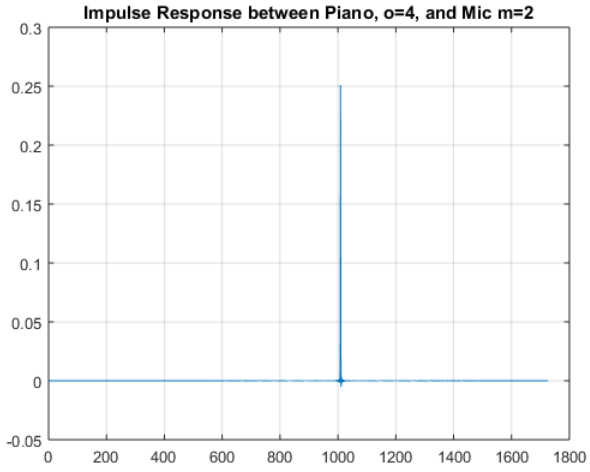

Figure D.33: Piano source to right microphone spatial impulse response filter

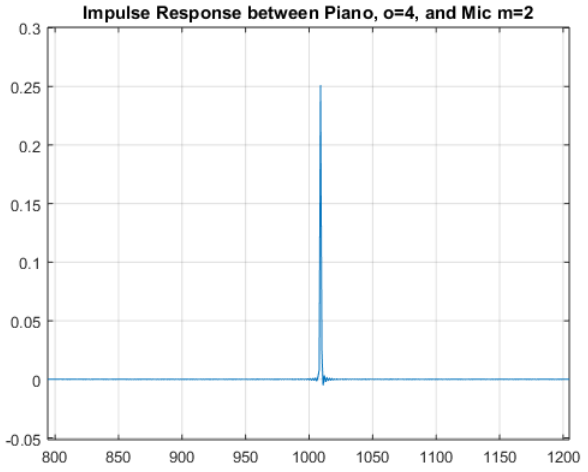

Figure D.34: Piano source to right microphone spatial impulse response filter, zoomed

D.4 Time Domain Impulse Response Filters for Test Case 1

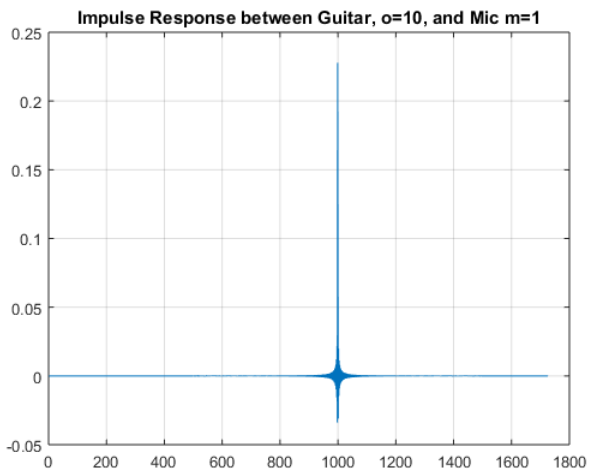

Figure D.35: Guitar source to left microphone spatial impulse response sequence

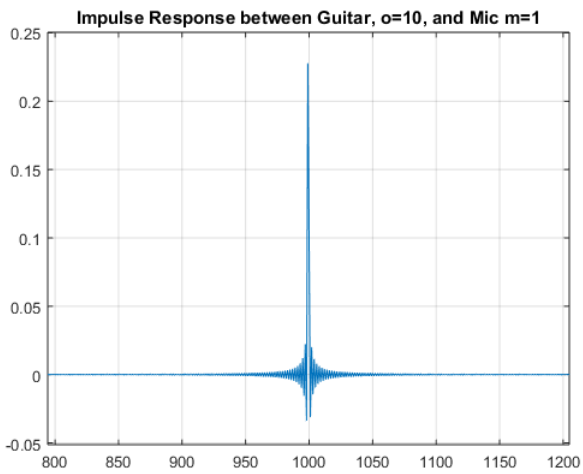

Figure D.36: Guitar source to left microphone spatial impulse response sequence, zoomed

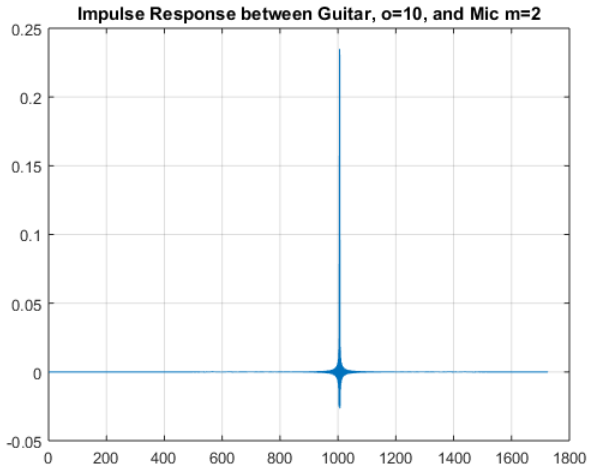

Figure D.37: Guitar source to right microphone spatial impulse response sequence

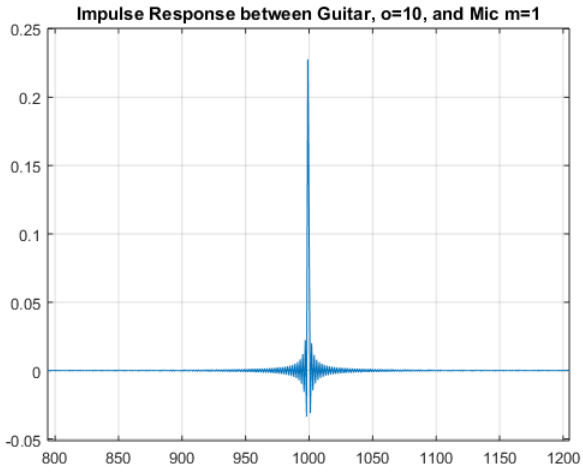

Figure D.38: Guitar source to right microphone spatial impulse response sequence, zoomed

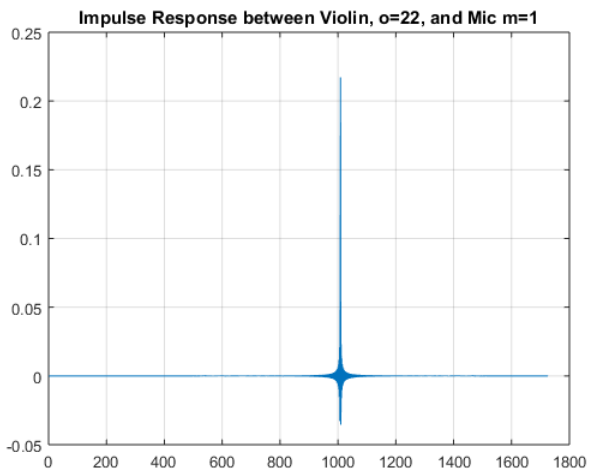

Figure D.39: Violin source to left microphone spatial impulse response sequence

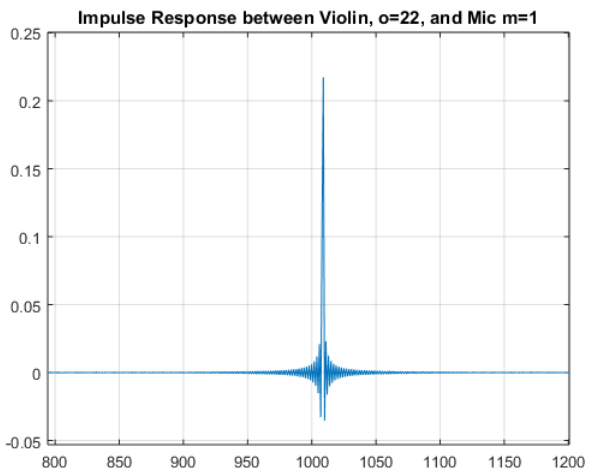

Figure D.40: Violin source to left microphone spatial impulse response sequence, zoomed

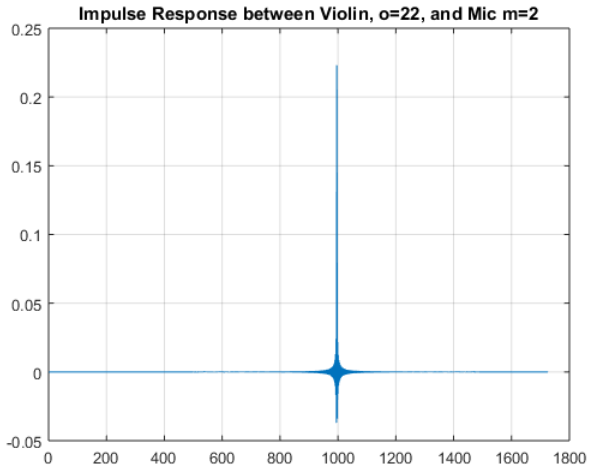

Figure D.41: Violin source to right microphone spatial impulse response sequence

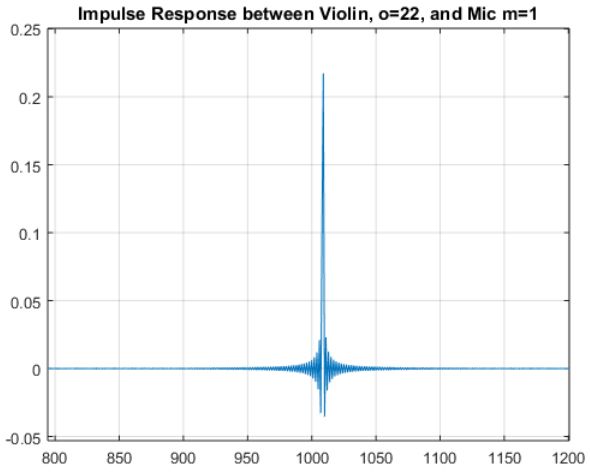

Figure D.42: Violin source to right microphone spatial impulse response sequence, zoomed

D.5 Input Signals for Supplementary Test Cases

D.5.0.1 Test Case 2

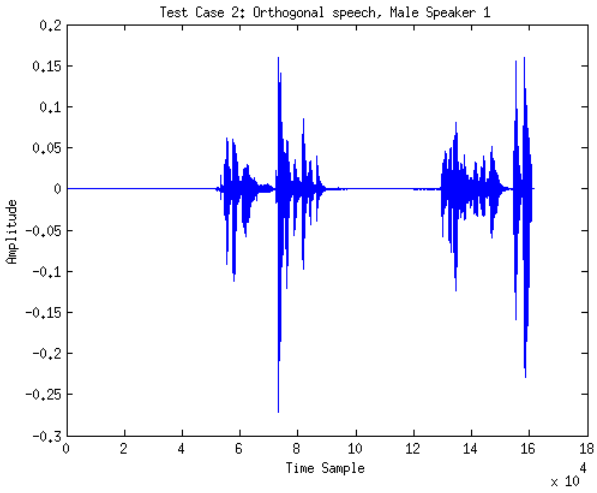

Figure D.43: Male Speaker 1, Time Domain

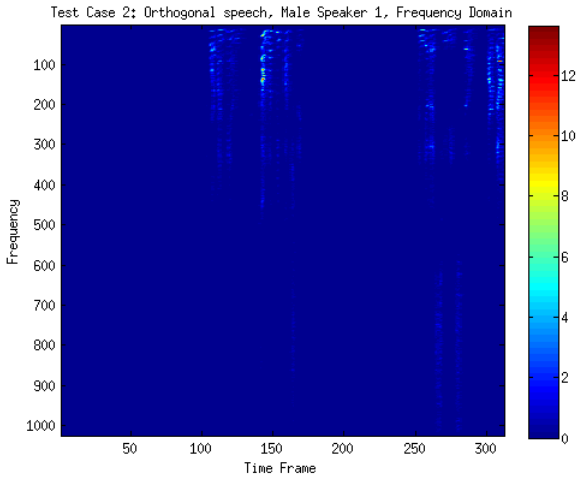

Figure D.44: Male Speaker 1, Frequency Domain

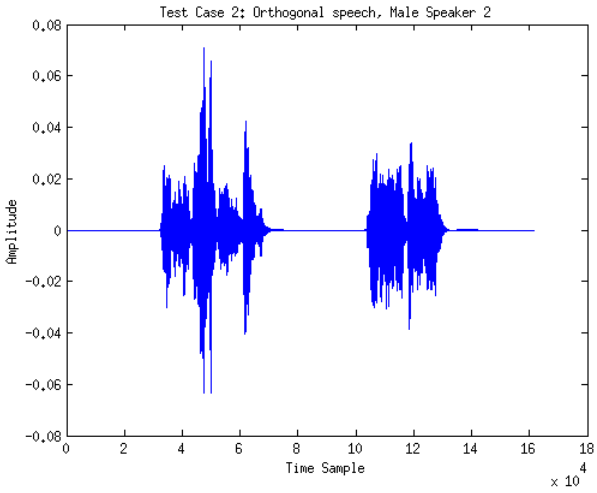

Figure D.45: Male Speaker 2, Time Domain

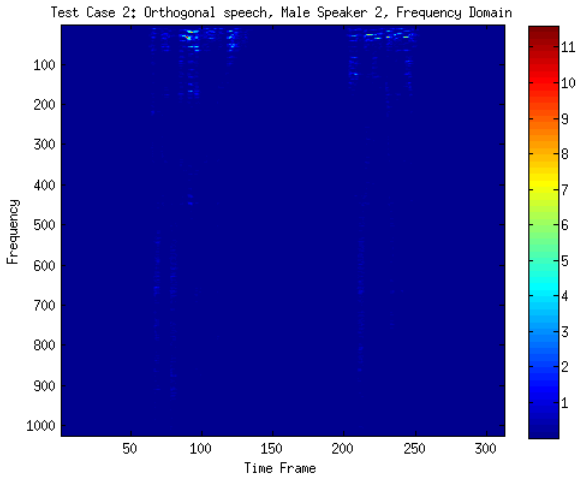

Figure D.46: Male Speaker 2, Frequency Domain

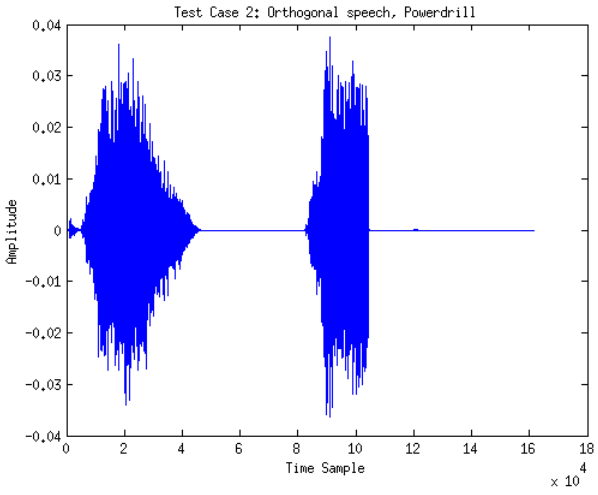

Figure D.47: Power Drill Signal, Time Domain

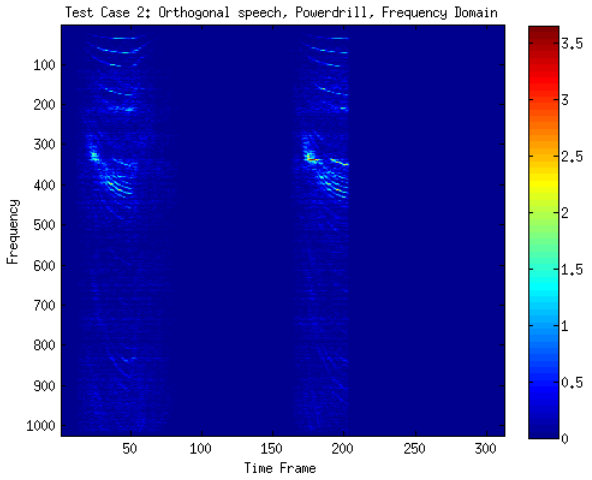

Figure D.48: Power Drill Signal, Frequency Domain

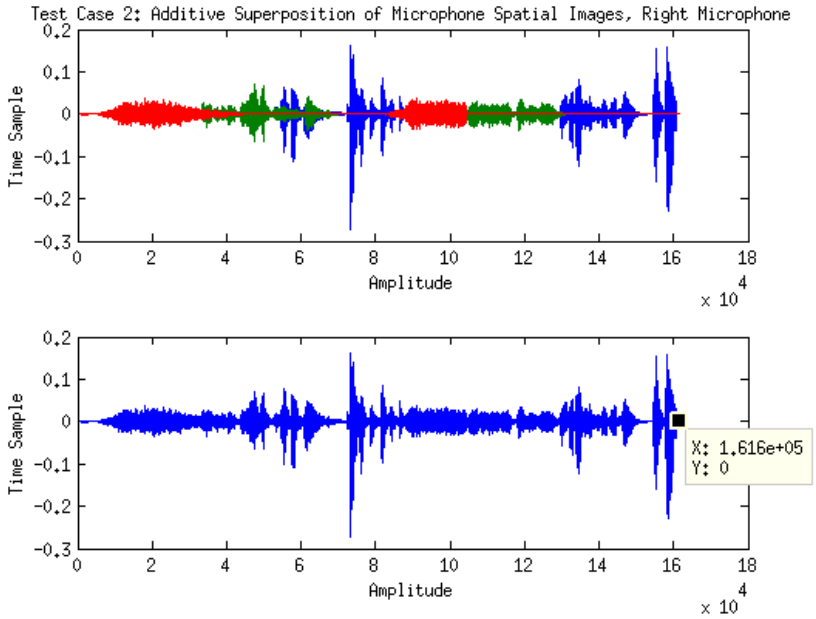

Figure D.49: Additive Superposition of the three source signal as seen by the right microphone

D.5.0.2 Test Case 3

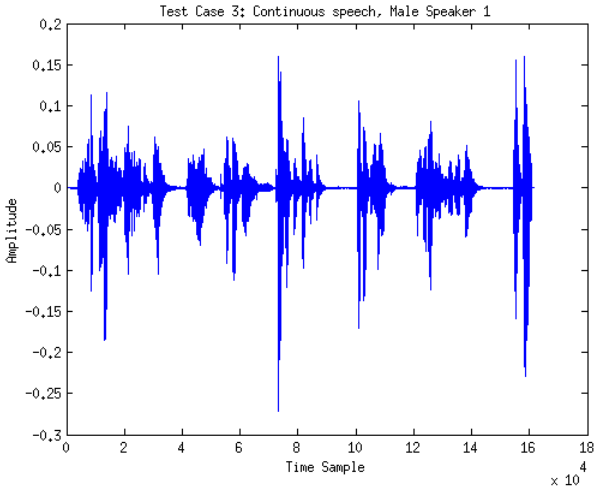

Figure D.50: Male Speaker 1, Time Domain

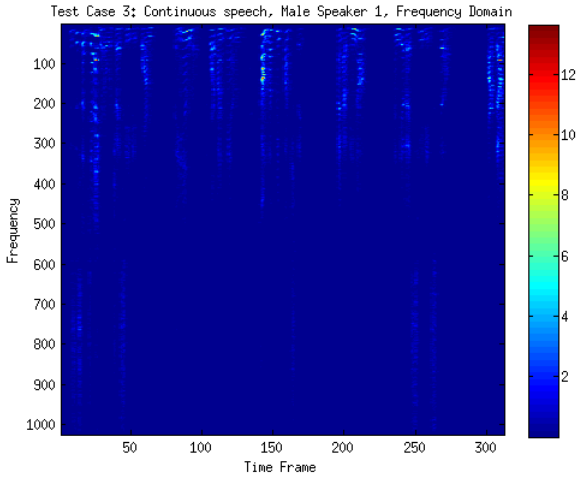

Figure D.51: Male Speaker 1, Frequency Domain

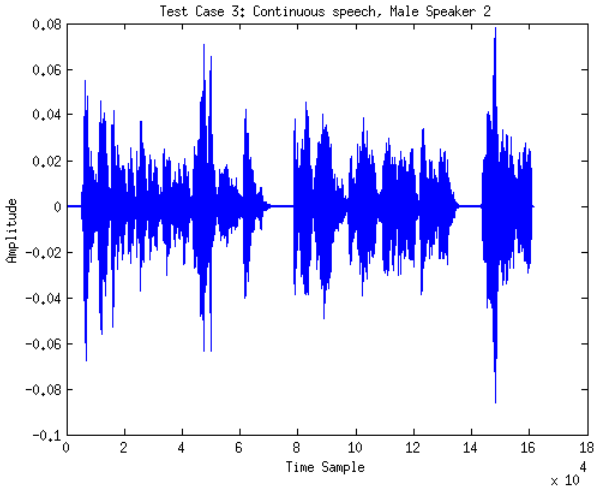

Figure D.52: Male Speaker 2, Time Domain

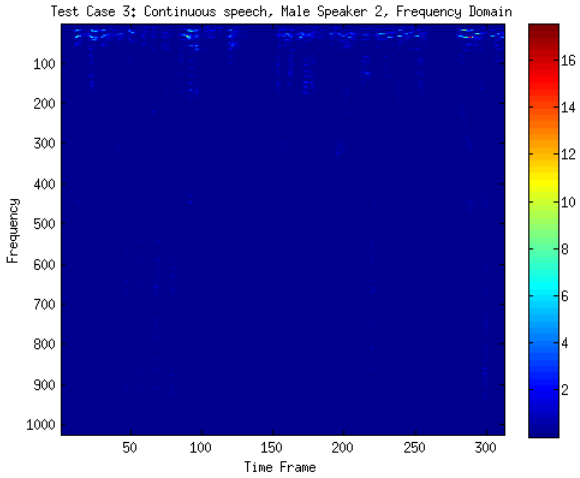

Figure D.53: Male Speaker 2, Frequency Domain

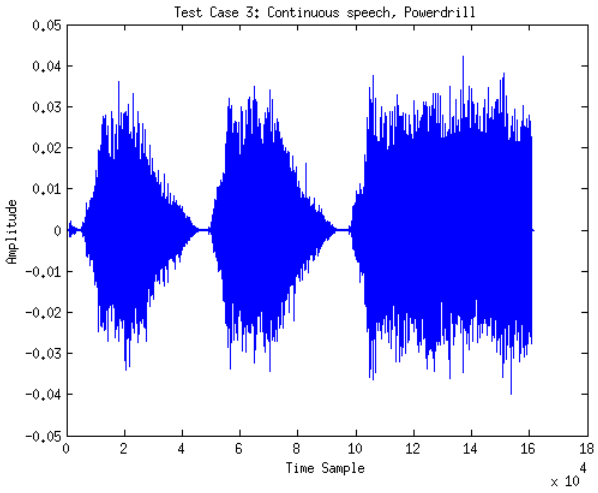

Figure D.54: Power Drill Signal, Time Domain

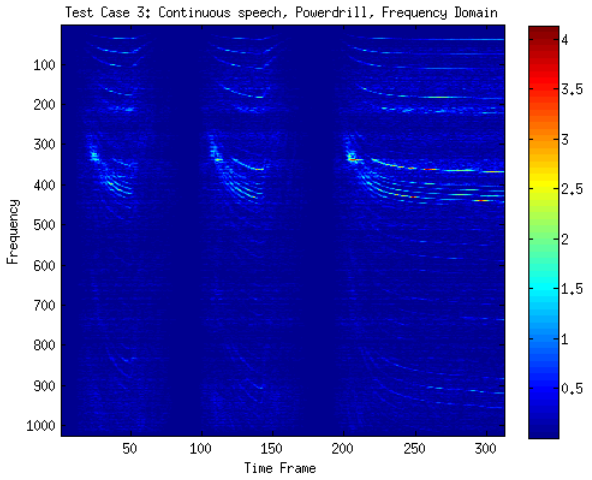

Figure D.55: Power Drill Signal, Frequency Domain

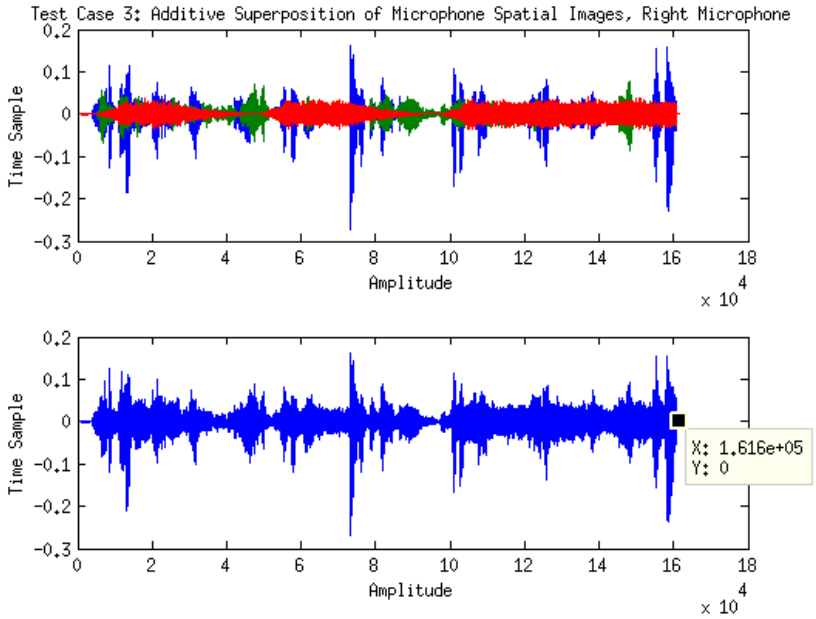

Figure D.56: Additive Superposition of the three source signal as seen by the right microphone

D.6 Simulation Results for Supplementary Test Cases

D.6.1 Test Case 2: Non Overlapping Speech

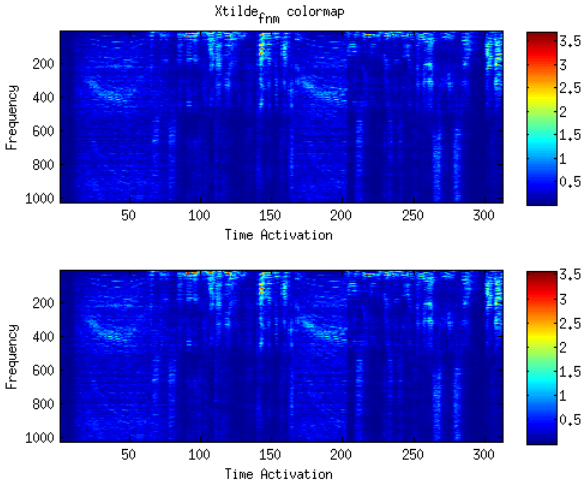

Figure D.57: Magnitude of Target Stereo STFT

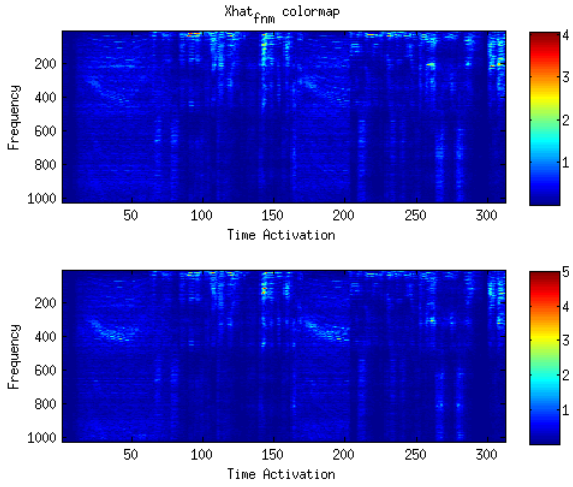

Figure D.58: Magnitude of Modelled Stereo STFT

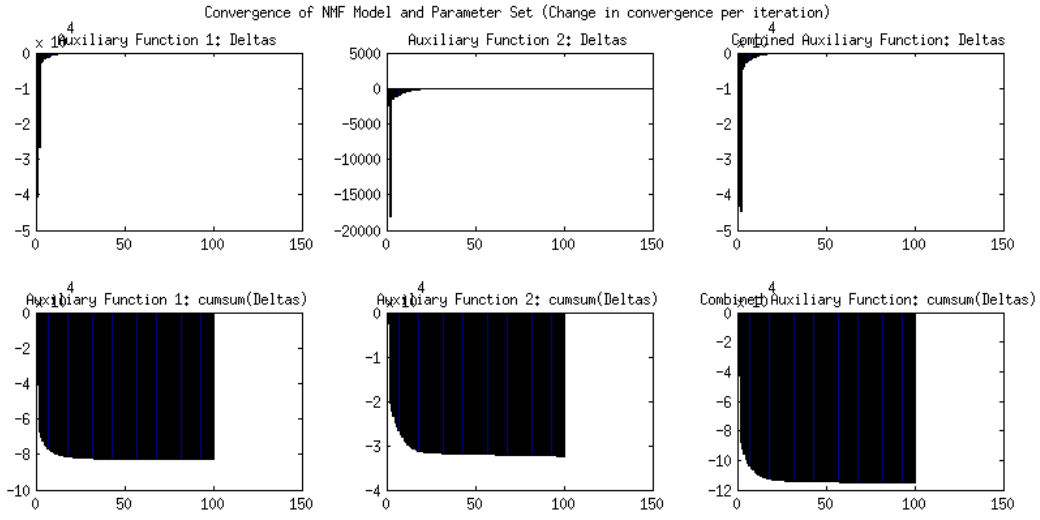

Figure D.59: Change in Convergence of NMF Parameter Set per Iteration

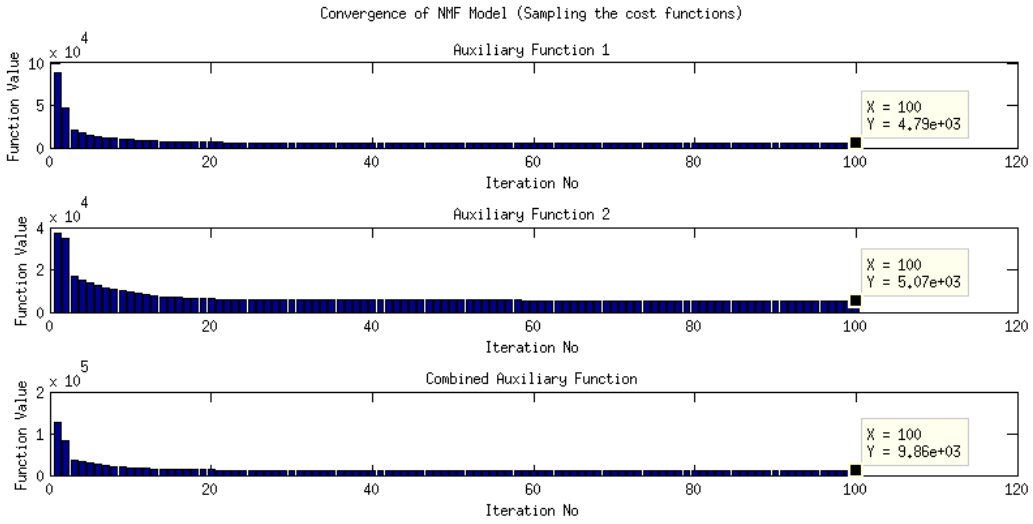

Figure D.60: Minimization of the Two Objective Functions separately and combined

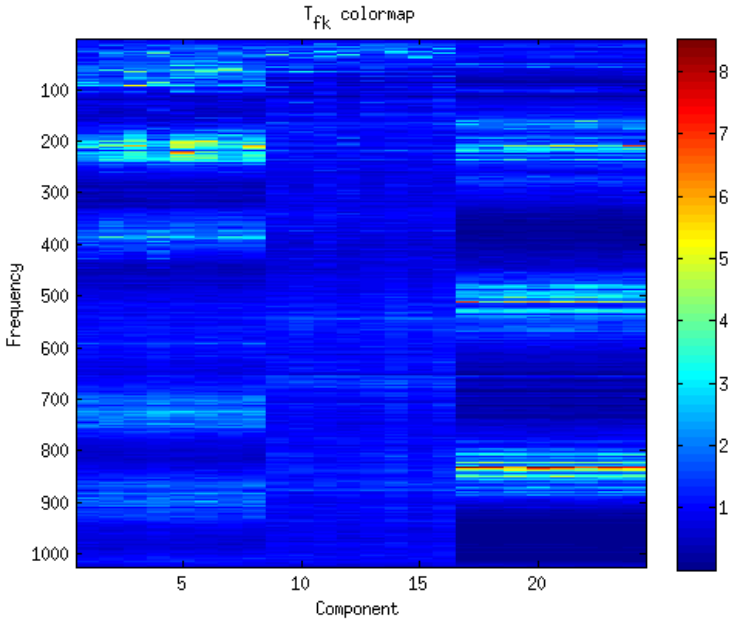

Figure D.61: Frequency Template Dictionary Matrix T_{fk}

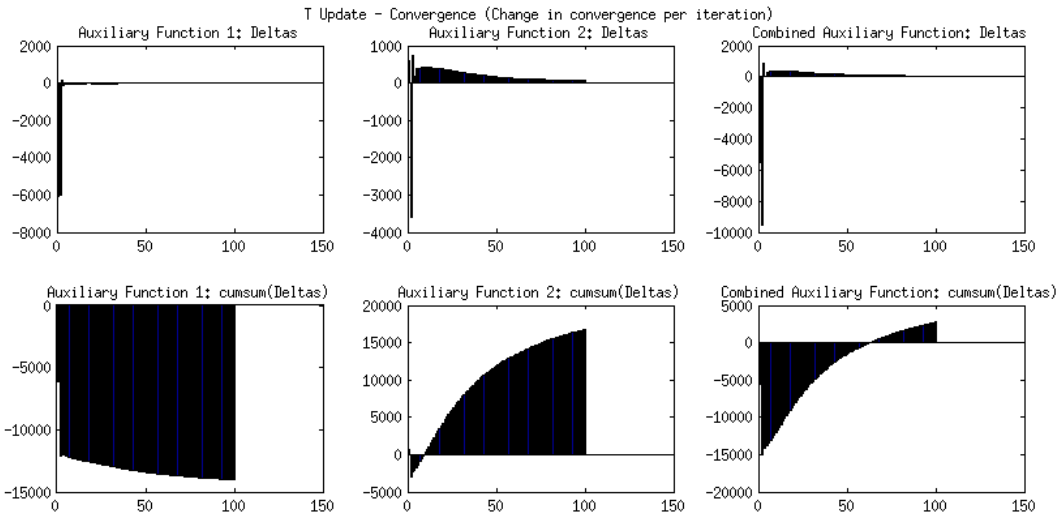

Figure D.62: Minimization Deltas of Cost Functions wrt Frequency Template Dictionary Matrix T_{fk}

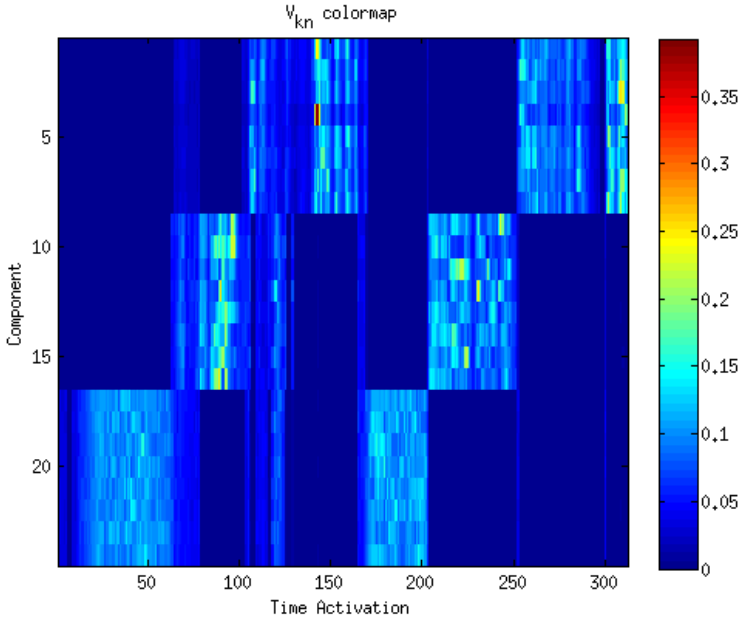

Figure D.63: Time Activation Matrix V_{kn}

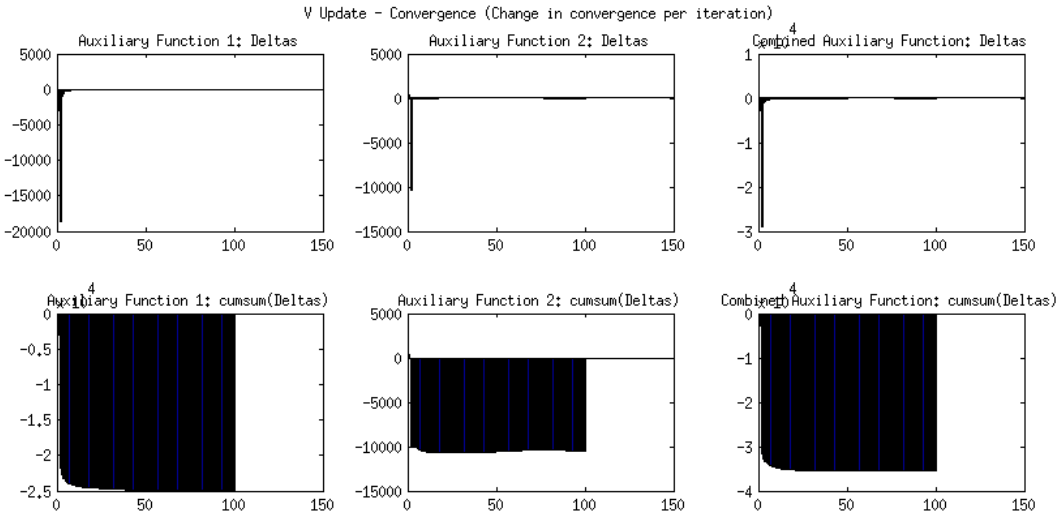

Figure D.64: Minimization Deltas of Cost Functions wrt Time Activation Matrix V_{kn}

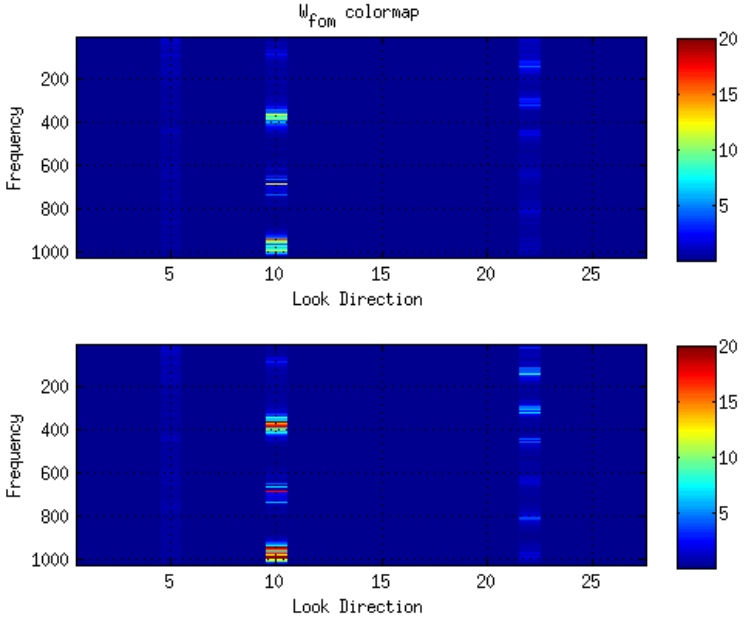

Figure D.65: Amplitude of Channel Mixing Tensor Parameter W_{fom}

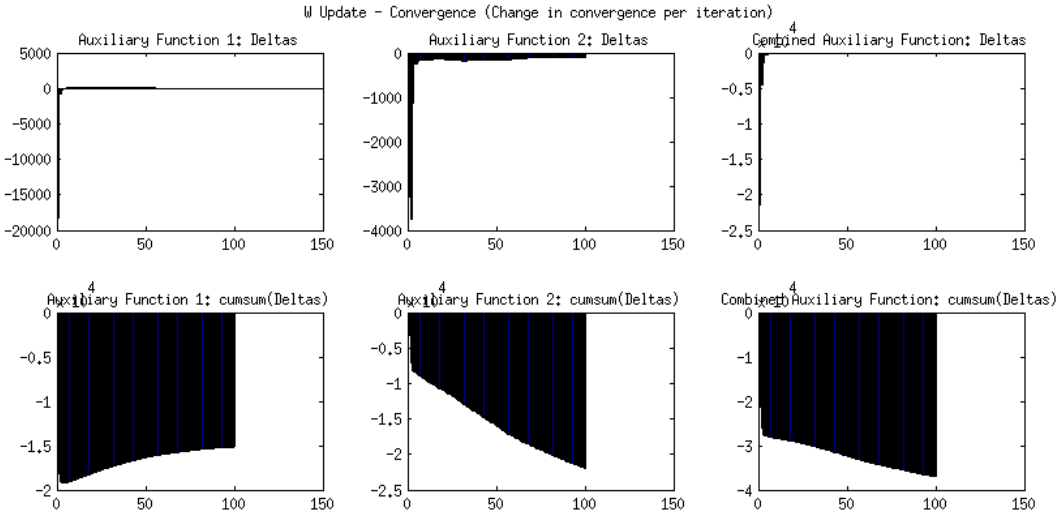

Figure D.66: Minimization Deltas of Cost Functions wrt Tensor Parameter W_{fom}

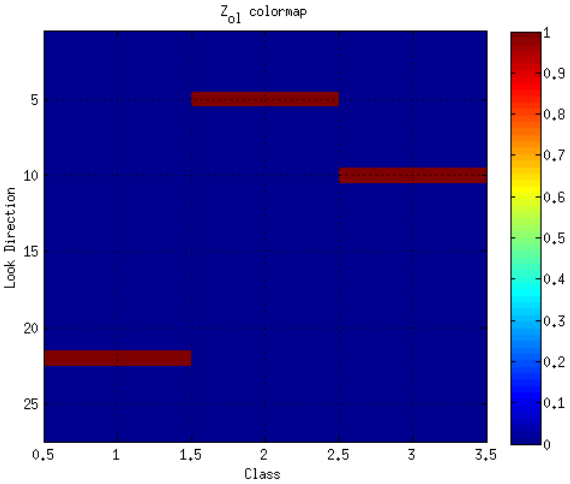

Figure D.67: Look Direction to Class Indicator Matrix Z_{ol}

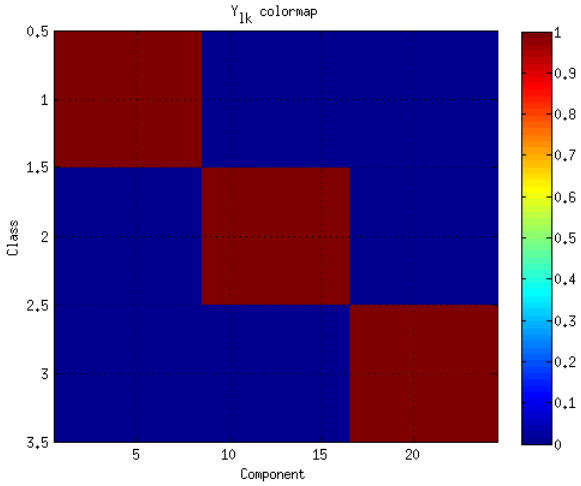

Figure D.68: Component to Class Indicator/Partition Matrix Y_{lk}

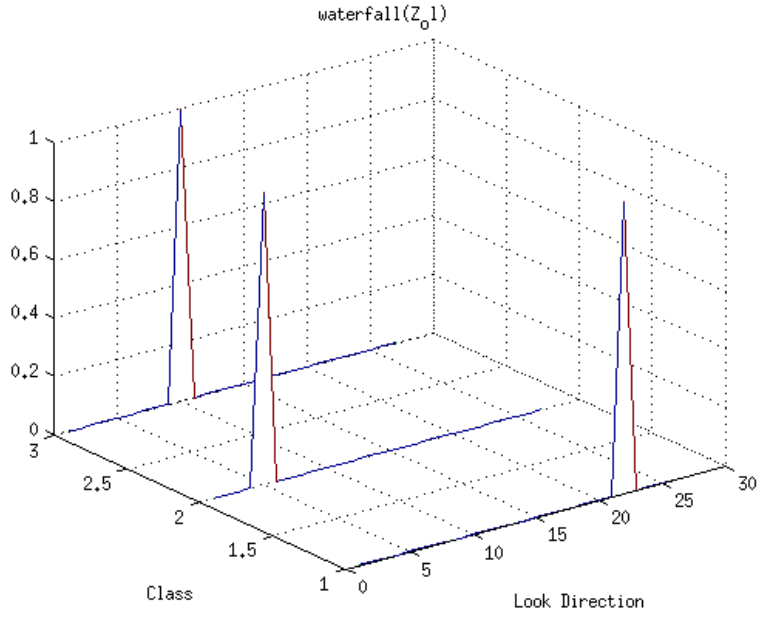

Figure D.69: Waterfall Plot of Look Direction to Class Indicator Matrix Z_{ol}

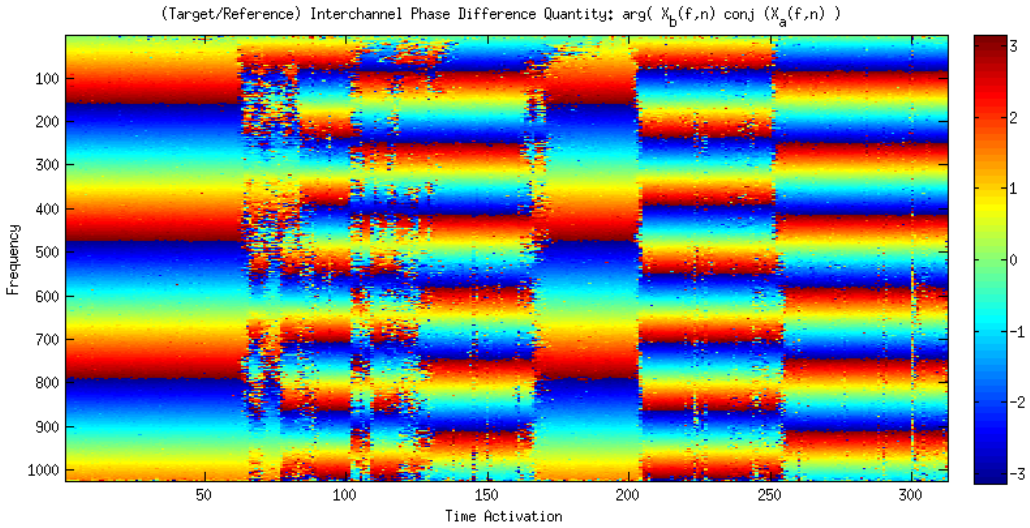

Figure D.70: Interchannel Phase Difference Quantity (obtained from the Reference STFT Data)

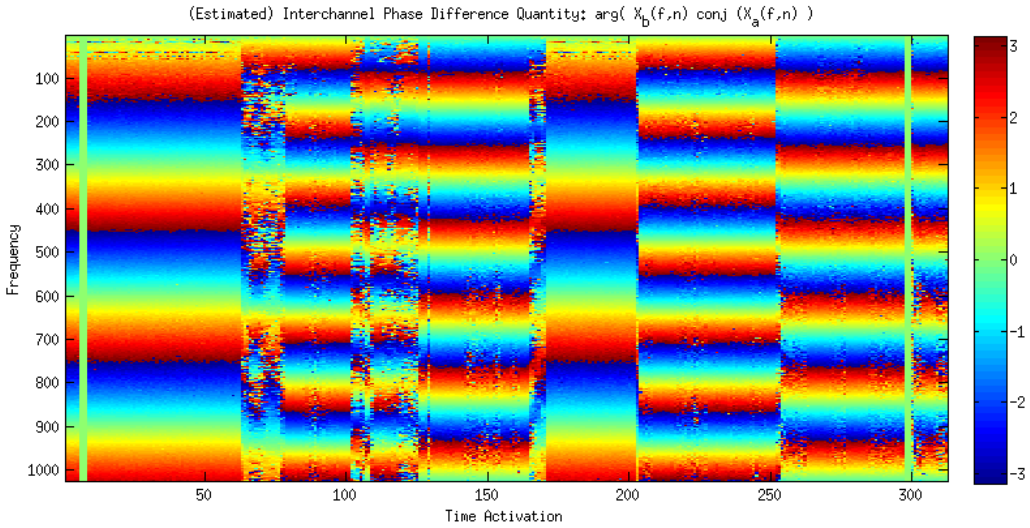

Figure D.71: Interchannel Phase Difference Quantity (modelled, estimated)

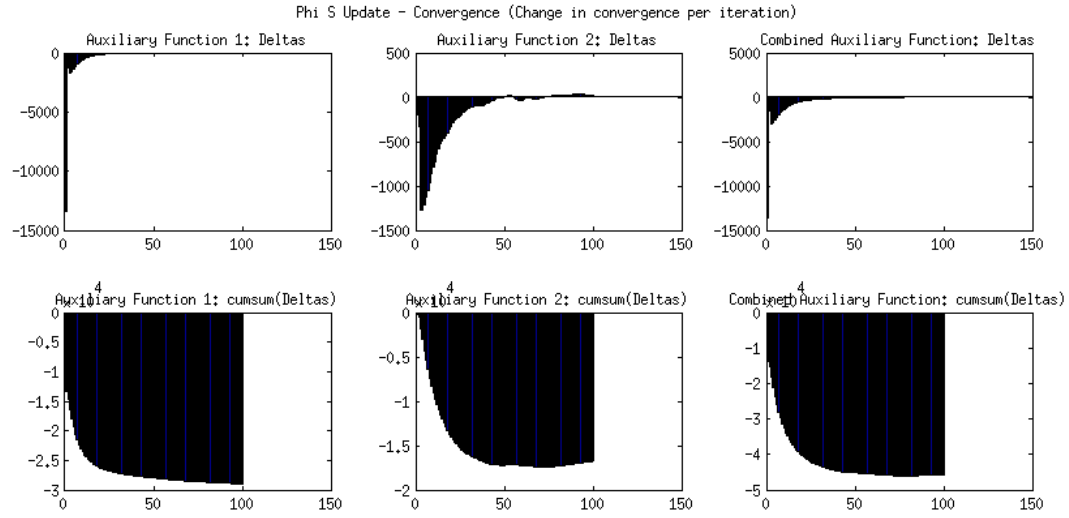

Figure D.72: Minimization Deltas of Cost Functions wrt complex Tensor Parameter $Phi_S(f, n, k)$

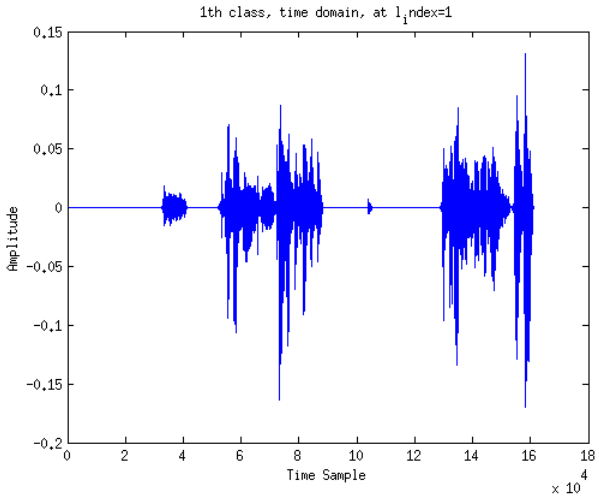

Figure D.73: Separated Output Class: Male Speaker 1, Time Domain

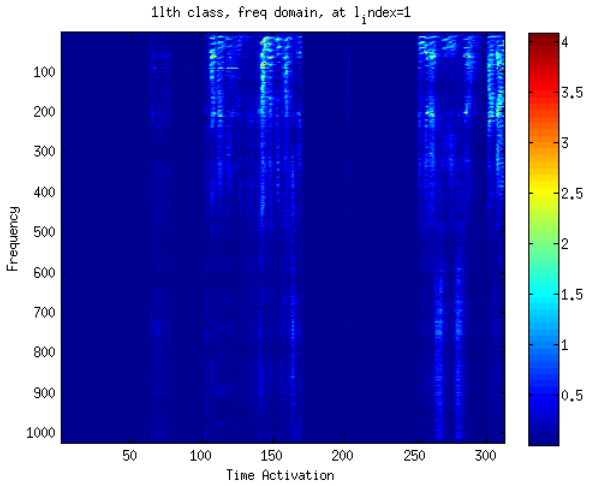

Figure D.74: Separated Output Class: Male Speaker 1, Frequency Domain

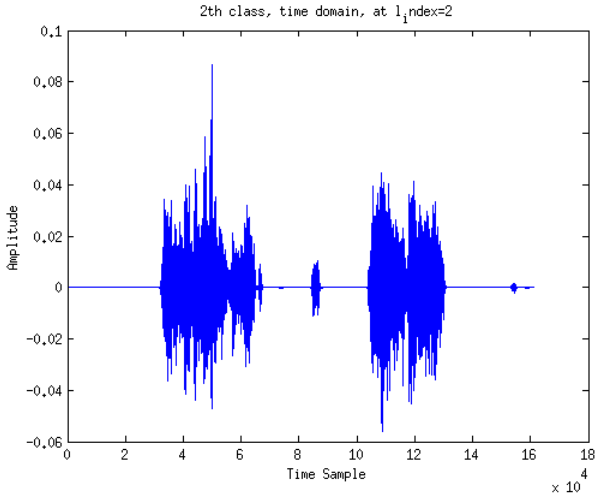

Figure D.75: Separated Output Class: Male Speaker 2, Time Domain

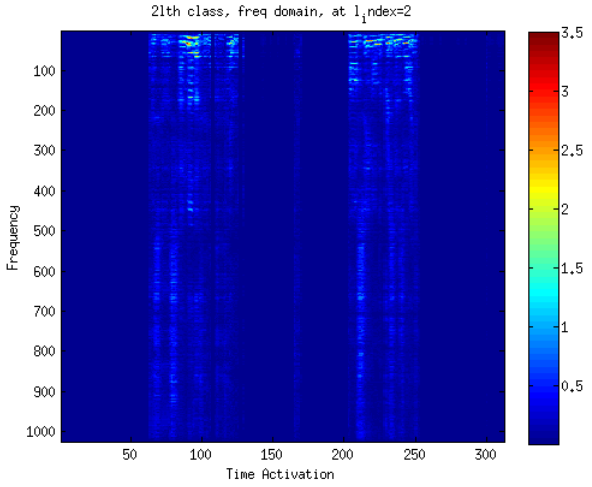

Figure D.76: Separated Output Class: Male Speaker 2, Frequency Domain

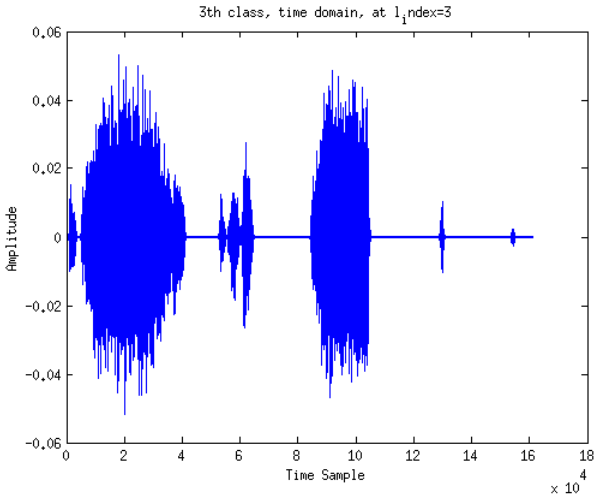

Figure D.77: Separated Output Class: Power Drill Signal, Time Domain

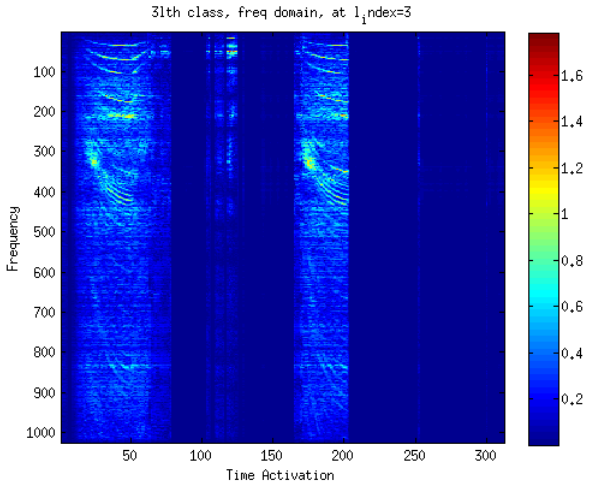

Figure D.78: Separated Output Class: Power Drill Signal, Frequency Domain

D.6.2 Test Case 3: Overlapping Speech

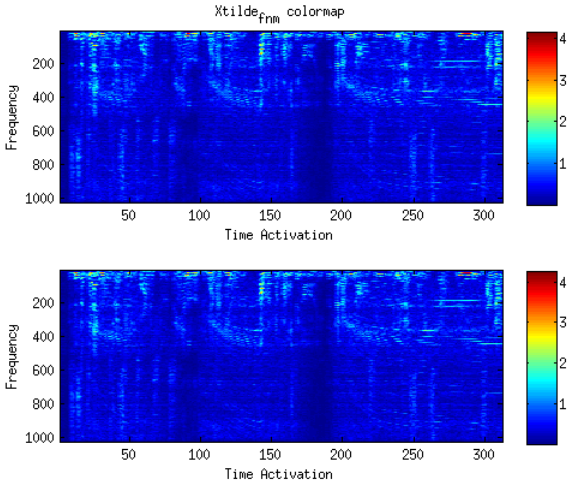

Figure D.79: Magnitude of Target Stereo STFT

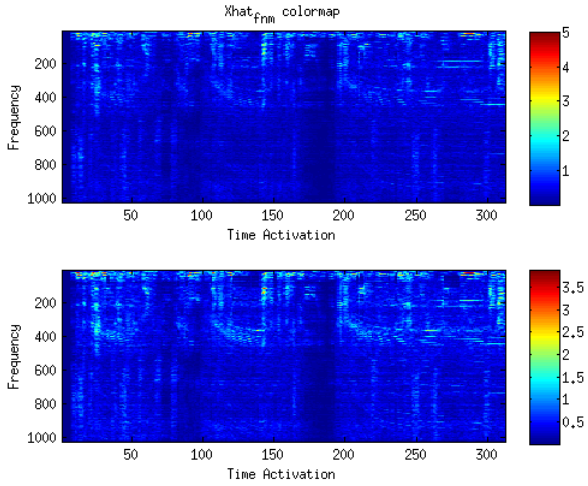

Figure D.80: Magnitude of Modelled Stereo STFT

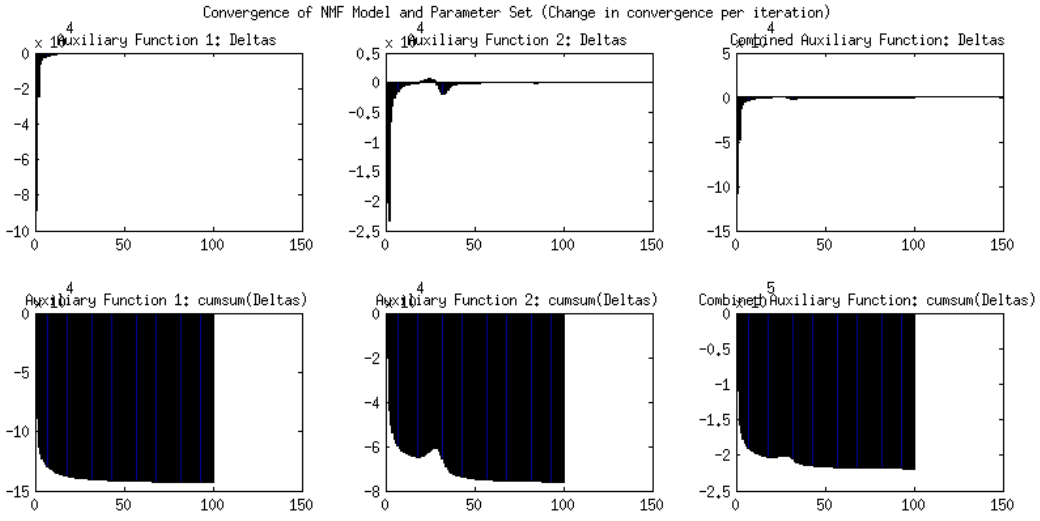

Figure D.81: Change in Convergence of NMF Parameter Set per Iteration

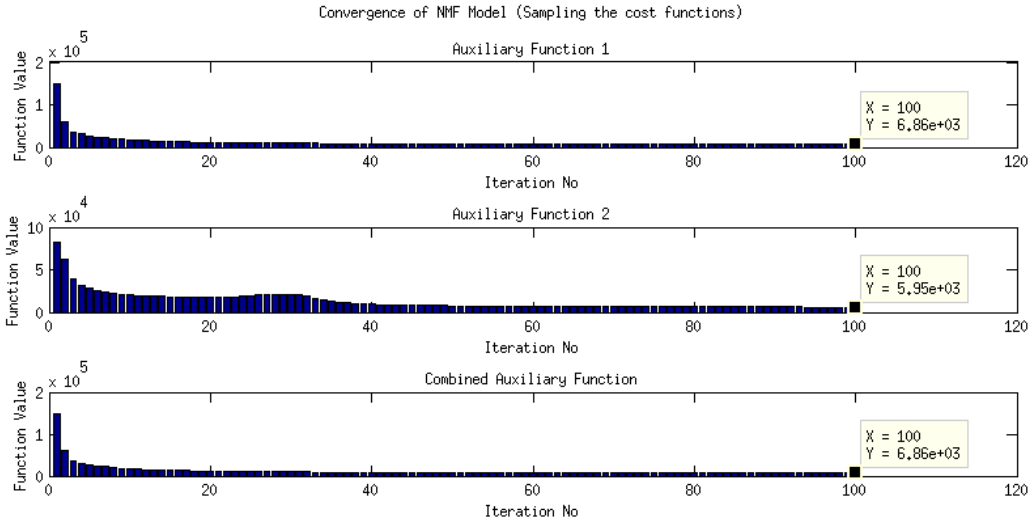

Figure D.82: Minimization of the Two Objective Functions separately and combined

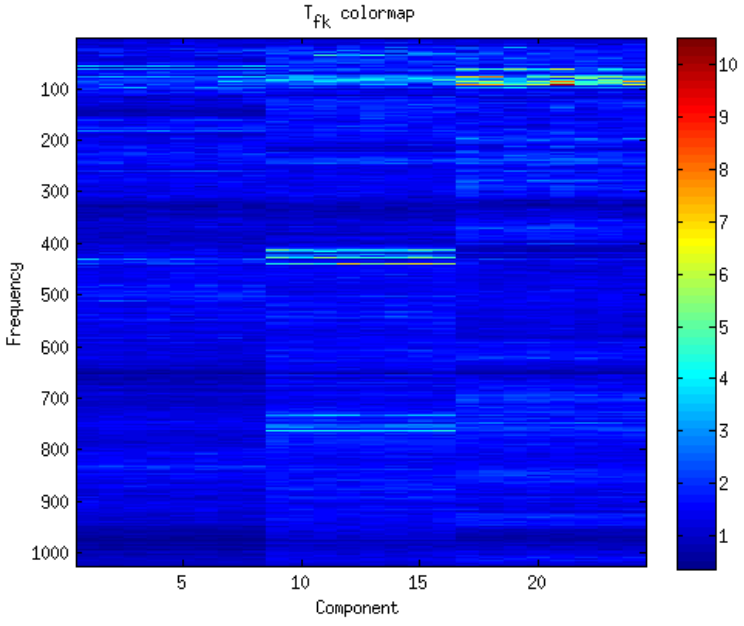

Figure D.83: Frequency Template Dictionary Matrix T_{fk}

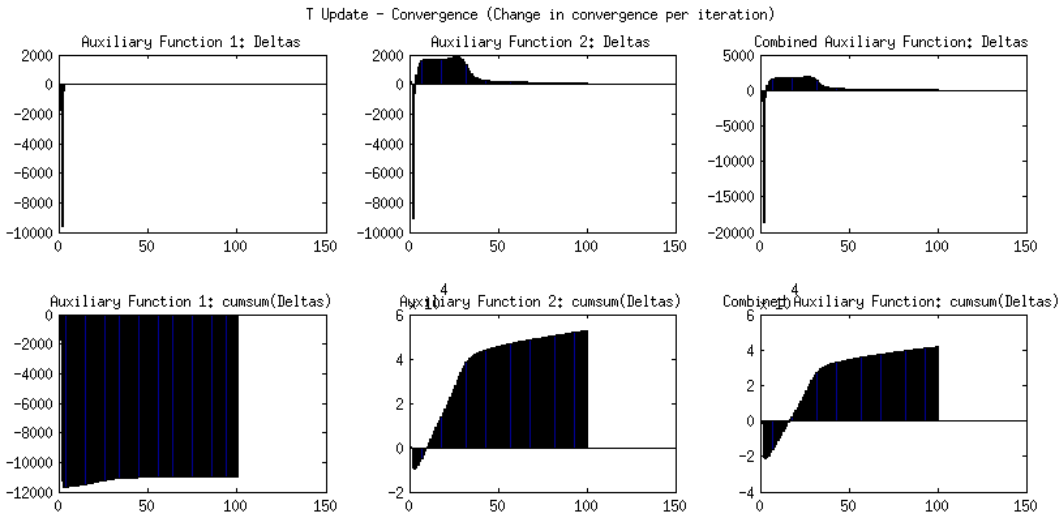

Figure D.84: Minimization Deltas of Cost Functions wrt Frequency Template Dictionary Matrix T_{fk}

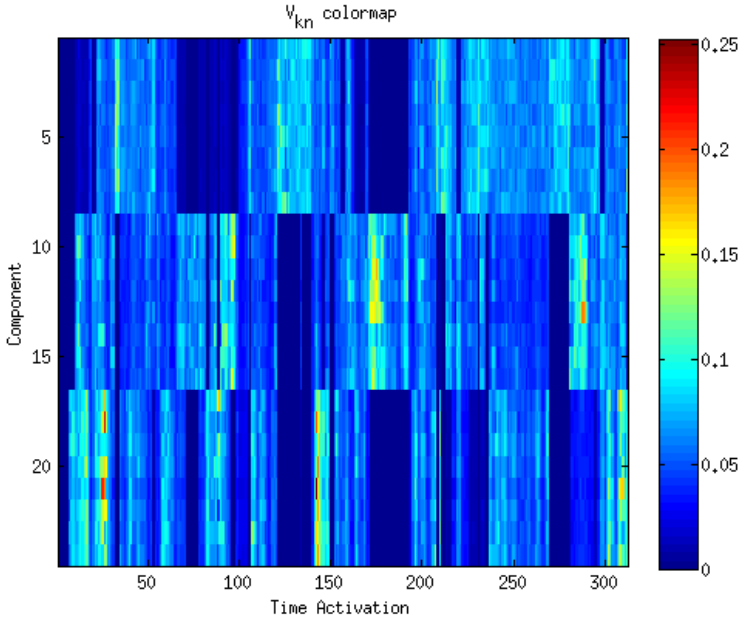

Figure D.85: Time Activation Matrix V_{kn}

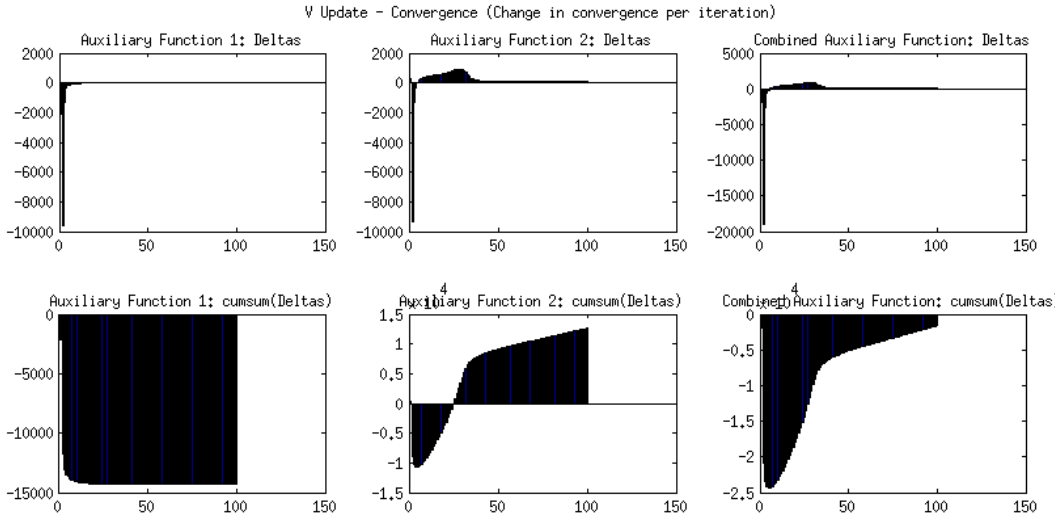

Figure D.86: Minimization Deltas of Cost Functions wrt Time Activation Matrix V_{kn}

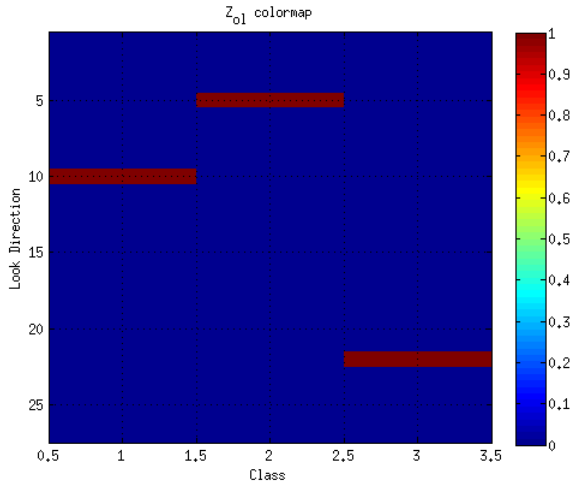

Figure D.87: Look Direction to Class Indicator Matrix Z_{ol}

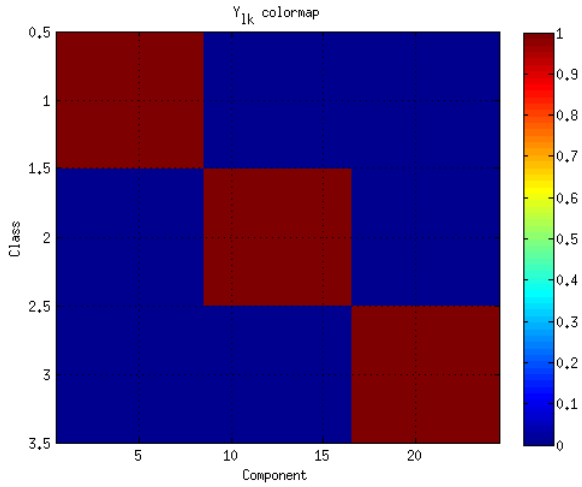

Figure D.88: Component to Class Indicator/Partition Matrix Y_{lk}

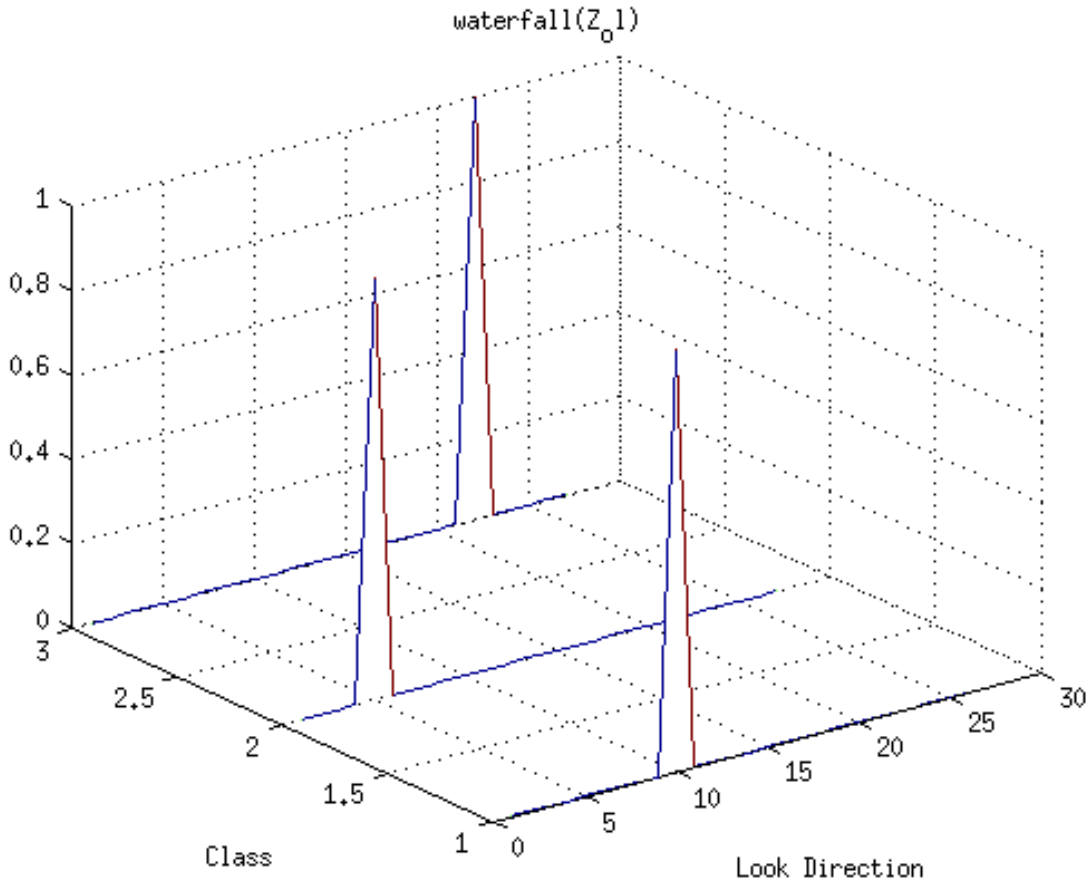

Figure D.89: Waterfall Plot of Look Direction to Class Indicator Matrix Z_{ol}

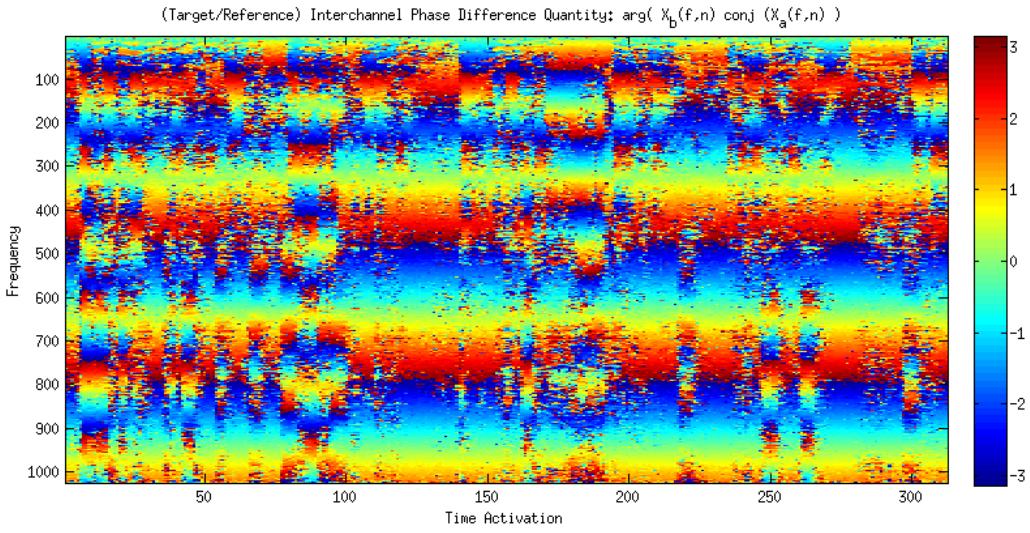

Figure D.90: Interchannel Phase Difference Quantity (obtained from the Reference STFT Data)

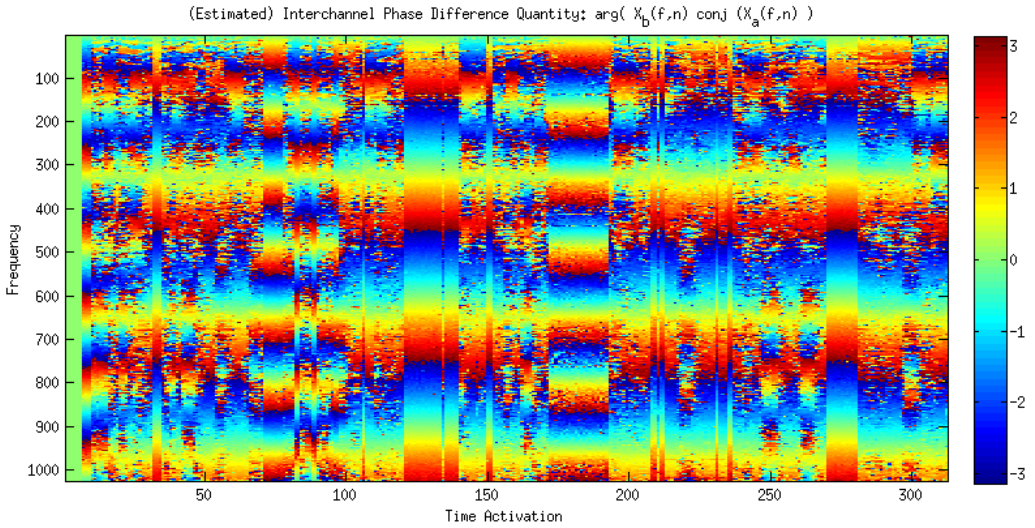

Figure D.91: Interchannel Phase Difference Quantity (modelled, estimated)

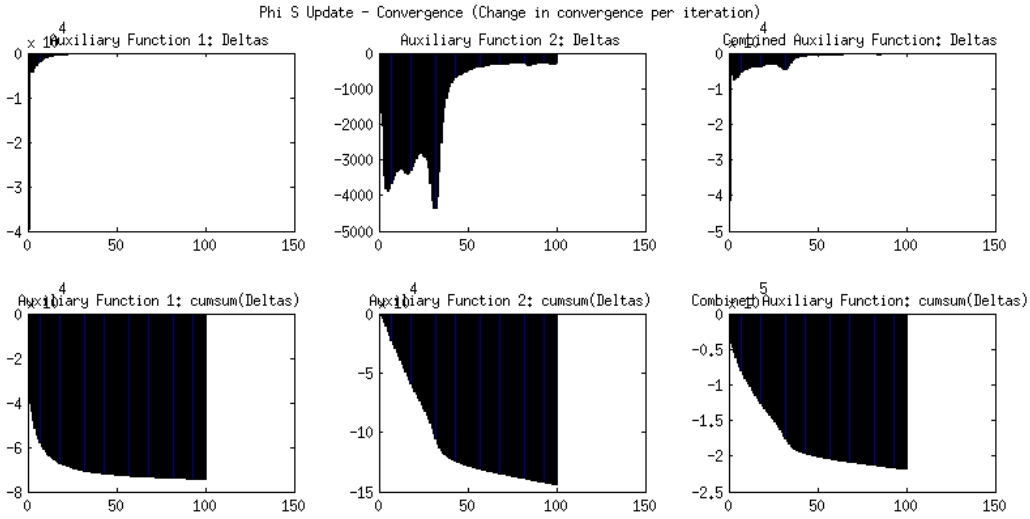

Figure D.92: Minimization Deltas of Cost Functions wrt complex Tensor Parameter $Phi_S(f, n, k)$

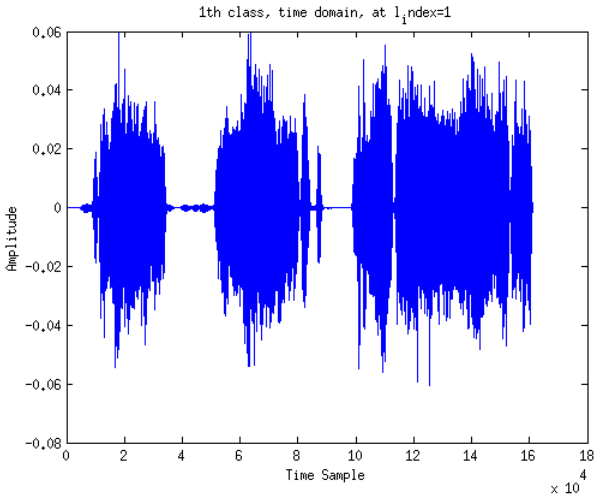

Figure D.93: Separated Output Class: Power Drill Signal, Time Domain

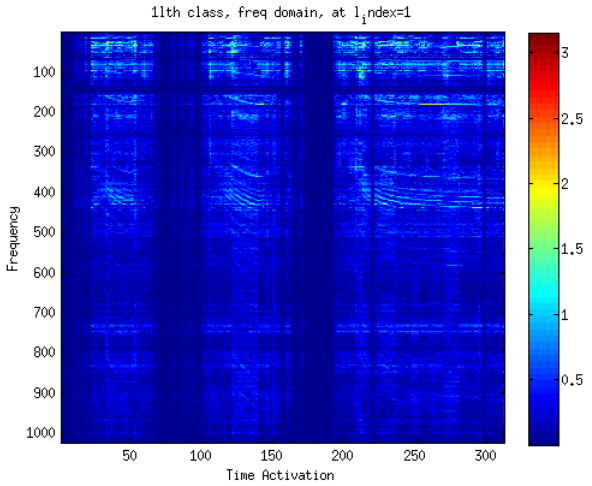

Figure D.94: Separated Output Class: Power Drill Signal, Frequency Domain

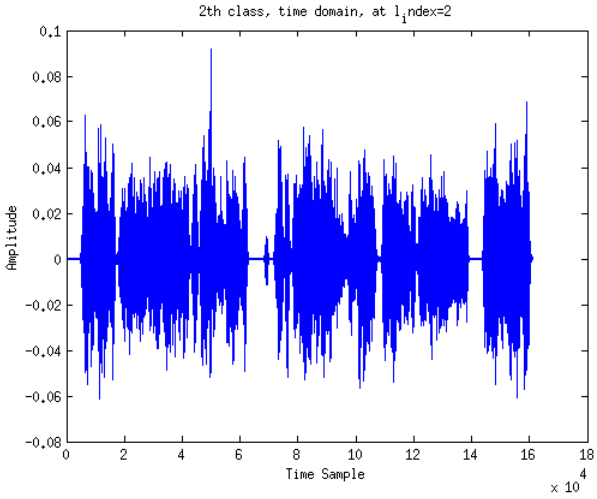

Figure D.95: Separated Output Class: Male Speaker 2, Time Domain

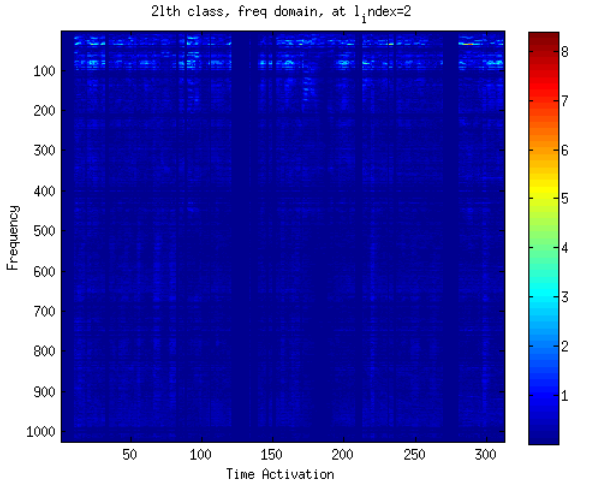

Figure D.96: Separated Output Class: Male Speaker 2, Frequency Domain

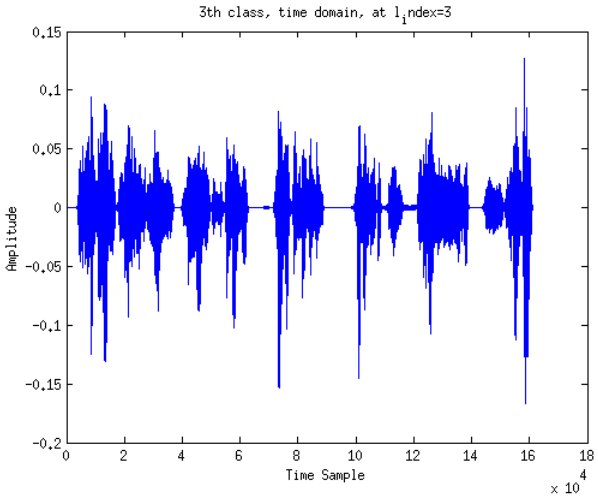

Figure D.97: Separated Output Class: Male Speaker 1, Time Domain

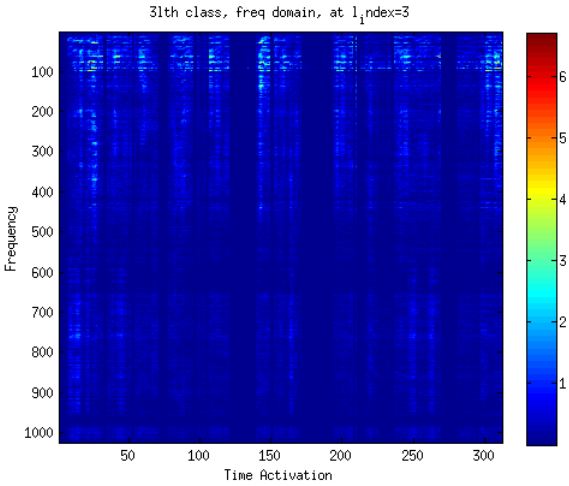

Figure D.98: Separated Output Class: Male Speaker 1, Frequency Domain