UMA INVESTIGAÇÃO SOBRE MÉTODOS DE SEPARAÇÃO CEGA DE FONTES SONORAS ENVOLVENDO REPRESENTAÇÕES NÃO-NEGATIVAS E DIVERSIDADE ESPACIAL

Claudio Romero

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

Orientadores: Wallace Alves Martins
Luiz Wagner Pereira Biscainho

Rio de Janeiro
Março de 2017

UMA INVESTIGAÇÃO SOBRE MÉTODOS DE SEPARAÇÃO CEGA DE FONTES SONORAS ENVOLVENDO REPRESENTAÇÕES NÃO-NEGATIVAS E DIVERSIDADE ESPACIAL

Claudio Romero

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

Examinada por:

Prof. Wallace Alves Martins, D.Sc.

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. Diego Barreto Haddad, D.Sc.

Prof. Flávio Rainho Ávila, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2017

Romero, Claudio

Uma Investigação sobre Métodos de Separação Cega de Fontes Sonoras Envolvendo Representações Não-negativas e Diversidade Espacial/Claudio Romero. – Rio de Janeiro: UFRJ/COPPE, 2017.

IX, 78 p.: il.; 29, 7cm.

Orientadores: Wallace Alves Martins

Luiz Wagner Pereira Biscainho

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2017.

Referências Bibliográficas: p. 74 – 78.

1. Separação de fontes. 2. NMF. 3. NTF. 4. Matriz de covariância espacial (SCM). 5. SRP-PHAT. I. Martins, Wallace Alves *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

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

UMA INVESTIGAÇÃO SOBRE MÉTODOS DE SEPARAÇÃO CEGA DE FONTES SONORAS ENVOLVENDO REPRESENTAÇÕES NÃO-NEGATIVAS E DIVERSIDADE ESPACIAL

Claudio Romero

Março/2017

Orientadores: Wallace Alves Martins
Luiz Wagner Pereira Biscainho

Programa: Engenharia Elétrica

A separação cega de fontes é um problema com diversas aplicações em várias áreas e que, por isso, vem sendo alvo de um grande número de pesquisas. Este trabalho foca no estudo do problema de separação cega de fontes sonoras utilizando representações não-negativas com o aproveitamento da diversidade espacial permitido pelo desenvolvimento de métodos multicanais, que abriram novas oportunidades de pesquisa e resultaram no surgimento de novas modelagens para o problema de separação de fontes.

Neste trabalho são estudados dois algoritmos distintos, a NMF-SCM (separação de áudio multicanal utilizando fatoração não-negativa de matrizes e com modelo de covariância espacial baseado em direção-de-chegada), que representa o estado da arte da modelagem deste problema, modelando não apenas as características das fontes, mas também o ambiente em que a mistura foi capturada; e a NTF (fatoração de tensores não-negativos), que apresenta uma modelagem simplificada do problema multicanal, de forma análoga à NMF (fatoração não-negativa de matrizes), e que não utiliza explicitamente a diversidade espacial.

Durante o desenvolvimento deste trabalho ambos os algoritmos foram implementados. Uma versão vetorizada e paralelizada da NMF-SCM é apresentada, assim como alterações ao algoritmo da NTF visando à melhoria em seu desempenho e também à utilização explícita da diversidade espacial. Por último, é proposto um método para a determinação cega do número de fontes presentes em misturas multicanais.

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

AN INVESTIGATION ON BLIND SOUND SOURCE SEPARATION METHODS INVOLVING NON-NEGATIVE REPRESENTATIONS AND SPATIAL DIVERSITY

Claudio Romero

March/2017

Advisors: Wallace Alves Martins
Luiz Wagner Pereira Biscainho

Department: Electrical Engineering

The problem of blind source separation finds many applications across different areas, thus justifying the ever increasing number of works in this topic. This work focuses on studying this problem for sound sources, employing non-negative signals' representations, while also taking advantage of the spatial diversity induced by the use of multiple channels; this particular feature has recently opened up new research directions regarding the proper modeling of multichannel source separation

This work studies two different algorithms: NMF-SCM (sound source separation using non-negative matrix factorization and direction-of-arrival-based spatial covariance model), whose model represents the state of the art, taking in consideration not only the characteristics of the sources but also the enviroment into which they were captured on; and NTF (non-negative tensor factorization), whose simplified model is the multichannel equivalent of NMF (non-negative matrix factorization).

During the development of this work both algorithms were implemented. A vectorized and parallelized NMF-SCM implementation is presented; and some improvements are proposed to the NTF algorithm, as well as a method for blind determination of the number of sources in multichannel mixtures.

Sumário

Lista de Figuras

Capítulo 1

Introdução

A separação de fontes é um problema que surge em diversas aplicações, como, por exemplo, o monitoramento do espectro eletromagnético para fins civis e militares em telecomunicações, a interpretação e identificação de sinais em eletroencefalogramas intracranianos na área biomédica e a separação (ou extração) de sinais provenientes de instrumentos musicais distintos na área de áudio.

A definição do que é uma fonte depende da aplicação; nos exemplos anteriores uma fonte seria desde um dispositivo que emite ondas eletromagnéticas, passando por uma região cerebral até instrumentos musicais, respectivamente. Uma fonte pode ser ainda algo abstrato. Por exemplo, pode-se interpretar como separação de fontes uma aplicação de remoção de ruídos; nesse caso, (ao menos) uma das fontes será similar aos exemplos anteriores, enquanto a outra fonte será uma abstração, já que ela compreende na verdade todo um conjunto de fontes de ruído. Uma interpretação equivalente pode ser feita para aplicações de extração de fontes.

O chamado problema da festa de coquetel (em inglês, *cocktail party problem*), um exemplo clássico de separação de fontes, ilustra nossa capacidade de reconhecer e entender o que uma pessoa está dizendo quando várias outras falam ao mesmo tempo. O problema propriamente dito é: como projetar uma máquina com essa mesma capacidade? Em [1], onde o problema foi nomeado, realizaram-se uma análise e alguns experimentos para avaliar objetivamente essa capacidade.

Nos últimos 20 anos a separação de fontes recebeu bastante atenção, o que resultou no desenvolvimento e aprimoramento de várias abordagens, como a análise de componentes independentes [2] (ICA – *Indendent Component Analysis*), muito utilizada na área biomédica e particularmente eficaz para misturas instantâneas, e a fatoração de matrizes não-negativas em matrizes não-negativas [3] (NMF – *Non-Negative Matrix Factorization*), também utilizada em várias outras aplicações, como mineração de texto [3]. Cada uma dessas abordagens apresenta particularidades (e limitações) que são alvo de estudos que resultaram na proposição de várias modificações e evoluções, como a análise de subespaços independentes [4] (ISA –

Independent Subspace Analysis, generalização da ICA) e a fatoração não-negativa e deconvolutiva de matrizes não-negativas [5] (NMFD – *Non-Negative Matrix Factor Deconvolution*, generalização da NMF).

As diferentes abordagens mencionadas acima utilizam ao menos uma das seguintes características dos sinais (ou de suas representações):

- Independência: As fontes originais são independentes entre si, logo um algoritmo de separação de fontes tratará de minimizar a dependência entre as estimativas. No caso do algoritmo de ICA, que será visto mais adiante, o objetivo é minimizar a gaussianidade das estimativas.

- Esparsidade: As fontes originais não estão ativas simultaneamente ao menos em alguns momentos (e/ou frequências) da mistura, logo um algoritmo de separação de fontes deverá penalizar a ativação simultânea das estimativas. A codificação esparsa (*Sparse Coding*), proposta em [6], é um exemplo de algoritmo de separação que utiliza essa característica da mistura. Outros algoritmos também utilizam a esparsidade como restrição adicional, como será mencionado adiante.

- Não-Negatividade: As fontes originais são não-negativas, logo não existe a possibilidade de cancelamento por adição; a NMF, que será vista mais adiante, é uma abordagem que utiliza essa característica para realizar a separação. Um exemplo de fonte não-negativa é a representação digital de uma imagem. Alternativamente, as fontes originais podem não ser não-negativas, porém uma representação não-negativa pode ser obtida para elas através de uma etapa de pré-processamento; por exemplo, para fontes sonoras, pode-se obter um espectrograma de magnitude do sinal original.

1.1 Contexto

1.1.1 Objetivos e aplicações da separação de fontes de áudio

Assim como as diferentes definições de fonte vistas acima, a separação de fontes de áudio pode ter diferentes objetivos e aplicações. Um objetivo natural seria a solução do problema da festa de coquetel, onde desejamos segregar uma das fontes, mas poderíamos também desejar segregar cada uma das fontes. Outro objetivo seria a detecção da presença de certas fontes em uma mistura.

Aplicações da separação de fontes de áudio implicam a escolha de um objetivo, de uma definição de fonte e de uma ou mais etapas de pós-processamento. Por exemplo, em uma aplicação de transcrição de músicas, gostaríamos de detectar

não só a presença de certos instrumentos, como também quais notas eles estão emitindo; uma etapa de pós-processamento, nesse caso, seria a transcrição propriamente dita. Para aplicações onde se deseja ouvir a fonte separada pode ser necessário um pós-processamento para a reconstrução do sinal a partir de um espectrograma de magnitudes, como ocorre com a utilização da NMF.

1.1.2 Separação cega vs. separação supervisionada

Algoritmos de separação de fontes são tipicamente divididos em algoritmos cegos, ou não-supervisionados, e algoritmos supervisionados. O trecho a seguir, extraído de [7], apresenta uma definição sucinta do que diferenciaria um algoritmo cego:

> "[...] *The scientific community used the word 'blind' for denoting all inversion methods based on output observations only.*[...]"

O texto ainda menciona que o termo não-supervisionado talvez seja o mais correto, porém não o mais difundido nessa área. Uma distinção análoga entre algoritmos supervisionados e não-supervisionados também existe em outras áreas de aprendizagem de máquina [8].

Genericamente, os algoritmos cegos são aqueles que visam a realizar a separação com o mínimo de informação possível sobre a mistura e as fontes originais, como por exemplo o número de fontes presentes na mistura, características dos sinais (como visto acima) e a definição de parâmetros (como tamanho da transformada de Fourier discreta – DFT) necessários para a execução com sucesso do algoritmo. Algoritmos supervisionados também utilizam essas informações, porém o que tipicamente os diferencia é o uso de exemplos das fontes originais separadas, embora essa distinção nem sempre seja clara.

1.1.3 Múltiplas misturas

Além dos aspectos mencionados anteriormente, existe ainda a possibilidade de se utilizar mais de uma mistura no processo de separação. O foco deste trabalho será exatamente estudar a utilização de múltiplas misturas em algoritmos de separação de fontes de áudio, particularmente aqueles baseados na fatoração de matrizes ou tensores[1] não-negativos. A ideia por trás da utilização de diversidade espacial neste tipo de algoritmo é de alguma forma melhorar a separação valendo-se de informações adicionais.

[1] Um tensor é simplesmente uma representação matemática de ordem N arbitrária. Um tensor (de ordem 3 ou mais) é análogo a uma matriz multidimensional; por exemplo, um tensor de ordem 3 pode ser pensado como uma matriz com 3 índices. Escalares, vetores e matrizes são casos particulares de tensores com ordens 0, 1 e 2, respectivamente.

As diferentes propostas para a utilização de múltiplas misturas podem ser classificadas em dois tipos: aquelas que se aproveitam das novas informações em etapas de pré e/ou pós-processamento, de certa forma estendendo algoritmos já existentes para o caso de múltiplas misturas; e aquelas onde as múltiplas misturas são intrínsecas ao modelo proposto, e acabam produzindo novos algoritmos (ou ao menos mudanças significativas em algoritmos já existentes).

O algoritmo proposto em [9] é um algoritmo (supervisionado) do primeiro tipo. Ele é composto, resumidamente, de uma etapa de pré-processamento (*directional clustering*) que visa a agrupar padrões espectrais (uma ou mais fontes sonoras, por exemplo) de acordo com a sua direção de chegada, seguida da execução do algoritmo da NMF sobre cada mistura pré-processada e uma etapa de pós-processamento para restaurar os padrões espectrais anteriores ao pré-processamento.

Os algoritmos propostos em [10] e [11] são algoritmos do segundo tipo. Em [10] propõe-se um modelo análogo ao utilizado na NMF, que considera que cada fonte está presente em todas as misturas. O algoritmo agrupa as misturas e realiza uma fatoração não-negativa sobre o tensor resultante (NTF – *Non-Negative Tensor Factorization*). Diferentemente do algoritmo em [9], este algoritmo (e o modelo correspondente) utiliza a diferença de ganhos existentes entre as misturas. A NTF, o modelo e o algoritmo completo de separação de fontes serão abordados de forma aprofundada no Capítulo 4. Em [11] a proposta é totalmente diferente; a ideia é não só modelar os padrões espectrais das fontes, e as diferentes intensidades com que ocorrem ao longo do tempo, mas também incluir no modelo o ambiente onde as fontes estão inseridas. Esse algoritmo será apresentado no Capítulo 3.

1.2 Objetivo

Este trabalho visa a estudar e implementar algoritmos de separação cega de fontes que utilizem representações não-negativas e múltiplas misturas, com foco também na utilização de diversidade espacial. O uso de múltiplas misturas e da diversidade espacial é de especial interesse por oferecerem novas informações que podem beneficiar um processo de separação, como também por permitir novas modelagens do processo de mistura em si.

1.3 Organização

No Capítulo 2 são apresentados algoritmos de separação e pré/pós processamento de misturas que serviram de base para o estudo deste trabalho; os Capítulos 3 e 4 apresentam os algoritmos de separação de múltiplas misturas e representam o núcleo dessa dissertação; no Capítulo 5 são apresentadas as propostas resultantes do estudo

apresentado nos capítulos anteriores; e finalmente, no Capítulo 6 são apresentadas as conclusões.

1.4 Símbolos

- $a \rightarrow$ representa uma variável escalar.
- $A \rightarrow$ representa um valor escalar pré-definido.
- $\mathbf{a} \rightarrow$ representa um vetor.
- $\mathbf{A} \rightarrow$ representa uma matriz.
- $\mathcal{A} \rightarrow$ representa um tensor.
- $\otimes \rightarrow$ representa o produto tensorial.
- $\odot \rightarrow$ representa o produto elemento a elemento entre vetores ou matrizes.
- $\frac{\mathbf{A}}{\mathbf{B}} \rightarrow$ representa uma divisão elemento a elemento entre vetores ou matrizes.

Capítulo 2

Alguns algoritmos preexistentes de separação e pré/pós-processamento de misturas

Para este trabalho, foi realizada uma revisão breve da literatura de separação de fontes, em particular o estudo da análise de componentes independentes (ICA), que por muitos anos foi o foco principal de estudos desta área, e da fatoração não-negativa de matrizes não-negativas (NMF) com algumas de suas variações, que além da separação em si podem fornecer representações mais abstratas de sinais musicais.

Figura 2.1: Etapas de um algoritmo genérico de separação de fontes.

A Figura 2.1 apresenta um diagrama de blocos de um algoritmo genérico de separação de fontes; como pode ser observado, além das técnicas de separação em si, estes algoritmos tipicamente envolvem também etapas de pré e pós-processamento. Este capítulo abordará tanto as técnicas de separação mencionadas acima quanto algumas etapas de pré/pós-processamento, particularmente aquelas aplicáveis a fontes sonoras, como a obtenção de espectrogramas de magnitude e técnicas para reconstrução da fase de sinais de áudio. Tais etapas são particularmente importantes no contexto de separação de áudio utilizando a NMF, já que esta opera sobre os espectrogramas de potência ou de magnitude das misturas, que uma vez decompostos nos espectrogramas das fontes, ainda requerem a informação de fase caso se deseje

reconstruir sinais de áudio.

2.1 Pré/Pós-processamento

2.1.1 Espectrograma de Magnitude

A utilização do espectro de frequências para análise e processamento de sinais de áudio já é algo quase ubíquo; essa representação, tipicamente utilizando a transformada discreta de Fourier (DFT — *Discrete Fourier Transform*), é o resultado da decomposição do sinal original em um determinado número de componentes frequenciais (às vezes conhecidos como *bins* ou raias de frequência) e permite a identificação de padrões que seriam indistinguíveis em sua representação original. A existência destes padrões (no caso mais simples, o fato de algumas fontes apresentarem apenas componentes em determinadas frequências), é o que motiva a utilização deste tipo de representação por algoritmos de separação de misturas.

O resultado da DFT pode ser representado como um vetor coluna onde cada elemento é um componente de frequência, ou seja, cada elemento representa a contribuição de uma frequência (na realidade, um intervalo de frequências) para a formação do sinal original:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_I \end{bmatrix} \in \mathbb{C}^I, \tag{2.1}$$

onde I é o número de componentes de frequência e i é o seu índice correspondente.

Para sinais com características espectrais que variam ao longo do tempo, como sinais de aúdio, a DFT não é realizada sobre o sinal completo, possibilitando a representação da evolução temporal das componentes de frequência presentes no sinal. De fato, a DFT é realizada sobre um determinado número de amostras, o que para sinais de áudio representa um determinado intervalo de tempo, tipicamente conhecido como um quadro. Em outras palavras, antes da aplicação da DFT, dá-se o processo de janelamento do sinal completo nesses quadros. O algoritmo completo é conhecido como transformada de Fourier de termo curto ou STFT (do inglês *Short-Time Fourier Transform*), o qual gera uma representação tempo-frequencial do sinal original que pode ser descrita em forma matricial por meio do seu espectrograma complexo:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \dots & x_{1L} \\ \vdots & \ddots & \vdots \\ x_{I1} & \dots & x_{IL} \end{bmatrix}, \tag{2.2}$$

onde L é o número de quadros, os quais serão indexados por l. Os elementos da

matriz $\mathbf{X}$ podem ser interpretados como componentes tempo-frequência, ideia que será bastante utilizada no Capítulo 3, já que cada coluna é o resultado da DFT realizada sobre um quadro e cada linha se refere a uma frequência.

A NMF e os vários outros algoritmos que se baseiam na não-negatividade não utilizam o espectro de frequências obtido diretamente pela DFT, utilizando em seu lugar o chamado espectrograma de magnitude, que pode ser definido como

$$|\mathbf{X}| = \begin{bmatrix} (x_{I1}x_{I1}^*)^{\frac{1}{2}} & \dots & (x_{IL}x_{IL}^*)^{\frac{1}{2}} \\ \vdots & \ddots & \vdots \\ (x_{11}x_{11}^*)^{\frac{1}{2}} & \dots & (x_{1L}x_{1L}^*)^{\frac{1}{2}} \end{bmatrix} \in \mathbb{R}_+^{I\times L}. \tag{2.3}$$

2.1.2 SRP-PHAT

O algoritmo *Steered Response Power* [12], algo como potência de resposta direcionada em inglês, é um algoritmo que visa a identificar a direção de chegada (DoA – *Direction of Arrival*) de sinais, quando detectados por um conjunto de sensores; o algoritmo efetivamente busca o DoA com máxima potência, calculando para isso um indicativo de potência referente a cada DoA. A premissa básica do algoritmo é que se um sinal for recebido em instantes diferentes pelos distintos sensores, então essa diferença deve ser causada pelo posicionamento de cada sensor no conjunto em relação à fonte. Se é possível determinar a diferença de tempo em que um mesmo sinal é detectado pelos diferentes sensores e se sabe a posição de cada sensor, então é possível determinar a direção de chegada daquele sinal em relação ao conjunto de sensores.

Para calcular a SRP, primeiramente calculamos a correlação cruzada de cada par do conjunto de sensores:

$$c_{n\eta}(\tau) = \int_{-\infty}^{+\infty} a_n(t)a_\eta(t+\tau)\mathrm{d}t, \tag{2.4}$$

onde a_n e a_η são sinais no domínio do tempo capturados pelo par de microfones n e η, respectivamente. A SRP será então obtida somando a correlação cruzada de todos os pares de sensores. Para o τ correspondente à diferença entre os atrasos de detecção do sinal entre dois sensores, a correlação teoricamente apresentará um pico; dessa forma, o cálculo da correlação serve para determinar qual o atraso observado por cada par de sensores.

Uma vez determinado o atraso observado por um par de sensores é possível utilizá-lo para estimar uma direção de chegada; isso pode ser realizado utilizando-se a seguinte relação:

$$\tau(\mathbf{k}_o) = (\mathbf{k}_o^T(\mathbf{n} - \boldsymbol{\eta}))/v, \tag{2.5}$$

onde o é um índice referente a cada direção que será avaliada, $\mathbf{k}_o$ é um vetor apontando para a direção o, v é a velocidade do som, e $\mathbf{n}$ e $\boldsymbol{\eta}$ são vetores contendo a posição dos microfones n e η, respectivamente.

A Figura 2.2 apresenta um exemplo da execução do algoritmo SRP com os sinais obtidos por um conjunto de 4 microfones no centro de uma sala contendo uma fonte a 135° e outra a 310° (mais detalhes na Subseção 5.3.1). Neste exemplo é possível estimar as direções das fontes, porém nenhuma está particularmente destacada e também são observados vários picos, possivelmente provocados por reverberações e ruídos.

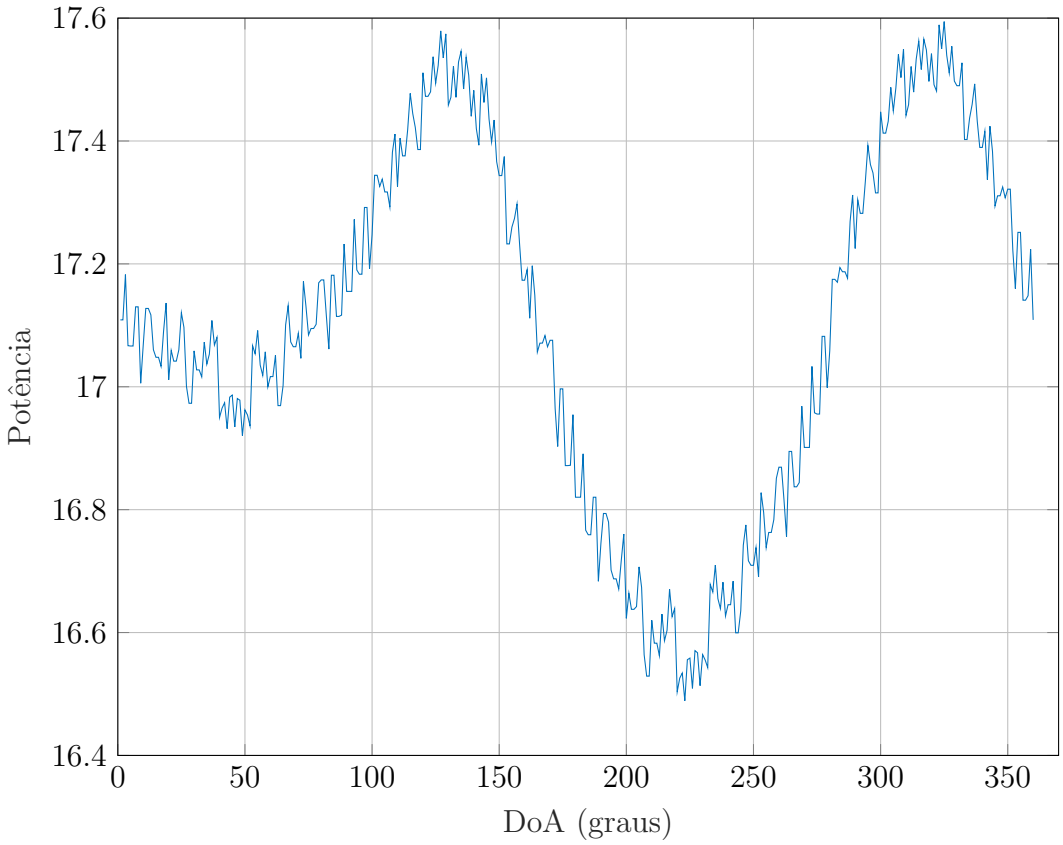

Figura 2.2: Gráfico da função objetivo do SRP, considerando duas fontes posicionadas a aproximadamente 135° e 310° do arranjo de microfones.

Para obter melhores resultados, em vez de calcular a correlação cruzada, calcula-se uma correlação cruzada generalizada (GCC — em inglês, *Generalized Cross Correlation*), com o objetivo de realçar o pico principal, referente ao atraso da DoA de cada par de sensores. Antes de descrever a GCC é interessante observar que a equação (2.4) pode ser vista como uma convolução, e portanto tem uma representação simples no domínio da frequência:

$$\mathcal{F}\{c_{n\eta}\} = S_n(\omega)S_\eta^*(\omega), \tag{2.6}$$

onde $\mathcal{F}$ representa o cálculo da transformada de Fourier. A equação (2.4) pode ser reescrita como a transformada de Fourier inversa da equação (2.6), levando assim a seguinte expressão:

$$c_{n\eta}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_n(\omega)S_\eta(\omega)^* \mathrm{e}^{\mathrm{j}\omega\tau}\mathrm{d}\omega. \tag{2.7}$$

A GCC nada mais é que uma espécie de correlação onde cada um dos sinais é ponderado por funções G. Utilizando a equação (2.7) encontramos

$$\mathrm{gcc}_{n\eta} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} (S_n(\omega)G_1(\omega))(S_\eta(\omega)G_2(\omega))^* \mathrm{e}^{\mathrm{j}\omega\tau}\mathrm{d}\omega, \tag{2.8}$$

o que pode ser simplificado para

$$\mathrm{gcc}_{n\eta} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \psi(\omega)S_n(\omega)S_\eta(\omega)^* \mathrm{e}^{\mathrm{j}\omega\tau}\mathrm{d}\omega, \tag{2.9}$$

onde

$$\psi(\omega) = G_1(\omega)G_2^*(\omega). \tag{2.10}$$

O SRP-PHAT [12], algo como resposta direcionada de potência com transformada de fase, é uma evolução do algoritmo SRP que pondera a correlação generalizada utilizando a função

$$\psi(\omega) = \frac{1}{|S_n(\omega)S_\eta^*(\omega)|}, \tag{2.11}$$

que ao eliminar a magnitude espectral transforma a representação de Fourier numa transformada de fase (PHAT, do inglês *phase transform*). Essa função acaba realizando o branqueamento do espectro de frequências; com isso, a correlação cruzada não será afetada pelas diferentes intensidades detectadas por cada sensor, efetivamente tornando o algoritmo imune a diferenças de amplitude entre os sinais. A Figura 2.3 apresenta o mesmo exemplo utilizado para gerar a Figura 2.2, porém agora utilizando o algoritmo SRP-PHAT; neste exemplo, as direções das fontes aparecem destacadas e podem ser observados significativamente menos picos que no exemplo anterior.

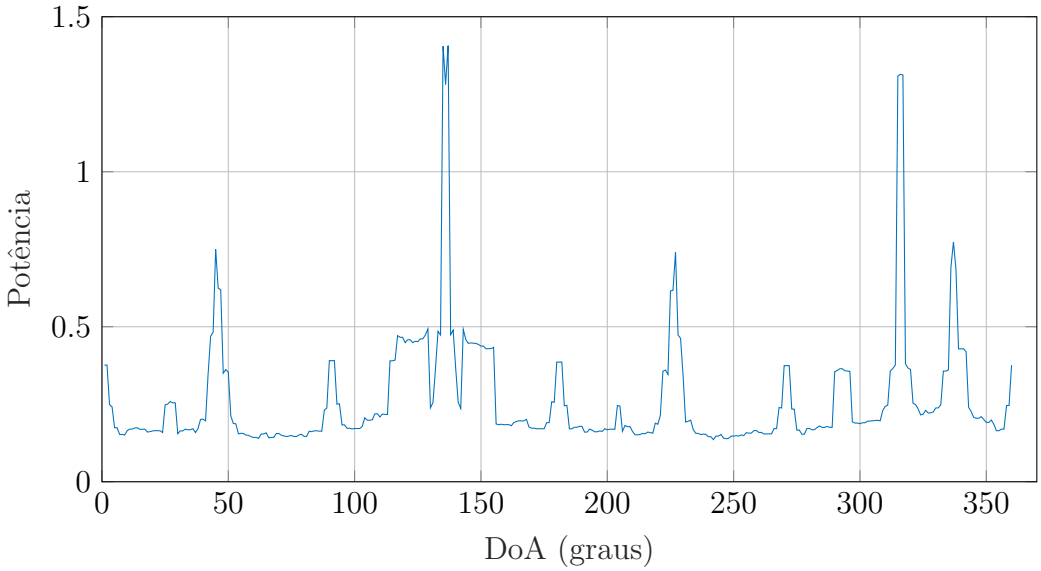

Figura 2.3: Exemplo SRP-PHAT.

Considerações Práticas

A direção de chegada é uma informação que pode ser útil para um algoritmo de separação de misturas: o processo de separação pode utilizar um algoritmo como o SRP-PHAT em uma etapa de pós-processamento para classificar seus resultados, ou na inicialização de um algoritmo que já utilize a informação de DoA, ou em ambos.

2.1.3 Reconstrução de Fase de Sinais de Áudio

Originalmente utilizados para aplicações de mudanças de escala de tempo e subtração espectral, algoritmos de reconstrução de fase são alvo de estudos em diversas áreas como microscopia eletrônica, cristalografia, astronomia e, dentre elas, a separação de misturas.

Os algoritmos apresentados a seguir foram alvo de um estudo em [13], onde eles foram implementados e seu desempenho foi comparado utilizando medidas objetivas. Todos estes algoritmos se baseiam na mesma ideia proposta em [14], no que ficou conhecido como o algoritmo Griffin & Lim, e que por isso será o primeiro a ser apresentado.

Griffin & Lim

Primeiro algoritmo de reconstrução proposto, o G & L consiste em uma ideia básica: minimizar uma função de distância entre a transformada de Fourier de dois sinais: $X(lT, \omega)$, correspondente ao sinal alvo (ou seja aquele que é a entrada do algoritmo e cuja fase se deseja reconstruir), e $\hat{X}(lT, \omega)$, correspondente a uma estimativa do sinal alvo. A equação 2.12 apresenta a função de distância proposta,

$$\mathrm{D}_{\hat{X},X} = \sum_{l=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} |\hat{X}(lT,\omega) - X(lT,\omega)|^2 \mathrm{d}\omega, \tag{2.12}$$

onde para cada quadro l (T é o número de amostras entre janelas consecutivas) calcula-se a integral do quadrado do módulo da diferença entre o sinal estimado $\hat{X}(lT,\omega)$ e o sinal alvo $X(lT,\omega)$.

Para minimizar a equação (2.12) calcula-se a sua função gradiente em relação ao sinal estimado, e a cada iteração encontra-se um ponto crítico, que será utilizado para atualizar a estimativa do sinal desejado.

Aplicando-se o teorema de Parseval à equação (2.12) encontra-se:

$$\mathrm{D}_{\hat{X},X} = \sum_{l=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |\hat{x}[lT,n] - x[lT,n]|^2. \tag{2.13}$$

Como na prática os sinais serão o resultado de uma STFT, $\hat{x}[lT,n]$ pode ser reescrito considerando-se uma função de janelamento:

$$\hat{x}[lT,n] = w[n - lT]x_{\mathrm{e}}[n], \tag{2.14}$$

onde $x_{\mathrm{e}}[n]$ é o sinal estimado propriamente dito.

Utilizando a equação (2.14) na equação (2.13) obtém-se:

$$\mathrm{D}_{\hat{X},X} = \sum_{l=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |w[n - lT]x_{\mathrm{e}}[n] - x[lT,n]|^2. \tag{2.15}$$

Pode-se ainda simplificar essa equação considerando que o objetivo real é minimizar a distância para cada amostra n_0 do sinal:

$$\mathrm{D}_{\hat{X},X,n_0} = \sum_{l=-\infty}^{\infty} |w[n_0 - lT]x_{\mathrm{e}n_0} - x[lT,n_0]|^2, \quad \forall n_0 \in \mathbb{Z}, \tag{2.16}$$

onde $x_{\mathrm{e}n_0}$ é o valor de $x_{\mathrm{e}}[n_0]$ para a amostra n_0. A minimização será realizada de forma independente para cada amostra; logo, para uma determinada amostra, o valor de n_0 será constante, mas o cálculo equivalente será realizado para cada amostra do sinal. Calculando o gradiente da equação (2.16) e igualando-o a zero, obtêm-se

$$\frac{\partial \mathrm{D}_{X,\hat{X},n_0}}{\partial x_{\mathrm{e}n_0}} = 2\left(\sum_{l=-\infty}^{\infty} w[n_0 - lT]^2 x_{\mathrm{e}n_0} - w[n_0 - mT]x[lT,n_0]\right) \tag{2.17}$$

e

$$x_{\mathrm{en}_0} = \frac{\sum_{l=-\infty}^{\infty} w[n_0 - lT]x[lT, n_0]}{\sum_{l=-\infty}^{\infty} w[n_0 - lT]^2}. \tag{2.18}$$

A cada iteração, o sinal estimado $\hat{X}(lT, \omega)$ é atualizado a partir da associação da magnitude do sinal original e da fase do sinal estimado na iteração anterior $\angle\hat{X}_{k-1}(lT, \omega)$:

$$\hat{X}_k(lT, \omega) = |X(lT, \omega)|\mathrm{e}^{\mathrm{j}\angle\hat{X}_{k-1}(lT,\omega)}. \tag{2.19}$$

O algoritmo é inicializado utilizando um sinal alvo com fase nula:

$$\hat{X}_0(lT, \omega) = |X(lT, \omega)|. \tag{2.20}$$

Ao final de K iterações, espera-se que o sinal estimado $x_{\mathrm{e}}[n]$ seja uma estimativa satisfatória da magnitude do sinal original $|X(lT, \omega)|$. No caso de separação de fontes, como já mencionado anteriormente, vários algoritmos têm como resultado espectrogramas de magnitude; estes serão utilizados como sinais alvo pelos algoritmos de reconstrução de fase.

RTISI — *Real-Time Iterative Spectrogram Inversion*

O RTISI [15], algo como inversão iterativa de espectrogramas em tempo real, é um algoritmo conceitualmente quase idêntico ao G & L; sua grande diferença, como o próprio nome já diz, é a ideia de trabalhar em tempo real, mais precisamente sem a necessidade do sinal completo. O RTISI reconstrói o sinal quadro a quadro utilizando apenas dados do atual quadro e dos anteriores, de forma que é teoricamente possível reconstruir o sinal à medida que ele tem seu espectrograma de magnitude estimado.

O RTISI-LA (*Look Ahead*) [16] é uma variação do RTISI que adicionalmente considera alguns quadros à frente. Essa modificação foi proposta para melhorar a qualidade do sinal reconstruído, já que tipicamente utilizam-se quadros com sobreposição — o que significa que parte da informação do quadro que é reconstruído a cada iteração está contida nos quadros seguintes (alguns tipos de sinais também apresentam naturalmente grande correlação temporal), característica que o RTISI ignorava na sua reconstrução.

MISI — *Multiple-Input Spectrogram Inversion*

O MISI [17], algo como inversão de espectrogramas utilizando múltiplas entradas, é um algoritmo baseado no G & L que foi pensando especificamente para a reconstrução de sinais provenientes de processos de separação de fontes; nele, o processo de atualização do G & L é modificado para receber um sinal de erro obtido a partir

da diferença entre a mistura original e uma mistura obtida a partir dos sinais reconstruídos. Efetivamente, no MISI ocorre a reconstrução em paralelo de múltiplas fontes e, entre iterações do G & L, as estimativas provisórias $x_e[n]$ de cada fonte são misturadas, de forma simplesmente aditiva, e comparadas com a mistura original. A diferença entre elas gera um sinal de erro $e_i[n]$ que corrige as estimativas de cada fonte, que por sua vez serão utilizadas na próxima iteração:

$$x_{\mathrm{e}}[n, s, i] = x_{\mathrm{e}}[n, s, i-1] + \frac{e_i[n]}{S},$$

onde S é o número de fontes, e n, i e s são os índices de amostra, iteração e fonte, respectivamente.

Comparação entre os algoritmos

Em [13] foram realizadas comparações entre os algoritmos acima utilizando como figura de mérito a razão sinal-erro (SER – *Signal-to-error Ratio*). Genericamente, o RTISI-LA apresentou os melhores resultados e o RTISI apresentou os piores. Para sinais provenientes de misturas, porém, o MISI se mostrou superior aos outros. O desempenho do MISI apresentou-se fortemente relacionado ao número de fontes presentes na mistura, o que é de certa forma natural: sua vantagem é utilizar a informação de fase presente nas misturas; à medida que o número de fontes aumenta, torna-se cada vez mais difícil detectar a fase referente a cada fonte, logo o impacto desse ganho de informação é limitado e em alguns testes o MISI apresentou desempenho inferior até ao RTISI.

2.1.4 BSS Eval

Apesar de não ser estritamente um pós-processamento, a ferramenta **BSS Eval** é comumente utilizada para a avaliação de algoritmos de separação cega. Apresentada em [18], essa ferramenta foi originalmente proposta para a avaliação de algoritmos de separação cega de fonte sonoras, e utiliza as seguintes métricas objetivas:

Razão fonte-distorção – *Source-to-Distortion Ratio*:

$$\mathrm{SDR} = 10 \log_{10} \frac{||s_{\mathrm{alvo}}||^2}{||e_{\mathrm{interf}} + e_{\mathrm{noise}} + e_{\mathrm{artif}}||^2}, \tag{2.21}$$

onde s_{alvo} é a fonte separada que se deseja obter, e_{interf} é a interferência causada por outras fontes, e_{noise} se refere ao ruído proveniente dos sensores e e_{artif} se refere ao impacto de artefatos causados pelo processo de separação.

Razão fonte-interferência – *Source-to-Interference Ratio*:

$$\mathrm{SIR} = 10\log_{10}\frac{||s_{\mathrm{alvo}}||^2}{||e_{\mathrm{interf}}||^2}. \tag{2.22}$$

Razão fontes-ruído – *Sources-to-noise Ratio*:

$$\mathrm{SNR} = 10\log_{10}\frac{||s_{\mathrm{alvo}} + e_{\mathrm{interf}}||^2}{||e_{\mathrm{noise}}||^2}, \tag{2.23}$$

Razão fontes-artefatos – *Sources-to-Artifacts Ratio*:

$$\mathrm{SAR} = 10\log_{10}\frac{||s_{\mathrm{alvo}} + e_{\mathrm{interf}} + e_{\mathrm{noise}}||^2}{||e_{\mathrm{artif}}||^2}. \tag{2.24}$$

Dadas as definições acima, é interessante apontar que a utilização desta ferramenta necessita da disponibilidade das fontes originais, já que os parâmetros e_{interf} e e_{artif} somente podem ser obtidos a partir delas e os parâmetros s_{alvo} são exatamente estas fontes.

2.2 ICA

No contexto de separação de fontes, a análise de componentes independentes, ICA, é uma técnica de separação cega (do termo em inglês *Blind Source Separation* – BSS), ou seja, não utiliza informações sobre as fontes originais separadas (a não ser o seu número); apesar disso, assumem-se certas características sobre as fontes e sobre o tipo de mistura. A ICA, como seu nome já diz, assume que as componentes a separar (nesse contexto, as fontes) sejam independentes; além disso, a separação de fontes utilizando ICA modela o processo de mistura como somas ponderadas dessas componentes. Cada observação é representada por uma dessas somas, e para se obter a separação de K componentes é necessário ter ao menos K observações distintas disponíveis. Um modelo de mistura pode ser descrito como

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{2.25}$$

onde $\mathbf{s}$ é o vetor de componentes, $\mathbf{A}$ é a matriz de mistura e $\mathbf{x}$ é o vetor de observações.

Com base no modelo acima, deve estar claro que o objetivo do processo de separação é encontrar a matriz $\mathbf{A}$ que gera a mistura (ou a matriz inversa que desfaz a mistura); para isso, a ICA, na sua forma clássica, assume uma última característica sobre as componentes: que elas não sejam gaussianas. Segundo o teorema central do limite, somas de variáveis aleatórias independentes tendem a distribuições gaussianas; no presente contexto, isso implica que a mistura deve ser

mais gaussiana (o que obviamente não teria sentido para componentes já gaussianas) do que cada componente individualmente. Por isso, o algoritmo básico da ICA busca maximizar (ou minimizar, dependendo da medida) medidas de não-gaussianidade (ou gaussianidade) como a curtose e a negentropia (ou aproximações delas, como no caso do algoritmo fastICA [19]).

Em [19] foram realizados testes utilizando a implementação fastICA; suas conclusões, assim como alguns testes rápidos, mostraram que o desempenho da separação utilizando ICA depende diretamente do tipo de mistura. Para misturas instantâneas[1], como é suposto no modelo, o desempenho da técnica é satisfatório e sua execução é rápida (principalmente comparando-a com a NMF); para misturas com ruído o desempenho também é satisfatório, porém está diretamente associado à razão sinal-ruído das componentes; e finalmente, para misturas com memória, conhecidas como convolutivas, o algoritmo falha na separação.

Vale ressaltar que existem diversas variações da técnica ICA que visam a resolver as deficiências do algoritmo básico, como a necessidade de que o número de observações seja no mínimo o mesmo número de fontes [20] e principalmente a capacidade de separar misturas com memória [4]. Atualmente, porém, outras técnicas como a NMF têm sido consideradas mais promissoras no contexto de separação de áudio.

2.3 NMF

A NMF é uma técnica de redução de dimensionalidade que, como seu nome já diz, permite apenas decomposições não-negativas. Para certas aplicações esta peculiaridade é potencialmente vantajosa, pois permite decomposições representativas do sinal original; por exemplo, para um espectrograma de magnitude, todas as decomposições resultantes da NMF poderão ser interpretadas como outros espectrogramas de magnitude. Para uma aplicação de separação de fontes, toda decomposição deve resultar nas fontes que compõem a mistura ou em padrões espectrais que compõem essas fontes.

2.3.1 NMF Básica

A ideia básica da NMF consiste em decompor uma matriz não-negativa $\mathbf{X} \in \mathbb{R}_+^{I \times L}$ em duas outras matrizes não-negativas $\mathbf{B} \in \mathbb{R}_+^{I \times Q}$ e $\mathbf{G} \in \mathbb{R}_+^{Q \times L}$, onde Q é o número de componentes em que a matriz é decomposta. A Figura 2.4 apresenta uma ilustração desta decomposição.

[1]Misturas instantâneas são aquelas onde as interações entre as partes, em um instante, independem dos seus estados em outros instantes; também podem ser conhecidas como misturas sem memória.

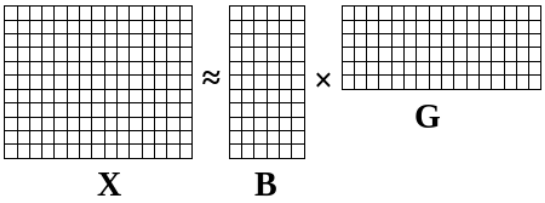

Figura 2.4: NMF – Decomposição.

O algoritmo da NMF procura as matrizes **B** e **G** que minimizam uma função custo $f : \mathbb{R}_+^{I\times Q} \times \mathbb{R}_+^{Q\times L} \to \mathbb{R}_+$, como por exemplo a norma de *Frobenius* da diferença entre a matriz original **X** e o produto das matrizes **B** e **G**:

$$f(\mathbf{B},\mathbf{G}) = \frac{1}{2}||\mathbf{X}-\mathbf{BG}||_F^2 = \frac{1}{2}\sum_{i=1}^{I}\sum_{l=1}^{L}|x_{ij}-\sum_{q=1}^{Q}b_{iq}g_{ql}|^2, \tag{2.26}$$

onde q é o índice das componentes-base em que a matriz será decomposta

O processo de minimização ocorre de forma iterativa, em que para uma determinada matriz **G** são calculados os novos valores da matriz **B**, com estes novos valores então são obtidos novos valores para a matriz **G**, e o processo se repete. A minimização é realizada desta forma porque o problema é convexo em **B** ou em **G**, mas não em ambos ([21]).

As decomposições resultantes possuem uma interpretação natural; por exemplo, as colunas da matriz **B** podem ser vistas como componentes-base que serão ponderadas pelas colunas da matriz **G**, que assim pode ser vista como uma matriz de ganhos. Em aplicações de áudio[2] as colunas de **B** podem ser vistas, de forma ainda mais representativa, como componentes espectrais que são ponderadas quadro a quadro pela matriz de ganhos **G**.

No contexto de separação de fontes, a NMF implica um modelo onde o sinal a ser fatorado é resultante de uma soma instantânea dos sinais das fontes originais, de forma similar ao modelo básico da ICA. Porém, diferentemente deste, o número de fontes que serão separadas independe do número de sensores. O número de fontes pode ser utilizado como o número de componentes-base, porém isso pode limitar a utilidade do algoritmo, que por exemplo, não seria capaz de representar fontes com padrões espectrais que variem ao longo do tempo.

Vários artigos, por exemplo [22] e [23], tratam de modificações do algoritmo básico da NMF como diferentes funções custo (por exemplo a divergência de Kullback-Liebler e a distância de Itakura-Saito), critérios de esparsidade e continuidade temporal e, como será visto a seguir, modificações que efetivamente alteram o modelo de combinação das fontes originais.

[2]Para aplicações de áudio a matriz **X** que será decomposta tipicamente representa um espectrograma de magnitude.

2.3.2 NMFD

A NMFD [5], algo como fatoração não-negativa e deconvolutiva de matrizes não-negativas, é uma modificação da NMF onde as componentes-base, ou as linhas da matriz de ganhos, deixam de ser vetores e passam a ser matrizes ($\mathbf{B}^t$ ou $\mathbf{G}^t$, onde t é a nova dimensão dessas matrizes) permitindo assim decomposições mais elaboradas. No contexto de separação de fontes, a definição de $\mathbf{B}^t$ permite representar melhor fontes com características espectrais que variem ao longo do tempo. Essa alteração é particularmente interessante para a separação de sinais musicais, já que as componentes-base agora podem melhor representar a variação do espectro das notas de um instrumento musical. A implementação dessa modificação envolve não só a mudança das componentes-base, que agora são efetivamente matrizes, mas também a introdução do operador de deslocamento horizontal $\overset{\rightarrow t}{(\cdot)}$ que será utilizado na matriz de ganhos $\mathbf{G}$ conforme o seguinte exemplo de uso:

$$\mathbf{G} = \overset{\rightarrow 0}{\mathbf{G}} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \quad \Longrightarrow \quad \overset{\rightarrow 1}{\mathbf{G}} = \begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}. \tag{2.27}$$

Com estas alterações o modelo da NMFD pode ser escrito como:

$$\mathbf{X} \approx \sum_{t=0}^{\rho-1} \mathbf{B}^t \overset{\rightarrow t}{\mathbf{G}}, \tag{2.28}$$

onde ρ é o número de deslocamentos (por exemplo, quadros) permitidos para cada componente.

NMF2D — *Non-Negative Matrix Factor 2-D Deconvolution*

A NMF2D [24], algo como fatoração não-negativa duplamente deconvolutiva de matrizes não-negativas, é a extensão natural da NMFD onde tanto as componentes-base quanto as linhas da matriz de ganho passam a ser matrizes. No caso de um espectrograma de magnitude, esta extensão permite a existência de padrões espectrais que variam ao longo de alguns quadros, e também que estes padrões possam ser deslocados na frequência; isso pode ser particularmente interessante para representar instrumentos musicais cujas "assinaturas" se desloquem na frequência ao longo dos quadros. A implementação desta extensão significa a introdução do operador deslocamento vertical para as matrizes de componentes-base de forma análoga ao que foi feito acima. O modelo estendido pode ser reescrito como:

$$\mathbf{X} \approx \sum_{t=0}^{\rho-1} \sum_{r=0}^{\phi-1} \overset{\downarrow r}{\mathbf{B}^t} \overset{\rightarrow t}{\mathbf{G}^r}, \tag{2.29}$$

onde ϕ é o número de possíveis deslocamentos na vertical (a NMFD descrita acima também poderia ser implementada utilizando o operador de deslocamento vertical em vez do horizontal; para fontes sonoras, isso significa que um mesmo instrumento poderia ser representado com menos — ou até mesmo apenas uma — componentes-base).

Para a utilização desta extensão com sinais musicais, é desejável que o espectrograma esteja em uma escala logarítmica de frequências, de modo que cada deslocamento unitário na vertical reflita uma mudança constante na escala de temperamento igual.

LNMF2D — *Linear Non-Negative Matrix Factor 2-D Deconvolution*

A LNMF2D [25], algo como fatoração linear, não-negativa e duplamente deconvolutiva de matrizes não-negativas, é uma alteração no modelo descrito acima que evita a necessidade da utilização de espectrogramas em escala logarítmica para sinais musicais, utilizando um operador de deslocamento vertical que realiza deslocamentos maiores ou menores de acordo com a frequência.

CNMF2D — *Constrained Non-Negative Matrix Factor 2-D Deconvolution*

A CNMF2D, algo como fatoração não-negativa e duplamente deconvolutiva de matrizes não-negativas com restrições, é uma generalização do algoritmo de NMF2D, proposta em [26], que permite a adição de novas restrições, como critérios de esparsidade temporal, esparsidade frequencial, continuidade temporal, autocorrelação das fontes e correlação cruzada entre as fontes.

Capítulo 3

Separação de misturas multicanais utilizando matrizes de covariância espacial e NMF

Neste capítulo será apresentado um algoritmo, proposto em [11], que combina conceitos de localização de fontes e matrizes de covariância espacial (*Spatial Covariance Matrix* — SCM) em uma nova modelagem para misturas de áudio, e um modelo de separação baseado na NMF que é compatível com essa modelagem.

A separação de áudio multicanal utilizando NMF e modelo de covariância espacial baseado em DoA (do inglês *multichannel audio separation by direction-of-arrival-based spatial covariance model and non-negative matrix factorization*), como o autor nomeou seu algoritmo que aqui será chamado apenas de NMF-SCM, apresenta como diferencial a assimilação, de forma intrínseca, de conceitos de localização de fontes e dos efeitos do ambiente em um processo de mistura. Outras propostas já utilizaram conceitos similares, porém tipicamente como etapas de pré-processamento (que este algoritmo também utiliza para inicialização) ou pós-processamento (como será proposto no Capítulo 5).

O capítulo está dividido em duas partes: a primeira apresentará o algoritmo em si, de início abordando as matrizes de covariância espacial, seguindo pelos modelos de mistura e separação, e terminando na apresentação do processo de extração das fontes após a execução do algoritmo; já a segunda parte consistirá principalmente de considerações práticas sobre a implementação do mesmo.

3.1 NMF-SCM

O problema prático que a NMF-SCM aborda é a separação de fontes capturadas por um arranjo de microfones, onde se sabe exatamente a posição de cada microfone no

arranjo e também o número de fontes. A utilização de um arranjo de microfones, o que obviamente implica a utilização de múltiplos canais pelo algoritmo, permite a determinação das diferentes DoAs das fontes, e a modelagem de reverberações e difrações causadas pelo ambiente.

Parte central do modelo de mistura que será apresentado a seguir são as matrizes de covariância espacial (SCM), que são matrizes de covariância de um único sinal que é percebido de forma diferente de acordo com a posição espacial dos sensores que o detectam (no caso, microfones). Para se obter as SCMs, a primeira etapa é agrupar as representações dos diferentes microfones em um vetor $\mathbf{x}_{il}$. Diferentemente da NMF, a representação utilizada será a STFT dos sinais e não diretamente o espectrograma de magnitude destes; o vetor $\mathbf{x}_{il}$ pode ser escrito como:

$$\mathbf{x}_{il} = \begin{bmatrix} x_{il1} \\ \vdots \\ x_{ilN} \end{bmatrix} \quad \in \mathbb{C}^N, \tag{3.1}$$

onde i é o índice referente ao componente de frequência, l é o índice referente ao quadro, N é o número de microfones e x_{iln} é um componente tempo-frequência do sinal de um dos microfones (n sendo o índice referente aos microfones). Uma vez obtidos os vetores $\mathbf{x}_{il}$, o cálculo das SCMs é o resultado do produto destes por seus conjugados transpostos (equivalente ao produto tensorial de cada vetor por ele mesmo)

$$\mathbf{X}_{il} = \mathbf{x}_{il}\mathbf{x}_{il}^H = \mathbf{x}_{il} \otimes \mathbf{x}_{il} \tag{3.2}$$

3.1.1 Modelo de mistura

Segundo o modelo proposto para representar a mistura, os vetores $\mathbf{x}_{il}$ serão aproximadamente (já que ruídos e possíveis não-linearidades não são modelados) iguais à soma dos produtos de componentes tempo-frequência correspondentes a cada fonte, $s_{ilp} \in \mathbb{C}$, e vetores $\mathbf{h}_{ip} \in \mathbb{C}^N$(que serão abordados logo adiante):

$$\mathbf{x}_{il} \approx \sum_{p=1}^{P} \mathbf{h}_{ip} s_{ilp}, \tag{3.3}$$

onde p é o índice correspondente a cada fonte e $\mathbf{h}_{ip}$ contém os coeficientes h_{ipn}, assim como $\mathbf{x}_{il}$ contém os coeficientes x_{iln}.

Aplicando a equação (3.3) na equação (3.2), obtém-se

$$\mathbf{X}_{il} \approx \sum_{p=1}^{P} \mathbf{H}_{ip} \hat{s}_{ilp}, \tag{3.4}$$

onde $\mathbf{X}_{il}$ são as SCMs referentes aos vetores $\mathbf{x}_{il}$, $\mathbf{H}_{ip}$ são as SCMs referentes aos vetores $\mathbf{h}_{ip}$ e $\hat{s}_{ilp}$ são as componentes tempo-frequência dos espectrogramas de magnitude de cada fonte. A expressão (3.4) será o modelo efetivamente utilizado pelo algoritmo.

Uma vez já definidos os vetores $\mathbf{x}_{il}$ e as matrizes $\mathbf{X}_{il}$, resta interpretar os coeficientes h_{ipn}. Considere o caso em que $P = 1$ e $N = 1$; para esses valores, os vetores $\mathbf{x}_{il}$ se tornam escalares

$$x_{il} = h_i \cdot s_{il}, \tag{3.5}$$

que seriam as componentes tempo-frequência da STFT do sinal recebido pelo único microfone. Pode-se observar que os coeficientes h_i não dependem do índice de quadro, logo não variam ao longo do tempo, porém dependem do índice de frequência. Esse tipo de dependência é o mesmo encontrado na representação de filtros no domínio da frequência. Pode-se, então, interpretar o conjunto de coeficientes h_i como um filtro que modela as alterações sofridas pelo sinal original s durante a sua propagação até o único microfone. Para o caso geral em que se tem P fontes e N microfones, os conjuntos de coeficientes h_{ipn} representam filtros correspondentes à propagação entre pares fonte-microfone.

O modelo de mistura apresentado acima representa na verdade um processo de filtragem que é único entre um par fonte-microfone, mas ainda não relaciona isso a efeitos físicos como reverberações e difrações. O autor desejou também relacionar as SCMs com DoA; isso será realizado através da modelagem das SCMs $\mathbf{H}_{ip}$ como a soma de núcleos de DoA ($\mathbf{W}_{io} \in \mathbb{C}^{N \times N}$) ponderados por coeficientes que relacionam direções e fontes ($z_{po} \in \mathbb{R}_+$):

$$\mathbf{H}_{ip} = \sum_{o=1}^{O} \mathbf{W}_{io} z_{po}, \tag{3.6}$$

onde o é o índice referente a cada direção (DoA) e O é o número de direções.

Os núcleos de DoA $\mathbf{W}_{io}$ também serão SCMs cujos elementos serão

$$[\mathbf{W}_{io}]_{n\eta} = \mathrm{e}^{\mathrm{j}2\pi f_i \tau_{n\eta}(\mathbf{k}_0)}, \tag{3.7}$$

onde n e η são índices referentes a microfones, $\mathbf{k}_0$ é um vetor correspondente a uma DoA, $\tau_{n\eta}(\mathbf{k}_0)$ vem da expressão (2.5), repetida a seguir por conveniência:

$$\tau_{n\eta}(\mathbf{k}_0) = (\mathbf{k}_0^T(\mathbf{n} - \boldsymbol{\eta}))/v,$$

e

$$f_i = (i-1)F_s/F_L, \tag{3.8}$$

onde F_s é a frequência de amostragem, F_L é o número de amostras da STFT, e portanto f_i são as frequências associadas a cada componente de frequência com o respectivo índice i.

Os coeficientes z_{po}, ao relacionar direções e fontes, permitem a modelagem de reverberações e difrações, já que agora uma mesma fonte pode estar associada a diferentes núcleos de DoA, e portanto diferentes percursos. A Figura 3.1 ilustra o papel dos coeficientes z_{po}: nessa ilustração, o círculo com a letra 'M' representa um conjunto de microfones, e os retângulos denotados por 'S1' e 'S2' representam fontes; as linhas representam caminhos de propagação correspondentes a diferentes DoAs e explicitam como os coeficientes z_{po} podem contribuir para a modelagem de reflexões (e também refrações) associando diferentes direções a uma mesma fonte.

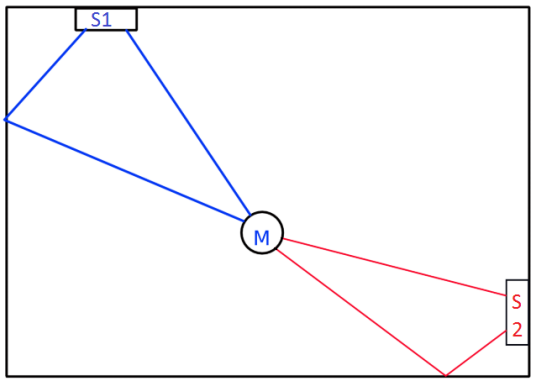

Figura 3.1: Ilustração do papel desempenhado pelos coeficientes z_{po}.

Cada associação fonte-direção pode também apresentar diferentes níveis de pertinência. Voltando ao exemplo da Figura 3.1, a visada direta entre cada fonte e o conjunto de sensores teria pertinência máxima, enquanto outras direções poderiam apresentar menores níveis de pertinência. A inicialização dos coeficientes z_{po} é realizada utilizando-se o algoritmo SRP-PHAT. Esse processo será apresentado mais à frente, porém vale destacar que após a inicialização, as direções consideradas mais significativas apresentarão pertinência máxima e todas as demais apresentarão pertinência mínima.

Juntando as expressões (3.4) e (3.6), obtemos o modelo completo:

$$\mathbf{X}_{il} \approx \sum_{p=1}^{P} \sum_{o=1}^{O} \mathbf{W}_{io} z_{po} \hat{s}_{ilp}. \tag{3.9}$$

Um detalhe relevante é que para permitir que as magnitudes das componentes tempo-frequência de cada microfone apareçam nas diagonais das SCMs $\mathbf{X}_{il}$, mantendo assim também uma relação direta com os algoritmos de NMF, o autor utiliza

uma versão alterada dos vetores x_{il} para obter as respectivas SCMs:

$$\hat{\mathbf{x}}_{il} = \begin{bmatrix} \sqrt{|x_{il1}|} \cdot \frac{x_{il1}}{|x_{il1}|} \\ \vdots \\ \sqrt{x_{ilM}} \cdot \frac{x_i lM}{|x_{ilM}|} \end{bmatrix}. \tag{3.10}$$

Para fins de comparação com outros algoritmos como a NMF e NTF, podemos explicitar as matrizes e tensores que seriam equivalentes às utilizadas por eles:

$$\begin{aligned} \mathcal{X} &\in \mathbb{C}^{I \times L \times N \times N}, \\ \mathcal{W} &\in \mathbb{C}^{I \times O \times N \times N}, \\ \mathbf{Z} &\in \mathbb{R}_{+,0}^{P \times O}, \\ \mathcal{S} &\in \mathbb{R}_{+,0}^{I \times L \times P}. \end{aligned} \tag{3.11}$$

3.1.2 Modelo e algoritmo de separação

Uma vez já definido o modelo de mistura, resta agora definir como será realizada a separação das fontes. Até a expressão (3.9) foi modelado apenas o ambiente em que ocorrem as misturas (além do processo de mistura propriamente dito), as fontes em si aparecendo em seu estado original, separado: $\hat{s}_{ilp}$. A separação das fontes equivalerá então à decomposição dos $\hat{s}_{ilp}$, e para isso o autor propõe um modelo de separação análogo ao utilizado pela NMF:

$$\hat{s}_{ilp} \approx \sum_{q=1}^{Q} m_{pq} b_{iq} g_{ql}, \quad m_{pq}, b_{iq}, g_{ql} \geq 0, \tag{3.12}$$

onde q é o índice referente a cada componente-base. Ignorando momentaneamente os escalares m_{pq} e o índice p, o modelo é muito similar ao utilizado pela NMF, sendo o escalar b_{iq} a participação da i-ésima frequência na q-ésima componente-base e o escalar g_{ql} a contribuição da q-ésima componente-base no l-ésimo quadro; essa similaridade fica mais clara se rescrevermos a equação (3.12) em uma forma matricial (ainda ignorando m_{pq} e o índice p):

$$\mathbf{S} \approx \mathbf{BG}. \tag{3.13}$$

O escalar m_{pq} atua como um coeficiente de pertinência da q-ésima componente-base em relação à p-ésima fonte. Dessa forma o algoritmo vai além da NMF básica, potencialmente permitindo que uma mesma componente-base componha fontes diferentes e, o que talvez seja ainda mais importante, incluindo no processo de separação uma relação entre fonte e componente-base que torna desnecessária uma etapa posterior de agrupamento das componentes-base. Outra interpretação equivalente da

atuação dos coeficientes m_{pq} é a de uma máscara que atua selecionando apenas os padrões correspondentes a cada fonte.

O modelo completo de mistura e separação é então obtido a partir das equações (3.9) e (3.12):

$$\mathbf{X}_{il} \approx \hat{\mathbf{X}}_{il} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{o=1}^{O}\mathbf{W}_{io}z_{po}m_{pq}b_{iq}g_{ql}. \tag{3.14}$$

Novamente, para fins de comparação, podemos explicitar as matrizes como feito anteriormente:

$$\begin{aligned}\mathbf{M} &\in \mathbb{R}_{+,0}^{P\times Q},\\ \mathbf{B} &\in \mathbb{R}_{+,0}^{I\times Q},\\ \mathbf{G} &\in \mathbb{R}_{+,0}^{Q\times L}.\end{aligned} \tag{3.15}$$

Para a implementação do algoritmo de separação é necessário definir também uma medida de erro e uma função custo. A medida de erro, assim como em outros algoritmos, é simplesmente a diferença entre o $\mathbf{X}_{il}$ original e o estimado:

$$\mathbf{E}_{il} = \mathbf{X}_{il} - \hat{\mathbf{X}}_{il} \quad = \quad \mathbf{X}_{il} - \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{o=1}^{O}\mathbf{W}_{io}z_{po}m_{pq}b_{iq}g_{ql}. \tag{3.16}$$

A função custo proposta é análoga à utilizada na NMF, também utilizando a norma de *Frobenius*:

$$f(\mathcal{W},\mathbf{Z},\mathbf{M},\mathbf{B},\mathbf{G}) = \sum_{i=1}^{I}\sum_{l=1}^{L}||\mathbf{X}_{il} - \hat{\mathbf{X}}_{il}||_F^2. \tag{3.17}$$

Como na NMF, a otimização de todos estes parâmetros pode ser descrita através de atualizações multiplicativas:

$$m_{pq}^{(\kappa+1)} \leftarrow m_{pq}^{(\kappa)}\left[1 + \frac{\sum_{i=1}^{I} b_{iq}\sum_{l=1}^{L} g_{ql}\sum_{o=1}^{O} z_{po}\,\mathrm{Tr}(\mathbf{E}_{il}\mathbf{W}_{io})}{\sum_{i=1}^{I} b_{iq}\sum_{l=1}^{L} g_{ql}\sum_{o=1}^{O} z_{po}\hat{x}_{il}}\right]^{(\kappa)}, \tag{3.18}$$

$$z_{po}^{(\kappa+1)} \leftarrow z_{po}^{(\kappa)}\left[1 + \frac{\sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{q=1}^{Q} m_{pq}b_{iq}g_{ql}\,\mathrm{Tr}(\mathbf{E}_{il}\mathbf{W}_{io})}{\sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{q=1}^{Q} m_{pq}b_{iq}g_{ql}\hat{x}_{il}}\right]^{(\kappa)}, \tag{3.19}$$

$$b_{iq}^{(\kappa+1)} \leftarrow b_{iq}^{(\kappa)}\left[1 + \frac{\sum_{l=1}^{L} g_{ql}\sum_{p=1}^{P} m_{pq}\sum_{o=1}^{O} z_{po}\,\mathrm{Tr}(\mathbf{E}_{il}\mathbf{W}_{io})}{\sum_{l=1}^{L} g_{ql}\sum_{p=1}^{P} m_{pq}\sum_{o=1}^{O} z_{po}\hat{x}_{il}}\right]^{(\kappa)}, \tag{3.20}$$

$$g_{ql}^{(\kappa+1)} \leftarrow g_{ql}^{(\kappa)}\left[1 + \frac{\sum_{i=1}^{I} b_{iq}\sum_{p=1}^{P} m_{pq}\sum_{o=1}^{O} z_{po}\,\mathrm{Tr}(\mathbf{E}_{il}\mathbf{W}_{io})}{\sum_{i=1}^{I} b_{iq}\sum_{p=1}^{P} m_{pq}\sum_{o=1}^{O} z_{po}\hat{x}_{il}}\right]^{(\kappa)}, \tag{3.21}$$

onde

$$\hat{x}_{il} = \sum_{q=1}^{Q} \sum_{o=1}^{O} z_{po} m_{pq} b_{iq} g_{ql}, \tag{3.22}$$

$(\kappa + 1)$ indica os valores para a nova iteração e (κ) indica os valores na iteração atual. Além disso, ainda é necessário atualizar $\mathbf{W}_{io}$, o que é realizado em 4 etapas. Inicialmente, faz-se:

$$\hat{\mathbf{W}}_{io} \leftarrow \mathbf{W}_{io} \left[\sum_{l=1}^{L} \sum_{p=1}^{P} \sum_{q=1}^{Q} (\hat{x}_{il} + \mathbf{E}_{il}) \right]; \tag{3.23}$$

depois, realiza-se a decomposição espectral da matriz $\hat{\mathbf{W}}_{io}$

$$\hat{\mathbf{W}}_{io} = \mathbf{V}\mathbf{D}\mathbf{V}^{H}, \tag{3.24}$$

onde $\mathbf{V}$ é uma matriz quadrada contendo os autovetores de $\mathbf{W}_{io}$ e $\mathbf{D}$ é uma matriz diagonal contendo os autovalores de $\mathbf{W}_{io}$. A matriz $\hat{W}_{io}$ é então reconstruída:

$$\hat{\mathbf{W}}_{io} \leftarrow \mathbf{V}\hat{\mathbf{D}}\mathbf{V}^{H}, \tag{3.25}$$

onde $\hat{\mathbf{D}}$ representa a matriz de autovalores $\mathbf{D}$ da equação (3.24) com todos os seus autovalores negativos substituídos por zero, de forma que $\hat{\mathbf{W}}_{io}$ seja positiva semidefinida; e finalmente, ocorre a atualização

$$\mathbf{W}_{io} \leftarrow |\hat{\mathbf{W}}_{io}| \mathrm{e}^{\mathrm{j}\angle \arg(\mathbf{W}_{io})}. \tag{3.26}$$

Adicionalmente, com o objetivo de evitar instabilidade numérica, o autor propõe utilizar parâmetros

$$\hat{a} = \left(\sum_{o=1}^{O} z_{po}^{2} \right)^{\frac{1}{2}} \quad \text{e} \quad \hat{c} = \left(\sum_{l=1}^{L} g_{ql}^{2} \right)^{\frac{1}{2}} \tag{3.27}$$

para realizar uma normalização dos parâmetros m_{pq}, z_{po}, b_{iq} e g_{ql}:

$$\begin{aligned} z_{po} &\leftarrow \frac{z_{po}}{\hat{a}} \\ m_{pq} &\leftarrow m_{pq}\hat{a} \\ g_{ql} &\leftarrow \frac{g_{ql}}{\hat{c}} \\ b_{iq} &\leftarrow b_{iq}\hat{c} \end{aligned} \tag{3.28}$$

Separação de fontes

Até a etapa anterior, o algoritmo descrito apenas decompõe as misturas originais nos diversos fatores definidos pelos modelos de mistura e separação. As estimativas das fontes serão efetivamente obtidas a partir dos vetores $\mathbf{x}_{il}$ originais utilizando-se um filtro de *Wiener*. Esse filtro será estimado usando-se os fatores obtidos pelo processo de separação, por exemplo: os coeficientes m_{pq} selecionam apenas as componentes-base que pertencem à fonte desejada, z_{po} delimitam a pertinência das contribuições de cada direção, b_{iq} representam padrões espectrais (de componentes-base) e g_{ql} são os ganhos dos respectivos padrões em cada quadro:

$$\mathbf{y}_{ilp} = \mathbf{x}_{il} \frac{\sum_{q=1}^{Q} \sum_{o=1}^{O} m_{pq} z_{po} b_{iq} g_{ql}}{\sum_{p=1}^{P} \sum_{q'=1}^{Q} \sum_{o'=1}^{O} m_{pq'} z_{po'} b_{iq'} g_{q'l}}, \tag{3.29}$$

onde $\mathbf{y}_{ilp} \in \mathbb{C}^N$ é um vetor contendo as estimativas de cada componente tempo-frequência para cada fonte p. Efetivamente, o resultado da separação é uma estimativa de cada fonte para cada microfone. Por fim, é interessante ressaltar que as estimativas obtidas não são estimativas de s_{ilp}, ou seja, das fontes originais, e sim do resultado da convolução das fontes originais com a resposta ao impulso do ambiente; para tentar esclarecer esse ponto, podemos reescrever a equação (3.3) incluindo $\mathbf{y}_{ilp}$:

$$\mathbf{x}_{il} \approx \sum_{p=1}^{P} \mathbf{h}_{ip} s_{ilp} = \sum_{p=1}^{P} \mathbf{y}_{ilp}. \tag{3.30}$$

3.2 Considerações práticas

O algoritmo proposto em [11] é interessante por apresentar, entre outras coisas, um modelo capaz de representar misturas convolutivas considerando DoA, refrações e difrações; além disso, ele não necessita de uma etapa de pós-processamento para determinar o agrupamento das componentes-base em relação às fontes, e a etapa de pós-processamento necessária para a separação das fontes é relativamente simples, já que não é necessário reconstruir a fase das fontes após a execução do algoritmo.

3.2.1 Implementação do algoritmo

Durante o desenvolvimento deste trabalho foram realizadas três principais implementações do algoritmo **NMF-SCM**, todas utilizando a linguagem R, com 3 focos distintos: validação, vetorização e paralelização. Nos itens a seguir serão abordadas estas implementações, suas mudanças, seus impactos no tempo de processamento necessário para execução do algoritmo e o processo de inicialização comum a todas elas. Antes disso, porém, é interessante apresentar a estrutura geral comum a todas

estas implementações:

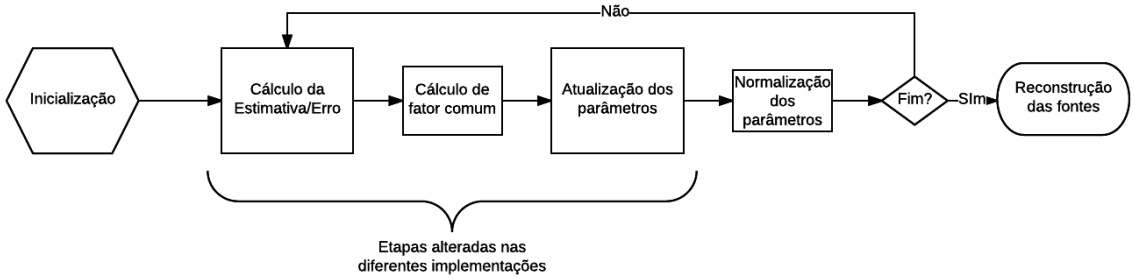

Figura 3.2: Estrutura geral de diferentes implementações do algoritmo NMF-SCM.

Como é possível observar na Figura 3.2, o algoritmo é composto de uma etapa de inicialização seguida da execução (durante várias iterações) de etapas responsáveis pelo cálculo das estimativas $\hat{\mathbf{X}}_{il}$, do erro $\mathbf{E}_{il}$ e da atualização dos parâmetros m_{pq}, z_{po}, b_{iq}, g_{ql} e $\mathbf{W}_{io}$; nestas etapas encontram-se as diferenças entre as três principais implementações. Finalizando, o algoritmo realiza uma etapa de normalização dos parâmetros atualizados (compartilhada por todas as implementações) e ao final de um determinado número de iterações ([11] utiliza 500) realiza-se uma etapa de reconstrução das fontes.

Inicialização

A etapa de inicialização do algoritmo envolve principalmente a obtenção das SCMs e a inicialização de todos os parâmetros em que elas serão decompostas. A obtenção das SCMs, e portanto do tensor $\mathcal{X}$, é realizada a partir dos espectrogramas de todas as misturas seguindo a expressão (3.2); porém, como já mencionado anteriormente, em vez de utilizar diretamente os elementos dos espectrogramas, as SCMs serão obtidas utilizando-se a alteração descrita na equação (3.10).

Os parâmetros m_{pq}, b_{iq}, g_{ql} são inicializados, de forma aleatória, com valores entre zero e um. Para a inicialização dos parâmetros z_{po} e $\mathbf{W}_{io}$ é necessário definir as DoAs que serão consideradas pelo algoritmo. Em [11] o autor menciona um conjunto de direções que amostram de maneira uniforme uma esfera cujo centro coincide com o do arranjo de microfones. Para este trabalho foi implementado um algoritmo que, dadas as resoluções de azimute e elevação a serem utilizadas, determina quais DoAs serão consideradas para obter uma amostragem aproximadamente uniforme. A Figura 3.3 ilustra um caso simples (à esquerda) e um caso próximo ao sugerido em [11] com 106 direções (à direita).

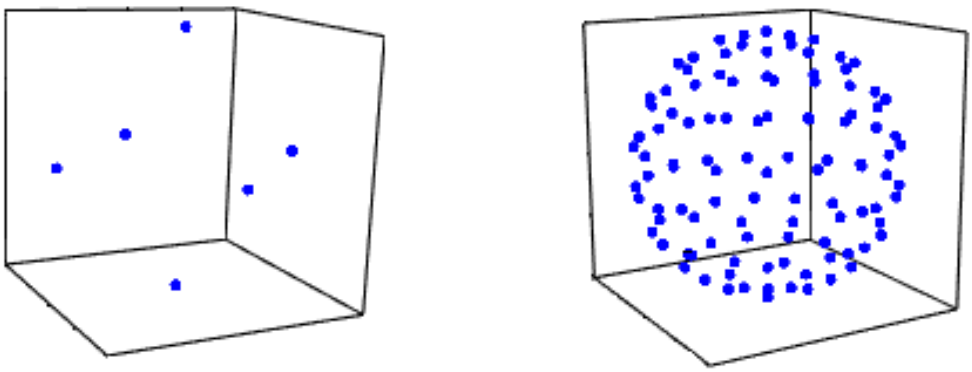

Figura 3.3: Exemplos de DoAs amostrados.

Umas vez obtidas as DoAs, os núcleos de DoA $\mathbf{W}_{io}$ são obtidos diretamente utilizando-se (3.7). Para a inicialização dos parâmetros z_{po}, executa-se o algoritmo SRP-PHAT; todo coeficiente z_{po} correspondente a uma direção a até 25° dos P (lembrando que P é o número de fontes) maiores picos encontrados recebe o valor 1 (correspondente à pertinência máxima) e todos os outros coeficientes recebem o valor 0 (correspondente à pertinência mínima).

Implementação de validação

Primeira implementação da NMF-SCM realizada durante o trabalho, seus objetivos originais foram a obtenção de uma codificação capaz de indicar o correto funcionamento do algoritmo. Para atingir estes objetivos, foi realizada de forma quase literal, de modo que as equações descritas em [11] pudesse ser identificadas, limitando assim o risco de fugir ao algoritmo proposto. Como pode ser visto nas equações (3.16), (3.18), (3.19), (3.20), (3.21), (3.22) e (3.23), a atualização de quase todos os parâmetros envolve três somatórios (3.22 envolve apenas dois somatórios), sendo que cada parâmetro apresenta dois índices; logo, para uma implementação direta, como desejado, são necessários cinco (quatro para a equação 3.22) laços de repetição, o que tende a ser ineficiente em linguagens como R e similares.

Antes de abordar o desempenho desta implementação, é interessante ressaltar a complexidade do algoritmo; por exemplo, utilizando os seguintes parâmetros, apresentados em [11]:

- sinais de 10 segundos de duração com frequência de amostragem 24 kHz
- STFT com 2048 amostras
- 50% de sobreposição de quadros

- Q (componentes-base) = 60
- O (DoA) = 110
- P (fontes) = 2
- M (microfones) = 4
- L (quadros) = 234 (valor aproximado calculado a partir dos outros valores)

o tensor $\mathcal{X}$ (com os elementos $\mathbf{X}_{il}$) terá dimensões $1024 \times 234 \times 4 \times 4$ (assim como $\mathcal{W}$ e $\hat{\mathcal{X}}$, para os elementos $\mathbf{W}_{io}$ e $\hat{\mathbf{X}}_{il}$, respectivamente), a matriz $\mathbf{M}$ (com os elementos m_{pq}) terá dimensões 2×60, a matriz $\mathbf{Z}$ (com os elementos z_{po}) terá dimensões 2×110, a matriz $\mathbf{B}$ (com os elementos b_{iq}) terá dimensões 1024×60 e a matriz $\mathbf{G}$ (com os elementos b_{ql}) terá dimensões 60×234. Esses valores, com a exceção dos tensores (cada um ocupa quase 30 MiB[1] — considerando representações de 8 bytes), são comparáveis aos tipicamente utilizados por outros algoritmos, como por exemplo, a NMF.

Para explicitar ainda mais a complexidade do algoritmo, considere a atualização dos fatores m_{pq} representada na equação (3.18): somente para calcular o termo no numerador é necessário calcular o traço de 26357760 ($= 110 \times 234 \times 1024$) matrizes 4×4, além do mesmo número de multiplicações e somas (o número de operações para o termo equivalente de cada fator pode ser visto abaixo).

- m_{pq} : 26357760 ($110 \times 234 \times 1024$)
- z_{po} : 14376960 ($60 \times 234 \times 1024$)
- b_{iq} : 51480 ($110 \times 2 \times 234$)
- g_{ql} : 225280 ($110 \times 2 \times 1024$)

Mesmo tratando-se de uma implementação direta, algumas medidas podem ser adotadas para que a execução do algoritmo seja mais eficiente; o traço de $\mathbf{E}_{il}\mathbf{W}_{io}$ só precisa ser calculado uma vez para todos i, l e o (as mesmas 26357760 vezes mencionadas anteriormente), assim como $\hat{x}_{il}$ também só precisa ser calculado uma vez para todos i e l. Porém, ainda é necessário calcular $\mathbf{E}_{il}$ (equação (3.16)), nada menos que 3162931200 operações, e a atualização de $\mathbf{W}_{io}$ (equação (3.24)) implica a decomposição em autovalores e autovetores (decomposição espectral) de 112640

[1]MiB ou mebibyte é equivalente a 2^{20} bytes. Kibi, mebi, gibi e outros, são sufixos criados pela *International Eletrotechnical Commision* e depois adotados por outras organizações como o IEEE (norma IEEE 1541-2002). Esses sufixos se referem a potências de dois com valores próximos aos dos sufixos mais comumente utilizados: kibi (KiB) = 2^{10}, mebi (MiB) = 2^{20}, gibi (GiB) = 2^{30} e etc

matrizes 4×4. A primeira implementação do algoritmo resultou em um tempo de execução de aproximadamente 40 minutos para cada iteração em uma máquina com as seguintes especificações: Core i7 4790, 16 GiB de memória RAM, utilizando o sistema operacional Fedora. Após as mudanças mencionadas acima, o tempo de execução diminuiu para cerca de 34 minutos.

Com o elevado custo computacional, a depuração de possíveis problemas nessa implementação se mostrou impraticável, o que motivou as implementações a seguir.

Implementação vetorizada

Para diminuir o tempo de execução observado na implementação anterior, a primeira abordagem adotada foi a da busca de uma implementação vetorizada do algoritmo.

Uma implementação vetorizada, ou em formato matricial, se refere à substituição de operações elemento-a-elemento (como os laços de repetição descritos acima) por operações vetoriais ou matriciais que muitas vezes são implementadas de forma mais eficiente através de otimizações de *software* e *hardware*.

Utilizando como base a implementação de validação, foi possível identificar que a atualização dos parâmetros b_{iq}, g_{ql}, $\mathbf{W}_{io}$ e $\mathbf{E}_{il}$ representavam (aproximadamente) 21%, 35%, 17% e 18%, respectivamente, do tempo de execução. Da mesma maneira, também foi possível identificar que o cálculo de $\hat{x}_{il}$ representava menos que 1% do tempo de execução, e que atualização de m_{pq} e z_{po} juntas representavam menos de 3%. Com base nessas informações, foram desenvolvidas implementações vetorizadas apenas para os parâmetros b_{iq}, g_{ql}, $\mathbf{W}_{io}$ e $\mathbf{E}_{il}$; durante esse processo, a implementação de validação foi utilizada para verificar que as novas implementações continuavam consistentes com o algoritmo proposto.

A obtenção de implementações vetorizadas não é um problema trivial: algumas das etapas apresentam dimensões compatíveis e logo podem ser diretamente descritas em formato matricial; porém, outras necessitam de algum processamento como o uso de matrizes auxiliares que contenham cópias de linhas/colunas de parâmetros para obter outras matrizes com dimensões compatíveis (assim como um processamento posterior para chegar aos valores corretos), e assim permitir eliminar laços de repetição. Não foi possível encontrar uma implementação que eliminasse todos os laços de repetição; porém, mesmo assim foi possível reduzir o tempo de execução de uma iteração em aproximadamente 70–80% (para 7–10 minutos).

Implementação paralelizada

Para diminuir ainda mais o tempo de execução de cada iteração do algoritmo, também foi desenvolvida uma implementação paralelizada, já baseada na implementação vetorizada. A Figura 3.4 apresenta como fica a estrutura do algoritmo

paralelizado: nela, apenas os parâmetros m_{pq}, z_{po}, b_{iq}, g_{ql} e $\mathbf{W}_{io}$ serão efetivamente calculados em paralelo, $\mathbf{E}_{il}$ e $\hat{x}_{il}$ continuarão sendo calculados independentemente.

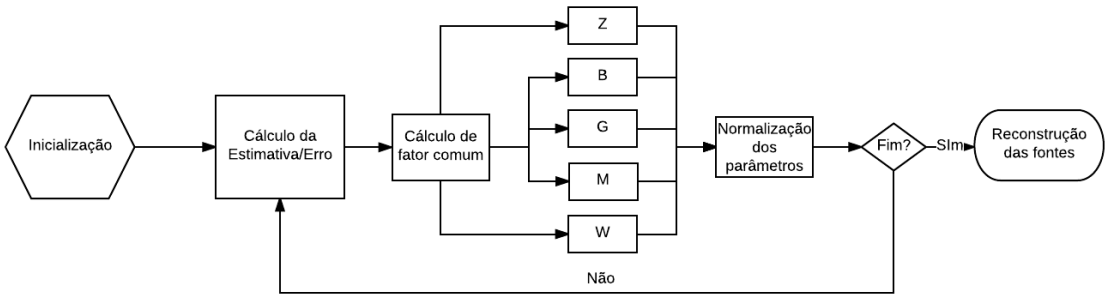

Figura 3.4: Estrutura paralelizada do algoritmo NMF-SCM.

Como resultado da implementação paralelizada, cada iteração do algoritmo apresenta um tempo de execução de cerca de 5 minutos.

3.2.2 Resultados

Em [11], o autor apresenta o resultado de testes com duas e três fontes, comparando o desempenho da NMF-SCM com três outros algoritmos: '*NMF clustering*' (algoritmo similar proposto pelo mesmo autor em [27]), '*NMF unconstrained*' (nome dado pelo autor para proposta realizada em [28]) e a ICA, especificamente a especificação proposta em [29]. Nestes testes os valores das métricas SDR e SAR da NMF-SCM foram maiores que para os outros algoritmos, e o valor da métrica SIR ficou abaixo apenas da '*NMF clustering*'.

Os testes realizados com todas as implementações desenvolvidas para este trabalho não foram bem sucedidos; todas se apresentaram numericamente instáveis e/ou muito dependentes do estado inicial: em algumas execuções alguns fatores convergiam rapidamente para zero e em outras, após dezenas de iterações, o algoritmo divergia.

Os trechos a seguir, extraídos de [30], ambos tratando deste tipo de algoritmo, o segundo especificamente do algoritmo aqui descrito, são particularmente correlacionados com as dificuldades observadas:

> "[...] *Although the technique performs well under any type of mixing conditions, the convergence of the cost function is unstable and much slower than that of conventional NMF techniques.*[...]"

> "[...]*It should be noted that the studies reported so far assume at most 3 or 4 microphones and are not suitable for a large number of microphones because the computational cost for SC-NMF is of order* $O(M^3)$ *per time-frequency (TF) bin* [...]."

O trecho a seguir, também extraído de [30], traz alguns comentários interessantes sobre a NTF que será apresentada no próximo capítulo:

> "[...] *Non-negative tensor factorization (NTF) is another promising technique for a multichannel scenario. The cost is only of order O(M) [10, 11] per TF bin. The drawback is that, in contrast to SC-NMF, it cannot model interchannel phase differences; that is, only intensity level differences between microphones are taken into account as spatial properties of the observed mixture. However, thanks to its low computational cost, NTF is being rigorously investigated for many types of applications*[...]"

Diante desses problemas, decidiu-se não prosseguir com o algoritmo neste trabalho.

Capítulo 4

Fatoração não-negativa de tensores

A fatoração não-negativa de tensores (NTF – *Non-Negative Tensor Factorization*), cujo primeiro algoritmo foi proposto em [31], é uma técnica similar à NMF, cujo objetivo é representar um tensor não-negativo utilizando outros tensores não-negativos de menor ordem ou dimensão.

Para melhor entender os algoritmos de NTF propostos por [31] e [10], é interessante explicitar duas propriedades de tensores:

1. Um tensor $\mathcal{X}$ de posto 1 e ordem N pode ser representado pelo produto tensorial, denotado aqui por $\otimes$, de N vetores:

$$\mathcal{X} = \mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \cdots \otimes \mathbf{u}_N. \tag{4.1}$$

2. Um tensor $\mathcal{Y}$ de posto Q e ordem N pode ser representado como a combinação linear de Q tensores de posto 1 e ordem N:

$$\mathcal{Y} = \sum_{q=1}^{Q} \mathcal{X}^q. \tag{4.2}$$

Aplicando a equação (4.1) na equação (4.2), pode-se reescrever o tensor $\mathcal{Y}$:

$$\mathcal{Y} = \sum_{q=1}^{Q} \mathbf{u}_1^q \otimes \mathbf{u}_2^q \otimes \cdots \otimes \mathbf{u}_N^q \tag{4.3}$$

A expressão (4.3) é particularmente interessante, pois já permite vislumbrar os Q fatores que resultarão da NTF e as estruturas (os N vetores) que os compõem. Fazendo uma analogia com a NMF, os vetores $\mathbf{u}_n^q$ seriam equivalentes às matrizes $\mathbf{B}$ e $\mathbf{G}$, porém representando uma única componente-base q. Em [31], que utilizou exatamente a expressão (4.3), o problema de fatoração é descrito como um problema de mínimos quadrados da diferença entre o tensor original $\mathcal{X}$ e uma estimativa obtida

a partir dos fatores (não-negativos):

$$\min_{\mathbf{u}_n^q} \ \frac{1}{2}||\mathcal{X} - \sum_{q=1}^{Q} \mathbf{u}_1^q \otimes \mathbf{u}_2^q \otimes \cdots \otimes \mathbf{u}_N^q||_F^2 \quad , \mathbf{u}_n^q \geq 0 \tag{4.4}$$

O algoritmo propriamente (Algoritmo 4.1) dito consiste na inicialização aleatória (com valores não-negativos) de cada vetor $\mathbf{u}_n^q$ e a subsequente atualização de cada elemento de forma iterativa, mantendo todos os outros valores constantes. O uso de um algoritmo de atualizações multiplicativas garante (já que todos os fatores são não-negativos no inicio da execução) a não-negatividade dos fatores obtidos pelo algoritmo.

Algoritmo 4.1 NTF

1: **function** NTF($\mathcal{X}, Q, k$)
   Entradas:
   $\mathcal{X} \rightarrow$ Tensor não-negativo.
   $Q \rightarrow$ Número de fatores (posto do tensor $\mathcal{X}$)
   $k \rightarrow$ Número de iterações
2: Inicializar os vetores $\mathbf{u}_n^q$ com valores aleatórios não-negativos
3: **for** 1,2,3, ...,k **do**
4: Atualizar elementos de cada vetor.
5: **end for**
6: **end function**

Em [10], a técnica foi estendida para utilizar a divergência de Kullback-Leibler generalizada [32] para tensores, utilizando também um tensor de ordem 3 já com foco na separação de áudio:

$$\mathrm{D_{KL}}(\mathcal{X}||\hat{\mathcal{X}}) = \sum_{i=1}^{I}\sum_{l=1}^{L}\sum_{c=1}^{C}\left(\mathcal{X}_{ilc}\log\frac{\mathcal{X}_{ilc}}{\hat{\mathcal{X}}_{ilc}} - \mathcal{X}_{ilc} + \hat{\mathcal{X}}_{ilc}\right), \quad \mathcal{X} \in \mathbb{R}_+^{I\times L\times C}, \tag{4.5}$$

onde I, L e C são respectivamente os números de faixas de frequência, quadros e canais. Os autores também apresentam uma decomposição equivalente à expressão (4.2), porém agrupando os Q vetores em N matrizes. Para seu modelo os autores nomearam as matrizes como $\mathbf{G}$, $\mathbf{A}$ e $\mathbf{S}$; neste trabalho, porém, estas letras foram substituídas por $\mathbf{H}$, $\mathbf{B}$ e $\mathbf{G}$ (vide Figura 4.1), respectivamente, para permitir uma analogia com as matrizes da NMF e suas interpretações.

$$\mathcal{X} \approx \hat{\mathcal{X}} = \sum_{q=1}^{Q} \mathbf{H}_{:q} \otimes \mathbf{B}_{:q} \otimes \mathbf{G}_{:q} \tag{4.6}$$

$$\mathbf{H} \in \mathbb{R}_{+,0}^{C\times Q}, \quad \mathbf{B} \in \mathbb{R}_{+,0}^{I\times Q}, \quad \mathbf{G} \in \mathbb{R}_{+,0}^{Q\times L}, \tag{4.7}$$

onde $\mathbf{A}_{:q}$ indica a seleção da coluna/linha referente à dimensão Q.

O algoritmo propriamente dito (Algoritmo 4.2) é praticamente idêntico ao anterior (com as devidas alterações referentes à diferente função custo); porém, com o modelo proposto as atualizações ocorrem em conjunto para cada matriz.

Algoritmo 4.2 NTF-KL

1: **function** NTF-KL($\mathcal{X}, Q, k$)
   Entradas:
   $\mathcal{X} \rightarrow$ Tensor não-negativo.
   $Q \rightarrow$ Número de fatores (posto do tensor $\mathcal{X}$)
   $k \rightarrow$ Número de iterações
2: Inicialize as matrizes $\mathbf{H}$, $\mathbf{B}$ e $\mathbf{G}$ com valores aleatórios não-negativos.
3: **for** 1,2,3, ...,k **do**
4: Atualizar a matriz $\mathbf{H}$
5: Atualizar o tensor $\hat{\mathcal{X}}$
6: Atualizar a matriz $\mathbf{B}$
7: Atualizar o tensor $\hat{\mathcal{X}}$
8: Atualizar a matriz $\mathbf{G}$
9: Atualizar o tensor $\hat{\mathcal{X}}$
10: **end for**
11: **end function**

As expressões das atualizações das matrizes $\mathbf{H}$, $\mathbf{B}$ e $\mathbf{G}$ são apresentadas abaixo em um formato matricial por simplicidade. Nessa representação, para ajustar as dimensões, é definida uma função $\mathcal{D}\{\mathbf{u}\}$ que, a partir de um vetor $\mathbf{u}$, constrói uma matriz quadrada cujos valores fora da diagonal são todos iguais a zero, e cuja diagonal contém os valores presentes no vetor $\mathbf{u}$. Além disso é preciso definir a matriz $\mathbf{1}_{I\times L}$, também para realizar um ajuste das dimensões.

$$\mathbf{B} = \mathbf{B} \odot \frac{\sum_{c=1}^{C} \frac{\mathbf{X}_c}{\hat{\mathbf{X}}_c} \mathbf{G}^T \mathcal{D}\{\mathbf{h}_c\}}{\sum_{c=1}^{C} \mathbf{1}_{IL} \mathbf{G}^T \mathcal{D}\{\mathbf{h}_c\}}; \tag{4.8}$$

$$\mathbf{G} = \mathbf{G} \odot \frac{\sum_{c=1}^{C} \mathcal{D}\{\mathbf{h}_c\} \mathbf{B}^T \frac{\mathbf{X}_c}{\hat{\mathbf{X}}_c}}{\sum_{c=1}^{C} \mathcal{D}\{\mathbf{h}_c\} \mathbf{B}^T \mathbf{1}_{IL}}; \tag{4.9}$$

$$\mathcal{D}\{\mathbf{h}_c\} = \mathcal{D}\{\mathbf{h}_c\} \odot \frac{\mathbf{B}^T \frac{\mathbf{X}_c}{\hat{\mathbf{X}}_c} \mathbf{G}^T}{\mathbf{B}^T \mathbf{1}_{IL} \mathbf{G}^T}, \tag{4.10}$$

onde $\odot$ representa multiplicação elemento a elemento, $\frac{\mathbf{A}}{\mathbf{B}}$ significa divisão elemento a elemento e $\mathbf{X}_c$, $\hat{\mathbf{X}}_\mathbf{c}$ são *slices*[1] dos tensores $\mathcal{X}$ e $\hat{\mathcal{X}}$, respectivamente.

Usando essa forma matricial a atualização da estimativa do tensor pode ser realizada *slice* por *slice*, como mostrado no Algoritmo 4.3 abaixo.

[1] *Slice* nesse contexto se refere ao colapsamento de uma das dimensões de um tensor. Esse nome é mais intuitivo quando utilizado em referência a um tensor de 3 dimensões, onde o colapso implica a obtenção de um tensor de duas dimensões ($2 \times 3 \times 4 \rightarrow 2 \times 3$, 3×4 ou 2×4), efetivamente uma fatia do tensor original.

Algoritmo 4.3 Atualização da estimativa do tensor

1: **for** 1,2,3, ...,C **do**
2: $\quad \hat{\mathcal{X}}_{\{:,:,c\}} = \mathbf{B}\mathcal{D}\{\mathbf{h}_c\}\mathbf{G}$
3: **end for**

A Figura 4.1 apresenta uma ilustração da fatoração na forma matricial descrita acima.

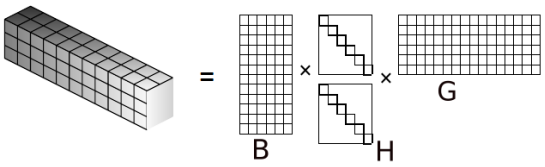

Figura 4.1: Ilustração de fatoração utilizando a NTF.

4.1 Separação de áudio utilizando NTF

A NTF, assim como a NMF e suas extensões, é uma técnica genérica de redução de dimensionalidade; para seu uso em algum problema específico é necessário um modelo que estabeleça uma relação entre a técnica e o problema.

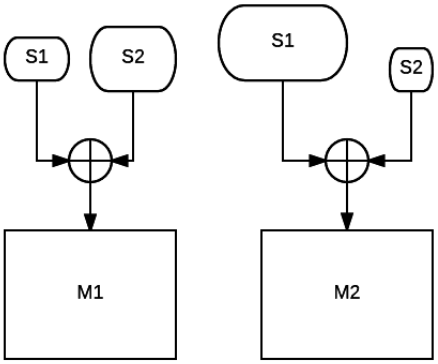

Figura 4.2: Modelo de misturas utilizado pelo algoritmo NTF.

A Figura 4.2 apresenta uma ilustração de um modelo (descrito em [33]) de dois canais que foi baseado na observação de que sons são tipicamente gravados de forma independente e misturados com diferentes ganhos em diferentes canais:

$$c^{\mathrm{e}}(t) = \sum_{p=1}^{P} g_p^{\mathrm{e}} s_p(t), \quad c^{\mathrm{d}}(t) = \sum_{p=1}^{P} g_p^{\mathrm{d}} s_p(t),$$

onde s_p se refere à fonte p, g_p^{e} é o ganho utilizado para a fonte p no canal esquerdo e g_p^{d} é o ganho equivalente para o canal direito (P, novamente, é o número de fontes).

Para a separação de fontes a NMF atua sobre espectrogramas de magnitude de misturas; no caso da NTF é necessário obter um tensor não-negativo. Logo, para

o modelo descrito acima, obtêm-se os espectrogramas de magnitude de ambos os canais, utilizando cada um deles como um *slice* do tensor desejado.

Analogamente à NMF, ao final da execução do algoritmo da NTF, obtêm-se as matrizes não-negativas **H**, **B** e **G** (para o caso genérico seriam N matrizes). Para esta aplicação, pode-se interpretar as informações que cada matriz contém sobre cada componente-base (fatores) da seguinte maneira:

- **H** $\rightarrow$ Intensidades das componentes-base (colunas) para cada canal (linhas).
- **B** $\rightarrow$ Padrões espectrais das componentes-base (colunas), ou seja, com que magnitude cada frequência (linhas) aparece em cada componente-base.
- **G** $\rightarrow$ Ganhos das componentes-base (linhas) em cada quadro (colunas).

Sinais de áudio tipicamente apresentam padrões espectrais complexos que não são bem representados por uma única componente-base, as fontes originais de uma mistura estarão na verdade distribuídas em um conjunto de componentes-base que deverão ser agrupadas.

4.1.1 Algoritmo de separação de fontes

O algoritmo completo de separação de fontes proposto em [10] consiste em:

1. Obter o espectrograma de cada canal.
2. Obter a magnitude do espectrograma de cada canal.
3. Obter um tensor a partir de todas as magnitudes e executar a NTF.
4. Determinar uma relação de intensidade intercanal das componentes-base.
5. Agrupar as componentes-base utilizando a relação do item 4.
6. Criar o espectrograma de magnitude de cada fonte utilizando os grupos do item 5.
7. Incorporar ao espectrograma de magnitude de cada fonte a informação de fase do canal onde ela é dominante.
8. Realizar a transformada inversa de Fourier sobre cada espectrograma.

No algoritmo acima, os passos 1, 2, 3, 6, 7 e 8 são autoexplicativos e não utilizam nenhuma informação específica sobre o modelo ou as fontes originais; estes aparecem nos passos 4 e 5, que serão abordados a seguir.

Relação de intensidade intercanal das componentes-base – Passo 4

Um dos maiores desafios na separação de áudio utilizando técnicas de fatoração não-negativas, como a NMF e a NTF, é que tipicamente melhores resultados são obtidos quando o número de componentes-base Q é comparativamente elevado em relação ao número de fontes, o que torna necessário agrupar as componentes-base segundo algum critério para obter as fontes originais. Vale ressaltar que a determinação do valor de Q é em si um grande desafio, e que algumas extensões da NMF, como a NMFD e NMF2D, tentam resolver este problema fazendo Q igual ao número de fontes, mas este não é o foco de estudo deste trabalho.

Em [10] o critério proposto para o agrupamento das componentes-base é uma relação logarítmica entre as suas intensidades nos respectivos canais:

$$\mathbf{i} = \log\left[\frac{\mathbf{H}(1,:)}{\mathbf{H}(2,:)}\right], \tag{4.11}$$

onde cada elemento da primeira linha da matriz $\mathbf{H}$ é dividido pelo correspondente elemento de sua segunda linha. O vetor $\mathbf{i}$ é então formado pelo logaritmo de cada um destes valores (um para cada componente-base).

A relação intercanal apresentada acima baseia-se no modelo proposto em [33], onde todas as fontes estão presentes em todos os canais, porém com ganhos diferentes. Essa relação como proposta em [10] está claramente limitada ao caso de dois canais, já que aquele trabalho tratou exatamente desse caso.

Agrupamento das bases – Passo 5

Seguindo o algoritmo proposto, o agrupamento das componentes-base é realizado utilizando-se a relação apresentada acima; especificamente, o agrupamento ocorre sobre o vetor de intensidades $\mathbf{i}$, onde componentes-base com valores similares de intensidade serão agrupadas.

Para o caso em que $P = 2$, ou seja, existem duas fontes e dois canais, o vetor $\mathbf{i}$ apresentará valores positivos para as componentes-base pertencentes à fonte com maior intensidade no canal 1, e valores negativos para a fonte com maior intensidade no canal 2. Na Figura 4.3 temos um exemplo desse caso (vide Seção 4.1.2), onde uma senoide decaindo exponencialmente foi misturada com um *chirp* (a senoide presente com maior intensidade no canal 1, e o *chirp* presente com maior intensidade no canal 2); neste exemplo, o algoritmo da NTF foi executado por 500 iterações utilizando $Q = 10$.

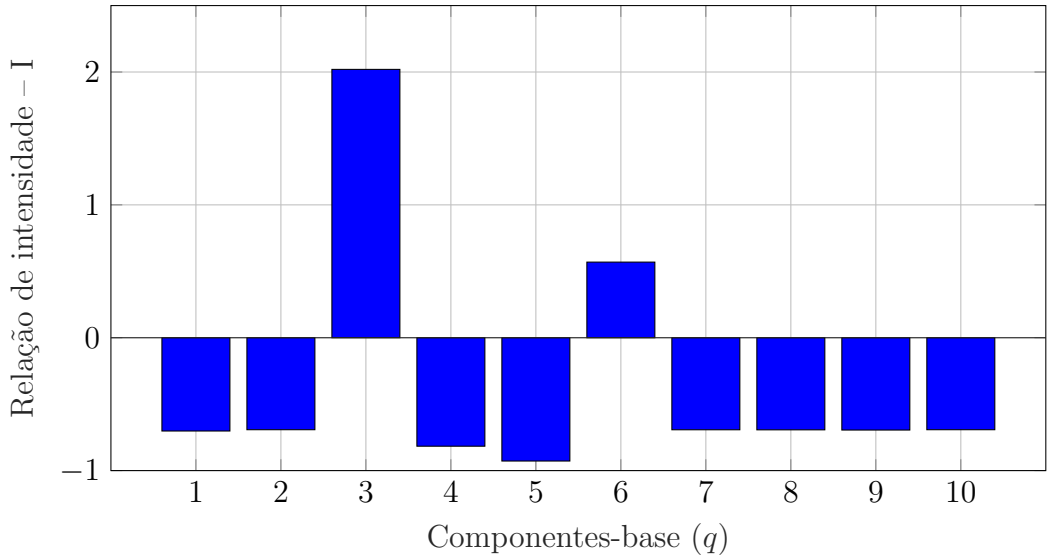

Figura 4.3: Gráfico de relação de intensidades da equação (4.11) para $Q = 10$.

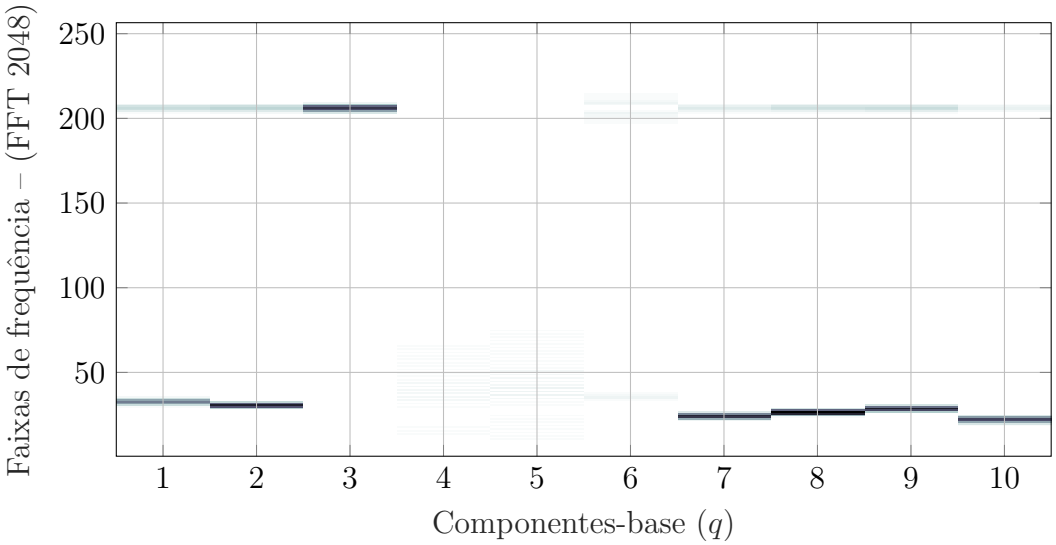

Figura 4.4: Representação da matriz de componentes-base.

A Figura 4.4 mostra uma representação da matriz de componentes-base **B**. Essa representação é análoga a um espectrograma, com as faixas de frequência no eixo vertical, porém em vez de identificar os quadros em que determinadas faixas de frequência estão presentes, ela indica a quais componentes-base os diferentes padrões espectrais pertencem. Dessa forma o eixo horizontal contém apenas índices, e para cada índice (componente-base) o eixo vertical indica quais frequências estão presentes; além disso a intensidade com as quais as frequências estão presentes em cada componente-base é representada pela respectiva intensidade do traço no gráfico. Pode-se observar nesse exemplo que apesar de o padrão espectral da senoide (padrão próximo à faixa de frequência 200) estar fortemente concentrado na

terceira componente-base, parte da informação correspondente à senoide acabou espalhada em praticamente todas as outras, além disso as componentes-base 4,5 e 6 quase não contêm informações sobre as fontes o que pode indicar que a separação seria possível com um número menor de componentes-base. Examinando a Figura 4.3, pode-se perceber que as componentes-base podem ser agrupadas utilizando o sinal da relação de intensidade.

Para o caso em que $P > 2$ ($P > C$, sendo $C = 2$ para o modelo proposto), ou seja, onde existem mais fontes do que canais, será necessário agrupar as componentes-base de acordo com os valores das intensidades, e não apenas seus sinais. Essa particularidade pode ser uma fraqueza desse tipo de agrupamento, pois uma mesma fonte pode vir a ser composta de componentes-base com diferentes intensidades.

Vale ressaltar que o agrupamento realizado sobre o vetor de intensidades $\mathbf{i}$ não utiliza explicitamente a informação temporal. Isso decorre diretamente do modelo utilizado, que enfatiza apenas a diferença de ganhos entre os canais. Por outro lado, pensando de forma genérica, a informação temporal poderia ser útil para separar fontes que não ocorrem de maneira simultânea.

O agrupamento das componentes-base é de certa forma independente do restante do algoritmo, logo diferentes algoritmos podem ser utilizados neste passo de forma modular. Em [10], o algoritmo utilizado foi o KNN – *K nearest neighbours*, que é um algoritmo supervisionado; como o foco deste trabalho será a separação de áudio cega, e o KNN requer um treinamento com exemplos, os testes foram realizados utilizando o algoritmo *K-means*.

4.1.2 Testes

Nos testes apresentados a seguir, o algoritmo descrito acima será utilizado para realizar a separação de diferentes misturas; além de gráficos apresentando diferentes aspectos do desempenho do algoritmo, também será apresentada uma avaliação subjetiva acompanhada do resultado das métricas objetivas SDR, SIR e SAR.

Teste 1

Neste teste, o algoritmo é aplicado sobre um sinal obtido a partir da mistura de um *chirp* decrescente (sinal senoidal cuja frequência varia linearmente entre 650 Hz e 450 Hz ao longo de 1 segundo) e uma senoide com amplitude decrescente (sinal senoidal de 4400 Hz cuja amplitude decresce de forma exponencial, atingindo 7% do valor original após 2 segundos). A Tabela 4.1 apresenta parâmetros da mistura e do algoritmo utilizados neste teste.

Tabela 4.1: Parâmetros do Teste 1.

Frequência de amostragem	44100 Hz
Tamanho da FFT	2048
Tamanho da Janela	1024
Percentual de sobreposição (janela)	75%
Número de componentes-base (Q)	10
Número de iterações	500

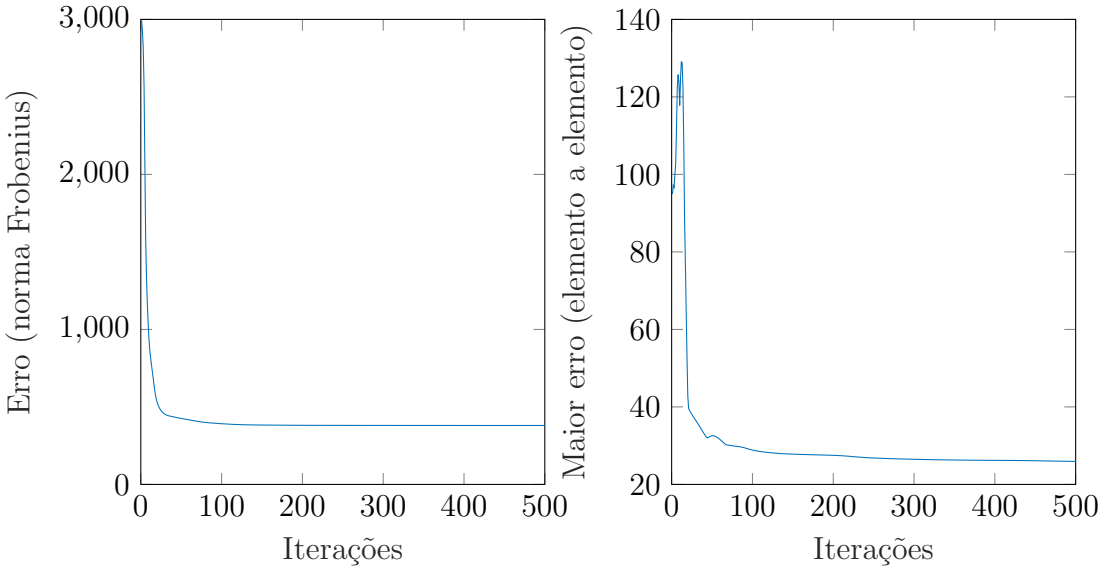

Figura 4.5: Convergência: Gráfico da função custo (Erro) e do maior erro da estimativa (teste 1).

A Figura 4.5 apresenta a variação de duas medidas de erro ao longo das iterações do algoritmo. A primeira medida de erro é a função custo utilizada pelo algoritmo, e a segunda apresenta o maior erro, ou diferença, encontrado entre o tensor estimado pelo algoritmo e o tensor original; para este teste o algoritmo converge após cerca de 100 iterações, próximo da iteração onde também ocorre o menor erro.

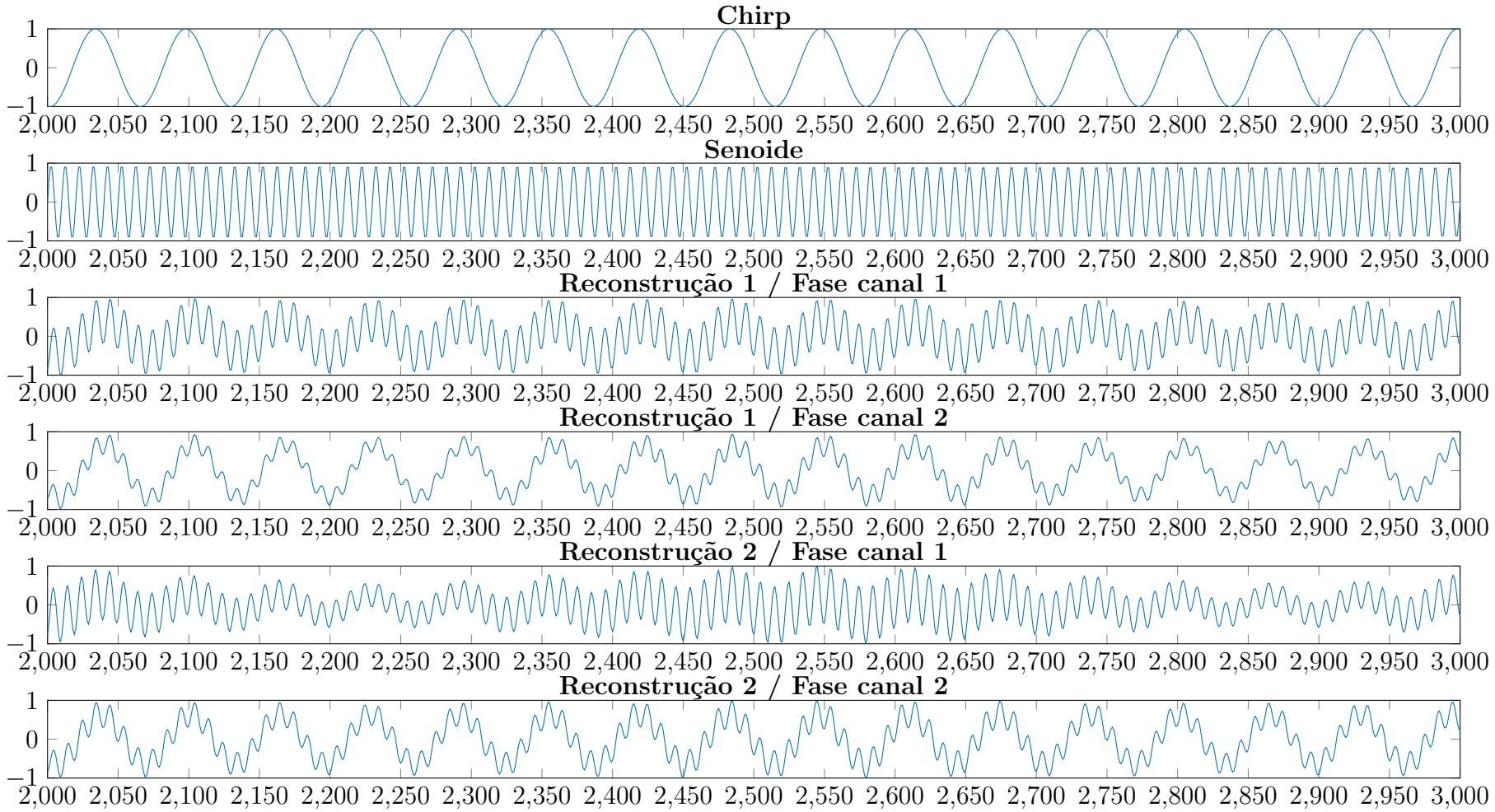

Figura 4.6: Fontes originais e estimadas.

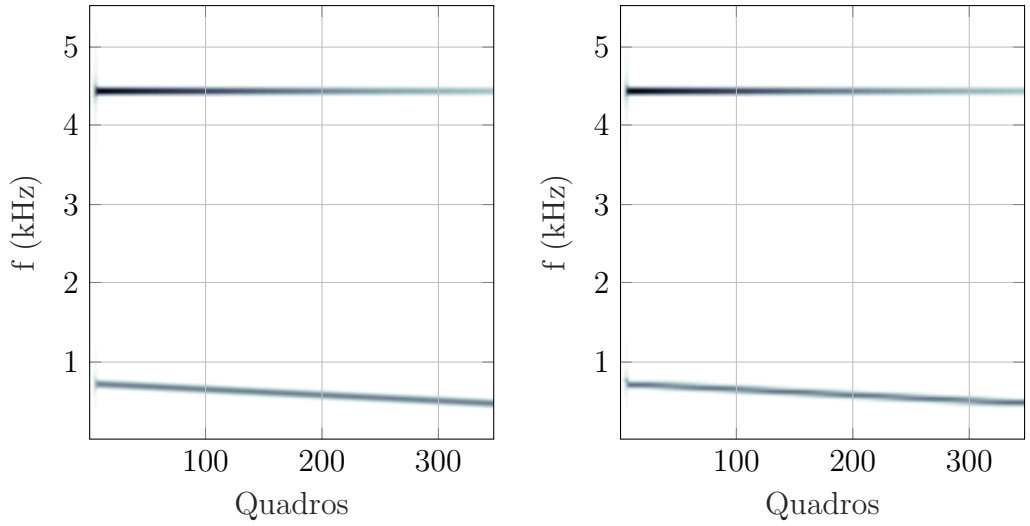

Figura 4.7: Espectrogramas de magnitude (original e estimado) – Canal 1 (teste 1).

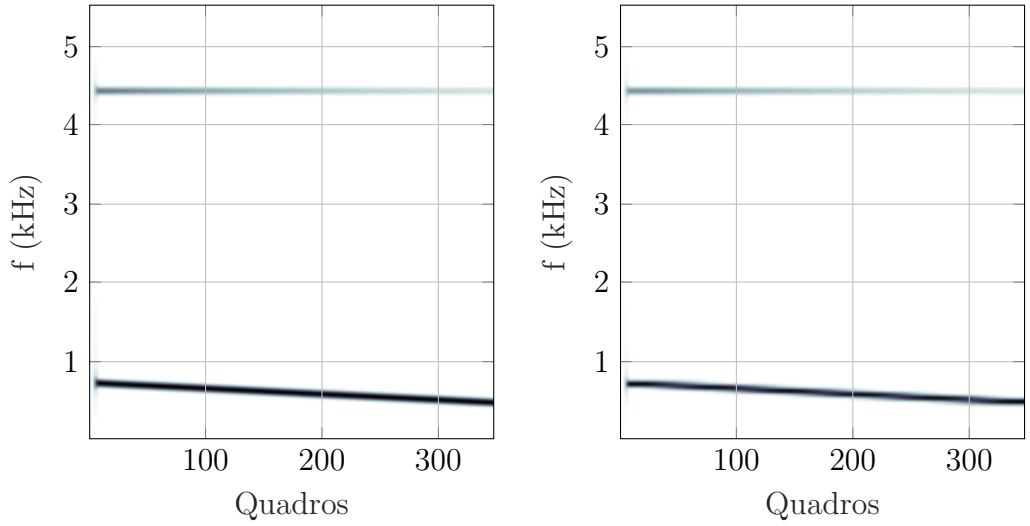

Figura 4.8: Espectrogramas de magnitude (original e estimado) – Canal 2 (teste 1).

A Figura 4.6 apresenta os sinais originais utilizados para gerar a mistura e as reconstruções realizadas a partir dos resultados da execução do algoritmo; 'Reconstrução 1/Fase canal 1' se refere ao sinal obtido pela associação entre a estimativa de uma das fontes e a fase do canal 1, da mesma forma 'Reconstrução 2/Fase canal 1' se refere à outra fonte associada à fase do canal 1 (reconstruções equivalentes também foram realizadas utilizando a fase do canal 2). Os resultados iniciais não parecem muito promissores, porém as Figuras 4.7 e 4.8 mostram que a NTF reproduziu fielmente o espectrograma de magnitude de ambos os canais. Além disso, como visto anteriormente nas Figuras 4.4 e 4.3, as componentes-base separaram as duas fontes. Dadas estas observações, pode-se tentar identificar qual passo, ou quais passos (vide seção 4.1.1), são responsáveis pelos resultados encontrados. Imediatamente os passos 1, 2 e 3 podem ser excluídos, pois tratam apenas da obtenção da

representação utilizada para a fatoração e da fatoração em si, que como já foi visto foi bem sucedida. Os passos 4, 6 e 8 também podem ser excluídos, pois a Figura 4.6 indica um problema na separação das fontes, que não é influenciado por estes passos. Finalmente, como já visto anteriormente, o agrupamento das bases é simples bastando separar as componentes-base de acordo com o sinal das suas respectivas relações de intensidade $\mathbf{i}$, o que também exclui o passo 5; isso implica que o problema encontra-se necessariamente no passo 7, cuja proposta talvez seja inadequada para casos tão simples quanto os deste exemplo. É interessante ressaltar que o algoritmo gerou 4 fontes, e não as 2 esperadas; isso ocorre porque é necessário escolher de qual mistura virá a informação de fase de cada fonte. Como o artigo original não apresenta isso de forma explícita, por simplicidade todas as combinações foram geradas na implementação utilizada no exemplo.

Os passos 7 e 8 também poderiam ser unificados em um único passo de síntese. Em [34] foi realizada uma revisão do estado da arte deste tipo de técnica; este estudo não tratou do caso multicanal; porém, todas as técnicas revisadas eram significativamente mais avançadas que o proposto nos passos 7 e 8, que por sua vez são de simples implementação e baixa complexidade computacional.

Tabela 4.2: Avaliação objetiva dos resultados do método original (teste 1).

métrica	resultado
SDR (fonte 1)	2,6289 dB
SDR (fonte 2)	-16,2040 dB
SIR (fonte 1)	18,1090 dB
SIR (fonte 2)	10,7171 dB
SAR (fonte 1)	-1,0783 dB
SAR (fonte 2)	-12,1492 dB

A Tabela 4.2 apresenta as métricas objetivas para a separação obtida pelo algoritmo neste teste. Os valores da métrica SIR indicam que ambas as fontes foram separadas, porém as métricas SDR e SAR indicam que a reconstrução de uma das fontes apresenta significativa quantidade de artefatos. Subjetivamente, é possível reconhecer as fontes destacadas nas respectivas reconstruções (principalmente a fonte 1), porém, em ambas as reconstruções todos os sinais estão presentes.

Teste 2

Neste teste, o algoritmo foi utilizado em uma mistura obtida a partir de dois sinais de fala com dois locutores distintos (um homem e uma mulher), gravados em estúdio profissional com palavra de 24 bits e frequência de amostragem de 48000 kHz; a

mistura foi realizada de forma a simular uma conversa entre dois locutores. A Tabela 4.3 apresenta parâmetros da mistura e do algoritmo utilizados neste teste.

Tabela 4.3: Parâmetros do Teste 2.

Frequência de amostragem	48000 Hz
Tamanho da FFT	2048
Tamanho da Janela	1024
Percentual de sobreposição (janela)	75%
Número de componentes-base (Q)	20
Número de iterações	400

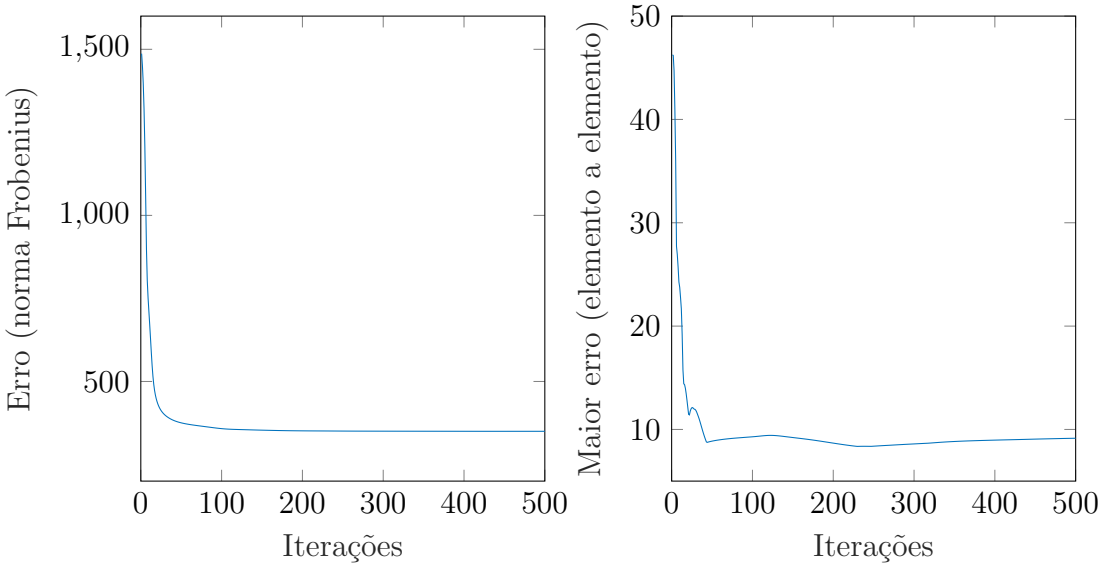

Figura 4.9: Convergência: Gráfico da função custo (Erro) e do maior erro da estimativa (teste 2).

Da mesma forma que no teste 1, como pode ser observado na Figura 4.9, o algoritmo converge após cerca de 100 iterações, sendo que a medida de maior erro se apresenta relativamente estável após 50 iterações.

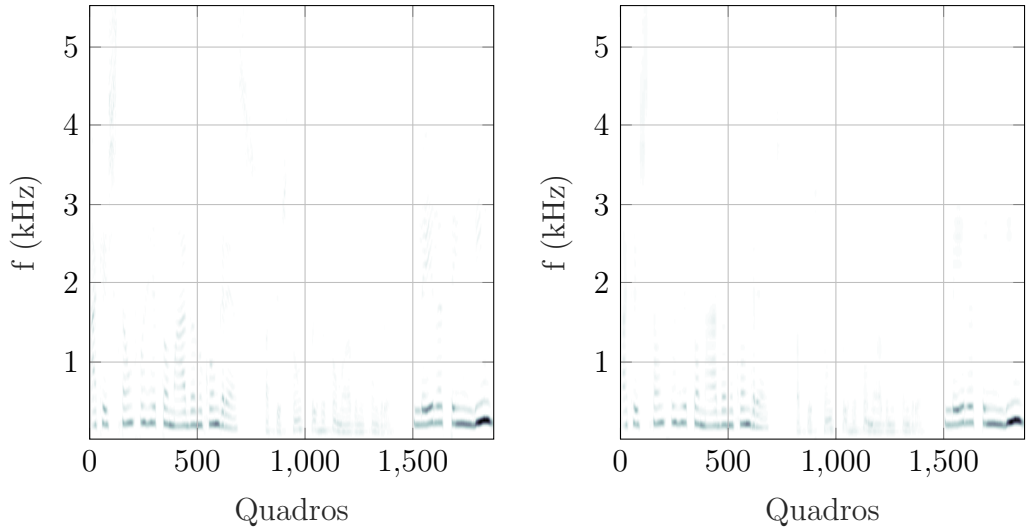

Figura 4.10: Espectrogramas de magnitude (original e estimado) – Canal 1 (teste 2).

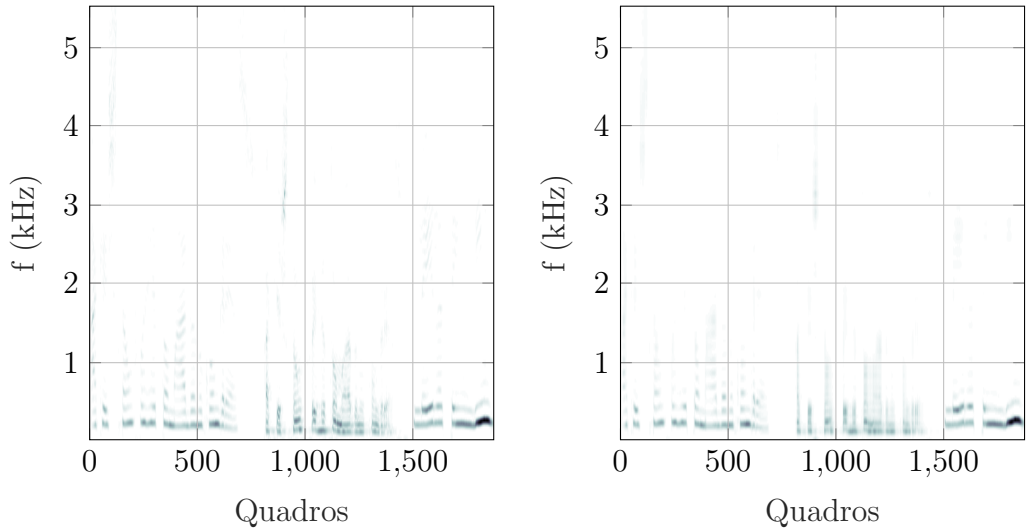

Figura 4.11: Espectrogramas de magnitude (original e estimado) – Canal 2 (teste 2).

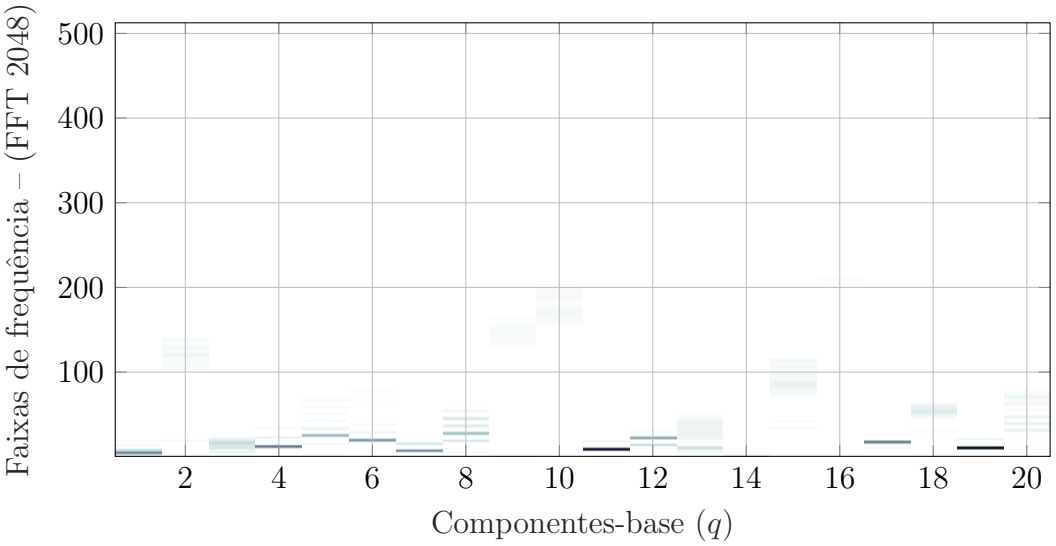

Figura 4.12: Representação da matriz de componentes-base (teste 2).

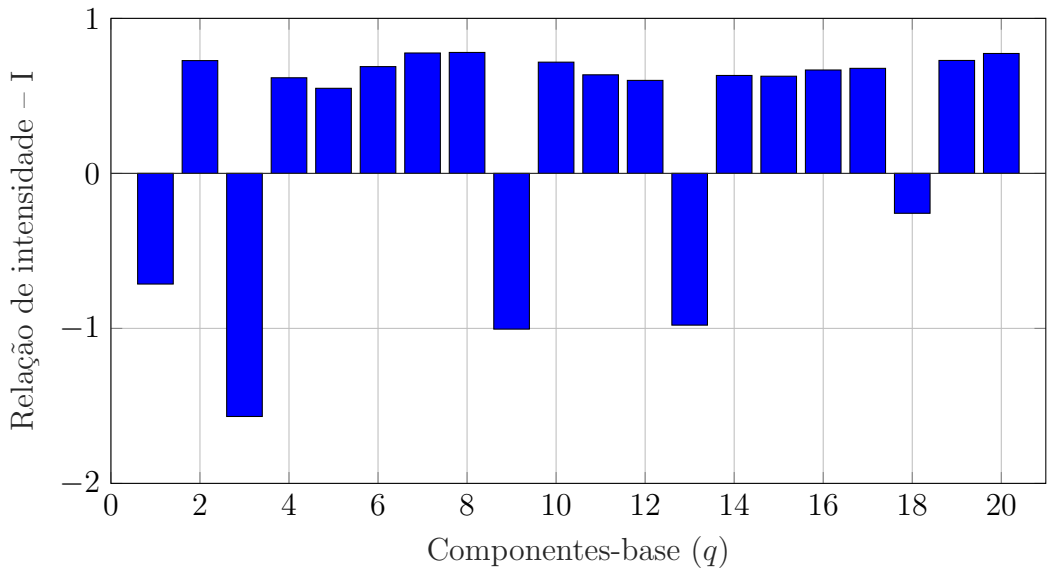

Figura 4.13: Gráfico da relação de intensidades (teste 2).

As Figuras 4.10, 4.11, 4.12 e 4.13 apresentam, respectivamente, os espectrogramas de magnitude originais e estimados de ambos os canais, a representação da matriz de componentes-base $\mathbf{B}$ e a relação de intensidades (assim como no teste anterior). Dados os padrões espectrais mais elaborados deste teste, as Figuras 4.12 e 4.13 são de difícil avaliação, apenas o que pode ser dito é que novamente o agrupamento das componentes-base a partir da relação de intensidades é simples, sendo dado novamente pelo sinal da relação de intensidade de cada componente-base. Por outro lado, as Figuras 4.10 e 4.11 indicam que o algoritmo foi capaz de reproduzir o espectrograma de magnitude das misturas.

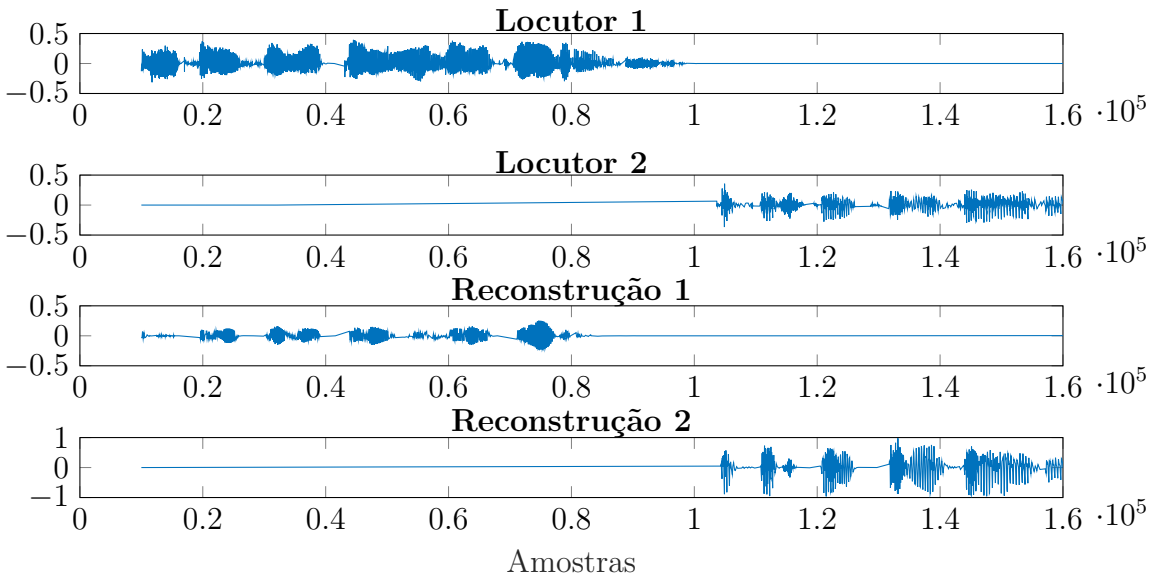

Figura 4.14: Fontes originais e reconstruídas.

A Figura 4.14 apresenta um trecho dos sinais originais e os trechos correspondentes dos sinais separados, identificados por 'Reconstrução' 1 e 2; pode-se observar que os locutores foram efetivamente separados; porém, em ambas as reconstruções também é possível detectar artefatos que não estavam presentes nos sinais originais. Essas observações coincidem com os resultados obtidos para os critérios objetivos que são apresentados na Tabela 4.4. Considerando apenas estes critérios, pode-se inferir que uma das fontes (fonte 1 na tabela) foi melhor extraída da mistura, porém os valores de SIR e SAR indicam uma baixa incidência de interferência proveniente de outras fontes e uma significativa incidência de artefatos nos sinais reconstruídos. Subjetivamente, ambos os locutores foram separados, porém, na reconstrução do locutor 2 podem ser ouvidos artefatos no momento em que o locutor 1 falaria; em ambas as reconstruções são percebidas distorções.

Tabela 4.4: Avaliação objetiva dos resultados do método original (teste 2).

métricas	resultados
SDR (fonte 1)	5,79 dB
SDR (fonte 2)	0,60 dB
SIR (fonte 1)	102,76 dB
SIR (fonte 2)	94,45 dB
SAR (fonte 1)	5,69 dB
SAR (fonte 2)	0,59 dB

Na Tabela 4.5 são apresentados os valores do critério SDR de acordo com o número de iterações: pode-se observar que em poucas iterações o SDR já está

próximo de seu máximo, e que após 100 iterações o SDR praticamente se mantém constante.

Tabela 4.5: Máxima SDR até dada iteração.

iteração	SDR máxima
1	2,01 dB
10	3,96 dB
20	5,16 dB
30	5,49 dB
40	5,57 dB
50	5,64 dB
60	5,68 dB
70	5,70 dB
80	5,72 dB
90	5,75 dB
100	5,78 dB
500	5,79 dB

4.2 Considerações Práticas

O algoritmo proposto em [10] apresenta uma modelagem simples do problema de separação multicanal de fontes sonoras. Diferentemente da NMF-SCM o algoritmo requer algumas etapas de pós-processamento, especificamente uma etapa de agrupamento das componentes-base e posteriormente uma etapa de reconstrução dos sinais das fontes separadas. O algoritmo em si é simples, e de certa forma análogo a uma versão multicanal da NMF básica. Essa simplicidade, tanto do algoritmo quanto da modelagem, é refletida também na sua complexidade computacional, que é relativamente baixa (principalmente se fizermos comparações com a NMF-SCM). Os parâmetros equivalentes aos utilizados na análise realizada no Capítulo 3 são:

- sinais de 10 segundos de duração com frequência de amostragem 24 kHz
- STFT com 2048 amostras
- 50% de sobreposição de quadros
- Q (componentes-base) = 60
- P (fontes) = 2

- M (microfones) = 2 (o algoritmo proposto não considera casos com mais de 4 microfones)
- L (quadros) = 234 (valor aproximado calculado a partir dos outros valores)

Utilizando estes parâmetros, o tensor $\mathcal{X}$ terá dimensões $1024 \times 234 \times 2$, 8 vezes menos elementos que o tensor equivalente para a NMF-SCM (ou 4 vezes, se considerarmos o caso de 4 canais); a matriz $\mathbf{H}$ terá dimensões 2×60, a matriz $\mathbf{B}$ terá dimensões 1024×60 e $\mathbf{G}$ terá dimensões 60×234. É interessante notar que, com a exceção do tensor $\mathcal{X}$ (e também da matriz $\mathbf{Z}$, que não existe na NTF), as dimensões das matrizes $\mathbf{M}$, $\mathbf{B}$ e $\mathbf{G}$, utilizadas na NMF-SCM, são exatamente as mesmas das matrizes $\mathbf{H}$, $\mathbf{B}$ e $\mathbf{G}$ utilizadas pela NTF. Vale ressaltar que, apesar de suas dimensões idênticas, as matrizes $\mathbf{M}$ e $\mathbf{H}$ têm interpretações claramente distintas, a primeira realizando um mapeamento de componente-base para fonte e a segunda um mapeamento de componente-base para canal. Comparando os dois algoritmos, pode-se notar uma clara analogia entre a NTF e o núcleo NMF presente na NMF-SCM.

A atualização das matrizes $\mathbf{H}$, $\mathbf{B}$ e $\mathbf{G}$ consiste basicamente em simples multiplicações e divisões que já são realizadas de maneira vetorizada, logo o tempo de execução é curto, claramente abaixo de um segundo por iteração (pelo menos 300 vezes mais rápido que a mais rápida implementação obtida para NMF-SCM). Em todos os testes realizados o algoritmo nunca apresentou nenhuma instabilidade numérica, quase sempre convergindo após 150-300 iterações.

Capítulo 5

Contribuições

Neste capítulo serão apresentadas contribuições desenvolvidas com base nos estudos e testes realizados nos capítulos anteriores. Todas as alterações são referentes ao algoritmo de separação do Capítulo 4, porém ao menos duas das propostas poderiam ser utilizadas com outros algoritmos. As implementações realizadas para os algoritmos dos Capítulos 3 e 4 também poderiam ser consideradas contribuições, mas nesse caso vale ressaltar que alguns aspectos das implementações foram inferidos (como a implementação do passo 7 descrito na Seção 4.1.1) ou alterados em relação aos descritos nos artigos (como o uso do algoritmo *K-means* em vez do KNN no passo 5).

5.1 Filtragem do espectrograma separado

Os resultados de algoritmos de separação de fontes como a NTF, tipicamente, são espectros de magnitude correspondentes às fontes separadas; algoritmos de reconstrução de fase como o RTISI e MISI podem ser utilizados para estimar a informação de fase correspondente a cada espectrograma de magnitude, porém o algoritmo apresentado no capítulo anterior apresenta uma etapa (passo 7) de menor custo computacional para obter a mesma informação de fase.

Durante os testes iniciais, cujos resultados foram apresentados na Seção 4.1.2, foi percebida uma possível deficiência no passo 7 utilizado por aquele algoritmo. Especificamente, a fatoração resultou em uma matriz $\mathbf{B}$ cujas componentes-base separam os padrões espetrais das duas fontes, e um uma relação de intensidade $\mathbf{i}$ capaz de agrupar corretamente estas componentes-base; mesmo assim, os sinais reconstruídos ainda apresentavam características de ambas as fontes.

Uma etapa de filtragem é potencialmente uma alternativa simples, e também de baixo custo computacional, para obter uma melhor estimativa do espectro correspondente a uma fonte. A ideia é simples: em vez de associar a fase da mistura ao espetrograma de magnitude estimado para cada fonte, um filtro é estimado e utili-

zado para extrair o espectrograma de fonte a partir do espectrograma da mistura, de forma análoga ao método de reconstrução utilizado pela NMF-SCM.

5.1.1 Novo passo 7

Em [35] foram comparadas algumas técnicas para a filtragem do espectrograma separado, e a filtragem de Wiener apresentou os melhores resultados. Um filtro de Wiener para a estimativa de uma das fontes pode ser aproximado pela seguinte expressão:

$$\mathbf{S}_{\mathrm{e}} = \frac{||\mathbf{S}_{\mathrm{e}}||^2}{||\mathbf{M}_c||^2} \odot \mathbf{M}_c \tag{5.1}$$

onde $\mathbf{S}_{\mathrm{e}}$ é o espectrograma estimado de uma fonte S, $\mathbf{M}_c$ é o espectrograma da mistura no canal c e tanto a multiplicação (simbolizada por $\odot$) quanto a divisão (simbolizada por $\frac{\mathbf{A}}{\mathbf{B}}$) serão realizadas elemento a elemento.

Teste 1 com novo passo 7

A Figura 5.1 é equivalente à Figura 4.6; porém, agora com o algoritmo utilizando o novo passo 7; percebem-se melhorias nas reconstruções, porém também novos artefatos.

Tabela 5.1: Avaliação objetiva dos resultados do método com passo 7 modificado.

métrica	resultado
SDR (fonte 1)	7,6 dB
SDR (fonte 2)	-14,8 dB
SIR (fonte 1)	77,9 dB
SIR (fonte 2)	11,0 dB
SAR (fonte 1)	-8,8 dB
SAR (fonte 2)	-11,1 dB

A Tabela 5.1 apresenta as métricas objetivas para a separação obtida com o algoritmo modificado. O valor da métrica SIR para a fonte 1 apresentou aumento expressivo, principalmente quando comparado ao pequeno aumento observado para a fonte 2; os valores da métrica SDR apresentaram uma melhora significativa para ambas as fontes, porém, a métrica SAR apresentou queda expressiva para a fonte 1 e uma pequena melhoria para a fonte 2. Estes resultados indicam que o novo passo 7 potencializou o desempenho observado com o algoritmo inicial, onde ambas as fontes são separadas, porém uma das fontes é extraída (com artefatos) enquanto a outra apresenta distorções. Subjetivamente os resultados são mais interessantes

que os do algoritmo original: ambas as fontes separadas podem ser identificadas e a interferência entre as fontes foi minimizada; em ambas as reconstruções, porém, são percebidos artefatos.

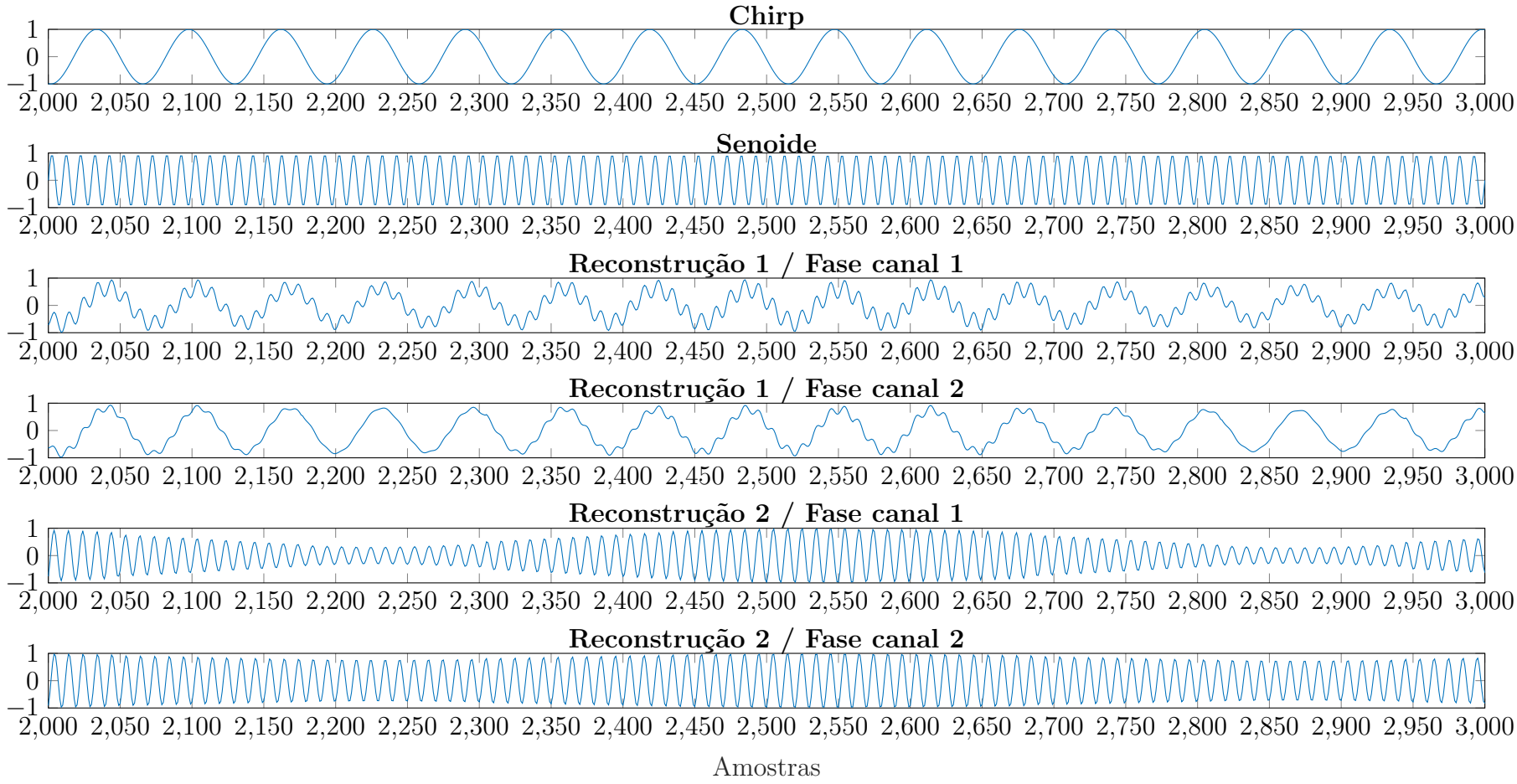

Figura 5.1: Fontes originais e estimadas (utilizando o filtro).

Teste 2 com novo passo 7

A Figura 5.1 é equivalente à Figura 4.6, com o algoritmo utilizando o novo passo 7; observam-se melhorias nas reconstruções, porém também percebem-se novos artefatos.

Tabela 5.2: Avaliação objetiva dos resultados do método com passo 7 modificado.

métrica	resultado
SDR (fonte 1)	19,10 dB
SDR (fonte 2)	0,76 dB
SIR (fonte 1)	80,61 dB
SIR (fonte 2)	62,48 dB
SAR (fonte 1)	18,97 dB
SAR (fonte 2)	1,2624 dB

A Tabela 5.2 apresenta o resultado das métricas objetivas deste novo teste. Para a fonte 1, observa-se uma melhoria expressiva das métricas SDR e SAR, acompanhadas de uma, também expressiva, queda na métrica SIR. Para a fonte 2 as métricas SDR e SAR apresentam apenas uma pequena melhoria, acompanhadas também de uma expressiva queda na métrica SIR.

Assim como no teste anterior, o novo passo 7 potencializou o desempenho do algoritmo original, favorecendo significativamente uma das fontes no processo de separação. As quedas expressivas da métrica SIR indicam um aumento do parâmetro e_{interf} (interferência causada pela presença de outras fontes na reconstrução); logo, o novo passo 7, apesar de diminuir as distorções e artefatos, acaba introduzindo informações de uma fonte na reconstrução da outra. Dito isso, vale ressaltar que os valores encontrados para esta métrica indicam um e_{interf} extremamente baixo; logo, mesmo pequenos aumentos neste parâmetro implicarão expressivas mudanças da métrica SIR.

5.2 Separação multicanal de áudio utilizando NTF

O algoritmo descrito na Seção 4.1 é genérico em relação ao número de canais; porém, como já abordado na Subseção 4.1.1, a relação intercanal proposta está limitada ao caso de dois canais. Esta relação intercanal tem como seu único propósito permitir o agrupamento das componentes-base e aproveita tanto a presença em ambos os canais de todas as fontes (e portanto de todas as componentes-base), quanto a diferença de ganho de cada fonte em cada canal.

Na separação cega de áudio, por definição, o objetivo é obter fontes de uma mistura sem saber, *a priori*, informações sobre as fontes que geraram tal mistura, ou seja, não existem exemplos dos sinais que se deseja obter; como consequência, algoritmos de agrupamento que necessitem de treinamento se mostram inadequados. Neste trabalho o algoritmo de agrupamento utilizado foi o *K-means* [36], pois ele atende aos requisitos da separação cega e também serve como uma referência de desempenho mínimo para a comparação com outros algoritmos.

Uma relação intercanal compatível com mais de dois canais deverá considerar as mesmas informações já utilizadas na proposta original. Uma possível extensão é simplesmente utilizar a mesma relação intercanal, porém agora entre todos os canais; outra opção, um pouco mais simples, é escolher arbitrariamente um canal como referência e obter a relação intercanal de todos os canais em relação a ele. Exemplo para 4 canais:

$$\begin{aligned} \mathbf{i}_{1,2} = \log\left[\frac{\mathbf{H}(1,:)}{\mathbf{H}(2,:)}\right] \quad \mathbf{i}_{1,3} = \log\left[\frac{\mathbf{H}(1,:)}{\mathbf{H}(3,:)}\right] \quad \mathbf{i}_{1,4} = \log\left[\frac{\mathbf{H}(1,:)}{\mathbf{H}(4,:)}\right] \\ \mathbf{i}_{2,3} = \log\left[\frac{\mathbf{H}(2,:)}{\mathbf{H}(3,:)}\right] \quad \mathbf{i}_{2,4} = \log\left[\frac{\mathbf{H}(2,:)}{\mathbf{H}(4,:)}\right] \\ \mathbf{i}_{3,4} = \log\left[\frac{\mathbf{H}(3,:)}{\mathbf{H}(4,:)}\right] \end{aligned} \tag{5.2}$$

ou

$$\mathbf{i}_1 = \log\left[\frac{\mathbf{H}(1,:)}{\mathbf{H}(2,:)}\right] \quad \mathbf{i}_2 = \log\left[\frac{\mathbf{H}(1,:)}{\mathbf{H}(3,:)}\right] \quad \mathbf{i}_3 = \log\left[\frac{\mathbf{H}(1,:)}{\mathbf{H}(4,:)}\right] \tag{5.3}$$

As duas variações propostas acima, porém, são incompatíveis com o algoritmo de agrupamento utilizado, já que a execução do agrupamento sobre cada uma das relações não teria garantida a unicidade dos índices; por exemplo: quando o algoritmo é executado para a relação $\mathbf{i}_1$ ele adota índice 1 para o padrão referente à fonte 1, porém durante a execução sobre a relação $\mathbf{i}_2$ ele poderia adotar para o mesmo padrão o índice 3 ou qualquer outro índice válido. Para compatibilizar uma relação intercanal entre mais de 2 canais com o algoritmo de agrupamento utilizado, é necessário que o agrupamento seja realizado uma única vez. Para isso, uma opção é considerar cada relação intercanal da equação (5.3) como uma dimensão ou variável que o algoritmo *K-means* utilizará:

$$\mathbf{I} = \begin{bmatrix} \mathbf{i}_1 \\ \mathbf{i}_2 \\ \vdots \\ \mathbf{i}_{C-1} \end{bmatrix} \in \mathbb{R}_+^{C-1 \times Q}. \tag{5.4}$$

5.2.1 Teste com 3 canais

De forma similar aos testes da Seção 4.1.2, o algoritmo de separação foi executado sobre uma mistura composta de uma senoide com decaimento exponencial (sinal senoidal de 4400 Hz cuja amplitude decresce de forma exponencial, atingindo 7% do valor original após 2 segundos), um *chirp* (sinal senoidal cuja frequência varia linearmente entre 1000 Hz e 200 Hz ao longo de 2 segundos) e um pulso composto por uma senoide modulada por outra senoide (com frequências 8800 Hz 7300 Hz).

Tabela 5.3: Parâmetros

Frequência de amostragem	44100 Hz
Tamanho da FFT	2048
Tamanho da Janela	1024
Percentual de sobreposição (janela)	87,5%
Número de componentes-base (Q)	20
Número de iterações	500

As figuras a seguir apresentam o comportamento do algoritmo modificado para atuar com 3 canais. As Figuras 5.2, 5.3 e 5.4 apresentam trechos dos sinais originais e suas respectivas reconstruções. O algoritmo não apresenta nenhum problema de convergência, como pode ser observado na Figura 5.5, e os espectrogramas de magnitudes de todos os canais foram estimados fielmente (como pode ser visto nas Figuras 5.6, 5.7 e 5.8).

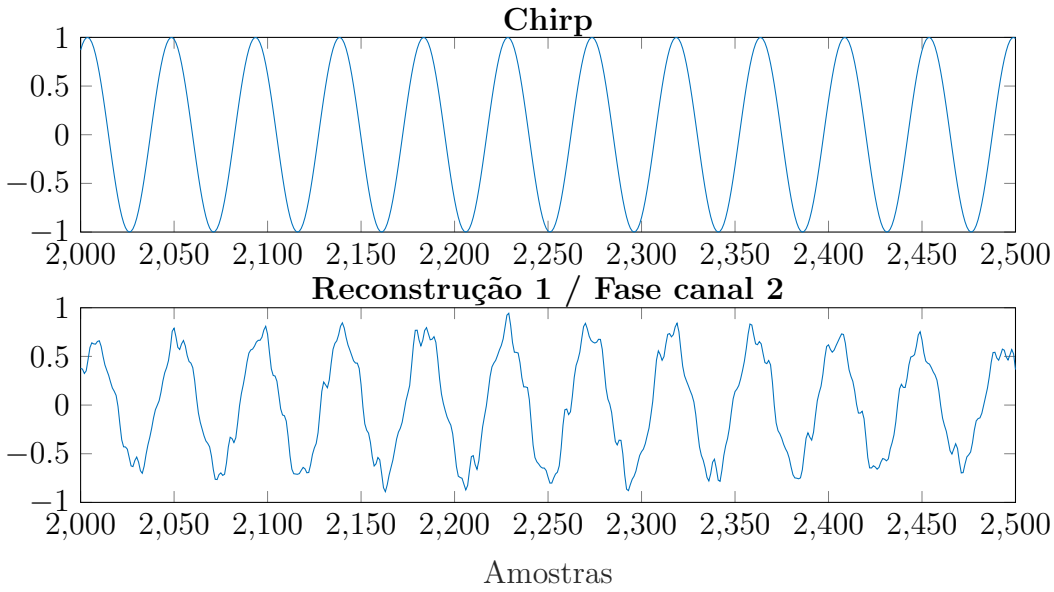

Figura 5.2: Trecho do sinal original e de sua reconstrução (*chirp*).

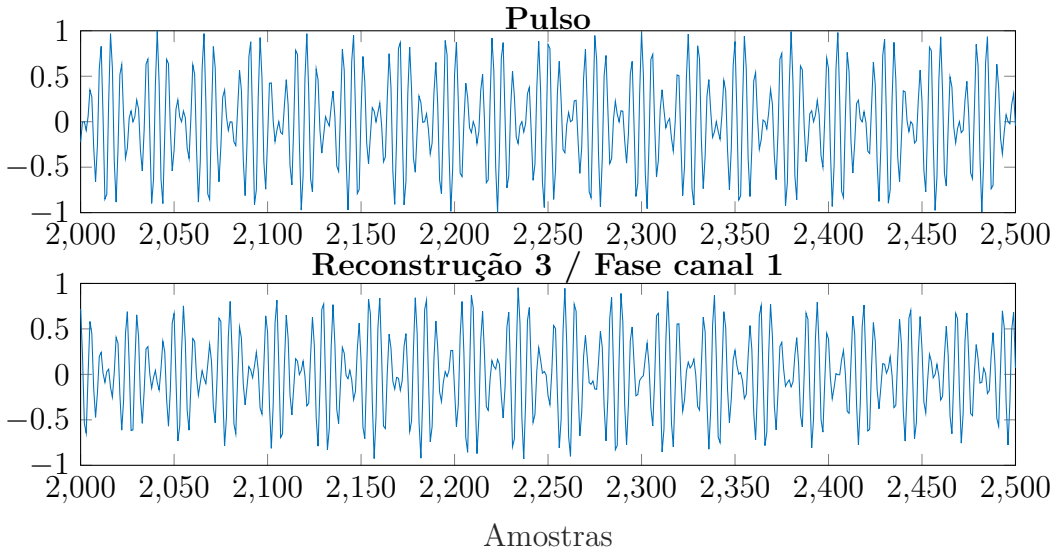

Figura 5.3: Trecho do sinal original e de sua reconstrução (pulso).

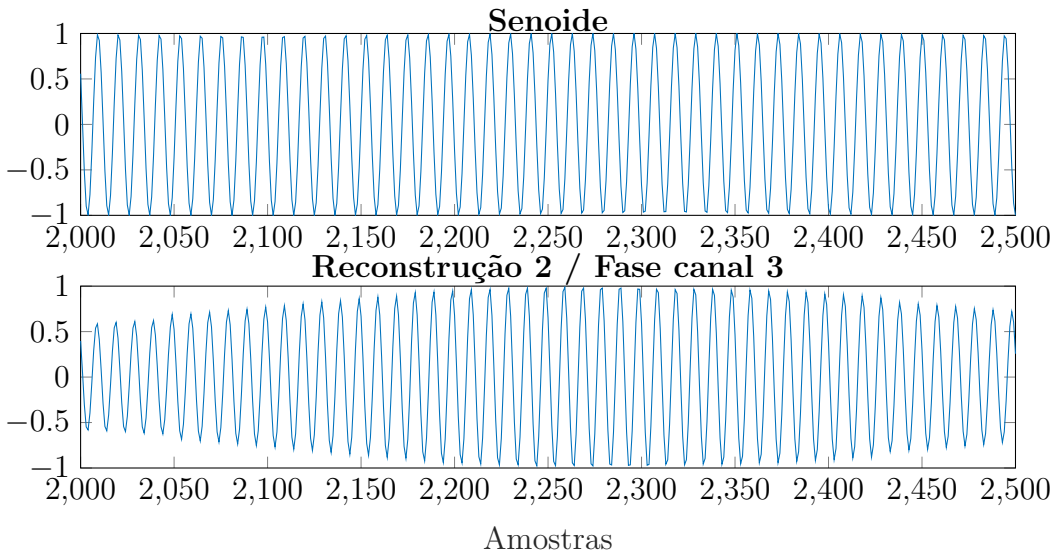

Figura 5.4: Trecho do sinal original e de sua reconstrução (senoíde).

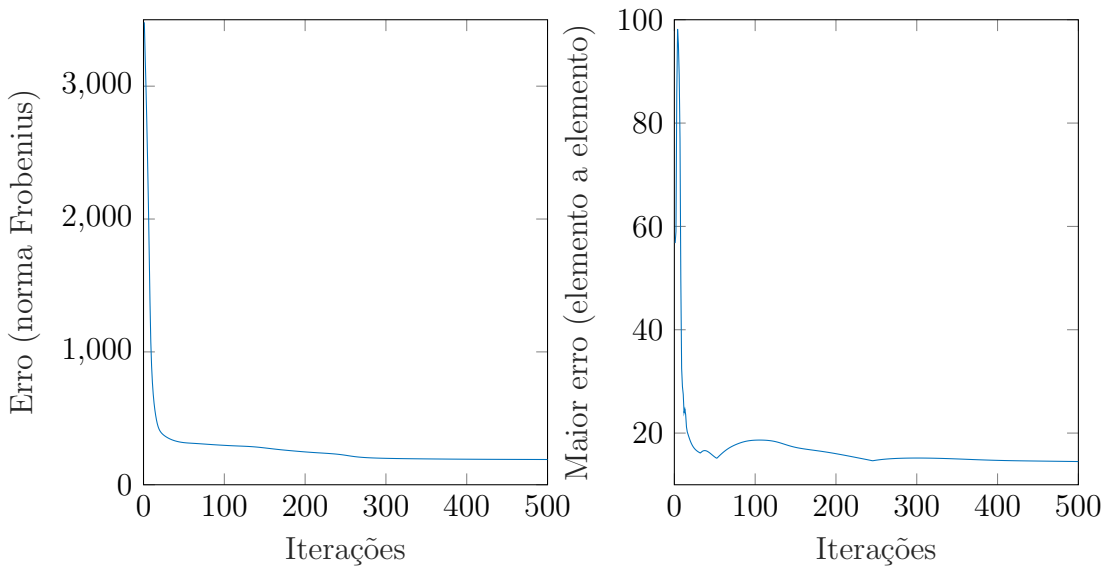

Figura 5.5: Convergência: Gráfico da função custo (Erro) e do maior erro da estimativa (teste 3 canais).

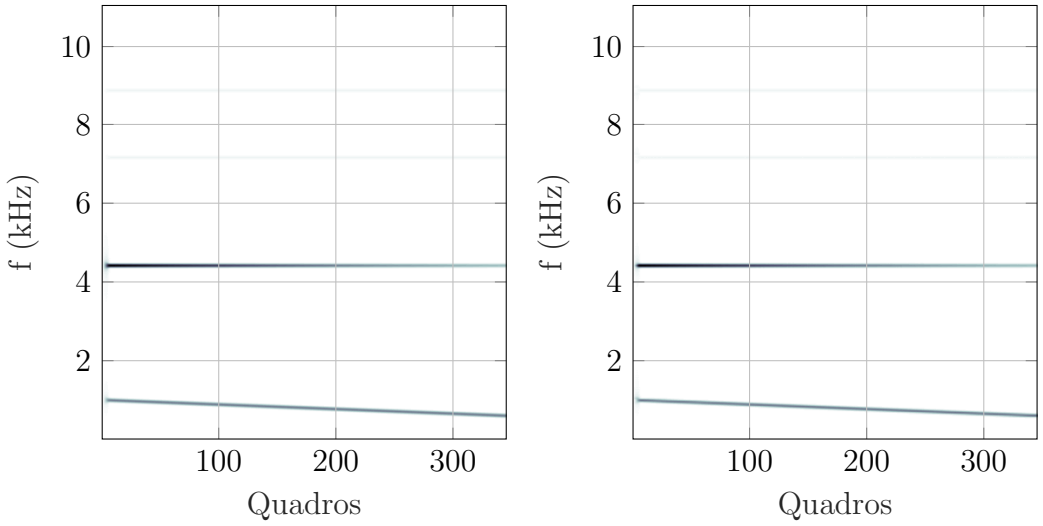

Figura 5.6: Espectrogramas de magnitude (original e estimado) – Canal 1.

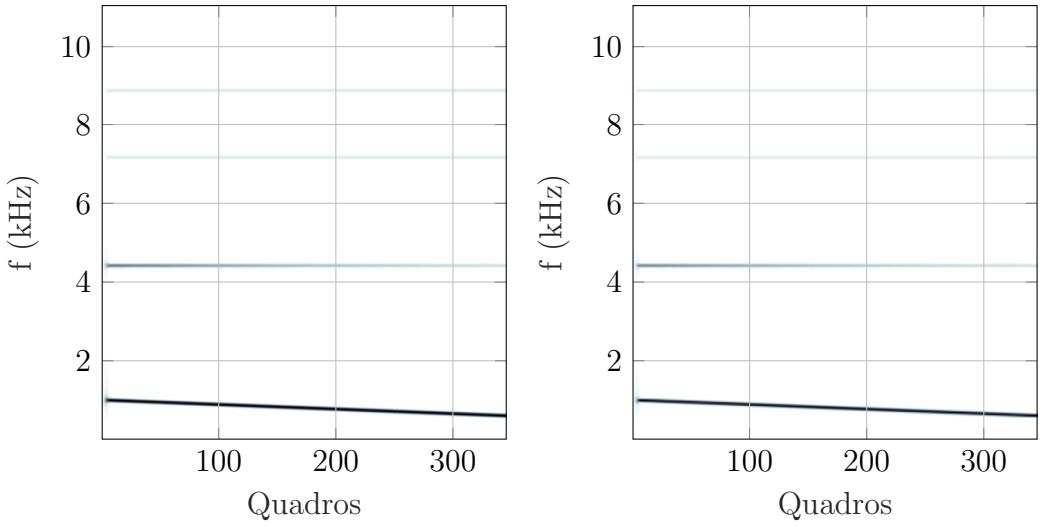

Figura 5.7: Espectrogramas de magnitude (original e estimado) – Canal 2.

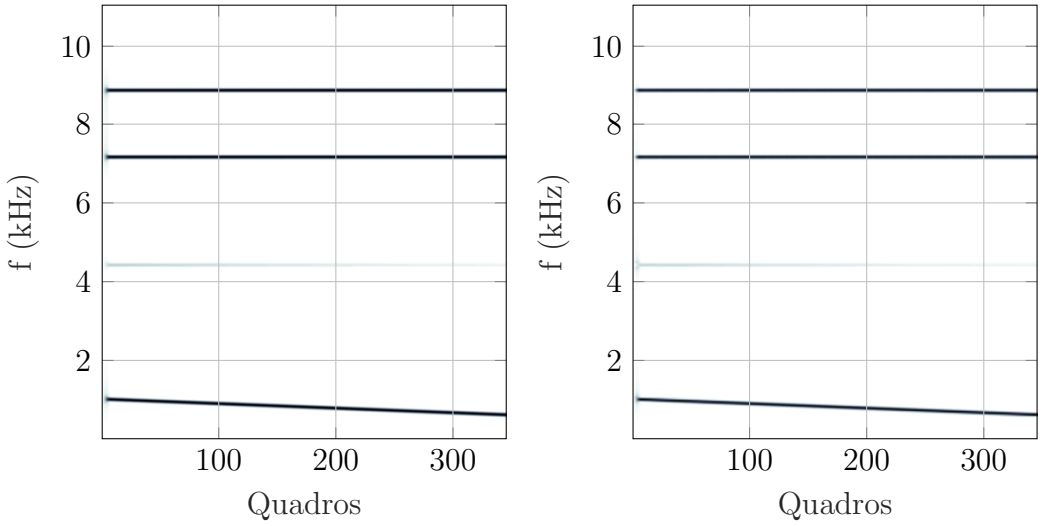

Figura 5.8: Espectrogramas de magnitude (original e estimado) – Canal 3.

As Figuras 5.9 e 5.10 apresentam os dois vetores de intensidade $\mathbf{i}$ referentes à relação entre o primeiro e segundo canais e entre o primeiro e o terceiro canal, respectivamente, que compõem a nova matriz de intensidade. O agrupamento das componentes-base deve considerar ao mesmo tempo as relações $\mathbf{i}_1$ e $\mathbf{i}_2$. Neste exemplo, a relação $\mathbf{i}_1$ apresenta uma clara distinção entre as componentes-base 9, 14 e16 e as demais (desconsiderando a componente-base 12 que, como pode ser observado na Figura 5.11, quase não apresenta informações de nenhuma fonte); já a relação $\mathbf{i}_2$ apresenta distinção similar (porém agora quase sem a presença das componentes-base 1 a 8 e 10, 11, 13, 17, 18, 19 e 20), exceto pela componente-base 15 que aparece com expressiva intensidade (com sinal negativo) em ambas as relações; com base nessas observações, o agrupamento deve seguir os três padrões observados: componentes-base com intensidade negativa em $\mathbf{i}_1$ e quase não presen-

tes em $\mathbf{i}_2$, componentes-base com intensidade positiva em $\mathbf{i}_1$ e $\mathbf{i}_2$, componentes-base com intensidade negativa em $\mathbf{i}_1$ e $\mathbf{i}_2$. Finalmente a Figura 5.11 apresenta os padrões espectrais referentes a cada componente-base, o que valida o agrupamento descrito acima, onde as componentes-base 9, 14 e 16 concentram a informação da senoide, a componente-base 15, representa o pulso e todas as outras ajudam a formar o *chirp*.

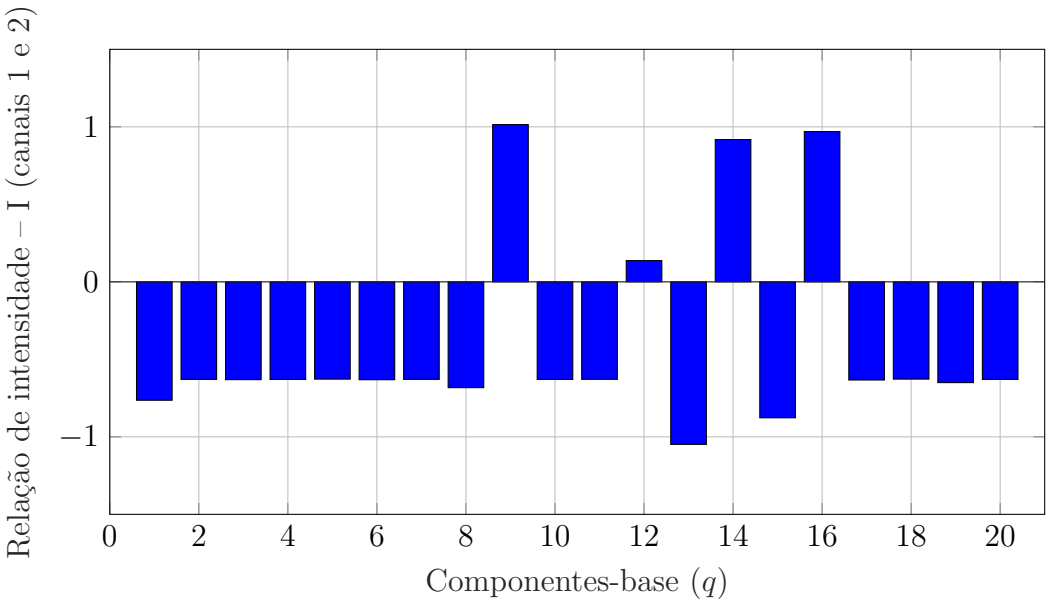

Figura 5.9: Relação de intensidade entre canais 1 e 2 ($\mathbf{i}_1$).

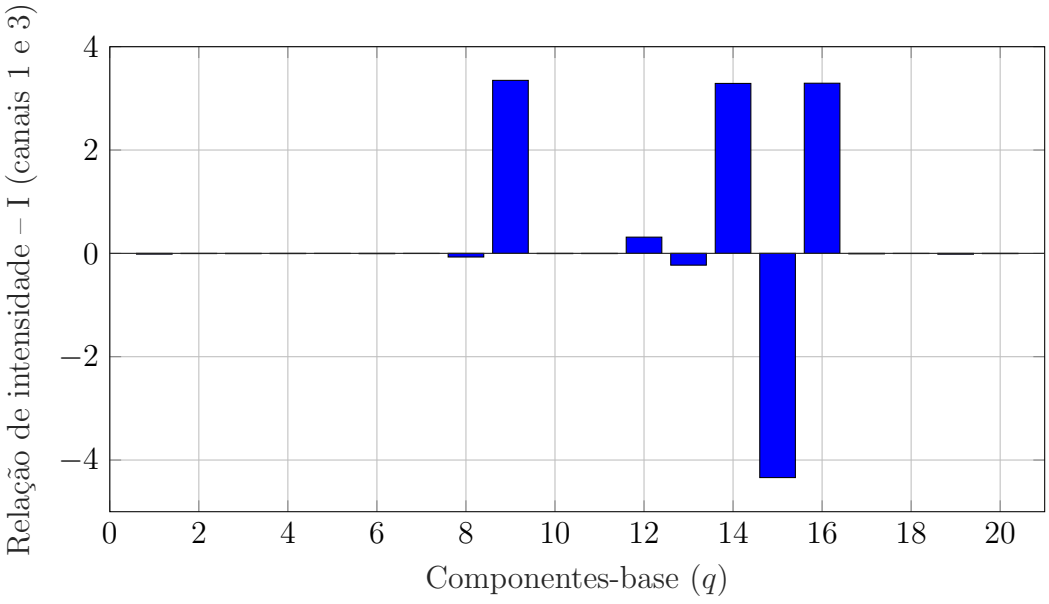

Figura 5.10: Relação de intensidade entre canais 1 e 3 ($\mathbf{i}_2$).

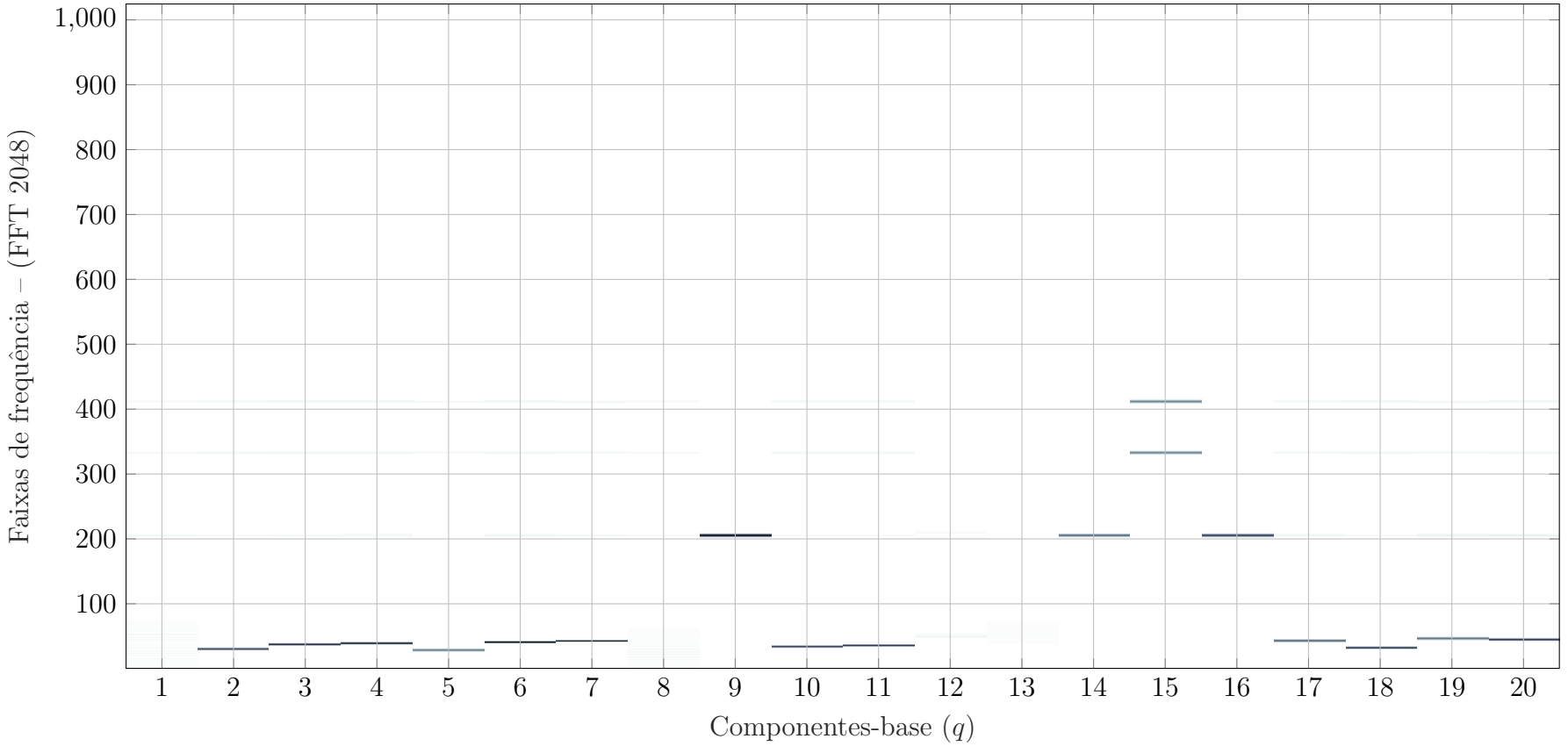

Figura 5.11: Representação do espectrograma de magnitude das componentes-base.

5.3 NTF e diversidade espacial

A modelagem apresentada no capítulo anterior não considera, a princípio, a informação de diversidade espacial que pode estar presente no conjunto de canais. No caso de uma captura e mixagem realizada em um estúdio, onde cada fonte sonora é obtida individualmente (exatamente o problema considerado no desenvolvimento do algoritmo apresentado no capítulo anterior), claramente a informação de diversidade espacial não estará presente. Por outro lado, se considerarmos o caso em que os sinais que irão compor o tensor são provenientes da captura simultânea de várias fontes sonoras, através de um conjunto de microfones, então é razoável inferir que a informação de diversidade espacial estará presente.

Para utilizar a NTF com este tipo de mistura, considerando o mesmo tipo de representação (tensor não-negativo formado pela associação dos espectrogramas de magnitude de cada canal), é necessário avaliar como aproveitar a diversidade espacial. Obviamente é possível executar o algoritmo sem nenhuma alteração, talvez utilizando a diversidade espacial de forma implícita. Isso porém, é potencialmente problemático, pois a lógica utilizada no agrupamento das componentes-base espera encontrar fontes com intensidades significativamente diferentes em cada canal, o que provavelmente não ocorrerá para este tipo de mistura. Uma possibilidade seria tentar realizar o agrupamento das componentes-base utilizando alguma informação espacial; inspirando-se na inicialização do algoritmo da NMF-SCM, pode-se calcular o SRP-PHAT sobre os sinais do conjunto de sensores e tentar determinar quais direções de chegada existem e a quais componentes-base elas se referem.

5.3.1 Agrupamento de componentes-base utilizando SRP-PHAT

Para realizar o agrupamento utilizando as diferentes DoAs é necessário calcular o SRP-PHAT referente a cada componente-base. Isso implica a construção de espectrogramas referentes a cada componente-base para cada canal. Essa construção pode ser realizada de maneira análoga à que foi apresentada na equação (5.1), obtendo os espectrogramas referentes a cada componente-base a partir dos espectrogramas da mistura em cada canal. A Figura 5.12 apresenta o SRP-PHAT de uma componente-base qualquer, resultante da fatoração. Este caso específico se refere à simulação (utilizando o *software CATT-Acoustic* e *Multivolver WCP* [37]) da captura (com frequência de amostragem de 48 kHz) de uma conversa entre dois locutores (utilizando os mesmos sinais descritos na Seção 4.1.2) por um arranjo de 4 microfones no centro de uma sala, estando os locutores posicionados a aproximadamente 135° e 310°, exatamente onde estão os dois maiores picos. O que a Figura 5.12 não mostra é

que todas as outras componentes-base apresentam aproximadamente o mesmo perfil. Isso ocorre pois o SRP-PHAT (particularmente o PHAT) resulta em um branqueamento dos sinais que acaba tornando mais significativa a informação presente na fase, que nesse caso vem da mistura que contém todas as fontes.

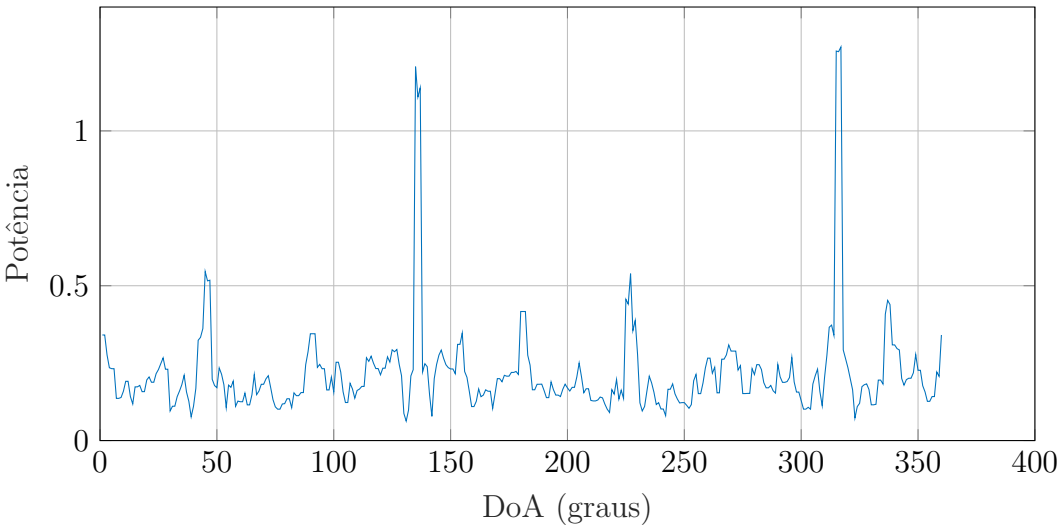

Figura 5.12: SRP-PHAT componente-base 1.

Para evitar esse problema, seria necessário que a construção do espectrograma de cada componente-base fosse feita com algo análogo a uma máscara binária, de modo que o espectrograma de cada componente-base contivesse apenas informações referentes à mesma. É possível obter algo similar a uma máscara binária alterando-se o espectrograma de magnitude referente a cada componente-base, normalizando-o e definindo um limiar mínimo de relevância de forma que toda região do espectrograma de magnitude que se encontre abaixo deste limiar receba o valor zero. As Figuras 5.13 (que apresenta a mesma componente-base da Figura 5.12) e 5.14 mostram exemplos do SRP-PHAT após essa modificação, onde não é mais possível determinar claramente uma direção e a impressão é a de que ocorreu um espalhamento da informação de direção. Essas figuras apresentam similaridades em relação à Figura 2.2, que é um exemplo de execução do algoritmo SRP; pode-se observar que, como efeito colateral do uso desta aproximação a uma máscara binária, também ocorre uma perda de informação de direção da componente-base desejada; dito isso, as figuras mostram que as direções presentes na mistura não estão todas igualmente representadas (particularmente as direções próximas de 135° e 300° nas Figuras 5.13 e 5.14, respectivamente, aparecem destacadas); essa discrepância torna possível diferenciar componentes-base referentes a direções diferentes, o que permite agrupá-las. Vale destacar que enquanto a alteração do espectrograma de magnitude das componentes-base é realizada apenas para a execução do SRP-PHAT, a separação e reconstrução das fontes ocorre sobre os resultados originais da NTF.

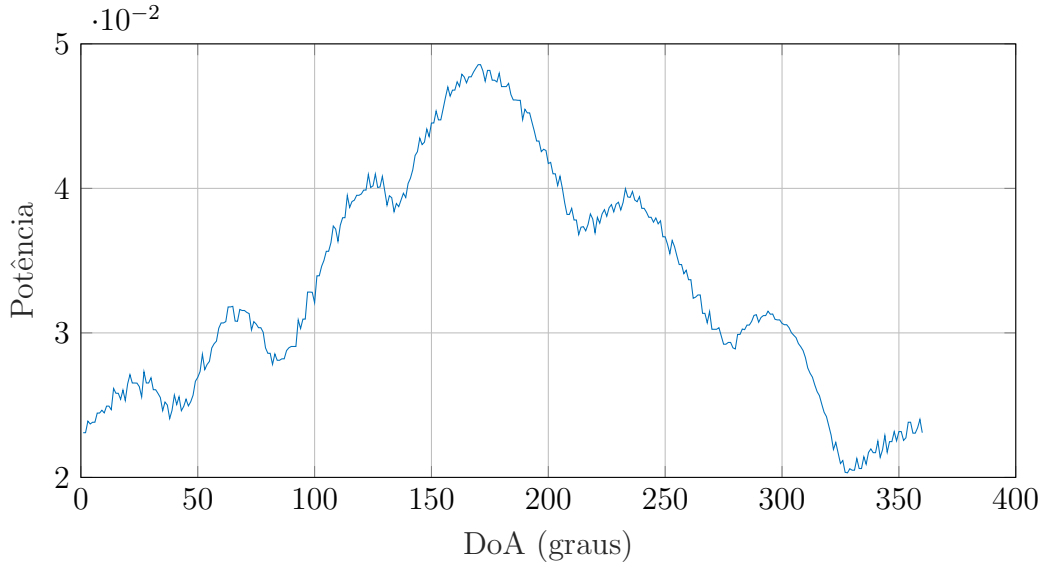

Figura 5.13: SRP-PHAT componente-base 1 após a alteração do espectrograma.

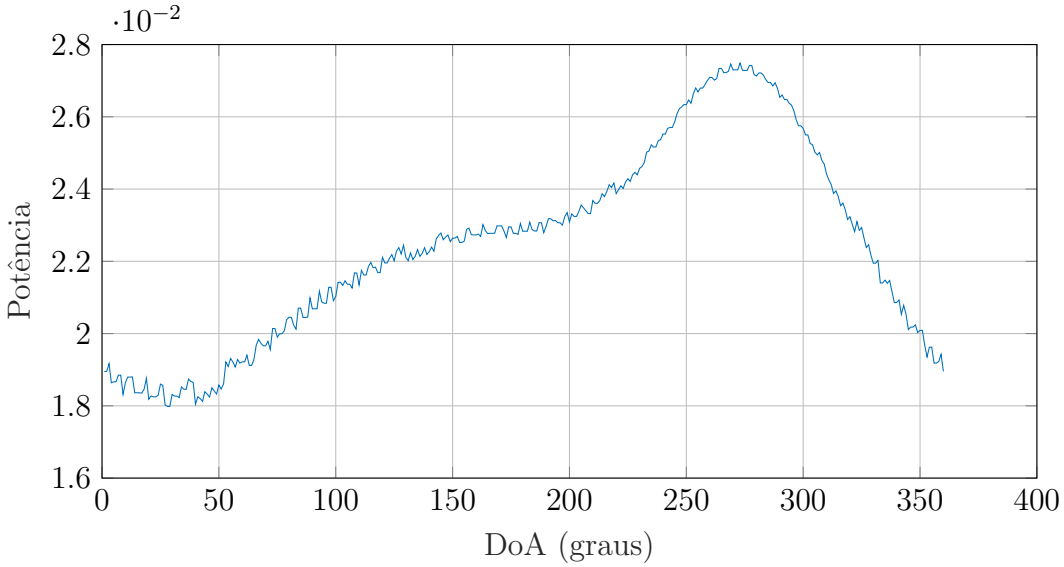

Figura 5.14: SRP-PHAT componente-base 3 após a alteração do espectrograma.

O algoritmo da NTF utilizando o método de agrupamento acima foi testado com misturas capturadas por uma arranjo de quatro microfones em salas com diferentes dimensões (por exemplo: 10 m ×10 m ×4 m, 2 m ×2 m ×2 m e 5 m ×5 m ×3 m) e características ambientais (paredes com diferentes tipos de materiais) como uma sala de estúdio e a sala de uma casa (todas as misturas são na realidade o resultados de simulações utilizando o *software CATT-Acoustic* e *Multivolver WCP* [37]). As misturas utilizadas nos testes eram compostas por diferentes instrumentos musicais (instrumentos percusivos como surdo e agogô) ou por conversas (utilizando as mesmas fontes originais utilizadas no Teste 2). Os testes foram repetidos variando-se o número de componentes-base utilizadas pelo algoritmo (de 10 até 1000), permitindo

assim avaliar o impacto deste parâmetro; todos os testes foram executados por centenas de iterações após a convergência da medida de erro do algoritmo (tipicamente até 1000 iterações para uma convergência próxima a 200 iterações).

Em nenhum dos testes realizados os resultados foram satisfatórios; no caso de testes com instrumentos percussivos foi possível reconstruir uma das fontes, porém com ainda clara presença da outra fonte; para os outros casos, não só não ocorria a separação, como também os sinais não eram reconstruídos com o mínimo de qualidade.

Os resultados encontrados indicam que a modelagem utilizada no desenvolvimento da NTF não é compatível com as dificuldades apresentadas pela separação de misturas que envolvam diversidade espacial. Fazendo uma analogia, a NMF só é capaz de modelar padrões espectrais constantes no tempo e na frequência; para contornar essa característica o algoritmo é tipicamente utilizado com um número relativamente elevado de componentes-base, de forma que padrões espectrais variantes no tempo ou na frequência possam vir a ser decompostos em múltiplos padrões constantes. Mesmo assim, foram desenvolvidas variantes do algoritmo que permitem melhor modelar padrões espectrais variantes no tempo ou frequência (NMFD) e até em ambos (NMF2D). Da mesma forma, a NTF precisaria representar também diferentes padrões gerados pelas caraterísticas ambientais implícitas a esse tipo de mistura, porém, não parece ser possível decompor os padrões gerados por estas características da mesma maneira que é feita pela NMF nesta analogia.

5.3.2 Determinação cega do número de fontes

No problema de separação cega de fontes, como já mencionado anteriormente, o objetivo é obter as fontes originais que compõem uma mistura sem dispor de informações *a priori* sobre as mesmas. Na prática, as diferentes abordagens tipicamente assumem determinados níveis de conhecimento sobre as fontes originais, como por exemplo características no tempo e/ou na frequência.

Uma informação que é assumida de forma quase universal nas diferentes referências utilizadas durante a realização deste trabalho, é a do número de fontes presentes na mistura. A Figura 5.12 da subseção anterior demostrou a capacidade do SRP-PHAT de destacar as principais direções de chegada percebidas por um arranjo de sensores. Considerando isso, o SRP-PHAT poderia ser utilizado como uma etapa inicial de processamento de qualquer algoritmo de separação cega multicanal, realizando uma determinação cega do número de fontes presentes na mistura.

O SRP-PHAT não é um algoritmo de determinação de número de fontes; para que ele cumpra esse papel é necessária a adoção de alguma heurística, como por exemplo a simples utilização de um limiar para determinar os picos relevantes.

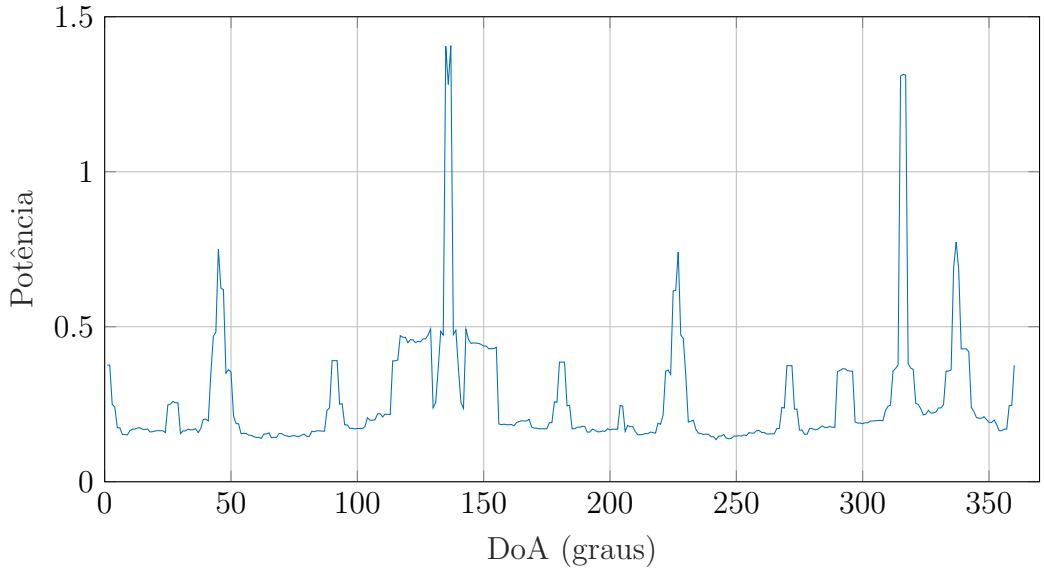

Figura 5.15: SRP-PHAT – 2 fontes (135° e 310°).

A Figura 5.15 (exatamente o mesmo exemplo apresentado na Figura 2.3) é um bom exemplo da execução do algoritmo SRP-PHAT; nela, percebem-se dois principais picos referentes às DoAs de 135° e 310° que correspondem justamente às posições das fontes presentes naquela mistura, e além destes picos existem também ao menos dois outros picos menores. Neste exemplo, se adotarmos um limiar de aproximadamente a metade do maior pico, o número de fontes será identificado corretamente.

A adoção de um limiar não deve ser genérica, sendo necessário observar ao menos três informações e seus impactos nos resultados do SRP-PHAT: resolução angular, resolução temporal, e a geometria do arranjo de sensores.

A resolução angular utilizada pelo algoritmo é basicamente o intervalo angular em que o mapeamento entre DoAs e atrasos é realizado. No exemplo da Figura 5.15 a resolução é de um grau. Uma consequência direta da escolha desta resolução é o intervalo angular mínimo entre duas fontes porém, na prática a escolha de um intervalo de 10° já implica um impacto significativo na determinação do número de fontes, como pode ser observado na Figura 5.16, que foi gerada a partir da mesma mistura que a Figura 5.15.

A resolução temporal utilizada pelo algoritmo é basicamente o menor intervalo de tempo considerado por ele, ou seja, apenas atrasos entre dois sensores maiores que este intervalo serão percebidos pelo algoritmo. Na prática este parâmetro está relacionado à taxa de amostragem da mistura e, o que é mais pertinente para este assunto, o tamanho da DFT utilizada. A Figura 5.17 apresenta o resultado da execução do algoritmo com uma DFT de tamanho 256 (utilizando tamanhos 512, 1024 os resultados são praticamente idênticos aos da Figura 5.15, que utilizou uma

DFT com tamanho 2048), onde é mais difícil identificar apenas as duas direções reais presentes na mistura.

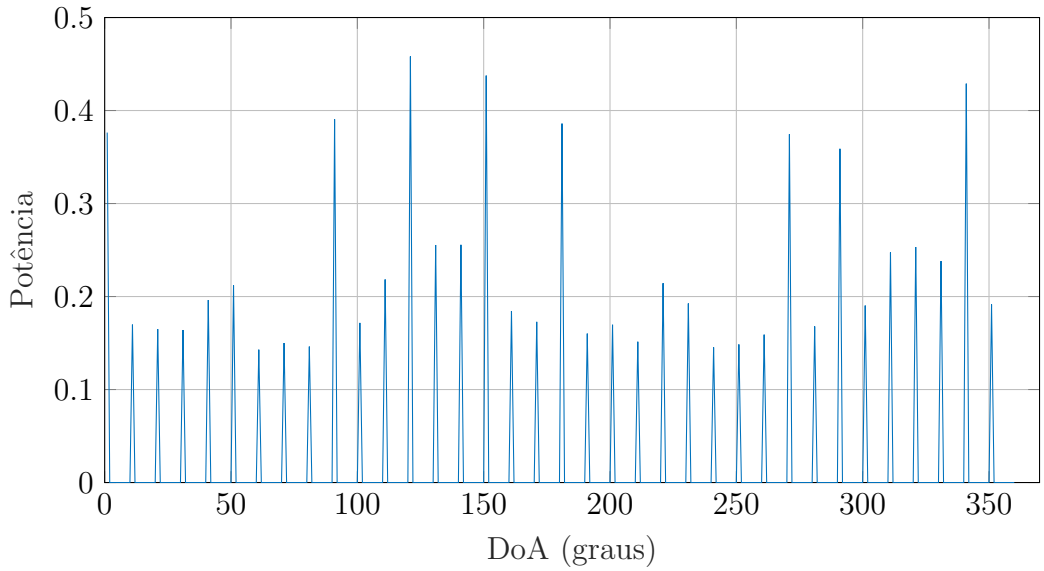

Figura 5.16: SRP-PHAT – resolução angular 10°.

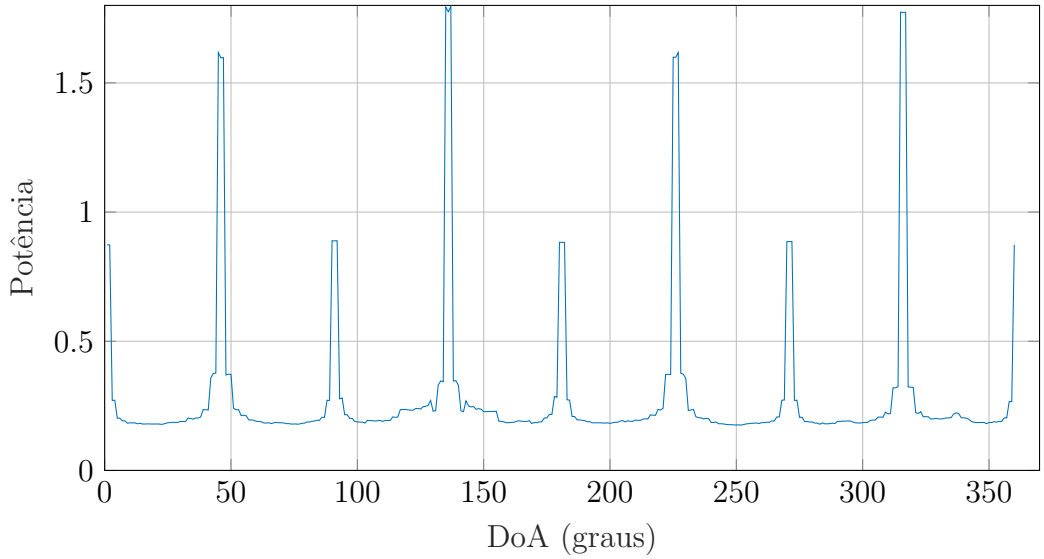

Figura 5.17: SRP-PHAT – Tamanho DFT 256.

A geometria do arranjo de sensores é um parâmetro importante, pois ela determinará as direções que o arranjo será capaz de identificar corretamente; especificamente, cada par de sensores é incapaz de determinar univocamente a direção de fontes que estejam alinhadas ao eixo de simetria do par. A Figura 5.18 apresenta um exemplo onde um arranjo de sensores simétrico é incapaz de identificar as direções verdadeiras, neste caso 45° e 135°.

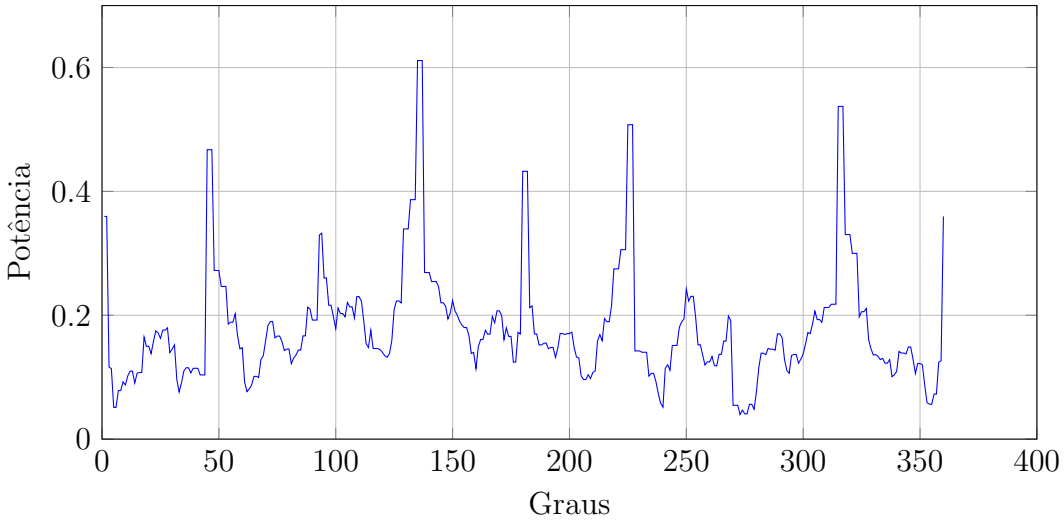

Figura 5.18: SRP-PHAT — fontes no eixo de simetria do arranjo

5.4 Considerações Práticas

As alterações propostas neste capítulo, com a exceção do uso da SRP-PHAT, não alteram significativamente a complexidade do algoritmo ou suas etapas de pós-processamento, representando menos de 1% do tempo de execução do algoritmo nas implementações aqui descritas. A execução do SRP-PHAT adiciona uma etapa de custo computacional significativo, particularmente se ele for calculado para todos os quadros; mesmo assim, para condições similares às descritas para o NMF-SCM, esta etapa adiciona algo como de 5 a 10 minutos à execução do algoritmo, o que pode representar até 300% do tempo de execução do algoritmo.

O SRP-PHAT se mostrou uma ferramenta interessante para a determinação cega do número de fontes presentes em uma mistura. O uso correto desta ferramenta, porém, requer uma breve análise para obtenção de resultados mais confiáveis; dito isso, dada uma geometria do arranjo de sensores que mitigue problemas de simetria, pode-se sugerir, ao menos para sinais sonoros típicos, o uso de uma resolução angular de 1°, um tamanho de DFT de 2048 e um limiar equivalente à metade do maior pico observado para a determinação do número de fontes presentes em uma mistura.

Capítulo 6

Conclusões e trabalhos futuros

Este trabalho estudou a separação multicanal de fontes sonoras utilizando algoritmos baseados na fatoração não-negativa de matrizes e tensores não-negativos. Além de um estudo sobre alguns algoritmos que utilizam apenas um canal, como a NMF e algumas de suas extensões, e etapas de pré/pós-processamento relacionadas a estes algoritmos, este trabalho abordou dois algoritmos de separação multicanal diferentes, a NMF-SCM e a NTF.

O Capítulo 3 apresentou a NMF-SCM, algoritmo este que apresenta uma das mais completas modelagens do problema de separação multicanal de fontes sonoras, utilizando diversidade espacial, direção de chegada (com uma etapa de inicialização que utiliza SRP-PHAT) e a não-negatividade da representação (mesmo o algoritmo não utilizando apenas uma representação não-negativa). Associada a essa modelagem completa está uma implementação computacionalmente complexa que motivou o desenvolvimento, para este trabalho, de uma implementação parcialmente vetorizada, e posteriormente a paralelização desta, que atingiram significativas reduções no tempo de execução do algoritmo. Em todos os testes realizados com as três implementações, porém, encontraram-se problemas de instabilidade numérica (confirmando o que também foi constatado em [30]).

O Capítulo 4 apresentou a NTF. Em contraste com a modelagem proposta na NMF-SCM, a NTF apresenta uma modelagem simplificada que utiliza apenas a não-negatividade da representação das fontes sonoras e as diferenças de intensidade de cada fonte nos diferentes canais. Por analogia, a NTF pode ser considerada como a NMF para o caso multicanal. Assim como para suas modelagens, também existe um contraste claro entre a complexidade computacional da NTF e a da NMF-SCM: não só o algoritmo da NTF apresenta menor complexidade como também é relativamente mais simples obter uma implementação vetorizada para ele. Essa simplicidade talvez seja uma fator determinante para o seu continuado estudo, como apresentado em [30].

O Capítulo 5 apresentou alterações e extensões ao algoritmo de separação uti-

lizando a NTF, visando a melhorar a qualidade dos sinais obtidos, estender o algoritmo para o caso de múltiplos canais e aproveitar a diversidade espacial potencialmente presente nos diferentes canais. Dentre essas mudanças, a utilização da diversidade espacial, potencialmente a mais interessante, se mostrou ineficaz. A modelagem simplificada utilizada pela NTF não parece ser capaz de desfazer as interações entre as fontes e o ambiente, ao menos com o método de agrupamento de componentes-base proposto. Por outro lado a proposta da utilização do algoritmo SRP-PHAT para a determinação cega do número de fontes se mostrou promissora, tornando, ao menos em alguns casos, desnecessária para algoritmos de separação cega a informação do número de fontes.

Com base no que foi estudado e desenvolvido para este trabalho, aparecem algumas vertentes naturais para trabalhos futuros:

- Nova implementação da NMF-SCM que reduza ainda mais o tempo de execução do algoritmo (por exemplo, utilizando uma unidade de processamento gráfico de propósito geral – GPGPU), que não só tornaria sua utilização mais interessante, como também tonaria mais prática a depuração de problemas. Vale ressaltar, porém, que com a exceção da atualização em paralelo dos diferentes parâmetros, toda redução no tempo de execução obtida na implementação realizada para este trabalho veio acompanhada de um aumento da dificuldade de compreensão da mesma, tornando mais complicada a depuração de problemas.

- Estudo da modelagem utilizada pela NMF-SCM, possivelmente aproveitando alguns de seus conceitos (assim como realizado no Capítulo 5) como, por exemplo, a matriz $\mathbf{M}$, que compõe a parte da modelagem que se refere ao espectrograma de magnitudes das fontes, que permite incorporar o agrupamento de componentes-base ao processo de otimização, prescindindo de uma etapa de pós-processamento (passo 5 do algoritmo da NTF e também necessária para a NMF).

- Obtenção de uma base de dados representativa de misturas capturadas por um arranjo de microfones para permitir a avaliação da efetividade da determinação cega do número de fontes utilizando o SRP-PHAT e a heurística proposta no Capítulo 5.

De modo geral, este trabalho apresentou dois extremos do vasto campo de separação multicanal de fontes sonoras. Primeiramente a NMF-SCM, um algoritmo do estado da arte com uma modelagem completa associada e uma demasiado grande complexidade computacional; do outro lado a NTF, talvez um dos primeiros algoritmos a tratar esse problema, com uma modelagem simples e mínima complexidade

computacional. Talvez o estudo destes dois extremos venha a servir como uma introdução satisfatória que permita a exploração de implementações mais simples da modelagem apresentada com a NMF-SCM ou outras alterações à NTF que possibilitem que ela lide melhor com misturas obtidas por arranjos de microfones.

Não por acaso, a referência [30] (cujos trechos tratando da NMF-SCM e da NTF já foram citados neste trabalho), propõe utilizar a modelagem proposta para a NMF-SCM em conjunto com a NTF através da utilização de uma representação no domínio espaço-frequência (do inglês *Spatial Frequency domain*).

Referências Bibliográficas

[1] CHERRY, C. E. "Some experiments on the recognition of speech", *The Journal of The Acoustical Society of America*, v. 25, n. 5, pp. 975–979, maio 1953.

[2] A. HYVÄRINEN, J. KARHUNEN, E. O. *Independent Component Analysis*. Helsinque, Finlândia, John Wiley and Sons, 2001.

[3] GILLIS, N. "The why and how of nonnegative matrix factorization". In: Suykens, J. A., Signoretto, M., Argyriou, A. (Eds.), *Regularization, Optimization, Kernels, and Support Vector Machines*, Machine Learning & Pattern Recognition Series, Chapman and Hall/CRC, cap. 12, pp. 257–291, Boca Raton, EUA, 2014.

[4] OMLOR, L., GIESE, M. "Blind source separation for over-determined delayed mixtures". In: *Advances in Neural Information Processing Systems 19: Proceedings of the 20th Annual Conference on Neural Information Procesing (NIPS)*, pp. 1049–1056, Vancouver, Canadá, dezembro 2006.

[5] SMARAGDIS, P. "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs". In: Carlos, G. P., Albert, P. (Eds.), *Lecture Notes in Computer Science*, v. 3195, Springer, pp. 494–499, Cambridge, EUA, 2004.

[6] OLSHAUSEN, B. A., FIELDT, D. J. "Sparse coding with an overcomplete basis set: a strategy employed by V1", *Vision Research*, v. 37, n. 23, pp. 3311–3325, dezembro 1997.

[7] JUTTEN, C., COMON, P. *Handbook of Blind Source Separation*. Oxford, Reino Unido, Academic Press, 2010.

[8] ALPAYDIN, E. *Introduction to Machine Learning*. Londres, Inglaterra, The MIT Press, 2004.

[9] KITAMURA, D., SARUWATARI, H., NAKAMURA, S., et al. "Hybrid multichannel signal separation using supervised nonnegative matrix factorization with spectrogram restoration". In: *Proceedings of the 2014 Asia-*

Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1–10, Siem Reap, Camboja, dezembro 2014. APSIPA.

[10] FITZGERALD, D., CRANITCH, M., COYLE, E. "Non-negative tensor factorisation for sound source separation". In: *Proceedings of the 16th IEE Irish Signals and Systems Conference (ISSC)*, pp. 1–5, Dublin, Irlanda, julho 2005. IEE.

[11] NIKUNEN, J., VIRTANEN, T. "Multichannel audio separation by direction of arrival based spatial covariance model and non-negative matrix factorization". In: *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6677–6681, Florença, Itália, maio 2014. IEEE.

[12] JOSEPH, H. D., HARVEY, F., S., BRANDSTEIN, M. S. "Robust localization in reverberant rooms". In: Michael, B., Darren, W. (Eds.), *Microphone Arrays*, Springer, cap. 8, pp. 157–178, Berlim, Alemanha, 2001.

[13] CAMPOS, C. V. C. *Algoritmos para Reconstruções da Fase de Sinais de Áudio*. Projeto de graduação, Universidade Federal do Rio de Janeiro, Escola Politécnica, Rio de Janeiro, Brazil, 2011.

[14] GRIFFIN, D., LIM, J. "Signal estimation from modified short-time Fourier transform", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 32, n. 2, pp. 236–243, April 1984.

[15] BEAUREGARD, G. T., ZHU, X., WYSE, L. "An Efficient Algorithm For Real Time Spectrogram Inversion". In: *Proceedings of the 8th international Conference on Digital Audio Effects (DAFx)*, pp. 116–121, Madri, Espanha, setembro 2005.

[16] ZHU, X., BEAUREGARD, G. T., WYSE, L. L. "Real-Time Signal Estimation From Modified Short-Time Fourier Transform Magnitude Spectra", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 15, n. 5, pp. 1645–1653, July 2007.

[17] GUNAWAN, D., SEN, D. "Music source separation synthesis using multiple input spectrogram inversion". In: *Proceedings of the 11th IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–5, Rio de Janeiro, Brazil, outubro 2009. IEEE.

[18] VINCENT, E., GRIBONVAL, R., FÉVOTTE, C. "Performance measurement in blind audio source separation", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 14, n. 4, pp. 1462–1469, julho 2006.

[19] HYVARINEN, A. "Fast and robust fixed-point algorithms for independent component analysis", *IEEE Transactions on Neural Networks*, v. 10, n. 3, pp. 626–634, maio 1999.

[20] DAVIES, M., JAMES, C. "Source separation using single channel ICA", *Signal Processing*, v. 87, n. 8, pp. 1819–1832, agosto 2007.

[21] LEE, D. D., SEUNG, H. S. "Algorithms for non-negative matrix factorization". In: *Advances in Neural Information Processing Systems 13: Proceedings of the 14th Annual Conference on Neural Information Procesing (NIPS)*, pp. 556–562, Pittsburgh, EUA, novembro 2000.

[22] CICHOCKI, A., ZDUNEK, R., ICHI AMARI, S. "Csiszár's divergences for non-negative matrix factorization: Family of new algorithms". In: *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, pp. 32–39, Charleston, EUA, março 2006.

[23] CICHOCKI, A., ZDUNEK, R., AMARI, S.-I. "New Algorithms for Non-Negative Matrix Factorization in Applications to Blind Source Separation". In: *Proceedings of the 2006 IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, v. 5, pp. 621–625, Toulouse, França, maio 2006. IEEE.

[24] SCHMIDT, M., MØRUP, M. "Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation". In: *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, pp. 700–707, Charleston, EUA, março 2006.

[25] TYGEL, A. F. *Métodos de Fatoração de Matrizes Não-Negativas Para Separação de Sinais Musicais*. Dissertação de mestrado, Universidade Federal do Rio de Janeiro – PEE/COPPE, Rio de Janeiro, Brasil, 2009.

[26] ALMEIDA, R. M. *Separação de Fontes Sonoras Por Fatoração Duplamente Deconvolutiva de Matrizes Não-Negativas com Uso de Restrições*. Dissertação de mestrado, Universidade Federal do Rio de Janeiro – PEE/COPPE, Rio de Janeiro, Brasil, 2014.

[27] NIKUNEN, J., VIRTANEN, T. "Direction of Arrival Based Spatial Covariance Model for Blind Sound Source Separation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 22, n. 3, pp. 727–739, March 2014.

[28] SAWADA, H., KAMEOKA, H., ARAKI, S., et al. "New formulations and efficient algorithms for multichannel NMF". In: *Proceedings of the 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 153–156, New Paltz, EUA, outubro 2011. IEEE.

[29] SAWADA, H., ARAKI, S., MUKAI, R., et al. "Grouping Separated Frequency Components by Estimating Propagation Model Parameters in Frequency-Domain Blind Source Separation", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 15, n. 5, pp. 1592–1604, julho 2007.

[30] MITSUFUJI, Y., KOYAMA, S., SARUWATARI, H. "Multichannel blind source separation based on non-negative tensor factorization in wavenumber domain". In: *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 56–60, Xanghai, China, março 2016. IEEE.

[31] SHASHUA, A., HAZAN, T. "Non-negative tensor factorization with applications to statistics and computer vision". In: *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 792–799, Bonn, Alemanha, agosto 2005. IMLS.

[32] DHILLON, I. S., SRA, S. "Generalized Nonnegative Matrix Approximations with Bregman Divergences". In: *Advances in Neural Information Processing Systems 18: Proceedings of the 19th Annual Conference on Neural Information Processing (NIPS)*, p. 283–290, Vancouver, Canadá, 2005.

[33] BARRY, D., LAWLOR, B., COYLE, E. "Sound Source Separation: Azimuth Discrimination and Resynthesis". In: *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx)*, pp. 1–5, Nápoles, Itália, outubro 2004.

[34] STURMEL, N., DAUDET, L. "Signal reconstruction from STFT magnitude: A state of the art". In: *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*, pp. 375–386, Paris, França, setembro 2011.

[35] TYGEL, A., BISCAINHO, L. W. P. "Sound source separation via nonnegative matrix factor 2-D deconvolution using linearly sampled spectrum". In:

Anais do 7o. Congresso de Engenharia de Áudio, pp. 58–65, São Paulo, Brasil, maio 2009. Audio Engineering Society – Seção Brasil.

[36] HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd ed. Upper Saddle River, EUA, Prentice Hall PTR, 1998.

[37] "CATT-Acoustic with Multivolver WCP, (v9.1a)". 2017. Disponível em: <http://www.catt.se/>.