



Universidade Federal  
do Rio de Janeiro  

---

Escola Politécnica

SEPARAÇÃO E ISOLAMENTO NÃO-SUPERVISIONADOS DE  
INSTRUMENTOS HARMÔNICOS MONOFÔNICOS USANDO O  
CONCEITO DE ESTRUTURA HARMÔNICA MÉDIA

Carlos Pedro Vianna Lordelo

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Luiz Wagner Pereira Biscainho

Rio de Janeiro, Brasil

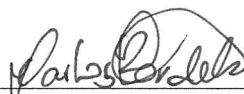
Setembro de 2017

SEPARAÇÃO E ISOLAMENTO NÃO-SUPERVISIONADOS DE  
INSTRUMENTOS HARMÔNICOS MONOFÔNICOS USANDO O  
CONCEITO DE ESTRUTURA HARMÔNICA MÉDIA

Carlos Pedro Vianna Lordelo

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO  
DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA PO-  
LITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO  
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU  
DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO

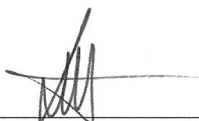
Autor:



---

Carlos Pedro Vianna Lordelo

Orientador:



---

Prof. Luiz Wagner Pereira Biscainho, D. Sc.

Examinador:



---

Prof. Diego Barreto Haddad, D. Sc.

Examinador:



---

Prof. Wallace Alves Martins, D. Sc.

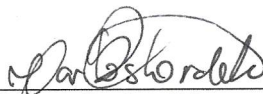
Rio de Janeiro, Brasil

Setembro de 2017

## Declaração de Autoria e de Direitos

Eu, *Carlos Pedro Vianna Lordelo*, CPF 143.373.267-07, autor da monografia *Separação e Isolamento Não-supervisionados de Instrumentos Harmônicos Monofônicos Usando o Conceito de Estrutura Harmônica Média*, subscrevo para os devidos fins, as seguintes informações:

1. O autor declara que o trabalho apresentado na disciplina de Projeto de Graduação da Escola Politécnica da UFRJ é de sua autoria, sendo original em forma e conteúdo.
2. Excetua-se do item 1. eventuais transcrições de texto, figuras, tabelas, conceitos e idéias, que identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos proprietários, quando necessárias.
3. O autor permite que a UFRJ, por um prazo indeterminado, efetue em qualquer mídia de divulgação, a publicação do trabalho acadêmico em sua totalidade, ou em parte. Essa autorização não envolve ônus de qualquer natureza à UFRJ, ou aos seus representantes.
4. O autor pode, excepcionalmente, encaminhar à Comissão de Projeto de Graduação, a não divulgação do material, por um prazo máximo de 01 (um) ano, improrrogável, a contar da data de defesa, desde que o pedido seja justificado, e solicitado antecipadamente, por escrito, à Congregação da Escola Politécnica.
5. O autor declara, ainda, ter a capacidade jurídica para a prática do presente ato, assim como ter conhecimento do teor da presente Declaração, estando ciente das sanções e punições legais, no que tange a cópia parcial, ou total, de obra intelectual, o que se configura como violação do direito autoral previsto no Código Penal Brasileiro no art.184 e art.299, bem como na Lei 9.610.
6. O autor é o único responsável pelo conteúdo apresentado nos trabalhos acadêmicos publicados, não cabendo à UFRJ, aos seus representantes, ou ao(s) orientador(es), qualquer responsabilização/ indenização nesse sentido.
7. Por ser verdade, firmo a presente declaração.



---

Carlos Pedro Vianna Lordelo

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica - Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro - RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es).

## AGRADECIMENTOS

Enfim, agradecimentos... Apesar de ainda não acreditar que o projeto tenha sido executado com sucesso depois de tantos anos e percalços no caminho, sim, chegou a hora de agradecer a todos que de alguma forma contribuíram para que este texto tenha finalmente tomado forma.

Sem sombra de dúvidas, uma das pessoas mais importantes nos últimos anos foi meu professor e amigo, Luiz Wagner. Obrigado pela sua confiança e perseverança no meu trabalho; sem você, provavelmente ainda estaria perdido, sem saber como finalizar a minha graduação e dar início à minha carreira profissional como engenheiro. Suas conversas e conselhos foram importantíssimos para me fazer seguir em frente em momentos difíceis de desmotivação e indecisão. Além disso, não poderia deixar de mencionar que suas ideias criativas e revisões minuciosas do texto foram fundamentais para o desenvolvimento deste projeto.

Agradeço aos amigos de longa data, principalmente, PV, Cianci, Jamil, Marcelo, Valter, Brilhante, Maju e João que, desde o ensino médio (alguns antes) me acompanham e ainda conseguem me descontraír em tempos de preocupação e tensão. Aos meus companheiros de UFRJ que me ajudaram nas matérias, nas provas e nos complicados trabalhos no decorrer do curso. Léo, Gustavo, Petraglia, Fonini, Lucas, Gabriel, Hélio, Baiano, Julio, Henrique e Michel, espero que continuemos nossa amizade por muito tempo.

Aos meus pais pelos exatos 26 anos de amor e carinho incondicionais e pelos 11 anos de muito sacrifício. Realizar toda a minha formação básica no Colégio de São Bento só foi possível graças a vocês.

Devo agradecer também a uma pessoa muito especial na minha vida. Carol, obrigado por praticamente tudo. É você a principal responsável por eu nunca ter me sentido sozinho ou desamparado, tanto emocionalmente quanto fisicamente. Sua companhia diária tem sido indispensável nos últimos anos, é com você que compartilho minhas angústias e minhas alegrias sistematicamente.

Não menos importantes são meus agradecimentos ao povo brasileiro que contribuiu de forma significativa à minha formação e estada nesta Universidade. Este projeto é uma pequena forma de retribuir o investimento e confiança em mim depo-

sitados.

Por último, antes de dar início ao trabalho propriamente dito, gostaria de dizer que todas as experiências, conversas, trocas e todos aqueles que passaram pela minha vida de alguma maneira e porventura não foram citados, também foram fundamentais direta ou indiretamente para a conclusão desta etapa que, aliás, não termina aqui, apenas se inicia.

## RESUMO

Apesar de existirem diversas maneiras de se tratar o problema da separação de fontes, a grande maioria dos métodos existentes podem ser inseridos em 3 grandes categorias: métodos baseados em independência, em esparsidade ou em não-negatividade. Alternativamente, este projeto aborda o tema sob uma perspectiva diferente encontrada na literatura, no contexto de processamento digital de áudio. A separação é realizada a partir da modelagem do timbre de um instrumento musical harmônico usando o conceito de estrutura harmônica média, que é o perfil espectral das notas por ele emitidas.

Dado apenas o número total de instrumentos harmônicos, a estrutura harmônica média correspondente a cada um deles é aprendida diretamente do sinal misturado e pode ser utilizada para extrair suas emissões sonoras. Além disso, como produto secundário, o método implementado consegue gerar estimativas para os *pitches* de cada nota presente na mistura e, portanto, ele pode também ser utilizado em outros tipos de aplicação.

Os resultados das simulações realizadas com instrumentos sintéticos e naturais mostram que o método pode ser utilizado para separar fontes harmônicas entre si ou fontes harmônicas de não-harmônicas.

Nesse sentido, este trabalho explica todos os fundamentos teóricos por trás do método e fornece as informações detalhadas para reproduzi-lo, com especial atenção aos algoritmos elaborados durante sua implementação.

Palavras-Chave: processamento digital de áudio, estrutura harmônica, separação de fontes, NMF, ICA, MPE, estimação de *pitch*.

## ABSTRACT

Although there are a lot of ways to deal with the problem of source separation, the great majority of the existent methods fall into 3 big categories: methods based on independence, on sparsity, or on non-negativity. Alternatively, this project approaches the theme from a different perspective found in the literature, in the area of digital audio processing. The separation is based on a model for the timbre of a harmonic musical instrument in the form of an average harmonic structure, which can be seen as the spectral profile of its emitted notes.

Given only the total number of harmonic instruments, the average harmonic structure corresponding to each of them is directly learned from the mixed signal and can be used to extract their sound emissions. Moreover, as a by-product, the implemented method estimates the pitch of each note in the mixture and, therefore, it can also be used in other types of applications.

The results of the simulations performed with synthetic and natural instruments show that the method can be used to separate harmonic sources from each other as well as harmonic from inharmonic sources.

In this sense, this work explains the theoretical foundations behind the method and provides detailed information to reproduce it, with special attention to the algorithms designed during its implementation.

Key-words: digital audio processing, harmonic structure, source separation, NMF, ICA, MPE, pitch estimation.



# Lista de Figuras

1.1	Diagrama de blocos do processo de mistura. . . . .	3
2.1	Representação de um sinal de áudio no domínio do tempo . . . . .	9
2.2	Representação de um sinal de áudio no domínio da frequência . . . . .	9
2.3	Espectrograma de magnitude de um sinal de áudio . . . . .	11
2.4	Vibrações naturais de uma corda em um instrumento de cordas . . . . .	12
2.5	Espectro de magnitude de diferentes notas de um flautim . . . . .	14
2.6	Estrutura harmônica média calculadas para alguns instrumentos co- muns. . . . .	18
2.7	Observações e modelagem estatística de um processo estocástico . . . . .	25
2.8	Comparação de ajustes polinomiais de diferentes ordens às observações de um fenômeno estocástico . . . . .	27
3.1	Diagrama de blocos do método implementado. . . . .	29
3.2	Comparação do resultado do algoritmo de detecção de picos em dife- rentes quadros de um sinal com duas fontes misturadas. . . . .	33
3.3	Exemplo de clusterização com a criação de arestas falsas. . . . .	48
3.4	Exemplo de funcionamento do algoritmo de agrupamento NK em $\mathbb{R}^2$ . . . . .	51
3.5	Diferenças dos resultados das duas etapas de extração de harmônicos. . . . .	58
4.1	Comparação das AHS estimadas pelo projeto e por [1] para o experi- mento #1. . . . .	63
4.2	Comparação dos resultados do algoritmo de MPE no experimento #1. . . . .	65
4.3	Comparação das AHS estimadas pelo projeto e por [1] para o experi- mento #2. . . . .	67
4.4	Comparação dos resultados do algoritmo de MPE para o experimento #2. . . . .	68

4.5	Comparação das estimativas para a AHS do oboé e dos resultados do algoritmo de MPE no experimento #3. . . . .	69
4.6	Comparação das estimativas para a AHS do flautim e dos resultados do algoritmo de MPE no experimento #4. . . . .	71
4.7	Comparação das estimativas para a AHS do órgão e dos resultados do algoritmo de MPE no experimento #5. . . . .	72

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Separação Cega de Fontes . . . . .	2
1.2	Métodos para Separação de Fontes . . . . .	2
1.2.1	Separação Usando Independência das Fontes . . . . .	4
1.2.2	Separação Usando Esparsidade das Fontes . . . . .	4
1.2.3	Separação Usando Não-negatividade . . . . .	5
1.3	Método Alternativo . . . . .	6
1.4	Objetivo . . . . .	7
1.5	Organização . . . . .	7
<b>2</b>	<b>Fundamentos Teóricos</b>	<b>8</b>
2.1	Espectrograma de Magnitude . . . . .	8
2.2	Frequências Fundamental e Harmônicas . . . . .	11
2.3	Estrutura Harmônica Média . . . . .	13
2.4	Escala de Frequência MIDI . . . . .	17
2.5	Estimação por Máxima Verossimilhança . . . . .	20
2.6	<i>Overfitting</i> e Critério de Informação Bayesiano . . . . .	24
<b>3</b>	<b>Separação de Fontes usando Estruturas Harmônicas Médias</b>	<b>28</b>
3.1	Pré-Processamento . . . . .	30
3.2	Modelagem . . . . .	30
3.2.1	Detecção de Picos . . . . .	32
3.2.2	Estimação Primária das Frequências Fundamentais . . . . .	34
3.2.3	Extração de Harmônicos e Criação das HS . . . . .	41
3.2.4	Agrupamento para Estimação das AHS . . . . .	43

3.3	Separação . . . . .	52
3.3.1	Estimação Secundária das Frequências Fundamentais . . . . .	52
3.3.2	Extração Retificada dos Harmônicos . . . . .	56
3.3.3	Reconstrução dos Sinais das Fontes . . . . .	59
<b>4</b>	<b>Testes e Resultados</b>	<b>60</b>
4.1	Sinais de Teste . . . . .	60
4.2	Avaliação Geral dos Experimentos . . . . .	61
4.3	Resultados . . . . .	62
4.3.1	Experimento #1 - FlautimOrgao.wav . . . . .	62
4.3.2	Experimento #2 - EufonioOboe.wav . . . . .	65
4.3.3	Experimento #3 - OboeFem.wav . . . . .	68
4.3.4	Experimento #4 e Experimento #5 . . . . .	69
<b>5</b>	<b>Conclusão</b>	<b>73</b>

# Capítulo 1

## Introdução

Atualmente, a separação de fontes está presente nas mais diversas áreas do conhecimento [2]. Na área de processamento de imagens, por exemplo, métodos dessa natureza são usados para a retirada de defeitos indesejados em imagens de satélite. Ademais, em telecomunicações, eles podem ser utilizados para retirar informações úteis de um dos diversos sinais que chegam a uma rede sem fio. Outra aplicação muito importante pode ser encontrada até mesmo na área da biologia, onde algoritmos de separação de fontes são utilizados em eletroencefalogramas para separar os sinais provenientes de diferentes ações executadas ao mesmo tempo, tais como o piscar de olhos e movimentos da língua ou de membros.

Este projeto aborda o problema da separação de fontes aplicada a sinais musicais. Mais especificamente, o projeto estuda um método alternativo encontrado na literatura para a extração das emissões sonoras provenientes de cada instrumento musical harmônico presente em um sinal de áudio composto. Algoritmos dessa natureza encontram grande aplicação na indústria da música, pois podem ser usados em estúdio para, por exemplo, transformar uma gravação mono em estéreo, substituir um cantor ou instrumentista por outro, modificar o balanço ou o arranjo original de uma gravação, etc.

Neste capítulo, o leitor terá uma visão geral acerca do tema da separação de fontes, com resumos dos métodos mais comuns na literatura para resolver o problema. Em seguida, será feita uma breve introdução a um método que motivou e tornou-se o foco principal deste projeto. Por último, os objetivos do projeto serão apresentados, e a organização deste texto será detalhada.

## 1.1 Separação Cega de Fontes

Geralmente, o termo separação cega de sinais (em inglês *Blind Source Separation*, BSS) [3] é utilizado na literatura para indicar que o algoritmo de separação em questão não utiliza nenhuma informação acerca dos sinais presentes na mistura. No entanto, não é bem isso o que acontece na maioria dos artigos a respeito do tema. Na verdade, alguma informação sobre o sinal é quase sempre utilizada, mesmo que indiretamente. Um exemplo é o próprio fato de saber que estamos processando sinais musicais, o que já nos traz informação a respeito do formato e da banda espectral dos sinais analisados. Na verdade, um termo mais preciso para definir um tipo de processamento que não utiliza amostras das fontes *a priori* seria “não-supervisionado”, em vez de “cego”.

Não se pode dizer, portanto, que o método aqui analisado realiza uma separação cega de fontes, já que a informação de que os sinais são musicais é utilizada. Entretanto, pode-se afirmar que o algoritmo é, sim, não-supervisionado, pois não é realizado nenhum tipo de análise dos instrumentos previamente.

É de se notar que, embora seja uma tarefa simples para a cognição humana, que a realiza facilmente quando ouvimos uma música, por exemplo, a segregação do sinal de um instrumento qualquer a partir de um sinal musical polifônico ainda é um problema muito complicado para as máquinas. Vários algoritmos continuam sendo propostos para tentar encontrar a solução desejada; porém, até o momento, esse ainda é um problema sem uma solução genérica robusta.

## 1.2 Métodos para Separação de Fontes

Para entender como a separação é realizada, faz-se necessária uma apresentação matemática do tema. Nesse sentido, considere que  $R$  fontes ( $s_1, s_2, s_3, \dots, s_R$ ) emitem sinais que são posteriormente captados por  $M$  sensores, dando origem às misturas ( $x_1, x_2, x_3, \dots, x_M$ ). O problema da separação não-supervisionada de fontes é simplesmente tentar descobrir quais são os valores de  $s_j$ , para  $j \in \{1, 2, \dots, R\}$  a partir somente da análise dos sinais misturados,  $x_i$ , para  $i \in \{1, 2, \dots, M\}$ . Pode-se inferir ainda, como subproduto do processamento, de que maneira as fontes foram misturadas.

Matematicamente, o problema na sua formulação instantânea pode ser descrito da seguinte forma:

$$\begin{array}{rclclclclcl}
x_1 & = & a_{11}s_1 & + & a_{12}s_2 & + & a_{13}s_3 & + & \dots & + & a_{1R}s_R \\
x_2 & = & a_{21}s_1 & + & a_{22}s_2 & + & a_{23}s_3 & + & \dots & + & a_{2R}s_R \\
x_3 & = & a_{31}s_1 & + & a_{32}s_2 & + & a_{33}s_3 & + & \dots & + & a_{3R}s_R \\
\vdots & & \vdots & & \vdots & & \vdots & & & & \vdots \\
x_M & = & a_{M1}s_1 & + & a_{M2}s_2 & + & a_{M3}s_3 & + & \dots & + & a_{MR}s_R,
\end{array}$$

onde cada termo  $a_{ij}$  é o peso dado à fonte  $s_j$  na mistura  $x_i$ . Podemos re-escrevê-lo na forma matricial como

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{1.1}$$

onde  $\mathbf{A} \in \mathbb{R}^{M \times R}$  contém os pesos da mistura,  $\mathbf{s} \in \mathbb{R}^{R \times 1}$  contém os sinais emitidos pelas fontes, ambos desconhecidos, e  $\mathbf{x} \in \mathbb{R}^{M \times 1}$  é o vetor com as diferentes misturas captadas pelos sensores. A Figura 1.1 ilustra a relação descrita na Equação (1.1).

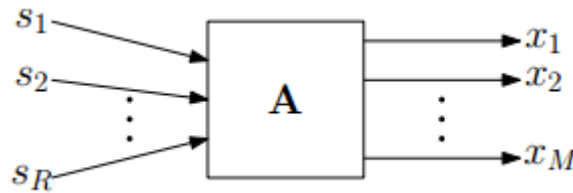


Figura 1.1: Diagrama de blocos do processo de mistura. Adaptado de [4].

Portanto, o problema de separação de fontes é justamente tentar descobrir os valores originais do vetor  $\mathbf{s}$ . Em primeira análise, um caminho para a solução seria utilizar mínimos quadrados no sistema linear  $\mathbf{A}\mathbf{s} = \mathbf{x}$  e resolvê-lo. No entanto, é exatamente aqui que entra o desafio: não sabemos quais são os coeficientes da matriz  $\mathbf{A}$ , pois não temos informação do sistema físico pelo qual o processo passou. Por isso, o problema passa a ser muito mais complicado.

Em resumo, é preciso estimar a matriz  $\mathbf{A}^{-1} \approx \mathbf{W}$  de alguma maneira tal que  $\mathbf{s} \approx \hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$ . É trivial perceber que o sistema não tem solução caso a matriz  $\mathbf{A}$  não tenha posto completo por colunas. Mesmo assim, nos últimos anos surgiram vários métodos, aplicáveis às mais diversas situações, cujo intuito é obter uma boa

representação dos sinais separados  $\hat{\mathbf{s}}$ .

Entretanto, para que as estimativas sejam razoáveis, é necessário impor algumas restrições sobre as fontes. Tais restrições dividem os métodos de separação em 3 grandes tipos: métodos baseados na independência das fontes, na esparsidade e/ou na não-negatividade.

### 1.2.1 Separação Usando Independência das Fontes

O Teorema do Limite Central [5] nos diz essencialmente que quanto mais variáveis aleatórias independentes somamos, mais a distribuição da variável aleatória resultante se aproxima de uma gaussiana. Portanto, supondo que as fontes são independentes umas das outras, podemos tentar estimar  $\mathbf{W}$  de maneira que a não-gaussianidade de  $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$  seja maximizada. Deste modo, no final, vamos obter os sinais mais estatisticamente independentes que forem possíveis, assemelhando-se dos sinais das fontes originais. Geralmente, para medir a não-gaussianidade de um sinal utiliza-se a negentropia [6] e/ou o módulo da curtose [7].

Esta é a ideia básica por trás da análise de componentes independentes (em inglês *Independent Component Analysis*, ICA). A ICA é um poderoso método utilizado em larga escala em mineração de texto, análise financeira, remoção de reflexões em imagens e em comunicações por CDMA (do inglês *Code-Division Multiple Access*) [8].

Para que a ICA possa ser utilizada, é necessário que o número de misturas (ou sensores),  $M$ , seja maior que ou igual ao número de fontes,  $R$ . Dessa forma, ela não pode ser utilizada diretamente em casos de separação de sinais musicais, por exemplo, onde há apenas um ou dois canais. Por isso, ela não é muito comum nesta área de aplicação. Para mais detalhes, veja [9].

### 1.2.2 Separação Usando Esparsidade das Fontes

Já que em muitos casos a ICA não pode ser utilizada, muitos algoritmos de supervisão tratam do problema analisando a esparsidade dos dados obtidos. Esta área de estudo é chamada de codificação esparsa (em inglês *Sparse Coding*) ou SCA (do inglês *Sparse Component Analysis*).



O princípio de funcionamento dos métodos dessa natureza se baseia no fato de os sinais musicais terem, em sua grande parte, *pitch* bem definido, o que reflete sua harmonicidade (componentes frequenciais múltiplas de uma fundamental, ver Seção 2.2), e, portanto, esparsidade espectral. Em contraste, sua natureza rítmica introduz por vezes eventos esparsos no tempo.

A *Sparse Coding* leva isto em consideração, e encontra a matriz  $\mathbf{W}$  forçando o vetor  $\hat{\mathbf{s}}$  a ficar o mais esparsos possível. Mais detalhes podem ser vistos em [10].

### 1.2.3 Separação Usando Não-negatividade

Assim como os anteriores, métodos deste tipo também impõem uma restrição muito comum sobre o problema que estamos enfrentando. Desta vez, há a imposição de não-negatividade de todos os elementos de todas as matrizes. São conhecidos como métodos de fatoração de matrizes não negativas, ou pela sigla NMF (do inglês *Non-negative Matrix Factorization*).

A imposição da não-negatividade de todos os elementos do cálculo facilita o problema da separação por garantir que não ocorre nenhum tipo de interferência destrutiva no processo de formação das misturas; por isso, teoricamente podemos desacoplar as fontes de volta, desde que cada uma apareça alguma vez isolada no sinal.

Provavelmente porque a não-negatividade das fontes é mais intuitiva na representação de imagens, onde os valores dos píxeis sempre são maiores ou iguais a zero, um método que utiliza essa restrição foi proposto pela primeira vez no contexto de processamento de sinais em 1999, pelos pesquisadores Lee e Seoung, em [11], para a aplicação em imagens. Este artigo pela primeira vez utilizava a NMF para separar fontes, que, no caso, eram as diferentes partes de objetos e de rostos humanos. Ele foi o pontapé inicial do método que atualmente é largamente conhecido e utilizado em diversas áreas.

Em se tratando de sinais musicais, foco principal deste trabalho, a não-negatividade pode parecer uma restrição muito forte e equivocada, uma vez que um sinal de música genérico pode ser negativo ou positivo. Entretanto, mesmo assim é possível utilizar este método se representarmos os sinais pelos seus respectivos espectrogramas de magnitude (ver Seção 2.1), cujos valores obrigatoriamente

são sempre não-negativos. Deste modo, é possível fatorar o espectrograma de uma única mistura em duas matrizes não negativas, uma representando padrões espectrais das fontes e a outra representando os ganhos de cada fonte na mistura original. Para descobri-las, é utilizado um algoritmo de otimização por gradiente descendente tentando minimizar o erro entre o espectrograma inicial e a multiplicação das duas matrizes geradas.

Hoje em dia, muitos algoritmos de separação de fontes com poucas misturas são baseados principalmente na NMF e, muitas vezes, a combinam com elementos de esparsidade e independência para aumentar sua complexidade e abrangência. O leitor interessado pode procurar por mais informações em [12].

### 1.3 Método Alternativo

Apesar de também haver métodos que misturam dois ou os até os três tipos de restrições acima, quase todos os métodos se enquadram nessas 3 grandes categorias da separação de fontes — independência, esparsidade e não-negatividade.

Pesquisando na literatura especializada, um método alternativo [1], apesar de possuir alguns elementos de esparsidade e não-negatividade que o aproximam dos métodos de SCA e de NMF, destaca-se devido à não utilização direta de nenhuma dessas restrições. Distintivamente, baseia-se somente em uma característica tímbrica muito interessante dos instrumentos musicais: ao contrário do trato vocal, que pode modificar seu formato e conseqüentemente variar o timbre das notas emitidas, os instrumentos harmônicos possuem um corpo fixo, o que, segundo o autor, gera um perfil espectral constante dos harmônicos por eles emitidos que, supostamente, permite identificá-los dos demais.

A este perfil típico dá-se o nome de estrutura harmônica média (em inglês *Average Harmonic Structure*, AHS), explicada com mais detalhes na Seção 2.3. Além disso, o método ainda é capaz de aprendê-las a partir da mistura, não havendo a necessidade, portanto, de uma análise prévia de cada instrumento separadamente. Isso permite que o método seja usado de maneira completamente não-supervisionada.

É este o método que motivou a realização do projeto e os próximos capítulos não só explicarão como reproduzi-lo, mas também relatarão os testes realizados e

resultados obtidos.

## **1.4 Objetivo**

O objetivo do projeto é fazer uma investigação e reprodução detalhadas de um método da literatura que modela as estruturas harmônicas médias (AHS) dos instrumentos e as utiliza para separar fontes sonoras de maneira completamente não-supervisionada [1]. O projeto deixa ainda como resultados uma plataforma de testes com as implementações documentadas e este texto de relatório, para utilização em novos trabalhos.

## **1.5 Organização**

O Capítulo 2 dá ao leitor os fundamentos teóricos básicos e necessários para o entendimento do projeto. Em seguida, o Capítulo 3 explica detalhadamente o método e todos os parâmetros necessários para implementar todos os algoritmos aqui utilizados. Já o Capítulo 4 documenta os testes e resultados obtidos no projeto. As conclusões são apresentadas no Capítulo 5.

# Capítulo 2

## Fundamentos Teóricos

Alguns conceitos básicos necessários para o entendimento do trabalho serão explicados neste capítulo, dando ao leitor uma visão geral da teoria que será utilizada nos algoritmos posteriormente apresentados.

### 2.1 Espectrograma de Magnitude

A maneira mais intuitiva de se representar um sinal digital de áudio é usando uma representação temporal. Um exemplo simples pode ser visto na Figura 2.1, a qual nos mostra um sinal de áudio composto por 3 notas musicais diferentes tocadas em sequência. Observando-a com cuidado, apesar de ser possível tirar conclusões acerca do momento exato em que as notas foram tocadas, infelizmente não conseguimos inferir nada sobre quais são as notas presentes no sinal. O motivo principal desta dúvida reside na falta de clareza na representação da informação frequencial, que uma representação puramente temporal pode não conseguir fornecer diretamente.

Por isso, para tentar contornar o problema, uma maneira alternativa para a representação de um sinal de áudio seria visualizá-lo no domínio da frequência, onde teríamos toda a informação frequencial acerca das notas ali presentes; assim, conseguiríamos descobrir as notas que foram tocadas. Entretanto, calculando a transformada de Fourier discreta (em inglês *Discrete Fourier Transform*, DFT) [14] da Figura 2.1, chega-se na Figura 2.2, na qual nos deparamos com um outro problema. Nesta nova representação, apesar de detectarmos a presença das componentes fundamentais das notas que foram tocadas (220 Hz, 440 Hz e 880 Hz) e seus harmônicos,

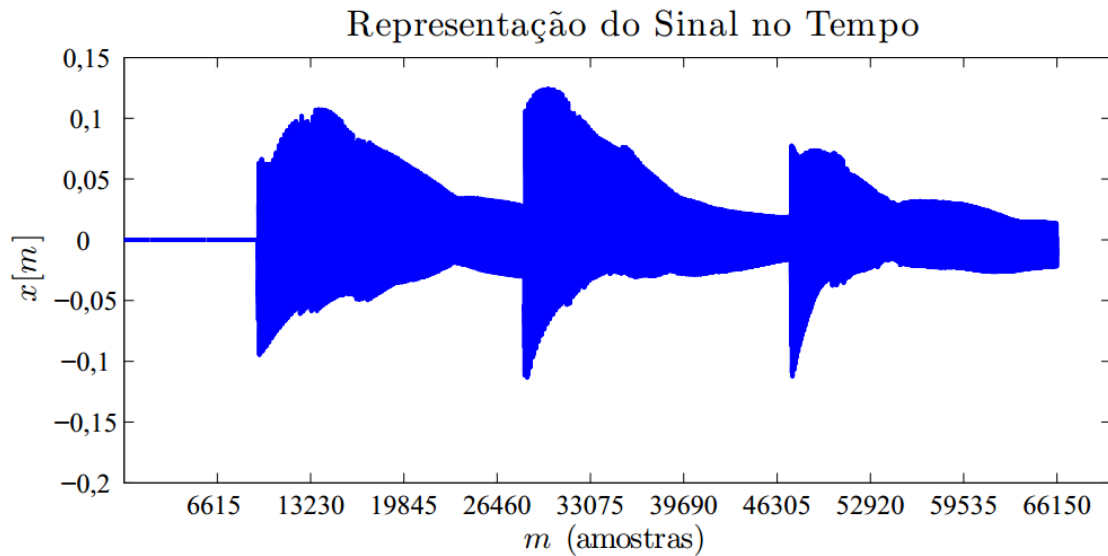


Figura 2.1: Representação no domínio do tempo de um sinal de áudio composto por 3 notas (Lá 220 Hz, Lá 440 Hz e Lá 880 Hz) tocadas em sequência. A taxa de amostragem é de 44,1 kHz. Adaptado de [13].

a informação temporal de quando e quantas notas foram tocadas, por exemplo, não está mais reconhecível.

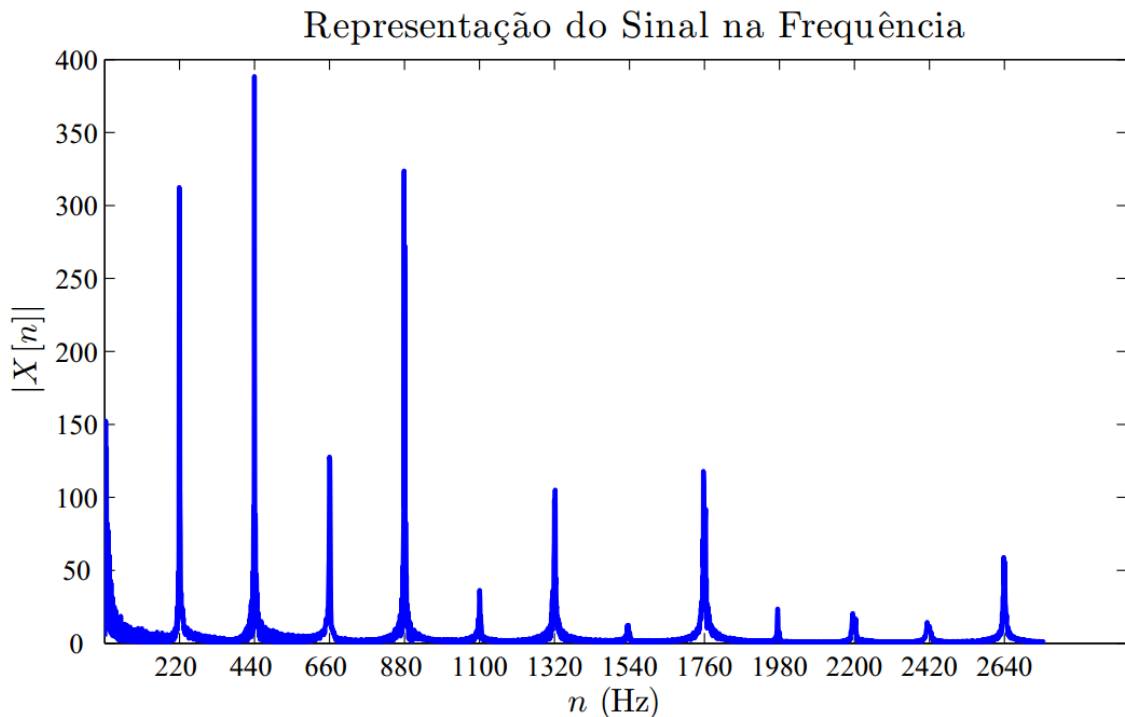


Figura 2.2: Um trecho da representação do mesmo sinal de áudio da Figura 2.1 no domínio da frequência. O eixo das ordenadas é a magnitude da transformada de Fourier discreta do sinal. Adaptado de [13].

Nesse sentido, na maior parte dos casos em que haja a necessidade de se obter

dados dos dois domínios, é de extrema importância escolher uma nova representação para os sinais de áudio que seja capaz de tornar mais evidentes as informações de tempo e de frequência. Com esse fim, foram criadas algumas representações tempo-frequenciais; a mais comum delas se baseia na técnica da transformada de Fourier de curta duração (em inglês *Short-Time Fourier Transform*, STFT) [15], gerando o que chamamos de espectrograma.

Em poucas palavras, para chegar nessa representação, basta aplicar a DFT em pequenos trechos (chamados de quadros) contíguos do sinal. Dessa maneira, conseguimos que a descrição frequencial do sinal evolua no tempo, juntando assim elementos das duas representações anteriores numa só. O procedimento padrão para se obter o espectrograma é descrito abaixo.

1. Partindo do sinal digital completo  $x[n]$  de duração  $L$  e uma função  $w[n]$  de duração  $W \ll L$ , realizamos uma sequência de deslocamentos e multiplicações para gerar  $M$  sinais de curta duração

$$x_m[n] = x[n + mD]w[n], \quad \text{com } n \in \{0, \dots, W - 1\} \text{ e } m \in \{0, \dots, M - 1\}, \quad (2.1)$$

a que damos o nome de quadros. Observe que a presença do deslocamento  $D \in \mathbb{N}^*$  na Equação (2.1) nos permite compreender  $x[n + mD]$  como um pequeno trecho de duração  $W$  do sinal original  $x[n]$ . Além disso, cada trecho precisa ainda ser multiplicado por uma função  $w[n]$  com o objetivo de amenizar o espalhamento frequencial causado pelo truncamento abrupto presente em  $x[n + mD]$  [16]. Tal função recebe o nome de função janela, e alguns tipos muito utilizados são as janelas de *Hamming*, de *Hann*, de *Blackman* e de *Kaiser* [16].

2. Aplicamos a DFT de tamanho  $N \geq W$  em cada um dos quadros  $x_m[n]$  do sinal  $x[n]$ , gerando os sinais  $X_m[k]$  dados por

$$X_m[k] = \sum_{n=0}^{N-1} x_m[n]e^{-j\frac{2\pi n}{N}k}, \quad \text{para } k \in \{0, \dots, N - 1\}. \quad (2.2)$$

3. Por fim, calculamos o espectro de magnitude  $|X_m[k]|$  para cada valor de  $m$  e, deste modo, montamos uma matriz  $N \times M$ , onde em cada coluna  $m$  colo-

camos  $|X_m[k]|$ . Essa matriz é, então, o que chamamos de espectrograma de magnitude do sinal digital  $x[n]$ . Note que um espectrograma de magnitude consegue evidenciar informações acerca dos dois domínios de interesse (tempo e frequência) do sinal.

O exemplo presente nas Figuras 2.1 e 2.2 tem seu espectrograma representado na Figura 2.3. Nela, o eixo das ordenadas representa as raiais de frequências e o eixo das abscissas representa cada quadro analisado. Quanto mais escuro for determinado ponto na figura, mais intensa é a presença de sua respectiva raia de frequência durante todo o quadro em questão. Dessa forma podemos descobrir não só as frequências presentes em cada nota, mas também os momentos aproximados em que estas foram tocadas e como se deu seu desvanecimento.

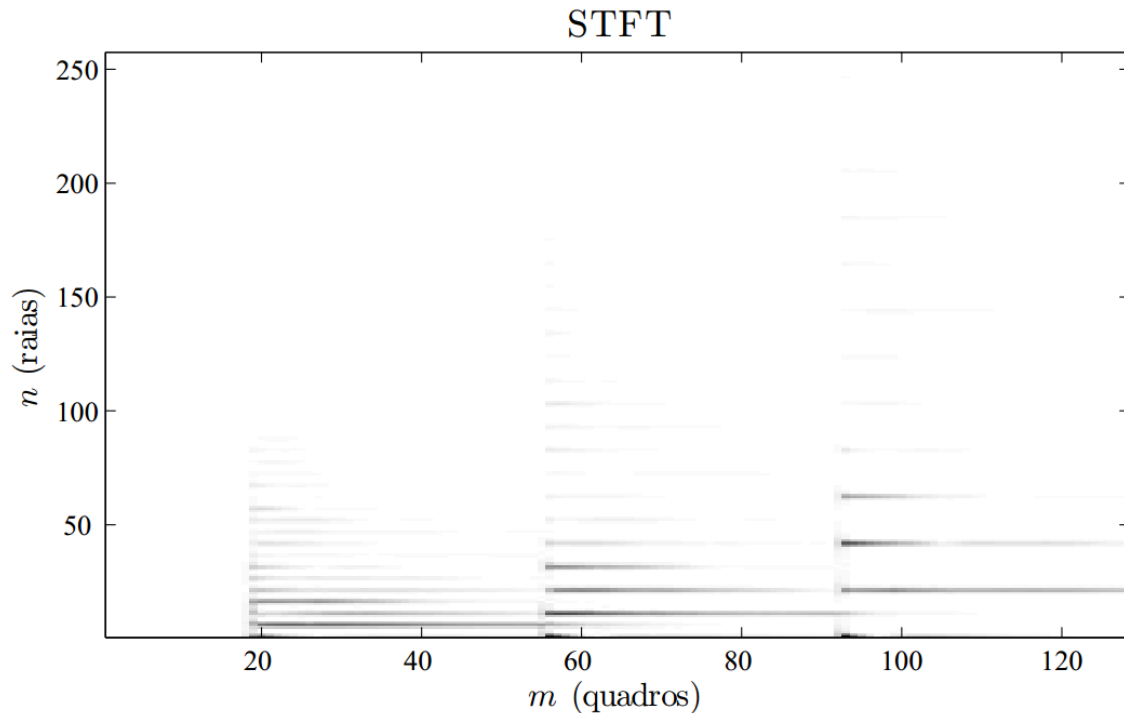


Figura 2.3: Espectrograma de magnitude do exemplo nas Figuras 2.1 e 2.2. Adaptado de [13].

## 2.2 Frequências Fundamental e Harmônicas

Na seção anterior, apareceram dois termos muito importantes no processamento digital de sinais de áudio: a “frequência fundamental” e suas “frequências

harmônicas”. Uma vez que são elementos centrais no algoritmo analisado neste trabalho, merecem uma seção para explicá-los mais detalhadamente.

Sabe-se que a menor frequência que se pode atribuir à vibração de um objeto qualquer é chamada de frequência principal ou frequência fundamental de vibração. Esta vibração, além de atingir as moléculas do objeto em questão, também induz uma onda de mesma frequência sobre as moléculas de ar ao seu redor, que, ao chegar em nossos ouvidos, dependendo da sua frequência fundamental, é percebida como um som.

Nesse sentido, suponha que um sinal de áudio é sintetizado usando apenas uma senoide cuja frequência varia dentro da faixa audível (de 20 Hz a 20 kHz, aproximadamente). Ao ouvirmos o som, nós o perceberemos como um sinal completamente artificial, sem timbre, sem “corpo”. Por isso, esta melodia de tons puros — termo muito utilizado na literatura — dificilmente poderá ser considerada música.

Isto acontece porque, na natureza, não conseguimos gerar tons puros sozinhos; aliás, uma senoide é uma idealização, já por ser perene. De todas as maneiras que conseguimos gerar som com altura definida, seja a partir de cordas, de tubos, ou até mesmo batendo em pedaços de madeira, por exemplo, uma gama de frequências é estimulada ao mesmo tempo. Para entender melhor, observe a Figura 2.4.

Quando vibramos uma corda ou sopramos um tubo, não definimos uma

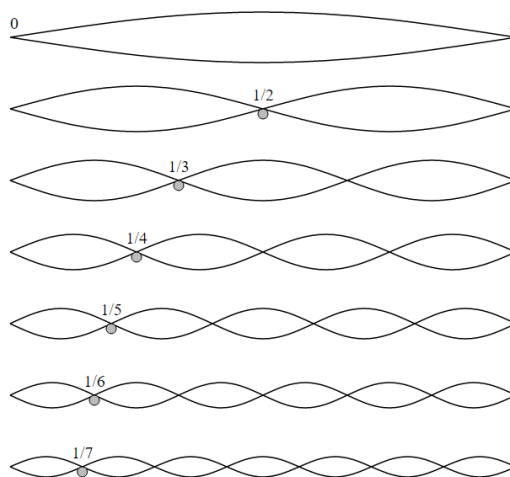


Figura 2.4: Exemplo de como é gerado o som em instrumentos de cordas. Note as diferentes frequências estimuladas a partir de uma vibração natural de uma corda genérica presa em suas extremidades. Um raciocínio análogo pode ser feito no caso de instrumentos de sopros, trocando a corda por tubos. Os valores dos comprimentos de onda estão normalizados.



frequência específica para sua vibração. Na verdade, o que fazemos é apenas definir a posição de alguns nós ou ventres na onda sonora e dar energia para a corda ou a coluna de ar vibrar. Desta forma, uma nota musical emitida pelo respectivo instrumento nada mais é do que uma combinação linear de ondas sonoras em frequências diferentes ressoando ao mesmo tempo.

Entretanto, é fácil perceber, olhando para a Figura 2.4 com mais atenção, que tais frequências não são geradas aleatoriamente. Todas têm algo em comum: são múltiplas da frequência principal ou fundamental de vibração do objeto em questão, usualmente representada por  $f_0$ .

Lembrando que o comprimento da onda é inversamente proporcional à sua frequência, ao olharmos a Figura 2.4, percebemos que a primeira onda exemplificada tem comprimento máximo dentre todas as presentes. É esta onda, portanto, a responsável pela frequência principal. Com cálculos simples é possível deduzir que as outras frequências presentes,  $f^i$ , serão múltiplos inteiros de  $f_0$ , isto é, para qualquer  $i \in \mathbb{N}$ ,

$$f^i = i f_0. \quad (2.3)$$

Além disso, as frequências  $f^i$  também recebem um nome especial:  $i$ -ésima frequência harmônica de  $f_0$ . Daí é que vem o termo ‘harmônico’, também utilizado para caracterizar alguns instrumentos. Os ditos instrumentos harmônicos são instrumentos musicais que geram notas musicais harmônicas, i.e., formadas por uma combinação linear de frequências harmônicas de suas frequências fundamentais. Pertencem a essa categoria<sup>1</sup> os instrumentos que serão misturados e separados neste projeto. Alguns exemplos são os instrumentos de cordas e de sopro. Uma discussão mais detalhada sobre o assunto pode ser encontrada em [17] e [18].

## 2.3 Estrutura Harmônica Média

De uma maneira geral, a quantidade de harmônicos gerados e a amplitude de cada um deles em relação à da frequência principal é o que nos permite distinguir os instrumentos harmônicos entre si. É por isso que, por exemplo, um Lá 220 Hz

---

<sup>1</sup>Utilizar um modelo de harmonicidade pura para instrumentos musicais ditos harmônicos também é uma abstração. Tal modelo explica, aproximadamente, o espectro característico desta categoria de instrumentos.

(frequência principal) tocado por um piano é percebido de maneira diferente pelos nossos ouvidos se comparado com a mesma nota Lá 220 Hz tocada por um violino. Apesar de ambas as notas possuírem a mesma frequência principal, a relação de energias entre os harmônicos é completamente diferente para os dois instrumentos.

Portanto, deve ser possível tirar conclusões acerca de qual fonte está emitindo sons apenas analisando-se o padrão espectral da nota tocada. De fato, se olharmos a Figura 2.5, podemos ver que, mesmo mudando as frequências estimuladas ao tocar diferentes notas, a relação de energia entre os harmônicos não varia muito. É como se cada fonte possuísse o seu próprio padrão espectral.

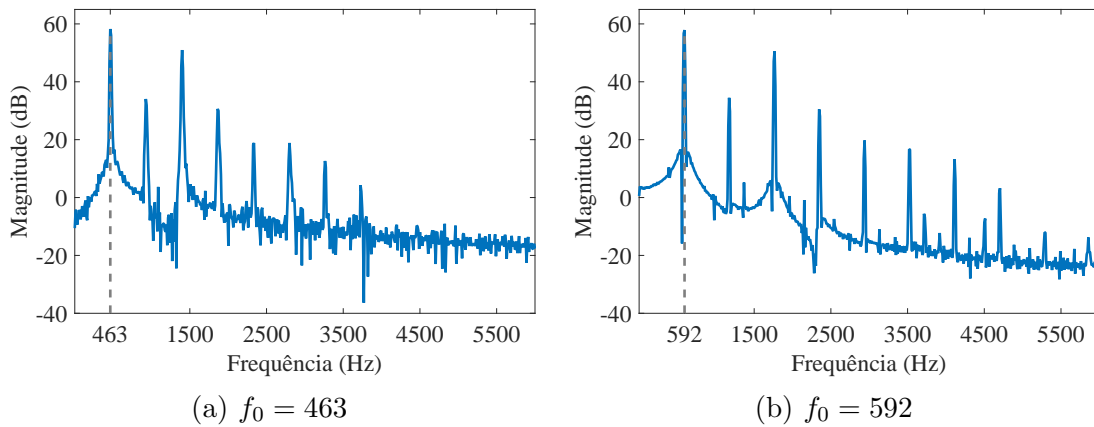


Figura 2.5: Espectro de magnitude de diferentes notas de um flautim. Observe a presença de um padrão espectral fixo, apesar de existirem diferentes harmônicos.

Nesse sentido, é possível descrever o timbre de um instrumento harmônico utilizando o que é conhecido como estrutura harmônica média (em inglês *Average Harmonic Structure*, AHS) [1].

Suponha que  $s(t)$  seja o sinal de áudio proveniente de uma fonte monofônica qualquer. Podemos representá-lo utilizando um modelo senoidal [19]

$$s(t) = \sum_{r=1}^R A_r(t) \cos[\theta_r(t)] + e(t), \quad (2.4)$$

onde  $e(t)$  é a componente ruidosa,  $A_r(t)$  e  $\theta_r(t) = \int_0^t 2\pi r f_0(\tau) d\tau$  são a amplitude instantânea e a fase do  $r$ -ésimo harmônico, respectivamente,  $f_0(\tau)$  é a frequência fundamental no tempo  $\tau$  e  $R$  é o número máximo de harmônicos presentes no sinal de áudio.

Podemos supor que o valor de  $A_r(t)$  é invariante dentro de um quadro de

duração bem pequena e, portanto, podemos reescrevê-lo como

$$A_r(t) \approx A_r^l, \quad (2.5)$$

no  $l$ -ésimo quadro. Assim, definimos a estrutura harmônica (em inglês *Harmonic Structure*, HS) no quadro  $l$  como o vetor com as amplitudes dos harmônicos significativos usando a escala decibel, isto é,

- Estrutura harmônica (HS) no quadro  $l$ :

$$\mathbf{B}^l = [B_1^l, \dots, B_R^l], \quad (2.6)$$

onde o coeficiente  $B_r^l$  é definido como

$$B_r^l = \begin{cases} 20 \log A_r^l, & \text{se } 20 \log A_r^l > 0 \\ 0, & \text{em caso contrário} \end{cases}, \quad \forall r \in \{1, 2, \dots, R\}. \quad (2.7)$$

★ Observe que os valores negativos de  $B_r^l$  devem ser considerados nulos. Esta não-negatividade é importante para que seja possível realizar a separação das fontes utilizando as estruturas harmônicas médias como “bases espectrais” (Subseção 3.3.2).

A estrutura harmônica é definida na escala decibel simplesmente porque o ouvido humano tem uma sensibilidade logarítmica à intensidade dos sinais de áudio. Além disso, utilizar uma escala logarítmica é interessante, pois a distância euclidiana entre duas estruturas (processo que será realizado na Subseção 3.2.4) representa razões de potência entre os coeficientes de cada estrutura.

Enfim, a estrutura harmônica média, assim como o próprio nome diz, é definida como o valor médio das estruturas harmônicas presentes em cada quadro. Além disso, podemos definir também a instabilidade da estrutura harmônica (em inglês *Harmonic Structure Instability*, HSI) [1] como sendo o desvio padrão médio dos seus coeficientes.

- Estrutura harmônica média (AHS):

$$\text{AHS} \equiv \bar{\mathbf{B}} = [\bar{B}_1, \bar{B}_2, \dots, \bar{B}_R], \quad (2.8)$$

$$\bar{B}_i = \frac{1}{L_i} \sum_{l=1, B_i^l \neq 0}^{L_i} B_i^l, \quad \forall i \in \{1, 2, \dots, R\}. \quad (2.9)$$

- Instabilidade da estrutura harmônica (HSI):

$$(\text{HSI})^2 = \frac{1}{R} \sum_{i=1}^R \left[ \frac{1}{L_i} \sum_{l=1, B_i^l \neq 0}^{L_i} (B_i^l - \bar{B}_i)^2 \right], \quad (2.10)$$

onde  $L_i$  é a quantidade total de quadros onde o  $i$ -ésimo coeficiente da estrutura harmônica é diferente de zero.

Observe que a fórmula utilizada para o cálculo da AHS (Equação (2.9)), definida por Duan et al. em [1], usa um valor de  $L_i$  diferente para calcular a média de cada harmônico, isto é, a média de cada harmônico é feita em relação à quantidade de quadros nos quais ele está presente. Por isso, na prática, caso algum harmônico apareça em pouquíssimos quadros apenas, devido à presença de ruído ou de lóbulos secundários [16] da função janela utilizada no processamento, ele terá um valor mais baixo para  $L_i$  do que os outros e, conseqüentemente, ganhará uma presença desproporcionalmente mais destacada na estrutura harmônica média.

Para evitar erros desta natureza, é utilizada aqui uma equação modificada para o cálculo da AHS, que define um valor mínimo  $L_{\min}$  para a quantidade de quadros que um harmônico qualquer deve aparecer para ser considerado no cálculo.

Enfim, chegamos a

$$\text{AHS} \equiv \bar{\mathbf{B}} = [\bar{B}_1, \bar{B}_2, \dots, \bar{B}_R], \quad (2.11)$$

$$\bar{B}_i = \begin{cases} \frac{1}{L_i} \sum_{l=1, B_i^l \neq 0}^{L_i} B_i^l, & \text{se } L_i \geq L_{\min}, \\ 0, & \text{senão,} \end{cases} \quad (2.12)$$

onde

$$L_{\min} = 30 \% \text{ de } L_1. \quad (2.13)$$

Além disso, como há a possibilidade de alguns dos  $R$  harmônicos analisados possuírem valores iguais a zero (não-significativos) na AHS de um instrumento genérico, a utilização direta da Equação (2.10) proposta por Duan et al. para calcular a HSI do instrumento causará uma polarização no resultado da média final dos coeficientes, pois estaremos sempre dividindo o resultado final por  $R$ , levando em consideração tanto os harmônicos significativos como os não-significativos nos quadros.

Assim, podemos criar uma nova fórmula corrigida para a HSI que será utilizada neste trabalho fazendo

$$(\text{HSI})^2 = \frac{1}{R'} \sum_{i=1, \bar{B}_i \neq 0}^{R'} \left[ \frac{1}{L_i} \sum_{l=1, B_i^l \neq 0}^{L_i} (B_i^l - \bar{B}_i)^2 \right], \quad (2.14)$$

onde  $R'$  é o número de harmônicos significativos, isto é, diferentes de zero, na AHS.

A Figura 2.6 mostra alguns exemplos de estruturas harmônicas médias e suas respectivas instabilidades para alguns dos instrumentos monofônicos mais comuns. Ela foi realizada utilizando a base de dados RWC [20], na qual existem gravações de alta qualidade e baixíssimo ruído com diversos instrumentos tocando as notas de seu alcance natural. Os instrumentos foram analisados em sua respectiva oitava central.

## 2.4 Escala de Frequência MIDI

Em 1983, com o crescimento da tecnologia digital, foi criado um padrão técnico para permitir a conexão e comunicação de uma grande variedade de instrumentos musicais eletrônicos e computadores. O nome dado ao padrão foi MIDI, sigla proveniente do inglês *Musical Instruments Digital Interface*, que, em tradução livre, significa interface digital para instrumentos musicais.

Nesse sentido, foi necessário especificar algumas novas formas de representar notas musicais numa mensagem de computador. Diversos parâmetros e sinais foram definidos, alguns controlando a duração e volume das notas, outros sincronizando as notas com o *clock* das máquinas, e até mesmo alguns representando efeitos sonoros como, por exemplo, o *vibrato*. Foi criada também uma notação diferente para

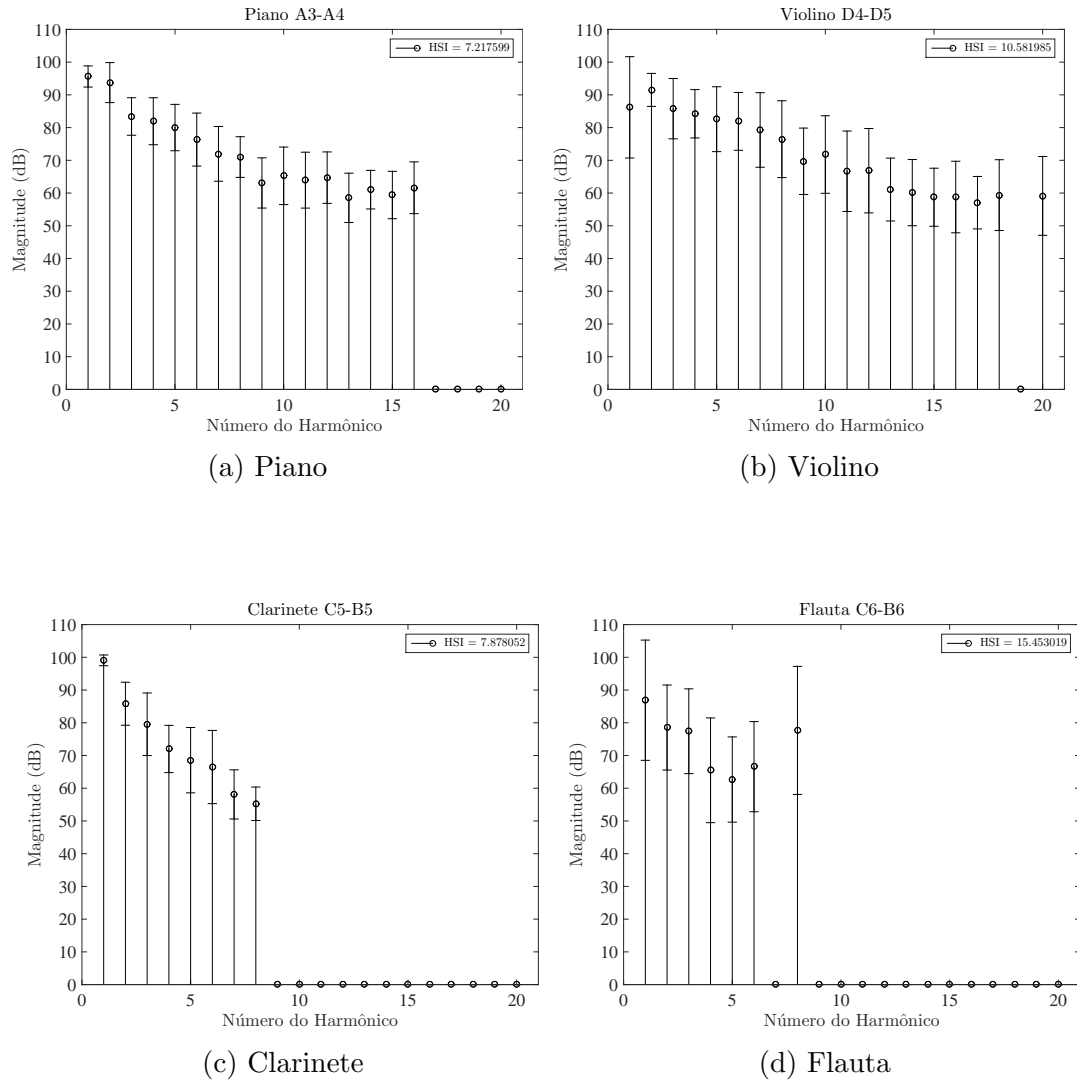


Figura 2.6: Estrutura harmônica média (AHS) calculadas para alguns instrumentos comuns.

representar as frequências fundamentais das notas musicais mais comuns na música ocidental. Essa representação é conhecida como o número MIDI de uma nota musical, e nada mais é do que a simples associação de um número inteiro de 7 bits a uma certa nota musical, numa tentativa de representá-la nos computadores mais antigos.

De uma maneira geral, os instrumentos harmônicos ocidentais utilizam o semitom como o menor intervalo entre duas notas sequenciais quaisquer. Para exemplificar, podemos dizer que um semitom é a diferença de altura produzida por duas teclas contíguas do piano.

Além disso, em teoria musical, um intervalo de 12 semitons define uma oitava ou, em outras palavras, é exatamente o intervalo necessário para encontrarmos duas

notas cujas frequências fundamentais,  $f_1$  e  $f_2$ , possuem a seguinte relação:

$$\frac{f_2}{f_1} = 2 \quad \Rightarrow \quad f_2 = 2f_1. \quad (2.15)$$

Note que, depois de 12 semitons,  $f_2$  torna-se exatamente o dobro de  $f_1$ . Entretanto, nossos ouvidos não percebem a diferença de altura das notas musicais linearmente, mas sim de uma forma logarítmica. Por isso, o semitom é definido utilizando-se uma escala deste tipo.

Seja  $s$  a representação de um semitom. Considerando a relação das frequências em uma oitava (12 semitons) dada pela Equação (2.15) e utilizando uma escala logarítmica, podemos escrever

$$\log_2 f_2 = \log_2 f_1 + 12 \log_2 s \quad \Rightarrow \quad f_2 = 2f_1 = f_1 s^{12} \quad (2.16)$$

$$\log_2 s = \frac{1}{12} \quad \Rightarrow \quad s = 2^{1/12} = \sqrt[12]{2} \quad (2.17)$$

Dessa forma, na notação MIDI, uma unidade, a menor unidade de medida de uma escala que utiliza apenas números inteiros, deve representar um intervalo de um semitom entre notas musicais, pois este é também o menor intervalo possível entre as notas. Utiliza-se o número 69 para representar a nota Lá 440 Hz (A4), por exemplo. Usando essa mesma nota como referência nos cálculos, podemos encontrar uma fórmula para o número MIDI<sup>2</sup>:

$$F = 69 + 12 \log_2 \left( \frac{f}{440 \text{ Hz}} \right), \quad (2.18)$$

onde  $f$  é a frequência em Hertz e  $F$  o número MIDI. Observe que o valor de  $\log_2 \left( \frac{f}{440} \right)$  nos dá o número de oitavas acima da referência (440 Hz). Multiplicá-lo por 12, portanto, converte o resultado em quantidade de semitons que, somada com o número 69 (número inteiro associado a 440 Hz) nos dá o número MIDI da frequência  $f$ .

O processo inverso também pode ser facilmente usando

$$f = 2^{(F-69)/12} \cdot 440 \text{ Hz}. \quad (2.19)$$

---

<sup>2</sup>Considerando o diapasão definido pela nota Lá 440 Hz (A4) e a escala de temperamento igual definida pela Equação (2.17), a Equação (2.18) sempre resulta em números inteiros para  $F$

Apesar de originalmente serem utilizados apenas números inteiros na notação MIDI, muitas vezes, ao processar sinais de áudio, expande-se a notação para números reais. Isto acontece porque representar um valor de frequência qualquer em MIDI facilita muito os cálculos em casos onde é desejado usar frações de semitom como intervalos frequenciais, pois dessa forma, em vez de se utilizar uma escala logarítmica (de fração de semitom em fração de semitom), utiliza-se uma escala linear, definida agora em todo o eixo real.

## 2.5 Estimação por Máxima Verossimilhança

Ao estudarmos um fenômeno de natureza imprevisível é conveniente analisá-lo à luz de um modelo estocástico [5]. Nesse contexto, criamos modelos matemáticos para representá-lo simplificadamente, ilustrando certos aspectos fundamentais para a sua análise, sem contemplar necessariamente todos os detalhes.

Vale ressaltar que não existem modelos certos ou modelos errados, modelos verdadeiros ou falsos. Há apenas modelos mais simples ou modelos mais complexos e detalhados, que implicam diferentes tipos de perda de informação. Resta ao pesquisador verificar, qual conjunto de perdas é o menos danoso à sua análise. Assim, ao analisarmos um processo estocástico, é de extrema importância seguir uma metodologia para a escolha de um modelo que julgamos “melhor” do que os demais em algum aspecto.

Para exemplificar matematicamente, vamos supor que tenhamos apenas um modelo genérico  $\mathcal{M}$  em nossas mãos e queremos torná-lo capaz de descrever um determinado fenômeno de nosso interesse. É claro que não sabemos se esse modelo é o mais adequado para o sistema que pretendemos analisar, porém é possível estimar os valores de seu conjunto de parâmetros,  $\boldsymbol{\theta}$ , de acordo com alguma metodologia estatística para que o modelo represente o fenômeno de forma satisfatória. Sem perda de generalidade podemos definir o conjunto de parâmetros como

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_D], \quad (2.20)$$

se consideramos os parâmetros do modelo inseridos num espaço de dimensão  $D$ . A dimensão do espaço de parâmetros muitas vezes também é referida, na literatura,



como a complexidade do modelo  $\mathcal{M}$ .

Um detalhe importante que deve ser levado em consideração é que, na prática, nunca conseguiremos obter infinitas observações do fenômeno que queremos analisar; portanto, será utilizado apenas um conjunto finito de observações, digamos, as  $N$  observações às quais temos acesso, para estimar os parâmetros  $\boldsymbol{\theta}$  do modelo. Enfim, o que desejamos fazer é adaptar os parâmetros do modelo  $\mathcal{M}$  de complexidade  $D$  aos  $N$  dados relativos a um determinado fenômeno.

Geralmente, a metodologia adotada para a estimação dos parâmetros é a da estimação por máxima verossimilhança [21] (em inglês *Maximum Likelihood*, ML), formalmente resumida nos passos abaixo.

1. Suponha que as únicas informações existentes sobre um determinado fenômeno de interesse sejam provenientes de  $N$  observações  $O_i$ , com  $i \in \{1, \dots, N\}$ . Podemos definir, portanto, o vetor de observações

$$\mathbf{O} = [O_1, O_2, \dots, O_N]. \quad (2.21)$$

2. Seja  $\mathcal{M}$  um modelo estatístico com complexidade  $D$  e conjunto de parâmetros  $\boldsymbol{\theta}$  que desejamos utilizar para representar o fenômeno. Podemos tirar conclusões acerca das probabilidades de cada uma das observações ocorrerem, dado um conjunto fixo de parâmetros, utilizando a função densidade de probabilidade condicional [5]

$$p(\mathbf{O}|\boldsymbol{\theta}), \quad (2.22)$$

também chamada de função de verossimilhança de  $\boldsymbol{\theta}$ , dadas as observações  $\mathbf{O}$ .

- A notação  $p(\mathcal{A}|\mathcal{B})$  é muito utilizada na literatura e representa a função de densidade de probabilidade do evento  $\mathcal{A}$  condicionada à ocorrência do evento  $\mathcal{B}$ .
3. Assuma que as observações  $O_i$  são condicionalmente independentes umas das outras. Esta é uma hipótese plausível, uma vez que os parâmetros do modelo estão fixos em um determinado valor. Portanto, é possível calcular a

verossimilhança escrevendo

$$p(\mathbf{O}|\boldsymbol{\theta}) = p(O_1, O_2, \dots, O_N|\boldsymbol{\theta}) = \prod_{i=1}^N p(O_i|\theta_1, \theta_2, \dots, \theta_D). \quad (2.23)$$

4. Finalmente, o valor de  $\hat{\boldsymbol{\theta}}_{\text{ML}}$  é encontrado maximizando-se a função de verossimilhança em relação a  $\boldsymbol{\theta}$ .

$$p(\mathbf{O}|\hat{\boldsymbol{\theta}}_{\text{ML}}) = \max_{\boldsymbol{\theta}} p(\mathbf{O}|\boldsymbol{\theta}). \quad (2.24)$$

- Em alguns casos, em vez de maximizar a verossimilhança propriamente dita, convém maximizar equivalentemente o seu logaritmo. Isso facilita não só a análise matemática, evitando expoentes nas fórmulas, mas também os próprios cálculos numéricos ao transformar as multiplicações em somas. Vale lembrar que uma grande quantidade de multiplicações de pequenos valores numéricos (produtório da verossimilhança) pode causar *underflow*, devido à precisão numérica do computador.

$$\operatorname{argmax}_{\boldsymbol{\theta}} p(\mathbf{O}|\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} \ln p(\mathbf{O}|\boldsymbol{\theta}) = \sum_{i=1}^N \ln p(O_i|\theta_1, \theta_2, \dots, \theta_D). \quad (2.25)$$

É possível entender também, intuitivamente, a ideia por trás da estimação por máxima verossimilhança. É como se esse procedimento nos desse os valores para o conjunto de parâmetros de um modelo estatístico que maximizassem as probabilidades de ocorrência das  $N$  observações que possuímos previamente. Dizemos que o modelo  $\mathcal{M}$  foi adaptado às  $N$  observações do fenômeno desejado e, por isso, podemos generalizá-lo com uma certa confiabilidade para retratar o fenômeno como um todo.

Vamos analisar agora um outro caso. Suponha que desta vez deseje-se utilizar um outro modelo  $\mathcal{M}'$ , de complexidade  $D' = D + 1$ , para representar o mesmo fenômeno. As observações à nossa disposição são as mesmas  $N$  do caso anterior, entretanto a dimensão do espaço de parâmetros do modelo é maior do que a do caso

anterior. Então, o seu conjunto de parâmetros pode ser definido como

$$\boldsymbol{\theta}' = [\theta'_1, \theta'_2, \dots, \theta'_D, \theta'_{D+1}]. \quad (2.26)$$

Se estimarmos os seus parâmetros da mesma maneira que antes, descobriremos que o valor máximo de sua verossimilhança é pelo menos igual ao valor máximo encontrado para o modelo  $\mathcal{M}$ .

Intuitivamente, isto acontece porque o modelo  $\mathcal{M}'$  tem mais graus de liberdade do que o modelo  $\mathcal{M}$  e por isso, é mais fácil ajustar os valores dos seus parâmetros às observações do fenômeno. Em outras palavras, ao maximizarmos a verossimilhança para o modelo  $\mathcal{M}$ , modelamos estatisticamente o fenômeno da melhor maneira possível sobre um espaço de dimensão  $D$  gerado pelos seus parâmetros. Ao utilizarmos o modelo mais complexo  $\mathcal{M}'$ , no entanto, representamos o fenômeno usando um espaço de parâmetros de dimensão maior. Por isso, é possível chegar a uma modelagem estatística no mínimo equivalente à anterior se ignorarmos a nova dimensão, ou aproveitá-la para procurar valores para os seus parâmetros que aumentem ainda mais a verossimilhança final.

Dessa forma podemos escrever que

$$\max_{\boldsymbol{\theta}'} p(\mathbf{O}|\boldsymbol{\theta}') \geq \max_{\boldsymbol{\theta}} p(\mathbf{O}|\boldsymbol{\theta}). \quad (2.27)$$

A Equação (2.27) traz à tona uma nova questão acerca da escolha do “melhor” modelo estatístico a ser empregado para representar um fenômeno de interesse. Se compararmos a qualidade de representação de  $\mathcal{M}$  com a de  $\mathcal{M}'$  nos baseando apenas na máxima verossimilhança que eles são capazes de oferecer, ou seja, na maior probabilidade de ocorrência das observações que cada um é capaz de prever ajustando seus parâmetros aos dados do fenômeno, escolheremos o modelo  $\mathcal{M}'$  como o vencedor. Entretanto a escolha foi consequência direta simplesmente do fato de o modelo  $\mathcal{M}'$  possuir maior complexidade do que o concorrente.

Assim, de uma maneira geral, numa “disputa” entre vários modelos de diferentes complexidades, sempre acabaríamos escolhendo um modelo com a maior complexidade possível, aumentando cada vez mais as chances de causar *overfitting* [21] nos dados.

Na seção seguinte é explicado a que se refere esse termo, e um exemplo é analisado mais detalhadamente. Entretanto, o leitor interessado pode procurar mais informações em [21, 22] acerca do tema da escolha de modelo estatístico.

## 2.6 *Overfitting* e Critério de Informação Bayesiano

*Overfitting* é um termo bastante utilizado para sinalizar que o modelo estatístico escolhido para representar certo fenômeno não é capaz de generalizar o comportamento do sistema para novas observações. Geralmente o *overfitting* ocorre quando utilizamos um modelo de complexidade mais alta do que a requerida pelo fenômeno, pois, por possuir mais graus de liberdade do que o necessário, ele acaba predizendo incorretamente novas observações, mesmo funcionando de maneira satisfatória para as  $N$  observações utilizadas previamente no cálculo dos seus parâmetros.

Para ilustrar, vamos supor que um fenômeno qualquer que se deseja analisar possa ser modelado corretamente através de uma função senoidal da seguinte forma:

$$f(x) = \text{sen}(2\pi x), \quad (2.28)$$

onde  $f(x)$  representa o fenômeno propriamente dito e  $x$  uma variável qualquer da qual o fenômeno é dependente. Esse modelo simples poderia descrever a variação da posição de um pêndulo com o passar do tempo, por exemplo. Neste caso,  $f(x)$  teria unidades de distância e  $x$ , de tempo.

Além disso, como há sempre um erro de medida nas posições do pêndulo, de uma maneira geral, podemos adicionar um ruído à função original criando um processo estocástico

$$d(x) = \text{sen}(2\pi x) + \eta(x), \quad (2.29)$$

onde  $\eta(x)$  é, por exemplo, um ruído branco gaussiano de média zero e pequena variância, para a qual foi utilizado o valor de 0,3 nas ilustrações abaixo. A Figura 2.7 mostra 10 observações  $d(x_i)$ , para  $i \in \{1, \dots, 10\}$ , do fenômeno espaçadas uniformemente no intervalo  $[0, 1]$  junto da função mãe,  $f(x) = \text{sen}(2\pi x)$ .

Suponha agora que, assim como acontece na prática, não se saiba a origem

do fenômeno, isto é, não é sabido previamente que foi a função  $f(x)$  que originou as 10 observações que possuímos acerca de um fenômeno genérico  $d(x)$ .

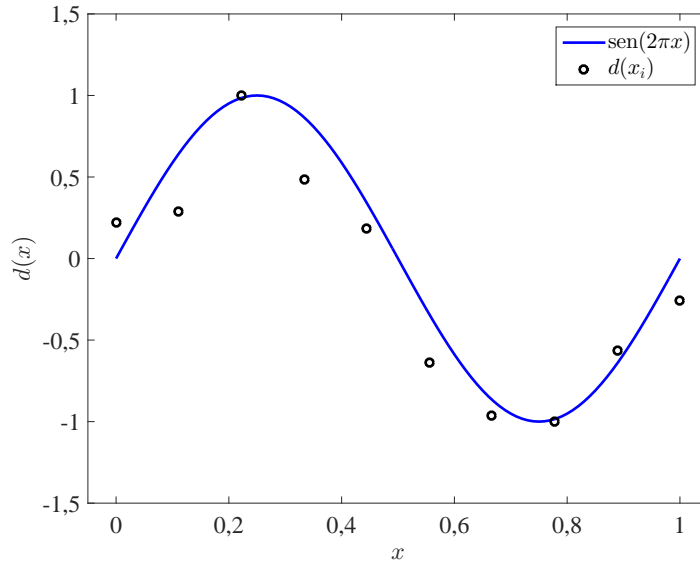


Figura 2.7: 10 observações de um fenômeno genérico qualquer  $d(x)$  que, idealmente, pode ser modelado como uma senoide  $\text{sen}(2\pi x)$ .

Nesse sentido, é necessário encontrar uma função  $\hat{d}(x)$  que tente modelar o sistema e consiga prever um valor de  $\hat{d}(x_o)$  mais próximo do real  $d(x_o)$  para qualquer valor de  $x_o \neq x_i, \forall i \in \{1, 2, \dots, 10\}$ .

Este problema, também conhecido como um problema de regressão, geralmente é abordado supondo-se que a função  $d(x)$  pode ser aproximada por um polinômio de grau  $M$ ,

$$\hat{d}_M(x) \approx d(x) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M. \quad (2.30)$$

Os valores do parâmetro  $\mathbf{w}$ , que são os coeficientes  $(w_0, w_1, w_2, \dots, w_M)$  do polinômio, são determinados ajustando-se o polinômio de grau  $M$  aos 10 valores provenientes das 10 observações  $d(x_i)$ . Isto pode ser feito de várias maneiras. Uma delas é através da minimização da função erro quadrático<sup>3</sup>, que pode ser definida como

$$E_2(\mathbf{w}) = \sum_{i=1}^{10} \left[ \hat{d}_M(x_i) - d(x_i) \right]^2. \quad (2.31)$$

---

<sup>3</sup>Em [21] pode-se encontrar a prova de que a análise de minimização do erro quadrático é equivalente a uma análise de maximização de uma função de verossimilhança.

A Figura 2.8 nos mostra o gráfico do polinômio encontrado para diferentes valores de  $M$  após o ajuste dos seus coeficientes aos dados para minimizar o resultado da Equação (2.31).

Visualmente, é fácil notar que com o aumento da ordem do polinômio, o erro quadrático final do ajuste vai diminuindo; basta perceber que ao chegar perto dos pontos  $x_i$ , a curva vermelha se aproxima cada vez mais dos valores obtidos nas observações  $d(x_i)$ . Mais do que isso, para ordens maiores do que 8, o erro quadrático final é capaz, inclusive, de tornar-se nulo, uma vez que sempre é possível encontrar um polinômio de ordem  $M \geq N - 1$  capaz de passar exatamente em  $N$  pontos do Plano Cartesiano.

Todavia, mesmo minimizando a Equação (2.31), observamos que fora dos pontos  $x_i$  o polinômio ajustado começa a distanciar-se cada vez mais da curva ideal em azul a partir das ordens  $M \geq 5$ . Assim, a modelagem começa a possuir tantos graus de liberdade que suas previsões acerca do fenômeno ficam cada vez mais distantes da realidade. É exatamente essa perda de generalização que chamamos de *overfitting*.

Para evitá-lo, adotamos alguns critérios de informação que procuram penalizar o modelo estocástico conforme sua dimensão aumenta. Um deles, chamado de Critério de Informação Bayesiano (em inglês *Bayesian Information Criterion*, BIC) foi introduzido por Schwarz em [23] e é utilizado na escolha de dimensão do modelo estatístico presente no algoritmo proposto em [1] por Duan et al.

Assim, supondo que a partir de  $N$  observações desejamos não só estimar os parâmetros  $\boldsymbol{\theta}_M$  de um modelo estatístico de dimensão  $M$  que melhor represente o fenômeno, mas também escolher o melhor valor de  $M$  que não cause *overfitting* nos dados, em vez de maximizar apenas a verossimilhança, maximizamos o que pode ser entendido como um logaritmo penalizado da verossimilhança representado pelo BIC, definido como

$$\text{BIC} = \ln p(\mathbf{O}|\boldsymbol{\theta}_M) - \frac{1}{2}M \ln N. \quad (2.32)$$

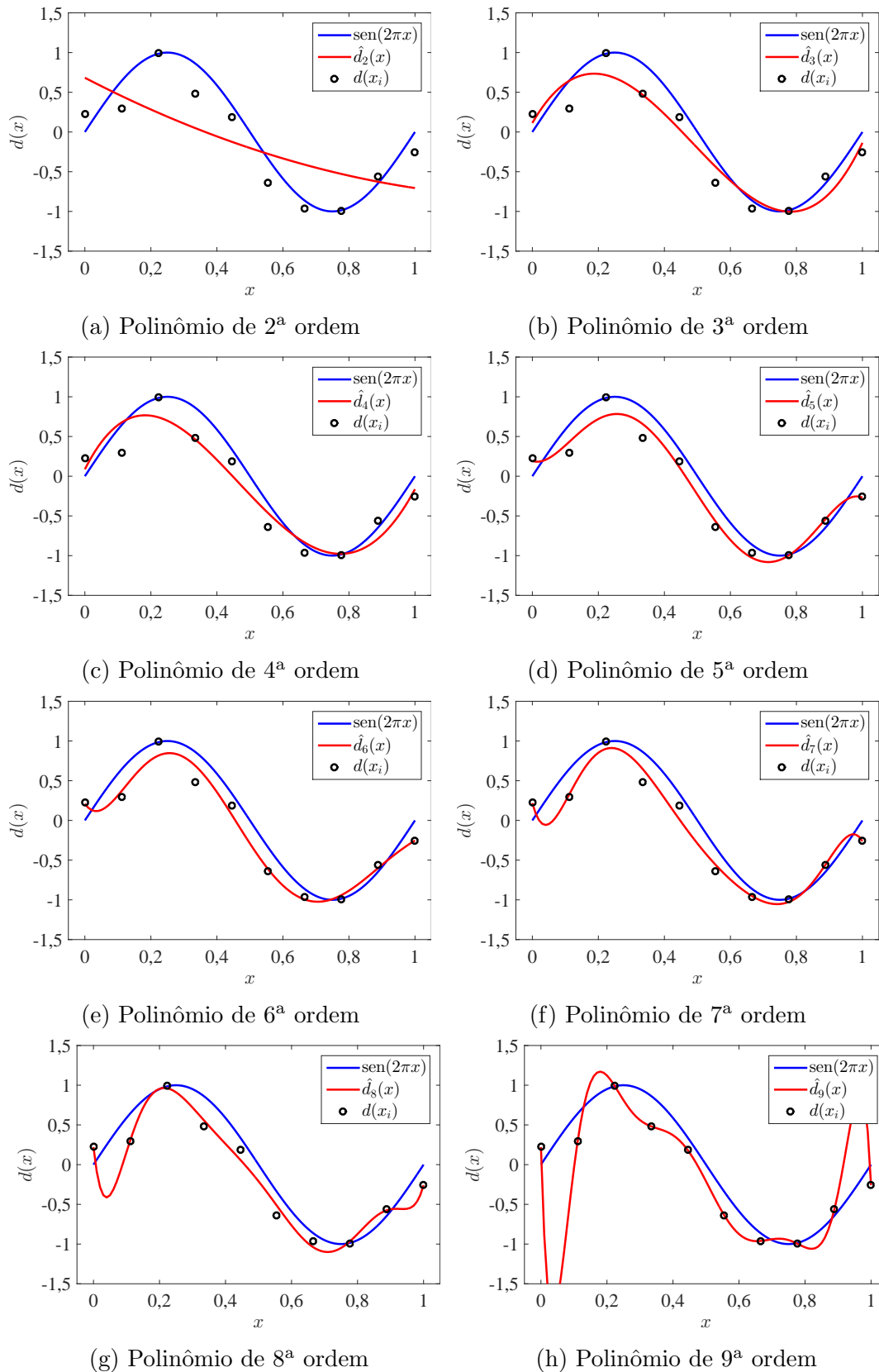


Figura 2.8: Comparação dos ajustes polinomiais de diferentes ordens às observações do fenômeno. Observe que o *overfitting* começa a acontecer a partir do 5º grau.

## Capítulo 3

# Separação Não-Supervisionada de Fontes usando Estruturas Harmônicas Médias

Este capítulo é destinado à descrição do método de separação não supervisionada de fontes utilizando as informações provenientes de uma estimação da estrutura harmônica média de cada uma das fontes presentes no sinal de áudio original.

O método implementado baseia-se principalmente no artigo de Duan et al. [1]. No entanto, este projeto apresenta uma explicação mais detalhada de como realizar a separação e como definir alguns parâmetros necessários para o algoritmo funcionar corretamente. Além disso, no artigo original, não está muito claro como o autor realiza algumas etapas do método ali proposto e, por isso, foram efetuadas algumas modificações no algoritmo original, numa tentativa de propor novos caminhos para resolver os problemas encontrados durante a implementação do projeto.

De uma maneira geral, a ideia central do método é tentar modelar o timbre de cada um dos instrumentos musicais misturados explorando a relação de energia entre os harmônicos presentes na mistura. Nesse sentido, o algoritmo de Duan et al., de uma forma criativa, propõe uma maneira completamente não-supervisionada de se descobrir a estrutura harmônica média de cada uma das fontes tendo em vista apenas o sinal de áudio misturado, isto é, com ele é possível inferir, somente a partir da mistura, o padrão espectral de cada instrumento. Por isso, não há a necessidade de tratá-los individualmente em etapas anteriores. Em seguida, o método utiliza



esta informação tímbrica para separar as fontes presentes na mistura.

Além disso, após o processamento do sinal, obtemos não só um arquivo de áudio para cada instrumento harmônico contendo sua respectiva participação na música, mas também obtemos uma estimativa para as frequências fundamentais de cada fonte, quadro a quadro do sinal. Portanto, este método pode ser também utilizado em aplicações que requeiram essa informação.

No entanto, antes da explicação detalhada do algoritmo, é conveniente reforçar ao leitor algumas definições e restrições importantes do projeto implementado. O algoritmo aqui presente lida com o problema de extração e isolamento de instrumentos harmônicos a partir de um sinal de áudio com apenas um único canal tratado de uma maneira completamente não-supervisionada<sup>1</sup>. Todos os instrumentos presentes são monofônicos, isto é, cada instrumento emite apenas uma nota musical por vez; não há, portanto, a emissão de acordes ou instrumentos polifônicos tocando várias notas simultaneamente.

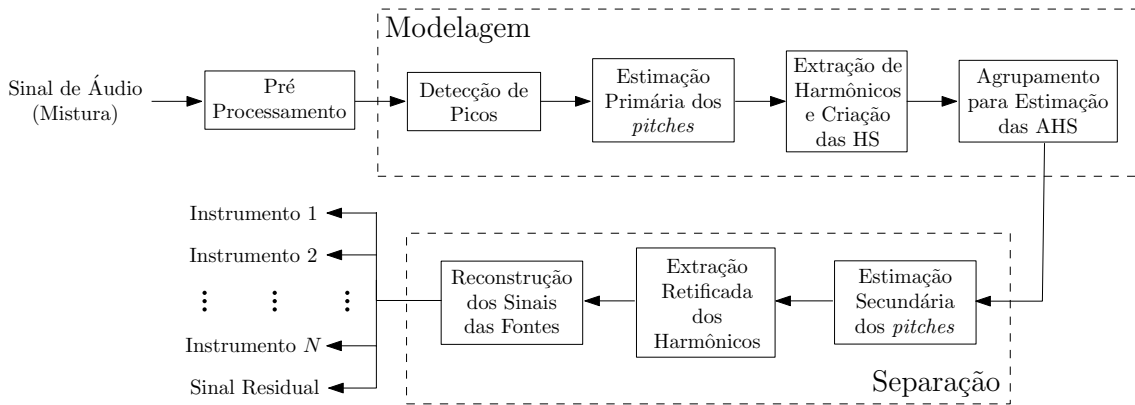


Figura 3.1: Diagrama de blocos do método implementado.

Podemos dividir o método em diversas etapas, conforme é ilustrado na Figura 3.1:

- Pré-Processamento;
- Detecção de Picos;
- Estimação Primária das Frequências Fundamentais ( $f_0$ );

<sup>1</sup>O termo “não supervisionado” é utilizado no sentido de não ser necessária nenhuma informação ou base de dados acerca dos instrumentos presentes; entretanto, é necessário, *a priori*, passar ao algoritmo a quantidade total de fontes misturadas no sinal original.

- Extração de Harmônicos e Criação das Estruturas Harmônicas (HS);
- Agrupamento para Estimação das Estruturas Harmônicas Médias (AHS);
- Estimação Secundária das Frequências Fundamentais ( $f_0$ );
- Extração Retificada dos Harmônicos;
- Reconstrução dos Sinais das Fontes.

As cinco primeiras etapas são responsáveis pela estimação não supervisionada das estruturas harmônicas médias de cada um dos instrumentos. A separação, propriamente dita, é executada apenas nas três últimas etapas. Dessa forma, podemos dizer que o método possui duas grandes fases. A fase de modelagem, onde o método tenta estimar as estruturas harmônicas médias que modelam os instrumentos presentes no sinal original de áudio, e a fase de separação, onde o método realiza a separação das fontes e extração dos instrumentos.

### 3.1 Pré-Processamento

Após a leitura ou conversão em canal único (mono) e a eliminação do valor médio (DC) do sinal de áudio, é realizada uma normalização de seu valor quadrático médio para a unidade. Em seguida é utilizada a STFT com uma janela de Hamming de duração aproximada de 93 milissegundos (2048 amostras a uma taxa de 22050 Hz) e sobreposição de 50 % para gerar o espectrograma da mistura. Por fim, o espectrograma de magnitude é convertido para decibéis com referência à unidade, isto é, cada elemento  $x$  é substituído por  $20 \log(x)$ , para ficarem de acordo com as definições da Seção 2.3. Todo o restante do processamento será feito nessa escala.

### 3.2 Modelagem

O foco principal da fase de modelagem do método é conseguir obter uma estimação confiável para as estruturas harmônicas médias (AHS) de cada instrumento conhecendo-se apenas a quantidade de fontes harmônicas monofônicas misturadas no sinal de áudio original.

Nesta fase, primeiramente, estimam-se as frequências principais de todas as notas presentes em um determinado quadro do espectrograma do sinal. Em seguida, constrói-se uma estrutura harmônica para cada nota estimada, extraíndo-se o respectivo padrão espectral de seus harmônicos.

Já que todos os instrumentos misturados são monofônicos, a estrutura relativa a uma nota qualquer, na realidade, pode ser associada ao instrumento responsável por tê-la emitido durante aquele quadro. Assim, ao realizarmos o mesmo procedimento em todos os quadros do espectrograma, conseguiremos uma grande quantidade de estruturas harmônicas que modelam os instrumentos presentes no sinal durante toda a música.

Para aprender a AHS de cada instrumento, podemos analisar o conjunto de estruturas calculadas de uma forma ligeiramente diferente. Ignorando o eixo temporal associado às estruturas (cada estrutura foi retirada de um quadro específico do sinal), podemos vê-las apenas como um conjunto de pontos espalhados em um espaço de dimensão  $R$ , que é a quantidade de harmônicos que são levados em consideração no cálculo de cada estrutura. A este espaço damos o nome de espaço de estruturas harmônicas ou, apenas, espaço de estruturas.

No caso deste projeto, o valor de  $R$  é definido como 20, assim como sugerido por Duan et al. Segundo os autores, mesmo que o valor de  $R$  varie com o instrumento analisado, ele pode ser fixado em 20, pois os harmônicos superiores ao vigésimo geralmente têm baixa amplitude e permanecem sob os lóbulos secundários dos harmônicos maiores, e para notas que contenham menos de 20 harmônicos, os coeficientes dos harmônicos superiores podem ser considerados nulos.

Dessa forma, cada estrutura seria um ponto em  $\mathbb{R}^{20}$ . Olhando dessa forma para o conjunto de dados e tendo em vista que o padrão espectral dos harmônicos estimulados ao tocarmos diferentes notas num mesmo instrumento não varia consideravelmente, é fácil prever que existirão muitos pontos relativamente próximos entre si, pertencentes ao mesmo instrumento, ao passo que possivelmente existirão outros mais distantes, pertencentes a instrumentos distintos, por exemplo.

Em outras palavras, se houver  $N$  fontes (instrumentos) misturadas no sinal, podemos concluir que existirão  $N$  grandes grupos de pontos no espaço de estruturas. A densidade de cada um deles está diretamente relacionada com a quantidade

de notas emitidas pelo respectivo instrumento durante a música. Nesse sentido, uma estimativa para a AHS de cada instrumento pode ser gerada encontrando-se o centroide de cada um dos  $N$  grupos mais significativos do espaço inteiro.

### 3.2.1 Detecção de Picos

Em cada quadro, os harmônicos provenientes das fontes são representados por picos no espectro de magnitude da STFT; por isso, uma etapa de detecção de picos é primordial.

Como podemos ver na Figura 3.2, os picos do espectro (curva em azul) são máximos locais; porém, apesar de alguns representarem os harmônicos corretamente, outros são gerados pelos lóbulos secundários ou pelo ruído presente no sinal. Assim, o algoritmo de detecção de picos deverá ignorar alguns máximos locais, detectando apenas os picos significativos, isto é, picos que têm maior probabilidade de estar relacionados com os harmônicos da nota que foi tocada no quadro.

A literatura é vasta em algoritmos para tratar este tipo de problema; no entanto, o algoritmo utilizado no projeto foi o mesmo proposto no artigo [1], pois este é a base principal do trabalho. O método de detecção de picos é detalhado a seguir para sinais de áudio com frequência de amostragem  $f_s = 22050$  Hz, divididos em quadros de 2048 amostras (aproximadamente 93 ms), parâmetros já utilizados no espectrograma.

- 1) Com o uso de um filtro média móvel de comprimento 9, calcula-se uma suavização do espectro de magnitude (em dB) do sinal, gerando as curvas em preto na Figura 3.2.
- 2) Define-se um limiar mínimo e global para os valores dos picos significativos como sendo o valor máximo do espectro menos 50 dB. Máximos locais abaixo deste limiar são mais prováveis de terem sido gerados por ruído.
  - No caso de o valor do limiar calculado ser menor do que zero, deve-se utilizar 0 dB como o verdadeiro limiar mínimo para a detecção de picos. O motivo de ignorarmos valores negativos de log-magnitude é a estrutura harmônica (definida na Equação (2.7)) obrigatoriamente precisar ser não-negativa para o algoritmo de separação funcionar corretamente.

- 3) São detectados como picos significativos os máximos locais cujos valores de magnitude excedem pelo menos em 8 dB os respectivos valores do espectro suavizado.
- 4) Por último, é realizada uma interpolação quadrática [24] nos valores encontrados para os picos com o objetivo de melhorar a precisão das posições dos máximos locais. Resumidamente, a interpolação é implementada processando-se cada pico em separado.
  - Calcula-se a parábola que passa pelo pico e por seus dois pontos mais próximos no espectro do quadro (um à esquerda e um à direita) e calcula-se a equação da parábola que passa nos três pontos;
  - Calculam-se, então, a nova frequência e a nova amplitude do pico como as coordenadas do ponto de máximo da parábola.

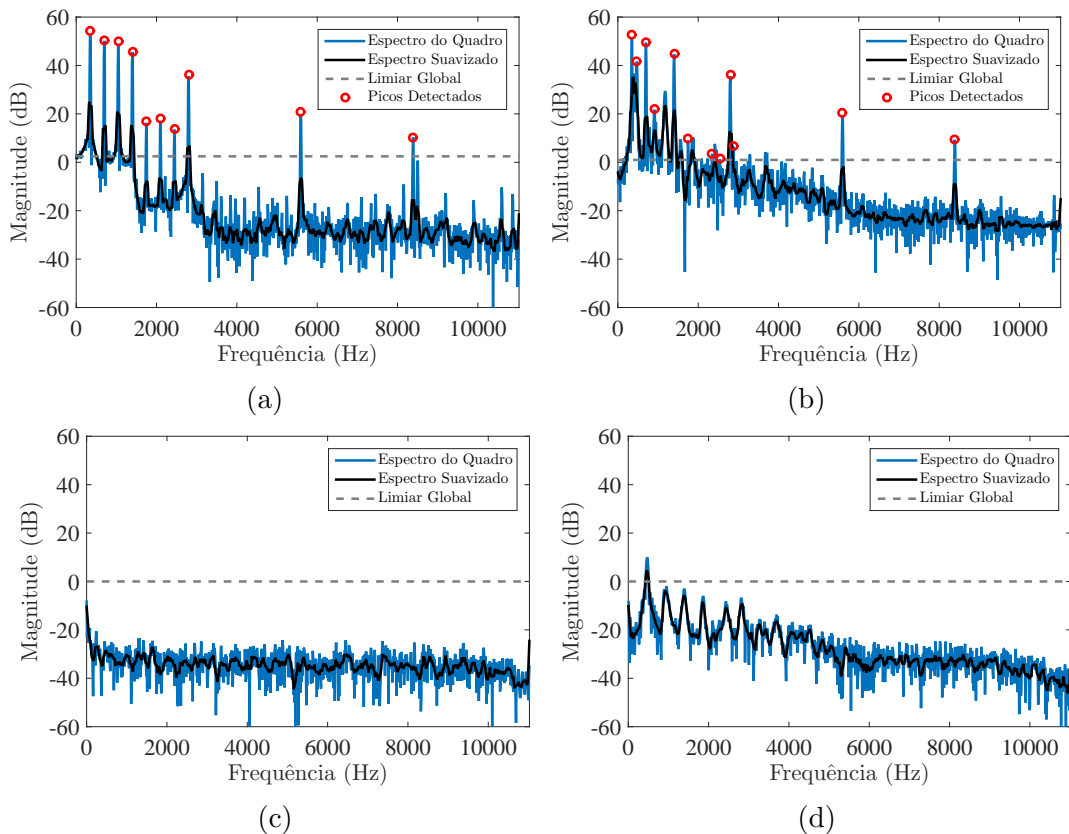


Figura 3.2: Comparação do resultado do algoritmo de detecção de picos em diferentes quadros de um sinal com duas fontes misturadas.

A Figura 3.2 ilustra o resultado do algoritmo aplicado em diversos quadros de um dos sinais de teste. No quadro ilustrado pela Figura 3.2a, há uma detecção

muito promissora dos picos mais significativos. Já na Figura 3.2b, é possível notar uma perda de fidelidade, uma vez que alguns picos importantes passam despercebidos para o algoritmo. Esse tipo de erro não será um grande problema para o método em geral, pois quadros onde isto acontece apenas acarretarão estruturas harmônicas completamente diferentes daquelas que desejamos estimar para os instrumentos e, portanto, no momento do agrupamento, serão eliminadas do cálculo como pontos de ruído no espaço de estruturas ou darão origem a pequeninos grupos muito distantes dos principais. Nesse caso, também poderão ser ignorados durante a etapa de estimação das estruturas harmônicas médias, pois os novos grupos possuirão densidades desprezíveis se comparados com os demais.

Nos outros dois quadros exemplificados não há picos significativos encontrados no sinal. Isto acontece porque não há nenhuma nota sendo emitida no quadro (Figura 3.2c), ou então o quadro faz parte do decaimento das notas ou do ataque dos instrumentos (Figura 3.2d). Ambos os casos não influenciam na estimação da AHS dos instrumentos; no entanto, na fase de separação alguns ataques ou decaimentos de notas serão incorretamente desconsiderados.

### 3.2.2 Estimação Primária das Frequências Fundamentais

As estruturas harmônicas de cada quadro do sinal misturado serão extraídas dos picos detectados na etapa anterior. No entanto, para isso, é necessário que antes haja uma estimação do número de fontes presentes no quadro analisado com suas respectivas frequências fundamentais. Na literatura, métodos para a estimação de frequências fundamentais de sinais de áudio são conhecidos com a sigla MPE, derivada do inglês *Multi-Pitch Estimation* [25] e, por isso, o algoritmo muitas vezes será referenciado com este nome neste trabalho. Além disso, o termo *pitch* (que a rigor é a altura percebida do som) também poderá ser utilizado como sinônimo de frequência fundamental.

Suponha que  $N$  fontes (instrumentos) estejam misturadas no sinal de áudio a ser processado. Além disso, para um determinado quadro qualquer da mistura, suponha que haja a detecção de  $P$  picos significativos pelo algoritmo da etapa anterior. Assim, podemos representar as frequências de cada pico como  $f_1, f_2, f_3, \dots, f_P$ , e suas amplitudes como  $A_1, A_2, A_3, \dots, A_P$ , respectivamente. Note que, mesmo com

a presença de  $P$  picos no quadro analisado e a informação *a priori* de que existem  $N$  instrumentos na mistura, não é possível, neste momento, ter a certeza da quantidade de fontes presentes no quadro. Pode ser, por exemplo, que todos os picos sejam harmônicos de apenas uma das fontes ou que um certo conjunto de harmônicos pertença a uma fonte e o resto pertença à outra, ou pode acontecer qualquer outra associação de harmônicos e fontes. Enfim, não é trivial inferir informações acerca da quantidade de fontes estimuladas no quadro.

Nesse sentido, considerando os  $P$  picos detectados num quadro genérico como uma observação de um processo estocástico, o algoritmo implementado estima não só a quantidade de notas presentes, mas também quais as suas frequências fundamentais a partir de uma busca exaustiva. Começemos modelando a verossimilhança no quadro analisado,  $p(\mathbf{O}|\mathbf{f}_0)$ , como

$$p(\mathbf{O}|\mathbf{f}_0) = p\left(f_1, f_2, \dots, f_P | f_0^1, f_0^2, \dots, f_0^{N'}\right) = \prod_{i=1}^P p\left(f_i | f_0^1, f_0^2, f_0^3 \dots, f_0^{N'}\right), \quad (3.1)$$

onde  $N'$  é o número real de notas (fontes) no quadro. Observe que a Equação (3.1) modela a verossimilhança geral de todos os picos do quadro como uma multiplicação das verossimilhanças individuais de todos os picos. Isso é válido se considerarmos as frequências dos picos significativos condicionalmente independentes dado o conjunto  $\mathbf{f}_0$  de frequências fundamentais do quadro, uma consideração muito utilizada na literatura [26].

Para simplificar o entendimento da modelagem utilizada no projeto para cada uma das verossimilhanças presentes na Equação (3.1), vamos primeiramente analisar o que acontece quando há apenas uma nota emitida no quadro analisado, ou seja,  $N' = 1$  e  $\mathbf{f}_0 = f_0^1$ . Neste caso, intuitivamente, é fácil perceber que  $p(f_i|f_0^1)$ , a probabilidade de um pico genérico  $f_i$  aparecer no espectro, dado que existe apenas uma nota de frequência principal  $f_0^1$  presente no quadro, é alta se  $f_i$  estiver próximo a qualquer um dos harmônicos de  $f_0^1$ , pois, teoricamente, qualquer um deles pode também ter sido estimulado durante a emissão da nota. Mais especificamente, podemos definir o harmônico ideal mais próximo do pico  $f_i$  como

$$f^{[f_i/f_0^1]} = \left\lceil \frac{f_i}{f_0^1} \right\rceil f_0^1, \quad (3.2)$$

onde o símbolo  $[\mathcal{X}]$  representa o número inteiro mais próximo de  $\mathcal{X}$ . Portanto, se  $f_i$  estiver muito perto de  $f^{[f_i/f_0^1]}$ ,  $p(f_i|f_0^1)$  será próxima de 1. Caso contrário, não é provável que o pico em  $f_i$  tenha sido gerado por uma nota com *pitch*  $f_0^1$ .

Assim, matematicamente, modelamos a verossimilhança como uma distribuição gaussiana [5] de variância  $\sigma_1^2$  dependente da distância relativa entre  $f_i$  e  $f^{[f_i/f_0^1]}$ .

$$p(f_i|f_0^1) = C_1 e^{-\left(\frac{d^2(f_i, f_0^1)}{2\sigma_1^2}\right)}, \quad (3.3)$$

$$d(f_i, f_0^1) = \frac{f_i - f^{[f_i/f_0^1]}}{f^{[f_i/f_0^1]}} = \frac{f_i - [f_i/f_0^1] f_0^1}{[f_i/f_0^1] f_0^1} = \frac{f_i/f_0^1 - [f_i/f_0^1]}{[f_i/f_0^1]}, \quad (3.4)$$

e  $C_1$  é apenas uma constante representando o fator de normalização necessário para a Equação (3.3) representar uma função de probabilidade válida. Note que a distribuição foi definida usando-se uma escala de frequências normalizada pelo harmônico ideal mais próximo de  $f_i$ . O valor utilizado para o desvio padrão  $\sigma_1$  foi arbitrariamente escolhido em todos os testes para corresponder à metade de um semitom, portanto

$$\sigma_1 = \frac{\sqrt[24]{2} f^{[f_i/f_0^1]} - f^{[f_i/f_0^1]}}{f^{[f_i/f_0^1]}} = \sqrt[24]{2} - 1 \approx 0,03. \quad (3.5)$$

Suponha agora que uma nova frequência fundamental  $f_0^2$  seja adicionada ao conjunto  $\mathbf{f}_0$ . Nesse caso,  $N' = 2$  e, devemos associar o pico  $f_i$  à frequência principal pertencente a  $\mathbf{f}_0$  mais provável de tê-lo estimulado. Portanto, matematicamente, teríamos:

$$p(f_i|f_0^1, f_0^2) = \max_j p(f_i|f_0^j), \quad \text{para } j \in \{1, 2\}. \quad (3.6)$$

Finalmente, definindo  $d(f_i)$  como sendo a menor de todas as distâncias relativas,  $d(f_i, f_0^j)$ , do pico  $f_i$  para as  $N'$  frequências fundamentais possíveis, chegamos



à modelagem final da verossimilhança:

$$p(f_i | f_0^1, f_0^2, \dots, f_0^{N'}) = C_1 e^{-\left(\frac{d^2(f_i)}{2\sigma_1^2}\right)}, \quad (3.7)$$

$$d^2(f_i) = \min_j d^2(f_i, f_0^j), \quad (3.8)$$

$$d(f_i, f_0^j) = \frac{f_i/f_0^j - [f_i/f_0^j]}{[f_i/f_0^j]}, \quad (3.9)$$

para  $j \in \{1, 2, \dots, N'\}$ .

Assim, o algoritmo de MPE deste trabalho funciona fixando um valor constante para  $N'$  e buscando exaustivamente  $N'$  valores para os *pitches* do quadro que maximizam a Equação (3.1). Para isso, é necessário não só definir um espaço de busca de tamanho tratável, pois trata-se de um problema de explosão combinatorial, mas também definir uma forma de escolher um valor plausível para  $N'$ .

Observe que os valores de amplitude,  $(A_1, A_2, \dots, A_P)$ , dos picos significativos foram ignorados nesta primeira etapa de estimação das frequências fundamentais. Conseqüentemente, os *pitches* em cada quadro são estimados utilizando-se apenas a informação frequencial dos picos do espectro segundo a Equação (3.8).

Além disso, da mesma forma que o exemplo da Seção 2.6, este método de MPE também sofre do típico problema de *overfitting* presente nos métodos de estimação por máxima verossimilhança. Isto significa que, ao adicionarmos mais um parâmetro livre  $f_0^{N'+1}$  em nosso espaço de parâmetros  $\mathbf{f}_0$ , tornamos nosso modelo mais complexo e, devido à operação de minimização da Equação (3.8), a nova função de verossimilhança será no mínimo equivalente à aplicada em um modelo de dimensão inferior, ou seja,

$$p(\mathbf{O} | f_0^1, f_0^2, \dots, f_0^{N'}) \leq p(\mathbf{O} | f_0^1, f_0^2, \dots, f_0^{N'}, f_0^{N'+1}). \quad (3.10)$$

Como já sabemos, o valor máximo para  $N'$  é a quantidade total de fontes misturadas no sinal, i.e.,  $N$ . Assim, para evitar que o algoritmo estime sempre  $N$  frequências fundamentais independentemente da informação existente no quadro, utiliza-se o Critério de Informação Bayesiano (BIC) para estimar a quantidade de notas simultâneas e, portanto, a complexidade do modelo que será utilizado em cada

quadro do sinal. Lembrando da definição apresentada na Equação (2.32),

$$\text{BIC}(N') = \ln p(f_1, f_2, \dots, f_P | f_0^1, f_0^2, \dots, f_0^{N'}) - \frac{1}{2} N' \ln P, \quad (3.11)$$

e o valor de  $N'$  em um quadro com  $P$  picos significativos será o que maximiza a Equação (3.11),

$$N' = \max_n \text{BIC}(n), \quad \text{com } n \in \{1, 2, \dots, N\}. \quad (3.12)$$

Analiseemos mais detalhadamente o comportamento da Equação (3.7) considerando, para facilitar os cálculos, que exista um  $f_0^l$  tal que cada um dos  $P$  picos do quadro seja exatamente um de seus harmônicos ideais, ou seja,

$$\frac{f_i}{f_0^l} = \left[ \frac{f_i}{f_0^l} \right], \quad \forall i \in \{1, 2, \dots, P\}. \quad (3.13)$$

Nesse caso, os valores encontrados para as distâncias relativas dos picos e da verossimilhança final podem ser calculados como

$$d(f_i, f_0^l) = \frac{f_i/f_0^l - [f_i/f_0^l]}{[f_i/f_0^l]} = 0, \quad (3.14)$$

$$d^2(f_i) = 0, \quad \forall i \in \{1, 2, \dots, P\}, \quad (3.15)$$

$$p(f_i | \mathbf{f}_0 = f_0^l) = 1 \quad \Rightarrow \quad p(\mathbf{O} | \mathbf{f}_0 = f_0^l) = 1. \quad (3.16)$$

Em outras palavras, segundo os cálculos, podemos afirmar que  $f_0^l$  será estimada como uma frequência fundamental do quadro se  $f_0^l$  estiver presente no vetor utilizado como espaço de busca exaustiva do algoritmo de MPE.

No entanto, se além de  $f_0^l$  também existir no vetor de busca um divisor qualquer de  $f_0^l$ , representado por

$$f_0^L = \frac{f_0^l}{L}, \quad \text{para qualquer } L \in \mathbb{N}^*, \quad (3.17)$$

após os cálculos teremos

$$\frac{f_i}{f_0^L} = L \frac{f_i}{f_0^l} = L \left[ \frac{f_i}{f_0^l} \right], \quad \text{como } L \in \mathbb{N}^* : \frac{f_i}{f_0^L} = \left[ \frac{f_i}{f_0^l} \right], \quad (3.18)$$

$$d(f_i, f_0^L) = \frac{f_i/f_0^L - [f_i/f_0^L]}{[f_i/f_0^L]} = 0, \quad (3.19)$$

$$d^2(f_i) = 0, \quad \forall i \in \{1, 2, \dots, P\}, \quad (3.20)$$

$$p(f_i | \mathbf{f}_0 = f_0^L) = 1 \quad \Rightarrow \quad p(\mathbf{O} | \mathbf{f}_0 = f_0^L) = 1. \quad (3.21)$$

Portanto, para o algoritmo, a chance de tanto  $f_0^l$  quanto qualquer um de seus divisores  $f_0^L$  serem uma frequência fundamental no quadro é de 100 %. Por isso, deve-se evitar que existam valores de frequência múltiplos inteiros no vetor de busca exaustiva que será utilizado pelo algoritmo. Nesse sentido, utilizam-se apenas alguns valores ao redor dos picos do espectro  $f_i$  como possíveis valores de frequência fundamental do sistema, isto é, como o espaço (vetor) de busca exaustiva.

Este modo de definir o vetor de busca parte do princípio de que o primeiro harmônico de cada nota existente no quadro, ou seja, a própria frequência fundamental de cada uma delas, será obrigatoriamente estimulado no espectro, independentemente do instrumento que o emitiu. Desse modo, o primeiro harmônico sempre estará presente em todas as estruturas harmônicas dos quadros, não sendo recomendada, portanto, a utilização do método com instrumentos harmônicos que não gerem um pico na frequência fundamental da nota musical que emitirem.

Para quadros onde não é detectada a presença de picos, como os exemplificados pelas Figuras 3.2c e 3.2d, é razoável inferir que não existe nenhum instrumento sendo tocado no momento e, portanto, não há frequências fundamentais nem estruturas harmônicas a serem estimadas. Já para o caso geral, o algoritmo é apresentado passo a passo nos tópicos abaixo:

- 1) Com  $i \in \{1, 2, \dots, P\}$ , converta os picos significativos do espectro,  $f_i$ , para a escala MIDI de acordo com a Equação (2.18), gerando as frequências  $F_i$  na nova unidade de medida.
- 2) Defina o raio de busca de *pitches* ao redor de cada pico como metade de um

semitom, isto é.,

$$r = 0,5 \text{ MIDI.} \quad (3.22)$$

- 3) Amostre valores de frequência nos intervalos  $[F_i - r, F_i + r]$  com período de 0,1 MIDI.
- 4) Concatene os valores amostrados, formando um vetor com todos os possíveis valores para as frequências fundamentais, porém, ainda na escala MIDI.
- 5) Converta o vetor de volta para Hertz com a Equação (2.19), gerando o vetor  $\mathbf{f}^0$ , sobre o qual será feita a busca exaustiva:

$$\mathbf{f}^0 = \{f_1^0, f_2^0, f_3^0, \dots, f_Q^0\}, \quad (3.23)$$

onde  $Q$  é a quantidade total de elementos em nosso espaço de busca.

★ Não se deve confundir o vetor de busca,  $\mathbf{f}^0$ , com o vetor de *pitches*,  $\mathbf{f}_0$ . O segundo representa as  $N'$  frequências fundamentais que serão estimadas pelo algoritmo. Já o primeiro representa todos os  $Q$  possíveis valores de frequência dentre os quais iremos procurar por  $\mathbf{f}_0$ . Independentemente do valor utilizado para a polifonia  $N'$  do quadro,

$$\mathbf{f}_0 \subset \mathbf{f}^0. \quad (3.24)$$

- 6) Faça  $N' = 1$ . Busque pelo valor de  $f_q^0$  que maximize a Equação (3.1), e guarde o valor de  $f_0^1$  como  $f_q^0$ :

$$f_0^1 \leftarrow f_q^0. \quad (3.25)$$

- 7)  $N' \leftarrow N' + 1$ .

- 8) Busque pelo valor de  $f_q^0$  que maximize a Equação (3.1), e guarde o valor de  $f_0^{N'}$  como  $f_q^0$ :

$$f_0^{N'} \leftarrow f_q^0; \quad (3.26)$$

- 9) Repita 7) – 8) até que  $N' = N$ .
- 10) Escolha como valor final para a polifonia  $N'$  do quadro o valor da complexidade do modelo que maximize a Equação (3.11). Os *pitches* estimados no quadro serão  $\{f_0^1, f_0^2, \dots, f_0^{N'}\}$ :

$$\mathbf{f}_0 \leftarrow \{f_0^1, f_0^2, \dots, f_0^{N'}\}. \quad (3.27)$$

Nesse algoritmo, o número de notas simultâneas  $N'$  e seus valores de frequências fundamentais podem acabar sendo estimados incorretamente em alguns quadros devido à baixa complexidade do algoritmo e à desconsideração dos valores da amplitude dos picos significativos do espectro. Porém, os resultados ainda são satisfatórios para a etapa de criação e agrupamento de estruturas harmônicas que virão a seguir. As estimações finais e retificadas serão realizadas na Subseção 3.3.1.

### 3.2.3 Extração de Harmônicos e Criação das HS

Após a estimação das frequências fundamentais de cada quadro, seus harmônicos devem ser extraídos com o intuito de criar estruturas harmônicas (HS) dos instrumentos. A seguir, damos a visão geral do algoritmo que tem essa finalidade.

- 1) Assuma como 20 a quantidade máxima de harmônicos possíveis em qualquer estrutura harmônica, conforme a definição na Equação (2.7).
- 2) Sejam  $f_0^j$ , com  $j = \{1, 2, \dots, N'\}$ , as frequências fundamentais estimadas no quadro. Seus  $r$ -ésimos harmônicos podem ser definidos como

$$f^r = r f_0^j. \quad (3.28)$$

A cada quadro, verifique em cada um dos intervalos

$$[0,997r f_0^j, 1,003r f_0^j], \quad \text{para } r \in \{1, 2, \dots, 20\}, \quad (3.29)$$

se existe algum pico significativo. Caso não exista, assumo o respectivo harmônico como desprezível, e coloque 0 dB em sua respectiva posição na Equação (2.7).

★ Note que foi utilizado o mesmo desvio padrão de meio semitom para verificar a presença dos harmônicos no sinal.

3) Crie  $N'$  estruturas harmônicas (HS) a cada quadro definidas de acordo com a Equação (2.7). Para isto, basta concatenar os harmônicos encontrados no passo anterior.

4) Normalize as HS de maneira que a energia total de seus harmônicos seja 100 dB;

★ Essa normalização é importantíssima para a etapa de agrupamento que virá a seguir, pois dessa forma, as estruturas harmônicas de um mesmo instrumento calculadas em diferentes quadros estarão garantidamente bem próximas no espaço de estruturas, independentemente da energia total de cada quadro.

5) As HS com quantidade de harmônicos menores do que 5 são excluídas, com base na observação de que o som proveniente de instrumentos harmônicos possui, em geral, pelo menos 5 harmônicos [1].

É importante lembrar que no espectro de um sinal polifônico alguns harmônicos provenientes de diferentes notas podem coincidir. Por isso, o valor da magnitude de alguns picos do espectro é fruto da influência coletiva dos harmônicos coincidentes. Como consequência, as estruturas harmônicas encontradas não são exatas. Entretanto, como a relação entre as notas não se mantém constante durante toda a música, a relação entre os harmônicos coincidentes também varia de quadro a quadro — em outras palavras, se o  $r$ -ésimo harmônico de um instrumento coincide com um harmônico qualquer de outro instrumento em um dos quadros, muito provavelmente em muitos outros quadros isto não acontecerá da mesma maneira; será possível, portanto, aprendê-lo corretamente depois da etapa de agrupamento.

No entanto, se durante grande parte da música a relação entre dois instrumentos se mantiver constante e houver coincidência de harmônicos entre as notas em muitos quadros, o algoritmo cometerá um erro no cálculo das estruturas e, mesmo após o agrupamento, a estimativa ficará polarizada em alguns harmônicos específicos.

Outro caso especial acontece quando há dois ou mais instrumentos tocando as mesmas notas em oitavas diferentes durante a maior parte da música. O algoritmo

utilizado não conseguirá distingui-los, pois, além de todos os harmônicos do instrumento da oitava superior coincidirem com os da inferior e causarem uma grande polarização em ambas as estruturas calculadas após o agrupamento, é muito difícil que o próprio algoritmo de estimação de frequências fundamentais estime corretamente a polifonia  $N'$  do quadro, pois ao utilizar 2 em vez de 1 para o valor de  $N'$ , a penalização relativa à complexidade do modelo seria muito maior, mas a verossimilhança permaneceria constante. Nesse caso, seria estimada apenas a frequência fundamental inferior nos quadros onde essa relação entre os instrumentos estivesse presente, pois o valor do BIC seria maior.

### 3.2.4 Agrupamento para Estimação das AHS

Seguindo ainda o artigo [1], o algoritmo de agrupamento (clusterização) implementado foi o chamado Agrupamento NK (em inglês *Neighborhood Knowledge Clustering*), originalmente proposto em [27]. Esta subseção apresenta sua explicação detalhada.

De uma maneira geral, este algoritmo de agrupamento infere informações sobre a densidade de pontos na vizinhança de cada ponto da base de dados ao calcular uma matriz de covariância local. Assim como é muito comum em algoritmos de análise de componentes principais (PCA), o estudo do comportamento dos autovalores dessa matriz nos dá informação acerca da densidade de pontos nas diferentes regiões do espaço. A este tipo de informação o autor dá o nome de *Knowledge*, originando o nome do algoritmo. Por fim, diferentes grupos são formados ligando-se os pontos vizinhos com densidade local parecida.

Considere as seguintes definições:

- A entrada do algoritmo é uma base de dados  $X$ , com  $M$  pontos em  $\mathbb{R}^d$ ,

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}, \quad \text{com } \mathbf{x}_i \in \mathbb{R}^d. \quad (3.30)$$

- Sejam  $\mathbf{x}_i$  e  $\mathbf{x}_j$  dois pontos aleatórios do espaço de entrada. Sabe-se que  $\mathbf{x}_j$  é o  $p$ -ésimo vizinho (calculado usando distância euclidiana) de  $\mathbf{x}_i$  e que  $\mathbf{x}_i$  é o  $q$ -ésimo vizinho de  $\mathbf{x}_j$ . Podemos definir o que chamamos de valor de vizinhança

mútua [28], ou MNV (do inglês *Mutual Neighborhood Value*) como

$$L_{ij} = p + q, \quad \text{com } p, q \geq 1. \quad (3.31)$$

★ Note que o MNV é capaz de nos dar uma informação acerca das características locais de cada ponto do espaço. Um MNV pequeno entre dois pontos significa que muito provavelmente os pontos pertencem a um mesmo grupo, pois estão bem próximos um do outro. Já um MNV alto deve implicar que pertençam a grupos distintos; mesmo se considerarmos o caso extremo em que  $p$  é muito pequeno e  $q$  é muito grande, muito provavelmente  $\mathbf{x}_i$  poderá ser considerado um ponto de ruído em nosso espaço de dados. Basta perceber que existiriam muitos pontos próximos de  $\mathbf{x}_j$ , porém, muito poucos ao redor de  $\mathbf{x}_i$ . Dessa forma, é muito mais interessante utilizar o MNV como uma métrica de distância para o algoritmo de agrupamento do que a distância euclidiana simples.

- A  $K$ -vizinhança de  $\mathbf{x}_i$ , encontrada usando os  $K$  pontos com os menores MNV em vez das menores distâncias euclidianas, é definida como<sup>2</sup>

$$\omega_i = \{\mathbf{x}_i, \mathbf{x}_i^1, \dots, \mathbf{x}_i^K\}, \quad \forall i \in \{1, \dots, M\}, \quad (3.32)$$

onde  $K$  é o número de vizinhos de  $\mathbf{x}_i$  presentes na vizinhança e  $\mathbf{x}_i^j$  (lê-se  $\mathbf{x}_i$  índice  $j$ ), com  $j \in \{1, \dots, K\}$ , representa o  $j$ -ésimo vizinho mais próximo de  $\mathbf{x}_i$ . Para facilitar a notação, podemos considerar  $\mathbf{x}_i^0 = \mathbf{x}_i$  e rescrever a Equação (3.32) como

$$\omega_i = \{\mathbf{x}_i^0, \mathbf{x}_i^1, \dots, \mathbf{x}_i^K\}. \quad (3.33)$$

- O centroide da vizinhança  $\omega_i$ , isto é, o seu ponto central, é definido como

$$\bar{\omega}_i = \frac{1}{K+1} \sum_{j=0}^K \mathbf{x}_i^j. \quad (3.34)$$

---

<sup>2</sup>Atenção com a notação utilizada aqui;  $j$  não é um expoente de  $\mathcal{X}_i$  quando escrevemos  $\mathcal{X}_i^j$ . Trata-se apenas do índice  $j$  de  $\mathcal{X}_i$ . Essa notação será utilizada em toda a seção.



- A matriz de covariância local do ponto  $\mathbf{x}_i$  é definida como

$$\mathbf{S}_{\mathbf{x}_i} = \frac{1}{K+1} \sum_{j=0}^K (\mathbf{x}_i^j - \bar{\boldsymbol{\omega}}_i) (\mathbf{x}_i^j - \bar{\boldsymbol{\omega}}_i)^\top. \quad (3.35)$$

★ Observe que cada ponto da base de dados possui sua própria matriz de covariância local, cujo cálculo é feito levando-se em consideração apenas seus  $K$  vizinhos mais próximos.

- O vetor com os  $d$  autovalores da matriz de covariância local de  $\mathbf{x}_i$  dispostos em ordem decrescente é definido como

$$\boldsymbol{\lambda}_{\mathbf{x}_i} = [\lambda_{\mathbf{x}_i}^1, \dots, \lambda_{\mathbf{x}_i}^d]^\top, \quad \text{com } \lambda_{\mathbf{x}_i}^1 \geq \lambda_{\mathbf{x}_i}^2 \geq \dots \geq \lambda_{\mathbf{x}_i}^d. \quad (3.36)$$

★ Observe que, como cada ponto da base de dados possui sua própria matriz de covariância local, cada ponto também possuirá seu próprio vetor de autovalores, o qual é responsável por nos dar uma informação acerca da densidade de pontos em volta de  $\mathbf{x}_i$ . É interessante que os autovalores sejam bem pequenos, pois, dessa forma, garantimos que os pontos da vizinhança de  $\mathbf{x}_i$  estão bem próximos dele mesmo. A este vetor é dado o nome de *knowledge*, ou, em tradução livre, vetor informação.

- A inadaptabilidade da vizinhança  $\boldsymbol{\omega}_i$  é definida como

$$a_{\boldsymbol{\omega}_i} = \frac{1}{d} \sum_{j=1}^d \frac{\lambda_{\mathbf{x}_i}^j}{\bar{\lambda}_i^j}, \quad (3.37)$$

onde  $\bar{\lambda}_i^j$  é o valor médio de todos os  $j$ -ésimos elementos de  $\boldsymbol{\lambda}_{\mathbf{x}_l}$ , para  $\mathbf{x}_l \in \boldsymbol{\omega}_i$ . Em outras palavras,  $\bar{\lambda}_i^j$  é a média dos  $j$ -ésimos autovalores das matrizes de covariância de todos os pontos da vizinhança de  $\mathbf{x}_i$ .

$$\bar{\lambda}_i^j = \frac{1}{K} \sum_{l=1}^K \lambda_{\mathbf{x}_i^l}^j. \quad (3.38)$$

Antes de explicar detalhadamente suas etapas, é importante citar que o algoritmo de agrupamento NK foi proposto com o intuito de funcionar em bases de dados ruidosas, onde alguns pontos estão espalhados de forma aleatória no espaço e

devem ser ignorados na hora do processamento.

Nesse sentido, logo após o cálculo dos  $K$  vizinhos mais próximos de cada ponto usando-se a métrica do MNV, existe um estágio responsável por verificar se um determinado ponto deve ser considerado como dado importante ou como ruído pelo algoritmo. Isto é feito utilizando-se os autovalores da matriz de covariância local (vetor informação) de cada ponto.

Como sabemos, os pontos de ruído geralmente são muito mais esparsos do que os pontos de dados significativos. Em outras palavras, os elementos do vetor informação dos pontos de ruído têm valores maiores se comparados com os elementos do vetor informação dos outros pontos do espaço de dados.

Tomando os  $M$  vetores informação da base de dados (há um total de  $M$  pontos), podemos definir sua matriz informação concatenando-os:

$$\mathbf{\Lambda}_X = \begin{bmatrix} \lambda_{\mathbf{x}_1}^1 & \lambda_{\mathbf{x}_2}^1 & \cdots & \lambda_{\mathbf{x}_M}^1 \\ \lambda_{\mathbf{x}_1}^2 & \lambda_{\mathbf{x}_2}^2 & \cdots & \lambda_{\mathbf{x}_M}^2 \\ \vdots & \vdots & \cdots & \vdots \\ \lambda_{\mathbf{x}_1}^d & \lambda_{\mathbf{x}_2}^d & \cdots & \lambda_{\mathbf{x}_M}^d \end{bmatrix}. \quad (3.39)$$

Calcula-se, então, um autovalor médio para a base de dados fazendo-se uma média nas linhas e nas colunas de  $\mathbf{\Lambda}_X$ ,

$$\bar{\lambda}_X = \mathbb{E}_d [\mathbb{E}_M [\mathbf{\Lambda}_X]], \quad (3.40)$$

onde  $\mathbb{E}_d$  e  $\mathbb{E}_M$  representam as médias nas  $d$  linhas e nas  $M$  colunas, respectivamente. Por fim, é definido um limiar de ruído como

$$\bar{\lambda}_{\text{ruído}} = \bar{\lambda}_X + P_{\text{ruído}} \bar{\sigma}_\lambda, \quad (3.41)$$

onde  $\bar{\sigma}_\lambda$  é o desvio padrão médio dos autovalores presentes na matriz informação da base de dados, isto é:

$$\bar{\sigma}_\lambda^2 = \mathbb{E}_d \left[ \mathbb{E}_M \left[ (\mathbf{\Lambda}_X - \bar{\lambda}_X)^2 \right] \right]. \quad (3.42)$$

Com este limiar podemos retirar alguns pontos da base de dados antes de

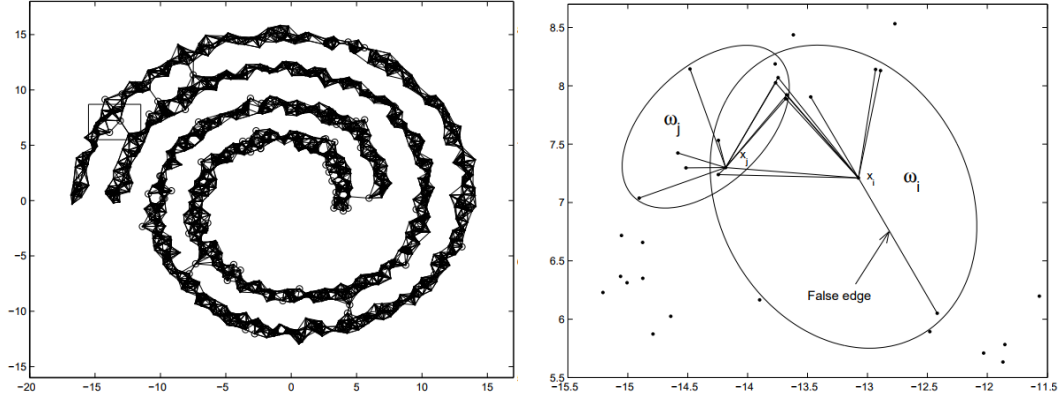
realizar o agrupamento, pois assumiremos serem estes pontos de ruído. Para isso, basta analisar o valor médio do vetor informação de cada ponto e comparar com um limiar na forma de um valor pré-definido para o parâmetro  $P_{\text{ruído}}$ . Se o valor médio passar do limiar, o ponto é considerado ruído e, portanto, pode ser ignorado pelo algoritmo.

Note que o parâmetro  $P_{\text{ruído}}$  controla a rigidez da análise. Caso não haja ruído nos dados, é recomendado utilizar um valor bem grande para o parâmetro (pelo menos 3), pois menos pontos serão ignorados com o limiar mais elevado. Entretanto, como o algoritmo vai ser utilizado para formar grupos de estruturas harmônicas, devemos utilizar valores entre 0 e 1 para o parâmetro, porque trata-se de um espaço de dados muito ruidoso, onde muitas estruturas podem ter sido incorretamente calculadas em alguns quadros.

Para encontrar grupos de pontos no espaço limpo (após a retirada dos pontos de ruído), parte-se de um ponto aleatório e liga-se este ponto a seus  $K$  vizinhos por arestas. Em seguida, para cada ponto neste grupo mais arestas devem ser formadas ligando-os também aos seus respectivos vizinhos. Este processo é feito até que não haja mais novos pontos vizinhos neste grupo. Em seguida, se ainda há pontos fora de grupos na base de dados, o processo é reiniciado partindo agora de um desses outros pontos, e formam-se novos grupos. O número total de grupos encontrados será determinado automaticamente pelo algoritmo, não sendo, portanto, controlado pelo usuário.

Entretanto, é muito comum que durante o agrupamento, algumas arestas sejam formadas de forma equivocada, isto é, que o algoritmo crie uma aresta falsa entre dois pontos, pois os considera vizinhos, mas eles na verdade não deveriam ser. Um exemplo é ilustrado pela Figura 3.3.

Quando uma aresta falsa é criada, é importante detectá-la para que o agrupamento seja feito de forma adequada. Observe a Figura 3.3b:  $\mathbf{x}_i$  é um ponto com aresta falsa e  $\omega_i$  é sua respectiva  $K$ -vizinhança. Já  $\mathbf{x}_j$  é um vizinho de  $\mathbf{x}_i$  que não possui arestas falsas, e  $\omega_j$  é a sua  $K$ -vizinhança correspondente. É fácil perceber que os elementos de  $\lambda_i$  tendem a ser maiores do que os elementos de  $\lambda_j$ . É exatamente esta informação que devemos levar em consideração para encontrar as vizinhanças que possuem arestas falsas.



(a) Formação de grupos utilizando a métrica MNV (b) Uma das arestas falsas é mostrada em detalhe. Ela deve ser quebrada.

Figura 3.3: Exemplo de agrupamento com a criação de arestas falsas. Adaptado de [27].

É importante lembrar que podem existir grupos de diferentes quantidades e densidades de pontos no espaço, com autovalores bem pequenos se forem muito densos ou autovalores maiores se forem mais esparsos. Em ambos os casos, porém, os autovalores de cada ponto dentro de uma certa vizinhança sempre possuem valores relativamente próximos entre si. Nesse exemplo, há, claramente, uma certa deformidade: um dos pontos da vizinhança  $\omega_i$  possui autovalores muito maiores do que os de seus outros vizinhos.

Para encontrar as arestas falsas utiliza-se um outro limiar, desta vez baseado na inadaptabilidade definida na Equação (3.37), a qual mede a similaridade entre a vizinhança  $\omega_i$  e todas as vizinhanças  $\omega_l$  de seus pontos vizinhos. Se  $\omega_i$  e todos os  $\omega_l$  são similares,  $a_{\omega_i}$  é pequeno; caso contrário,  $a_{\omega_i}$  é grande.

Portanto, a vizinhança que possui arestas falsas pode ser detectada a partir do limiar

$$\bar{a}_{\text{aresta}} = \bar{a} + P_{\text{aresta}} \sigma_a, \quad (3.43)$$

onde  $\bar{a}$  e  $\sigma_a$  são o valor médio e o desvio padrão das inadaptabilidades, respectivamente, e  $P_{\text{aresta}}$  é um parâmetro de projeto.

Se  $a_i > \bar{a}_{\text{aresta}}$ , a vizinhança  $\omega_i$  possui uma aresta falsa. Agora é necessário descobrir qual das  $K$  arestas devemos quebrar. Para isso é utilizado um simples método de descida mais íngreme:

- 1) Calcule o centroide da vizinhança de acordo com a Equação (3.34).
- 2) Calcule as distâncias entre o centroide e cada ponto da vizinhança.
- 3) Elimine da vizinhança o ponto mais distante de seu centroide. A vizinhança possui agora  $K - 1$  pontos;
- 4) Recalcule a inadaptabilidade da vizinhança modificada.
- 5) Verifique se a inadaptabilidade é menor do que o limiar  $\bar{a}_{aresta}$ . Se for, pare; caso contrário, volte para o passo 1).

Note que o parâmetro  $P_{aresta}$  funciona de maneira semelhante ao  $P_{ruído}$ , controlando o quão rígido é o processo de quebra de arestas entre os pontos. Para os testes realizados agrupando as estruturas harmônicas, este parâmetro foi mantido constante em 2, o valor mais recomendado segundo [27]. No entanto, este fator não influencia muito no cálculo das estruturas harmônicas médias dos instrumentos, já que só utilizaremos apenas alguns dos grupos encontrados: os  $N$  (quantidade de fontes) grupos que possuírem as maiores quantidades de pontos.

Somente depois das etapas de eliminação de pontos de ruído e quebra de arestas falsas é que se ligam os pontos a seus vizinhos para formar os diferentes grupos no espaço. O algoritmo completo é resumido nos passos abaixo para um melhor entendimento:

- 1) Detecção dos  $K$  vizinhos de cada ponto da base de dados.
  - Calcule a matriz de distância euclidiana  $D$ , onde cada elemento  $d_{ij}$  é a distância do ponto  $\mathbf{x}_i$  ao ponto  $\mathbf{x}_j$ ;
  - Calcule a matriz MNV dos pontos, onde cada elemento  $L_{ij}$  é o MNV entre o ponto  $\mathbf{x}_i$  e  $\mathbf{x}_j$ ;
  - Encontre os  $K$  vizinhos mais próximos de cada ponto usando a matriz MNV.
- 2) Cálculo das características locais.
  - Calcule a matriz de covariância local  $\mathbf{S}_{\mathbf{x}_i}$  de cada ponto  $\mathbf{x}_i$  da base de dados;

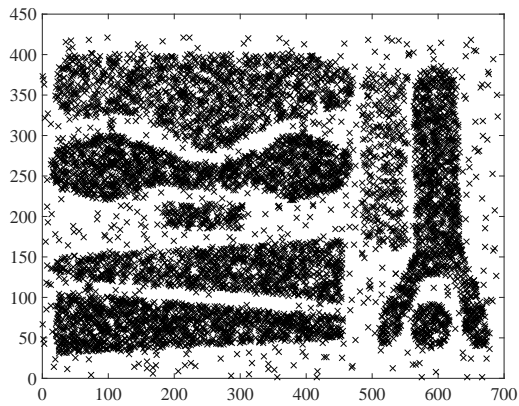
- Calcule seus autovalores e construa o vetor informação  $\boldsymbol{\lambda}_i$  de cada ponto  $\boldsymbol{x}_i$ .
- 3) Eliminação de pontos de ruído.
- Calcule o limiar  $\bar{\lambda}_{\text{ruído}}$  e elimine os pontos de ruído da base de dados;
  - Repita a Etapa 1) para a base de dados limpa, calculando novamente os  $K$  vizinhos de cada ponto significativo.
- 4) Quebra de arestas falsas.
- Recalcule o vetor informação  $\boldsymbol{\lambda}_i$  de cada ponto  $\boldsymbol{x}_i$ ;
  - Calcule a inadaptabilidade  $a_i$  de cada ponto  $\boldsymbol{x}_i$ ;
  - Calcule o limiar  $\bar{a}_{\text{aresta}}$  e encontre as vizinhanças  $\boldsymbol{\omega}_j$  com arestas falsas;
  - Elimine as arestas entre os pontos  $\boldsymbol{x}_j$  e  $\boldsymbol{x}_l$  — os pontos das vizinhanças  $\boldsymbol{\omega}_j$  mais distantes de seus centroides;
  - Recalcule as inadaptabilidades  $a_j$  das vizinhanças  $\boldsymbol{\omega}_j$  e verifique se estão abaixo do limiar  $\bar{a}_{\text{aresta}}$ ; caso ainda estejam acima, elimine mais arestas das vizinhanças.
- 5) Agrupamento dos pontos vizinhos.
- Ligue os pontos vizinhos por arestas e forme grupos no espaço.

Alguns testes para verificar o funcionamento do algoritmo de agrupamento implementado foram realizados em bases de dados em  $\mathbb{R}^2$ . Dessa forma, podemos visualizar os grupos formados com mais facilidade. A base de dados utilizada foi obtida de [29], e a Figura 3.4 mostra os resultados.

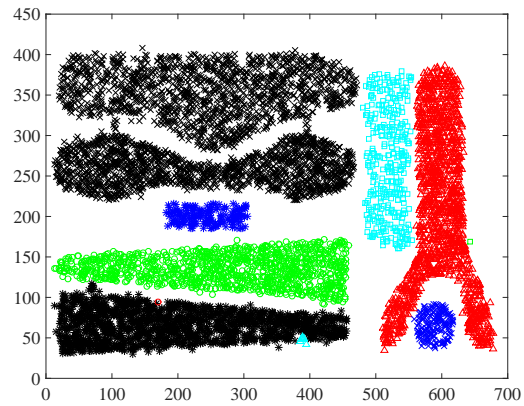
É conveniente lembrar que se deseja encontrar, ao fim do algoritmo, estimativas para  $N$  estruturas harmônicas médias (AHS) e suas respectivas instabilidades (HSI). Isto é feito calculando-se os centroides dos  $N$  grupos mais densos encontrados pelo algoritmo de agrupamento e seus respectivos desvios padrão.

Note que não temos controle sobre a quantidade de grupos que serão formados. Assim, é recomendado utilizar valores pequenos para  $K$  (entre 5 e 25) e procurar  $N$  grupos com significativamente mais pontos do que os outros. Caso o

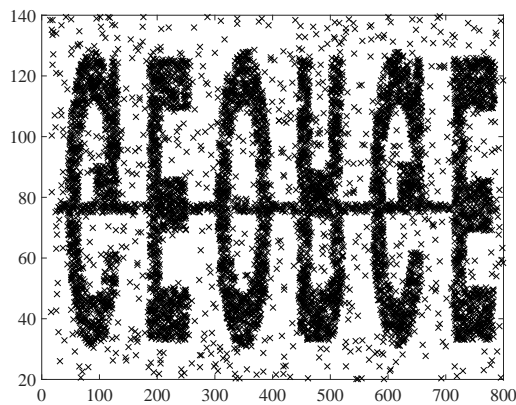
número de grupos totais seja menor do que  $N$ , diminui-se o valor de  $K$ . Diminuir o valor de  $P_{\text{ruído}}$  usualmente induz melhores resultados quando existem instrumentos percussivos ou vozes na mistura, pois são tratados como ruído de fundo e devem ser ignorados pelo algoritmo.



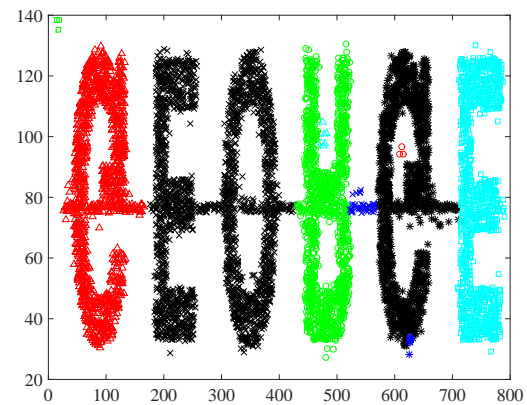
(a) Conjunto completos dos dados com 8000 pontos.



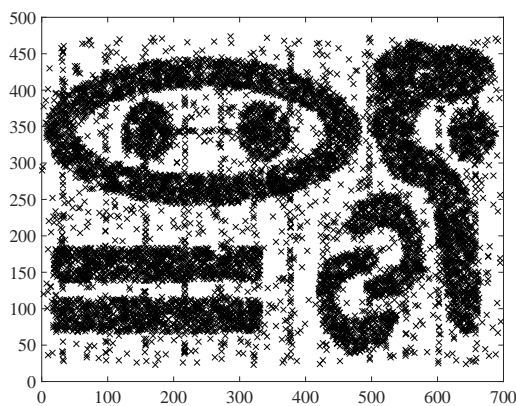
(b) Resultado para  $K = 6$ ,  $P_{\text{ruído}} = 0.8$  e  $P_{\text{aresta}} = 2$ .



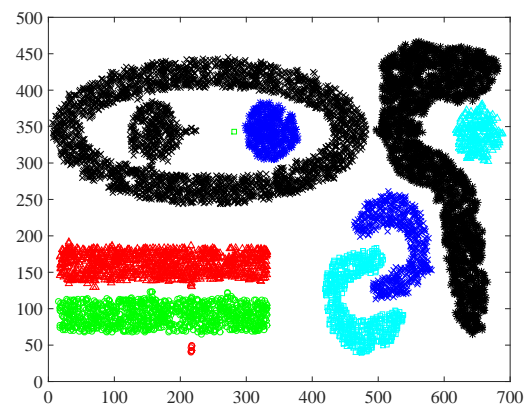
(c) Conjunto completo dos dados com 7700 pontos.



(d) Resultado para  $K = 7$ ,  $P_{\text{ruído}} = 0.5$  e  $P_{\text{aresta}} = 1$ .



(e) Conjunto completo dos dados com 10000 pontos.



(f) Resultado para  $K = 11$ ,  $P_{\text{ruído}} = 0.2$  e  $P_{\text{aresta}} = 2$ .

Figura 3.4: Exemplo de funcionamento do algoritmo de agrupamento NK em  $\mathbb{R}^2$ .

### 3.3 Separação

De posse das estimativas para as estruturas harmônicas médias de cada instrumento harmônico presente no sinal, finalizamos a fase de modelagem do processo. Dá-se início, então, à fase da separação propriamente dita, cujo principal objetivo é extrair as emissões sonoras de cada instrumento sem perder a qualidade dos sinais presentes na mistura original.

Esta seção explica como a separação é realizada e dá ao leitor todas as ferramentas necessárias para a reprodução da implementação do método, detalhando todos os passos dos algoritmos.

#### 3.3.1 Estimação Secundária das Frequências Fundamentais

O primeiro passo é obter melhores estimativas para as frequências principais das notas presentes na mistura. Isto pode ser feito seguindo-se o mesmo raciocínio e metodologia do algoritmo de MPE anterior; contudo, como agora dispomos das AHS das fontes, podemos usar também a informação da amplitude dos picos nos cálculos dos *itches*.

Neste caso, a função de verossimilhança é definida como

$$\begin{aligned} p(\mathbf{O}|f_0, \bar{\mathbf{B}}) &= p(f_1, f_2, \dots, f_P, A_1, A_2, \dots, A_P|f_0, \bar{\mathbf{B}}) \\ &= \prod_{i=1}^P p(f_i, A_i|f_0, \bar{B}_1, \bar{B}_2, \dots, \bar{B}_{20}), \end{aligned} \quad (3.44)$$

onde  $f_i$  e  $A_i$ , para  $i \in \{1, 2, \dots, P\}$ , são as frequências e amplitudes dos picos no quadro, respectivamente,  $\bar{\mathbf{B}} = [\bar{B}_1, \bar{B}_2, \dots, \bar{B}_{20}]$  é a estrutura harmônica média de um dos instrumentos da mistura e  $f_0$  é sua respectiva frequência fundamental.

Observe que, diferentemente da Equação (3.1), que só definia um valor máximo para a polifonia (quantidade de notas sendo emitidas simultaneamente) em cada quadro do sinal, a partir desta nova função de verossimilhança, obrigatoriamente, estima-se exatamente uma frequência fundamental por instrumento, a cada quadro. Assim, se existirem  $N$  fontes na mistura, necessariamente  $N$  frequências fundamentais serão estimadas por quadro.

A verossimilhança de cada pico  $(f_i, A_i)$  pode ser estimada usando a regra da



cadeia e lembrando-se que  $f_i$  é independente de  $\bar{\mathbf{B}}$ . Depois de alguns cálculos, é possível chegar a

$$p(f_i, A_i | f_0, \bar{\mathbf{B}}) = p(f_i | f_0, \bar{\mathbf{B}}) \cdot p(A_i | f_i, f_0, \bar{\mathbf{B}}) = p(f_i | f_0) p(A_i | f_i, f_0, \bar{\mathbf{B}}). \quad (3.45)$$

Assim como no algoritmo MPE anterior, a função  $p(f_i | f_0)$  é modelada como uma distribuição gaussiana também de variância  $\sigma_1^2 = 0,03$  dependente da distância relativa entre  $f_i$  e  $f^{[f_i/f_0]}$ . Portanto, seguindo as equações (3.3) e (3.4), podemos escrever

$$p(f_i | f_0) = C_1 e^{-\left(\frac{d^2(f_i, f_0)}{2\sigma_1^2}\right)}, \quad (3.46)$$

$$d(f_i, f_0) = \frac{f_i - f^{[f_i/f_0]}}{f^{[f_i/f_0]}} = \frac{f_i - [f_i/f_0] f_0}{[f_i/f_0] f_0} = \frac{f_i/f_0 - [f_i/f_0]}{[f_i/f_0]}. \quad (3.47)$$

Note que foram utilizadas novamente a constante  $C_1$  apenas para garantir área unitária sob o gráfico da função de verossimilhança  $p(f_i | f_0)$  e a notação  $[\mathcal{X}]$  para representar o número inteiro mais próximo de  $\mathcal{X}$ .

Já a outra verossimilhança,  $p(A_i | f_i, f_0, \bar{\mathbf{B}})$ , também será modelada como uma gaussiana, mas dependente de outra distância: a distância entre a amplitude normalizada do pico em  $f_i$ ,  $\hat{A}_i$ , e  $\bar{B}_{[f_i/f_0]}$ , que é a componente da AHS que corresponde ao harmônico ideal (relativo a  $f_0$ ) mais próximo de  $f_i$ . Matematicamente,

$$p(A_i | f_i, f_0, \bar{\mathbf{B}}) = C'_1 e^{-\left(\frac{D^2(\hat{A}_i, \bar{B}_{[f_i/f_0]})}{2\sigma_2^2}\right)}, \quad (3.48)$$

$$D^2(\hat{A}_i, \bar{B}_{[f_i/f_0]}) = \left(\hat{A}_i - \bar{B}_{[f_i/f_0]}\right)^2. \quad (3.49)$$

$C'_1$  é apenas o fator de normalização da verossimilhança (não confundir com a normalização do espectro de magnitude) e  $\sigma_2$  é a respectiva instabilidade (HSI) da estrutura harmônica média  $\bar{\mathbf{B}}$ .

A normalização do espectro de magnitude é necessária para garantir que o cálculo da distância seja realizado de maneira correta e sem polarização, colocando a energia efetiva do quadro exatamente igual à energia total da estrutura harmônica média. O termo “efetivo” é empregado no sentido de que nem todos os picos do

quadro foram necessariamente estimulados por um mesmo instrumento, pode haver mais de uma fonte sendo executada simultaneamente. Por isso, é importante verificar, antes da normalização e do cálculo da distância, quais picos do quadro devem ser considerados como efetivos, isto é, como possíveis picos provenientes de uma determinada AHS.

Nesse sentido, o primeiro passo antes de encontrar o valor da Equação (3.49) é verificar, para cada valor possível de  $f_0$  do vetor de busca, qual dos seus harmônicos ideais está mais próximo de cada pico em  $f_i$ .

Utilizando o mesmo vetor de busca exaustiva  $\mathbf{f}^0 = \{f_1^0, f_2^0, f_3^0, \dots, f_Q^0\}$  da Subseção 3.2.2, o algoritmo é detalhado a seguir para o  $q$ -ésimo valor possível de  $f_0$ , isto é, supondo que  $f_0 = f_q^0$ , para  $q \in \{1, 2, \dots, Q\}$ .

$$\mathbf{O} = [f_1, f_2, \dots, f_P] \iff [h_1, h_2, \dots, h_P], \quad (3.50)$$

$$h_i = \left[ \frac{f_i}{f_q^0} \right], \quad \forall i \in \{1, 2, \dots, P\}. \quad (3.51)$$

Aplicando-se a Equação (3.50), os picos  $f_i$  (em Hz) são transformados em  $h_i$  (valores inteiros). Dizemos que  $f_i$  é o  $h_i$ -ésimo harmônico de  $f_q^0$ , ou, simplesmente, que a harmonicidade de  $f_i$  é  $h_i$ .

Em seguida examinam-se quais picos deverão ser desconsiderados para o cálculo da normalização e da Equação (3.49), de acordo com o seguinte procedimento:

- Se algum  $h_i$  é menor do que 1, isto é, o pico  $f_i$  é bem menor do que o valor de  $f_q^0$  analisado, então sua magnitude,  $A_i$ , deve ser desconsiderada nos próximos cálculos;
- Se algum  $h_i$  é maior do que 20, isto é, o pico  $f_i$  é no mínimo o vigésimo primeiro harmônico de  $f_q^0$ , então sua magnitude,  $A_i$ , também deve ser desconsiderada;
- Se  $h_i = h_j$ , com  $i \neq j$ , isto é, existem dois picos bem próximos do mesmo harmônico ideal, então o pico mais distante do harmônico ideal deve ser também ignorado.

Se tudo foi realizado corretamente até aqui, alguns picos foram ignorados e os picos efetivos do quadro serão os que possuírem valores inteiros de 1 a 20 para suas

respectivas harmonicidades. Os valores de  $\hat{A}_i$  já podem ser calculados normalizando-se a energia total deste “espectro efetivo” para a energia total da estrutura harmônica média da fonte que será utilizada no cálculo.

Em seguida, a partir dos valores de  $\hat{A}_i$  e suas harmonicidades  $h_i$ , calculam-se os valores da Equação (3.49) fazendo

$$D^2(\hat{A}_i, \bar{B}_{[f_i/f_q^0]}) = \left( \hat{A}_i - \bar{B}_{h_i} \right)^2. \quad (3.52)$$

Por último, o comportamento da Equação (3.44) é analisado para todos os valores possíveis de  $f_0$  presentes no vetor de busca exaustiva  $\mathbf{f}^0$ , e o valor de  $f_m^0$  que a maximiza é utilizado como a estimativa final para a frequência fundamental relativa ao instrumento de AHS  $\bar{\mathbf{B}}$  no quadro.

Observe que o valor da polifonia de todos os quadros foi fixado em  $N$ , o número total de instrumentos harmônicos misturados. Entretanto, como sabemos que o valor da polifonia pode variar dependendo do número de instrumentos simultâneos, é utilizada no final do processamento a mesma polifonia  $N'$  dada pelo BIC da Subseção 3.2.2 como a verdadeira polifonia de cada quadro. Este procedimento não é citado em [1], mas foi utilizado no projeto na tentativa de melhorar o resultado da polifonia dos quadros. Os autores originais não deixaram muito claro qual procedimento deveria ser utilizado com este objetivo.

Assim, os  $N - N'$  instrumentos com os menores valores de  $p(\mathbf{O}|f_0, \bar{\mathbf{B}})$  são considerados não existentes no quadro, isto é, com frequência fundamental igual a zero.

Por fim, com o intuito de evitar erros provocados por mudanças abruptas dos valores estimados para as frequências principais, as estimativas passam por uma cascata de um filtro mediana de tamanho 3 com outro de tamanho 7, assim como sugerido em [1].

Além disso, um detalhe de implementação ainda proposto por Duan et al. e que melhora o resultado final das estimativas do algoritmo de MPE deve ser citado. Como há a multiplicação de duas gaussianas para encontrar a função de verossimilhança do sistema, é recomendado utilizar um valor mínimo para cada uma delas na tentativa de prevenir uma penalização muito severa de um dos fatores

no cálculo do produto. Assim, utilizam-se

$$d^2(f_i, f_q^0) = \min \left( \left( \frac{f_i/f_q^0 - [f_i/f_q^0]}{[f_i/f_q^0]} \right)^2, 4\sigma_1^2 \right) \quad \text{e} \quad (3.53)$$

$$D^2(\hat{A}_i, \bar{\mathbf{B}}) = \min \left( \left( \hat{A}_i - \bar{B}_{[f_i/f_q^0]} \right)^2, 4\sigma_2^2 \right) \quad (3.54)$$

como as verdadeiras funções de distância do algoritmo. Repare que a operação de minimização dessas duas equações faz com que o sistema se torne mais tolerante a picos  $(f_i, A_i)$  que estiverem com  $f_i$  fora do raio de um semitom ( $2\sigma_1$ ) de um harmônico ideal de  $f_q^0$  ou com  $\hat{A}_i$  fora do raio de duas HSI ( $2\sigma_2$ ) de seu valor ideal descrito na AHS.

### 3.3.2 Extração Retificada dos Harmônicos

Cada estimativa de  $f_0$  está associada a uma estrutura harmônica média de uma das fontes. Nesta etapa, utilizamos essas duas informações para extrair os harmônicos necessários à reconstrução das emissões sonoras de cada instrumento harmônico presente no sinal.

A extração é feita de uma maneira semelhante à descrita na Subseção 3.2.3, utilizando o mesmo intervalo para verificar se o  $r$ -ésimo harmônico de  $f_0$  está presente no espectro, isto é,  $f_i$  é o  $r$ -ésimo harmônico de  $f_0$  se e somente se

$$\exists r \in \{1, 2, \dots, 20\} \mid f_i \in [0,97r f_0, 1,03r f_0]. \quad (3.55)$$

Entretanto, há um processamento adicional para verificar se o harmônico efetivamente pertence ao instrumento e deve ser realmente extraído.

Supondo que um instrumento harmônico com AHS  $\bar{\mathbf{B}}$  emitiu uma nota com *pitch* estimado em  $f_0$  no quadro em questão, o procedimento é detalhado a seguir.

- 1) Tome o espectro de magnitude do quadro e normalize sua energia total de acordo com a energia total de  $\bar{\mathbf{B}}$ .
- 2) Utilize a Equação (3.55) e encontre os picos que são harmônicos de  $f_0$ . Caso mais de um pico represente o mesmo  $r$ -ésimo harmônico, ignore os mais distantes do valor ideal  $r f_0$ .

- 3) Verifique se o harmônico correspondente em  $\bar{\mathbf{B}}$  é igual a zero. Se este for o caso, o pico não foi provocado por este instrumento, e deve ser ignorado e mantido no espectro da mistura.
- 3) Compare o valor absoluto da diferença entre as log-magnitudes dos picos efetivos do espectro (normalizados conforme Passo 1)) e as respectivas log-magnitudes dos harmônicos em  $\bar{\mathbf{B}}$ :
  - Se o valor for igual ou inferior à HSI do instrumento, o pico foi provocado somente por esta fonte. Extraia-o do espectro da mistura utilizando toda a sua amplitude original (antes da normalização feita no Passo 1). Esta será a amplitude final deste harmônico no espectro de magnitude relativo ao instrumento de AHS  $\bar{\mathbf{B}}$  neste quadro.
  - Se o valor for superior à HSI do instrumento, o pico foi, provavelmente, provocado por mais de um instrumento. Extraia-o do espectro da mistura utilizando a amplitude resultante da normalização da energia total de  $\bar{\mathbf{B}}$  para a energia total original dos harmônicos (antes da normalização feita no Passo 1). Esta será a amplitude final deste harmônico no espectro de magnitude relativo ao instrumento de AHS  $\bar{\mathbf{B}}$  neste quadro.

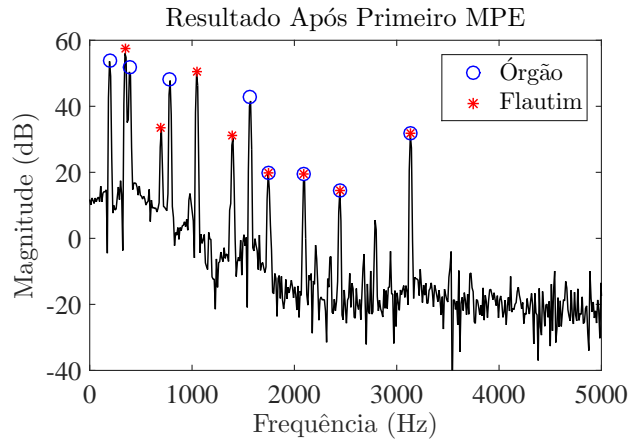
Depois de efetuar o procedimento acima em todos os quadros do espectrograma da mistura, é possível obter uma estimativa do espectrograma de magnitude de uma das fontes harmônicas (instrumento de AHS  $\bar{\mathbf{B}}$ ) misturadas.

O procedimento pode ainda ser repetido  $N - 1$  vezes com o objetivo de extrair os harmônicos provenientes dos outros instrumentos. Para isso, basta tratar o espectrograma residual como uma nova mistura e refazer os Passos 1) a 3) utilizando uma nova estrutura harmônica média associada a uma das outras fontes.

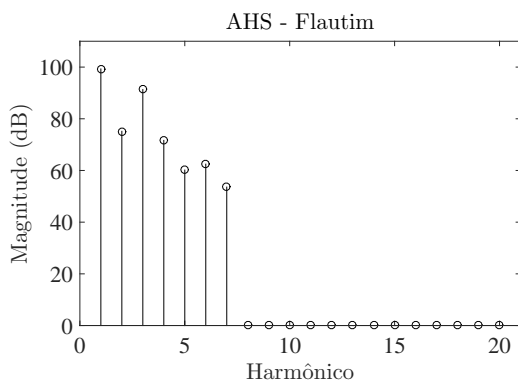
No fim do processamento, é possível obter, então, estimativas para  $N$  espectrogramas de magnitude, cada um relativo a um dos instrumentos harmônicos; e uma estimativa para o espectrograma de magnitude residual, onde podem ainda estar presentes os sinais provenientes de instrumentos não-harmônicos além dos resíduos remanescentes da separação.

Vale a pena notar que esta segunda etapa de extração de harmônicos conserta muitos erros causados pela etapa de extração anterior (Subseção 3.2.3), a qual utili-

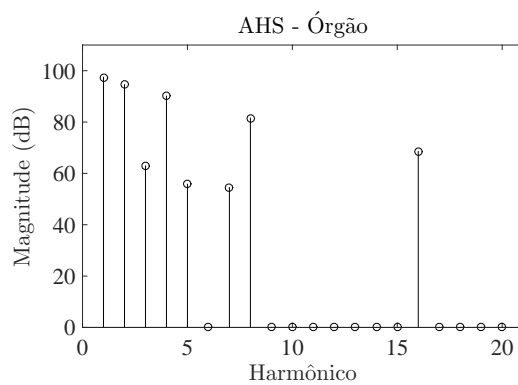
zou apenas a informação frequencial dos picos do espectro. Um exemplo é ilustrado em detalhes na Figura 3.5, que compara os resultados das duas etapas de extração de harmônicos aplicadas a uma mistura com duas fontes harmônicas monofônicas.



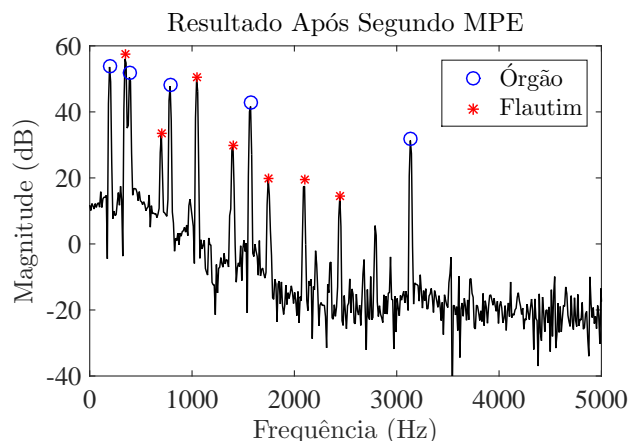
(a) Harmônicos extraídos conforme Subseção 3.2.3 para a formação das HS do quadro.



(b) Estimativa para a AHS do flautim de acordo com a Subseção 3.2.4.



(c) Estimativa para a AHS do órgão de acordo com a Subseção 3.2.4.



(d) Harmônicos extraídos de acordo com a Subseção 3.3.2 para a formação das estimativas do sinal de cada fonte.

Figura 3.5: Diferenças dos resultados das duas etapas de extração de harmônicos.

Observe que os últimos 4 picos da mistura que haviam sido equivocadamente associados às duas fontes presentes (Figura 3.5a) são tratados de maneira correta após a segunda etapa de extração de harmônicos (Figura 3.5d). O ajuste é possível devido à utilização das estruturas harmônicas médias (Figuras 3.5b e 3.5c) aprendidas em etapas intermediárias durante o algoritmo de separação.

### 3.3.3 Reconstrução dos Sinais das Fontes

Para criar estimativas dos espectrogramas complexos de cada fonte, utilizam-se os espectrogramas de magnitude encontrados na Subseção 3.2.3 junto com a informação de fase proveniente da mistura.

Os espectrogramas complexos são finalmente convertidos para o domínio do tempo com o uso da ISTFT [16] (do inglês *Inverse Short-Time Fourier Transform*) e a técnica do *overlap-add* [16].

Formam-se, então, no fim do processamento,  $N + 1$  sinais no domínio do tempo. Entre eles,  $N$  estão associados às estimativas das emissões sonoras de cada instrumento harmônico presente e o outro representando a forma de onda residual, a qual possui também estimativas para as fontes não-harmônicas, como instrumentos percussivos e/ou vozes que, por apresentarem variância muito alta, não se recomenda representar a partir de estruturas harmônicas médias.

# Capítulo 4

## Testes e Resultados

Este capítulo descreve ao leitor os experimentos realizados durante o projeto com o objetivo de avaliar o desempenho do algoritmo de separação implementado. Há também a exposição e análise dos resultados obtidos, bem como comentários a respeito das limitações do método.

Todos os testes realizados tiveram como principal objetivo a extração completa do som dos instrumentos harmônicos presentes nas misturas, conservando a qualidade sonora original. No entanto, é importante lembrar que as etapas intermediárias do algoritmo de separação implementado também geram estimações para os *pitches* de cada instrumento da mistura; por isso, julgou-se conveniente fazer também uma breve análise desses resultados.

Antes de dar início ao capítulo é importante reforçar que todos os materiais necessários para a execução dos experimentos do projeto foram provenientes do laboratório de Sinais, Multimídia e Telecomunicações (SMT) da Universidade Federal do Rio de Janeiro (UFRJ), destacando-se o uso da base de dados de alta qualidade RWC [20] para análise tímbrica de instrumentos.

### 4.1 Sinais de Teste

Os sinais utilizados nas simulações foram os mesmos presentes no artigo original de Duan et al. [1] para uma melhor referência na comparação dos resultados. O autor os disponibiliza livremente em [30]. Esses sinais são sinais de música mono (contêm um único canal) pré-gravados ou sintetizados a uma taxa de 22.050 Hz.



Neles podem estar misturados os sons de um ou mais instrumentos musicais e/ou vozes.

A Tabela 4.1 mostra os detalhes de cada uma das misturas utilizadas nas simulações, bem como os valores dos parâmetros fornecidos ao algoritmo de agrupamento. Caso o leitor queira acessar os arquivos de áudio, todos os experimentos podem ser encontrados em [31].

Tabela 4.1: Tabela com as informações dos sinais de teste.

Nome do Arquivo	$N$	Natureza	Tipo	$K$	$P_{\text{ruído}}$	$P_{\text{aresta}}$
FlautimOrgao.wav	2	Sint.	H	30	0,2	2
EufonioOboe.wav	2	Nat.	H	5	0,2	2
OboeFem.wav	1	Sint.	H+V	4	0,1	2
FlautimFem.wav	1	Sint.	H+V	6	0,1	2
OrgaoMasc.wav	1	Sint.	H+V	10	0,1	2

Legenda:

- $N$ : Total de fontes harmônicas misturadas;
- Natureza: Se os sinais são provenientes de instrumentos sintéticos ou naturais;
- Tipo: Se os sinais possuem instrumentos harmônicos (H) e/ou vozes (V);
- $K$ : Número de vizinhos utilizados pelo algoritmo Agrupamento NK (Subseção 3.2.4);
- $P_{\text{ruído}}$ : Parâmetro que controla a rigidez da detecção de pontos ruidosos pelo algoritmo de agrupamento (Equação (3.41)),
- $P_{\text{aresta}}$ : Parâmetro que controla a rigidez da detecção de arestas falsas pelo algoritmo de agrupamento (Equação (3.43)),

## 4.2 Avaliação Geral dos Experimentos

Como sabemos, a modelagem tímbrica dos instrumentos harmônicos usando o artifício da estrutura harmônica média ignora todas as partes não-estacionárias de suas emissões sonoras. Isso faz com que o método de separação não inclua os ataques dos instrumentos nos seus respectivos sinais separados. Mesmo nos casos em que o algoritmo de MPE funciona bem, gerando estruturas harmônicas bem próximas das que seriam encontradas se utilizarmos as fontes originais separadamente, ao

ouvirmos o sinal residual dos experimentos fica nítida a presença dos ataques de cada nota presente na mistura, bem como outros tipos de defeitos indesejados.

Há também claramente erros causados por falhas cometidas pelo algoritmo de MPE utilizado. Em alguns experimentos (como será facilmente percebido no sinal `FlautimOrgao.wav`), existem trechos do sinal separado nos quais certas notas que foram tocadas originalmente pela fonte alvo não estão presentes, exemplificando uma subestimação do valor da polifonia (número total de frequências fundamentais) de alguns quadros da mistura. Já em outros, a polifonia é até estimada corretamente, mas os valores das frequências fundamentais dos instrumentos podem ser estimados de maneira equivocada. Isto é agravado quando os sinais possuem efeitos de *vibrato* e reverberação, elementos muito comuns em sinais gravados com instrumentos reais. A mistura `Eufonio0boe.wav` é um exemplo onde isso ocorre.

## 4.3 Resultados

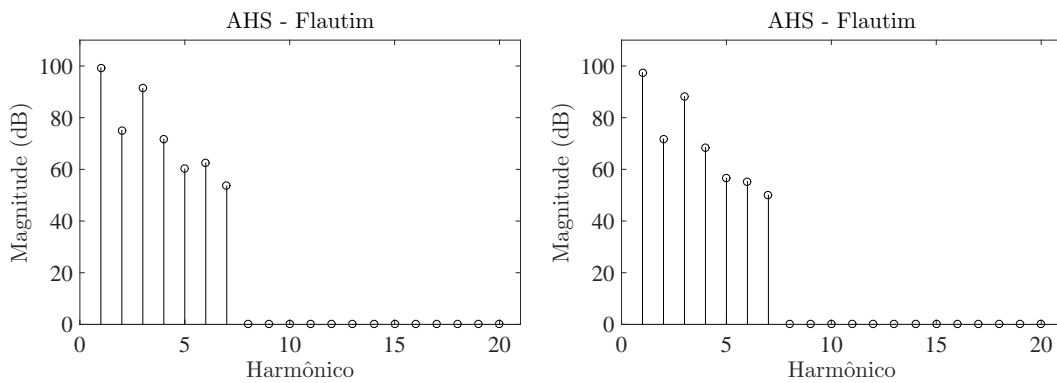
De acordo com a Tabela 4.1, cinco simulações foram feitas com o objetivo de testar as limitações e avaliar a qualidade das separações obtidas pelo método implementado. Esta seção relata cada um deles e mostra os resultados obtidos em cada mistura.

### 4.3.1 Experimento #1 - `FlautimOrgao.wav`

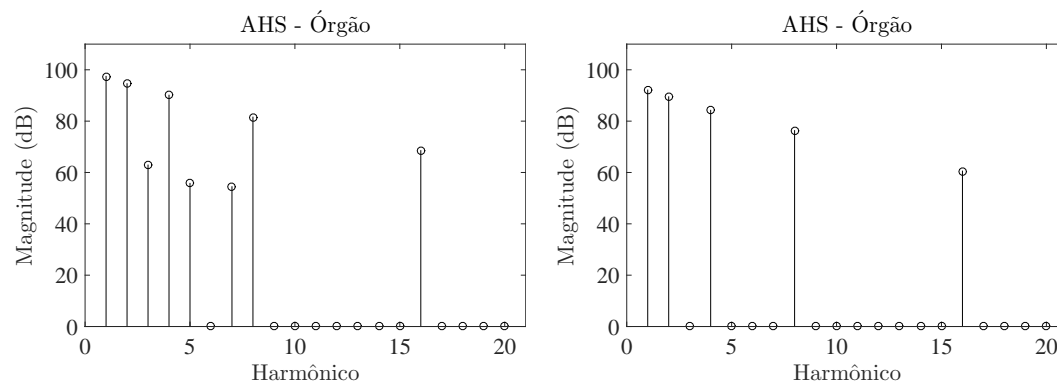
Este sinal é uma mistura de órgão e flautim sintetizada a partir de seus modelos MIDI. Ele foi criado sem a adição de ruídos e ambos os instrumentos possuem aproximadamente a mesma energia.

As estimativas das estruturas harmônicas médias encontradas pelas simulações são as mesmas ilustradas nas Figuras 3.5b e 3.5c, que são reinseridas aqui por conveniência (Figuras 4.1a e 4.1c). Já as presentes em [1], o artigo que este projeto tenta reproduzir, aparecem nas Figuras 4.1b e 4.1d. Comparando-as, podemos perceber que os resultados dos dois algoritmos são muito parecidos, mesmo havendo diferenças consideráveis em ambas as implementações. Muitas decisões tomadas por Duan et al. não foram explicitadas no artigo original e, por isso, este projeto seguiu caminhos um pouco diferentes. Uma delas reflete-se nos harmônicos extras (terceiro,

quinto e sétimo) com valores diferentes de zero na AHS do órgão encontrada pelas simulações deste trabalho. Esse resultado foi fruto do algoritmo de detecção de picos utilizado no projeto, que, por ser ligeiramente diferente do algoritmo original apresentado em [1], é capaz de detectá-los como picos significativos nos espectros de magnitude de muitos quadros do sinal — enquanto o algoritmo de Duan et al. utiliza um filtro gaussiano de comprimento e variância não especificados no artigo para a suavização do espectro, o algoritmo implementado aplica uma média móvel de comprimento 9, como foi explicado na Subseção 3.2.1.



(a) Estimativa para a AHS do flautim a partir da análise da mistura com  $N = 2$ . (b) Estimativa de Duan et al. para a AHS do flautim. Adaptado de [1].



(c) Estimativa para a AHS do órgão a partir da análise da mistura com  $N = 2$ . (d) Estimativa de Duan et al. para a AHS do órgão. Adaptado de [1].

Figura 4.1: Comparação das AHS estimadas pelo projeto e por [1] para o experimento #1.

Como a mistura é livre de ruídos e as duas fontes presentes puderam ser modeladas usando-se o artifício da AHS, existem três maneiras diferentes de isolar os sinais provenientes de cada instrumento:

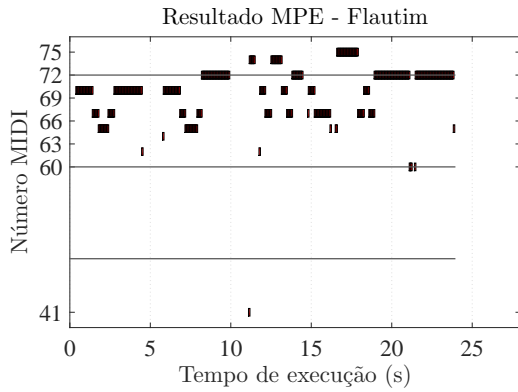
1. Extrair as emissões sonoras do flautim a partir da utilização de sua AHS e

- deixando o sinal do órgão como resíduo do processamento,
2. Extrair as emissões sonoras do órgão a partir da utilização de sua AHS e deixando o sinal do flautim como resíduo do processamento,
  3. Extrair ambos os instrumentos da mistura utilizando suas respectivas AHS.

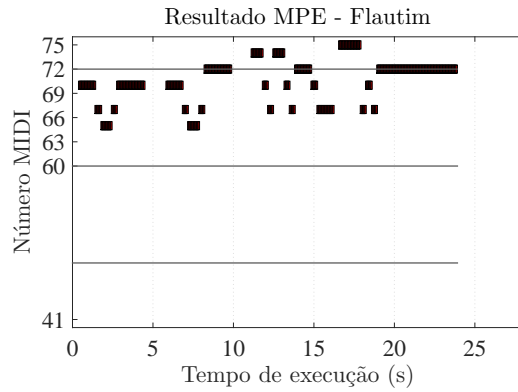
Assim como em [1], os três métodos tiveram um desempenho similar nas simulações deste projeto. Entretanto, o terceiro método foi o escolhido para a análise, visto que o sinal residual evidencia melhor as falhas da separação. Facilmente, ao ouvi-lo, é possível perceber a presença do ataque das notas dos dois instrumentos e outros resíduos do processo da separação, como, por exemplo, alguns harmônicos mais altos que foram ignorados e algumas notas que permaneceram sem estar associadas a nenhum dos instrumentos.

Os *pitches* estimados para cada instrumento podem ser vistos na Figura 4.2. Enquanto as Figuras 4.2a e 4.2c mostram os resultados do algoritmo de MPE aplicado no sinal misturado usando  $N = 2$ , as Figuras 4.2b e 4.2d servem como uma referência para o “melhor” resultado possível, pois mostram os resultados que foram calculados aplicando-se o método implementado em cada uma das fontes originais separadamente com  $N = 1$ . Se considerarmos os *pitches* presentes nas duas últimas como as verdadeiras frequências fundamentais dos instrumentos, podemos dizer que houve um acerto em 94,05 % dos quadros para as estimações dos *pitches* do flautim e em 93,54 % dos quadros para as estimações dos *pitches* do órgão.

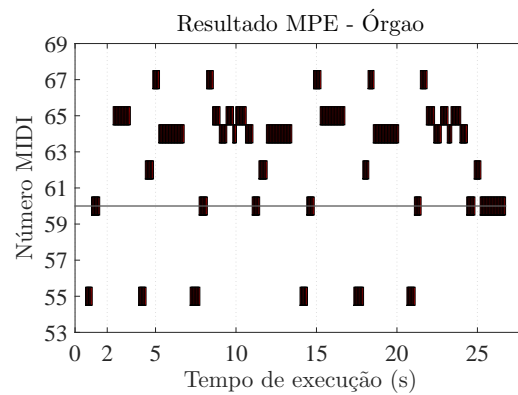
Vale a pena citar um fato curioso. Como podemos ver na Figura 4.2d, existe uma nota tocada pelo órgão com  $f_0 = 67$  MIDI e duração muito curta, por volta dos 2 segundos do sinal. O método implementado não conseguiu detectá-la e por isso, ao ouvirmos o sinal residual notamos facilmente que há uma nota com um timbre muito similar ao do órgão original esquecida junto aos resíduos da separação.



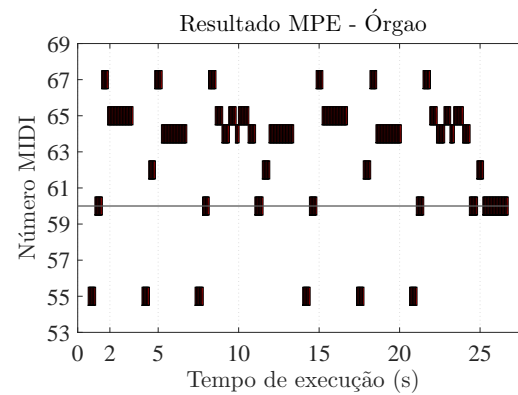
(a) Estimativa para os *pitches* do flautim a partir da análise da mistura com  $N = 2$ .



(b) Estimativa de referência para os *pitches* do flautim a partir da análise da fonte original com  $N = 1$ .



(c) Estimativa para os *pitches* do órgão a partir da análise da mistura com  $N = 2$ .



(d) Estimativa de referência para os *pitches* do órgão a partir da análise da fonte original com  $N = 1$ .

Figura 4.2: Comparação dos resultados do algoritmo de MPE no experimento #1. Figuras geradas usando o pacote MIDIToolbox 1.1 [32].

### 4.3.2 Experimento #2 - EufonioOboe.wav

Segundo [1], esta mistura possui dois instrumentos naturais gravados: o oboé (E5-F5) e o eufônio (G3-G4), ambos extraídos de CDs comerciais não correlacionados. Entretanto, possuem uma relação harmônica, além de elementos como *vibrato* e reverberação. O sinal também foi criado sem ruído utilizando uma razão de 2,3 dB de energia entre o eufônio e o oboé.

A Figura 4.3 mostra os resultados das estruturas harmônicas encontradas pelo método, bem como as encontradas por Duan et al. Já a Figura 4.4 mostra os resultados da estimação das frequências fundamentais.

Note que a principal diferença entre a AHS do eufônio encontrada pelo

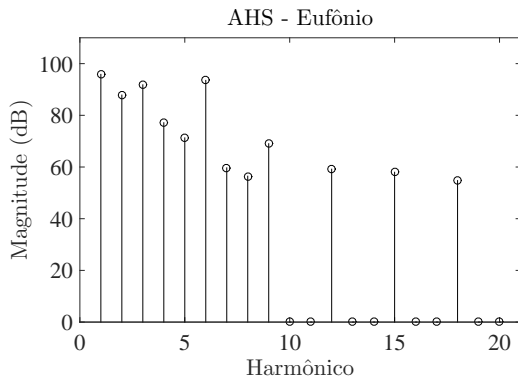
método implementado e a encontrada por [1] é a presença de harmônicos superiores. Isso aconteceu porque os sinais das fontes estão em oitavas muito distantes. Os maiores harmônicos do eufônio, que está numa oitava muito inferior, foram contaminados pelos harmônicos do oboé — é como se a presença do oboé na mistura enganasse o algoritmo e provocasse associações erradas de alguns harmônicos que deveriam ser do oboé com harmônicos superiores do eufônio.

Para exemplificar, observe a região após os 2,5 segundos na Figura 4.4. Durante esse trecho, com aproximadamente 2 segundos de duração, o algoritmo trata a mistura como se houvesse apenas um instrumento tocando. O  $f_0$  da nota do oboé é zero, quando na verdade deveria ser 76 MIDI, o que faz com que todos os seus harmônicos sejam associados apenas ao eufônio.

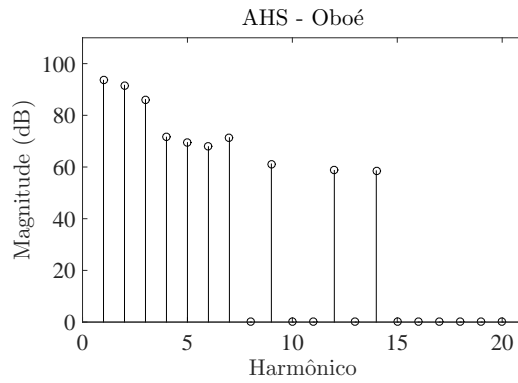
Isso é refletido também nos outros resultados da estimação dos *pitches* das fontes. Dificultados ainda pelo *vibrato* e pela reverberação presentes no sinal, os cálculos das frequências principais do oboé tiveram resultados muito ruins: o algoritmo de MPE erra em 100% dos quadros da mistura, conseguindo apenas detectar os *pitches* inferiores do eufônio.

Consequentemente, a estrutura harmônica média do oboé é aprendida de forma errada e, por isso, a separação fica com melhores resultados se extrairmos o sinal do eufônio da mistura e deixarmos o sinal do oboé como resíduo do processamento. Com a audição dos sinais, observa-se que o timbre do eufônio extraído fica mais agudo do que o original devido à presença indevida dos harmônicos superiores em sua AHS.

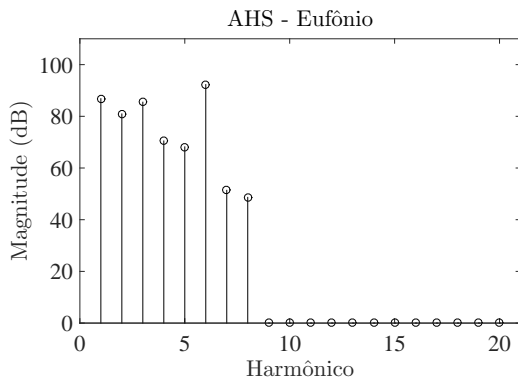
Duan et al. não disponibilizaram os seus resultados de MPE para esse experimento; por isso, não podemos comparar o comportamento do método original com o reproduzido. No entanto, é importante lembrar que algumas etapas do algoritmo de MPE proposto em [1] não estão muito claras, e o algoritmo implementado teve que sofrer algumas mudanças. Então, é muito provável que realmente exista uma diferença nos *pitches* estimados.



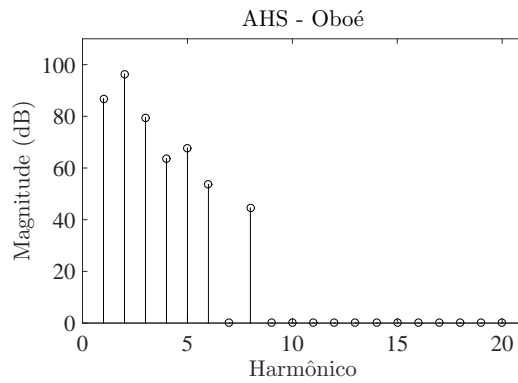
(a) Estimativa para a AHS do eufônio a partir da análise da mistura com  $N = 2$ .



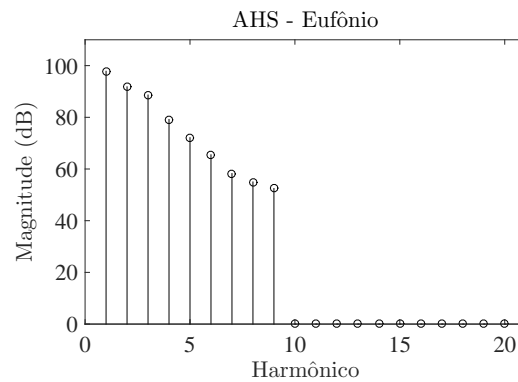
(b) Estimativa para a AHS do oboé a partir da análise da mistura com  $N = 2$ .



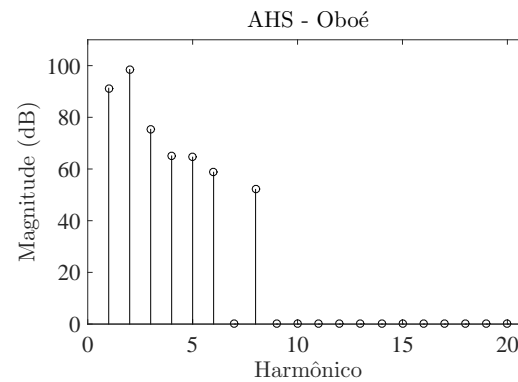
(c) Estimativa de Duan et al. para a AHS do eufônio. Adaptado de [1].



(d) Estimativa de Duan et al. para a AHS do oboé. Adaptado de [1].



(e) Estimativa de referência para a AHS do eufônio a partir da análise da fonte original com  $N = 1$ .



(f) Estimativa de referência para a AHS do oboé a partir da análise da fonte original com  $N = 1$ .

Figura 4.3: Comparação das AHS estimadas pelo projeto e por [1] para o experimento #2.

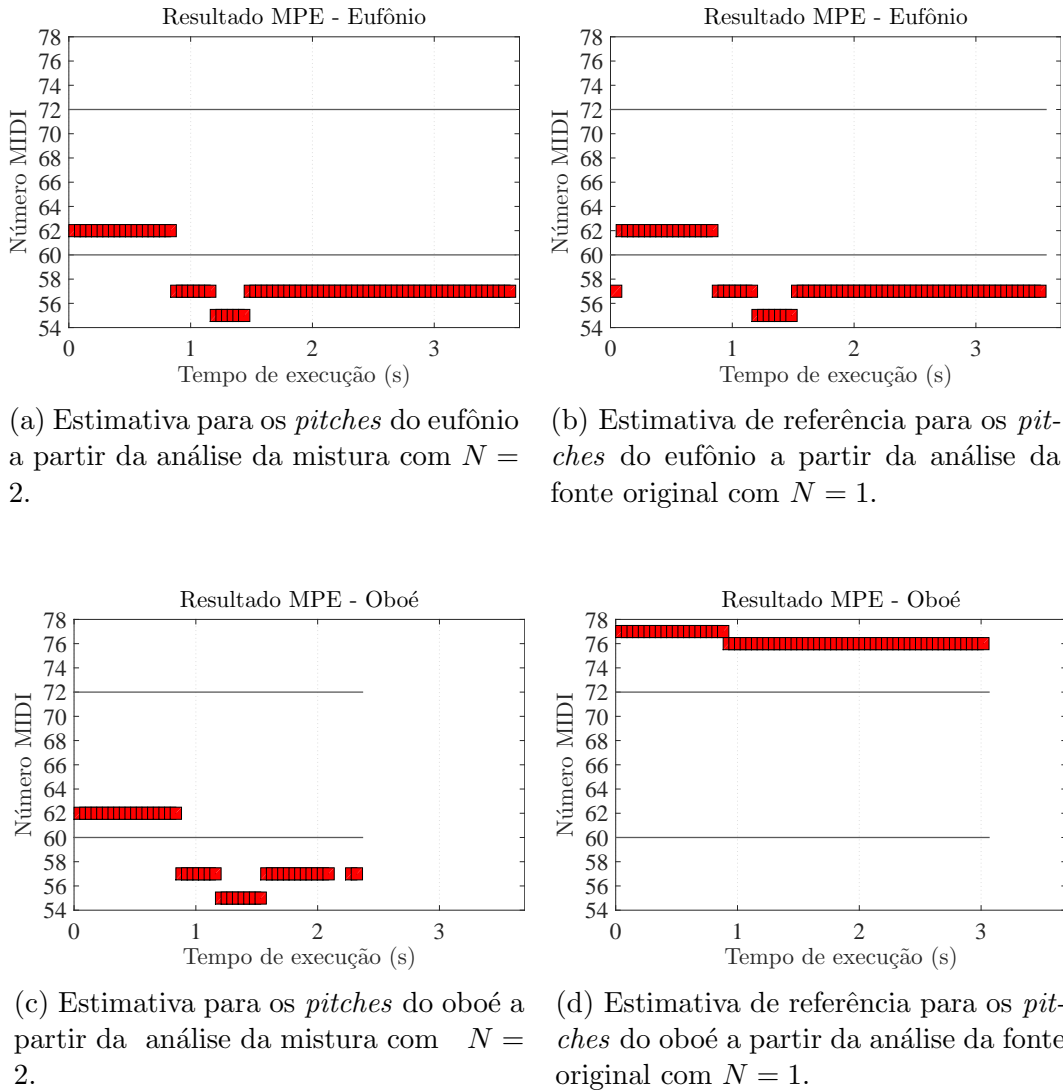


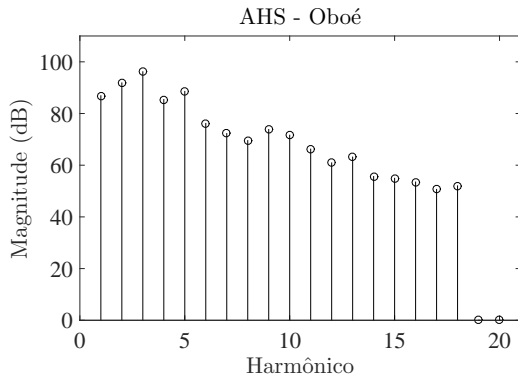
Figura 4.4: Comparação dos resultados do algoritmo de MPE para o experimento #2. Figuras geradas usando o pacote `MIDIToolbox 1.1` [32].

### 4.3.3 Experimento #3 - OboeFem.wav

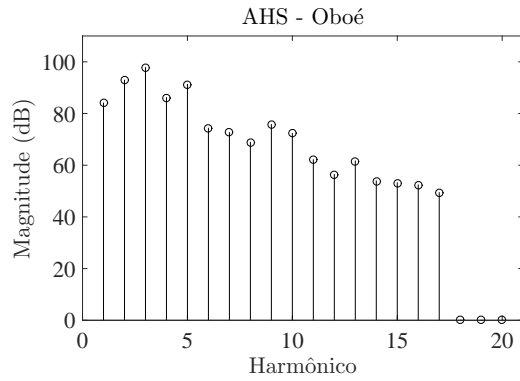
A mistura utilizada no terceiro experimento é sintetizada utilizando-se um sinal de voz feminina sem muita relação com o sinal do oboé. Apesar da forte presença de *vibratos* e de reverberação, o algoritmo conseguiu bons resultados. Isto foi possível porque o sinal de voz é tratado como um ruído de fundo, e a mistura é processada utilizando sempre  $N = 1$ . Repare que é justamente por isso que se utilizou um valor menor ainda para o parâmetro  $P_{\text{ruído}}$ .

A Figura 4.5 compara o método implementado atuando na mistura e no sinal original do instrumento. Podemos dizer que o método extraiu bem o sinal da fonte harmônica (oboé), deixando a voz no sinal residual.

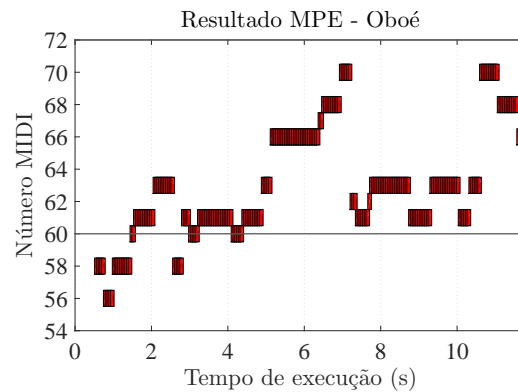




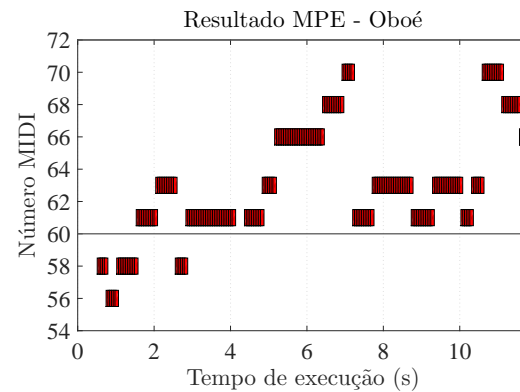
(a) Estimativa para a AHS do oboé a partir da análise da mistura com  $N = 1$ .



(b) Estimativa de referência para a AHS do oboé a partir da análise dada fonte original com  $N = 1$ .



(c) Estimativa para os *pitches* do oboé a partir da análise da mistura com  $N = 1$ .



(d) Estimativa de referência para os *pitches* do oboé a partir da análise da fonte original com  $N = 1$ .

Figura 4.5: Comparação das estimativas para a AHS do oboé e dos resultados do algoritmo de MPE no experimento #3. Figuras 4.5c e 4.5d geradas usando o pacote `MIDIToolbox 1.1` [32].

#### 4.3.4 Experimento #4 e Experimento #5

Esses dois experimentos são exemplos do caso especial explicado com mais detalhes no fim da Subseção 3.2.3. Trata-se de sinais nos quais existem fontes tocando exatamente as mesmas notas, mas em oitavas diferentes.

A mistura do Experimento 4 (`FlautimFem.wav`) é composta pelo sinal do flautim utilizado no primeiro experimento com a adição de uma voz feminina cantando a mesma melodia, mas na oitava inferior.

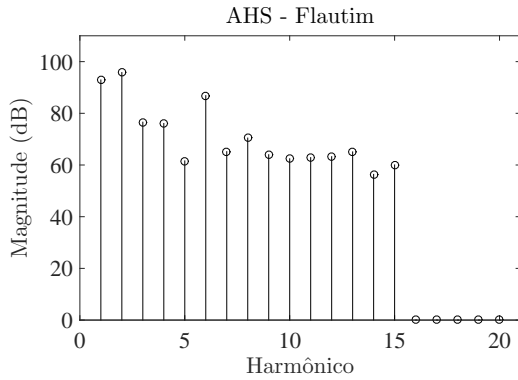
Já a mistura do Experimento 5 (`OrgaoMasc.wav`) foi feita adicionando-se uma voz masculina ao outro sinal do Experimento 1. Nesta mistura o sinal de voz também segue a melodia original tocada pelo órgão, emitindo as notas na oitava

anterior.

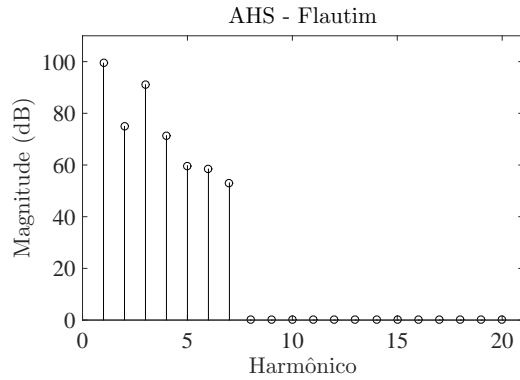
As Figuras 4.6 e 4.7 mostram seus respectivos resultados. Ao analisá-las é fácil perceber que houve uma nítida polarização nas estimativas das estruturas harmônicas médias devido à defasagem de exatamente 12 MIDI (uma oitava) nos valores estimados para os *itches* dos instrumentos. Em outras palavras, os sinais de voz mascaram os sinais dos instrumentos, pois, por estarem exatamente 12 semitons abaixo, todos os harmônicos provenientes dos instrumentos coincidem com os das respectivas vozes. Esse resultado foi previsto na Subseção 3.2.3.

Vale a pena citar que se pode consertar a polarização ao “decimar” por 2 as estimativas encontradas para as AHS. Isto faz sentido, uma vez que os *itches* estimados foram exatamente a metade dos originais.

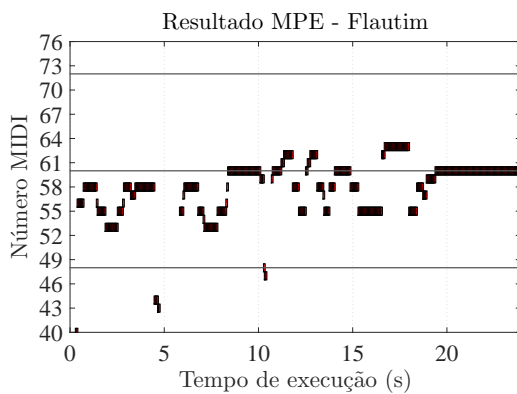
Ouvindo os sinais separados, percebemos que os instrumentos realmente ficaram mais graves; já as vozes, remanescentes no sinal residual, perderam a naturalidade.



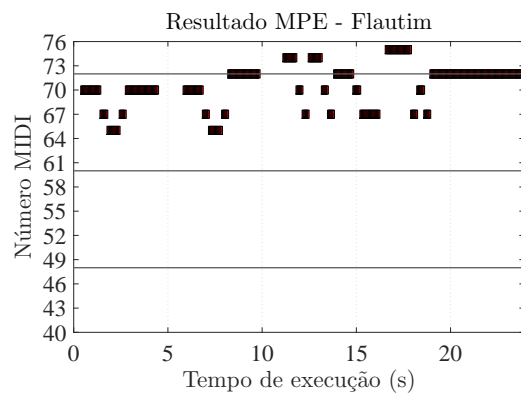
(a) Estimativa para a AHS do flautim a partir da análise da mistura com  $N = 1$ .



(b) Estimativa de referência para a AHS do flautim a partir da análise da fonte original com  $N = 1$ .

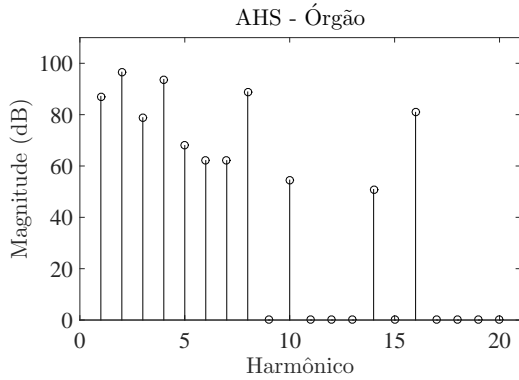


(c) Estimativa para os *pitches* do flautim a partir da análise da mistura com  $N = 1$ .

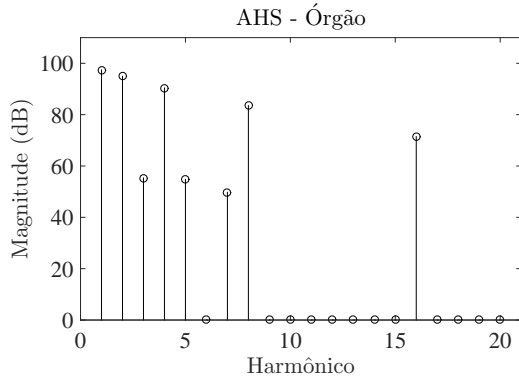


(d) Estimativa de referência para os *pitches* do flautim a partir da análise da fonte original com  $N = 1$ .

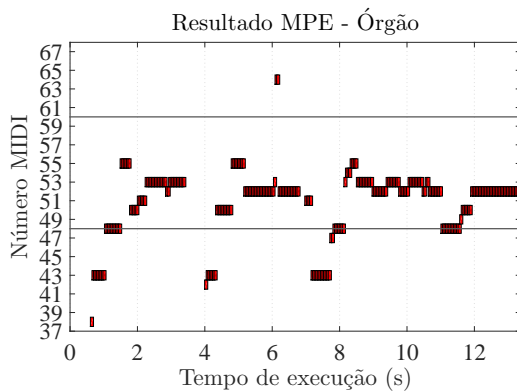
Figura 4.6: Comparação das estimativas para a AHS do flautim e dos resultados do algoritmo de MPE no experimento #4. Figuras 4.6c 4.6d geradas usando o pacote MIDIToolbox 1.1 [32].



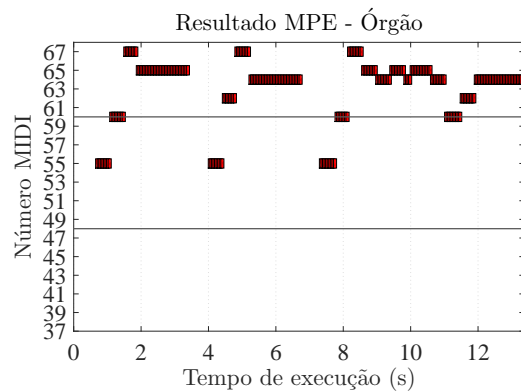
(a) Estimativa para a AHS do órgão a partir da análise da mistura com  $N = 1$ .



(b) Estimativa de referência para a AHS do órgão a partir da análise da fonte original com  $N = 1$ .



(c) Estimativa para os *pitches* do órgão a partir da análise da mistura com  $N = 1$ .



(d) Estimativa de referência para os *pitches* do órgão a partir da análise da fonte original com  $N = 1$ .

Figura 4.7: Comparação das estimativas para a AHS do órgão e dos resultados do algoritmo de MPE no experimento #5. Figuras 4.7c e 4.7d geradas usando o pacote MIDIToolbox 1.1 [32].

# Capítulo 5

## Conclusão

Este projeto foi uma tentativa de reprodução de um método alternativo [1] presente na literatura que realiza a separação e o isolamento não-supervisionados de instrumentos harmônicos monofônicos usando o conceito de estrutura harmônica média. Pôde-se concluir que, sob a condição de estarem tocando em estreitas faixas de frequência, as diferentes fontes instrumentais harmônicas presentes em um sinal de áudio composto possuíam estruturas harmônicas estáveis, ainda que distintas. Tal estrutura foi capaz de identificar cada instrumento presente na mistura e pôde ser usada para extrair suas emissões sonoras.

Nesse sentido, dado o número total de fontes harmônicas, o algoritmo analisado no projeto conseguiu aprender o modelo de cada instrumento diretamente do sinal misturado a partir do agrupamento das estruturas harmônicas extraídas de diferentes quadros. Suas fontes correspondentes foram, então, extraídas da mistura utilizando-se os modelos aprendidos.

Entretanto, há algumas limitações. O método não é capaz de lidar com misturas que possuem mais de uma fonte não-harmônica ou ruidosas, tais como instrumentos percussivos e vozes. Esses tipos de fontes não são bem representadas usando a AHS e permanecem misturadas no sinal residual após o processamento. Além disso, outra limitação é o fato de a estrutura harmônica conseguir modelar apenas a parte invariante no tempo das notas emitidas por um mesmo instrumento. Como consequência, o sinal residual fica com muitas componentes que o algoritmo não consegue associar à nenhuma das fontes (e que muitas vezes fazem parte delas). A presença de efeitos como reverberação e *vibrato* também geram padrões espectrais

variantes no tempo, e pioram os resultados do método.

No que diz respeito ao algoritmo de estimação de *pitches*, concluiu-se que o método conseguiu estimar os *pitches* das fontes instrumentais com certa flexibilidade, não só detectando automaticamente, em muitos casos, o número de notas simultâneas, mas também identificando corretamente as notas sobrepostas.

Como trabalho futuro, é proposta uma extensão do modelo da AHS que seja capaz de modelar instrumentos tocando notas com maiores extensões frequenciais, ou até mesmo a utilização de várias estruturas harmônicas diferentes para um mesmo instrumento dependendo da região frequencial da nota emitida. Outra ideia é a adição da informação temporal na análise, pois ela pode ser utilizada para melhorar a modelagem das partes variantes no tempo e tornar o método mais robusto a variações de *pitch*.

Para finalizar, podemos dizer que o tipo de abordagem para o problema da separação de fontes usado neste trabalho não é muito comum; por isso, não existe na literatura uma documentação muito clara para o método implementado. Assim, o texto deste relatório torna-se uma referência ideal para introdução ao tema e o ponto de partida para novos trabalhos, pois possui não só explicações teóricas, mas também uma descrição detalhada de todos os algoritmos implementados.

Os códigos implementados e todas as suas documentações estão disponíveis para *download* em [31].

# Referências Bibliográficas

- [1] Z. Duan, Y. Zhang, C. Zhang, and Z. Shi. Unsupervised single-channel music source separation by average harmonic structure modeling. *IEEE Transactions on Audio, Speech and Language Processing*, 16(4):766–778, May 2008.
- [2] P. Comon and C. Kitten, editors. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, Oxford, UK, 2010.
- [3] S. Haykin. *Unsupervised Adaptive Filtering: Blind source separation*, volume 1. Wiley, New York, USA, 2000.
- [4] I. M. Quintanilha. Algoritmos para a fatoração de matrizes não-negativas com aplicação em transcrição de instrumentos percussivos. Monografia de B. Sc, POLI/UFRJ, Rio de Janeiro, Brasil, 2016.
- [5] P. Z. Peebles. *Probability, Random Variables, and Random Signal Principles*. McGraw-Hill, New York, USA, 2000.
- [6] T. W. Lee, M. Girolami, A. J. Bell, and T. A. Sejnowski. A unifying information-theoretic framework for independent component analysis. *Computers & Mathematics with Applications*, 39(11):1–21, June 2000.
- [7] A. Hyvarinen. Gaussian moments for noisy independent component analysis. *IEEE Signal Processing Letters*, 6(6):145–147, June 1999.
- [8] E. Oja. Applications of independent component analysis. In N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal, and S. K. Parui, editors, *Neural Information Processing*, pages 1044–1051. Springer, Berlin, Germany, 2004.

- [9] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, USA, 2001.
- [10] T. Virtanen. Unsupervised learning methods for source separation in monaural music signals. In A. Klapuri and M. Davy, editors, *Signal Processing Methods for Music Transcription*, chapter 9, pages 267–296. Springer, New York, USA, 2006.
- [11] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–91, October 1999.
- [12] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley, Chichester, UK, 2009.
- [13] Tygel A. F. Métodos de fatoração de matrizes não-negativas para separação de sinais musicais. Dissertação de M. Sc, COPPE/UFRJ, Rio de Janeiro, Brasil, 2009.
- [14] S. Haykin and B. Van Veen. *Signals and Systems*. Wiley, New York, USA, 2nd edition, 2002.
- [15] L. Cohen. *Time Frequency Analysis: Theory and Applications*. Prentice Hall, Upper Saddle River, USA, 1994.
- [16] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto. *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, Cambridge, UK, 2nd edition, 2010.
- [17] M. Bosi and R. E. Goldberg. *Introduction to Digital Audio Coding and Standards*. Springer, Berlin, Germany, 2002.
- [18] H. Fastl and E. Zwicker. *Psychoacoustics: Facts and Models*. Springer Series in Information Sciences. Springer, Berlin, Germany, 3rd edition, 2006.
- [19] X. Serra. Musical sound modeling with sinusoids plus noise. In C. Roads, S. Pope, A. Piccialli, and G. De Poli, editors, *Musical Signal Processing*, Studies



- on New Music Research, chapter 8, pages 91–122. Swets & Zeitlinger, Lisse, The Netherlands, 1997.
- [20] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Music genre database and musical instrument sound database. In *International Symposium on Music Information Retrieval (ISMIR)*, number 4, pages 229–230, Barcelona, Spain, October 2003.
- [21] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information science and statistics. Springer, Berlin, Germany, 2006.
- [22] Y. S. Abu-Mostafa, M. Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data. A Short Course*. AMLBook, 2012.
- [23] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, March 1978.
- [24] J. O. Smith and X. Serra. PARSHL: An Analysis/Synthesis Program for Non-harmonic Sounds Based on a Sinusoidal Representation. In *Proceedings of the 1987 International Computer Music Conference (ICMC)*, 1987.
- [25] M. Christensen and A. Jakobsson. *Multi-Pitch Estimation*, volume 5 of *Synthesis Lectures on Speech and Audio Processing*. Morgan & Claypool, San Rafael, USA, 2009.
- [26] M. Davy, S. Godsill, and J. Idier. Bayesian analysis of polyphonic western tonal music. *The Journal of the Acoustical Society of America (JASA)*, 119(4):2498–2517, April 2006.
- [27] Y. Zhang, C. Zhang, and S. Wang. Clustering in knowledge embedded space. In *Transactions of the 14th European Conference on Machine Learning (ECML)*, pages 480–491, September 2003.
- [28] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

- [29] G. Karypis, Eui-Hong Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32(8):68–75, August 1999.
- [30] Z. Duan. AHS separation results. <https://sites.google.com/site/mperesult/musicseparationresults>. Acessado em: 01/09/2017.
- [31] C. Lordelo. Link com os arquivos de áudio utilizados nos experimentos e com os códigos implementados. <http://www.smt.ufrj.br/~carlos.lordelo/ProjetoFinal/>. Acessado em: 01/09/2017.
- [32] P. Toiviainen and T. Eerola. MIDI toolbox 1.1. <https://github.com/miditoolbox/1.1>, 2016. Acessado em: 01/09/2017.