Universidade Federal de Santa Catarina
Curso de Pós-Graduação em Matemática e
Computação Científica

Análise e Ressíntese de Sinais Musicais

Saulo Castilho

Florianópolis - SC
2008

Universidade Federal de Santa Catarina
Curso de Pós-Graduação em Matemática e Computação Científica

Análise e Ressíntese de Sinais Musicais

Dissertação apresentada ao Curso de Pós-Graduação em Matemática e Computação Científica, do Centro de Ciências Físicas e Matemáticas da Universidade Federal de Santa Catarina, para obtenção do grau de Mestre em Matemática, com Áera de Concentração em Básico em Matemática Aplicada.

Saulo Castilho
Florianópolis - SC
2008

Análise e Ressíntese de Sinais Musicais

Saulo Castilho

Essa Dissertação foi julgada adequada para a obtenção do Título de "Mestre", Área de Concentração em Básico em Matemática Aplicada, e aprovada em sua forma final pelo Curso de Pós-Graduação em Matemática e Computação Científica.

Clóvis Caesar Gonzaga (Coordenador)

Comissão Examinadora:

Prof. Dr. Licio Hernanes Bezerra (Orientador - UFSC)

Prof. Dr. Edson Cataldo (UFF)

Prof. Dr. Juliano de Bem Francisco (UFSC)

Prof. Dr. Paulo Rafael Bösing (UFSC)

30 de Maio de 2008

Agradecimentos

Agradeço aos meus pais que sempre tiveram minha formação acadêmica como prioridade em suas vidas, à minha namorada pelo apoio nos momentos difíceis, aos meus familiares pela confiança que depositaram em mim, aos meus amigos, ao orientador deste trabalho por trilhar junto comigo este desafio, à comissão examinadora pelas sugestões que enriqueceram este trabalho, aos meus colegas de mestrado pelo intercâmbio de conhecimento, à secretaria do curso de pós-graduação em matemática e computação científica, a todos os professores com quem tive o privilégio de aprender, à CAPES pelo apoio financeiro e ao meu estilo musical favorito, o heavy metal, pela inspiração.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

Esta dissertação apresenta um estudo sobre duas famílias de métodos para análise e ressíntese de sinais musicais: métodos no domínio do tempo (Iteração Ortogonal Adaptativa, Strobach, NP3 e OPAST) e métodos no domínio das frequências (Phase Vocoder, PARSHL e SMS). Começamos apresentando algumas características dos sinais sonoros e introduzimos alguns modelos para sua representação matemática. Depois, introduzimos métodos no domínio do tempo, que utilizam matrizes de Hankel com os dados do sinal, a partir das quais buscamos os parâmetros que o descrevem. Além dos métodos de Iteração Ortogonal e de Strobach, utilizamos implementações próprias dos algoritmos NP3 e OPAST em MATLAB. Em seguida, apresentamos resultados de testes realizados com sinais musicais monofônicos. Finalmente, após uma introdução à análise de Fourier, que fundamenta os métodos no domínio das freqüências apresentados a seguir, apresentamos resultados de testes com esses métodos. Para os métodos Phase Vocoder e SMS, são utilizadas implementações públicas e, para o método PARSHL, uma implementação própria, todas programadas em MATLAB.

Abstract

This dissertation presents a study on two families of methods for analysis and resynthesis of music signals: time domain methods (Adaptive Orthogonal Iterations, Strobach, NP3 and OPAST) and frequency domain methods (Phase Vocoder, PARSHL and SMS). First, we present some features about sound signals and some models for their mathematical representation. After, we present time domain methods that utilize Hankel matrices, from which we search parameters that describe the signal. Besides Adaptive Orthogonal Iterations and Strobach methods, we utilize personal implementations of NP3 and OPAST algorithms in MATLAB. After, we present some results from experiments with monophonic music signals. Finally, after an introduction to Fourier analysis, which is the background for the explanation of those frequency domain methods, we present results of experiments with those methods. For Phase Vocoder and SMS we use public implementations, and for PARSHL we use a personal implementation, all of them in MATLAB.

Introdução

O objeto do nosso trabalho é o processamento de sinais digitais, mais especificamente, de sinais de áudio, que é utilizado em áreas diversas, como reconhecimento de voz, reconhecimento de palavras, detecção de patologias vocais, automatização de sistemas de telefonia, softwares de edição musical, pedais de efeitos para instrumentos, entre outras.

O presente trabalho apresenta um estudo sobre alguns métodos para a análise e ressíntese de sinais musicais. Ainda que para muitos o título seja claro, é interessante definirmos bem os termos envolvidos, de forma a não deixar dúvidas sobre o que será feito.

O termo análise refere-se à extração e ao estudo de parâmetros que representam um determinado fenômeno. Para isso, deve-se possuir um conhecimento sobre o fenômeno a ser analisado e quais os parâmetros que o descrevem. Portanto, surge a necessidade de definirmos o que é um sinal musical e de que maneira podemos analisá-lo.

Utilizando o termo ressíntese, expressamos a intenção de reconstruir o fenômeno estudado através dos parâmetros que dele extraimos. Se, por outro lado, houvéssemos usado o termo "síntese", poderia surgir a interpretação de que tentaríamos gerar parâmetros para criar um fenômeno inexistente, neste caso, um sinal musical com o qual ainda não havíamos nos deparado. Tal processo também é de grande importância, sendo exemplos de sua aplicação a música eletrônica e efeitos sonoros. Entretanto, nos restringiremos aqui à reconstituição de sinais que conhecemos de antemão, utilizando os parâmetros obtidos após sua respectiva análise, sujeitos apenas a leves modificações intencionais em alguns casos, cuja importância explicaremos no devido momento.

Resta-nos definir o fenômeno que estudaremos e de que maneira podemos analisá-lo e ressintetizá-lo. O termo sinal é muito abrangente e possui várias interpretações. Em nosso caso, denominamos sinal uma função que descreve o comportamento de um determinado fenômeno. Logo, podemos interpretar sinal musical como uma função que descreve uma música. Por sua vez, música pode ser definida como a sucessão de sons e silêncio organizados ao longo do tempo. Portanto, a unidade que compõe a música é o som. Os

métodos de análise e síntese aqui apresentados podem ser utilizados para a descrição e reconstituição de sons de uma forma geral. Porém, a qualidade dos resultados obtidos dependerá de quão bem o som descrito pelo sinal se encaixará no modelo que utilizaremos para representá-lo.

No capítulo 1, introduziremos alguns conceitos de física, biologia, engenharia e computação, e transportaremos esses conceitos à matemática aplicada, por meio de diferentes modelagens. Também será citado um breve histórico do processamento de sinais digitais de áudio, com informações sobre o surgimento e evolução dos métodos que apresentaremos.

No capítulo 2, veremos os métodos de análise e síntese musical baseados no domínio do tempo, também conhecidos como métodos de alta resolução. Começaremos revisando conceitos básicos sobre matrizes, depois nos aprofundaremos no modelo no qual se baseiam esses métodos, e mostraremos formas de encontrarmos os parâmetros necessários para análise e ressíntese dos sinais musicais. Finalmente, estudaremos formas de otimização dos cálculos envolvidos nesses métodos e apresentaremos os resultados obtidos nos testes executados. Para esses métodos, utilizaremos apenas sinais quase harmônicos correspondentes a sons monofônicos, provenientes de instrumentos de orquestra e de curta duração.

No capítulo 3, apresentaremos métodos baseados no domínio de freqüências. Começaremos pela teoria de Fourier, com enfoque voltado à Transformada de Fourier, suas variações e propriedades. Em seguida, mostraremos através de exemplos simples como a Transformada de Fourier fornece o espectro de um sinal e como pode ser interpretado esse espectro, visando a extração dos parâmetros que procuramos. Por último, apresentaremos passo a passo os métodos estudados e os resultados dos testes realizados com eles. Para esses métodos, utilizaremos também sinais correspondentes a sons mais elaborados, obtidos a partir de diversos instrumentos. Finalmente, apresentamos alguns resultados de modificações de tempo e freqüência de alguns sinais.

Capítulo 1

Modelagem de Sinais Musicais

1.1 O Que é o Som?

O som é gerado através da vibração de um corpo, exercendo pressão sobre o ar e se propagando por esse meio em forma de ondas. Dessa forma, o som chega aos nossos ouvidos, onde há uma estrutura que recebe essas vibrações, as interpreta e as envia ao cérebro, gerando a percepção que temos do som. Nessa estrutura, destaca-se a membrana basilar, que é composta por conjuntos de células, cada um vibrando em resposta a uma determinada freqüência. Assim, a membrana basilar decompõe o som de acordo com as freqüências presentes nele.

Portanto, o entendimento do comportamento do som passa pelo estudo do comportamento das ondas e de como nosso organismo as recebe. As partículas presentes no meio onde uma onda se propaga não acompanham o movimento da onda, elas apenas vibram localmente e transmitem as vibrações às partículas vizinhas pelo contato.

Há quatro características que descrevem um som e que podem ser vistas tanto sobre o aspecto físico do comportamento da onda, quando sobre o aspecto da nossa percepção.

1. Amplitude: corresponde ao deslocamento total da partícula vibratória em relação ao seu ponto de equilíbrio. É interpretada como a intensidade ou volume do som. Em termos espaciais, o deslocamento das partículas da onda sonora é muito pequeno, da ordem de frações de mílimetros. Para quantizar a intensidade do som, utilizamos uma medida chamada decibel (dB), que é a dada pelo logaritmo da pressão exercida pela vibração no ar.

2. Freqüência fundamental: é o número de vezes que a partícula completa seu movimento vibratório e volta ao seu estado inicial em uma

determinada unidade de tempo. A unidade de freqüência mais utilizada é Hertz (Hz), ou número de ciclos por segundo. A freqüência é interpretada como a altura do som. O termo altura é frequentemente confundido com volume. A diferença de volume refere-se a quanto um som é mais forte ou fraco que outro, enquanto a diferença de altura refere-se a quanto um som é mais agudo ou grave que outro.

3. Espectro de freqüências: raramente um som é composto de uma única freqüência, geralmente ele é uma combinação de vibrações em várias freqüências diferentes simultaneamente. O espectro de freqüências determina quais as freqüências que compõem o som e quais suas intensidades. Interpretamos essa característica como o timbre do som, e isso é o que diferencia as fontes sonoras. Dessa forma, se uma mesma nota musical é tocada em um violino ou em uma flauta, podemos distinguir o intrumento em que a nota é tocada por seu timbre, ou seja, pela intensidade das diferentes freqüências que compõem o som gerado pelo instrumento. No caso de uma nota musical isolada, o timbre é dado pela intensidade dos harmônicos, i. e., dos múltiplos da freqüência fundamental.

4. Duração: refere-se ao tempo entre o início e o fim da percepção do som.

A Figura 1.1 mostra alguns exemplos de sinais correspondentes a instrumentos musicais diversos. Os três primeiros possuem mesma freqüência fundamental, ou seja, correspondem à mesma nota, mas diferenciam-se por seu timbre, que pode ser visualizado pelo formato das ondas. Por possuírem freqüência fundamental identificável, esses sons são ditos **harmônicos**. Já o quarto sinal corresponde ao som de uma caixa de bateria que, por ser um instrumento **percussivo**, não apresenta freqüência fundamental definida e, portanto, é dito um sinal **não harmônico**.

1.2 Modelos Matemáticos

Vamos agora analisar o comportamento de um ponto situado na membrana basilar e, portanto, estudar o som como o percebemos. Para isso, considere uma partícula de massa m sujeita a uma força F direcionada a uma posição de equilíbrio $y_0 = 0$, com magnitude proporcional à distância y da posição de equilíbrio, sendo k a constante dessa proporção.

$$F = -ky \tag{1.1}$$

Pela lei de Newton, temos:

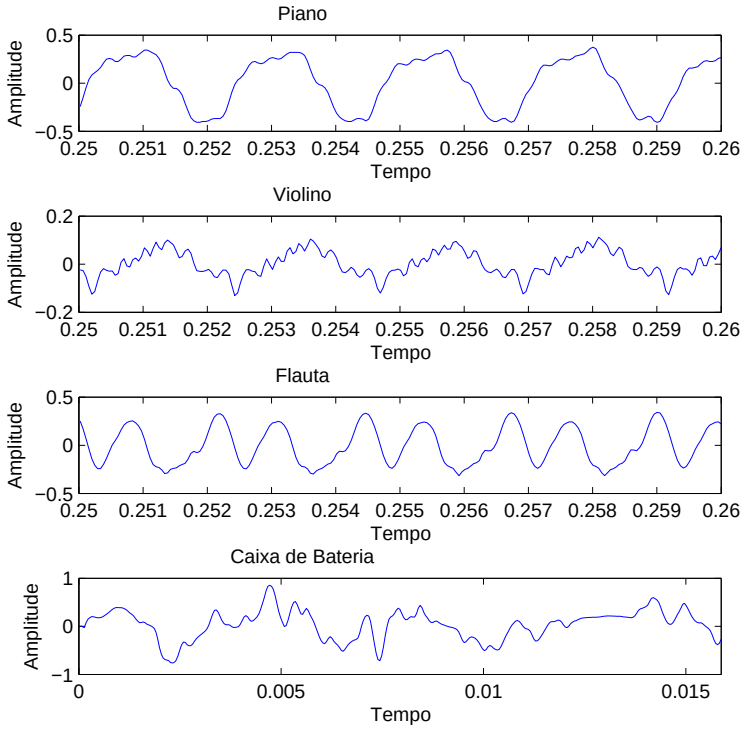

Figura 1.1: Exemplos de sinais gerados por instrumentos musicais.

$$F = ma \tag{1.2}$$

Sendo a a aceleração da partícula dada por:

$$a = \frac{d^2y}{dt^2} \tag{1.3}$$

Substituindo (1.1) e (1.3) em (1.2), temos:

$$\frac{d^2y}{dt^2} + \frac{ky}{m} = 0$$

As soluções dessa equação diferencial são dadas por:

$$y = A\cos(\sqrt{\frac{k}{m}}t) + B\,\mathrm{sen}\,(\sqrt{\frac{k}{m}}t)$$

Por outro lado, $\mathrm{sen}\,(t) = \cos(t - \frac{\pi}{2})$ e $\cos(t) = \sin(t + \frac{\pi}{2})$, ou seja, as duas funções descrevem o mesmo movimento no tempo, diferindo apenas na posição de início da trajetória, que chamaremos de fase inicial. Assim, o movimento de partículas vibratórias pode ser descrito através da soma de senóides, que podem ser representadas por funções seno ou funções cosseno. Vamos escolher o cosseno para representar essas parcelas, que são chamadas, também, de parciais.

Vamos agora analisar a função $A\cos(bt + \theta)$. O termo θ, como vimos, será nossa fase inicial. Sabemos que os valores máximo e mínimo da função cosseno são 1 e -1, respectivamente. O termo A modifica esses valores máximo e mínimo, correspondendo à amplitude. Sabemos também que o cosseno é uma função periódica de período 2π, ou seja, $\cos(t + 2\pi) = \cos(t)$. Já o termo b modifica a função de forma que seu novo período seja $\frac{2\pi}{b}$. Como a freqüência é dada pelo inverso do período, se quisermos uma função cosseno de freqüência f, devemos ter $b = 2\pi f$.

Desse modo, utilizaremos modelos de representação de um sinal sonoro que se baseiam na soma de funções senoidais, com diferentes freqüências e amplitudes. Esses modelos visam aproximar o comportamento do som em nosso ouvido e como ele é enviado ao cérebro. Na membrana basilar, as diferentes freqüências presentes em um determinado som são separadas e enviadas ao cérebro [5]. Veremos posteriormente um procedimento que nos permitirá separar, de forma semelhante, diferentes freqüências presentes em um som, junto com suas respectivas intensidades.

1.2.1 Modelo Senoidal

O primeiro modelo que apresentaremos é o mais simples. Esse modelo baseia-se em somas de funções cosseno, de amplitudes e freqüências constantes.

$$x(t) = \sum_{k=1}^{M} A_k \cos(2\pi f_k t + \theta_k) \tag{1.4}$$

1.2.2 Modelo Senoidal com Amortecimento Exponencial

Sons emitidos por instrumentos musicais possuem certos aspectos em comum. Por exemplo, o som de uma nota musical emitido por um instrumento caracteriza-se por um ataque de intensidade grande e, na seqüência, uma diminuição gradativa de volume. Ou seja, o som da nota musical atinge seu volume máximo no momento em que é tocada e, após isso, diminui progressivamente sua intensidade. Para modelar esse decaimento, o modelo senoidal com amortecimento exponencial insere um parâmetro $e^{d_k t}$, com $d_k < 0$:

$$x(t) = \sum_{k=1}^{M} e^{d_k t} A_k \cos(2\pi f_k t + \theta_k) \tag{1.5}$$

Utilizaremos esse modelo nos métodos de análise e síntese de sinais musicais no domínio do tempo.

1.2.3 Modelo Senoidal Variável no Tempo

Neste modelo, cada componente senoidal tem seus parâmetros de amplitude e freqüência variáveis no tempo. Como a freqüência não é mais constante, substituimos o termo $f_k + \theta_k$, pelo termo $\Theta_k(t)$, que diz respeito à fase instantânea da senóide, estando relacionado tanto à fase inicial quanto à freqüência instantânea. Em geral, representamos essa fase por $\Theta_k(t) = \theta_{k_0} + 2\pi \int_0^t f_k(s)ds$. Observemos, além disso, que o número de senóides presentes em cada momento, $M(t)$, também pode ser variável. Nesse caso, o sinal é representado por:

$$x(t) = \sum_{k=1}^{M(t)} A_k(t) \cos(\Theta_k(t)) \tag{1.6}$$

Tomando $A_k(t) = e^{d_k t} A_{k_0}$, vemos que o modelo anterior é um caso particular deste. Utilizaremos esse modelo para o algoritmo PARSHL.

1.2.4 Modelo Senoidal com Resíduo

Este modelo, também conhecido como Modelo Determinístico mais Estocástico, considera que há componentes do som que correspondem a ruídos, a sons percussivos, ou a sons característicos do ataque de notas musicais, como o ruído de um dedo batendo em uma corda de violão, ou o martelo batendo em uma corda de piano. Para essas componentes a modelagem senoidal não funciona bem. O sinal é então expresso por:

$$x(t) = \sum_{k=1}^{M(t)} [A_k(t) \cos(\Theta_k(t))] + e(t) \tag{1.7}$$

Esse é o modelo utilizado no algoritmo SMS.

1.2.5 Modelagem Física

Uma outra abordagem da análise e síntese musical é a modelagem física. Ao contrário dos modelos que vimos, a modelagem física é voltada a um instrumento específico, utilizando, para isso, equações diferenciais que descrevem seu comportamento.

A grande vantagem da modelagem física é a aproximação fiel ao instrumento que emite o som, tendo mais qualidade com menor processamento. Sua desvantagem é a falta de generalidade, pois é aplicável apenas ao instrumento modelado, enquanto os outros modelos podem ser aplicados a diversas categorias diferentes de som.

Podemos dizer que os modelos físicos representam a fonte emissora do som, enquanto os modelos baseados na decomposição em parciais (tanto no domínio do tempo quanto no das freqüências) descrevem a fonte receptora do som.

1.3 Histórico do Processamento de Sinais de Áudio

O processamento de sinais é uma área muito ampla com diversas aplicações. Vamos nos restringir ao processamento de sinais de áudio, cuja importância se destaca em áreas como música, telefonia, fonoaudiologia, entre outras.

A música é uma arte milenar, porém apenas no início do século XX foram encontradas formas de armazená-la e reproduzi-la. Ainda assim, esses meios eram pouco flexíveis e não permitiam muitas alterações ao som depois de gravado.

Com o advento dos computadores, surgiu a possibilidade de armazenamento de músicas em forma de sinais discretos e sua modificação. Surgiram também os sintetizadores, despertando o interesse de compositores em explorar formas musicais impossíveis de serem reproduzidas pelos instrumentos tradicionais. Surgiam então a música eletrônica e a eletro-acústica.

Desde o século XVIII, era conhecida uma forma de análise de ondas sonoras através de sua decomposição em senóides de freqüências diferentes. Conhecia-se também uma versão dessa análise aplicável a pontos discretos, assim como os que compõem um sinal digital, que se chama DFT (Discrete Fourier Transform). Porém, essa técnica demandava um grande número de operações computacionais, impossibilitando sua utilização em computadores rudimentares, como os da metade do século XX.

Em 1965, Cooley e Tukey[8] desenvolveram um algoritmo que permitiu o cálculo da DFT com muito menos operações. Essa ferramenta ficou conhecida como FFT (Fast Fourier Transform) e foi um marco na história do processamento de sinais e da análise musical. Com isso, foi possível a decomposição do espectro de freqüências que compõem um sinal sonoro de forma rápida e automática. Outro algoritmo, baseado na FFT, se tornou importante na análise de sinais. Trata-se da STFT (Short-Time Fourier Transform), que consiste na divisão do sinal em quadros menores e na aplicação da FFT em cada um desses quadros, gerando uma seqüência de espectros de freqüências que variam no tempo.

As primeiras aplicações da STFT no processamento de audio digital datam do início da década de 1970, quando foi utilizada para o cálculo da amplitude e freqüência instantâneas individuais de uma componente senoidal de um sinal sonoro. Baseado nisso, surgiu em 1976 o Phase Vocoder [17], o primeiro algoritmo completo de análise e síntese de sinais sonoros. Porém, essa ferramenta era pouco flexível, não gerando bons resultados para sons não-harmônicos ou com componentes cujas freqüências apresentam grandes variações ao decorrer do tempo.

Na década de 1980, dois trabalhos independentes geraram algoritmos semelhantes para a análise e síntese de sinais sonoros. McAulay e Quatieri [15] desenvolveram um algoritmo para análise e síntese de sinais correspondentes à fala. Na mesma época, Smith e Serra [19] desenvolveram o PARSHL, um algoritmo semelhante ao de McAuley e Quatieri, aplicável a sons mais gerais, incluindo sinais não-harmônicos. A principal evolução desses algoritmos em relação ao Phase Vocoder foi a forma de detectar as freqüências das componentes senoidais, através de interpolações, e acompanhar suas modificações ao longo do tempo.

Na década de 1990, os mesmos criadores do PARSHL utilizaram o modelo determinístico mais estocástico para melhorar a análise e síntese de sinais

com muito ruído. Surgiu então, como uma evolução natural do PARSHL, o algoritmo SMS (Spectral Modelling Synthesis) [20]. Sua grande vantagem foi a melhor representação de ruídos e sinais percussivos, que não se encaixam no modelo senoidal.

Paralelamente aos métodos citados acima, surgiu na década de 1980 o ESPRIT [18], um outro método para a estimativa de parâmetros de sinais. Esse algoritmo deu origem a uma nova família de métodos de análise e síntese musical, conhecidos como métodos de alta resolução. Nesses algoritmos, prioriza-se o domínio do tempo em detrimento do domínio das freqüências, ou seja, raramente visualiza-se o espectro do sinal, mesmo assim é possível encontrar suas componentes.

Os métodos de alta resolução que apresentaremos são baseados no modelo senoidal com amortecimento exponencial [2] [9]. Neles, utilizam-se os pontos discretos do sinal para a formação de uma matriz de Hankel, que pode ser decomposta como $H = WAW^T$, em que W é uma matriz de Vandermonde que contém as informações de freqüência e A é uma matriz diagonal contendo as informações de amplitude das componentes do sinal. Como H é simétrica, as colunas de W são geradas pelos autovetores de H. Através da decomposição espectral de H, encontra-se uma matriz, chamada de matriz espectral, cujos autovalores são os pólos da matriz de Vandermonde W.

Por outro lado, para sons mais longos, deve-se proceder de forma semelhante à STFT nos modelos espectrais, dividindo o som em quadros e analisando suas características locais. Assim, a matriz H passará a depender do tempo e será denominada $H(t)$, sendo chamada de matriz deslizante, pois a cada avanço de tempo, uma linha/coluna é removida e outra é acrescentada.

Para suprir a necessidade de análise local dos sinais, surgiram os métodos de alta resolução adaptativos. No entanto, a necessidade de se realizar uma decomposição espectral (ou QR), a cada passo, torna o método ainda mais dispendioso computacionalmente. Logo, procuraram-se métodos iterativos para obter uma matriz ortogonal que represente uma base para o subespaço dominante de H e, posteriormente, encontrar a matriz espectral. Além disso, esses métodos deveriam fornecer atualizações simples de todas essas matrizes.

Recentemente, foi proposto em [4] uma forma de atualização da matriz espectral, supondo que a matriz deslizante fosse atualizada segundo a expressão: $H(t+1) = H(t) + E(t)G(t)^T$. Alguns algoritmos [1] [13] foram construídos de forma que a atualização da matriz deslizante se comportasse dessa maneira. Em geral, esses algoritmos baseiam-se no algoritmo PAST [25], que busca diminuir a complexidade computacional de métodos tradicionais, como o método de iteração ortogonal [12]. Uma tentativa pioneira da redução dessa complexidade foi o método QR adaptativo criado por Strobach [24].

1.4 Digitalização

Vimos exemplos de modelos matemáticos utilizados para a modelagem de ondas sonoras. Tais modelos consideram que o som pode ser representado por uma função contínua que descreve a pressão aérea gerada pela fonte sonora. Porém, quando se trata da análise e síntese computacional de sinais sonoros, devemos levar em conta que computadores são dispositivos discretos, os quais não representam funções contínuas, e sim, amostras em pontos isolados. Sinais sonoros contínuos, como os presentes na natureza, em discos de vinil e em fitas cassete são chamados de analógicos, enquanto sinais armazenados computacionalmente ou presentes em CDs são conhecidos como digitais.

A conversão do sinal digital para analógico (por exemplo, reproduzir um som armazenado no computador) é feita através de interpolação. Já a conversão de um sinal analógico para digital (por exemplo, gravar um som no computador através de um microfone) é feita através de um processo um pouco mais complexo que requer no mínimo quatro etapas: filtragem, amostragem, quantização e codificação. A etapa de filtragem, apesar de ser a primeira, deve ser explicada posteriormente à etapa de amostragem, para que se entenda sua necessidade. Já as etapas de quantização e codificação dizem respeito à forma como os dados coletados são armazenados pelo computador, sendo irrelevantes no presente momento.

A etapa de amostragem é, sem dúvida, a mais importante no processo de conversão de um sinal analógico para digital. Essa etapa consiste em se tomar o valor da função contínua em pontos equidistantes. Porém, ao fazer-se isso, levantam-se as principais questões referentes ao processo de digitalização, sendo a principal delas a perda de informação ao se ignorar os valores da função entre dois pontos subseqüentes.

Um resultado importante em relação a essa pergunta é o Teorema de Nyquist, o qual fundamenta a teoria da amostragem.

Teorema 1.1 (Nyquist)**.** *Seja $g(t)$ uma função que não contenha freqüências maiores que W. Então, $g(t)$ pode ser completamente determinada a partir do conjunto $\{g(t_k)\}$, em que $\{t_k\}$ é um conjunto de pontos igualmente espaçados no domínio de g, de forma que $t_{k+1} - t_k = \Delta t < \frac{1}{2W}$.*

Segundo este teorema, a maior freqüência representável em uma digitalização, ou seja, a maior freqüência que pode estar presente no sinal para que sua representação digital não possua perda de informações, deve ser menor que a metade da freqüência de amostragem.

Na Figura 1.2, temos uma freqüência de amostragem suficiente, maior que o dobro da freqüência da senóide representada. Assim, podemos reconstituir

exatamente a senóide original com os dados discretos que possuímos.

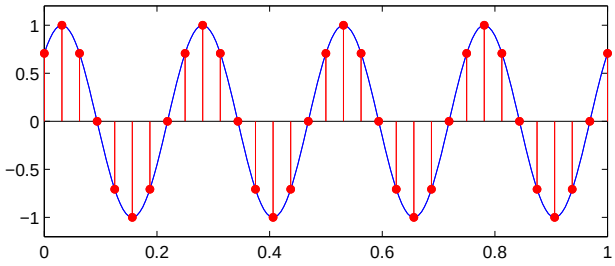

Figura 1.2: $2f_{senóide} < f_{amostragem}$

Já na Figura 1.3, temos uma situação mais delicada, quando a freqüência de amostragem é exatamente o dobro da freqüência da senóide e, portanto, podem ocorrer três casos diferentes: no primeiro deles, a amostragem acontece exatamente nos picos da senóide, sendo então possível sua reconstrução completa; no segundo caso, a amostragem ocorre sempre no ponto onde a função é nula, sendo que nenhuma senóide é detectada; no terceiro caso, pode ser recuperada a freqüência da senóide, porém sua amplitude e fase sofrem deformações.

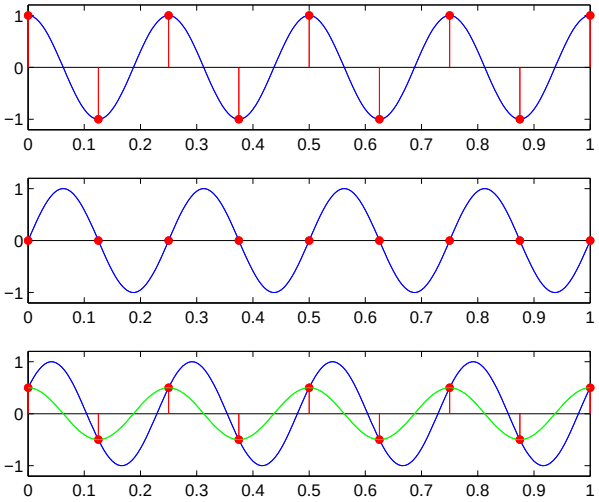

Figura 1.3: $2f_{senóide} = f_{amostragem}$

A Figura 1.4 mostra o que ocorre quando a freqüência de amostragem é insuficiente, ou seja, inferior ao dobro da freqüência da senóide. Neste caso,

os valores dos pontos de amostragem são ambíguos e podem ser interpretados como uma freqüência não presente no sinal, fenômeno conhecido na literatura por mascaramento (aliasing).

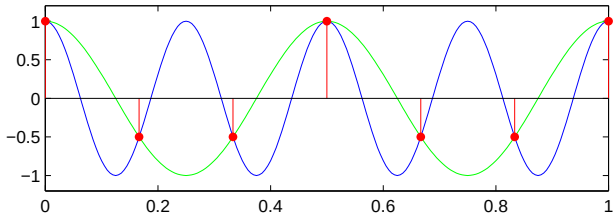

Figura 1.4: $2f_{senóide} > f_{amostragem}$

Sabendo-se qual será a freqüência de amostragem, deve-se aplicar um filtro ao sinal de entrada para eliminar as freqüências superiores à metade da taxa de amostragem, para que elas não gerem senóides não presentes no sinal original, antes da etapa de amostragem.

Podemos escolher a taxa de amostragem ideal baseada na aplicação com a qual estamos trabalhando. Por exemplo, sabemos que o ouvido humano apenas reconhece freqüências abaixo de 20000Hz. Portanto, em aplicações de áudio, freqüências de amostragem acima de 40000Hz são suficientes para uma ótima qualidade sonora, representando todo o espectro audível. Já uma taxa de amostragem muito superior acarretaria no armazenamento de informações irrelevantes e aumentaria o gasto de memória demandada.

A freqüência de amostragem mais utilizada para aplicações de áudio digital de alta qualidade é de 44100Hz, sendo também a taxa de amostragem em CDs de música. Por outro lado, a freqüência de amostragem de um telefone convencional é de 8000Hz, já que este visa apenas a conversação inteligível entre duas pessoas e, em uma conversa regular, a freqüência fundamental de uma voz masculina adulta é em média de 120Hz e de uma voz feminina adulta é de 250Hz, raramente ultrapassando 500Hz.

Capítulo 2

Algoritmos no Domínio do Tempo

Os métodos para análise e ressíntese de sinais musicais no domínio do tempo são métodos que utilizam apenas informações temporais do sinal para procurar os parâmetros que o descrevem. Esses métodos pressupõem que os sinais musicais são quase harmônicos, isto é, são bem descritos pelo modelo senoidal com amortecimento exponencial. Nesse caso, a matriz de Hankel H, definida por $H_{ij} = x(i+j-2)$, pode ser decomposta na forma WAW^T, em que W é uma matriz de Vandermonde e A é uma matriz diagonal, conforme veremos adiante. A partir de A e W, calculamos os parâmetros desejados para ressintetizar o sinal. A seguir, vamos revisar alguns conceitos de álgebra linear e provar alguns lemas que utilizaremos no desenvolvimento desses métodos.

2.1 Preliminares de Álgebra Linear

Vamos agora revisar alguns conceitos da Teoria de Matrizes. Primeiro, vamos apresentar algumas notações:

$tr(A)$ é o traço de A.

Observação: $tr(A^T) = tr(A)$ e $tr(AB) = tr(BA)$.

$\|A\|_2 \stackrel{def}{=} \max_{\|x\|_2=1} (\|Ax\|_2)$

$Im(A)$ é o espaço coluna de A.

$\lambda(A)$ é o espectro de A.

$\rho(A)$, o raio espectral de A, é o autovalor de A cujo valor absoluto é máximo.

Observação: $\|A\|_2 = \sqrt{\rho(A^T A)}$.

$A(k,:)$ denota a k-ésima linha de A e $A(:,k)$ denota a k-ésima coluna de

A. Sejam $u = (i_1, ..., i_r)$, em que $1 \leq i_1 < ... < i_r \leq m$ e $v = (j_1, ..., j_s)$, em que $1 \leq j_1 < ... < j_s \leq n$. Então $A(u, v)$ é a submatriz de A definida a partir de u e v. Definimos $1 : m = (1, 2, 3, ..., m)$.

Lema 2.1 (Sherman-Morrison)**.** *Seja A uma matriz inversível e sejam U e V tais que $(I - V^T A^{-1} U)$ seja também inversível. Então,*

$$(A - UV^T)^{-1} = A^{-1} + A^{-1}U(I - V^T A^{-1}U)^{-1}V^T A^{-1}$$

Demonstração.

$$[A^{-1} + A^{-1}U(I - V^T A^{-1}U)^{-1}V^T A^{-1}][A - UV^T] = I - A^{-1}UV^T +$$

$$+A^{-1}U(I - V^T A^{-1}U)^{-1}V^T - A^{-1}U(I - V^T A^{-1}U)^{-1}V^T A^{-1}UV^T =$$

$$= I - A^{-1}UV^T + A^{-1}U(I - V^T A^{-1}U)^{-1}(I - V^T A^{-1}U)V^T =$$

$$= I - A^{-1}UV^T + A^{-1}UV^T = I$$

□

Definição 1. *Seja $A \in \mathbb{R}^{m \times n}$. Dizemos que A é uma matriz de posto completo se posto(A) $= n$.*

Definição 2. *Sejam $A \in \mathbb{R}^{n \times n}$ e $\{v_1, ..., v_m\} \subset \mathbb{R}^n$. Se $[Av_1, ..., Av_m] \subset [v_1, ..., v_m]$, então $[v_1, ..., v_m]$ é um subespaço invariante sob A.*

Proposição 2.1. *Sejam $A \in \mathbb{R}^{n \times n}$ e $X \in \mathbb{R}^{n \times k}$ de posto completo. Se $\exists B \in \mathbb{R}^{k \times k}$ tal que $AX = XB$, então a imagem de X é um subespaço invariante sob A.*

Lema 2.2. *Sejam $A \in \mathbb{R}^{n \times n}$ e $X \in \mathbb{R}^{n \times k}$ de posto completo e $B \in \mathbb{R}^{k \times k}$ tais que $AX = XB$. Então o conjunto de autovalores de B está contido no conjunto de autovalores de A.*

Demonstração. Seja λ um autovalor de B e v um autovetor não nulo associado a λ. $Bv = \lambda v \Rightarrow AXv = XBv = \lambda Xv$. Como X é de posto completo e $v \neq 0$ então $Xv \neq 0$. Logo, λ é autovalor de A associado ao autovetor Xv. □

Definição 3. *Seja $Q \in \mathbb{R}^{n \times n}$. Diz-se que Q é ortogonal se $Q^T Q = QQ^T = I$.*

Observação: se Q é uma matriz ortogonal, $\|Qx\|_2 = \|x\|$, pois $\|Qx\|_2^2 = x^T Q^T Qx = x^T x = \|x\|_2^2$; $\|QA\|_2 = \|A\|_2$, pois dado x tal que $\|x\|_2 = 1$, $\|QAx\|_2^2 = x^T A^T Q^T QAx = x^T A^T Ax = \|Ax\|_2^2$. O mesmo vale para $\|AQ\|_2$.

Definição 4. *Seja S um subespaço de $\mathbb{R}^n$. $P \in \mathbb{R}^{n\times n}$ é a matriz de projeção ortogonal de $\mathbb{R}^n$ sobre S se $Im(P) = S$, $P^2 = P$ e $P^T = P$.*

Observação: se V é uma matriz cujas colunas formam uma base ortonormal para S, então $VV^T = P$ é a matriz de projeção ortogonal em S.

Lema 2.3. *Se P_1 e P_2 são matrizes de projeção ortogonal, então $\|P_1 - P_2\|_2 \leq 1$*

Demonstração. Dado um espaço S de dimensão n, sejam S_1 e S_2 dois subespaços de dimensão k e k', respectivamente. Seja $W_1 \in \mathbb{R}^{n\times k}$ uma matriz de k colunas ortonormais que gera o subespaço S_1. Assim, $W_1^T W_1 = I$ e $W_1 W_1^T = P_1$, matriz de projeção ortogonal de S sobre S_1. Analogamente, tome $Z_1 \in \mathbb{R}^{n\times k'}$ que gera S_2, com $Z_1^T Z_1 = I$ e $Z_1 Z_1^T = P_2$, matriz de projeção de S sobre S_2. Completaremos agora W_1 e Z_1 de forma a obtermos duas bases ortonormais W e Z para o espaço S: $W = (W_1|W_2)$ e $Z = (Z_1|Z_2)$, com $W_2 \in \mathbb{R}^{n\times(n-k)}$ e $Z_2 \in \mathbb{R}^{n\times(n-k')}$, que podem ser escritos da seguinte forma: $W_2 = W_1^{\perp}$ e $Z_2 = Z_1^{\perp}$. Portanto, $W_1^T W_2 \equiv 0$ e $Z_1^T Z_2 \equiv 0$.

Assim, $\|P_1 - P_2\|_2 = \|W_1 W_1^T - Z_1 Z_1^T\|_2 = \|W^T(W_1 W_1^T - Z_1 Z_1^T)Z\|_2$, pois a norma euclidiana é invariante por transformações ortogonais. Note agora que $W^T W_1 = \begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix} W_1 = \begin{pmatrix} I \\ 0 \end{pmatrix}$ e $Z_1^T Z = Z_1^T(Z_1|Z_2) = (I|0)$. Então

$$\begin{aligned}
\|P_1 - P_2\|_2 &= \left\|W^T(W_1 W_1^T - Z_1 Z_1^T)Z^T\right\|_2 = \\
&= \left\|W^T W_1 W_1^T Z - W^T Z_1 Z_1^T Z\right\|_2 = \\
&= \left\|\begin{pmatrix} I \\ 0 \end{pmatrix} W_1^T Z - W^T Z_1 \,(I|0)\right\|_2 = \\
&= \left\|\begin{pmatrix} W_1^T Z_1 & W_1^T Z_2 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} W_1^T Z_1 & 0 \\ W_2^T Z_1 & 0 \end{pmatrix}\right\|_2 = \\
&= \left\|\begin{pmatrix} 0 & W_1^T Z_2 \\ -W_2^T Z_1 & 0 \end{pmatrix}\right\|_2
\end{aligned}$$

Para encontrarmos a norma desta última matriz, considere a matriz ortogonal $W^T Z$:

$$W^T Z = \begin{pmatrix} W_1^T Z_1 & W_1^T Z_2 \\ W_2^T Z_1 & W_2^T Z_2 \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} = Q$$

Como W^T e Z são matrizes ortogonais, então $W^T Z$ é ortogonal e, portanto, $\|W^T Z\|_2 = 1$. Como $Q_{12} = W_1^T Z_2$ e $Q_{21} = W_2^T Z_1$ são submatrizes de $Q = W^T Z$, temos $\|Q_{12}\|_2 \leq \|Q\|_2$ e $\| - Q_{21}\|_2 = \|Q_{21}\|_2 \leq \|Q\|_2$. Note que $Q_{12} = W_1^T Z_2 \in \mathbb{R}^{k\times(n-k')}$ e $Q_{21} = W_2^T Z_1 \in \mathbb{R}^{(n-k)\times k'}$. Tome então $x \in \mathbb{R}^{n\times 1}$

tal que $\|x\|_2 = 1$ e $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, com $x_1 \in \mathbb{R}^{k' \times 1}$ e $x_2 \in \mathbb{R}^{(n-k') \times 1}$. Então $1 = \|x\|_2 = \|x_1\|_2 + \|x_2\|_2$. Logo:

$$\begin{aligned}
\|(P_1 - P_2)x\|_2^2 &= \left\| \begin{pmatrix} 0 \\ -Q_{21}x_1 \end{pmatrix} + \begin{pmatrix} Q_{12}x_2 \\ 0 \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} Q_{12}x_2 \\ -Q_{21}x_1 \end{pmatrix} \right\|_2^2 = \\
&= \|Q_{12}x_2\|_2^2 + \|Q_{21}x_1\|_2^2 \leq \\
&\leq \|Q_{12}\|_2^2 \|x_1\|_2^2 + \|Q_{21}\|_2^2 \|x_2\|_2^2 \leq \\
&\leq \|Q\|_2^2 \|x_1\|_2^2 + \|Q\|_2^2 \|x_2\|_2^2 = \\
&= \|x_1\|_2^2 + \|x_2\|_2^2 = 1 \\
&\Rightarrow \|P_1 - P_2\|_2^2 \leq 1
\end{aligned}$$

□

Definição 5. *Uma matriz $A \in \mathbb{R}^{n \times n}$ é dita ser diagonalizável se A possui n autovetores linearmente independentes.*

Se A é uma matriz diagonalizável então A pode ser escrita como $A = S\Lambda S^{-1}$, em que S é uma matriz formada pelos autovetores linearmente independentes de A e Λ é uma matriz diagonal formada pelos autovalores de A.

Esta decomposição é chamada forma diagonal e não é única, pois um múltiplo de um autovetor continua sendo autovetor associado ao mesmo autovalor.

Se A é uma matriz real e simétrica, então ela pode ser decomposta da forma $A = Q\Lambda Q^T$, com Q ortogonal e Λ diagonal real.

Lema 2.4. *Seja $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{n \times m}$. Se AB é uma matriz diagonalizável e $0 \notin \lambda(BA)$, então BA também é diagonalizável.*

Demonstração. Como AB é diagonalizável, então $AB = PDP^{-1}$, com $D = diag(\lambda_1, ..., \lambda_p, 0, ..., 0)$. Se $0 \notin \lambda(BA)$, então o posto de A é completo.

Se A é quadrada, então $BA = A^{-1}ABA = A^{-1}PDP^{-1}A$, logo BA é diagonalizável.

Seja A uma matriz $m \times n$. Logo, $m > n$ (pois A é de posto completo). Seja C uma matriz $m \times (m-n)$, de forma que a matriz $F = (A|C)$ seja quadrada e inversível. Considere também a matriz $G = \begin{pmatrix} -B \\ 0 \end{pmatrix}$, quadrada. Então, $FG = (A|C) \begin{pmatrix} -B \\ 0 \end{pmatrix} = AB = PDP^{-1}$. Por outro lado, $\begin{pmatrix} BA & BC \\ 0 & 0 \end{pmatrix} =$

$GF = F^{-1}(FG)F = F^{-1}PDP^{-1}F$. Assim,

$$\begin{pmatrix} BA & BC \\ 0 & 0 \end{pmatrix}(F^{-1}P) = (F^{-1}P)D.$$

Seja agora $\lambda \neq 0$ um autovalor de GF. Então $\begin{pmatrix} BA & BC \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \lambda\begin{pmatrix} x \\ y \end{pmatrix}$. Daí, $y = 0$ e $BA = \lambda x$. Ou seja, $(F^{-1}P)e_k$ é da forma $\begin{pmatrix} x_k \\ 0 \end{pmatrix}$, $k = 1, ..., p$. Logo, como as colunas de $F^{-1}P$ são LI, $\{x_1, ..., x_p\}$ é LI, ou seja, BA é diagonalizável. □

(Decomposição SVD)**.** *Uma matriz $A \in \mathbb{R}^{m\times n}$ pode ser decomposta como $A = U\Sigma V^T$, em que $U \in \mathbb{R}^{m\times m}$ é ortogonal, $V \in \mathbb{R}^{n\times n}$ é ortogonal e $\Sigma \in \mathbb{R}^{m\times n}$ é diagonal com elementos $\sigma_1 \geq ... \geq \sigma_r$, em que r é o posto de A.*

Observação: se $m \geq n$, A pode também ser decomposta por uma **SVD econômica** da forma $A = U_1\Sigma_1 V^T$, em que $U_1 = U(:, 1:n) \in \mathbb{R}^{m\times n}$, as n primeiras colunas de U e $\Sigma_1 = \Sigma(1:n, 1:n) \in \mathbb{R}^{n\times n}$.

(Decomposição QR)**.** *Dada uma matriz $A \in \mathbb{R}^{m\times n}$, ela pode ser escrita como $A = QR$, em que $R \in \mathbb{R}^{m\times n}$ é triangular superior e $Q \in \mathbb{R}^{m\times m}$ é ortogonal. Se A possui posto coluna completo, então Q fornece uma base ortonormal para $Im(A)$.*

Observação: se $m \geq n$, A pode também ser decomposta por uma **QR econômica** da forma $A = Q_1R_1$, em que $Q_1 = Q(:, 1:n) \in \mathbb{R}^{m\times n}$, as n primeiras colunas de Q e $R_1 = R(1:n, 1:n) \in \mathbb{R}^{n\times n}$.

Teorema 2.1. *O conjunto das matrizes $A \in \mathbb{R}^{n\times m}$, $m \leq n$, tais que posto(A) $= m$ é denso em $\mathbb{R}^{n\times m}$.*

Demonstração. Seja $A \in \mathbb{R}^{m\times n}$, tal que *posto*$(A) = r < m$. Seja $A = U\Sigma V^T$ a decomposição SVD de A. Temos que:

$$\Sigma = \left(\begin{array}{ccccc} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & 0 & \\ & & & & \ddots \\ \hline & & & & 0 \\ & & 0 & & \end{array}\right)$$

Defina:

$$\Sigma_k = \begin{pmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & \frac{1}{k} & & \\ & & & & \ddots & \\ & & & & & \frac{1}{k} \\ \hline & & & 0 & & \end{pmatrix}$$

Considere agora as matrizes $A_k = U\Sigma_k V^T$. Então,

$$A = \lim_{k\to\infty} A_k$$

□

Definição 6. *Seja $A \in \mathbb{R}^{m\times n}$ e seja $A = U\Sigma V^T$ sua decomposição SVD. A pseudo-inversa de A é definida por $A^\dagger = V\Sigma^\dagger U^T$, em que os elementos da diagonal de $\Sigma^\dagger$ são os inversos dos elementos não nulos da diagonal de Σ.*

Observação: a pseudo-inversa é a única inversa generalizada que satisfaz as 4 condições de Moore-Penrose.

(i) $AA^\dagger A = A$

(ii) $A^\dagger AA^\dagger = A^\dagger$

(iii) $(AA^\dagger)^T = AA^\dagger$

(iv) $(A^\dagger A)^T = A^\dagger A$

A pseudo-inversa possui as seguintes propriedades [23]:

1. $(A^\dagger)^\dagger = A$.
2. $(\bar{A})^\dagger = \overline{(A^\dagger)}$.
3. $(A^T)^\dagger = (A^\dagger)^T$.
4. $posto(A) = posto(A^\dagger) = posto(AA^\dagger) = posto(A^\dagger A)$.
5. $(AA^H)^\dagger = A^{H\dagger}A^\dagger$, $(A^HA)^\dagger = A^\dagger A^{H\dagger}$.
6. $(AA^H)^\dagger AA^H = AA^\dagger$, $(A^HA)^\dagger A^HA = A^\dagger A$.

7. Se $A \in \mathbb{C}^{m\times n}$ tem posto n, então $A^\dagger = (A^H A)^{-1}A^H$ e $A^\dagger A = I^{(n)}$.

8. Se $A \in \mathbb{C}^{m\times n}$ tem posto m, então $A^\dagger = A^H(AA^H)^{-1}$ e $AA^\dagger = I^{(m)}$.

9. Se A tem fatoração de posto completo $A = FG^H$, em que $posto(F) = posto(G) = posto(A)$, então $A^\dagger = G(F^H AG)^{-1}F^H$ e $A^\dagger = (G^\dagger)^H F^\dagger$.

Definição 7. *Uma matriz $A \in \mathbb{R}^{n\times n}$ é simétrica definida positiva (SDP) se $A = A^T$ e $(\forall x \in \mathbb{R}^n)\ x \neq 0$, vale que $x^T Ax > 0$.*

Observação: uma matriz é SDP se e somente se todos os seus autovalores são positivos. Se λ e v são autovalores de A, então $v^T Av = \lambda v^T v = \lambda\|v\|_2^2$ e, portanto, $v^T Av$ tem o mesmo sinal de λ.

Lema 2.5. *Se C é SDP e Y é de posto completo, então $Y^T CY$ é SDP.*

Demonstração. Seja x um vetor não nulo. Defina $y = Yx$. Se Y tem posto completo, então y é não nulo, logo $x^T Y^T CYx = y^T Cy > 0$, pois C é SDP. Logo, $Y^T CY$ é SDP. □

Lema 2.6. *Se A é SDP, $B = B^T$ e $AB + BA = 0$, então $B = 0$.*

Demonstração. Se B é simétrica, então B possui decomposição espectral da forma: $B = U\Sigma U^T$. $AB + BA = 0 \Leftrightarrow AU\Sigma U^T + U\Sigma U^T A = 0 \Leftrightarrow U^T AU\Sigma + \Sigma U^T AU = 0$. Pelo lema 2.5, $\hat{A} = U^T AU$ é SDP e $\hat{A}\Sigma + \Sigma\hat{A} = 0 \Rightarrow (\forall i)\,(\hat{A}\Sigma + \Sigma\hat{A})_{ii} = 0$. Ou seja, os elementos da diagonal de $\hat{A}\Sigma + \Sigma\hat{A}$ são nulos. Por outro lado, $0 = (\hat{A}\Sigma + \Sigma\hat{A})_{ii} = e_i^T(\hat{A}\Sigma + \Sigma\hat{A})e_i = e_i^T\hat{A}\Sigma e_i + e_i^T\Sigma\hat{A}e_i$. Como Σ é diagonal, os vetores canônicos são seus autovetores e seus autovalores os elementos da diagonal. Então $\Sigma e_i = \lambda_i e_i$ e $e_i^T\Sigma = e_i^T\lambda_i = \lambda_i e_i^T$. Assim, $e_i^T\hat{A}\Sigma e_i + e_i^T\Sigma\hat{A}e_i = \lambda_i(e_i^T\hat{A}e_i) + \lambda_i(e_i^T\hat{A}e_i) = 0 \Rightarrow 2\lambda_i\hat{A}_{ii} = 0$. Como $\hat{A}$ é SDP, $(\forall i)\,\hat{A}_{ii} > 0$, logo $(\forall i)\,\lambda_i = 0$ e, portanto, $\Sigma \equiv 0$. Conseqüentemente, $B \equiv 0$. □

Proposição 2.2. *Seja A uma matriz real simétrica definida não negativa. Então, existe uma única matriz real simétrica definida não negativa P tal que $P^2 = A$.*

Demonstração. Seja $A = QDQ^T$ uma decomposição espectral de A, em que $D \geq 0$. Então, se $P = QD^{\frac{1}{2}}Q^T$, $P^2 = A$. Suponha que $A = R^2$ para alguma matriz real simétrica definida não negativa R. Então, como A comuta com R, A e R são simultaneamente diagonalizáveis, isto é, existe W ortogonal tal que $A = WDW^T$ e $R = WEW^T$. Como $R^2 = A$ e R é uma matriz com autovalores não negativos, $E = D^{\frac{1}{2}}$. Logo, $R = P$. □

Definição 8. *Seja A uma matriz simétrica definida não negativa. Seja P a matriz simétrica definida não negativa tal que $A = P^2$. A matriz P é dita a raiz quadrada de A (Notação: $P = A^{\frac{1}{2}}$).*

Definição 9. *Seja A uma matriz real simétrica definida positiva. Seja P uma matriz real não simétrica tal que $A = PP^T$. A matriz P é dita uma raiz quadrada assimétrica de A.*

Proposição 2.3. *Seja A uma matriz real simétrica definida positiva. Seja P uma raiz quadrada assimétrica de A. Então, existe uma matriz ortogonal U tal que $A^{\frac{1}{2}} = PU = U^T P^T$.*

Demonstração. Seja $U = P^{-1}A^{\frac{1}{2}}$. $U^T U = A^{\frac{1}{2}}P^{-T}P^{-1}A^{\frac{1}{2}} = A^{\frac{1}{2}}A^{-1}A^{\frac{1}{2}} = I$. Como $A^{\frac{1}{2}}$ é SDP, $(PU)^T = PU$, isto é, $PU = U^T P^T$. □

Corolário 2.1. *Seja C uma matriz real SDP. Sejam A e B duas matrizes reais SDP tais que $C = ABA$. Então, $AB^{\frac{1}{2}}$ é uma raiz assimétrica de C.*

Demonstração. $(AB^{\frac{1}{2}})(AB^{\frac{1}{2}})^T = (AB^{\frac{1}{2}})(B^{\frac{1}{2}}A) = ABA = C$. □

Proposição 2.4. *Seja A real SDP. Então, $(A^{\frac{1}{2}})^{-1} = (A^{-1})^{\frac{1}{2}}$.*

Demonstração. Seja $P = A^{\frac{1}{2}}$. Então $(A^{\frac{1}{2}})^{-1} = P^{-1}$. Por outro lado, $(P^{-1})^2 = P^{-1}P^{-1} = (PP)^{-1} = A^{-1}$. □

Definição 10. *Seja A uma matriz real SDP. Definimos $A^{-\frac{1}{2}} = (A^{\frac{1}{2}})^{-1} = (A^{-1})^{\frac{1}{2}}$.*

Corolário 2.2. *Seja C uma matriz real SDP. Sejam A e B duas matrizes reais SDP tais que $C = ABA$. Então, existe U ortogonal tal que $C^{-\frac{1}{2}} = A^{-1}B^{-\frac{1}{2}}U = U^T B^{-\frac{1}{2}}A^{-1}$.*

Demonstração. Pelo corolário 2.1, $AB^{\frac{1}{2}}$ é uma raiz assimétrica de C. Portanto, existe U tal que $C^{\frac{1}{2}} = AB^{\frac{1}{2}}U$. Logo, $C^{-\frac{1}{2}} = (C^{\frac{1}{2}})^{-1} = (AB^{\frac{1}{2}}U)^{-1} = U^T(B^{\frac{1}{2}})^{-1}A^{-1} = U^T B^{-\frac{1}{2}}A^{-1}$. Analogamente, tomando $C^{\frac{1}{2}} = AB^{\frac{1}{2}}$, chegamos a $C^{\frac{1}{2}} = A^{-1}B^{-\frac{1}{2}}U$. □

2.2 Estimativa de Parâmetros

Começaremos agora o estudo dos métodos de alta resolução para análise e síntese de sinais sonoros. Esses métodos baseiam-se no modelo senoidal com amortecimento exponencial ([2], [9]), no qual o sinal sonoro discreto $x(t)$ é descrito como uma soma de senóides com amplitude a_k, freqüência f_k, fase inicial θ_k e um decaimento exponencial $e^{d_k t}$, $t = 0, ..., N-1$:

$$x(t) = \sum_{k=1}^{M} e^{d_k t} a_k \cos(2\pi f_k t + \theta_k) \tag{2.1}$$

Sem perda de generalidade, vamos supor que a amplitude $a_k \in \mathbb{R}^+$ pois, caso alguma amplitude a_{k_0} fosse negativa, então teríamos $a_k \cos(2\pi f_k + \theta_k) = -a_k \cos(2\pi f_k + \theta_k + \pi) = \hat{a}_k \cos(2\pi f_k + \hat{\theta}_k)$, com $\hat{a}_k = -a_k$ e $\hat{\theta}_k = \theta_k + \pi$. O fator de amortecimento d_k é, por definição, menor que zero, pois o modelo exige que a senóide tenha um decaimento e, portanto, reduza sua intensidade com o tempo.

As freqüências f_k neste modelo são divididas pela freqüência de amostragem, sendo normalizadas para o intervalo $[-\frac{1}{2}, \frac{1}{2}]$, mas sendo diferentes de zero, pois nesse caso teríamos um sinal constante que representaria apenas ruído. As freqüências também devem ser diferentes entre si pois, caso houvesse duas freqüências iguais, elas deveriam ser agrupadas na mesma senóide, recalculando sua amplitude e amortecimento. As fases θ_k pertencem ao intervalo $[-\pi, \pi)$, mas podem ser normalizadas para qualquer intervalo de tamanho 2π. A equação (2.1) pode ser escrita como:

$$\begin{aligned}
x(t) &= \sum_{k=1}^{M} a_k e^{d_k t} \cos(2\pi f_k t + \theta_k) = \\
&= \sum_{k=1}^{M} \frac{a_k}{2} (e^{d_k t} e^{i(2\pi f_k t + \theta_k)} + e^{d_k t} e^{-i(2\pi f_k t + \theta_k)}) = \\
&= \sum_{k=1}^{M} \frac{a_k}{2} (e^{(d_k + i2\pi f_k)t + i\theta_k} + e^{(d_k - i2\pi f_k)t - i\theta_k}) = \\
&= \sum_{k=1}^{M} \frac{a_k}{2} (e^{i\theta_k} z_k^t + e^{-i\theta_k} \bar{z}_k^t)
\end{aligned}$$

Ou seja, se $\alpha_k = \frac{1}{2} a_k e^{i\theta_k}$ e $z_k = e^{d_k + i2\pi f_k}$,

$$x(t) = \sum_{k=1}^{M} \alpha_k z_k^t + \bar{\alpha}_k \bar{z}_k^t \tag{2.2}$$

O número de pontos do sinal discretizado é N, que suporemos ser ímpar. Buscamos detectar $2M$ exponenciais $z_1, \bar{z}_1, z_2, \bar{z}_2, ..., z_M, \bar{z}_M$, conjugadas duas a duas, que representam as M senóides presentes no sinal. Assumimos, por enquanto, que conhecemos M e $M < \frac{N+1}{2}$.

Observe também que, como $d_k < 0$, então $e^{d_k} < 1$ e, como $z_k = e^{d_k} e^{i2\pi f_k}$, temos $|z_k| = |e^{d_k}| < 1$.

Seja agora $L = \frac{N+1}{2}$. Vamos definir a seguinte matriz:

$$H = \begin{pmatrix} x(0) & x(1) & \cdots & x(L-1) \\ x(1) & x(2) & \cdots & x(L) \\ \vdots & \vdots & \ddots & \vdots \\ x(L-1) & x(L) & \cdots & x(N-1) \end{pmatrix}$$

Sabemos que:

$$\begin{aligned} x(0) &= \alpha_1 + \bar{\alpha}_1 + ... + \alpha_M + \bar{\alpha}_M \\ x(1) &= \alpha_1 z_1 + \bar{\alpha}_1 \bar{z}_1 + ... + \alpha_M z_M + \bar{\alpha}_M \bar{z}_M \\ &\vdots \\ x(L-1) &= \alpha_1 z_1^{L-1} + \bar{\alpha}_1 \bar{z}_1^{L-1} + ... + \alpha_M z_M^{L-1} + \bar{\alpha}_M \bar{z}_M^{L-1} \end{aligned}$$

Assim, para a primeira coluna de H, temos a seguinte equação matricial:

$$\begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(L-2) \\ x(L-1) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ z_1 & \bar{z}_1 & \cdots & z_M & \bar{z}_M \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_1^{L-2} & \bar{z}_1^{L-2} & \cdots & z_M^{L-2} & \bar{z}_M^{L-2} \\ z_1^{L-1} & \bar{z}_1^{L-1} & \cdots & z_M^{L-1} & \bar{z}_M^{L-1} \end{pmatrix}}_{W} \begin{pmatrix} \alpha_1 \\ \bar{\alpha}_1 \\ \vdots \\ \alpha_M \\ \bar{\alpha}_M \end{pmatrix} =$$

$$= W \underbrace{\begin{pmatrix} \alpha_1 & & & & \\ & \bar{\alpha}_1 & & & \\ & & \ddots & & \\ & & & \alpha_M & \\ & & & & \bar{\alpha}_M \end{pmatrix}}_{A} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} =$$

$$= WA \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}$$

Portanto,

$$\begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(L-2) \\ x(L-1) \end{pmatrix} = WA \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x(1) \\ x(2) \\ \vdots \\ x(L-1) \\ x(L) \end{pmatrix} = WA \begin{pmatrix} z_1 \\ \bar{z}_1 \\ \vdots \\ z_M \\ \bar{z}_M \end{pmatrix}$$

$$\vdots$$

$$\begin{pmatrix} x(L-1) \\ x(L) \\ \vdots \\ x(N-2) \\ x(N-1) \end{pmatrix} = WA \begin{pmatrix} z_1^{L-1} \\ \bar{z}_1^{L-1} \\ \vdots \\ z_M^{L-1} \\ \bar{z}_M^{L-1} \end{pmatrix}$$

Assim, concatenando todas as colunas de H, temos:

$$H = WAW^T \tag{2.3}$$

Portanto, $W \in \mathbb{C}^{L \times 2M}$ e posto$(W) = 2M =$ posto(H). Logo, $Im(H) = Im(W)$. Nota-se, também, que W é uma matriz de Vandermonde associada aos pólos $z_1, \bar{z}_1, ..., z_M, \bar{z}_M$.

Seja $H = U\Sigma V^T$ a decomposição em valores singulares (SVD) econômica de H. Logo, $Im(H) = Im(U)$. Sabemos que H é real e simétrica. Por conseguinte, U é uma matriz de $2M$ autovetores de H ortonormais ($U^T U = I$). Por sua vez, $V = US$, em que S é uma matriz diagonal formada por 1 e -1. Este valor está relacionado às colunas associadas aos autovalores negativos. Desta forma, a matriz Σ contém o valor absoluto dos autovalores não nulos de H. Mas, como vimos que $Im(H) = Im(W)$, as colunas de U são combinações lineares das colunas de W, ou seja, elas se relacionam através de uma matriz inversível C:

$$U = WC^{-1} \tag{2.4}$$

Agora, vamos definir as matrizes $W_\downarrow, W_\uparrow \in \mathbb{C}^{L-1 \times 2M}$. $W_\downarrow$ é a matriz W sem a última linha e $W_\uparrow$ é a matriz W sem a primeira linha. Podemos descrevê-las da seguinte forma:

$$W_\downarrow = \underbrace{\begin{pmatrix} e_1^T \\ \vdots \\ e_{L-1}^T \end{pmatrix}}_{I_\downarrow} W \qquad W_\uparrow = \underbrace{\begin{pmatrix} e_2^T \\ \vdots \\ e_L^T \end{pmatrix}}_{I_\uparrow} W \tag{2.5}$$

em que os vetores e_k são os vetores canônicos de $\mathbb{R}^L$.

Seja Z a matriz diagonal tal que $Z_{11} = z_1$, $Z_{22} = \bar{z}_1$, ..., $Z_{2M-1,2M-1} = z_M$, $Z_{2M,2M} = \bar{z}_M$. Então, $W_\downarrow$ e $W_\uparrow$ se relacionam pela equação:

$$W_\downarrow = W_\uparrow Z \tag{2.6}$$

Essa propriedade é chamada de invariância rotacional. Observe que $W_\downarrow$ é de posto completo, pois $2M \leq L - 1$. Então,

$$Z = W_\downarrow^\dagger W_\uparrow$$

Note que a pseudo-inversa de $W_\downarrow$ é dada por $W_\downarrow^\dagger = (W_\downarrow^H W_\downarrow)^{-1} W_\downarrow^H$.

A partir de 2.4 e 2.5, obtemos:

$$\begin{aligned} I_\downarrow U = I_\downarrow W C^{-1} &\Rightarrow U_\downarrow = W_\downarrow C^{-1} \\ I_\uparrow U = I_\uparrow W C^{-1} &\Rightarrow U_\uparrow = W_\uparrow C^{-1} \end{aligned} \tag{2.7}$$

Como $posto(W_\downarrow) = posto(C) = posto(U_\downarrow) = 2M$, temos que $U_\downarrow^\dagger = C W_\downarrow^\dagger$. Então:

$$\begin{aligned} Z = W_\downarrow^\dagger W_\uparrow &= C^{-1} U_\downarrow^\dagger U_\uparrow C \\ \Rightarrow U_\downarrow^\dagger U_\uparrow &= C Z C^{-1} \end{aligned} \tag{2.8}$$

Observe que os autovalores da matriz $U_\downarrow^\dagger U_\uparrow$ são os pólos $z_1, \bar{z}_1, ..., z_M, \bar{z}_M$. Seja $\hat{C} Z \hat{C}^{-1}$ uma outra decomposição espectral de $U_\downarrow^\dagger U_\uparrow$. Como os elementos diagonais de Z são distintos dois a dois, as matrizes C e $\hat{C}$ se relacionam por uma matriz diagonal D: $C = \hat{C} D$, ou $\hat{C} = C D^{-1}$.

Vamos agora encontrar α_1, $\bar{\alpha}_1$, ... Temos que:

$$He_1 = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(L-2) \\ x(L-1) \end{pmatrix} = WA \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} = W \begin{pmatrix} \alpha_1 \\ \bar{\alpha}_1 \\ \vdots \\ \alpha_M \\ \bar{\alpha}_M \end{pmatrix} = UC \begin{pmatrix} \alpha_1 \\ \bar{\alpha}_1 \\ \vdots \\ \alpha_M \\ \bar{\alpha}_M \end{pmatrix} \tag{2.9}$$

Em [2], sugere-se que o cálculo das amplitudes, isto é, a resolução do problema $W\alpha = He_1$ pode ser feita sem a necessidade de se explicitar a matriz W. Buscaremos explicitar de maneira original essa solução.

Como U tem colunas ortonormais,

$$U^T He_1 = C \begin{pmatrix} \alpha_1 \\ \bar{\alpha}_1 \\ \vdots \\ \alpha_M \\ \bar{\alpha}_M \end{pmatrix} = CD^{-1}D \begin{pmatrix} \alpha_1 \\ \bar{\alpha}_1 \\ \vdots \\ \alpha_M \\ \bar{\alpha}_M \end{pmatrix} = \hat{C}D\alpha$$

Logo,

$$\alpha = D^{-1}\hat{C}^{-1}U^T He_1 \tag{2.10}$$

Para encontrarmos α, falta conhecermos a matriz D. Temos que $W = UC$ e $C = \hat{C}D$. Multiplicando W à esquerda pelo vetor canônico e_1^T, obtemos sua primeira linha, a qual é composta apenas de números 1.

$$\begin{aligned} W &= UC \\ e_1^T W &= e_1^T UC \\ \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} &= e_1^T U\hat{C}D \end{aligned}$$

Vamos agora multiplicar à direita os dois lados da equação, separadamente, por cada um dos vetores canônicos e_k, para $k = 1, ..., 2M$. Note que, ao multiplicarmos a matriz linha (1...1) por e_k, temos como resultado o número 1; quando multiplicamos a matriz diagonal D por e_k, temos $e_k d_{kk}$. Logo,

$$\begin{aligned} 1 = e_1^T U\hat{C}e_1 d_{11} \quad &\Rightarrow \quad d_{11} = \frac{1}{e_1^T U\hat{C}e_1} \\ &\vdots \\ 1 = e_1^T U\hat{C}e_{2M} d_{2M,2M} \quad &\Rightarrow \quad d_{2M,2M} = \frac{1}{e_1^T U\hat{C}e_{2M}} \end{aligned}$$

Portanto, α é dado por:

$$\begin{aligned} \alpha &= D^{-1}\hat{C}^{-1}U^T He_1 \\ \begin{pmatrix} \alpha_1 \\ \vdots \\ \bar{\alpha}_M \end{pmatrix} &= \begin{pmatrix} e_1^T U\hat{C}e_1 & & \\ & \ddots & \\ & & e_1^T U\hat{C}e_{2M} \end{pmatrix} \hat{C}^{-1}U^T He_1 \end{aligned} \tag{2.11}$$

Temos agora os parâmetros α_k e z_k do nosso modelo. Lembrando que $\alpha_k = \frac{1}{2}a_k e^{i\theta_k}$ e $z_k = e^{d_k}e^{i2\pi f_k}$, para cada senóide encontramos sua amplitude,

fase, freqüência e fator de amortecimento:

$$\begin{aligned} a_k &= 2|\alpha_k| \\ \theta_k &= \angle\alpha_k \\ f_k &= \frac{\angle z_k}{2\pi} \\ d_k &= \log|z_k| \end{aligned} \tag{2.12}$$

Parte do que foi feito acima está incluído no algoritmo ESPRIT, abreviação de Estimation of Signal Parameters via Rotational Invariance Techniques [18]. Esse algoritmo está resumido abaixo:

ESPRIT

Algoritmo 1 ESPRIT

$[U, \Sigma, V^T] = svd(H, 0)$
$\Phi = U_{\downarrow}^{\dagger} U_{\uparrow}$
$[\hat{C}, Z] = eig(\Phi)$

Utilizamos, na descrição de parte do algoritmo, a sintaxe do MATLAB: a instrução $[U, \Sigma, V^T] = svd(H, 0)$ significa que às matrizes U, Σ e V^T serão atribuídos os valores tais que $H = U\Sigma V^T$ seja a decomposição SVD econômica de H; a instrução $[\hat{C}, Z] = eig(\Phi)$ significa que a matriz $\hat{C}$ será a dos autovetores de Φ e a matriz diagonal Z será composta dos autovalores correspondentes na diagonal.

2.3 Ordem do Modelo

Na seção anterior, foram feitas deduções supondo-se que o som não possui ruídos e que sabemos exatamente o número M de senóides (que correspondem a $2M$ exponenciais complexas) a serem rastreadas. Se não houver ruído, $2M$ é o posto de H. Chamamos de **ordem do modelo** ao valor M (às vezes, neste texto, utilizamos $2M$ como a ordem do modelo). No caso em que não há ruído, o cálculo numérico do posto de H pode ser feito eficientemente via SVD, se M não for muito grande.

Na prática, dado um sinal qualquer, não sabemos quantas são suas componentes senoidais. Para um sinal de procedência desconhecida, essa quantidade pode ser infinita. Porém, para os métodos de alta resolução, vamos nos restringir a sons **harmônicos com um número limitado de parciais**, por exemplo, sons monofônicos gerados por instrumentos de orquestra.

Precisamos, então, de uma estimativa do valor de M para podermos aplicar o algoritmo. Há duas alternativas: usarmos um número pré-determinado ou usarmos um método para estimar o número de senóides do som em questão. Antes, vamos estudar o que acontece quando superestimamos o número de senóides. Nesse caso, os parâmetros que rastreamos estão incluídos no conjunto de parâmetros encontrados. Porém, se subestimarmos a ordem do problema, não teremos informações suficientes para reconstruir o sinal completo, embora os parâmetros encontrados nesse caso sejam aproximações de alguns dos $2M$ parâmetros verdadeiros.

2.3.1 Modelo de ordem superestimada

Vamos supor que temos M senóides presentes em um sinal, mas que trabalharemos com um modelo de ordem superestimada $\hat{M} > M$. Suponha que temos a matriz H e encontramos sua SVD numérica, da forma $H \cong \hat{U}\Sigma\hat{V}^T$. A matriz $\hat{U}$ é $L \times L$, $\hat{U} = [U_1|U_2]$, em que $U_1 \in \mathbb{R}^{L\times 2\hat{M}}$ é a matriz obtida na decomposição econômica. Por outro lado, a matriz U_1 também é composta de duas submatrizes, $U_1 = [U|X]$, sendo $U \in \mathbb{R}^{L\times 2M}$ a matriz que realmente procuramos, com a verdadeira ordem do problema, e X uma outra matriz que é estranha ao nosso modelo.

Vamos mostrar agora que, mesmo que tenhamos superestimado a ordem do modelo, ainda assim encontraremos as informações das senóides. Lembremos que os pólos que procuramos são $z_1, \bar{z}_1, ..., z_M, \bar{z}_M$, autovalores de $U_\downarrow^\dagger U_\uparrow$.

Teorema 2.2. $\{z_1, \bar{z}_1, ..., z_M, \bar{z}_M\} \subset \lambda(U_{1\downarrow}^\dagger U_{1\uparrow})$

Demonstração. Seja v autovetor de $U_\downarrow^\dagger U_\uparrow$ associado a algum autovalor $\lambda \in \{z_1, \bar{z}_1, ..., z_M, \bar{z}_M\}$, ou seja, $U_\downarrow^\dagger U_\uparrow v = \lambda v$. Seja $\Phi = U_\downarrow^\dagger U_\uparrow$. Como $posto(U_\downarrow) = posto(U_\uparrow) = 2M$, pela invariância rotacional, Φ é a única solução de $U_\uparrow = U_\downarrow \Phi$ (ver equações (2.6), (2.7) e (2.8)). Multiplicando por v à direita, temos:

$$U_\uparrow v = U_\downarrow \Phi v = U_\downarrow U_\downarrow^\dagger U_\uparrow v = \lambda U_\downarrow v$$

Seja agora o vetor $v_1 = \begin{pmatrix} v \\ 0 \end{pmatrix}$

$$U_1 v_1 = \begin{pmatrix} U_\uparrow & | & X_\uparrow \end{pmatrix} \begin{pmatrix} v \\ 0 \end{pmatrix} = U_\uparrow v = \lambda U_\downarrow v = \begin{pmatrix} U_\downarrow & | & X_\downarrow \end{pmatrix} \begin{pmatrix} v \\ 0 \end{pmatrix}$$

Como $\begin{pmatrix} U_\downarrow & | & X_\downarrow \end{pmatrix}$ é de posto completo, temos que:

$$\begin{pmatrix} U_\downarrow & | & X_\downarrow \end{pmatrix}^\dagger \begin{pmatrix} U_\uparrow & | & X_\uparrow \end{pmatrix} \begin{pmatrix} v \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} v \\ 0 \end{pmatrix}$$

Concluímos, assim, que λ é também autovalor de $(U_{1\downarrow}^{\dagger}U_{1\uparrow})$. $\square$

2.3.2 Algoritmos para estimativa de M

Apresentamos aqui dois algoritmos: o primeiro procura determinar M a partir da minimização de uma função; o segundo, que é utilizado quando a ordem do problema for superestimada, é uma estratégia para separar os autovalores que procuramos dos outros encontrados pela superestimação de M.

ESTER

Vamos apresentar um método para a estimativa do parâmetro M chamado ESTER (ESTimation ERror) [3]. Esse método busca encontrar um valor μ mais próximo de M possível, por meio de uma função que o avalia. A função de avaliação será a norma de uma matriz $E(\mu)$, que será definida a seguir. Esse processo é mais barato que a obtenção de U pela SVD de H [3].
Para um determinado μ, começamos o processo iterativo definindo:

$$U_0 = \begin{pmatrix} I_{2\mu} \\ 0 \end{pmatrix}_{N\times 2\mu}$$

As iterações serão dadas por:

$$\begin{cases} A_k &= HU_{k-1} \\ U_kR_k &= \mathrm{qr}\,(A_k, 0) \end{cases}$$

em que $\mathrm{qr}\,(A_k, 0)$ indica que o produto U_kR_k é a fatoração QR econômica da matriz A_k. Esse método, dito método de iteração ortogonal [12], converge para uma matriz que denominaremos $U(\mu)$. Definimos agora:

$$\begin{aligned} \Phi(\mu) &= U(\mu)_{\downarrow}^{\dagger}U(\mu)_{\uparrow} \\ E(\mu) &= U(\mu)_{\uparrow} - U(\mu)_{\downarrow}\Phi(\mu) \end{aligned}$$

A seguir, demonstramos uma proposição que descreve parcialmente o comportamento de $\|E(\mu)\|_2$.

Proposição 2.5. $(\forall \mu \in \{M, M+1, ..., N\})\ \|E(\mu)\|_2 \leq 1$

Demonstração. $E(\mu) = \left[U(\mu)_{\uparrow}(U(\mu)_{\uparrow}^{\dagger}) - U(\mu)_{\downarrow}(U(\mu)_{\downarrow}^{\dagger})\right] U(\mu)_{\uparrow}$, pois como vale que $\forall A$, $AA^{\dagger}A = A$, vale particularmente para $A = U(\mu)_{\uparrow}$. Assim,

$$\|E(\mu)\|_2 \leq \|U(\mu)_{\uparrow}(U(\mu)_{\uparrow}^{\dagger}) - U(\mu)_{\downarrow}(U(\mu)_{\downarrow}^{\dagger})\|_2 \cdot \|U(\mu)_{\uparrow}\|_2$$

Vamos agora calcular separadamente as duas normas. A primeira delas é a norma de uma diferença entre duas matrizes de projeção e, portanto, pelo Lema 2.3, $\|U(\mu)_\uparrow(U(\mu)_\uparrow^\dagger) - U(\mu)_\downarrow(U(\mu)_\downarrow^\dagger)\|_2 \leq 1$. Na segunda, temos que:

$$\begin{aligned} U(\mu)_\uparrow &= \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} U(\mu) \\ U(\mu)_\uparrow &= I_\uparrow U(\mu) \\ \Rightarrow \|U(\mu)_\uparrow\|_2 &\leq \|I_\uparrow\|_2 \cdot \|U(\mu)\|_2 \end{aligned}$$

Como $(\forall A)\ \|A^H\|_2 = \|A\|_2 = \sqrt{\max\lambda(A^H A)}$, temos que $\max\lambda(I_\uparrow) = 1$. Logo, $\|U(\mu)\|_2 \leq 1 \cdot 1 = 1$. Ou seja, $\|E(\mu)\|_2 \leq 1$. □

Consideremos, agora, a seguinte função J:

$$\begin{aligned} J : \{M, M+1, ..., N\} &\longmapsto [0,1] \\ \mu &\longmapsto \|E(\mu)\|_2 \end{aligned}$$

Note agora que, se $\mu = M$, então $U(\mu) = U$ e, portanto, $E(M) \equiv 0$ e $J(M) = 0$. Na prática, porém, a matriz $U(\mu)$ é obtida iterativamente. Logo, é difícil que encontremos $J(M) = 0$. Assim, busca-se μ tal que $J(\mu)$ seja um valor razoavelmente próximo de 0, sendo esse μ uma boa estimativa para o nosso M verdadeiro.

Método de predição linear para estimativa de M

Vamos, agora, propor um novo método para a estimativa de M, baseado em predição linear. Seja $\overrightarrow{H}$ a matriz H sem a última coluna. Seja $\overleftarrow{H}$ a matriz H sem a primeira coluna. Considere a equação:

$$\overleftarrow{H}\ c = \overrightarrow{H}\ e_1$$

Observe que, se existir solução c, então:

$$\overrightarrow{H} = \overleftarrow{H}\ S,$$

em que:

$$S = \begin{pmatrix} c_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{L-2} & 0 & \cdots & 1 \\ c_{L-1} & 0 & \cdots & 0 \end{pmatrix}$$

Em [6], foi mostrado que, se c é a solução de quadrados mínimos de menor norma, então os autovalores da matriz S que se encontram fora da bola unitária, ou seja, cujo valor absoluto seja maior que 1, correspondem aos parâmetros que nos interessam. Basta contá-los e teremos o nosso valor para $2M$.

Para calcular os parâmetros desejados, procederemos de forma análoga à que utilizamos com o ESPRIT. Para isso, vamos considerar a decomposição espectral $S^T = \hat{W}^T \Lambda \hat{W}^{-T}$, ou seja, $S = \hat{W}^{-1} \Lambda \hat{W}$. Vemos que os autovalores de S e S^T são os mesmos. Seja $v = (v_1, ..., v_{L-1})$ autovetor de S^T associado a λ. Temos:

$$\begin{cases} \lambda v_1 &= c_1 v_1 + ... + c_{L_1} v_{L-1} \\ \lambda v_2 &= v_1 \\ \lambda v_3 &= v_2 \\ \vdots & \vdots \\ \lambda v_{L-1} &= v_{L-2} \end{cases} \Rightarrow \begin{cases} v_{L-1} &= 1 \\ v_{L-2} &= \lambda \\ v_{L-3} &= \lambda^2 \\ \vdots & \vdots \\ v_1 &= \lambda^{L-1} \end{cases}$$

Portanto, os autovetores associados a λ_k são da forma $(\lambda_k^{L-1}, ..., \lambda_k, 1)$. Ao fatorarmos S^T, obtemos $\hat{W}^T$ da forma:

$$\begin{pmatrix} \lambda_1^{L-1} & \cdots & \lambda_n^{L-1} \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} = \begin{pmatrix} d_1 \lambda_1^{L-1} & \cdots & d_n \lambda_n^{L-1} \\ \vdots & & \vdots \\ d_1 & \cdots & d_n \end{pmatrix}$$

Para encontrarmos a matriz W do nosso problema, basta dividirmos cada coluna k de $\hat{W}$ por $d_k \lambda_k^{L-1}$, ou seja:

$$W = \hat{W} \begin{pmatrix} \frac{1}{e_1^T \hat{W} e_1} & & \\ & \ddots & \\ & & \frac{1}{e_1^T \hat{W} e_n} \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ \lambda_1^{-1} & \cdots & \lambda_n^{-1} \\ \vdots & & \vdots \\ \lambda_1^{-(L-1)} & \cdots & \lambda_n^{-(L-1)} \end{pmatrix}$$

Podemos agora encontrar nossos parâmetros α_k, resolvendo o sistema:

$$W\alpha = \begin{pmatrix} x(0) \\ \vdots \\ x(L-1) \end{pmatrix}$$

A solução de mínimos quadrados é:

$$\alpha = W^{\dagger} \begin{pmatrix} x(0) \\ \vdots \\ x(L-1) \end{pmatrix}$$

2.4 Algoritmos Adaptativos

Os métodos de alta resolução que vimos até agora detectam as características globais de um determinado sinal, a partir da matriz H. Em geral, essa matriz H, formada por todos os pontos do sinal, é muito grande para processamento numérico. Uma idéia inicial para lidar com essa impossibilidade é reduzir o número de pontos a serem analisados e avançar no sinal (isto é, no tempo) através de janelas, dando origem aos algoritmos adaptativos. Este é o conceito intuitivo de janelas deslizantes. Nossa variável L será agora arbitrária, maior que $2M$ e, obviamente, bem menor que $\frac{N+1}{2}$. Vamos agora formalizar o conceito de janelas deslizantes. Seja $H(t)$ definida por:

$$H(t) = \begin{pmatrix} x(t) & x(t+1) & \cdots & x(t+L-1) \\ x(t+1) & x(t+2) & \cdots & x(t+L) \\ \vdots & \vdots & \cdots & \vdots \\ x(t+L-1) & x(t+L) & \cdots & x(t+2L-2) \end{pmatrix} \tag{2.13}$$

Chamaremos as matrizes $H(0)$, $H(1)$, ..., $H(N-2L)$ de janelas deslizantes, pois, a cada avanço no tempo, a matriz $H(t+1)$ tem uma coluna retirada e outra acrescentada em relação à matriz $H(t)$, enquanto as colunas comuns entre elas são deslocadas para a esquerda. A seguir descreveremos alguns algoritmos que utilizam essa técnica de janelas deslizantes.

2.4.1 Método de Iteração Ortogonal Adaptativa

Para contornarmos o cálculo da SVD de $H(t)$, $H(t) = U(t)\Sigma(t)V^T(t)$, vamos aproximar $U(t)$ por uma matriz $Q(t)$ proveniente do método de iteração ortogonal [24]: dada a matriz inicial $Q(0) = \begin{pmatrix} I_{2M} \\ 0 \end{pmatrix}$, a cada passo $t = 0, 1, 2, ...$, fazemos:

$$\begin{cases} A(t) &= H(t)Q(t-1) \\ Q(t)R(t) &= A(t) \end{cases}$$

Porém, dessa maneira, começamos com uma má aproximação para $U(0) = Q(0)$. Uma alternativa é realizar o processo iterativo inicialmente com a matriz $H(0)$ fixa, para obtermos $Q(0)$ mais próxima de $U(0)$. Para $k = 1, 2, ..., T$ teríamos então:

$$\begin{cases} A_k &= H(0)Q_{k-1} \\ Q_kR_k &= A_k \end{cases}$$

e, finalmente, tomaríamos $Q(0) = Q_T$.

A complexidade desse método é de $\mathcal{O}(ML^2)$ operações por iteração.

2.4.2 Método de Strobach

Vamos agora apresentar outro processo iterativo, computacionalmente mais econômico, para aproximarmos $U(t)$ em cada janela. Esse algoritmo é chamado método de Strobach [24]. Começaremos definindo o vetor linha:

$$h(t) = x(t : t + L - 1)$$

Assim, temos que a matriz $H(t)$ é escrita em função dos vetores $h(t)$ da seguinte forma:

$$H(t) = \begin{pmatrix} h(t) \\ \vdots \\ h(t + L - 1) \end{pmatrix}$$

Vamos definir agora a matriz $\tilde{H}(t)$ acrescentando uma linha à matriz $H(t-1)$.

$$\tilde{H}(t) = \begin{pmatrix} H(t-1) \\ h(t + L - 1) \end{pmatrix} = \begin{pmatrix} h(t-1) \\ H(t) \end{pmatrix}$$

Consideremos agora as matrizes $\tilde{A}(t)$ e $A(t)$ definidas a seguir:

$$\tilde{A}(t) = \tilde{H}(t)Q(t-1) = \begin{pmatrix} h(t-1) \cdot Q(t-1) \\ H(t) \cdot Q(t-1) \end{pmatrix} = \begin{pmatrix} h(t-1) \cdot Q(t-1) \\ A(t) \end{pmatrix}$$

Por outro lado, também temos que:

$$\tilde{A}(t) = \tilde{H}(t)Q(t-1) = \begin{pmatrix} H(t-1) \cdot Q(t-1) \\ h(t + L - 1) \cdot Q(t-1) \end{pmatrix}$$

Vamos agora aproximar $H(t)$ por $\hat{H}(t) = Q(t)R(t)Q(t-1)^T$. Assim,

$$A(t) = H(t)Q(t-1) = Q(t)R(t) = \hat{H}(t)Q(t-1).$$

Defina agora $\hat{A}(t)$ de tal forma que:

$$\begin{pmatrix} \hat{h}(t-1)Q(t-1) \\ A(t) \end{pmatrix} = \begin{pmatrix} \hat{H}(t-1)Q(t-1) \\ h(t + L - 1)Q(t-1) \end{pmatrix} = \tilde{A}(t)$$

em que $\hat{h}(t-1) = e_1^T \hat{H}(t-1)$. Observe que:

$$\hat{H}(t-1)Q(t-1) = Q(t-1)R(t-1)Q(t-2)^T Q(t-1) = A(t-1)\theta(t-1)$$

em que $\theta(t) = Q(t-1)^T Q(t)$. Definindo $a(t) = \hat{h}(t-1)Q(t-1)$ e $b(t) = h(t+L-1)Q(t-1)$, chegamos ao seguinte algoritmo para encontrar a matriz $Q(t)$ para cada t:

Esse algoritmo reduz a ordem do problema para $\mathcal{O}(M^2 L)$ operações por iteração. Porém, essa complexidade ainda não é satisfatória.

Algoritmo 2 Método de Strobach

$Q_0 = \begin{pmatrix} I_{2M} \\ 0 \end{pmatrix}$
$\theta(0) = I_{2M}$
$A(0) = H(0)Q(0)$
for $t = 1, 2, ...$ **do**
  $b = h(t + L - 1)Q(t - 1)$
  $\tilde{A} = [A(t - 1)\theta(t - 1); b]$
  $A = \tilde{A}(2 : L + 1, :)$
  $[Q(t), R(t)] = \text{ qr }(A, 0)$
  $\theta(t) = Q(t - 1)^T Q(t)$
end for

2.4.3 Algoritmos baseados em Aproximações Projetivas

Buscaremos formas econômicas de encontrarmos a matriz U_k, sem a necessidade de fatoração SVD ou QR a cada passo. Para isso, utilizaremos algoritmos baseados em aproximações projetivas, derivados do método PAST [25]. Suas principais variantes são os algoritmos NP3 [13] e OPAST [1], que foram desenvolvidos originalmente para rastrear subespaços dominantes de matrizes de covariância C_k, cuja atualização é de posto 1. Nossa intenção é adaptar esses métodos para a matriz $C_k = H_k^2 = H(k)^2$, cujo espaço coluna é o mesmo de $H(k)$ e cuja atualização será de posto 2. Esses métodos possuem complexidade de $\mathcal{O}(ML)$ operações por iteração.

Vamos supor que os autovalores de H são da forma $|\lambda_1| > |\lambda_2| > ... > |\lambda_{2M}| >> |\lambda_{2M+1}| \approx ... \approx |\lambda_{2L}| \approx 0$, ou seja, o autovalor λ_{2M} de H é muito maior que λ_{2M+1}, sendo que a partir daí os autovalores são muito próximos de 0. O subespaço correspondente aos $2M$ maiores autovalores em valor absoluto é conhecido como subespaço dominante. Temos, então, que $U = [U_{2M}|U_{L-2M}]$ e queremos um método iterativo que convirja para U_{2M}.

Seja $C = HH^T$. C é uma matriz simétrica definida positiva: $C = U\Lambda U^T$, em que $\Lambda = diag(\lambda_1^2, ..., \lambda_{2L}^2) = \begin{pmatrix} \Sigma & \\ & \Lambda_{L-2M} \end{pmatrix}$, e $\Sigma = \Lambda_{2M}$ é a matriz diagonal com os quadrados dos $2M$ maiores autovalores de H.

Consideremos agora a função $J : \mathbb{R}^{L \times 2M} \to \mathbb{R}$, definida por

$$J(Y) = tr(C) - 2tr(Y^T CY) + tr(Y^T CYY^T Y) \tag{2.14}$$

Então, temos o seguinte teorema:

Teorema 2.3. *Y é de posto completo e $\nabla J(Y) = 0 \Leftrightarrow Y = VQ^T$, em que*

V é uma matriz de autovetores linearmente independentes associados a $2M$ autovalores de C, $V^TV = I$ e Q é uma matriz ortogonal de ordem $2M$.

Demonstração.

$$J(Y+H) = tr(C) - 2tr((Y+H)^TC(Y+H)) +$$
$$+tr((Y+H)^TC(Y+H)(Y+H)^T(Y+H)) =$$
$$= tr(C) - 2tr(Y^TCY) - 2tr(H^TCY + Y^TCH) +$$
$$+tr(Y^TCYY^TY) + tr(Y^TCHY^TY) + tr(H^TCYY^TY) +$$
$$+tr(Y^TCYH^TY) + tr(Y^TCYY^TH) + O(\|H\|^2)$$

Como o traço é uma função linear, $tr(A^T) = tr(A)$ e $tr(AB) = tr(BA)$,

$$J(Y+H) = J(Y) - 4tr(H^TCY) + tr(H^TCYY^TY) + tr(Y^TCHY^TY) +$$
$$+tr(Y^TCYH^TY) + tr(Y^TCYY^TH) + O(\|H\|^2) =$$
$$= J(Y) - 4tr(H^TCY) + 2tr(H^TCYY^TY) + 2tr(H^TYY^TCY) + O(\|H\|^2) =$$
$$= J(Y) + tr(H^T[-4CY + 2CYY^TY + 2YY^TCY]) + O(\|H\|^2)$$

Assim, temos que $\nabla J(Y) = (-4CY + 2CYY^TY + 2YY^TCY)$. Vamos, agora, verificar para quais matrizes Y, de posto completo, $\nabla J(Y) = 0$.

$$\begin{aligned} \nabla J(Y) = 0 \quad &\Rightarrow \quad 0 = Y^T\nabla J(Y) = -2Y^TCY + Y^TCYY^TY + Y^TYY^TCY \\ &\Rightarrow \quad Y^TCY(Y^TY - I) + (Y^TY - I)(Y^TCY) = 0 \end{aligned}$$

Como C é SDP e Y é de posto completo, então, pelo Lema 2.5, Y^TCT é SDP. Portanto, pelo Lema 2.6, temos que $Y^TY - I = 0$, ou seja, $Y^TY = I$. Logo,

$$0 = \frac{1}{2}\nabla J(Y) = -2CY + CY + YY^TCY = -CY + YY^TCY$$

e, como Y^TCY é SDP, logo inversível, temos:

$$CY = YY^TCY \Leftrightarrow Y = CY(Y^TCY)^{-1} \tag{2.15}$$

Por ser SDP, Y^TCY possui decomposição espectral: $Y^TCY = Q\Sigma_YQ^T$, em que Q é uma matriz ortogonal de ordem $2M$ e Σ_Y é diagonal. Assim, por (2.15), $CY = YQ\Sigma_YQ^T \Rightarrow C(YQ) = (YQ)\Sigma_Y$. Como Y é de posto completo e Q ortogonal, YQ é de posto completo. Assim, YQ é uma matriz de autovetores LI de C associados às entradas diagonais de Σ_Y correspondentes, que são autovalores de C. Como C é uma matriz simétrica, existe uma base ortonormal de autovetores de C associados às entradas de Σ_Y. Seja V a matriz cujas colunas são os $2M$ vetores dessa base. Logo, $Y = VQ^T$.

Para provarmos a recíproca, basta substituirmos $Y = V_{2M}Q$ em $\nabla J(Y)$. □

Teorema 2.4. *O mínimo global de $J(Y)$ é alcançado se $Y = U_{2M}Q$, em que Q é uma matriz ortogonal qualquer.*
A demonstração desse teorema pode ser encontrada em [25].

Se Y é uma matriz de posto completo e C é uma matriz simétrica, então Y^TC^2Y é uma matriz simétrica definida não negativa. Logo, existe uma única matriz simétrica definida não negativa Z tal que $Z^2 = Y^TC^2Y$: $Z = (Y^TC^2Y)^{\frac{1}{2}}$. Se C é uma matriz SDP, então $(Y^TC^2Y)^{-\frac{1}{2}} = ((Y^TC^2Y)^{\frac{1}{2}})^{-1}$.

Proposição 2.6. *Sejam Y uma matriz de posto completo e C uma matriz SDP. Se $G = CY(Y^TC^2Y)^{-\frac{1}{2}}$, então $G^TG = I$.*

Demonstração.

$$G^TG = (Y^TC^2Y)^{-\frac{1}{2}}Y^TC^TCY(Y^TC^2Y)^{-\frac{1}{2}} =$$

$$= (Y^TC^2Y)^{-\frac{1}{2}}(Y^TC^2Y)(Y^TC^2Y)^{-\frac{1}{2}} = I$$

. □

Podemos aplicar o algoritmo de iteração ortogonal adaptativo nas matrizes $C(t) = H(t)H(t)^T$. Observemos que a matriz de colunas ortonormais $Q(t)$, quando t avança, converge para o subespaço dominante de $H(t)$, que é o mesmo de $C(t)$. Obtemos $Q(t)$ a partir da decomposição QR de $C(t)Q(t-1)$. Mas poderíamos obter $Q(t)$ a partir de outra decomposição de $C(t)Q(t-1)$. Uma idéia é usar a proposição acima para obtermos $Q(t)$:

$$Q(t) = C(t)Q(t-1)(Q(t-1)^TC^2(t)Q(t-1))^{-\frac{1}{2}} \tag{2.16}$$

Os dois algoritmos a seguir procuram calcular essa expressão de forma aproximada, porém eficiente.

2.4.4 Algoritmo NP3

Seja $C(t) = H(t)H(t)^T = H(t)^2$, em que $H(t)$ é a matriz de Hankel definida em (2.13). Vamos utilizar a notação $C_k = C(k)$ e $H_k = H(k)$. Sejam $h_1 = H_{k-1}e_1$, $h_2 = H_ke_n$, $r_k = (h_1|h_2)$ e $\hat{r}_k = (h_1| - h_2)$. Temos:

$$\begin{aligned} C_{k-1} = H_{k-1}H_{k-1}^T &= (h_1|H_{\uparrow k-1})\begin{pmatrix} h_1^T \\ H_{\uparrow k-1}^T \end{pmatrix} = \\ \Rightarrow C_{k-1} &= H_{\uparrow k-1}H_{\uparrow k-1}^T + h_1h_1^T \end{aligned}$$

Por outro lado:

$$\begin{aligned} C_k &= H_k H_k^T = (H_{\uparrow k-1}|h_2) \begin{pmatrix} H_{\uparrow k-1}^T \\ h_2^T \end{pmatrix} = \\ &= H_{\uparrow k-1} H_{\uparrow k-1}^T + h_2 h_2^T = \\ &= H_{k-1} H_{k-1}^T - h_1 h_1^T + h_2 h_2^T = \\ &= C_{k-1} - (h_1|h_2) \begin{pmatrix} h_1^T \\ -h_2^T \end{pmatrix} = \\ &= C_{k-1} - r_k \hat{r}_k^T \end{aligned} \tag{2.17}$$

Agora, definiremos $Y_k = C_k U_{k-1}$, $S_k = U_{k-1}^T \hat{r}_k$ e faremos a seguinte aproximação, considerando que $U_k \approx U_{k-1}$:

$$\begin{aligned} Y_k &= C_k U_{k-1} = (C_{k-1} - r_k \hat{r}_k^T) U_{k-1} = C_{k-1} U_{k-1} - r_k S_k^T \\ \Rightarrow Y_k &\approx C_{k-1} U_{k-2} - r_k S_k^T = Y_{k-1} - r_k S_k^T \end{aligned}$$

Vamos agora definir Z_k:

$$\begin{aligned} Z_k &= Y_k^T Y_k = (Y_{k-1} - r_k S_k^T)^T (Y_{k-1} - r_k S_k^T) = \\ &= (Y_{k-1}^T - S_k r_k^T)(Y_{k-1} - r_k S_k^T) = \\ &= Y_{k-1}^T Y_{k-1} - S_k r_k^T Y_{k-1} - Y_{k-1}^T r_k S_k^T + S_k r_k^T r_k S_k^T = \\ &= Z_{k-1} - S_k r_k^T Y_{k-1} - Y_{k-1}^T r_k S_k^T + S_k r_k^T r_k S_k^T \end{aligned}$$

Utilizando (2.16):

$$\begin{aligned} U_k &= C_k U_{k-1} (U_{k-1}^T C_k^2 U_{k-1})^{-\frac{1}{2}} = \\ &= C_k U_{k-1} (U_{k-1}^T C_k^T C_k U_{k-1})^{-\frac{1}{2}} = \\ &= Y_k (Y_k^T Y_k)^{-\frac{1}{2}} = \\ &= Y_k Z_k^{-\frac{1}{2}} \end{aligned} \tag{2.18}$$

Temos, por outro lado, que $Z_k =$

$$Z_{k-1}^{\frac{1}{2}} (I - Z_{k-1}^{-\frac{1}{2}} S_k r_k^T Y_{k-1} Z_{k-1}^{-\frac{1}{2}} - Z_{k-1}^{-\frac{1}{2}} Y_{k-1}^T r_k S_k^T Z_{k-1}^{-\frac{1}{2}} + Z_{k-1}^{-\frac{1}{2}} S_k r_k^T r_k S_k^T Z_{k-1}^{-\frac{1}{2}}) Z_{k-1}^{\frac{1}{2}}$$

Vamos calcular uma raiz quadrada não simétrica para Z_k pelo Corolário 2.2:

$$\begin{aligned} Z_k^{-\frac{1}{2}} = Z_{k-1}^{-\frac{1}{2}} (I - Z_{k-1}^{-\frac{1}{2}} S_k r_k^T Y_{k-1} Z_{k-1}^{-\frac{1}{2}} - \\ - Z_{k-1}^{-\frac{1}{2}} Y_{k-1}^T r_k S_k^T Z_{k-1}^{-\frac{1}{2}} + Z_{k-1}^{-\frac{1}{2}} S_k r_k^T r_k S_k^T Z_{k-1}^{-\frac{1}{2}})^{-\frac{1}{2}} \end{aligned}$$

$$\Rightarrow Z_k^{-\frac{1}{2}} = Z_{k-1}^{-\frac{1}{2}}(I - ab^T - ba^T + aca^T)^{-\frac{1}{2}}$$

em que $a = Z_{k-1}^{-\frac{1}{2}} S_k$, $b = Z_{k-1}^{-\frac{1}{2}} Y_{k-1}^T r_k$ e $c = r_k^T r_k$. Observemos que a matriz $(-ab^T - ba^T + aca^T)$ é simétrica e pode ser escrita como:

$$-ab^T - ba^T + aca^T = \underbrace{\begin{pmatrix} a & b \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} -b^T + ca^T \\ -a^T \end{pmatrix}}_{B}$$

Considere, agora:

$$BA = \begin{pmatrix} -b^T + ca^T \\ -a^T \end{pmatrix} \begin{pmatrix} a & b \end{pmatrix} = \begin{pmatrix} -b^T a + ca^T a & -b^T b + ca^T b \\ -a^T a & -a^T b \end{pmatrix} \quad (2.19)$$

A matriz BA é 4×4 e seus autovalores são σ_1, σ_2, σ_3, σ_4, os mesmos autovalores não nulos de $-ab^T - ba^T + aca^T$. Podemos escrever AB como $\sigma_1 v_1 v_1^T + \sigma_2 v_2 v_2^T + \sigma_3 v_3 v_3^T + \sigma_4 v_4 v_4^T$, em que v_1, v_2, v_3, v_4 são autovetores ortonormais relacionados a $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ respectivamente.

Seja $F = (I + \sigma_1 v_1 v_1^T + ... + \sigma_4 v_4 v_4^T)^{-\frac{1}{2}}$.

$$F = (I + \sigma_1 v_1 v_1^T + \sigma_2 v_2 v_2^T + \sigma_3 v_3 v_3^T + \sigma_4 v_4 v_4^T)^{-\frac{1}{2}} =$$

$$= (v_1 \dots v_4 \dots v_n) \begin{pmatrix} (1+\sigma_1)^{-\frac{1}{2}} & & & & & \\ & \ddots & & & & \\ & & (1+\sigma_4)^{-\frac{1}{2}} & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} (v_1 \dots v_4 \dots v_n)^T =$$

$$= V \begin{pmatrix} (1+\sigma_1)^{-\frac{1}{2}} - 1 + 1 & & & & & \\ & \ddots & & & & \\ & & (1+\sigma_4)^{-\frac{1}{2}} - 1 + 1 & & & \\ & & & 0+1 & & \\ & & & & \ddots & \\ & & & & & 0+1 \end{pmatrix} V^T =$$

$$= I + (v_1 v_2 v_3 v_4) \begin{pmatrix} (1+\sigma_1)^{-\frac{1}{2}} - 1 & & \\ & \ddots & \\ & & (1+\sigma_4)^{-\frac{1}{2}} - 1 \end{pmatrix} (v_1 v_2 v_3 v_4)^T =$$

$$= I - \gamma_1 v_1 v_1^T - \gamma_2 v_2 v_2^T - \gamma_3 v_3 v_3^T - \gamma_4 v_4 v_4^T$$

em que $\gamma_i = 1 - \frac{1}{\sqrt{\sigma_i + 1}}$. Falta agora acharmos os autovetores ortonormais v_1, v_2, v_3, v_4 de AB, de forma que $AB = \sigma_1 v_1 v_1^T + ... + \sigma_4 v_4 v_4^T$.

Como AB é simétrica, então é diagonalizável. Já BA é uma matriz 4×4. Vamos supor que BA é de posto completo e, portanto, não possui autovalores

nulos. Logo, pelo Lema 2.4, BA também é diagonalizável e pode ser escrita como $BA = J\Lambda J^{-1}$. Portanto, $ABA = AJ\Lambda J^{-1} \Rightarrow (AB)(AJ) = (AJ)\Lambda$. Ou seja, AJe_i, para $i = 1, .., 4$ são autovetores LI associados a $\sigma_1, ..., \sigma_4$. Falta apenas aplicarmos a eles o processo de ortonormalização, que pode ser feito por fatoração QR ou Gram-Schmidt, para obtermos v_1, v_2, v_3, v_4.

Na prática, essa forma de encontrar os autovalores e autovetores da matriz AB não gerou bons resultados e buscamos outra forma mais eficiente para isso, descrita a seguir.

Considere a fatoração QR econômica da matriz A: $QR = A \Leftrightarrow R = Q^T A$. Como AB é uma matriz simétrica, $RBQ = Q^T ABQ = (Q^T (AB)^T Q)^T$ também é uma matriz simétrica. Logo, pode ser decomposta como $RBQ = PDP^T$, com D diagonal contendo seus autovalores e P ortogonal com seus respectivos autovetores. Por outro lado, $RBQ = PDP^T \Leftrightarrow QRBQ = QPDP^T \Leftrightarrow (AB)(QP) = (QP) \Leftrightarrow AB = (QP)D(QP)^T$. Assim, D é a matriz contendo os autovalores de AB e QP é uma matriz ortogonal contendo seus autovetores. Ou seja, utilizamos Q e R gerados pela decomposição QR econômica da matriz A, $L \times 4$, para definir a matriz RBQ, 4×4; calculada uma matriz de autovetores P ortogonal, as colunas de QP são os vetores $v_1, ..., v_4$ procurados.

Retomando (2.18), vamos buscar uma atualização de U_k em função de U_{k-1} e daquela raiz assimétrica de Z_k.

$$\begin{aligned}
U_k &= Y_k Z_k^{-\frac{1}{2}} = \\
&= [Y_{k-1} - r_k S_k^T][Z_{k-1}^{-\frac{1}{2}}(I - \gamma_1 v_1 v_1^T - \gamma_2 v_2 v_2^T - \gamma_3 v_3 v_3^T - \gamma_4 v_4 v_4^T)] = \\
&= Y_{k-1} Z_{k-1}^{-\frac{1}{2}}[I - \gamma_1 v_1 v_1^T - ... - \gamma_4 v_4 v_4^T] - r_k S_k^T Z_{k-1}^{-\frac{1}{2}}[I - ...] = \\
&= U_{k-1}[I - pq^T] - r_k S_k^T Z_{k-1}^{-\frac{1}{2}}[I - pq^T] = \\
&= U_{k-1} - U_{k-1} pq^T - r_k a^T + r_k a^T pq^T \qquad (2.20)
\end{aligned}$$

Se definirmos $w = r_k a^T p - U_{k-1} p$, $E = [w| - r_k]$ e $G = [q|a]$, então poderemos reescrever (2.20) como:

$$U_k = U_{k-1} + EG^T$$

O Algoritmo 3 implementa em pseudocódigo o método NP3 para a atualização da matriz $U(t)$, cujas colunas são uma base ortogonal para o subespaço dominante de $C(t)$.

2.4.5 OPAST

Esse método baseia-se no fato de que, se

$$\hat{U}_k = C_k U_{k-1} (U_{k-1}^T C_k U_{k-1})^{-1},$$

Algoritmo 3 NP3

$U = \begin{pmatrix} I_{2M} \\ 0 \end{pmatrix}$
$Y = H(0)H(0)U$
$Z = (Y^T Y)^{-\frac{1}{2}}$
$U = YZ$
for $k = 1, ..., N - 2L$ **do**
  $h_1 = x(k : k + L - 1)$
  $h_2 = x(k + L : k + 2L - 1)$
  $r = [h_1|h_2]$
  $s = U[h_1| - h_2]$
  $a = Zs$
  $b = Z(Yr)$
  $c = r^t r$
  $A = [a|b]$
  $B = \begin{pmatrix} -b^T + ca^T \\ -a^T \end{pmatrix}$
  $[Q, R] = qr(A, 0)$
  $F = RBQ$
  $[V, D] = eig(F)$
  $q = QV$
  for $j = 1, ..., 4$ **do**
    $\gamma(j) = 1 - \frac{1}{\sqrt{1 + D(j, j)}}$
    $p(:, j) = \gamma(j)q(:, j)$
  end for
  $w = (ra^T)p - Up$
  $E = [w| - r]$
  $G = [q|a]$
  $Z = (I - pq^T)Z$
  $Y = Y - rs^T$
  $U = U + EG^T$
end for

então,

$$U_k = \hat{U}_k(\hat{U}_k^T\hat{U}_k)^{-\frac{1}{2}}$$

é uma matriz cujas colunas são vetores ortonormais que geram C_kU_{k-1}. Note que:

$$\hat{U}_k^T\hat{U}_k = (U_{k-1}^TC_kU_{k-1})^{-1}(U_{k-1}^TC_k^2U_{k-1})(U_{k-1}^TC_kU_{k-1})^{-1}$$

Logo,

$$(\hat{U}_k^T\hat{U}_k)^{-\frac{1}{2}} = (U_{k-1}^TC_kU_{k-1})(U_{k-1}^TC_k^2U_{k-1})^{-\frac{1}{2}}Q$$

em que Q é uma matriz ortogonal. Além disso,

$$U_k = \hat{U}_k(\hat{U}_k^T\hat{U}_k)^{-\frac{1}{2}} = C_kU_{k-1}(U_{k-1}^TC_k^2U_{k-1})^{-\frac{1}{2}}$$

Manteremos $Y_k = C_kU_{k-1} \approx Y_{k-1} - r_kS_k^T$, como no método anterior, e redefiniremos $Z_k = (U_{k-1}^TC_kU_{k-1})^{-1} = (U_{k-1}^TY_k)^{-1}$, de forma que $\hat{U}_k = Y_kZ_k$. Vamos calcular a seguir a atualização para $\hat{U}_k$.

$$\begin{aligned}
Z_k &= [U_{k-1}^TY_k]^{-1} = [U_{k-1}^T(Y_{k-1} - r_kS_k^T)]^{-1} = \\
&= [U_{k-1}^TY_{k-1} - U_{k-1}^Tr_kS_k^T]^{-1} \approx \\
&\approx [U_{k-2}Y_{k-1} - U_{k-1}^Tr_kS_k^T]^{-1} = \\
&= [U_{k-2}^TY_{k-1} - \underbrace{U_{k-1}r_k}_{y_k}S_k^T]^{-1} = \\
&= [U_{k-2}Y_{k-1} - y_kS_k^T]^{-1} = \\
&\overset{(S.M.)}{=} Z_{k-1} + \underbrace{Z_{k-1}y_k}_{q_k}\underbrace{(I - \overbrace{S_k^TZ_{k-1}}^{t_k^T}y_k)^{-1}}_{\gamma_k}\underbrace{S_k^TZ_{k-1}}_{t_k} = \\
&= Z_{k-1} + q_k\gamma_kt_k^T
\end{aligned}$$

A igualdade sinalizada por (S.M.) justifica-se pela aplicação da fórmula de Sherman-Morrison para a obtenção da inversa (Lema 2.1). Vamos agora

estudar a atualização para $\hat{U}_k$:

$$\begin{aligned}
\hat{U}_k &= Y_k Z_k = (Y_{k-1} - r_k S_k^T)(Z_{k-1} + q_k \gamma_k t_k^T) = \\
&= Y_{k-1} Z_{k-1} - r_k S_k^T Z_{k-1}^T + Y_{k-1} q_k \gamma_k t_k^T - r_k S_k^T q_k \gamma_k t_k = \\
&= U_{k-1} - r_k t_k^T + Y_{k-1} Z_{k-1} y_k \gamma_k t_k^T - r_k S_k^T Z_{k-1} y_k \gamma_k t_k^T = \\
&= U_{k-1} - (r_k - U_{k-1} y_k \gamma_k + r_k t_k^T y_k \gamma_k) t_k^T = \\
&= U_{k-1} - [r_k (I + t_k^T y_k \gamma_k) - U_{k-1} y_k \gamma_k] t_k^T = \\
&= U_{k-1} - [r_k (\gamma_k - \gamma_k + I + t_k^T y_k \gamma_k) - U_{k-1} y_k \gamma_k] t_k^T = \\
&= U_{k-1} - [r_k [\gamma_k + I - \overbrace{(I - t_k^T y_k)\gamma_k}^{I}] - U_{k-1} y_k \gamma_k] t_k^T = \\
&= U_{k-1} - (r_k \gamma_k - U_{k-1} y_k \gamma_k) t_k^T = \\
&= U_{k-1} - \underbrace{(r_k - U_{k-1} y_k)\gamma_k}_{p_k} t_k^T = \\
&= U_{k-1} - p_k t_k^T
\end{aligned}$$

Temos que $U_{k-1}^T p_k = (U_{k-1}^T r_k - U_{k-1}^T U_{k-1} y_k)\gamma_k$. Observemos que, se U_{k-1} é ortogonal, então $U_{k-1}^T p_k = (U_{k-1}^T r_k - y_k)\gamma_k = 0\gamma_k = 0$. Vamos supor U_{k-1} ortogonal. Logo,

$$\begin{aligned}
\hat{U}_k &= U_{k-1} - p_k t_k^T = \\
\Rightarrow \hat{U}_k^T \hat{U}_k &= (U_{k-1}^T - t_k p_k^T)(U_{k-1} - p_k t_k^T) = \\
&= I + (t_k p_k^T p_k t_k^T) - \overbrace{U_{k-1}^T p_k}^{=0} t_k^T - t_k \overbrace{p_k^T U_{k-1}}^{=0} = \\
&= I + (t_k p_k^T p_k t_k^T) = \\
&= I + v_k v_k^T
\end{aligned}$$

em que $v_k = t_k p_k^T$.

Se $t_k = n_k m_k$ é a decomposição QR econômica de t_k e $f_k d_k f_k^T$ é a decomposição espectral de $m_k (p_k^T p_k) m_k^T$, então

$$v_k v_k^T = n_k f_k d_k f_k^T n_k^T$$

Logo,

$$(\hat{U}_k^T \hat{U}_k^T)^{-\frac{1}{2}} = I - n_k f_k [I - (I + d_k)^{-\frac{1}{2}}] f_k^T n_k^T = I - n_k F_k n_k^T.$$

Assim,

$$\begin{aligned}
U_k &= \hat{U}_k(\hat{U}_k^T\hat{U}_k)^{-\frac{1}{2}} = [U_{k-1} - p_k t_k^T][I - n_k f_k n_k^T] = \\
&= [U_{k_1} - p_k m_k^T n_k^T][I - n_k F_k n_k^T] = \\
&= U_{k-1} - p_k m_k^T n_k^T - U_{k-1} n_k F_k n_k^T + p_k m_k^T n_k^T n_k F_k n_k^T = \\
&= U_{k-1} - p_k m_k^T n_k^T - U_{k-1} n_k F_k n_k^T + p_k m_k^T F_k n_k^T = \\
&= U_{k-1} - (p_k m_k^T + U_{k-1} n_k F_k - p_k m_k^T F_k) n_k^T = \\
&= U_{k-1} - [p_k m_k^T (I - F_k) + U_{k-1} n_k F_k] n_k^T = \\
&= U_{k-1} - E_k G_k^T
\end{aligned}$$

em que $E_k = [p_k m_k^T (I - F_k) + U_{k-1} n_k F_k]$ e $G_k = n_k$.

Algoritmo 4 OPAST

$U = \begin{pmatrix} I_{2M} \\ 0 \end{pmatrix}$
$Y = H(0)H(0)U$
$Z = (U^T Y)^{-1}$
$\hat{U} = YZ$
$U = \hat{U}(\hat{U}^T\hat{U})^{-\frac{1}{2}}$
for $k = 1, ..., N - 2L$ **do**
  $h_1 = x(k : k + L - 1)$
  $h_2 = x(k + L : k + 2L - 1)$
  $r = [h_1|h_2]$
  $s = U[h_1| - h_2]$
  $y = U^T r$
  $q = Zy$
  $t = Zs$
  $\gamma = (I_2 - s^T q)^{-1}$
  $p = (r - Uy)\gamma$
  $[G, m] = qr(t, 0)$
  $\hat{f} = mp^T pm^T$
  $[f, d] = eig(\hat{f})$
  $F = f(I - (d + I_2)^{-\frac{1}{2}})f^T$
  $E = -(pm^T(I_2 - F) + UGF)$
  $Z = Z + q\gamma t^T$
  $Y = Y - rs^T$
  $U = U + EG^T$
end for

2.4.6 Atualização da Matriz Espectral

Vimos métodos para atualizarmos a matriz U a cada passo, sem precisarmos aplicar a decomposição SVD, que é de alto custo computacional. Porém, interessa-nos encontrar os autovalores da matriz $(U_\downarrow)^\dagger U_\uparrow$. Ao invés de usarmos a pseudo-inversa, que também possui alta demanda computacional, vamos procurar um método de encontrarmos a matriz $\Phi = (U_\downarrow)^\dagger U_\uparrow$ e atualizá-la a cada passo. Em [4], é fornecida uma atualização para a matriz espectral, no caso em que existe uma atualização de posto 1 para U_k da forma $U_k = U_{k-1} + eg^T$. Adaptaremos essa atualização para o nosso caso, em que U_k possui atualização de posto 2.

Sabemos que, se A é uma matriz de posto completo, sua pseudo-inversa é $A^\dagger = (A^T A)^{-1} A^T$. Vamos supor que $U_\downarrow$ é de posto completo, então:

$$(U_\downarrow)^\dagger = (U_\downarrow^T U_\downarrow)^{-1} U_\downarrow^T$$

Portanto,

$$\begin{aligned} I = U^T U &= (U_\downarrow^T | \nu) \begin{pmatrix} U_\downarrow \\ \nu^T \end{pmatrix} = U_\downarrow^T U_\downarrow + \nu\nu^T \\ \Rightarrow U_\downarrow^T U_\downarrow &= I - \nu\nu^T \end{aligned}$$

Logo,

$$(U_\downarrow^\dagger) U_\uparrow = \underbrace{(U_\downarrow^T U_\downarrow)^{-1}}_{\Omega} \underbrace{U_\downarrow^T U_\uparrow}_{\Psi}$$

$$(I - \nu\nu^T)^{-1} = I + \frac{1}{1 - \|\nu\|^2} \nu\nu^T$$

$$\begin{aligned} \Phi = \Omega\Psi &= \Psi + \frac{1}{1 - \|\nu\|^2} \nu\nu^T \Psi \\ \Phi &= \Psi + \frac{1}{1 - \|\nu\|^2} \nu\varphi^T \end{aligned} \tag{2.21}$$

em que $\varphi = \Psi^T \nu$.

Encontramos, portanto, uma forma econômica de obtermos a matriz espectral. Precisamos agora de uma atualização simples a cada passo. Sabendo que $U_k = U_{k-1} + E_k G_k^T$, vamos definir:

$$\begin{aligned} e_k^- &= U_{\downarrow k-1}^T E_{\uparrow k} \\ e_k^+ &= U_{\uparrow k-1}^T E_{\downarrow k} \\ e_k^* &= e_k^+ + G_k (E_{\uparrow k}^T E_{\downarrow k}^T) \end{aligned}$$

Assim,

$$\begin{aligned}
\Psi_k &= U_{\downarrow k}^T U_{\uparrow k} = \\
&= (I_\downarrow(U_{k-1} + E_k G_k^T))^T (I_\uparrow(U_{k-1} + E_k G_k^T)) = \\
&= (U_{\downarrow k-1} + E_{\downarrow k} G_k^T)^T (U_{\uparrow k-1} + E_{\uparrow k} G_k^T) = \\
&= (U_{\downarrow k-1}^T + G_k E_{\downarrow k}^T)(U_{\uparrow k-1} + E_{\uparrow k} G_k^T) = \\
&= U_{\downarrow k-1}^T U_{\uparrow k-1} + G_k E_{\downarrow k}^T U_{\uparrow k-1} + U_{\downarrow k-1}^T E_{\uparrow k} G_k^T + G_k E_{\downarrow k}^T E_{\uparrow k} G_k^T = \\
&= \Psi_{k-1} + G_k e_k^{+T} + e_k^- G_k^T + G_k E_{\downarrow k}^T E_{\uparrow k} G_k^T = \\
&= \Psi_{k-1} + e_k^- G_k^T + G_k (e_k^{+T} + E_{\downarrow k}^T E_{\uparrow k} G_k^T) = \\
&= \Psi_{k-1} + e_k^- G_k^T + G_k e_k^{*T}
\end{aligned}$$

Também temos que ν é a última linha de U transposta. Portanto, podemos obter sua atualização por:

$$\begin{aligned}
\nu_k &= (e_L^T U_k)^T = \\
&= [e_L^T (U_k - 1 + E_k G_k^T)]^T = \\
&= (e_L^T U_k - 1 + e_L^T E_k G_k^T)^T = \\
&= (\nu_{k-1}^T + e_L^T E_k G_k^T)^T = \\
&= \nu_{k-1} + G_k \hat{e}_k
\end{aligned}$$

em que $\hat{e}_k$ é a última linha de E_k transposta, que contém 2 elementos. Vamos então retomar (2.21):

$$\Phi_k = \Psi_k + \frac{1}{1 - \|\nu_k\|^2} \nu_k \varphi_k^T =$$

$$= \Psi_{k-1} + e_k^- G_k^T + G_k e_k^{*T} + \frac{\nu_k}{1 - \|\nu_k\|^2} \varphi_k^T =$$

$$= \Psi_{k-1} + e_k^- G_k^T + G_k e_k^{*T} + \frac{\nu_{k-1}}{1 - \|\nu_k\|^2} \varphi_k^T + \frac{G_k \hat{e}_k}{1 - \|\nu_k\|^2} \varphi_k^T =$$

$$= \Psi_{k-1} + e_k^- G_k^T + G_k e_k^{*T} + \frac{\nu_{k-1}}{1 - \|\nu_k\|^2} \varphi_k^T + \frac{G_k \hat{e}_k}{1 - \|\nu_k\|^2} \varphi_k^T +$$

$$+ \frac{\nu_{k-1} \varphi_{k-1}^T}{1 - \|\nu_{k-1}\|^2} - \frac{\nu_{k-1} \varphi_{k-1}^T}{1 - \|\nu_{k-1}\|^2} =$$

$$= \underbrace{\Psi_{k-1} - \frac{\nu_{k-1} \varphi_{k-1}^T}{1 - \|\nu_{k-1}\|^2}}_{\Phi_{k-1}} + e_k^- G_k^T + G_k e_k^{*T} + \frac{G_k \hat{e}_k}{1 - \|\nu_k\|^2} \varphi_k^T +$$

$$+\nu_{k-1}\underbrace{[\frac{\varphi_k^T}{1-\|\nu_k\|^2}-\frac{\varphi_{k-1}^T}{1-\|\nu_{k-1}\|^2}]}_{\Delta\varphi_k^T}=$$

$$=\Phi_{k-1}+e_k^- G_k^T+G_k\underbrace{[e_k^{*T}+\frac{\hat{e}_k}{1-\|\nu_k\|^2}\varphi_k^T]}_{e_k'^T}+\nu_{k-1}\Delta\varphi_k^T=$$

Logo,

$$\Phi_k=\Phi_{k-1}+e_k^- G_k^T+G_k e_k'^T+\nu_{k-1}\Delta\varphi_k^T$$

E assim, atualiza-se a matriz espectral de forma econômica a cada passo. A seguir, apresentaremos o algoritmo para essa atualização. Esse algoritmo foi acoplado aos métodos NP3 e OPAST em nossas implementações.

Algoritmo 5 Atualização da Matriz Espectral

$\Psi_0 = U_{\downarrow 0}^T U_{\uparrow 0}$
$\nu_0 = U_0(end,:)^T$
$\varphi_0 = \Psi_0^T \nu_0$
$\Phi_0 = \Psi_0 + \frac{\nu_0}{1-\nu_0^T\nu_0}\varphi_0^T$
for $k = 1, ..., N-2L$ **do**
  $e_1 = U_\downarrow^T E_\uparrow$
  $e_2 = U_\uparrow^T E_\downarrow$
  $e_3 = e_2 + G E_\uparrow^T E_\downarrow$
  $\Psi_k = \Psi_{k-1} + e_1 G^T + G e_3^T$
  $\nu_k = \nu_{k-1} + G(E(end,:)^T)$
  $\varphi_k = \Psi_k^T \nu_k$
  $e_4 = e_3 + \varphi_k \frac{E(end,:)}{1-\nu_k^T\nu_k}$
  $\Delta\varphi = \frac{\varphi_k}{1-\nu_k^T\nu_k} - \frac{\varphi_{k-1}}{1-\nu_{k-1}^T\nu_{k-1}}$
  $\Phi_k = \Phi_{k-1} + G e_4^T + e_1 G^T + \nu_{k-1}\Delta\varphi$
end for

2.5 Testes e Resultados

Serão agora apresentados os resultados provenientes de testes realizados com os 4 algoritmos adaptativos de alta resolução estudados até aqui: Método de Iteração Ortogonal Adaptativo, Strobach, NP3 e OPAST. As implementações

foram feitas em MATLAB e os testes em um computador com dois processadores de 2GHz e 2Gb de memória RAM. Foi processado um arquivo contendo o som da nota A4 (freqüência fundamental: 440 Hz) de um piano, com 24646 pontos à freqüência de amostragem 22050 Hz, resultando em uma duração de 1,118 segundos. Nos algoritmos adaptativos, trabalhamos inicialmente com uma janela de tamanho $L = 300$ e $2M$ autovalores a serem detectados, $2M$ variando entre 50 e 250. Posteriormente utilizou-se $L = 500$ e $2M$ entre 50 e 350. Também testou-se a detecção do valor de $2M$ ideal e obteve-se 154 para $L = 300$ e 284 para $L = 500$.

Os parâmatros utilizados para a avaliação dos algoritmos foram o tempo de processamento e o erro relativo na norma infinito, ou seja, dado x o sinal original e $\hat{x}$ o sinal ressintetizado, o erro relativo é $E = \dfrac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} = \dfrac{\max_k(|x_k - \hat{x}_k|)}{\max_j |x_j|}$. Chamaremos $\max_k(|x_k - \hat{x}_k|)$ de erro absoluto.

O primeiro algoritmo testado foi o método de iteração ortogonal adaptativo. Neste método, a única aproximação que temos é a de $U(t)$ por uma matriz $Q(t)$ obtida por iterações ortogonais. Porém, seu tempo de processamento é muito alto. Foi utilizada uma rotina que otimiza a multiplicação matriz-vetor, no caso em que a matriz é de Hankel, o que reduziu o tempo de processamento, mas que ainda foi muito superior aos outros algoritmos.

Os resultados dos testes com esse algoritmo são apresentados em duas tabelas: a Tabela 2.1 mostra o erro e o tempo de processamento para uma para janela de tamanho $L = 300$; a Tabela 2.2 mostra os resultados para $L = 500$. Nota-se em alguns casos que o erro aumentou ao tomar-se um L maior. Isso ocorre pois os erros numéricos aumentam proporcionalmente à ordem da matriz.

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.10648	0.11358	8min 45.360s
100	0.023262	0.024813	29min 39.328s
150	0.0066893	0.0071353	1h 04min 11.156s
154	0.0067844	0.0072367	1h 05min 16.672s
200	0.0053554	0.0057125	1h 59min 38.875s
250	0.0017188	0.0018333	3h 24min 33.156s

Tabela 2.1: Método de Iteração Ortogonal Adaptativo, com $L = 300$.

O algoritmo de Strobach manteve os erros muito baixos e também obteve uma melhora no tempo de processamento em relação ao método de iteração ortogonal. Porém, esse algoritmo ainda requer a fatoração QR a cada passo, o que faz com que sua velocidade ainda não seja satisfatória. Vê-se na Tabela

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.12102	0.12908	13min 53.718s
150	0.017534	0.018702	1h 23min 42.641s
250	0.010714	0.011428	4h 11min 33.282s
284	0.0112542	0.0120045	6h 01min 36.828s
350	0.0036743	0.0039193	10h 01min 59.578s

Tabela 2.2: Método de Iteração Ortogonal Adaptativo, com $L = 500$.

2.3 que, para $L = 300$, os erros relativos foram muito baixos. Porém, para $L = 500$ e $2M > 50$ o problema tornou-se mal-condicionado em um intervalo, elevando os erros além do esperado.

Na Figura 2.1, vemos o intervalo no qual o problema tornou-se mal condicionado, para $L = 500$, e ocorreram os maiores erros. Percebe-se que há uma perturbação no comportamento do sinal, no sentido que o sinal não é muito periódico naquele intervalo, o que acarreta esse problema. Porém, na Figura 2.2, vemos que, do ponto 900 em diante, os erros relativos foram ínfimos, da ordem de 10^{-5} para $2M = 150$ e da ordem de 10^{-6} para $2M = 250$ e 350.

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.0071264	0.0076015	5min 22.078s
100	0.0063285	0.0067504	23min 32.187s
150	0.0053932	0.0057528	57min 2.329s
154	0.0061024	0.0065092	1h 00min 21.750s
200	0.0048342	0.0051564	1h 52min 48.437s
250	0.0043716	0.004663	3h 23min 36.110s

Tabela 2.3: Algoritmo de Strobach, com $L = 300$.

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.0098189	0.010473	8min 14.766s
150	0.9439	1.0068	1h 15min 7.750s
250	1.0586	1.1292	4h 06min 36.563s
284	0.3173246	0.338479	5h 39min 39.625s
350	1.0027	1.0696	10h 12min 1.843s

Tabela 2.4: Algoritmo de Strobach, com $L = 500$.

O algoritmo NP3 apresentou um ganho considerável de velocidade em relação ao método de Strobach e ao método de iteração ortogonal. Houve um aumento no erro devido à aproximação $U_k \approx U_{k-1}$. Porém, esse erro está

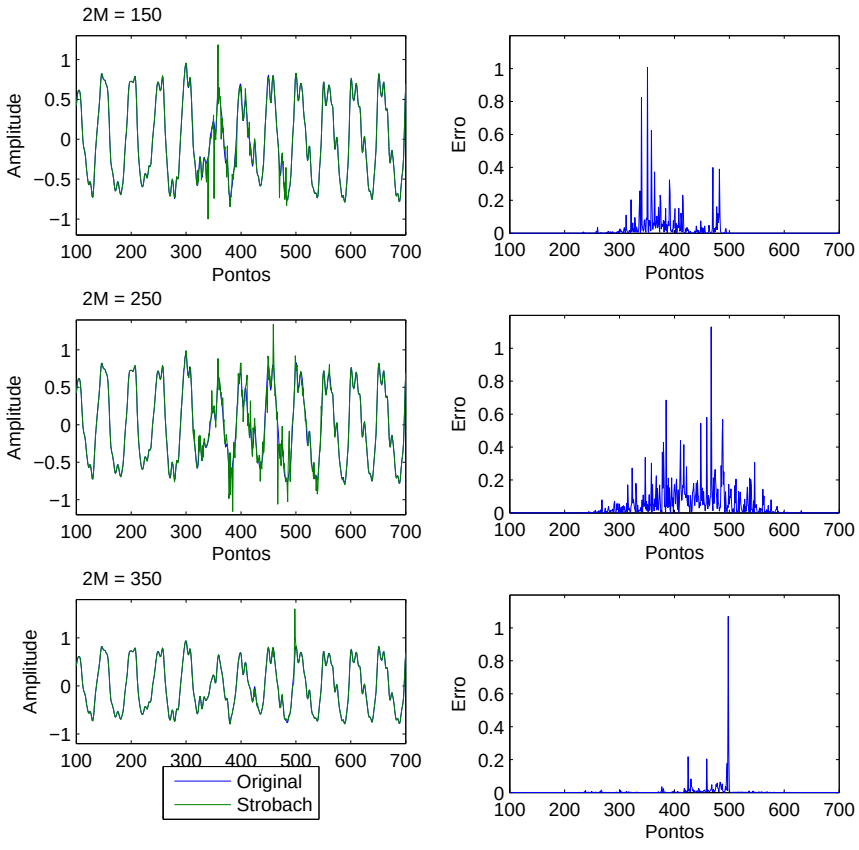

Figura 2.1: Erro no método de Strobach para $L = 500$, entre os pontos 100 e 700.

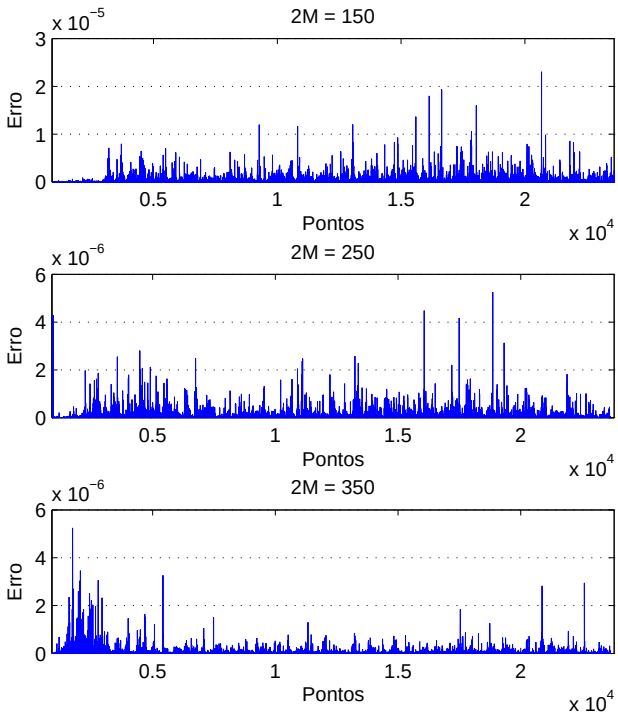

Figura 2.2: Erro no método de Strobach para $L = 500$, a partir do ponto 900.

dentro de uma margem imperceptível ao ouvido, quando se compara o som original ao ressintetizado. Nota-se, também, que o erro do algoritmo NP3 diminui até um certo valor de $2M$, a partir do qual aumenta novamente. Isso acontece porque o erro que tomamos é o maior erro entre o sinal sintetizado e o original. Nesse caso, tivemos um erro dessa magnitude em alguns pontos de uma região no início do sinal, próxima à mesma região em que o método de Strobach também apresentou problemas.

A Figura 2.3 ilustra o erro do algoritmo NP3 para $L = 300$ e $2M = 250$, dividindo-o em duas regiões: os primeiros 200 pontos e o restante. Vemos que na primeira região está o erro máximo, junto com erros de mesma magnitude em alguns pontos. Já na segunda região, o erro foi inferior a 0.003.

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.14402	0.15362	2min 50.219s
100	0.025428	0.027123	12min 50.219s
150	0.007384	0.0078763	35min 42.765s
154	0.0064135	0.0068411	37min 47.640s
200	0.0042121	0.0044929	1h 20min 38.922s
250	0.067566	0.07207	2h 11min 42.547s

Tabela 2.5: Algoritmo NP3, com $L = 300$.

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.1814	0.19349	3min 15.703s
150	0.039712	0.042359	36min 42.562s
250	0.0094178	0.010046	2h 32min 21.000s
284	0.054354	0.057978	3h 54min 21,125s
350	0.31675	0.33787	6h 46min 57.609s

Tabela 2.6: Algoritmo NP3, com $L = 500$.

O algoritmo OPAST teve o melhor desempenho no quesito tempo, porém teve o maior erro relativo dos métodos de alta resolução. É importante mencionar que, mesmo que os erros pareçam indicar maus resultados, muitas vezes eles correspondem apenas a uma leve diferença de volume entre o som original e o ressintetizado, acarretando pouca diferença perceptível em sua comparação auditiva.

A Figura 2.4 mostra o gráfico do erro para o algoritmo OPAST, quando $L = 300$ e $2M = 150$. No gráfico superior, apresenta-se a diferença entre o som original e o sintetizado. No gráfico inferior, está o erro relativo em valor absoluto. Percebe-se, novamente, que os maiores erros encontram-se

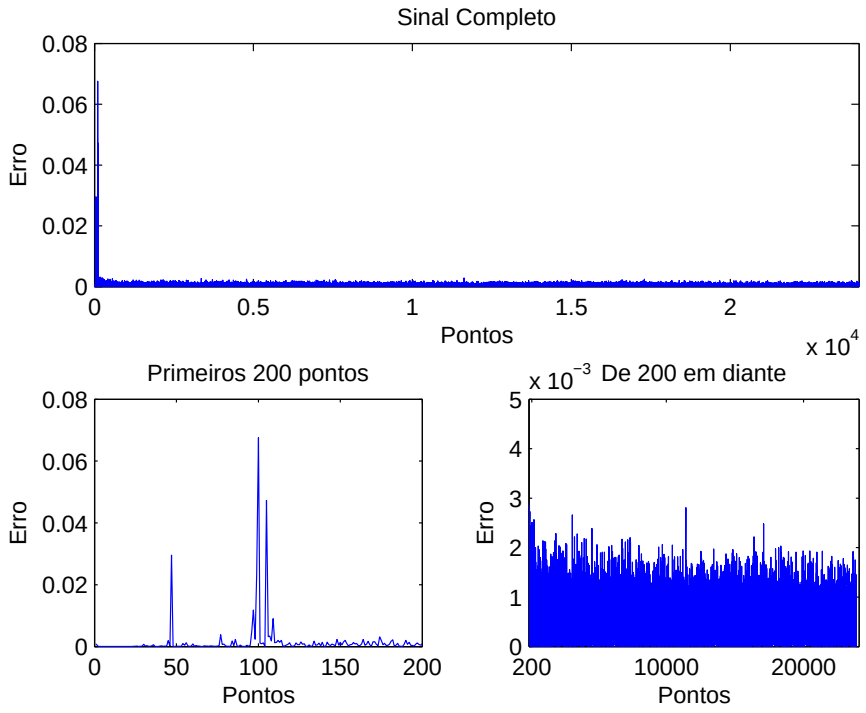

Figura 2.3: Erro relativo no algoritmo NP3 para $L = 300$ e $2M = 250$.

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.40039	0.42708	2min 30.375s
100	0.40852	0.43575	12min 0.579s
150	0.34052	0.36322	35min 22.922s
154	0.35717	0.38098	37min 10.281s
200	0.16994	0.18127	1h 16min 22.828s
250	0.09364	0.099882	2h 28min 33.265s

Tabela 2.7: Algoritmo OPAST, com $L = 300$.

2M	Erro Absoluto	Erro Relativo	Tempo de Processamento
50	0.51008	0.54408	2min 55.328s
150	0.52621	0.56129	34min 23.125s
250	0.29478	0.31443	2h 30min 29.547s
284	0.220549	0.235252	3h 41min 15.062s
350	0.091043	0.097113	7h 20min 39.000s

Tabela 2.8: Algoritmo OPAST, com $L = 500$.

no começo do sinal, ou seja, nos primeiros passos do algoritmo, nos quais a convergência ainda é frágil.

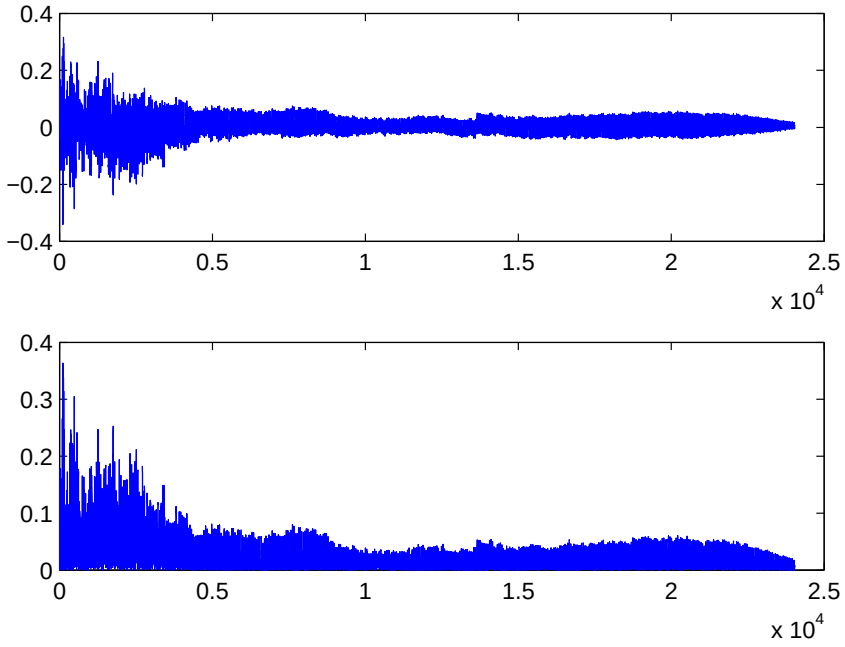

Figura 2.4: Erro relativo do algoritmo OPAST para $L = 300$ e $2M = 150$.

A Figura 2.5 apresenta os primeiros 300 pontos do sinal original e dos sinais ressintetizados pelos algoritmos testados, com $L = 300$ e $2M = 50$. Nota-se, mais uma vez, que no trecho inicial os algoritmos ainda não estão próximos da convergência. Na Figura 2.6, tomamos um trecho do meio do sinal, em que os algoritmos já possuem uma melhor estabilidade. Porém, pelo fato de $2M$ ser pequeno, ainda há uma diferença visível entre o sinal original e os sinais ressintetizados. Na Figura 2.7, temos o mesmo trecho com $2M = 250$. Nesse caso, nota-se que os resultados dos métodos implementados aproximam-se mais do sinal original.

Por fim, os gráficos visualizados nas Figuras 2.8, 2.9, 2.10 e 2.11, descrevem a comparação entre os métodos testados, com relação aos seus erros relativos e tempos de processamento.

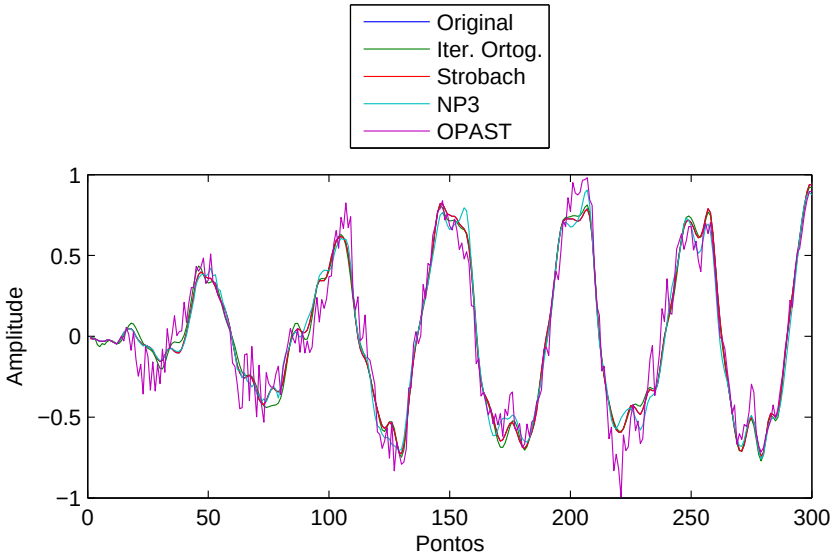

Figura 2.5: Trecho inicial do sinal original e sinais ressintetizados pelos métodos testados, com $L = 300$ e $2M = 50$.

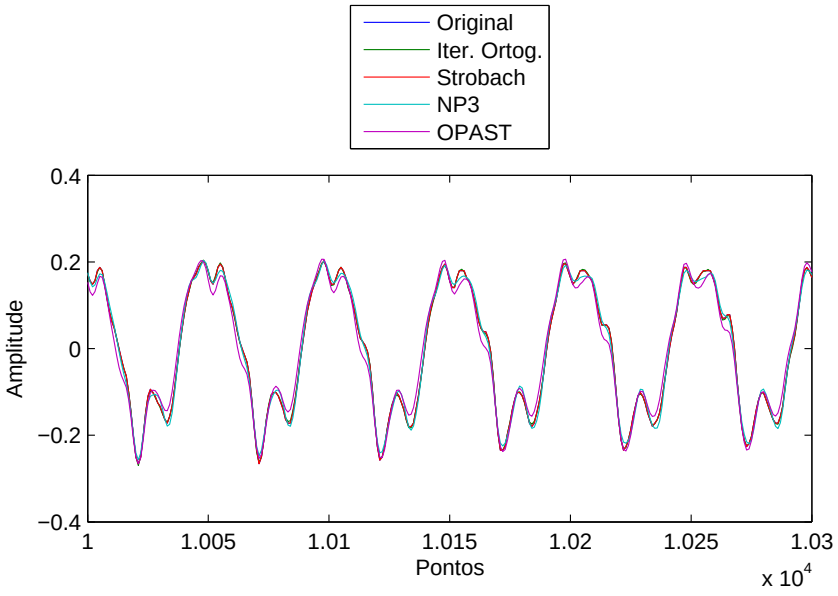

Figura 2.6: Trecho intermediário do sinal original e sinais ressintetizados pelos métodos testados, com $L = 300$ e $2M = 50$.

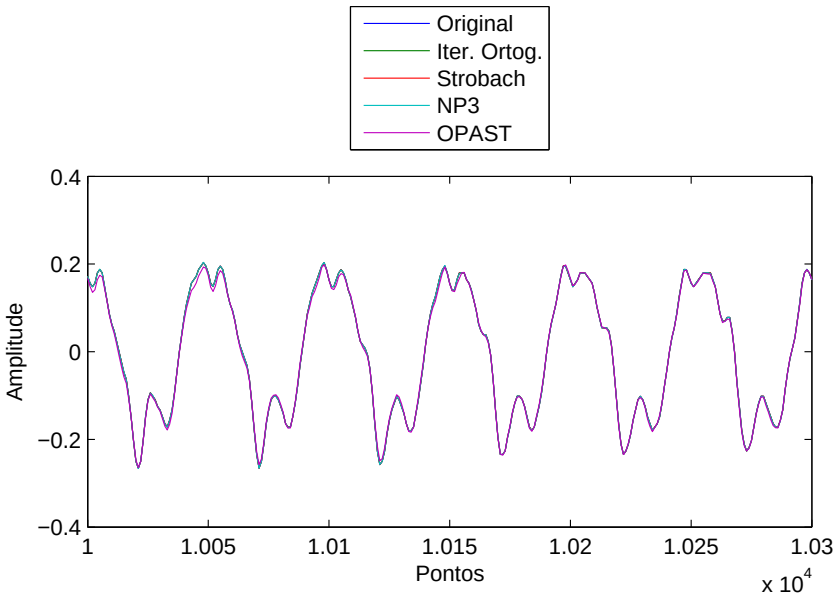

Figura 2.7: Trecho intermediário do sinal original e sinais ressintetizados pelos métodos testados, com $L = 300$ e $2M = 250$.

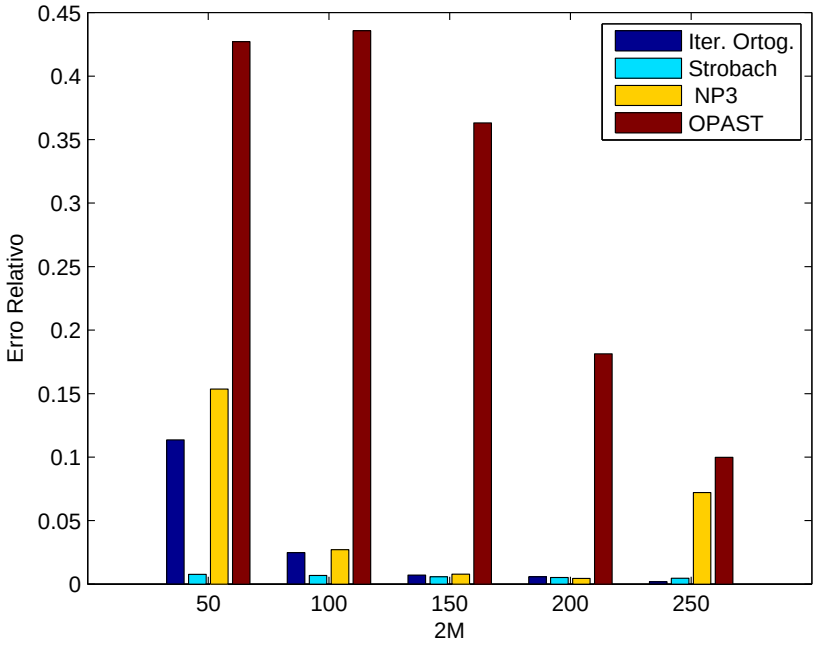

Figura 2.8: Erro Relativo para $L = 300$.

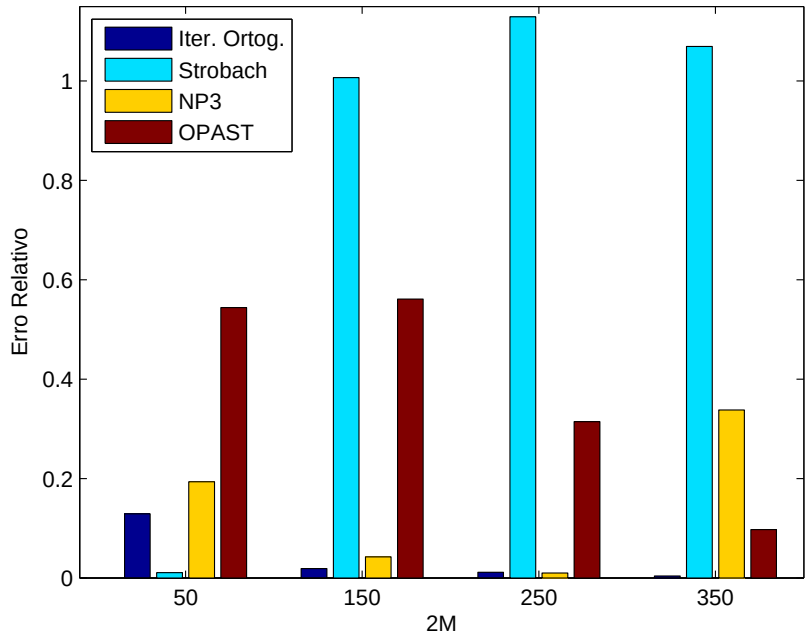

Figura 2.9: Erro Relativo para $L = 500$.

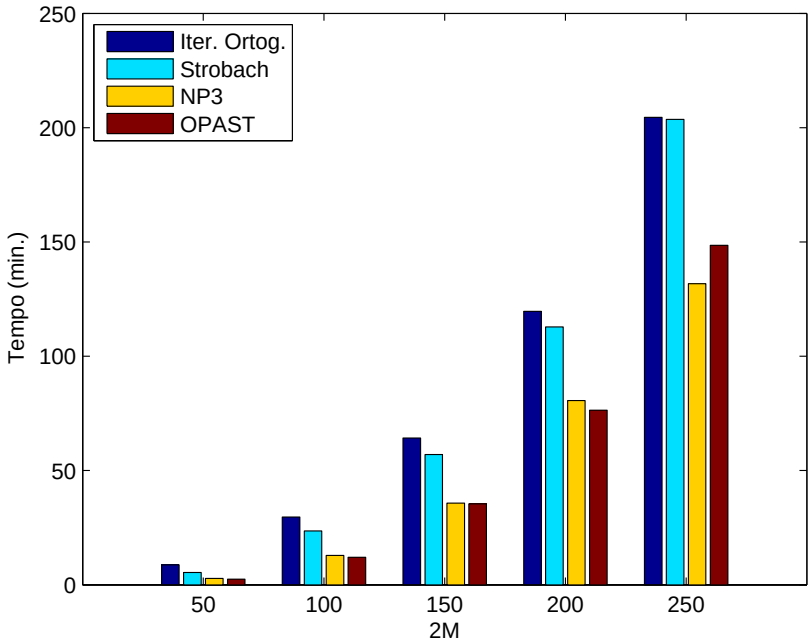

Figura 2.10: Tempo de Processamento em Minutos para $L = 300$.

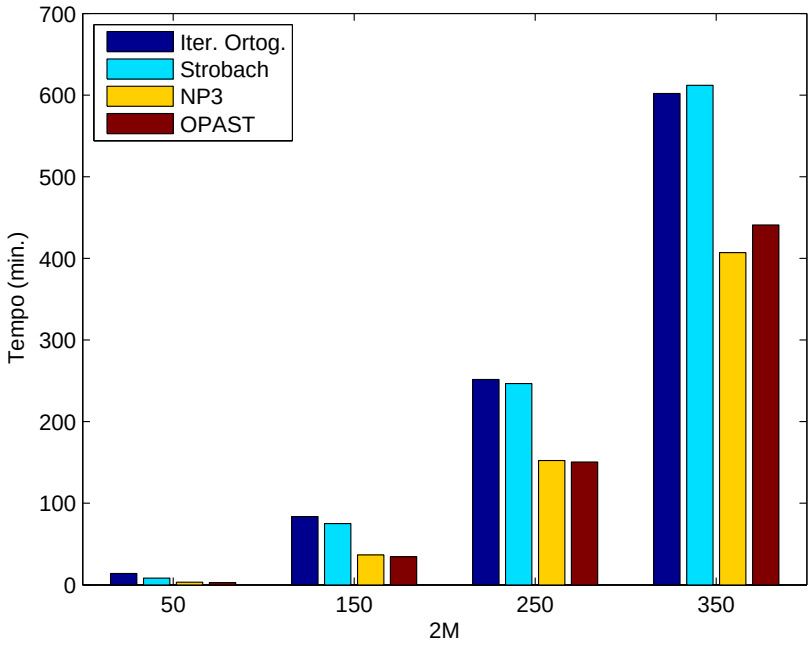

Figura 2.11: Tempo de Processamento em Minutos para $L = 500$.

Capítulo 3

Algoritmos no Domínio das Freqüências

Neste capítulo, estudaremos os métodos no domínio das freqüências para análise e ressíntese de sinais musicais. Estes métodos buscam uma representação do sinal baseada nas freqüências que o compõem e, a partir disso, procuram os parâmetros que descrevem o sinal. A ferramenta que utilizaremos para obter essa representação é a Transformada de Fourier. Assim, iniciaremos esse capítulo com um estudo sobre a Teoria de Fourier.

3.1 Teoria de Fourier

3.1.1 Introdução

A Teoria de Fourier é um assunto muito importante em processamento de sinais. Entre os temas que esta teoria aborda estão a transformada de Fourier, através da qual é possível avaliar o espectro de freqüências de uma função no tempo, e a aproximação de funções via polinômios trigonométricos e séries de Fourier.

Nesta seção veremos um pouco de cada um desses temas, em suas formas contínua e discreta, de forma que nosso principal objetivo será deduzir e introduzir a Transformada Discreta de Fourier, ou DFT (do inglês, Discrete Fourier Transform), que será a principal ferramenta utilizada nos modelos espectrais de análise e síntese de sinais sonoros aqui apresentados.

3.1.2 Aproximação por polinômios trigonométricos e coeficientes de Fourier

Seja E um espaço vetorial, real ou complexo, de dimensão finita n com produto interno $<,>$. Seja S um subespaço de E, de dimensão $k \leq n$. Considere $e_1, ..., e_k$ uma base ortogonal de S. Sabemos que, para todo vetor $x \in S$,

$$x = \frac{< x, e_1 >}{\|e_1\|^2} e_1 + ... + \frac{< x, e_k >}{\|e_k\|^2} e_k \tag{3.1}$$

Seja $v \in E$. A projeção ortogonal de v em S é dada por:

$$\bar{v} = \frac{< v, e_1 >}{\|e_1\|^2} e_1 + ... + \frac{< v, e_k >}{\|e_k\|^2} e_k, \tag{3.2}$$

que é o vetor de S mais próximo de v em relação à norma definida pelo produto interno.

Se E é um espaço de dimensão infinita e S é um subespaço de E, também de dimensão infinita, estender (3.1) a um somatório infinito da forma

$$\sum_{k=1}^{\infty} \frac{< x, e_k >}{\|e_k\|^2} e_k,$$

sendo $e_1, e_2, ...$ um conjunto ortogonal LI infinito, é temerário, pois não há como saber de antemão se a série irá convergir. Para representar uma série formal, da qual não temos garantia de convergência, utilizaremos a seguinte notação:

$$x \sim \sum_{k=1}^{\infty} \frac{< x, e_k >}{\|e_k\|^2} e_k$$

Posteriormente, uma vez provado que a série converge em média para x (ver definição a seguir), substitui-se o símbolo $\sim$ por uma igualdade. De qualquer forma, $\frac{<x,e_k>}{\|e_k\|^2}$ são chamados coeficientes de Fourier generalizados de x em relação aos vetores ortogonais $e_1, e_2, ...$.

Definição 11. *Seja E um espaço vetorial real de dimensão infinita com produto interno. Uma série infinita $\sum_{k=1}^{\infty} x_k$ em que, para todo k, $x_k \in E$, converge* **em média** *para um vetor $x \in E$ se, dado $\epsilon > 0$, existe um inteiro K tal que, para todo $n > K$,*

$$\left\| \sum_{k=1}^{n} x_k - x \right\| < \epsilon \tag{3.3}$$

Definição 12. *Sejam $e_1, e_2, ...$ vetores ortogonais LI pertencentes a um espaço E de dimensão infinita. Diz-se que esse conjunto é uma e-base de E se, para todo $x \in E$, existem únicos $\alpha_1, \alpha_2, ...$ tais que a seguinte série converge em média para x:*

$$x = \sum_{k=1}^{\infty} \alpha_k e_k$$

Vamos agora definir um espaço de dimensão infinita com produto interno de nosso interesse. Dado um intervalo $[a, b]$, considere $\mathcal{C}[a, b]$, o espaço vetorial das funções contínuas em $[a, b]$, com as operações usuais de soma e produto por escalar, e produto interno definido por:

$$< g, h >= \int_a^b g(t)h(t)dt \tag{3.4}$$

Desejamos estender esse produto interno a outro espaço de maior relevância ao nosso estudo. Para definir esse espaço, vamos primeiro definir funções contínuas por partes.

Definição 13. *Uma função g é dita ser contínua por partes em um intervalo $[a, b]$, se ela possui apenas um número finito de pontos de descontinuidade t_k em $[a, b]$ e em cada um desses pontos t_k, g possui limites laterais $g(t)^+$ e $g(t)^-$.*

É importante notar que a função g não precisa estar definida nos pontos t_k e os limites laterais não precisam ser iguais, contanto que ambos existam. Na prática, uma função contínua por partes em um intervalo é uma função que pode ser particionada em subintervalos de forma que seja contínua no interior de cada um desses subintervalos abertos.

Podemos denominar $\mathcal{CP}[a, b]$ o conjunto das funções contínuas por partes, do qual $\mathcal{C}[a, b]$ é um subconjunto. É válido que, se $g, h \in \mathcal{CP}[a, b]$ e $\alpha \in \mathbb{R}$, então $g + h$ e $\alpha g \in \mathcal{CP}[a, b]$. Portanto, $\mathcal{CP}[a, b]$ é um espaço vetorial. Porém, (3.4) não é um produto interno em $\mathcal{CP}[a, b]$. Isso porque, pela definição de produto interno, $< g, g >= 0 \Rightarrow g \equiv 0$. No entanto, dada a função h tal que $h(t) = 0$ se $t \in [a, b)$ e $h(b) = 1$, temos que $< h, h >= 0$. Para contornar esse problema, vamos definir uma relação de equivalência: diremos que uma função em $\mathcal{CP}[a, b]$ é equivalente a outra se elas diferirem apenas em um conjunto finito de pontos. Dessa forma, uma função não nula em um número finito de pontos pertence à mesma classe de equivalência da função nula e, portanto, (3.4) torna-se um produto interno no conjunto $\overline{\mathcal{CP}}[a, b]$ das classes de equivalência daquela relação definida em $\mathcal{CP}[a, b]$.

Vamos agora definir um novo subespaço, o dos polinômios trigonométricos.

Definição 14. *Um polinômio trigonométrico de grau m é uma expressão da forma:*

$$P_m(t) = \frac{a_0}{2} + \sum_{k=1}^{m-1}[a_k \cos(kt) + b_k \operatorname{sen}(kt)] + a_m \cos(mt)$$

Denominaremos $\mathcal{PT}[a,b]$ o conjunto dos polinômios trigonométricos. O conjunto das classes de equivalência de $\overline{\mathcal{CP}}[a,b]$ representadas pelos polinômios trigonométricos forma um supespaço de $\overline{\mathcal{CP}}[a,b]$.

Temporariamente, substituiremos o intervalo genérico $[a,b]$, pelo intervalo específico $[-\pi,\pi]$.

Proposição 3.1. *O conjunto* $\{1, \cos t,\ \operatorname{sen} t, ..., \cos(kt),\ \operatorname{sen}(kt), ...\}$ *é ortogonal em* $\overline{\mathcal{CP}}[-\pi,\pi]$.

Portanto, o conjunto acima também é uma base de $\mathcal{PT}[-\pi,\pi]$. Assim, usando (3.2), podemos projetar uma função $g \in \overline{\mathcal{CP}}[-\pi,\pi]$ em $\mathcal{PT}[-\pi,\pi]$. Obteremos, então, a seguinte série formal:

$$G(t) \sim \frac{a_0}{2} + \sum_{k=1}^{\infty}[a_k \cos(kt) + b_k \operatorname{sen}(kt)]$$

em que a_k e b_k são obtidos por:

$$\begin{aligned} a_k &= \frac{1}{\pi}\int_{-\pi}^{\pi} g(t)\cos(kt)dt \\ b_k &= \frac{1}{\pi}\int_{-\pi}^{\pi} g(t)\operatorname{sen}(kt)dt \end{aligned}$$

Ainda não podemos garantir a convergência dessa série, mas vimos que ela surge como projeção ortogonal no espaço dos polinômios trigonométricos. Veremos que a convergência dessa série ocorrerá para funções com algumas características que apresentaremos adiante. Mas antes de enunciarmos o teorema que garante esse resultado, definiremos a série de Fourier.

3.1.3 Série de Fourier para funções períodicas

Definição 15. *Uma função g é dita ser periódica de período P se $(\forall t \in \mathbb{R})$ e $(\forall k \in \mathbb{Z})$, $g(t+Pk) = g(t)$.*

Definição 16. *Seja g períodica de período P. A série de Fourier de g é definida formalmente por:*

$$g(t) \sim \sum_{k=-\infty}^{\infty} c_k e^{\frac{i2\pi kt}{P}}$$

em que os coeficientes c_k são expressos por:

$$c_k = \frac{1}{P}\int_0^P g(t)e^{-\frac{i2\pi kt}{P}}dx$$

Se g é uma função real, usando as relações $\cos(\phi) = \frac{e^{i\phi}+e^{-i\phi}}{2}$ e $\operatorname{sen}(\phi) = \frac{e^{i\phi}-e^{-i\phi}}{2i}$, temos:

$$g(t) \sim \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos\frac{2\pi kt}{P} + \sum_{k=1}^{\infty} b_k \operatorname{sen}\frac{2\pi kt}{P} \tag{3.5}$$

em que os coeficientes a_k e b_k são obtidos por:

$$\begin{aligned} a_k &= \frac{2}{P}\int_0^P g(t)\cos\frac{2\pi kt}{P}dt \\ b_k &= \frac{2}{P}\int_0^P g(t)\operatorname{sen}\frac{2\pi kt}{P}dt \end{aligned} \tag{3.6}$$

Dada uma prova da convergência de (3.5), teremos que o sinal poderá ser aproximado via senos e cossenos de diferentes freqüências, ponderados pelos coeficientes dados por (3.6). Poderemos, então, interpretar esses coeficientes como os pesos das freqüências que compõem o sinal e, portanto, sua obtenção por (3.6) poderá ser considerada uma etapa de análise do sinal. Por outro lado, se já tivermos os pesos das freqüências que descrevem o sinal, poderemos aproximá-lo por (3.5) e, assim, estaríamos realizando uma síntese, um processo inverso ao anterior.

Até aqui, entretanto, temos uma definição da série de Fourier para uma função. Porém, não temos nenhuma garantia de que essa série aproxime a função, ou mesmo que convirja. Para funções com uma determinada característica, enunciaremos um teorema que resolverá essa questão.

Definição 17. *Uma função g é dita ser suave por partes em um intervalo I se tanto g quanto g' são contínuas por partes neste intervalo. Para g definida em toda reta real, g é suave por partes em toda a reta se for suave por partes em cada intervalo limitado.*

Teorema 3.1. *Seja g suave por partes e períodica de período P. Nesse caso, sua série de Fourier converge pontualmente para:*

$$\frac{g(t)^+ + g(t)^-}{2}$$

A demonstração desse teorema não será explicitada aqui, mas pode ser facilmente encontrada na literatura sobre o assunto ([7], [14]). Esse é um teorema de fundamental importância, pois nos fornece um resultado que garante a aproximação de funções períodicas via séries trigonométricas. Assim, se possuíssemos um sinal perfeitamente período, contínuo e sem ruídos, poderíamos aproximá-lo com o erro que desejássemos, apenas truncando sua série de Fourier.

Na prática, os sinais obtidos raramente são perfeitamente periódicos, pois estão suscetíveis à interferência de vários fatores como ruídos, falta de precisão nos instrumentos de medição, discretização do sinal e, principalmente, ao fato de que as próprias fontes sonoras naturais não geram sinais perfeitamente períodicos, havendo oscilações em suas freqüências, mesmo que quase imperceptíveis. Ainda podemos citar sons percussivos e alguns tipos de ruídos nos quais pode haver algum interesse de estudo e, nesse caso, o sinal obtido pode ser completamente não-períodico.

Para funções que estejam definidas apenas em um intervalo finito, podemos definir sua extensão períodica, ou seja, podemos estendê-la a toda reta real de forma a obtermos uma função períodica, repetindo seu comportamento no intervalo que conhecemos.

Definição 18. *Seja g uma função definida no intervalo $[a, b]$. Sua extensão períodica de período $P = (b - a)$ é definida por:*

$$h(t + (b - a)k) = g(t)$$

Para $t \in [a, b)$ e $k \in \mathbb{Z}$.

Assim, se g é uma função suave por partes, a série de Fourier para a função h converge para $\frac{g(t)^+ + g(t)^-}{2}$ em intervalos da forma $(a + (b-a)k, b + (b-a)k)$ e para $\frac{g(a)^+ + g(b)^-}{2}$ nos pontos da forma $a + (b - a)k$, para $k \in \mathbb{Z}$.

Para garantirmos que um sinal sonoro possa ser aproximado pontualmente por sua série de Fourier, precisamos que esse sinal satisfaça as condições de ser suave por partes e periódico. Quanto à questão da periodicidade, não há garantias de que o som seja perfeitamente periódico, mas podemos tomar trechos do sinal (ou o sinal todo) e utilizarmos sua extensão periódica para uma análise.

Na prática, ao processarmos computacionalmente um sinal sonoro, não estamos trabalhando com um sinal contínuo, e sim, um sinal discreto, composto de amostras igualmente espaçadas, em geral. Portanto, ao trabalharmos com a análise contínua, podemos considerar uma interpolação suave desses pontos de amostragem, de forma a termos uma função suave por partes, satisfazendo as condições suficientes para a convergência de sua série de Fourier.

3.1.4 Transformada Contínua de Fourier

Agora que vimos que um sinal sonoro pode ser aproximado por meio de uma série trigonométrica, podemos relacionar esse tipo de aproximação com sua transformada de Fourier.

Definição 19. *Seja g definida em $(-\infty, \infty)$, tal que $\int_{-\infty}^{\infty} |g(t)|dt < \infty$. Sua Transformada de Fourier $\mathcal{F}(w)$ é dada por:*

$$\mathcal{F}(w) = \int_{-\infty}^{\infty} g(t)\, e^{-i2\pi wt} dt$$

3.1.5 DFT

Queremos agora aplicar a Transformada de Fourier a uma função definida apenas em um conjunto finito de pontos igualmente espaçados, como é o caso dos sinais musicais. Seja g uma função definida em toda a reta real, que seja nula fora do intervalo $[0, P]$ e que, para todo w, satisfaça:

$$\int_{-\infty}^{\infty} g(t)\, e^{-i2\pi wt} dt = \int_{0}^{P} g(t)\, e^{-i2\pi wt} dt < \infty \tag{3.7}$$

Logo, $\mathcal{F}(w) = \int_0^P g(t)\, e^{-i2\pi wt} dt$. Essa função g pode representar, por exemplo, um sinal de duração finita P, começando no instante $t = 0$, o qual será digitalizado e transformado em pontos discretos.

Considere agora uma partição do intervalo $[0, P]$ dada por $\{t_0, ..., t_N\}$, em que $t_n = n\Delta t$ e $\Delta t = \frac{P}{N}$. Definindo $g_w(t) = f(t)\, e^{-i2\pi wt}$, vamos aproximar a integral (3.7) por:

$$\mathcal{F}(w) = \int_0^P g_w(t)dt \approx \Delta t \sum_{n=0}^{N-1} g_w(t_n) = \frac{P}{N} \sum_{n=0}^{N-1} g(t_n)\, e^{-i2\pi wt_n} \tag{3.8}$$

Assim, obtemos uma função que pode ser calculada para qualquer freqüência w. Porém, como estamos trabalhando com pontos discretos, a função da freqüência deve ser discreta também, para podermos utilizá-la computacionalmente. Portanto, devemos escolher quantos e em quais pontos w realizaremos esse cálculo. Como são usados N pontos no domínio do tempo em (3.8) e queremos estabelecer uma relação única e inversível entre o domínio do tempo e o das freqüências, é natural que obtenhamos N pontos também no domínio das freqüências, da forma:

$$w_k = k\Delta w \tag{3.9}$$

para $k = 0, ..., N-1$.

Precisamos encontrar Δw para sabermos quem serão os pontos w_k. Sabemos que $\Delta w = w_1$, ou seja, Δw é a menor freqüência não nula representável para o sinal. A menor freqüência que pode ser encontrada no sinal corresponde ao maior período representado, nesse caso P. Assim,

$$\Delta w = \frac{1}{P} \tag{3.10}$$

Mais adiante, discutiremos mais sobre freqüências máxima e mínima obtidas pela transformada de Fourier. Os pontos que encontramos para o domínio das freqüências localizam-se no intervalo $[0, \Omega]$, com $\Omega = N\Delta w = \frac{N}{P}$. Também temos que $N\Delta t = P = \frac{1}{\Delta w}$. Disso, surgem as relações de reciprocidade:

$$\begin{aligned} P\Omega &= N \\ \Delta t \Delta w &= \frac{1}{N} \end{aligned} \tag{3.11}$$

Assim, usando (3.11), temos que:

$$w_k t_n = kn\Delta w \Delta t = \frac{kn}{N}$$

Portanto,

$$\mathcal{F}(w_k) \approx P\frac{1}{N}\sum_{n=0}^{N-1} g(t_n)\, e^{-\frac{i2\pi kn}{N}}$$

Baseado nesse resultado, podemos definir a Transformada Discreta de Fourier da seguinte forma:

Definição 20 (DFT)**.** *Seja um conjunto de pontos $x_0, x_1, ..., x_{N-1}$. A Transformada Discreta de Fourier para esses pontos é:*

$$X_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n\, e^{-\frac{i2\pi kn}{N}}$$

Para $k = 0, ..., N-1$.

Assim, para o caso anterior, $\mathcal{F}(w_k) \approx PX_k$. Porém, para um conjunto de pontos do qual não se sabe a procedência, essa definição é razoável. Analogamente, define-se a inversa da DFT, ou IDFT, como:

$$x_n = \sum_{k=0}^{N-1} X_k\, e^{\frac{i2\pi kn}{N}}$$

Assim como tomamos a extensão periódica de funções para aplicar a transformada contínua, faremos a extensão periódica para a transformada discreta: $(\forall m \in \mathbb{Z})\ x_{n+mN} = x_n$.

A definição da DFT e da IDFT varia de autor para autor. Alguns autores colocam o coeficiente $\frac{1}{N}$ na IDFT ao invés da DFT. Outros preferem utilizar a DFT normalizada, colocando tanto na DFT quanto na IDFT o coeficiente $\frac{1}{\sqrt{N}}$. Outra diferença comum entre definições é o índice do somatório, ou seja, a indexação dos pontos. Alguns autores preferem utilizar n e k no intervalo que vai de $-\frac{N}{2}$ a $\frac{N}{2}$. Neste trabalho, vamos usar a notação $\omega_N^{kn} = e^{-i\frac{2\pi kn}{N}}$. Note que ω_N^j é a j-ésima raiz complexa da unidade e também que $\omega_N^{j+mN} = \omega_N^j$. Assim,

$$x_{n+mN}\,\omega_N^{(k+mN)(n+mN)} = x_n\omega_N^{kn+(km+mn+m^2N)N} = x_n\omega_N^{kn}.$$

Portanto, não importa se começamos com índice 0 ou $-\frac{N}{2}$, contanto que x_0 seja sempre o mesmo. Adiante, ao apresentarmos operadores discretos, utilizaremos as duas indexações para que fiquem claros os efeitos das operações que aplicaremos sobre sinais discretos.

Outra forma de expressar a DFT é matricialmente: $\mathbf{X} = \mathbf{W}\mathbf{x}$ em que $\mathbf{X}$ é o vetor dos pontos no domínio das freqüências, também conhecido como **espectro**; $\mathbf{x}$ é o vetor dos pontos no domínio do tempo, ou seja, o próprio sinal; e W é a matriz de Fourier de ordem N, dada por $(W)_{ij} = \omega_N^{(i-1)(j-1)}$. As coordenadas do espectro, ou pontos no domínio das freqüências, também são chamados de **bins**. O valor absoluto de $\mathbf{X}$ é chamado de **espectro de potência**.

$$\begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)^2} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{pmatrix}$$

Computacionalmente, entretanto, a Transformada Discreta de Fourier tem uma complexidade muito grande, da ordem de N^2 operações. Utiliza-se, portanto, uma forma alternativa de calcular-se a DFT, chamada Transformada Rápida de Fourier, ou FFT (do inglês, Fast Fourier Transform). A FFT baseia-se em uma fatoração matricial que reduz a complexidade de cálculo da DFT para $\mathcal{O}(N \log N)$ operações. Nas seções seguintes, referimo-nos apenas à FFT, por ser a ferramenta de real uso computacional. Porém, fica aqui a observação de que, conceitualmente, a DFT e a FFT são iguais, diferindo apenas em sua forma de implementação.

3.1.6 Operadores Discretos

Para definirmos os métodos no domínio de freqüências para análise e ressíntese de sinais sonoros, precisaremos estudar algumas propriedades da DFT. Muitas dessas propriedades envolvem a modificação do sinal a partir de certas operações e a maneira como elas interferem em seu espectro. Para isso, vamos usar a notação $x \leftrightarrow X$ para denotar o fato de que X é o espectro de x, ou $X = Wx$, como já vimos matricialmente.

Antes de mostrarmos as propriedades, devemos definir algumas operações que podem ser feitas em sinais discretos, ou melhor, os operadores discretos que podemos aplicar sobre um sinal. A notação que usaremos será $\mathrm{op}(x)_n = x_k$, ou seja, a n-ésima coordenada do vetor $\mathrm{op}(x)$ é a k-ésima coordenada do vetor x.

Definição 21. *O operador* $\mathrm{Reverso}\,(x)$ *é definido por:*

$$\mathrm{Reverso}\,(x)_n = x_{-n}$$

O efeito desse operador é mais claro se usarmos indexação de $-\frac{N}{2}$ a $\frac{N}{2}$. Com a aplicação desse operador, o sinal é invertido horizontalmente, ou seja, começa do último ponto e termina com o primeiro.

Definição 22. *O operador* $\mathrm{Deslocamento}_{\,\Delta}(x)$ *é definido por:*

$$\mathrm{Deslocamento}_{\,\Delta}(x)_n = x_{n-\Delta}$$

O operador de deslocamento também é chamado de operador de deslocamento circular, pois $x_{(n+N)} = x_n$, e seu efeito independe de indexações.

$\mathrm{Deslocamento}_{\,1}([1,2,3,4])$	$=$	$[4,1,2,3] = \mathrm{Deslocamento}_{\,-3}([1,2,3,4])$
$\mathrm{Deslocamento}_{\,-1}([1,2,3,4])$	$=$	$[2,3,4,1] = \mathrm{Deslocamento}_{\,3}([1,2,3,4])$
$\mathrm{Deslocamento}_{\,N}(x)$	$=$	x
$\mathrm{Deslocamento}_{\,\Delta}(x)$	$=$	$\mathrm{Deslocamento}_{\,\Delta+kN}(x),\ k \in \mathbb{Z}$

Tabela 3.1: Exemplos da ação do operador de deslocamento.

Definição 23. *Dados dois vetores* $x, y \in \mathbb{C}^N$*, sua convolução é definida como:*

$$(x * y)_n = \sum_{m=0}^{N-1} x_m y_{n-m}$$

O operador de convolução é um operador binário que associa a dois vetores x e y de tamanho N, o vetor $x * y$. A convolução é comutativa, isto é, $(x * y) = (y * x)$. Prova-se isso, usando a mudança de variáveis $l = n - m$ e, depois, mudando o índice inicial do somatório. Como os vetores são períodicos, isso apenas reordena os termos.

$$(x * y)_n = \sum_{m=0}^{N-1} x_m y_{n-m} = \sum_{l=n}^{n-N+1} x_{n-l} y_l = \sum_{l=0}^{N-1} y_l x_{n-l} = (y * x)_n$$

Definição 24. *Sejam $x, y \in \mathbb{C}^N$, sua multiplicação pontual $x \cdot y$ é definida como:*

$$(x \cdot y)_n = x_n y_n$$

A multiplicação de dois vetores ponto a ponto também terá importância fundamental neste trabalho.

Definição 25. *Seja x um sinal discreto de tamanho M. Para $N > M$, o operador* $\text{AcZeros}_N(x)$ *é definido por:*

$$\text{AcZeros}_N(x)_n = \begin{cases} x_n, & 0 \leq n \leq M-1 \\ 0, & M \leq n \leq N-1 \end{cases}$$

O operador de acréscimo de zeros (zero-padding) corresponde ao aumento do tamanho do sinal através da introdução de zeros ao seu final. Chamamos $\frac{N}{M}$ de fator de acréscimo de zeros. Seja N ímpar. Se M é impar, $M < N$, e x é um sinal indexado de $-\frac{M-1}{2}$ até $\frac{M-1}{2}$ $N > M$, então o operador $\text{AcZeros}_N(x)$, pode ser definido por:

$$\text{AcZeros}_N(x)_n = \begin{cases} x_n, & |n| \leq \frac{M-1}{2} \\ 0, & \frac{M-1}{2} < |n| \leq \frac{N-1}{2} \end{cases}$$

A definição acima pode ser ajustada de acordo com a paridade de M e N. O importante é observar que esta definição pode variar de acordo com a indexação. Para sinais indexados de 0 a $M-1$, acrescentam-se os zeros no final, enquanto para sinais com índices positivos e negativos, acrescentam-se zeros em suas extremidades.

$$\text{AcZeros}_9([1,3,2,5]) = [1,3,2,5,0,0,0,0,0]$$

3.1.7 Propriedades da DFT

Teorema 3.2. *Sejam $x, y \in \mathbb{C}^N$, com $x \leftrightarrow X$ e $y \leftrightarrow Y$ e sejam $\alpha, \beta \in \mathbb{C}$. Então:*

$$\alpha x + \beta y \leftrightarrow \alpha X + \beta Y$$

Demonstração. Voltando à nossa definição, $x \leftrightarrow X \Leftrightarrow X = Wx$. Então $W(\alpha x + \beta y) = \alpha W x + \beta W y \Leftrightarrow \alpha x + \beta y \leftrightarrow \alpha X + \beta Y$. □

Teorema 3.3. *Para todo $x \in \mathbb{C}^N$, com $x \leftrightarrow X$,*

$$\overline{x} \leftrightarrow \text{Reverso}\,(\overline{X})$$

Demonstração.

$$\begin{aligned}
(W\bar{x})_k &= \sum_{n=0}^{N-1} \overline{x_n}\, e^{-\frac{i2\pi kn}{N}} = \overline{\sum_{n=0}^{N-1} x_n\, e^{\frac{i2\pi kn}{N}}} = \\
&= \overline{\sum_{n=0}^{N-1} x_n\, e^{-\frac{i2\pi(-k)n}{N}}} = \overline{X}_{-k} = \\
&= \text{Reverso}\,(\overline{X})_k
\end{aligned}$$

□

Teorema 3.4. *Para todo $x \in \mathbb{C}^N$, com $x \leftrightarrow X$,*

$$\text{Reverso}\,(\overline{x}) \leftrightarrow \overline{X}$$

Demonstração. Tomando $m = N - n$.

$$\begin{aligned}
(W\text{Reverso}\,(\overline{x}))_k &= \sum_{n=0}^{N-1} \overline{x}_{N-n}\, e^{-\frac{i2\pi kn}{N}} = \sum_{m=N}^{1} \overline{x}_m\, e^{-\frac{i2\pi k(N-m)}{N}} = \\
&= \sum_{m=0}^{N-1} \overline{x}_m\, e^{\frac{i2\pi km}{N}} = \overline{\sum_{m=0}^{N-1} x_m\, e^{-\frac{i2\pi km}{N}}} = \\
&= \overline{X}_k
\end{aligned}$$

□

Teorema 3.5. *Para todo $x \in \mathbb{C}^N$, com $x \leftrightarrow X$,*

$$\text{Reverso}\,(x) \leftrightarrow \text{Reverso}\,(X)$$

Demonstração.

$$\begin{aligned}(W\text{Reverso}\,(x))_k &= \sum_{n=0}^{N-1} x_{N-n}\, e^{-\frac{i2\pi kn}{N}} = \sum_{m=N}^{1} x_m\, e^{-\frac{i2\pi k(N-m)}{N}} = \\ &= \sum_{m=0}^{N-1} x_m\, e^{\frac{i2\pi km}{N}} = X_{-k} = \\ &= \text{Reverso}\,(X)_k\end{aligned}$$

□

A partir do Teorema 3.4 e do Teorema 3.5, podemos provar um resultado importante na análise do espectro de sinais reais, como os sonoros.

Teorema 3.6. *Para todo $x \in \mathbb{R}^N$, com $x \leftrightarrow X$,*

$$\text{Reverso}\,(X) = \overline{X}$$

Demonstração. Como x é real, $\overline{x} = x$. Logo,

$$\text{Reverso}\,(x) = \text{Reverso}\,(\overline{x}) \leftrightarrow \overline{X}$$

Por outro lado,

$$\text{Reverso}\,(x) \leftrightarrow \text{Reverso}\,(X)$$

Portanto,

$$\text{Reverso}\,(X) = \overline{X}$$

□

Isso significa que, se X é o espectro de um sinal real, $X_{-k} = \overline{X}_k$. Esse fato será muito importante quando interpretaremos o espectro de sinais sonoros. Veremos que toda a informação relevante à nossa análise está contida em apenas metade do espectro, sendo a outra metade redundante.

Teorema 3.7. $(\forall x \in \mathbb{C}^N)\ (\forall \Delta \in \mathbb{Z})$

$$(W\text{Deslocamento}_{\,\Delta}(x))_k = e^{-\frac{i2\pi k\Delta}{N}} X_k$$

Demonstração.

$$\begin{aligned} W[\text{Deslocamento}_{\Delta}(x)]_k &= \sum_{n=0}^{N-1} x_{n-\Delta} e^{-\frac{i2\pi nk}{N}} \\ &= \sum_{m=-\Delta}^{N-1-\Delta} x_m e^{-\frac{i2\pi(m+\Delta)k}{N}} \\ &= \sum_{m=0}^{N-1} x_m e^{-\frac{i2\pi mk}{N}} e^{-\frac{i2\pi\Delta k}{N}} \\ &= e^{-\frac{i2\pi\Delta k}{N}} \sum_{m=0}^{N-1} e^{-\frac{i2\pi mk}{N}} \end{aligned}$$

□

Esse teorema mostra que um deslocamento no sinal ocasiona uma mudança de fase linear em seu espectro.

Teorema 3.8. $(\forall x, y \in C^N)$, *vale:*

$$x * y \leftrightarrow X \cdot Y$$

Demonstração.

$$\begin{aligned} W(x*y)_k &= \sum_{n=0}^{N-1} (x*y)_n e^{-\frac{i2\pi nk}{N}} \\ &= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x_m y_{n-m} e^{-\frac{2\pi nk}{N}} \\ &= \sum_{m=0}^{N-1} x_m \sum_{n=0}^{N-1} y_{n-m} e^{-\frac{i2\pi nk}{N}} \\ &\overset{(*)}{=} \sum_{m=0}^{N-1} x_m [e^{-\frac{i2\pi mk}{N}} Y_k] \\ &= [\sum_{m=0}^{N-1} x_m e^{-\frac{i2\pi mk}{N}}] Y_k \\ &= X_k Y_k \end{aligned}$$

□

A igualdade $(*)$ é justificada pela utilização do Teorema 3.7. Porém, é o dual desse teorema que nos interessa mais em nossa análise.

Teorema 3.9. *$(\forall x, y \in \mathbb{C}^N)$, vale:*

$$x \cdot y \leftrightarrow \frac{1}{N} X * Y$$

A demonstração desse teorema segue os mesmos passos do anterior. Sua importância está no fato de que ele fundamenta a utilização de janelas na FFT.

Teorema 3.10. *Seja $x \in \mathbb{C}^N$, $N, M \in \mathbb{N}$ e $M > N$. O espectro de* $\text{AcZeros}_M(x)$ *corresponde a uma interpolação do espectro de x.*

Não demonstraremos esse último teorema. Porém, adiante veremos que isso realmente ocorre e será outra ferramenta muito importante em nossa análise para obtermos um espectro interpolado e com as informações que procuramos mais precisas. A demonstração desse teorema e outros operadores e propriedades da DFT podem ser encontrados em [22].

3.1.8 STFT

A Transformada Discreta de Fourier, que vimos até agora, é uma transformação que se aplica a um conjunto discreto e finito de pontos, associando a ele um outro conjunto discreto. Para isso, utilizam-se somatórios de todos os pontos do conjunto-domínio, ponderados por exponenciais complexas. Às vezes, porém, a aplicação da DFT em todos os pontos do domínio pode não ter um resultado satisfatório, fazendo-se necessária a divisão desses pontos em subconjuntos, nos quais a DFT é aplicada independentemente. Uma justificativa para esse procedimento surgirá naturalmente no decorrer deste capítulo.

A variação da DFT que apresentaremos aqui chama-se STFT, do inglês "Short Time Fourier Transform", que significa Transformada de Fourier em Tempo Curto. Esse nome vem da interpretação do conjunto de pontos discretos como sendo uma função no tempo. Assim, aplicaria-se a DFT em subconjuntos de pontos contíguos, os quais corresponderiam a intervalos curtos de tempo.

Não formalizaremos uma definição precisa da STFT, pois ela não corresponde a um conceito novo, e sim, a uma estratégia de aplicação da DFT. Apenas definiremos um quadro relacionado a um conjunto de pontos discretos e a Transformada de Fourier desse quadro.

Definição 26. *Seja $x(t)$, $t \in \mathbb{Z}$ um conjunto de pontos discretos. Um quadro x_{t_0} de x de tamanho M é definido por $x_{t_0}(n) = x(t_0+n)$, para $n = 0, ..., M-1$, e com $t_0 \in \mathbb{Z}$ fixo.*

Podemos também utilizar a notação $x_{t_0} = x(t_0 : t_0 + M - 1)$.

Definição 27. *Seja x_{t_0} um quadro de tamanho M de $x(t)$, $t \in \mathbb{Z}$, a Transformada Discreta de Fourier associada a esse quadro é dada por:*

$$X_{t_0}(k) = \sum_{n=0}^{M-1} x_{t_0}(n) e^{-\frac{i2\pi nk}{M}}$$

Assim, pela STFT, relacionamos cada quadro de um conjunto discreto ao seu espectro. Mais detalhes desse procedimento surgirão durante sua utilização. Por exemplo, não há a necessidade de aplicarmos a DFT a todos os quadros possíveis dentro de um conjunto discreto. Dependendo da aplicação, escolhem-se os quadros relevantes ao problema. Estamos usando aqui o ponto inicial do quadro para indexar, ou identificar, cada quadro, pois nosso conjunto discreto é geral, podendo ser infinito. Na prática, a indexação pode ser diferente, pois, para um conjunto finito de pontos, haverá um conjunto finito de quadros de um determinado tamanho, que podem ser indexados de forma ordenada.

3.2 Extração de Parâmetros no Domínio das Freqüências

Vejamos, na prática, como a Teoria de Fourier pode ser aplicada à análise de sinais. Os exemplos, aqui apresentados, descreverão passo a passo o procedimento, que é o princípio fundamental da análise de sinais: a estimativa de freqüência, amplitude e fase de senóides.

3.2.1 Análise de Uma Senóide

Começaremos analisando uma simples senóide de freqüência f, amplitude A e fase inicial θ, que pode ser gerada através da função $\cos(2\pi ft + \theta)$. Para facilitar a compreensão, vamos trabalhar com a freqüência de amostragem $f_s = 1$, de forma que o índice de cada ponto equivale ao tempo absoluto que ele representa. Assim, temos $t = 0, 1, ..., N - 1$. Pelo Teorema de Nyquist (Teorema 1.1), temos que a maior freqüência representável em uma amostragem é $\frac{f_s}{2}$. Portanto, não podemos utilizar freqüências maiores que $0,5$. Em nosso primeiro exemplo, vamos supor $A = 1$, $f = 0.25$, $\theta = 0$ e $N = 8$. Assim, teremos que nossa senóide discretizada S será a função $\cos(\frac{\pi}{2}t)$ aplicada sobre os pontos $t = 0, 1, ..., 7$. Logo, $S = (1, 0, -1, 0, 1, 0, -1, 0)$.

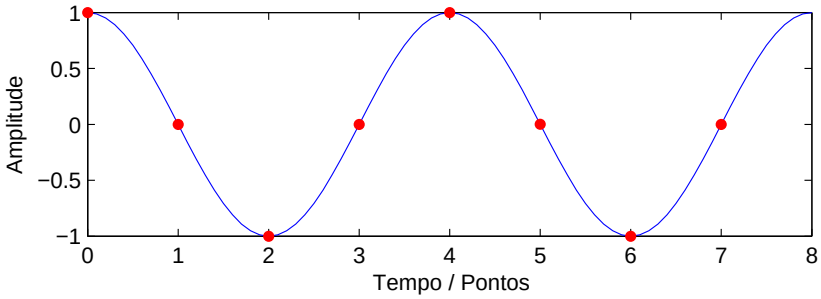

Figura 3.1: $\cos(\frac{\pi}{2}t)$

Aplicando a FFT em S, obtemos $X = FFT(S) = (0, 0, 4, 0, 0, 0, 4, 0)$. Neste caso, como o cosseno é uma função real e par, temos que sua transformada também é real e par que, neste caso específico, já é seu próprio valor absoluto. Temos também, pelo Teorema 3.6, que, para um sinal real, os dados contidos em seu espectro, que se encontram após o índice $\frac{N}{2} + 1$, são redundantes. Estes podem ser eliminados, durante a etapa de análise, e reconstruídos na etapa de síntese, antes da utilização da FFT inversa. Esses pontos serão descartados. Relembremos, por (3.9) e (3.10), que os pontos de X representam a intensidade das freqüências $w_k = k\Delta w$, em que $\Delta w = \frac{1}{P}$. Nesse caso, $P = N\Delta t = \frac{N}{f_s} = 8$. Assim, temos que os valores que obtivemos no espectro $(0, 0, 4, 0, 0)$ correspondem às freqüências $(0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8},)$, respectivamente. Esse fato está de acordo com o Teorema de Nyquist, já que a maior freqüência representada é $0.5 = \frac{f_s}{2}$.

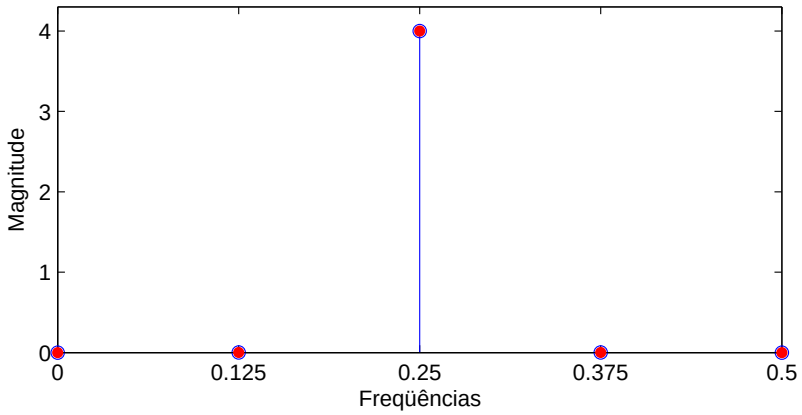

Figura 3.2: Espectro de $\cos(\frac{\pi}{2}t)$.

Neste exemplo, a freqüência da senóide é exatamente a freqüência de um

dos bins do espectro, sendo este o único não nulo no espectro, com valor absoluto igual a $\frac{AN}{2}$. Já a fase pode ser encontrada tomando-se o valor correspondente do ângulo neste mesmo bin, que é 0, pois a parte imaginária é 0. Portanto, quando isso ocorre, podemos facilmente recuperar, com exatidão (desconsiderando-se erros inerentes aos cálculos númericos), os valores das freqüência, amplitude e fase de uma senóide simples.

Ainda podemos apresentar outros exemplos deste caso com parâmetros diferentes. Por exemplo, se a senóide fosse gerada por uma função seno ao invés de cosseno, a fase inicial encontrada seria $-\frac{\pi}{2}$, pois $\sin(2\pi ft) = \cos(2\pi ft - \frac{\pi}{2})$.

Tomemos agora $f = \frac{3}{8}$, $A = 4$ e $\theta = \frac{\pi}{4}$. Nesse caso, temos a senóide discreta:

$$S = (2.8284, -4, 2.8284, 0, -2.8284, 4, -2.8284, 0)$$

e sua transformada:

$$X = (0, 0, 0, 11.3137 + 11.3137i, 0, 11.3137 - 11.3137i, 0, 0, 0).$$

Excluindo-se os valores redundantes e tomando o valor absoluto, que representa seu espectro de potência, temos que $(0, 0, 0, 16, 0)$ são as magnitudes correspondentes às freqüências $(0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8})$. Assim, vemos que a magnitude não nula está na freqüência $\frac{3}{8} = 0.375$, sendo esta a freqüência da senóide. A amplitude é $A = \frac{16}{\frac{N}{2}} = 4$. A fase é obtida pelo ângulo do complexo $11.3137 + 11.3137i$, que é $\frac{\pi}{4}$, pois as partes real e imaginária são iguais e positivas.

No entanto, o caso do qual acabamos de tratar é muito raro na prática. É difícil que a freqüência da senóide seja exatamente a freqüência de uns dos bins presentes no domínio de seu espectro. De fato, na prática, além de isso nunca ocorrer, ainda há muitas outras senóides envolvidas, tornando a tarefa de identificar suas freqüências, amplitudes e fases muito mais complexa.

Apresentaremos agora um outro exemplo que ainda é muito bem comportado e de fácil tratamento. Mantendo os parâmetros do primeiro exemplo, $A = 1$, $f = 0.25$, $\theta = 0$, vamos apenas mudar o valor de N para 10. Com isso, temos dois períodos e meio da função $\cos(\frac{\pi}{2}t)$, sendo que o período total do sinal de entrada agora vale 10. Mas isto implica que o intervalo entre os bins será de $\frac{1}{10}$ e não haverá nenhum bin com a freqüência da senóide. Assim, não haverá apenas um bin não nulo e a energia da freqüência da senóide será distribuída por todas as freqüências do domínio. Aplicando a FFT no sinal, temos

$$X = (1, 1 + 0.727i, 1 + 3.078i, 1 - 3.078i, 1 - 0.727i,$$
$$1, 1 + 0.727i, 1 + 3.078i, 1 - 3.078i, 1 - 0.727i)$$

Omitindo os dados redundantes e obtendo o valor absoluto, temos:

$$\hat{X} = (1, 1.236, 3.236, 3.236, 1.236, 1)$$

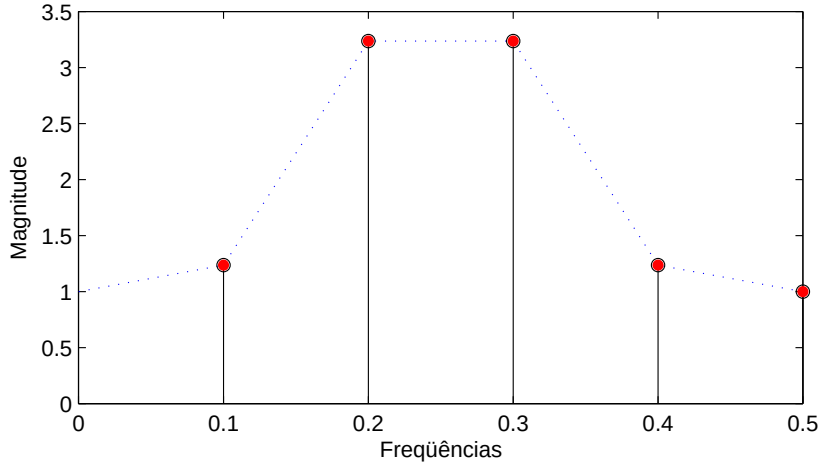

Figura 3.3: Espectro de $\cos(\frac{\pi}{2}t)$ com $N = 10$.

Como a freqüência da senóide é de 0.25, era de se esperar que suas freqüências vizinhas fossem as de maior magnitude e, também, que fossem iguais, já que a primeira se encontra exatamente no meio das duas. Porém, não temos como recuperar a amplitude exata nem a fase com esses dados. Entretanto, como sabemos que haverá um pico entre as freqüências de maior intensidade, podemos utilizar uma parábola para estimar tal pico. Segundo dados empíricos encontrados na literatura, o pico da interpolação será duas vezes mais preciso em relação ao pico real se a parábola for traçada no espectro normalizado em decibéis [19]. Aplicando-se tal transformação a $\hat{X}$ temos:

$$\tilde{X} = (0, 1.841, 10.2, 10.2, 1.841, 0)$$

Em geral, para traçar a parábola, toma-se o bin de maior magnitude e seus dois vizinhos. Porém, nesse caso, temos dois bins de mesma magnitude, assim como os bins laterais. Portanto, há duas escolhas equivalentes possíveis. Tomaremos o bin de freqüência 0.2 como o central, sendo lhe atribuído o valor 0 no eixo horizontal da parábola, enquanto seus bins vizinhos, de freqüência 0.1 e 0.3, terão o valor de -1 e 1, respectivamente, com a finalidade de simplificar os cálculos. Assim que encontrarmos o pico da parábola, este é transladado novamente para sua posição absoluta para que saibamos seu real valor. Precisamos encontrar uma parábola $p(x) = ax^2 + bx + c$ tal que

$p(-1) = \alpha$, $p(0) = \beta$ e $p(1) = \gamma$, no caso que $\alpha = 1.841$, $\beta = 10.2$ e $\gamma = 10.2$. Sabendo-se os coeficientes da parábola, poderemos encontrar o máximo em $x_{max} = \frac{-b}{2a}$. Resolvendo o sistema

$$\begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

temos:

$$\begin{aligned} a &= \frac{\gamma + \alpha}{2} - \beta \\ b &= \frac{\gamma - \alpha}{2} \\ c &= \beta \end{aligned}$$

Portanto, $x_{max} = \frac{1}{2}\frac{\alpha - \gamma}{\alpha - 2\beta + \gamma}$
Substituindo os valores de α, β e γ, temos que $x_{max} = 0.5$ é a posição relativa do pico em relação aos bins. Sua real localização é $\hat{f} = 0.2 + 0.1 \times 0.5 = 0.25$. Para obter a amplitude, calculamos $p(0.5) = 11.245$ e, passando de dB para a escala linear, temos que a magnitude do pico é 3.65. Falta-nos estimar apenas a fase inicial da senóide. Para isso vamos considerar os ângulos dos bins do espectro:

$$(0, 0.628, 1.257, -1.257, 0.628, 0)$$

Diferentemente da magnitude, a fase não tem um comportamento que se aproxima de uma parábola. Portanto, tentaremos a interpolação linear como estimativa e veremos se foi ou não uma boa escolha. Interpolando linearmente a fase entre os bins de freqüência 0.2 e 0.3, temos que ela será dada por:

$$\hat{\theta} = 0.5 \times 1.257 + (1 - 0.5)(-1.257) = 0$$

Assim, temos que, neste caso, a interpolação linear foi suficiente para recuperarmos a fase inicial da senóide. Dessa forma, conseguimos recuperar, com precisão, a freqüência e a fase inicial da senóide, obtendo uma estimativa para sua amplitude.

3.2.2 Acréscimo de Zeros

Nos exemplos que apresentamos, foram utilizados poucos pontos de amostragem para facilitar os cálculos e a compreensão de todo o fenômeno. Devido a isso, podemos identificar apenas uma faixa pequena de freqüências desses sinais. Na prática, a FFT é aplicada sobre um número muito maior de

pontos. Duas maneiras de acrescentarmos pontos ao nosso sinal discreto de entrada são: aumento do intervalo de amostragem e aumento da taxa de amostragem. Para analisarmos a forma como cada um desses procedimentos altera o espectro gerado pela FFT, relembremos, novamente, que $w_k = k\Delta w$, em que $\Delta w = \frac{1}{P}$.

Quando aumentamos a taxa de amostragem, temos mais pontos representando o sinal no mesmo período de tempo. Neste caso, como o período P continua igual, Δw também é o mesmo, ou seja, a menor freqüência representável e o intervalo entre as freqüências de bins consecutivos se mantém. Porém, aumentando a taxa de amostragem, aumentamos o valor da maior freqüência detectável. Logo, esse procedimento aumenta a banda de freqüências do espectro.

Por outro lado, se mantivermos a mesma taxa de amostragem e aumentarmos o período da amostra, teremos a mesma banda de freqüências detectável, porém o intervalo entre os bins será menor, ou seja, teremos maior precisão na detecção dos valores de freqüência. Em nosso segundo exemplo, poderíamos aumentar o período das senóides até que um dos bins do espectro correspondesse à freqüência da senóide que gostaríamos de identificar. Porém, na prática, raramente temos uma senóide exatamente períodica ao longo de um grande período de tempo. Também devem ser analisados períodos curtos do sinal, para que possamos enxergar suas características instantâneas, por exemplo, a nota que está sendo tocada no momento. Portanto, não devemos aumentar muito o período da amostra para evitar grandes variações no quadro em que aplicaremos a FFT.

Existe ainda uma outra forma de aumentarmos a precisão do espectro sem alterarmos o conteúdo do quadro. Utilizando o operador definido em (3.12), vamos acrescentar zeros ao sinal. Assim, não alteramos a taxa de amostragem e aumentamos o período sem acrescentar novas informações ao sinal. Logo, temos Δw menor e um espectro mais preciso, ou seja, uma interpolação no espectro (Teorema 3.10). Um fator de acréscimo de zeros de $\frac{N}{M}$ acarreta em um mesmo fator de interpolação, ou seja, o número de bins no espectro aumenta em $\frac{N}{M}$.

Apesar de muito utilizado, o acréscimo de zeros possui um efeito colateral conhecido como **vazamento** (leakage): mesmo que, com a interpolação do espectro, a freqüência da senóide passe a coincidir com a freqüência de um dos bins (e nesse caso poderíamos encontrá-la com exatidão), ainda assim, os demais bins não serão nulos, formando picos de menor intensidade conhecidos como **morros laterais** (side lobes), cujas freqüências não pertencem ao sinal. Chamaremos de **morros principais** (main lobes) os conjuntos dos máximos locais (e seus bins adjacentes), oriundos de freqüências pertencentes ao sinal.

Para ilustrar esse fato, vamos tomar um exemplo parecido com o último

que estudamos. Seja uma senóide com parâmetros $A = 1$, $f = 0.25$, $\theta = \frac{\pi}{4}$. Vamos analisá-la com $N = 10$ pontos e $f_s = 1$. A única diferença para o exemplo anterior é a fase inicial. Porém, apenas essa mudança é suficiente para que agora os bins vizinhos do bin que representa a freqüência da senóide, mesmo que equidistantes deste, não tenham mesma magnitude. Isso faz com que a interpolação por uma parábola falhe em acusar a freqüência real da senóide. Se estendêssemos a senóide de forma a completar mais um período ($N = 12$ ou $N = 16$), teríamos um bin correspondente à freqüência da senóide do qual extrairíamos a amplitude e os outros bins nulos. Porém, supondo que não sabemos como se comporta em seguida o sinal analisado, não podemos completar seu período, a não ser acrescentando zeros. Na Figura 3.4, vemos um gráfico que mostra, em vermelho, o espectro apenas dos 10 pontos do sinal e, em azul, o espectro do sinal com 16 pontos, sendo 6 zeros extras.

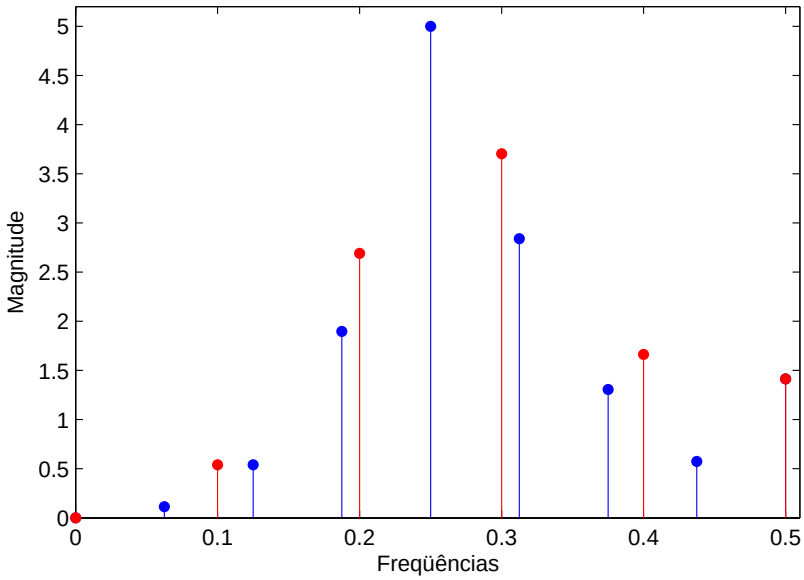

Figura 3.4: Espectro de $\cos(\frac{\pi}{2}t)$ com $N = 10$ e acréscimo de zeros.

Por se tratarem de poucos pontos discretos, fica difícil termos uma visão mais ampla do fenômemo que ocorre com o espectro, devido ao acréscimo de zeros no domínio do tempo. Faremos agora uma interpolação mais densa do espectro, acrescentando zeros até que o tamanho do sinal seja de 256 pontos. Essa nova interpolação está ilustrada na Figura 3.5. Nessa mesma figura, os pontos do espectro, sem acréscimo de zeros, estarão destacados. A escala desse gráfico será agora em decibéis. Assim, vemos claramente o efeito dos morros laterais causados pelo acréscimos de zeros e como eles geram novos

falsos picos no espectro. Porém, há um jeito de amenizar este problema, que é aplicando janelas sobre o sinal.

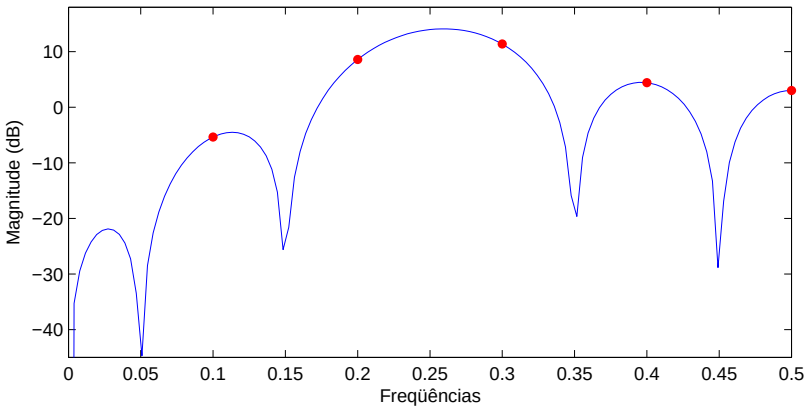

Figura 3.5: Espectro de $\cos(\frac{\pi}{2}t)$ densamente interpolado.

3.2.3 Aplicação de Janelas

Aplicar uma janela a um sinal x significa multiplicar pontualmente x por uma função discretizada w de mesmo número de pontos que x (Definição 24). A multiplicação pontual $x \cdot y$, no domínio do tempo, corresponde a uma convolução no domínio das freqüências (Teorema 3.9), ou seja, o espectro do sinal é filtrado pelo espectro da janela. O espectro de uma senóide simples multiplicada por uma janela é o espectro da janela ponderado pela amplitude da senóide e centrado em sua freqüência. Assim, escolhendo uma janela cujo espectro possua um pico central e um decaimento lateral, podemos diminuir a altura dos morros laterais causados pelo acréscimo de zeros. A não utilização explícita de uma janela corresponde ao uso de uma janela chamada retangular. Logo adiante, veremos exemplos de algumas outras janelas. Em geral, definimos janelas de tamanho M ímpar formadas por pontos $w(n)$, $n = -\frac{M-1}{2}, ..., \frac{M-1}{2}$ e $w(k) = w(-k)$. Em alguns casos, podemos também considerar $w(n)$, definida para todo $n \in \mathbb{Z}$, tal que $w(n) = 0$, para $|n| > \frac{M-1}{2}$.

Vamos voltar ao nosso exemplo anterior e aplicar uma janela sobre a senóide para vermos o que ocorre com o espectro. Utilizaremos uma janela de Hamming, que é definida por:

$$w_h(n) = 0.54 - 0.46\cos(2\pi\frac{n}{N}), 0 \leq n \leq N.$$

Neste caso, para $N = 10$, temos que a janela será dada pelos pontos:

$$(0.08, 0.188, 0.46, 0.77, 0.972, 0.972, 0.77, 0.46, 0.188, 0.8).$$

Observe que, se N fosse ímpar, teríamos uma janela com um ponto central, cujo valor seria 1. Na Figura 3.6, vemos o gráfico do espectro do sinal transformado utilizando-se essa janela. Como no gráfico anterior, temos o espectro interpolado, com alto fator de acréscimo de zeros, os pontos do espectro, sem acréscimo de zeros, destacados.

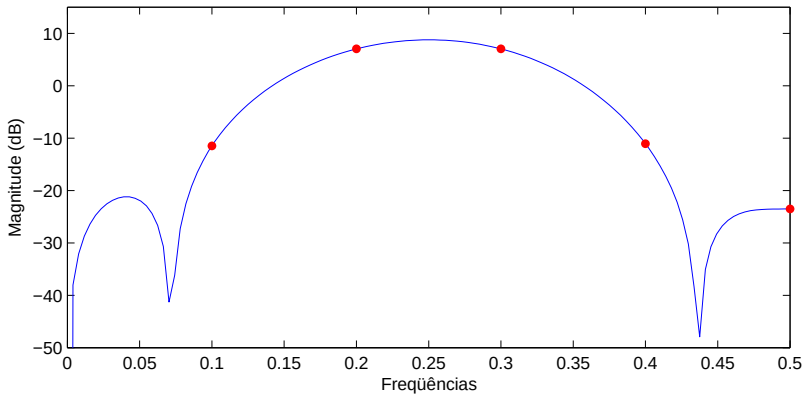

Figura 3.6: Espectro de $\cos(\frac{\pi}{2}t)$ com janela de Hamming.

Comparando os dois gráficos, percebe-se que, com a utilização da janela de Hamming, os morros laterais ficaram muito menores em relação ao morro principal, quando comparados com o espectro sem o uso de janelas (ou com o uso da janela retangular) no sinal. Porém, desta vez, o efeito colateral foi o de tornar mais largo o morro principal, i. e., aquele onde se localizam a freqüência e a amplitude da senóide. Isso pode afetar a precisão da freqüência da senóide detectada e na distinção entre dois picos próximos.

3.2.4 Análise de Duas Senóides

Vamos agora considerar o caso em que o sinal a ser analisado é composto de duas senóides. Quando as freqüências das senóides são suficientemente distantes uma da outra, apenas ocorrerá uma combinação dos casos anteriores. O caso que merece mais atenção é quando suas freqüências são muito próximas. Vamos considerar, por exemplo, duas senóides de freqüências $f_1 = 4.85$ e $f_2 = 5.1$ e amplitudes $A_1 = 1$ e $A_2 = 2$, representadas em um intervalo de tempo de tamanho 4 e $f_s = 20$. Apresentaremos, na Figura 3.7, o gráfico do espectro da soma das duas senóides. O traçado contínuo, em azul, representa o espectro com um fator de acréscimo de zeros igual a 4, ou seja, acrescentou-se ao sinal de entrada uma quantidade de zeros tal que o tamanho final foi 4 vezes superior ao sinal sem os zeros extras. Os pontos

destacados em vermelho correspondem ao espectro do sinal sem acréscimo de zeros.

O gráfico da direita apresenta o espectro do sinal com janela retangular (ou sem janela), enquanto o gráfico da direita apresenta o espectro do sinal aplicando-se janela de Hamming ao sinal antes da FFT. Os espectros estão em decibéis e ambos os gráficos estão na mesma escala, para que possa ser notada a diferença de altura entre os morros laterais com e sem a janela.

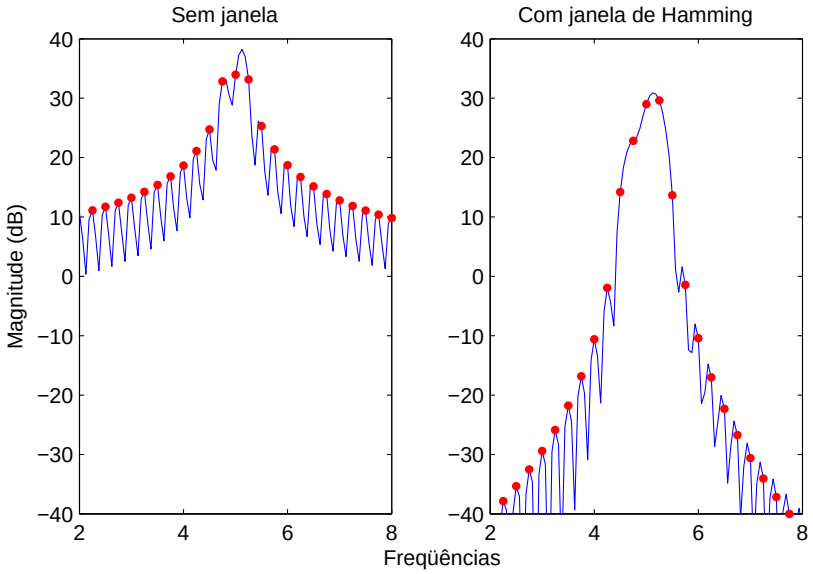

Figura 3.7: Espectro de duas senóides próximas.

Pode-se perceber o quanto a janela de Hamming atenua os morros laterais, de tal forma que não são confundidos com morros principais. Porém, a proximidade entre as freqüências das duas senóides presentes no sinal torna difícil sua separação. No gráfico à esquerda, o espectro sem janelas e sem acréscimo de zeros não permite a diferenciação entre os dois picos. Porém, o acréscimo de zeros resolve o problema. Já no gráfico à direita, com janela de Hamming, torna-se impossível distinguir os dois picos, mesmo com acréscimo de zeros: a aplicação da janela ao sinal torna o morro principal mais largo, fazendo com que os dois picos se fundam em um só, o que torna impossível sua separação.

Esses são os passos para se analisar um quadro isolado de um sinal. Porém, como um sinal sonoro, em geral, pode ser longo e com muitas modulações, esta análise deve ser aplicada em pequenos pedaços do sinal para detectar suas características instantâneas e acompanhar suas mudanças ao longo do tempo. Assim, nos algoritmos que apresentaremos, o real procedimento para análise de um som consiste: em dividir o sinal em quadros; aplicar

a FFT separadamente em cada quadro (STFT); e interpretar o espectro de cada quadro como acabamos de fazer. Antes de apresentarmos esses algoritmos, vamos exemplificar algumas janelas muito utilizadas no processamento de sinais.

3.2.5 Exemplos de Janelas

Veremos, agora, alguns tipos de janelas utilizados na STFT e como se comportam seus espectros de potência. As janelas serão definidas por $w(n)$, $n = -\frac{M-1}{2}, ..., \frac{M-1}{2}$, para M ímpar, de modo que sejam simétricas e alcancem o ápice no ponto $n = 0$. Nos gráficos, as janelas serão de tamanho $M = 31$ e seu espectro será interpolado via acréscimo de 225 zeros ($N = 256$). O espectro será normalizado, resultando no domínio $[-0.5, 0.5]$, e terá magnitude expressa em decibéis, com valor máximo 0. Dessa maneira, poderemos visualizar a altura dos morros laterais em relação ao morro principal.

Janela Retangular

A janela retangular corresponde à não utilização explícita de janelas e pode ser definida por

$$w_R(n) = 1, n = -\frac{M-1}{2}, ..., \frac{M-1}{2}$$

No espectro dessa janela, o maior morro lateral encontra-se 13dB abaixo do morro principal.

Janela Triangular

É definida por:

$$w_T(n) = \left(1 - \frac{2|n|}{M-1}\right)$$

No espectro dessa janela, o maior morro lateral encontra-se 26dB abaixo do morro principal.

Janela de Hanning

A Janela de Hanning é definida por:

$$w(n) = 0.5 - 0.5\cos(\frac{2\pi n}{N})$$

No espectro dessa janela, o maior morro lateral encontra-se 31dB abaixo do morro principal.

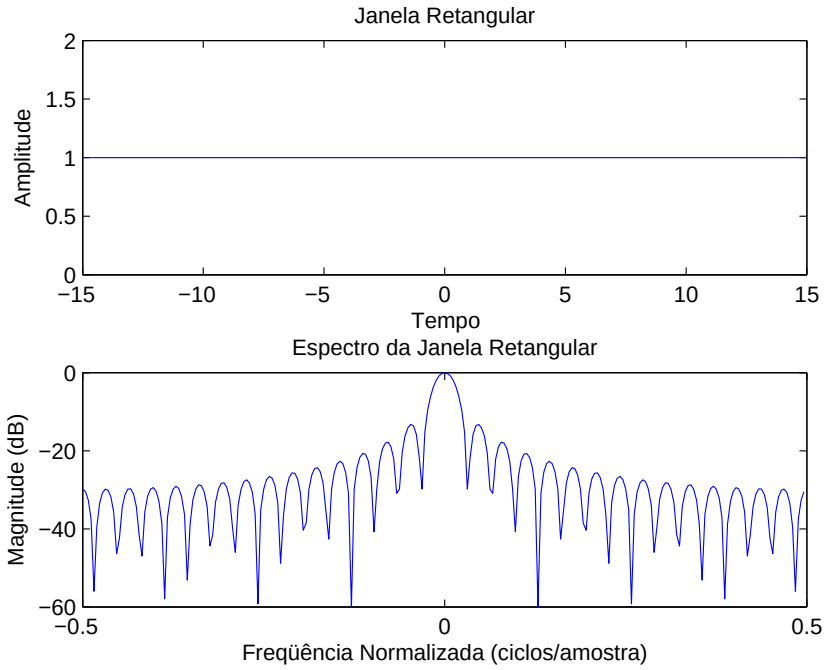

Figura 3.8: Janela Retangular e seu espectro.

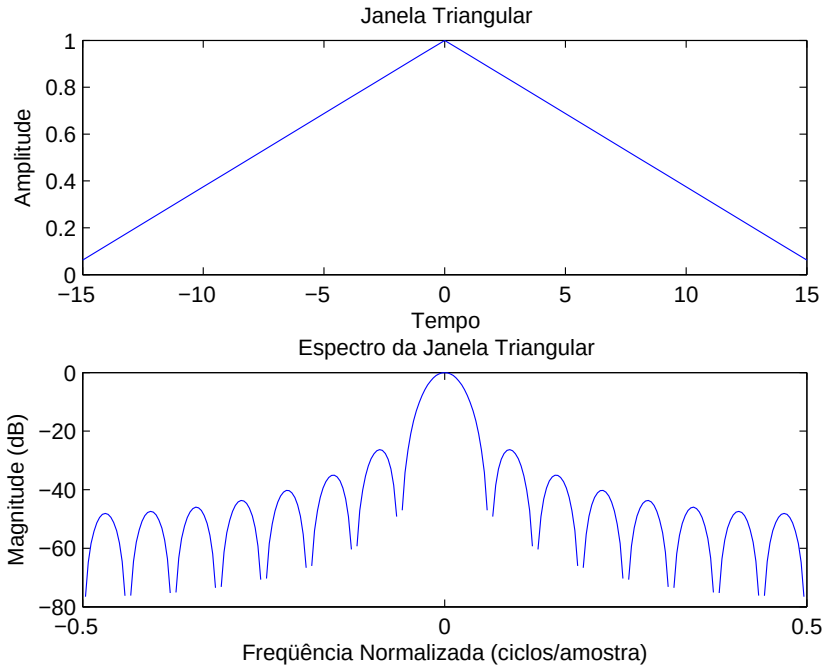

Figura 3.9: Janela Triangular e seu espectro.

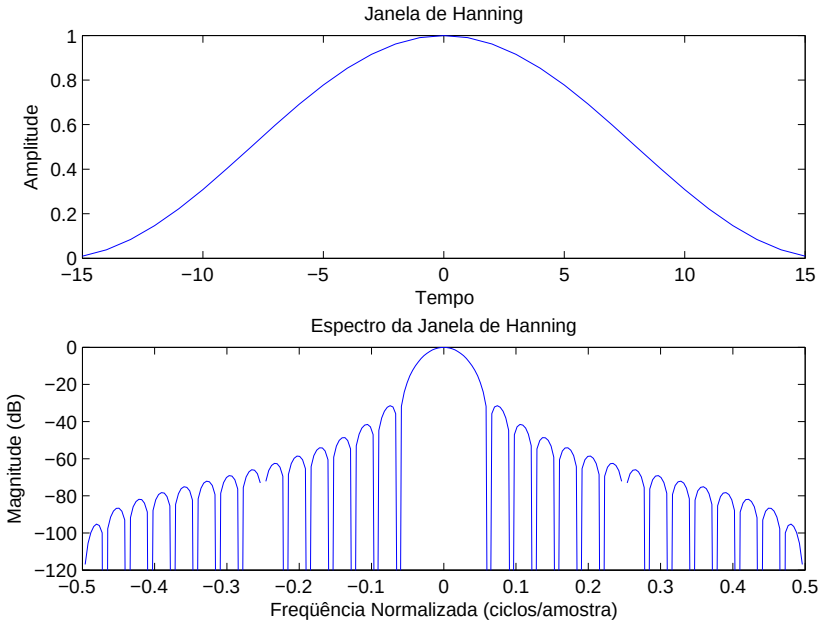

Figura 3.10: Janela de Hanning e seu espectro.

Uma variante da janela de Hanning é a janela de Hann. A diferença entre elas é que a janela de Hann possui dois zeros extras, um em cada extremidade. Assim, a janela de Hann de tamanho M é equivalente à janela de Hanning de tamanho $M-2$, com 0 nas extremidades.

Janela de Hamming

A janela de Hamming é dada por:

$$w(n) = 0.54 - 0.46\cos(2\pi\frac{n}{N})$$

No espectro dessa janela, o maior morro lateral encontra-se 41dB abaixo do morro principal.

Janela de Blackman-Harris

A janela de Blackman-Harris é dada por:

$$w(n) = 0.35875 - 0.48829\cos(\frac{2\pi n}{M}) + 0.14128\cos(\frac{4\pi n}{M}) - 0.01168\cos(\frac{6\pi n}{M})$$

No espectro dessa janela, o maior morro lateral encontra-se 92dB abaixo do morro principal.

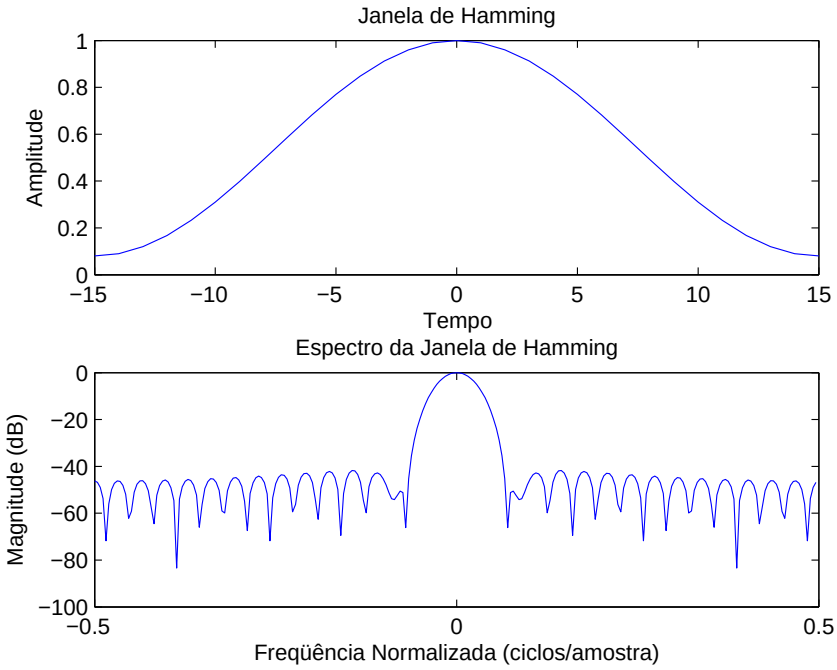

Figura 3.11: Janela de Hamming e seu espectro.

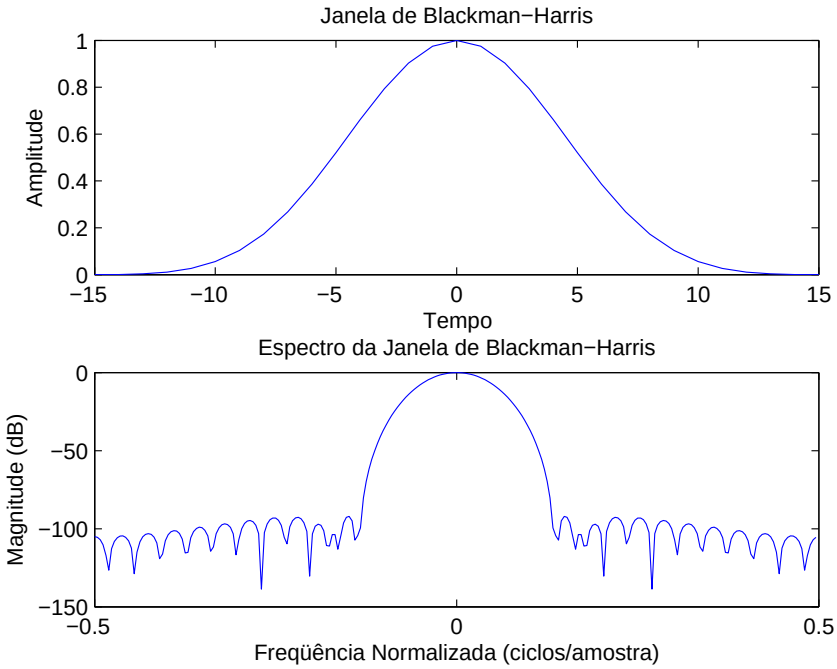

Figura 3.12: Janela de Blackman-Harris e seu espectro.

3.3 Phase Vocoder

O Phase Vocoder é um algoritmo para análise e modificação de sinais sonoros, que baseia-se na utilização da STFT para o tratamento espectral quadro a quadro [17] [10]. Uma consideração importante para se entender o phase vocoder é a percepção do fenômeno físico que relaciona a freqüência e a fase de uma senóide. Até agora, tratamos apenas da fase inicial de senóides, sem entrar em detalhes de como a fase se comporta com a variação do tempo.

Imagine um ponto em movimento percorrendo o círculo trigonométrico. Cada vez que o ponto retorna à origem do círculo, dizemos que ele completou um período. Podemos relacionar o movimento do ponto com uma senóide, de forma que o período da senóide será o tempo que o ponto leva para completar uma volta, enquanto sua freqüência será o número de vezes que o ponto descreve o círculo por unidade de tempo. Para tornar essa relação ainda mais concreta, a senóide pode ser definida como o cosseno do ângulo que o ponto ocupa no círculo.

Considere agora que o tempo estivesse congelado em um dado momento do movimento do ponto. Poderíamos, então, observar a posição angular que o ponto ocupa no círculo naquele instante. Essa é a fase instantânea da senóide. Note que a freqüência da senóide depende da velocidade com a qual o ponto se move no círculo. Portanto, a freqüência nada mais é do que a variação de posição do ponto no círculo (ou fase da senóide), ou seja, a freqüência é a derivada da fase.

No phase vocoder, aplica-se a FFT e, como anteriormente, obtemos um conjunto discreto de bins no espectro, cada um representando uma freqüência. Porém, como já vimos, a freqüência real pode não corresponder a nenhum dos bins que compõem o espectro. Considera-se, aqui, que cada bin é um filtro que permite a passagem de uma banda limitada ao seu redor. Estima-se a freqüência real derivando-se a fase no mesmo bin entre dois quadros consecutivos.

Mas, nesse processo, deve ser observada uma outra questão: de um quadro para outro, não sabemos exatamente quantos ciclos foram percorridos, já que temos apenas o valor da fase em um intervalo de tamanho 2π. Deve-se, portanto, aplicar um processo chamado **desencapsulamento** (unwrapping), para que o valor da fase do quadro seguinte seja estendido de forma a permitirmos um intervalo entre fases superior a 2π. Podemos estimar quantos ciclos foram percorridos entre as duas fases, baseando-nos na freqüência do bin em questão.

Porém, o que mais nos interessa sobre o phase vocoder é sua parte de ressíntese, de onde surgem suas duas principais aplicações, as quais se encontram presentes nesse trabalho.

Para entender sua importância, imagine um disco de vinil em uma vitrola na qual a rotação possa ser alterada. Imagine agora que queremos acelerar a música aumentando a rotação da vitrola. Quando isso ocorre, a onda sonora é reproduzida mais rapidamente, de forma que sua variação de fase aumenta. Fazendo a analogia com o círculo trigonométrico, o ponto varia mais rapidamente sua posição angular, aumentando a freqüência de seu movimento e, conseqüentemente, a freqüência da senóide. Da mesma forma, se reduzirmos a rotação da vitrola, a música ficará mais lenta e suas freqüências ficarão menores, ou seja, o tom da música ficará mais grave.

Na música, no entanto, é muito importante podermos alterar separadamente a freqüência (altura) e a velocidade, sem que uma interfira na outra. A mudança de tom, ou seja, multiplicação de todas as freqüências presentes no sinal por um mesmo fator, também conhecida na música como transposição, é muito importante, por exemplo, para que um cantor ajuste uma música ao seu tom de voz, ou para que um instrumentista ajuste-a à afinação de seu instrumento. Por outro lado, a alteração da velocidade sem mudança de tom é igualmente importante. Pode ser citada, como aplicação, a necessidade de um instrumentista começar a prática da execução de uma peça lentamente, até chegar à velocidade desejada, sendo esse processo altamente recomendado aos estudantes de música que desejam atingir a excelência em seus instrumentos. Ou seja, o phase vocoder se adapta às aplicações citadas por separar informações temporais de informações espectrais e ressintetizar o sinal modificado de forma simples e rápida.

Para alterar a velocidade sem mudança de tom, vamos pensar em um quadro de tempo isolado. Nele, através dos bins, temos as informações das freqüências presentes naquele instante, que dura a distância de um quadro para outro. Podemos, então, alterar a velocidade aumentando a distância entre um quadro e outro. Isso faz com que aquele instante seja prolongado e as freqüências do quadro sejam emitidas por um intervalo maior de tempo. Para isto, basta mudarmos a distância entre os quadros do sinal durante a etapa de ressíntese. Ao fazermos isso, ocorre que, em cada quadro, a freqüência, amplitude e fase das senóides são mantidas. Porém, pensando novamente no ponto do círculo trigonométrico, a senóide será prolongada. O ponto percorrerá uma distância maior e dará mais voltas no círculo. Portanto, sua fase será alterada, ocasionando uma mudança indesejada na freqüência como derivada da fase, que será incompatível com a freqüência armazenada nas informações do quadro, o que gera distorções.

A sincronia entre as informações de fase e freqüência é o ponto mais delicado da ressíntese sonora e deve ser muito bem observada para que não ocorram distorções e ruídos no resultado final. No caso acima, deve-se recalcular a posição da fase em cada quadro para que ela corresponda à freqüência

esperada. Fazendo-se isso, o phase vocoder gera um resultado muito bom.

Já a mudança de tom, sem alteração de velocidade, é feita pelo phase vocoder da seguinte maneira: primeiro é utilizado o processo anterior para mudar a velocidade da música (inicialmente sem alteração de tom), na mesma proporção que deseja-se modificar o tom; posteriormente, altera-se a freqüência de amostragem do sinal, de forma que, durante sua reprodução, a velocidade volte à original, dessa vez alterando o tom na forma desejada. Seguindo a analogia anterior, é como se a vitrola tivesse a rotação alterada, fazendo o disco voltar à velocidade original, mas acarretando uma mudança de tom, que desta vez é desejada.

Na prática, a alteração da freqüência de amostragem é inconveniente, por ocasionar perda de qualidade ou aumento desnecessário de armazenamento de dados. Portanto, é feita uma interpolação dos dados (conversão digital para analógico) e, depois, uma reamostragem, mudando o intervalo entre os pontos.

3.4 PARSHL

Apresentaremos agora o método espectral de análise e síntese de sinais digitais chamado PARSHL [19], que se baseia no modelo senoidal variável no tempo. As etapas desse algoritmo são as seguintes:

1. Divisão do sinal em quadros e aplicação da STFT;
2. Análise individual do quadro;
3. Análise entre quadros;
4. Síntese.

A seguir veremos como funciona cada uma dessas estapas e discutiremos as questões levantadas por ela.

3.4.1 Divisão em Quadros e Aplicação da STFT

A transformada de Fourier é uma ferramenta utilizada correntemente para analisar o espectro das freqüências que compõem um som. Porém, como um som em geral pode ser longo e com muitas modulações, a transformada deve ser aplicada em pequenos pedaços dos som, de modo a detectar suas características instantâneas e acompanhar suas mudanças ao longo do tempo. Portanto, divide-se o sinal em quadros e utiliza-se a STFT (Short-Time Fourier Transform) para realizar essa tarefa.

Seja $x(t)$, $t = 0, ..., P-1$, um sinal discreto, com freqüência de amostragem f_s em Hz (pontos por segundo). $T = \frac{1}{f_s}$ é o intervalo de tempo, em segundos, entre as amostras. Seja M ímpar, $M < P$. Definimos $M_h = \frac{M-1}{2}$. Vamos considerar os quadros de x da seguinte forma:

$$x_m(n) = x((m-1)R + n + M_h),$$

em que $m = 1, ..., Q$, $n = -M_h, ..., M_h$ e R é o número de pontos entre um quadro e outro. O intervalo de tempo entre dois quadros será de RT segundos.

Estamos supondo que o número total de quadros, $Q = \frac{P-M}{R} + 1$, seja inteiro. Caso isso não aconteça, acrescentaremos zeros ao sinal $x(t)$, gerando um novo sinal $x'(t)$ de tamanho P'. Esse procedimento, diferentemente do que vimos anteriormente, não terá influência no domínio das freqüências do sinal, pois não mudará o tamanho do quadro. Sua influência é em seu domínio de tempo, correspondendo ao acréscimo de silêncio ao sinal. A maneira como esses zeros serão incluídos será discutida mais adiante.

Agora que temos quadros bem definidos, vamos aplicar em cada um deles a FFT. Como já vimos, podemos aplicar um pré-processamento nesses dados para que sua interpretação seja mais fácil, ou seja, para que as características que devemos extrair (freqüência, amplitude e fase) fiquem mais evidentes. Por motivos já apresentados, aplicamos sobre o quadro uma janela $w(n)$, obtendo:

$$\tilde{x}_m(n) = x_m(n)w(n)$$

Posteriormente, acrescentamos zeros ao resultado, de forma que o quadro passe a ter um tamanho N. Para facilitar o cálculo da FFT e tornar o processamento mais rápido, escolhe-se N como sendo uma potência de 2. Definimos $N_h = \frac{N}{2}$. Após o acréscimo de zeros, temos o seguinte quadro:

$$\tilde{x}'_m(n) = \begin{cases} \tilde{x}_m(n), & |n| \leq M_h \\ 0, & M_h < n \leq N_h \text{ ou } -N_h + 1 \leq n < -M_h \end{cases}$$

Agora sim, aplica-se a Transformada de Fourier em sua forma discreta:

$$X_m(k) = \sum_{n=-N_h}^{N_h-1} \tilde{x}'_m(n) e^{-\frac{j2\pi nk}{N}} = \sum_{n=-N_h}^{N_h-1} \tilde{x}'_m(n) e^{-jw_k nT}$$

em que $w_k = \frac{2\pi k f_s}{N}$ e k é o índice das amostras discretas no domínio de freqüências (bins).

A próxima etapa será a interpretação dos dados obtidos pela FFT em cada quadro isolado. Porém, para que obtenhamos um resultado satisfatório,

que facilite a interpretação dos parâmetros que compõem o som, precisamos escolher adequadamente as variáveis envolvidas na STFT. Portanto, antes de prosseguirmos, faremos uma análise dessas variáveis, que são tipo de janela ($w(n)$), tamanho do salto (R), tamanho do quadro (M) e tamanho da FFT (N).

I) Janelas:

Dois parâmetros devem ser considerados na escolha da janela: a largura do morro principal e altura dos morros laterais. O desejado são morros principais estreitos e morros laterais baixos. Morros secundários altos podem ser vistos, equivocadamente, como morros principais, resultando na adição de freqüências indesejadas (ruído) ao sinal. Morros principais muito largos diminuem a precisão na detecção da freqüência; morros principais próximos, se forem muito largos, podem ser interpretados como um único morro principal. A diferença de largura também pode ser utilizada como critério para classificar os morros como principais ou secundários, já que os primeiros são muito mais largos que os segundos. Há um balanço entre os dois parâmetros, de forma que janelas com menor largura do morro principal apresentam maiores morros laterais e vice-versa. Deve-se procurar um equilíbrio entre esses fatores.

II) Tamanho do salto

O tamanho do salto, R, é o número de pontos entre dois inícios de quadros consecutivos. Um salto pequeno resulta em mais quadros e maior qualidade de análise. Porém, demanda maior esforço computacional. Novamente, deve-se buscar um equilíbrio entre esses fatores. A escolha do R ideal também depende da janela escolhida. Dada uma janela $w(n)$, tal que $w(n) = 0, \forall |n| > |M_h|$, o tamanho do salto R deve satisfazer a seguinte equação:

$$(\forall t)\, A_w(t) = \sum_{m=-\infty}^{\infty} w(t - mR) = C, \tag{3.12}$$

em que C é uma constante. Se isso acontecer, todos os pontos de amostragem recebem o mesmo peso na soma dos quadros e, portanto, a análise do sinal não sofrerá distorções. Para a janela retangular de largura M, por exemplo, o salto pode ser de $\frac{M}{k}$, para qualquer k inteiro tal que $\frac{M}{k}$ seja também inteiro.

III) Tamanho do quadro

O tamanho do quadro, M, é outro parâmetro a ser escolhido no PARSHL. Se usarmos um quadro muito grande, perderemos as características instantâneas

do som. Por outro lado, um quadro muito curto dificulta a detecção de freqüências mais graves, pois o quadro pode ser menor que o período de tais freqüências. Recomenda-se, portanto, que o tamanho do quadro seja, aproximadamente, o tamanho do período da freqüência mais grave presente no sinal. Para aplicações de áudio, como o ouvido humano não reconhece freqüências abaixo de 20 Hz, podemos tomar um quadro de tamanho 0,05s (50ms). Por outro lado, o ouvido humano também não identifica freqüências acima de 20KHz. Logo, podemos usar $f_s \cong 40KHz$. Assim,

$$T = \frac{1}{f_s} = \frac{1}{4.10^4} = 0,25.10^{-4}s$$

Como a duração do quadro, MT, é igual a 0,05s, $M = \frac{0,05}{0,25.10^{-4}} = 0,2.10^4$. Ou seja, $M \approx 2000$.

Se soubermos que o sinal não possui freqüências tão graves, pode-se diminuir o tamanho de M. Para sinais com f_s menor, podemos fazer o mesmo. Por outro lado, M deve ser ímpar para que o quadro possua um ponto central $x_m(0)$, no qual a janela que será aplicada ao quadro deverá assumir seu valor máximo. Esse fato reflete-se no espectro do quadro, acentuando seus picos principais.

IV) Tamanho da FFT

O tamanho da FFT, N, também é um parâmetro arbitrário. N deve ser um número maior ou igual a M, para preencher o quadro com $N - M$ zeros. Também deve ser uma potência de 2, para facilitar a aplicação da FFT. Uma escolha padrão para N é a primeira potência de 2 maior que $2M$ (em nosso exemplo de $M = 2001$, teríamos $N = 2^{12} = 4096 \geq 2M = 4002$). A importância desse procedimento é que, se um quadro recebe acréscimo de $\frac{N}{M}$ zeros, seu espectro terá $\frac{N}{M}$ vezes mais pontos do que se não houvesse acréscimo de zeros. Assim, a resolução do espectro aumenta, gerando maior precisão na hora de encontrar a freqüência na qual ocorre o pico de magnitude.

Vimos que o quadro é centrado no ponto 0, possuindo parte positiva e negativa. Porém, computacionalmente, não se usam índices negativos. Logo, no quadro a ser transformado, a parte negativa aparecerá no final e os zeros aparecerão no meio.

3.4.2 Análise Individual do Quadro

Na etapa anterior, utilizamos a STFT para dividir o sinal em quadros e associar a cada quadro seu espectro dado pela FFT. Nesta etapa, nosso

foco é um quadro isolado. Interpretaremos seu espectro, de forma a encontrar as freqüências presentes naquele instante do sinal, junto com suas respectivas amplitudes e fases. Já vimos anteriormente, passo a passo, o que acontece com o espectro de uma ou duas senóides depois de aplicada uma janela e acrescentados zeros ao quadro. Vimos, também, como determinar a freqüência que cada bin representa e, conseqüentemente, determinar sua amplitude e fase. Faremos apenas uma breve recaptulação, voltada ao nosso estado atual dentro do algoritmo.

Denotamos $X_m(k)$, $k = 0, .., N-1$, o espectro complexo do quadro x_m. Tomando apenas os dados relevantes, consideramos $k = 1, ..., N_h + 1$. Assim, para os pontos discretos do domínio de freqüências, temos:

$$\begin{aligned} \text{Freqüência: } f_m(k) &= \frac{k}{NT} \\ \text{Amplitude: } A_m(k) &= |X_m(k)| \\ \text{Fase: } \theta_m(k) &= \angle X_m(k) \end{aligned}$$

Sabemos que as freqüências que compõem o sinal raramente encontram-se presentes com exatidão entre as freqüências representadas pelos bins e, com isso, sua energia é dividida entre as freqüências vizinhas. Portanto, para aproximarmos as reais freqüências, devemos analisar os morros encontrados no módulo do espectro, ou seja, onde encontram-se as maiores amplitudes $A_m(k)$ em seu espectro de potência.

Como o espectro de potência é discreto, a maior magnitude (eixo y) nele encontrada corresponderá a uma freqüência (eixo x) com precisão de meio bin, para mais ou para menos. Para uma janela retangular, uma precisão de 0,1% seria obtida apenas com fator de acréscimo de zeros superior a 1000, o que é impraticável. Porém, utilizando interpolação quadrática nos 3 pontos de maior magnitude em um mesmo morro, pode-se atingir a mesma precisão com um fator de acréscimo de zeros igual ou superior a 5. Dados empíricos também mostram que, se a interpolação for feita utilizando a magnitude em decibéis (escala logarítmica), a precisão também aumentará duas vezes em relação à interpolação com magnitude linear [19]. Assim, dado um máximo local no espectro de potência em decibéis, usamos interpolação quadrática para aproximar a real localização do pico da parábola, cujas coordenadas correspondem à freqüência (eixo x) e amplitude (eixo y) que buscamos.

Porém, resta-nos a questão de quantos (ou quais) máximos locais podem ser considerados como componentes senoidais do sinal. Há duas maneiras para tratarmos isso. Uma delas é a de distinção entre os morros correspondentes, a componentes senoidais, e os gerados por ruídos ou vazamento. Os critérios para essa seleção são altura e largura dos morros. No entanto,

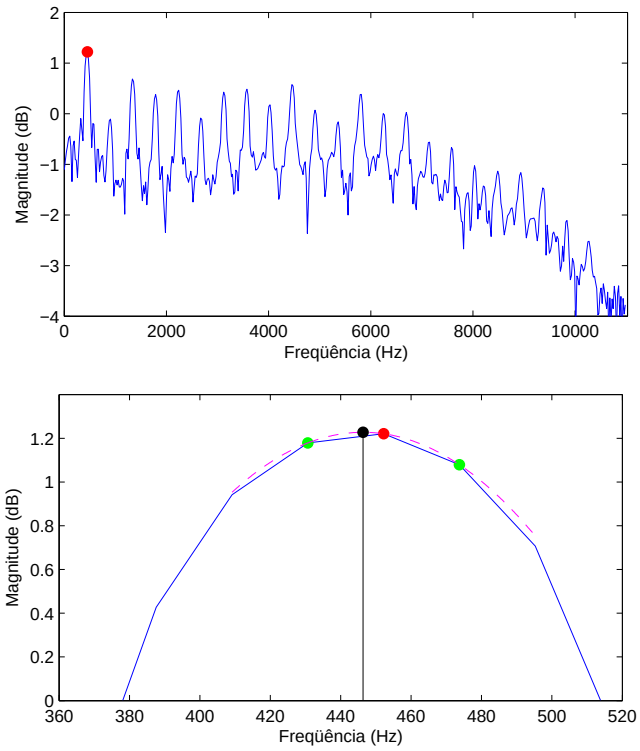

Figura 3.13: Detecção de picos no espectro de um violino.

em um sinal com muitas componentes e de amplitudes variadas, essa tarefa torna-se muito subjetiva, sendo que morros secundários correspondentes a amplitudes altas podem ser confundidos com morros principais de amplitudes mais baixas.

Portanto, o critério mais adotado é o de se utilizar um número fixo de picos a serem detectados em cada quadro. Dessa forma, procura-se o ponto de maior magnitude no espectro, utiliza-se seus pontos vizinhos para a determinação de uma parábola e estima-se seu pico. Traduzem-se, então, suas coordenadas em amplitude e freqüência associadas a uma senóide. Depois disso, retira-se os pontos associados a esse morro do espectro de potência e procura-se o próximo ponto de maior magnitude. Faz-se isso até que o número estipulado de picos por quadro tenha sido detectado.

3.4.3 Análise Entre Quadros

Analisando cada quadro, separadamente, obtemos as características instantâneas do som. Relacionando as informações encontradas em um quadro com as do quadro seguinte, podemos ver como essas características se comportam com o passar do tempo, criando uma interpretação dinâmica do sinal. Faremos isso em duas etapas: primeiro, vamos relacionar o espectro todo de um quadro ao do quadro seguinte, vendo como amplitude e fase variam entre os bins; depois, relacionaremos apenas os dados referentes aos picos encontrados na etapa anterior.

Variações de parâmetros entre quadros

Um detalhe crucial para o entendimento do próximo passo do processo é observar que, como M e N são constantes, assim como o intervalo entre as amostras T, o domínio das freqüências será sempre o mesmo, ou seja, os bins são fixos. Logo, $X_{m-1}(k)$ e $X_m(k)$ correspondem a duas amostras subseqüentes do mesmo bin. Dessa forma, se quisermos ver como cada bin se comporta no decorrer do tempo, transformamos as funções $X_m(k)$ (uma função para cada quadro cuja variável é a freqüência) em funções de freqüência fixa, cuja variável é o tempo. Assim, teremos as funções de amplitude e fase para cada bin, sendo a variável de tempo o índice do quadro.

$$\begin{aligned} \text{Amplitude: } A_k(m) &= |X_m(k)| \\ \text{Fase: } \theta_k(m) &= \angle X_m(k) \end{aligned}$$

Assim, cada bin é associado a uma **faixa** ou um **oscilador** com freqüência constante. Osciladores são dispositivos que geram um movimento senoidal.

Diremos que duas freqüências pertecem a mesma faixa, ou são geradas pelo mesmo oscilador, se forem os parâmetros instantâneos de uma mesma senóide em tempos diferentes. Se não considerarmos todos os bins, e sim, somente as freqüências obtidas no processo de deteção de picos, teremos faixas que não terão freqüências constantes. Portanto, devemos decidir como associar dois picos entre dois quadros diferentes.

Interligação de picos

O processo de detecção de picos nos fornece as principais freqüências presentes em um quadro, junto com suas respectivas amplitudes e fases. Sabemos que os quadros avançam no tempo através de um período arbitrário. Logo, de um quadro para outro podem ocorrer algumas situações com as freqüências encontradas:

a) uma freqüência pode se manter de um quadro para o outro ou sofrer modulações muito leves;

b) uma freqüência presente no quadro anterior pode não estar no seguinte;

c) uma freqüência não presente no quadro anterior pode surgir no seguinte.

Consideraremos que o sinal será gerado por osciladores, cada um gerando uma senóide com freqüência e amplitude variáveis no tempo. Assim, de um quadro para outro, cada oscilador pode continuar a gerar a senóide, parar de gerá-la, ou começar a gerá-la.

Suponhamos que as freqüências de um quadro são $f_1, f_2, ..., f_p$ e as do quadro seguinte são $g_1, g_1, ..., g_r$. Dado Δf, se, para algum f_i houver g_j tal que

$$|f_i - g_j| \leq \Delta f, \tag{3.13}$$

podemos considerar g_j como continuação de f_i, sendo gerados pelo mesmo oscilador (situação a)). Assim, dizemos que a freqüência g_j pertence à mesma faixa que a freqüência f_i.

Se algum f_i não satisfaz a desigualdade (3.13), ou seja, não há nenhum g_j com freqüência suficientemente próxima à sua, então consideraremos que a faixa correspondente a f_i é desligada de um quadro para outro (situação b)). Assim, a faixa correspondente a f_i manterá essa mesma freqüência de um quadro para outro, porém sua amplitude diminuirá suavemente até 0 no quadro seguinte.

Se algum g_j não satisfaz a desigualdade (3.13) para nenhum f_i, então consideraremos que uma nova faixa, de freqüência inicial g_j é criada de um

quadro para o outro (situação c)). Assim, inclui-se no quadro anterior a faixa com freqüência inicial g_j e amplitude inicial 0.

Caso haja conflito, isto é, existam g_{j_1} e g_{j_2} que satisfaçam (3.13), a faixa será prolongada para a freqüência mais próxima (menor diferença), enquanto a outra procurará uma nova faixa para se ligar. Esse processo continua até que todas as faixas sejam encaixadas em uma das 3 situações acima.

Na prática, não usamos Δf fixo, pois o ouvido humano é sensível a variações relativas de freqüências. Δf será definido em função da freqüência f_i, para cuja faixa buscamos uma continuação. Outra questão interessante a ser discutida nesse ponto é a relação entre o número de picos detectados por quadro e o número de faixas existentes entre um quadro e outro. Sendo p o número de picos por quadro fixo, se utilizarmos todos os picos detectados em ambos os quadros, o número de faixas entre dois quadros estaria entre p (caso todas as faixas sejam prolongadas) e $2p$ (caso nenhuma faixa obtenha uma continuação). Entretanto, o caso de nenhum pico ser combinado de um quadro para outro é indesejado, pois acarreta uma grande descontinuidade no processo de síntese. Para evitar que isso aconteça, é comum a detecção de um número de picos superior ao número dos picos que serão efetivamente utilizados. Assim, prioriza-se a ligação das faixas já existentes de forma que, para cada faixa desligada, cria-se uma nova faixa para o pico de maior intensidade que surgir no quadro seguinte.

3.4.4 Síntese

Após o processo de análise, podemos montar uma linha de tempo mostrando como cada faixa varia sua fase, amplitude e freqüência com o tempo para reconstruir o som original. Temos duas alternativas para a reconstrução do sinal: síntese por sobreposição e síntese aditiva.

Síntese por Sobreposição

Uma alternativa para reconstruir o som é a síntese por sobreposição, que consiste em inverter pela FFT inversa cada quadro e depois fazer a sobreposição, somando pontos comuns a mais de um quadro, conforme descrito a seguir.

Inversão do quadro:

$$\hat{x}'_m(n) = \frac{1}{N} \sum_{-N_h}^{N_h-1} X_m(k) e^{jw_k n}$$

Reconstrução por sobreposição:

$$\hat{x}(t) = \sum_{m=1}^{Q} \hat{x}'_m(n)$$

em que $n = t - (m-1)R - M_h$ e $n \in [-M_h, M_h]$.

Os processos de análise e síntese por sobreposição são inversos (sua composição forma uma identidade), caso as janelas obedeçam a propriedade dada pela equação (3.12), com $C = 1$. Caso a condição (3.12) não seja satisfeita, haverá uma modulação de amplitude de período R distorcendo o sinal original, ou seja, uma distorção periódica de freqüência $\frac{f_s}{R}$.

Síntese Aditiva

A síntese aditiva, ao contrário da síntese por sobreposição, não é baseada no espectro dos quadros individuais, e sim, nas faixas que se estendem ao longo do tempo. Nela, a reconstrução do som pode ser interpretada como instruções passadas a osciladores para gerarem senóides de amplitude e freqüência variáveis no tempo. Podemos considerar dois casos: cada bin pode ser interpretado como uma faixa; ou ressintetizamos um número de faixas menor, contendo apenas as informações mais importantes do sinal. No primeiro caso, a síntese aditiva se equivale à síntese por sobreposição, tendo a desvantagem de não usar a FFT inversa para diminuir a quantidade de operações.

Para cada faixa, temos sua amplitude, freqüência e fase $(\hat{A}^r_m, \hat{f}^r_m, \hat{\theta}^r_m)$ em cada quadro. Vamos escolher um ponto de cada quadro para assumir esses valores e faremos interpolações desses parâmetros entre os pontos escolhidos. Na STFT, utilizamos janelas de tamanho ímpar, em que o ponto central assume o valor máximo. Portanto, escolheremos o ponto central de cada quadro como seu representante. Portanto, temos:

$$A_r((m-1)R + M_h) = \hat{A}^r_m$$

em que $\hat{A}^r_m$ é a amplitude da faixa r no quadro m e $A_r(t)$ é a amplitude da faixa r no ponto t.

Agora que definimos qual ponto de cada quadro assumirá as propriedades de suas faixas, antes de estabelecermos como será a interpolação de um ponto ao outro, vamos voltar a uma questão levantada anteriormente. Quando falamos sobre o número de quadros em que o sinal seria dividido, argumentamos que deveriam ser acrescentados zeros ao sinal, de forma que seu novo tamanho fosse de P' pontos, com $\frac{P'-M}{R}$ inteiro. Porém, deixamos em aberto a questão de como esses zeros seriam inseridos. Agora que decidimos que o ponto central de cada quadro será seu representante, podemos acrescentar M_h

zeros no início do sinal, para que o primeiro quadro tenha como representante o primeiro ponto de amostragem. Caso contrário, o ponto inicial receberia peso inferior aos outros e, como não haveria quadro anterior do qual ele fizesse parte, a condição (3.12) não seria satisfeita, a não ser que uma janela retangular fosse usada. O mesmo ocorre para o último ponto. Acrescentamos, portanto, ao final do sinal, M_h zeros mais a quantidade necessária para que $\frac{P'-M}{R}$ seja inteiro. Assim, todos os pontos do sinal original recebem mesma ponderação, satisfazendo a equação (3.12). As eventuais desproporções das janelas inicial e final cairiam sobre os zeros acrescentados e seriam anuladas por eles.

Voltando à interpolação da síntese aditiva, temos Q pontos para os quais sabemos $(A_r(t), f_r(t), \theta_r(t))$. Precisamos de uma interpolação satisfatória para os pontos que se localizam entre eles, de forma que as transições de amplitude, fase e freqüência ocorram suavemente, ou seja, sem causar ruídos ou efeitos bruscos indesejados, perceptíveis auditivamente.

Interpolação entre quadros

Para cada faixa temos, então, os parâmetros iniciais e os parâmetros finais, relativos ao seu comportamento entre dois quadros, e precisamos reconstruí-la em todo esse intervalo. Então, para cada faixa r, teremos os parâmetros $(\hat{A}^r_{m-1}, \hat{f}^r_{m-1}, \hat{\theta}^r_{m-1})$, referentes à amplitude, freqüência e fase no quadro anterior, e $(\hat{A}^r_m, \hat{f}^r_m, \hat{\theta}^r_m)$, referentes ao quadro posterior. Vamos definir esses parâmetros para todos os pontos intermediários $n = 1, ..., S-1$, em que S é o número de pontos entre os inícios de dois quadros da etapa de síntese (se não há a intenção de alterar a velocidade do sinal, $S = R$). Como estaremos fazendo a interpolação dentro de uma mesma faixa, os índices de faixa r serão omitidos por conveniência.

A amplitude pode ser interpolada linearmente sem grandes problemas, de forma que o volume do som seja alterado gradativamente de um quadro para outro. Portanto, teremos:

$$\hat{A}_m(n) = \hat{A}_{m-1} + \frac{\hat{A}_m - \hat{A}_{m-1}}{S} n \tag{3.14}$$

Já a freqüência e a fase não podem ser interpoladas separadamente desta forma, pois a fase instantânea é dependente das fases e freqüências iniciais e finais, enquanto a freqüência instantânea é a derivada da fase. Assim, procuramos uma função para gerar uma interpolação suave de fase e freqüência. Por termos 4 parâmetros disponíveis, utilizaremos uma polinomial cúbica, da forma:

$$\Theta(n) = \zeta + \gamma n + \alpha n^2 + \beta n^3 \tag{3.15}$$

Apesar de $\Theta(n)$ ser discreta, vamos tratá-la como uma função contínua e derivável, para que possamos encontrar os parâmetros necessários para a interpolação e, depois, calculá-la nos pontos discretos. Como a freqüência instantânea é a derivada da fase, teremos a seguinte função para as freqüências:

$$\Theta'(n) = \gamma + 2\alpha n + 3\beta n^2$$

Sabemos os valores iniciais e finais de fase e freqüência. Logo,

$$\begin{aligned} \hat{\theta}_{m-1} &= \Theta(0) = \zeta \\ \hat{f}_{m-1} &= \Theta'(0) = \gamma \\ \hat{f}_m &= \Theta'(S) = \hat{f}_{m-1} + 2\alpha S + 3\beta S^2 \\ \hat{\theta}_m + 2\pi\eta &= \Theta(S) = \hat{\theta}_{m-1} + \hat{f}_{m-1} S + \alpha S^2 + \beta S^3 \end{aligned}$$

A condição $\Theta(S) = \hat{\theta}_m + 2\pi\eta$ é necessária pois $\hat{\theta}_M$ está em um intervalo de comprimento 2π e, assim, não sabemos quantas voltas a fase percorreu até chegar no ponto atual. Portanto, o fator $2\pi\eta$ é necessário como forma de desencapsulamento da fase. Dessa forma, para cada $\eta \in \mathbb{Z}$, temos uma função de interpolação diferente, com parâmetros $\alpha = \alpha(\eta)$ e $\beta = \beta(\eta)$. Posteriormente, decideremos qual delas será a mais suave, a qual corresponderá à melhor interpolação de fase.

Vamos agora encontrar $\alpha(\eta)$ e $\beta(\eta)$ em função das outras variáveis:

$$\left\{ \begin{array}{rcrcl} S^2\alpha(\eta) & + & S^3\beta(\eta) & = & \hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1} + 2\pi\eta \\ 2S\alpha(\eta) & + & 3S^2\beta(\eta) & = & \hat{f}_m - \hat{f}_{m-1} \end{array} \right.$$

Matricialmente, temos:

$$\begin{pmatrix} S^2 & S^3 \\ 2S & 3S^2 \end{pmatrix} \begin{pmatrix} \alpha(\eta) \\ \beta(\eta) \end{pmatrix} = \begin{pmatrix} \hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1} S + 2\pi\eta \\ \hat{f}_m - \hat{f}_{m-1} \end{pmatrix}$$

Podemos resolver o sistema invertendo a matriz 2×2:

$$\begin{pmatrix} \alpha(\eta) \\ \beta(\eta) \end{pmatrix} = \begin{pmatrix} \frac{3}{S^2} & \frac{-1}{S} \\ \frac{-2}{S^3} & \frac{1}{S^2} \end{pmatrix} \begin{pmatrix} \hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1} S + 2\pi\eta \\ \hat{f}_m - \hat{f}_{m-1} \end{pmatrix}$$

Falta apenas definirmos qual η gerará a função de interpolação mais suave. Lembrando que, como estamos tomando intervalos pequenos entre quadros e freqüências próximas de um quadro para outro, a tendência é que a freqüência tenha variação mínima. No caso ideal, de freqüência constante, teríamos que a derivada da freqüência seria nula durante todo o intervalo. Procuraremos, então, a função mais próxima deste caso, ou seja, a função em que o quadrado

da integral da derivada da freqüência seja mínima. Como a freqüência no intervalo é dada por $\Theta'(n)$, sua derivada será $\Theta''(n)$. Temos que encontrar η que minimize:

$$\int_0^S [\Theta''(n,\eta)]^2 dn \tag{3.16}$$

Para seguirmos a notação tradicional da matemática, vamos esquecer temporariamente que já estamos usando as variáveis x e t, pois queremos usá-las como variáveis contínuas de tempo e espaço. Nossa função Θ será agora de duas variáveis.

$$\Theta(t,x) = \hat{\theta}_{m-1} + \hat{f}_{m-1}t + \alpha(x)t^2 + \beta(x)t^3$$

em que:

$$\begin{aligned}
\alpha(x) &= \frac{3}{S^2}(\hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1}S + 2\pi x) - \frac{1}{S}(\hat{f}_m - \hat{f}_{m-1}) \\
\beta(x) &= \frac{-2}{S^3}(\hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1}S + 2\pi x) + \frac{1}{S^2}(\hat{f}_m - \hat{f}_{m-1})
\end{aligned}$$

Para encontrarmos η que minimize (3.16), vamos buscar x^* que minimize a função contínua em x e depois tomar η^* como o inteiro mais próximo de x^*. Devemos, então, minimizar em x a seguinte função:

$$\int_0^S [\frac{d^2\Theta}{dt^2}(t,x)]^2 dt \tag{3.17}$$

Temos que:

$$\begin{aligned}
[\frac{d^2\Theta}{dt^2}(t,x)]^2 &= [2\alpha(x) + 6\beta(x)t]^2 \\
&= 4\alpha^2(x) + 24\alpha(x)\beta(x)t + 36\beta^2(x)t^2
\end{aligned}$$

Substituindo em (3.17),

$$\begin{aligned}
\int_0^S [\frac{d^2\Theta}{dt^2}(t,x)]^2 dt &= \int_0^S (4\alpha^2(x) + 24\alpha(x)\beta(x)t + 36\beta^2(x)t^2)dt \\
&= 4\alpha^2(x)S + 12\alpha(x)\beta(x)S^2 + 12\beta^2(x)S^3 = \Phi(x)
\end{aligned}$$

Temos agora que minimizar $\Phi(x)$. Como $\Phi(x)$ tem comportamento quadrático, com termos de ordem 2 positivos, basta encontrar x^* tal que $\Phi'(x^*) = 0$. Porém,

$$\Phi'(x) = 8S(\alpha(x)\alpha'(x)) + 12S^2(\alpha'(x)\beta(x) + \alpha(x)\beta'(x)) + 24S^3\beta(x)\beta'(x) \tag{3.18}$$

em que:

$$\begin{aligned}\alpha'(x) &= \frac{6\pi}{S^2} \\ \beta'(x) &= \frac{-4\pi}{S^3}\end{aligned}$$

A título de simplificação, vamos definir :

$$u(x) = (\hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1}S + 2\pi x)$$

e

$$v = (\hat{f}_m - \hat{f}_{m-1})$$

Assim,

$$\alpha(x) = (\frac{3}{S^2}u(x) - \frac{1}{S}v)$$

e

$$\beta(x) = (\frac{-2}{S^3}u(x) + \frac{1}{S^2}v)$$

Substituindo $\alpha(x)$, $\alpha'(x)$, $\beta(x)$ e $\beta'(x)$ em (3.18), temos:

$$\Phi'(x) = (\frac{144\pi}{S^3}u(x) - \frac{48\pi}{S^2}v) + (\frac{72\pi}{S^2} - \frac{144\pi}{S^3}u(x)) +$$

$$+(\frac{48\pi}{S^2}v - \frac{144\pi}{S^3}u(x)) + (\frac{192\pi}{S^3}u(x) - \frac{96\pi}{S^2}v) =$$

$$= \frac{48\pi}{S^3}u(x) - \frac{24\pi}{S^2}v$$

Logo,

$$\Phi'(x) = \frac{48\pi}{S^3}(\hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1}S + 2\pi x) - \frac{24\pi}{S^2}(\hat{f}_m - \hat{f}_{m-1})$$

Impondo $\Phi'(x) = 0$:

$$\begin{aligned}0 &= \frac{48\pi}{S^3}(\hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1}S) + \frac{96\pi^2}{S^3}x - \frac{24\pi}{S^2}(\hat{f}_m - \hat{f}_{m-1}) \\ \frac{96\pi}{S}x &= \frac{-48}{S}(\hat{\theta}_m - \hat{\theta}_{m-1} - \hat{f}_{m-1}S) + 24(\hat{f}_m - \hat{f}_{m-1}) \\ x &= \frac{1}{2\pi}(\hat{\theta}_{m-1} + \hat{f}_{m-1}S - \hat{\theta}_m) + \frac{S}{4\pi}(\hat{f}_m - \hat{f}_{m-1}) \\ x^* &= \frac{1}{2\pi}[(\hat{\theta}_{m-1} + \hat{f}_{m-1}S - \hat{\theta}_m) + \frac{S}{2}(\hat{f}_m - \hat{f}_{m-1})]\end{aligned}$$

Assim, conseguimos uma interpolação suave de amplitude, fase e freqüência entre os pontos que representam cada quadro. Se no intervalo entre os quadros m e $m-1$ tivermos F_m faixas, esse intervalo será reconstruído por:

$$\hat{x}_m(n) = \sum_{r=1}^{F_m} \hat{A}_m^r(n) \cos(\Theta_m^r(n))$$

em que $n = 0, ..., S-1$, $\hat{A}_m^r(n)$ é dado por (3.14) e $\Theta_m^r(n)$ é dado por (3.15), com os parâmetros que encontramos.

O som final é obtido concatenando-se os intervalos entre os quadros:

$$\hat{x}(t) = \hat{x}_m(n)$$

de forma que $t \in 0, ..., P-1$, e m, n são inteiros tais que $t = \frac{m-1}{S} + n$.

3.5 SMS

O SMS (Spectral Modeling Synthesis) é outro algoritmo de modelagem espectral para análise e síntese de sinais sonoros [20]. O SMS surge como uma evolução natural do PARSHL, sendo baseado nos mesmos princípios. Como as idéias e etapas que fundamentam esse método são muito parecidas, não nos aprofundaremos nos detalhes do SMS, e sim, nos focaremos em suas principais diferenças com relação ao PARSHL. Não implementamos o SMS em MATLAB. Utilizamos a implementação original elaborada por seus idealizadores [21]. A grande vantagem do SMS em relação aos modelos espectrais anteriores é a grande variadade de efeitos que ele permite aplicar aos sinais sonoros. Também não nos aprofundaremos muito nesses efeitos, mas os descreveremos brevemente de modo a abrir um leque de possibilidades do que se pode fazer com os modelos espectrais, incentivando futuros trabalhos que estudem mais detalhadamente esses efeitos.

3.5.1 O Modelo

No modelo usado pelo SMS, determinístico mais estocástico, o sinal é representado da seguinte maneira:

$$x(t) = \sum_{r=1}^{R} A_r(t) \cos(\theta_r(t)) + e(t) \tag{3.19}$$

Dessa forma, encontra-se a parte senoidal, ou determinística, de forma semelhante ao PARSHL e, posteriormente, subtrai-se essa informação do sinal

para encontrarmos a componente estocástica, ou residual. Essa componente pode conter sons percussivos, como o barulho de um dedo batendo em uma corda de violão, ou o martelo batendo nas cordas do piano.

3.5.2 Análise de Quadro

As etapas do SMS são muito parecidas com a do PARSHL. O primeiro passo é a aplicação da STFT no sinal a ser analisado e ressintetizado. A escolha dos parâmetros – tipo de janela, tamanho do quadro, tamanho da FFT e tamanho do salto – segue a mesma idéia que já discutimos. Uma única diferença é que o tipo de janela pode ser ajustado dinamicamente de acordo com as propriedades do espectro do sinal. A detecção dos picos do espectro de potência de cada quadro também é muito parecida, com uma leve diferença de implementação. No PARSHL, o máximo global é detectado e seus vizinhos são utilizados para traçar a parábola que determinará a localização do pico. Após isso, o morro referente a esse pico é retirado e procura-se o novo máximo global. No SMS, diferencia-se o espectro de potências para encontrar-se todos os máximos locais. Também há um parâmetro que determina o espaço mínimo entre dois máximos locais para que sejam considerados picos diferentes.

3.5.3 Detecção de Freqüência Fundamental

Após a detecção de picos, executa-se um procedimento não presente no PARSHL, mas que é de imensa importância na áera da música, que é a detecção da freqüência fundamental, ou altura. Por meio disso, para um som monofônico, pode-se descobrir qual a nota musical que está sendo processada. Isso pode ser aplicado em afinadores digitais de instrumentos, transcrições melódicas automáticas, entre outros.

Primeiro, é feita uma verificação da harmonicidade do som, pois em sons não harmônicos não há sentido a busca por uma freqüência fundamental. Ao contrário do que se possa pensar, a freqüência fundamental não é necessariamente a mais forte nem a mais grave presente em um sinal, podendo até não estar presente. Definiremos a freqüência fundamental como a freqüência que melhor divide as freqüências encontradas no quadro. Também não se deve esperar que essa divisão seja exata, pois além da presença de ruídos e erros numéricos, os próprios sons da natureza não são perfeitamente periódicos, influenciando na precisão das freqüências detectadas.

O SMS utiliza um método chamado "Two-Way Mismatch"para a detecção da freqüência fundamental, que busca a freqüência f_0 que minimiza o resto da divisão das freqüências dos picos encontrados por f_0.

3.5.4 Interligação de Picos

A etapa da análise entre quadros, mais especificamente a ligação dos picos de um quadro para outro, é onde aparece a maior diferença entre o PARSHL e o SMS. No PARSHL, devido à sua modelagem exclusivamente senoidal, todos os picos detectados, ou pelo menos uma quantidade especificada, são obrigatoriamente associados a uma faixa. As faixas são criadas e destruídas dinamicamente com base na existência ou não de freqüências dentro de seu alcance. No SMS, a modelagem determinística mais estócastica requer que a parte determinística detecte somente as partes mais estáveis do som, ou seja, encontre componentes senoidais bem definidas e com continuidade. O algoritmo de ligação de picos anterior não garante que isso aconteça, pois uma faixa pode se afastar rapidamente de sua freqüência inicial, disputando um pico que deveria pertencer a outra faixa.

Na etapa da detecção da freqüência fundamental, é feita uma análise da harmonicidade do som. Portanto, na ligação de picos consideram-se dois casos: sons harmônicos e sons não harmônicos.

Para sons não harmônicos, o procedimento é muito semelhante ao do PARSHL: as faixas são criadas e destruídas dinamicamente, conforme o surgimento de novas freqüências. Porém, após as ligações serem completadas, conhecendo-se o comportamento futuro das faixas, descartam-se faixas de comprimento muito curto e preenchem-se faixas com leves descontinuidades. Entretanto, esse procedimento não pode ser aplicado no caso de processamento em tempo real, pois não se tem o conhecimento do comportamento futuro de cada faixa.

Para sons harmônicos e monofônicos, detecta-se a freqüência fundamental e, a partir de seus harmônicos, criam-se guias para as faixas. Assim, as faixas não são criadas pelos picos presentes, ou seja, pelo comportamento encontrado, e sim, pelo comportamento esperado para um sinal monofônico e harmônico. As faixas serão, portanto, as parciais ou harmônicos do som; os picos que não são associados a nenhum desses harmônicos são descartados e considerados como componentes estocásticos.

3.5.5 Obtenção da Componente Estocástica

Após a ligação dos picos, tem-se a parte determinística do sinal bem definida. Pode-se agora sintetizá-la e subtraí-la do sinal original para a obtenção de sua componente estocástica. Recomenda-se a normalização da amplitude da componente determinística sintetizada, pois uma diferença de volume em relação ao sinal original resultaria em um resíduo com senóides, cuja amplitude seria essa diferença. Assim, com a normalização do volume, as com-

ponentes senoidais são todas anuladas, resultando apenas a componente estocástica desejada. Esse procedimento nos fornece a componente estocástica no domínio do tempo. Por outro lado, podemos também obtê-la no domínio das freqüências, subtraindo-se o espectro da componente determinística do espectro original, não havendo, assim, necessidade da síntese das senóides. Após isso, é feita uma análise do espectro da componente residual, para o estudo de seu comportamento geral, o que será usado para ressintetizá-lo posteriormente. Tal comportamento pode ser obtido através de uma interpolação suave entre os máximos locais do espectro.

3.5.6 Síntese

O SMS prioriza a síntese por sobreposição ao invés da síntese aditiva, devido à primeira ser muito mais rápida, por utilizar a FFT inversa. As modificações desejadas sobre o som são aplicadas diretamente sobre seu espectro. A própria adição de senóides é feita gerando-se seus morros no espectro. Essa é uma grande diferença entre o SMS e o PARSHL. A síntese da parte estocástica do sinal também é uma tarefa nova a ser realizada. Isso pode ser feito de duas formas: uma delas é simplesmente adicionar o espectro das amplitudes e o espectro das fases obtido na análise ao espectro sintetizado da componente determinística; outra forma é modelar o ruído como uma convolução de ruído branco por um filtro variável no tempo, dado pelo comportamento geral do espectro da componente estocástica. Na prática, mantém-se esse comportamento como espectro de amplitude e gera-se um espectro de fase aleatório. Somam-se os espectros de amplitude e fase da componente determinística e estocástica para obter-se o espectro sintetizado. Após isso, aplica-se a FFT inversa para gerar o som final sintetizado no domínio do tempo.

3.5.7 Efeitos e Transformações

Após a etapa de análise e antes da etapa de síntese, pode-se aplicar efeitos ou transformações, visando modificar-se o som. Citaremos aqui algumas transformações que podem ser realizadas pelo SMS, porém, não as explicaremos detalhadamente (para mais detalhes, ver [26]). Nosso objetivo é expor ao leitor um pouco dos efeitos e aplicações que podem ser alcançados através dos métodos de análise e síntese espectral, e despertar o interesse em estudos mais aprofundados sobre essas transformações.

Filtros e Equalização

Filtros são muito importantes em diversas áreas tecnológicas. Através deles, pode-se detectar a intensidade de uma faixa específica de freqüências e extraí-las do som. Isso pode ser facilmente implementado através de convoluções ou modificações diretas no espectro do sinal.

Mudança de Velocidade

A mudança de velocidade de um som sem alteração de suas freqüências é uma transformação cuja importância, aplicação e implementação já discutimos na seção sobre o algoritmo Phase Vocoder. O SMS tira proveito da modelagem determinística mais estocástica para refinar essa transformação. A parte determinística deve ser estendida no tempo da forma como já fizemos anteriormente com o phase vocoder. Porém, isso deve ser evitado na parte estocástica para que ela não perca sua característica transiente. Portanto, o SMS separa essas componentes, reescala a componente senoidal e mantém as características da componente estocástica, sendo necessária apenas a sincronização entre ambas.

Mudança de Freqüências

A mudança de freqüências é outra transformação que já estudamos aqui. No SMS ela é feita diretamente em seu espectro, podendo ser feita de duas formas: a primeira delas é uma mudança uniforme, ou seja, todas as freqüências são reescaladas pelo mesmo fator, correspondendo musicalmente a uma mudança de tom ou transposição; a segunda é uma mudança de freqüências heterogênea, ou seja, cada freqüência do sinal pode ser reescalada por um fator diferente.

Harmonização

A harmonização é implementada utilizando-se o efeito anterior para transpor uma melodia e somar o resultado ao sinal original. Com isso, cria-se o efeito de dois instrumentos tocando a mesma melodia em diferentes tons, geralmente um intervalo harmônico fixo.

Discretização à Escala Temperada

Discretização à escala temperada significa transpor cada freqüência determinística encontrada à freqüência mais próxima que corresponde a uma nota musical na escala temperada que conhecemos. Uma aplicação disso

é a afinação automática de uma melodia tocada ou cantada, ou seja, essa transformação permite a correção de desafinações durante a gravação de uma música, por exemplo.

Vibrato e Trêmolo

Vibrato e trêmolo são dois efeitos muito comuns em execuções de peças musicais. Eles correspondem a variações pequenas e contínuas no som, sendo o vibrato uma variação de freqüência e o trêmolo, uma variação de amplitude.

3.6 Testes e Resultados

Nesta seção, apresentaremos os resultados obtidos nos testes realizados com os algoritmos baseados na modelagem espectral. Os algoritmos que apresentamos foram: Phase Vocoder, PARSHL e SMS. Por sua simplicidade, o Phase Vocoder foi acoplado ao PARSHL, sendo utilizado apenas para modificações de tempo ou de freqüência e ressíntese por sobreposição, deixando toda a parte de análise do sinal original ao PARSHL.

Começaremos fazendo um simples teste de ressíntese. Seguindo o modelo dos testes realizados com os algoritmos de alta resolução, utilizaremos a STFT para dividirmos o som, obter o espectro de cada quadro e depois ressintetizá-los e uni-los, gerando o som original via síntese por sobreposição. A utilização da FFT para a obtenção do espectro de cada quadro e sua posterior ressintetização por meio da FFT inversa tornam o processo extremamente rápido e eficiente.

Os parâmetros da STFT utilizados foram: tamanho do quadro $M = 511$, tamanho do salto $R = 256$, tamanho da FFT $N = 1024$ e janela de Hanning. Vemos na Figura 3.14 o erro obtido através desse método. O erro é da ordem de 10^{-16}, ou seja, praticamente desprezível.

Utilizando os métodos de alta resolução para essa mesma tarefa, obtivemos erros da ordem de 10^{-3} e o tempo de processamento foi da ordem de minutos, ou até mesmo horas. Pela síntese por sobreposição, o erro foi da ordem de 10^{-16} e o tempo de processamento foi em torno de 40ms.

Faremos agora um novo teste, simplesmente substituindo a síntese por sobreposição pela síntese aditiva. Ainda não serão realizadas etapas de análise, como detecção e ligação de picos. Considera-se que cada bin do espectro de freqüências é uma faixa e define-se a fase instantânea pela integral da freqüência.

A desvantagem da síntese aditiva fica evidente tanto no erro quanto no tempo de processamento. Enquanto a STFT com síntese por sobreposição

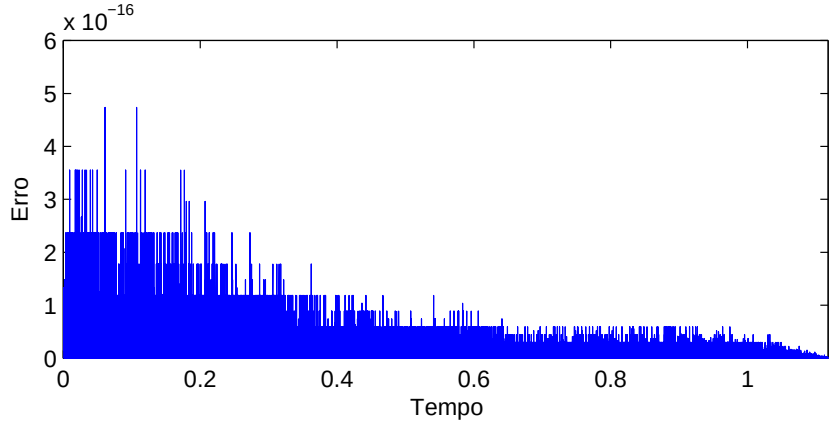

Figura 3.14: Erro da nota A4 do piano com síntese por sobreposição, via STFT.

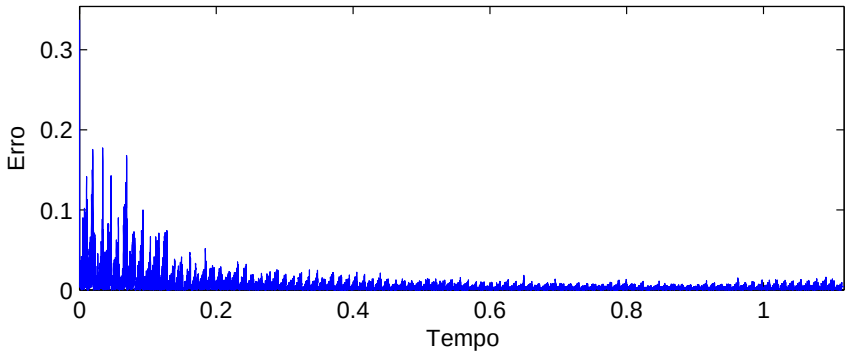

Figura 3.15: Erro da nota A4 do piano com síntese aditiva, via STFT.

teve tempo de processamento da ordem de milissegundos, a síntese aditiva levou em torno de 10.1 segundos. O erro relativo também foi alto em relação à síntese por sobreposição e até mesmo em relação aos métodos de alta resolução. Porém, muitas vezes, esse tipo de erro não é perceptível auditivamente. Além disso, os métodos que testaremos em seguida não produzem um sinal ressintetizado idêntico ao original, pois escolhe-se um certo número de freqüências a serem ressintetizadas. No entanto, os resultados assemelham-se muito aos originais. Isso ocorre pois nosso sistema auditivo não interpreta diferenças de fases. Assim, se um conjunto de senóides sobrepostas difere de outro apenas pelas fases iniciais de cada uma, o resultado sonoro será o mesmo, apesar dos sinais finais serem funções diferentes. Nesse caso, o erro não é a melhor forma de avaliarmos a semelhança entre o sinal sintetizado e o original. Portanto, apresentaremos um tipo de gráfico muito utilizado para análise e comparação entre sinais: o espectrograma.

O espectrograma é um gráfico tridimensional adaptado para duas dimensões, em que o eixo x representa o tempo, o eixo y, as freqüências, e o eixo z, as magnitudes, sendo seus valores expressados por cores. Utilizaremos um padrão em que cores frias (azul, verde) representam baixas magnitudes, e cores quentes (amarelo, laranja, vermelho) representam altas magnitudes.

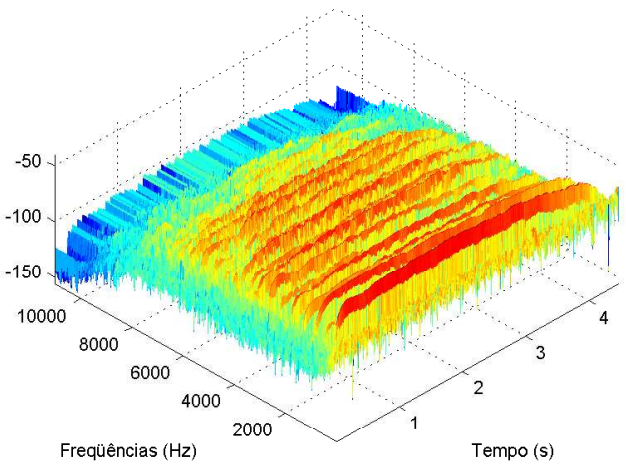

Figura 3.16: Espectrograma tridimensional de uma nota de violino.

Apresentaremos, então, o espectrograma para a nota A4 do piano, a qual utilizamos em todos os testes até agora, junto com o espectro do sinal ressintetizado pela síntese aditiva. Na Figura 3.17, aparecem todas as freqüências representáveis no espectro de um sinal digital, ou seja, até $\frac{f_s}{2}$. Como, para

esse sinal $f_s = 22050$Hz, temos freqüências até 11025Hz. Já na Figura 3.18, focamos as principais freqüências presentes nesse sinal específico.

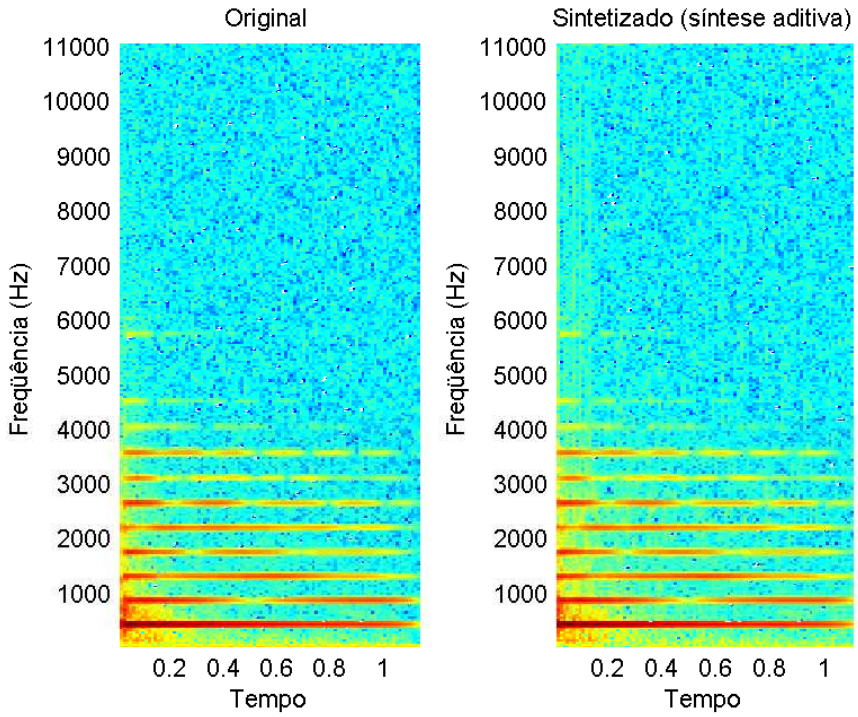

Figura 3.17: Espectrograma da nota A4 do piano.

Esse tipo de gráfico é muito útil para observarmos como o som se comporta. Nesse caso, temos uma nota de piano com freqüência fundamental 440Hz e podemos visualizar como se comportam seus harmônicos, ou seja, vemos a intensidade das freqüências múltiplas de 440Hz. Observa-se também uma região de maior intensidade no começo do sinal, que corresponde ao ruído gerado pelo ataque à nota.

Vamos agora prosseguir com os testes do algoritmo PARSHL, que consiste em 3 etapas: STFT com detecção de picos, definição das faixas através da ligação entre picos e síntese aditiva com interpolação cúbica de fase. Além dos quatro parâmetros anteriores, são necessários mais três: número de picos a serem detectados por quadro, número de picos a serem sintetizados por quadro e Δf, que representa a diferença máxima entre duas freqüências de quadros consecutivos para que elas possam ser associadas à mesma faixa. Utilizaremos $M = 511$, $R = 256$ e $N = 1024$.

O número de picos a serem detectados e sintetizados depende muito da complexidade do sinal a ser analisado. Vamos ver, por exemplo, o que acon-

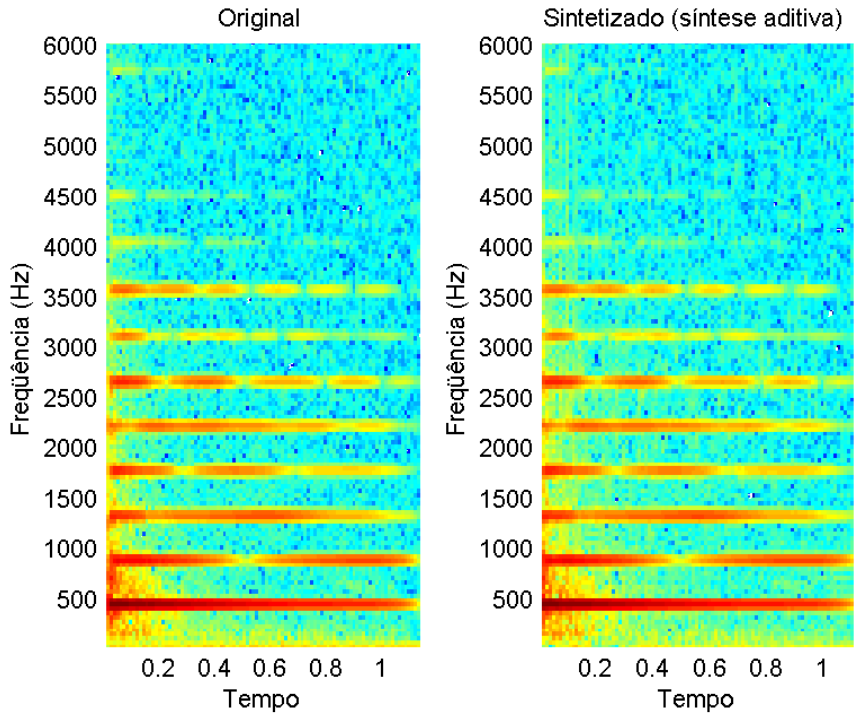

Figura 3.18: Espectrograma da nota A4 do piano, até 6000Hz.

tece se optarmos por sintetizarmos apenas 4 picos para cada quadro no som da nota A4 do piano.

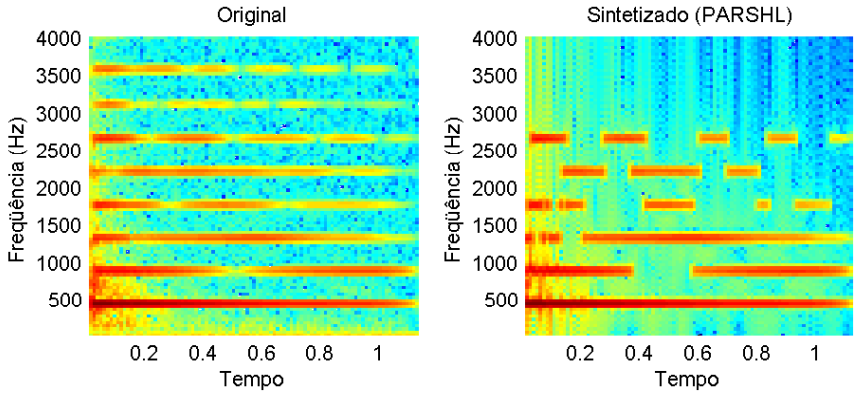

Figura 3.19: Nota A4 do piano com apenas 4 faixas simultâneas.

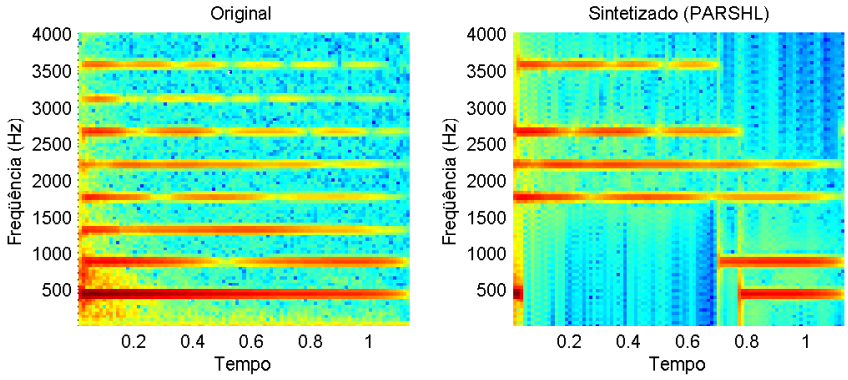

Figura 3.20: Nota A4 do piano com 4 faixas simultâneas e 10 picos por quadro.

A Figura 3.19 apresenta o resultado da detecção de 4 picos por quadro e a síntese da mesma quantidade de faixas. Percebe-se a grande descontinuidade das faixas sintetizadas, gerando muitas oscilações bruscas no som. Na Figura 3.20, detectamos 10 picos por quadro, mas sintetizamos apenas 4. Assim, priorizam-se as faixas já existentes, mantendo sua continuidade por mais tempo, gerando um som mais estável. Porém, nota-se que 4 é um número de faixas muito pequeno para a reconstrução desse som. Agora, aumentaremos o número de faixas progressivamente para vermos as modificações do espectro. A partir de 15 faixas simultâneas vemos que as características mais importantes do som já são bem representadas. Com o aumento no número

de faixas, características menos relevantes vão sendo adicionadas ao som e o espectro vai se completando, se aproximando do som original.

Faixas/Picos	STFT	Def. Faixas	Síntese aditiva	Total
10/20	0.232s	0.136s	2.814s	3.182s
20/30	0.236s	1.764s	6.727s	8.727s
30/50	0.319s	6.074s	10.165s	10.558s
50/80	0.454s	11.770s	15.190s	27.414s
100/150	0.499s	44.744s	27.021s	72.264s

Tabela 3.2: Tempo de processamento para o som do piano (24646 pontos).

Vamos agora utilizar o PARSHL em outros sinais. Primeiro, utilizaremos o som da nota A4 (440Hz) tocada em uma flauta. O sinal tem freqüência de amostragem também de 22050Hz e duração de 1,7 segundos, totalizando 37485 pontos. A Figura 3.22 mostra o espectrograma do sinal original e sintetizado com 20, 50 e 100 faixas.

Outro som que testaremos é o som de um violino, tocando também a nota A4. Esse som tem freqüência de amostragem também de 22050Hz e duração de 4.45 segundos, totalizando 98160 pontos. A Figura 3.23 mostra o espectrograma do sinal original e sintetizado com 20, 50 e 100 faixas. A Tabela 3.3 e a Figura 3.24 mostram o tempo de processamento para esse arquivo, para cada etapa e em sua totalidade.

Faixas/Picos	STFT	Def. Faixas	Síntese aditiva	Total
20/30	1.144s	40.158s	23.664s	1 min 5s
50/80	1.83s	10min 26.9s	1min 3.8s	11min 32s
100/150	2s	1h 00min 18.9s	1min 47.9s	1h 2min 8.8s

Tabela 3.3: Tempo de processamento para o som do violino (98160 pontos).

Até agora, todos os testes que fizemos com os diversos algoritmos apresentados foram realizados apenas para sons muito simples, formados apenas por uma única nota de um único instrumento. Faremos agora um teste com um som muito mais complexo, desta vez polifônico, com diferentes melodias em guitarra, baixo, teclado, voz e até mesmo bateria, que é um instrumento percussivo. Como a música é muito longa (1352583 pontos, totalizando 61.24 segundos), vamos tomar pequenos trechos dela. Na Figura 3.25, temos o espectro do segundo inicial desse sinal, em que ainda não há voz. Já na Figura 3.26, temos um trecho seguinte de 4 segundos, em que já aparecem todos os instrumentos. Para esse caso específico, utilizaremos $M = 1023$, $N = 2048$ e $R = 512$.

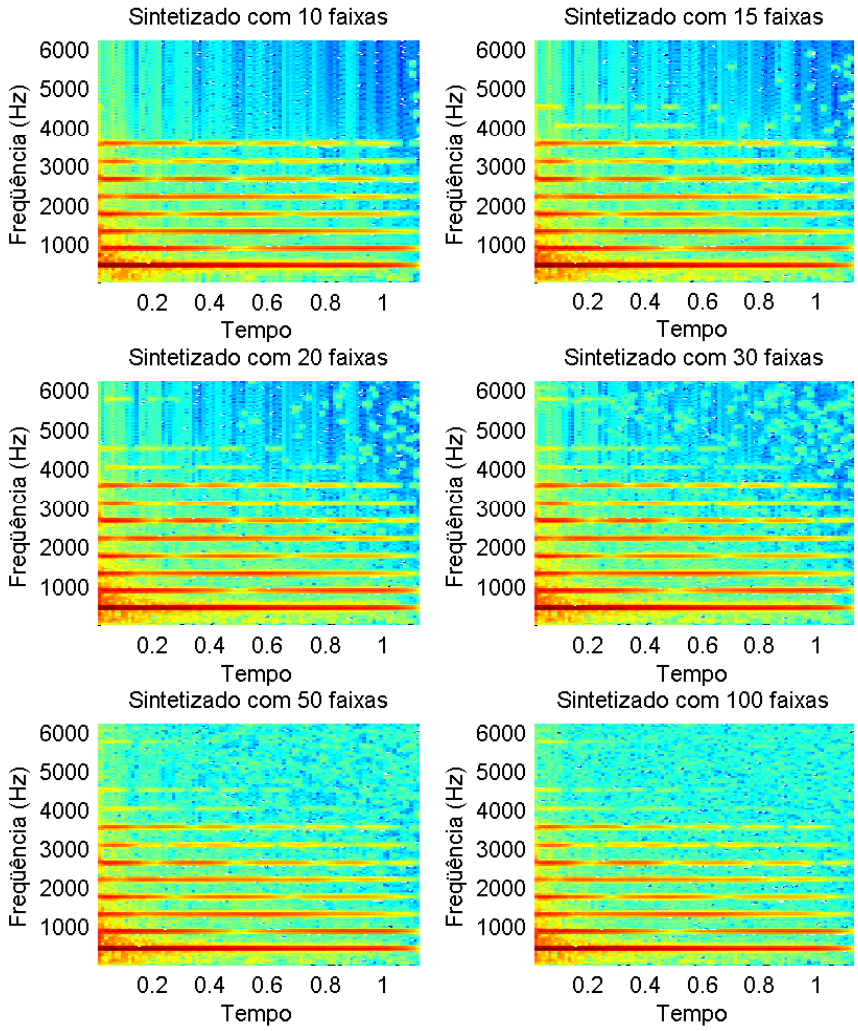

Figura 3.21: Nota A4 do piano com número de faixas variando entre 10 e 100.

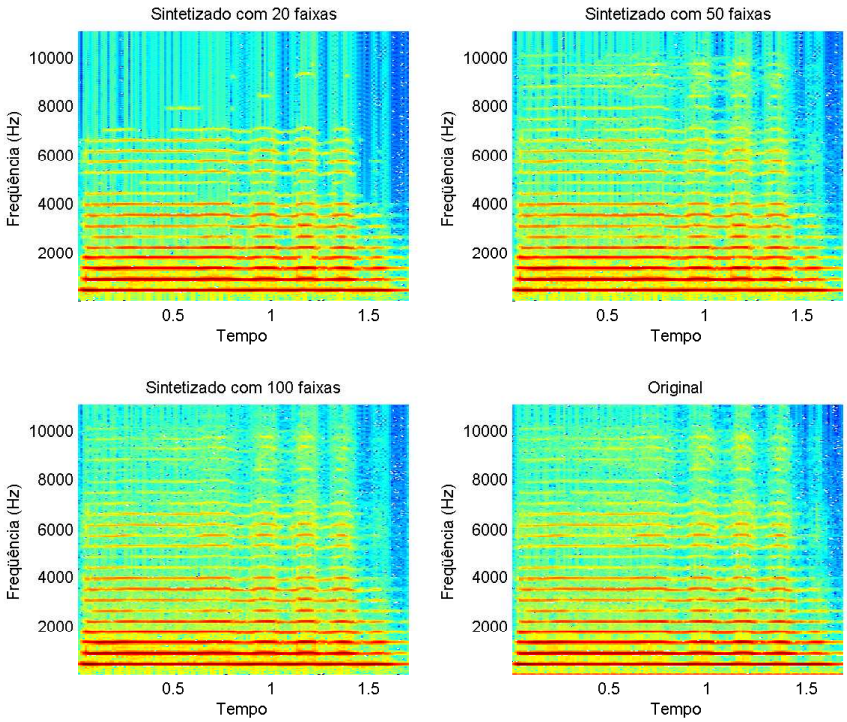

Figura 3.22: Espectro da nota A4 (440Hz) em uma flauta.

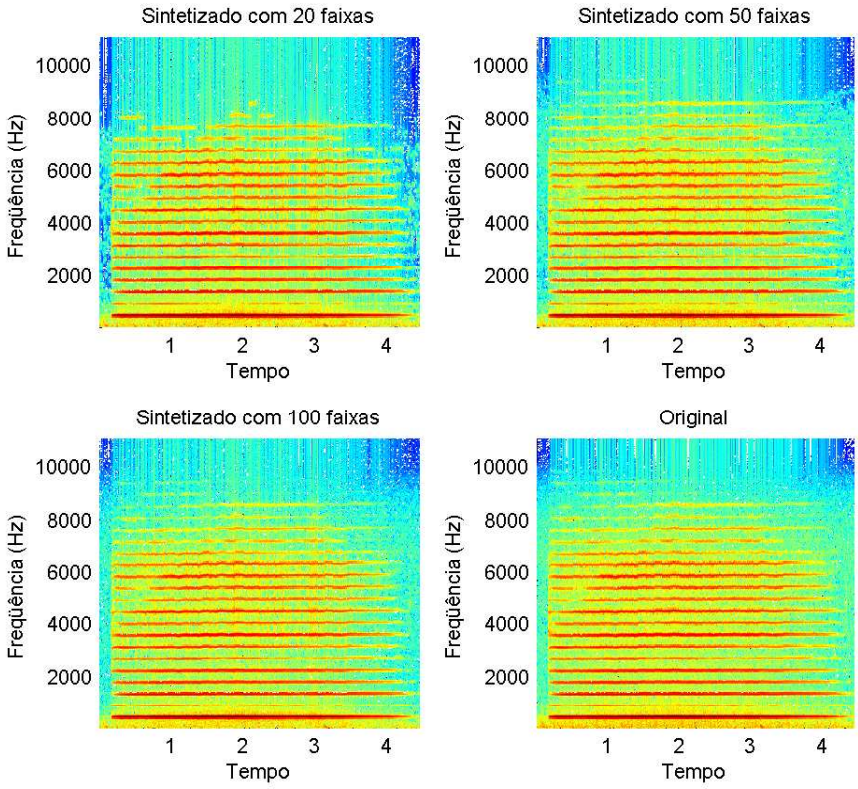

Figura 3.23: Espectro da nota A4 (440Hz) em um violino.

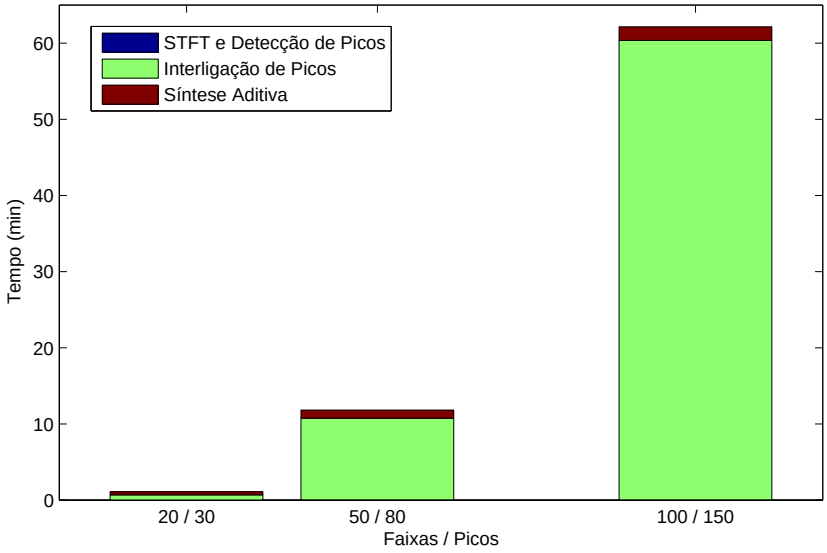

Figura 3.24: Tempo de processamento para o som do violino (98160 pontos).

O Algoritmo PARSHL se comporta bem para sons complexos, como uma música polifônica. Porém, deve-se aumentar o número de picos a serem detectados e o número de faixas para ressíntese, pois sons polifônicos são compostos de muitas parciais. Um número baixo de faixas resultará em pouca continuidade e muitas faixas de curta duração, que ficam descaracterizadas no som sintetizado. Em um sinal ressintetizado com poucos picos, ocorre um efeito semelhante ao de um som de baixa qualidade transmitido em tempo real pela internet. Devido às restrições de velocidades de conexão, o som transmitido geralmente é de qualidade inferior, com perda de informações, o que também acontece quando se sintetiza o sinal com poucas faixas.

Faremos agora alguns testes com o método SMS. Na Figura 3.27, vemos como o sinal correspondente à nota A4 do piano é dividido em componentes determinísticas e estocásticas. Percebe-se que a componente determinística detecta bem as características harmônicas do som, ou seja, as parciais estáveis múltiplas da freqüência fundamental. Já a componente estocástica sofre algumas interferências da componente determinística. O tempo de processamento da nota A4 do piano pelo Algoritmo SMS foi: 20.7s para 30 picos e 20 faixas; 127.7s para 80 picos e 50 faixas; e 188.2s para 150 picos e 100 faixas.

A Figura 3.28 exibe a mesma separação de componentes para uma nota de violino, enquanto a Figura 3.29 o faz para um som de flauta. As Figuras 3.30 e 3.31 ilustram, respectivamente para esses sinais, a ligação entre picos na

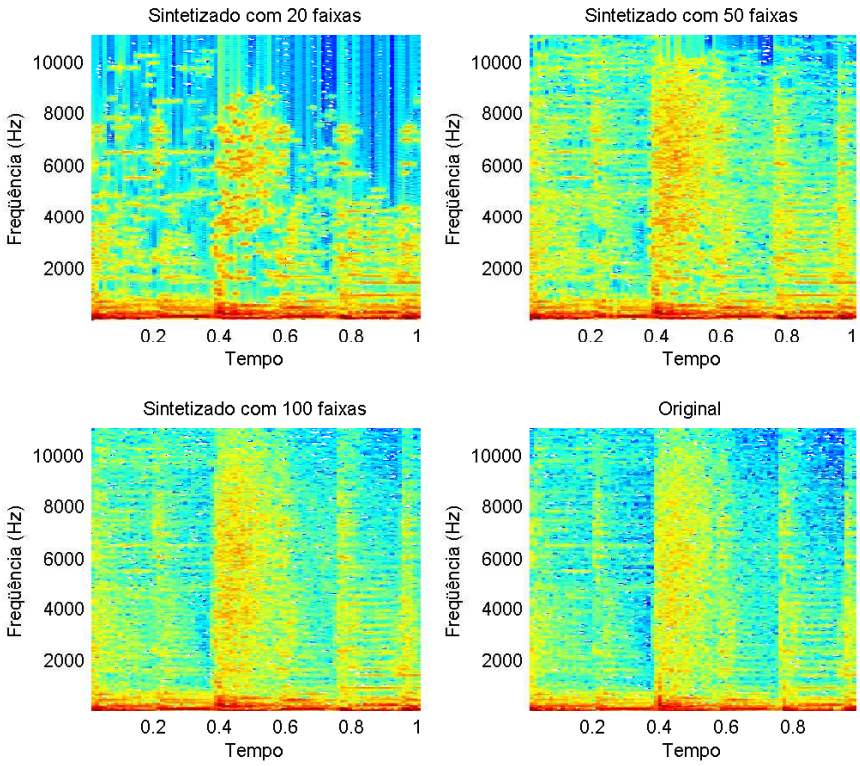

Figura 3.25: Espectro de uma música polifônica, durante 1 segundo.

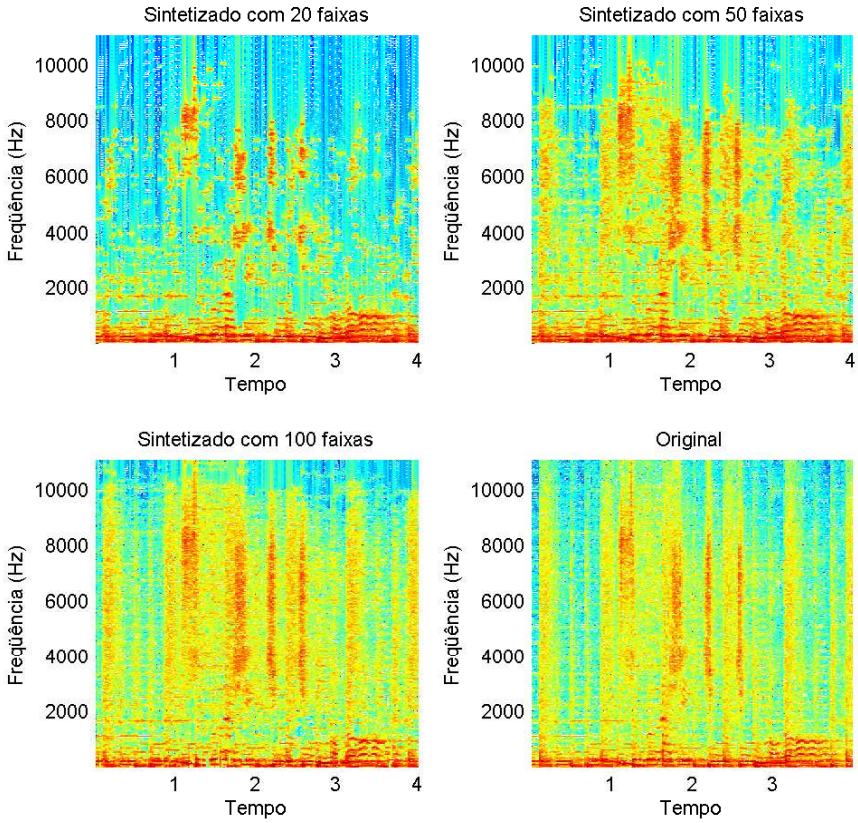

Figura 3.26: Espectro de uma música polifônica, durante 4 segundos.

seqüência de quadros. A linha escura mais grossa representa a freqüência fundamental. As outras parciais são associadas aos múltiplos dessa freqüência, de forma a detectarem-se apenas as componentes estáveis do som. Em alguns pontos, vê-se que a detecção da freqüência fundamental não obteve êxito. Nesses pontos, a tentativa de encaixar as freqüências detectadas nas faixas estabelecidas pelas supostas parciais ocasionou uma irregularidade no sinal. Esse não foi o caso no começo do sinal do violino, que corresponde a um ataque. Percebe-se também que não há irregularidades no começo do sinal da flauta, pois se trata de um instrumento de sopro, não caracterizado por um ataque. O tempo de processamento da nota A4 do violino foi: 86.1s para 30 picos e 20 faixas; 29.8s para 80 picos e 50 faixas; e 44s para 150 picos e 100 faixas.

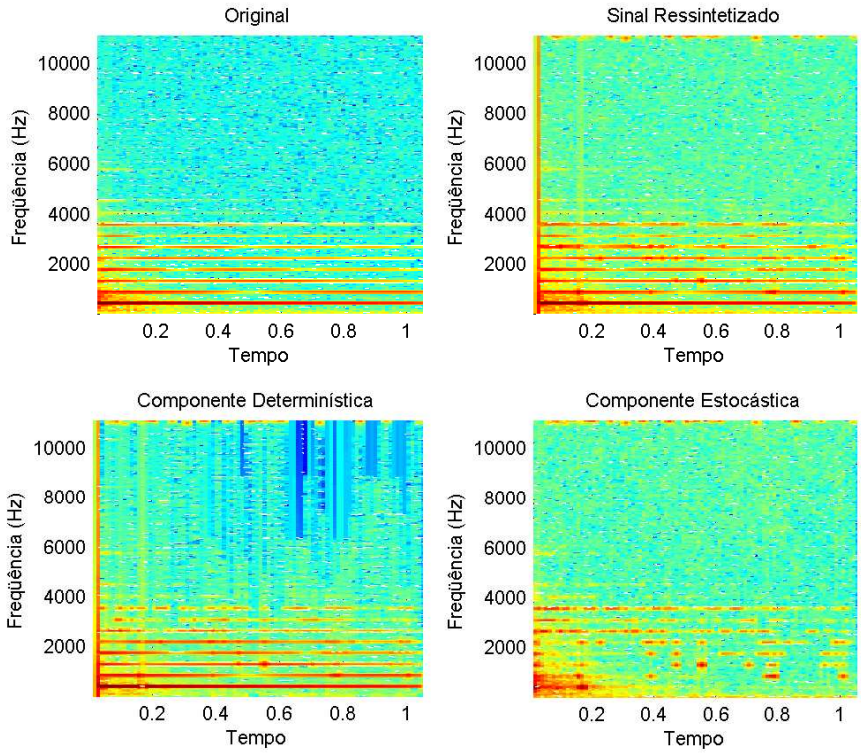

Figura 3.27: Espectro da nota A4 do piano, com divisão entre componentes determinística e estocástica.

Aplicaremos agora uma modificação ao som antes da ressíntese. Primeiro, vamos modificar a velocidade da música sem alterarmos seu tom, ou seja, suas freqüências. Depois alteraremos as freqüências sem modificarmos a veloci-

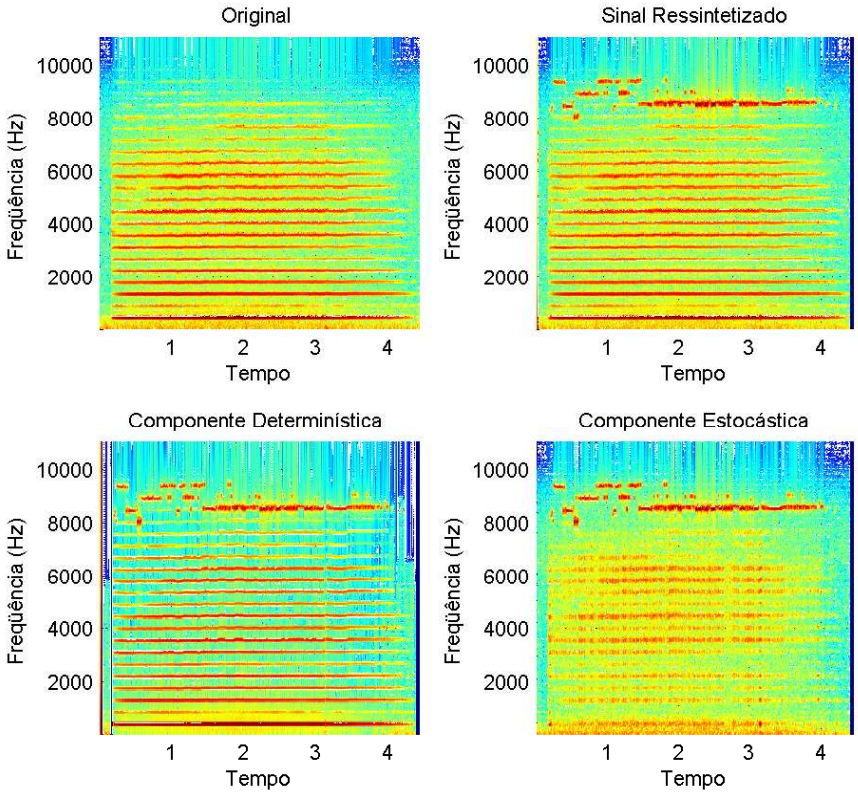

Figura 3.28: Espectro da nota A4 de um violino, com divisão entre componentes determinística e estocástica.

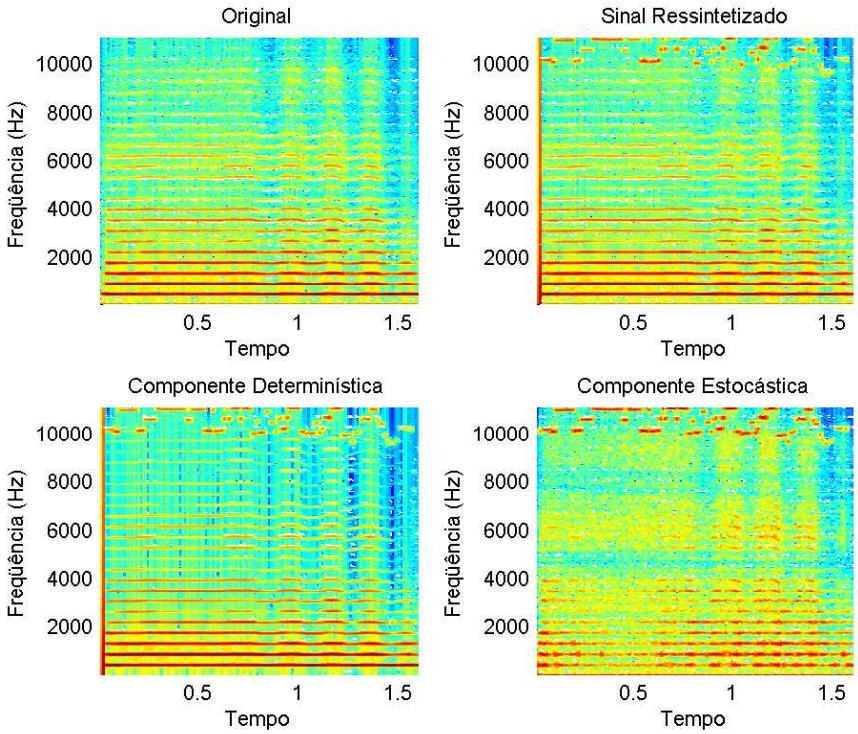

Figura 3.29: Espectro da nota A4 de uma flauta, com divisão entre componentes determinística e estocástica.

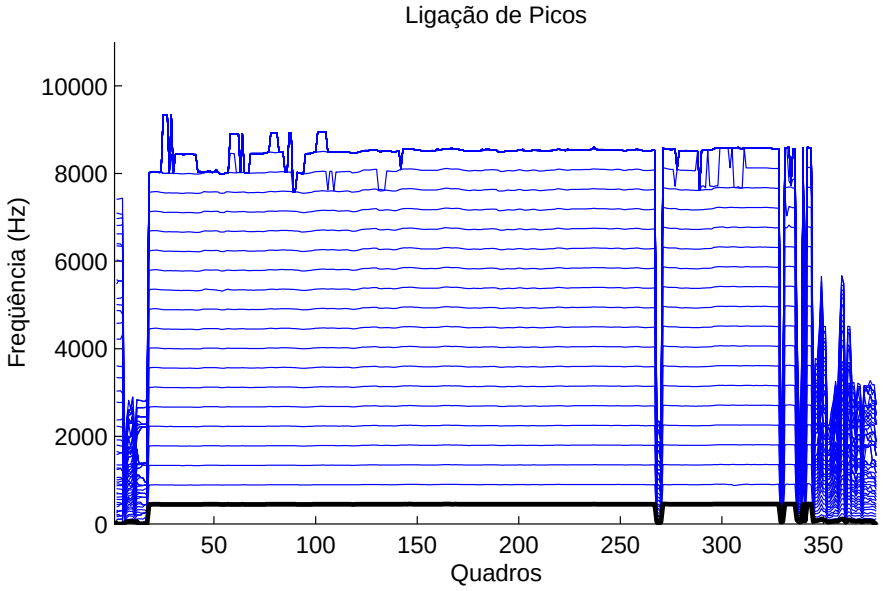

Figura 3.30: Interligação de picos para o sinal de um violino.

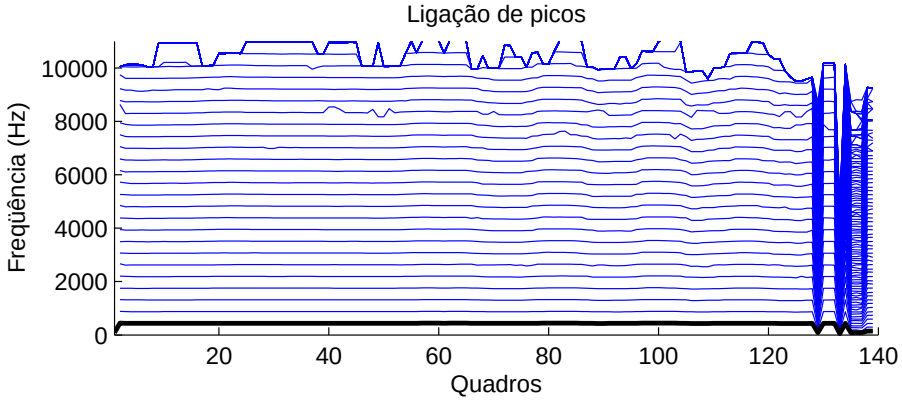

Figura 3.31: Interligação de picos para o sinal de uma flauta.

dade. Para isso, utilizaremos o algoritmo Phase Vocoder [†].

A Figura 3.32 mostra o espectro de uma música polifônica durante um período de 6 segundos. A duração da música é multiplicada ou dividida por um fator de $\frac{3}{2}$, gerando sinais de duração 9s e 4s, respectivamente. Vê-se na figura como os espectros são muito semelhantes, diferindo apenas na escala temporal.

Para testarmos a mudança de freqüências sem alteração de velocidade, vamos usar outro sinal. No espectro de um sinal polifônico essa mudança é mais difícil de ser visualizada, porém funciona igualmente bem. Multiplicaremos e dividiremos as freqüências de dois sons, do violino e do piano, pelo valor $\frac{3}{2}$. Vê-se que, ao multiplicar-se as freqüências por esse fator, as parciais (no espectro) se afastam umas das outras, aumentando a freqüência fundamental e, logo, a altura do som. Quando dividem-se as freqüências, o processo inverso ocorre: as parciais se aproximam umas das outras, diminuindo a altura do som.

[†]implementado via função pvsample obtida em [11]

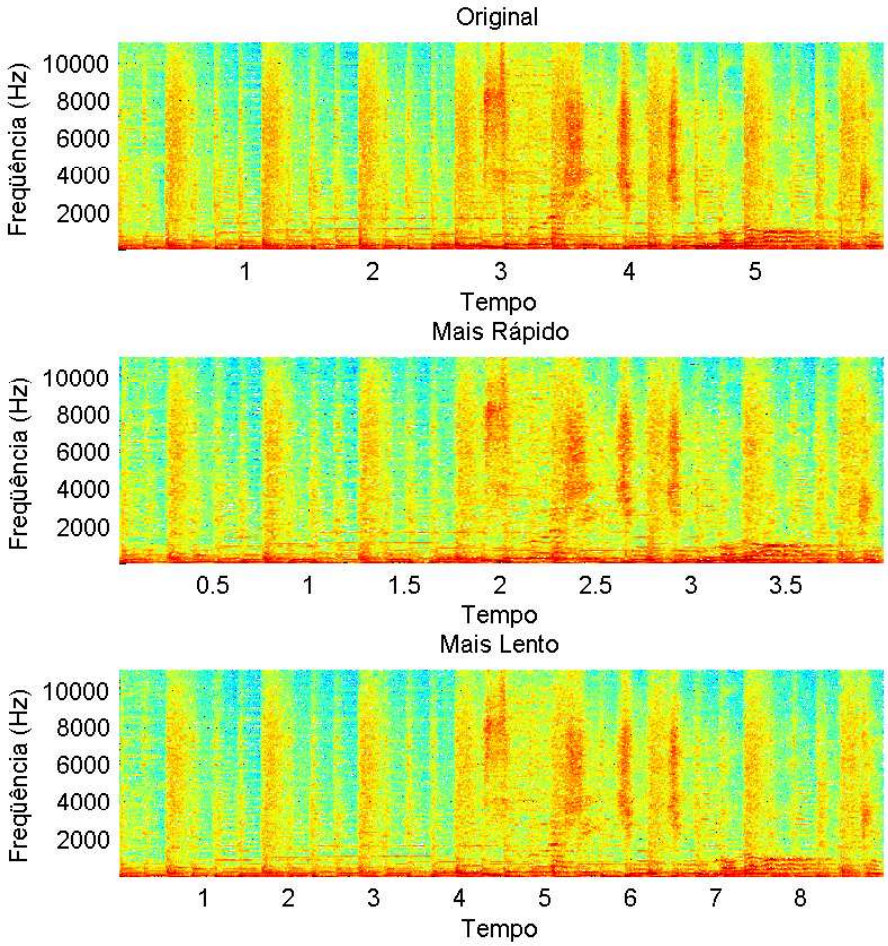

Figura 3.32: Espectro de uma música polifônica, com duração primeiro dividida e, depois, multiplicada pelo fator $\frac{3}{2}$.

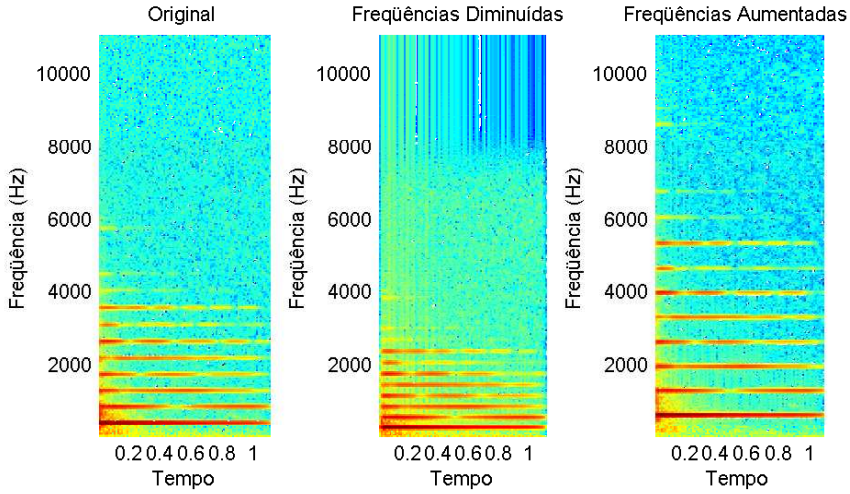

Figura 3.33: Espectro da nota A4 do piano, com freqüências primeiro divididas e, depois, multiplicadas pelo fator $\frac{3}{2}$.

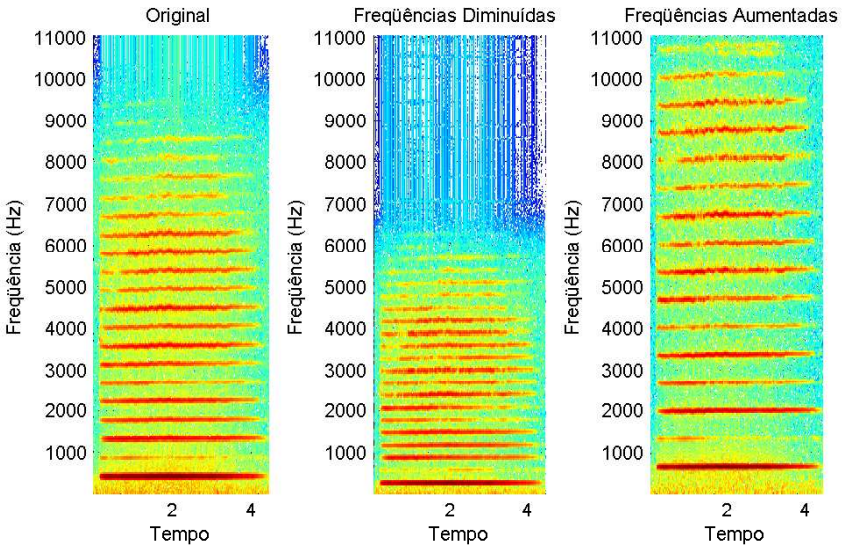

Figura 3.34: Espectro da nota A4 do piano, com freqüências primeiro divididas e, depois, multiplicadas pelo fator $\frac{3}{2}$.

Considerações Finais

Neste trabalho, estudamos duas famílias de métodos para análise e ressíntese de sinais musicais: os métodos baseados no domínio do tempo e os métodos baseados no domínio das freqüências.

Os métodos baseados no domínio do tempo que apresentamos aqui correspondem ao estado atual dessa abordagem em processamento de sinais. Pudemos fornecer nossa contribuição a essa família de métodos através de algumas novas maneiras de obtermos alguns parâmetros e de realizarmos alguns cálculos. A desvantagem desses métodos é o número de operações necessárias para realizá-los que, por ser muito alto, demandam muito tempo de processamento, em comparação aos algoritmos no domínio das freqüências. Esse fato inviabiliza sua utilização em aplicações de tempo real em computadores atuais. Apesar disso, os testes que realizamos mostram que os métodos de alta resolução possuem bons resultados, com erros inferiores aos dos métodos no domínio das freqüências que utilizam síntese aditiva.

Um método pioneiro baseado no domínio das freqüências para a ressíntese de sinais musicais foi o Phase Vocoder. Esse método não conta com uma etapa de análise. Para cada quadro, o espectro é obtido via STFT, porém não é interpretado, é apenas utilizado para a ressíntese. Na síntese aditiva, cada bin do espectro é interpretado como uma faixa de freqüência fixa, o que torna esse método pouco flexível e com desempenho inferior para sons com oscilações de freqüências. A utilização desse algoritmo, aqui, foi restrita à modificação independente de duração e freqüências de um sinal musical. Os métodos de análise e ressíntese no domínio das freqüências, que foram apresentados aqui, são o PARSHL e o SMS. O PARSHL foi um dos primeiros algoritmos baseados na STFT para a análise e síntese de sinais musicais. Seu estudo nos forneceu uma base de conhecimento que fundamenta alguns princípios das transformações sonoras modernas. Porém, ainda há muito a ser estudado nessa área. Vimos que o algoritmo SMS surgiu como uma evolução do PARSHL. Além de refinar suas etapas básicas de análise e síntese, o SMS torna possível a utilização de diversos efeitos sonoros na ressíntese.

Nossa pesquisa dos métodos de alta resolução iniciou-se com o algoritmo

ESPRIT (Algoritmo 1). Em [2], sugere-se que o cálculo das amplitudes, isto é, a resolução do problema $W\alpha = He_1$ (equação (2.9)), em que a matriz W é de Vandermonde, pode ser feita sem a necessidade de se explicitar a matriz de Vandermonde. Neste trabalho, deduzimos uma forma original para essa resolução, que é a nossa equação (2.11).

Em [4], há um modo de se calcular adaptativamente a matriz espectral Φ_k nos casos em que os subespaços, dados pelas colunas ortonormais das matrizes U_k, tenham uma atualização de posto 1 a partir de uma equação do tipo $U_k = U_{k-1} + eg^T$. Fizemos uma generalização desse algoritmo ao nosso caso, uma vez que em nossos algoritmos adaptativos, os subespaços que queríamos rastrear eram atualizados por uma equação do tipo $U_k = U_{k-1} + EG^T$, em que E e G são matrizes de posto 2. Por sua vez, os nossos algoritmos adaptativos foram generalizações dos algoritmos NP3 e OPAST aos casos em que as matrizes C_k tinham atualizações de posto 2, em vez de posto 1, como pensado originalmente. Os algoritmos [13] e [1] foram desenvolvidos originalmente para rastrear subespaços dominantes de matrizes de covariância C_k, cuja atualização é de posto 1. No nosso caso, as matrizes C_k eram definidas por $C_k = H_k^2 = C_{k-1} + r_k \hat{r}_k^t$ (equação 2.17). Todas essas adaptações originaram alguns problemas matemáticos de pequeno porte, que foram resolvidos nessa dissertação. Por exemplo, foi necessário enunciar o Lema 2.4 para concluir que a matriz BA da equação 2.19 era diagonalizável.

As implementações apresentadas aqui foram feitas em MATLAB e, com exceção dos algoritmos Phase Vocoder e SMS, os códigos são todos de nossa autoria. Para o algoritmo Phase Vocoder foi utilizada a implementação feita em [11]. Já os testes com o método SMS foram todos feitos utilizando o código original, disponível em [21]. Em nossa pesquisa do PARSHL, não encontramos nenhuma implementação pública desse algoritmo, e fizemos sua implementação, com base no artigo [19].

Após estudarmos e realizarmos testes com os métodos no domínio do tempo e os métodos no domínio das freqüências, concluimos que ambas as abordagens possuem vantagens e desvantagens. Assim, não podemos dizer que exista um melhor método para análise e ressíntese de sinais sonoros. A escolha do método adequado dependerá das características da aplicação para o qual este será utilizado. Para sons monofônicos, com poucas variações e que se encaixem no modelo senoidal com amortecimento senoidal, os métodos de alta resolução apresentaram melhores resultados. Também recomendaríamos esses métodos para reconhecimento de locutor, por exemplo, no qual é necessária uma representação mais fiel, ou seja, um resultado numérico mais preciso. Por outro lado, os métodos baseados no domínio das freqüências apresentaram melhores resultados com sons mais complexos, nos quais as imperfeições da ressíntese passam desapercebidas pelos ouvidos humanos.

Isso se deve à maior quantidade de informações a serem decodificadas pelo sistema auditivo. Além disso, esses métodos também são mais recomendados para a aplicação de transformações e efeitos na ressíntese de sinais, por oferecerem maior flexibilidade no domínio das freqüências. A utilização dos métodos no domínio do tempo em outras aplicações, como reconhecimento de voz, por exemplo, e o estudo mais aprofundado dos efeitos que podem ser aplicados a sinais sonoros via métodos no domínio de freqüências surgem como possíveis continuações deste estudo, as quais deixamos como sugestões a trabalhos futuros.

Referências Bibliográficas

[1] K. Abed-Meraim, A. Chkeif and Y. Hua, Fast Orthonormal PAST Algorithm, *IEEE Signal Processing Letters* 7 (3), 2000.

[2] R. Badeau, R. Boyer and B. David, EDS parametric modeling and tracking of audio signals, *5th Int. Conf. on Digital Audio Effects (DAFx-02)* 139-144, Hamburg, 2002, .

[3] R. Badeau, B. David and G. Richard, A new perturbation analysis for signal enumeration in rotational invariance techniques, *IEEE Transactions on Signal Processing*, 54 (2), 2006.

[4] R. Badeau, G. Richard and B. David, Fast Adaptive Esprit Algorithm, *IEEE 13th Workshop on Statistical Signal Processing* 05, 289-294, 2005.

[5] D. Benson, Music: A Mathematical Offering, Cambridge: University Press, 2007.

[6] L. H. Bezerra and F. S. V. Bazán, Eigenvalue locations of generalized companion pradictor matrices, *SIAM J. Matrix Anal. Appl.* 19(4), 886-897, 1998.

[7] W. L. Briggs, V. E. Henson, The DFT: An Owner's Manual for the Discrete Fourier Transform, Philadelphia: Society for Industrial Mathematics, 1995.

[8] J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, *Math. Comput.* 19, 297-301, 1965.

[9] B. David, G. Richard and R. Badeau, An EDS Modelling Tool for Tracking and Modifying Musical Signals, *Proc. of the Stockholm Music Acoustics Conference*, Stockholm, 2003.

[10] M. Dolson, The Phase Vocoder: A Tutorial, *Computer Music Journal* 10, 14-27, 1986.

[11] D. P. W. Ellis, A Phase Vocoder in Matlab, `http://www.ee.columbia.edu/~dpwe/resources/matlab/pvoc/`, 2002.

[12] G. H. Golub and C. F. Van , Matrix Computations 3rd. ed., Baltimore: Johns Hopkins, 1996.

[13] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim and Y. Miao, A new look at the power method for fast subspace tracking, *Digital Signal Processing* 9, 297-314, 1999.

[14] D. L. Kreider et al., Introdução à Análise Linear, 3 volumes, Rio de Janeiro: Ao Livro Técnico, 1972.

[15] R. J. McAulay and T. F. Quatieri, Speech analysis/synthesis based on a sinusoidal representation, *IEEE Transactions on Acoustics, Speech, Signal Processing* 34, 744-754, 1986.

[16] M. R. Petersen, Musical Analysis and Synthesis in Matlab, *MAA's College Mathematics Journal* 35 (5), 396-401, 2004.

[17] M. R. Portnoff, Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform, *IEEE Transavtions on Acoustics, Speech and Signal Processing* 24 (3), 243-248, 1976.

[18] R. Roy and T. Kailath, ESPRIT - estimation of signal parameters via rotational invariance techniques, *IEEE Trans. on Acoustics, Speech, and Signal Processing* 37, 984-995, 1989.

[19] X. Serra and J. O. Smith, PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation, *Proceedings of the International Computer Music Conference (ICMC-87)*, Computer Music Association, Tokyo, 1987 .

[20] X. Serra and J. O. Smith, Spectral Modeling Synthesis: a sound analysis/synthesis based on a deterministic plus stochastic decomposition, *Computer Music Journal* 14 (4), 12-24, 1990.

[21] X. Serra, Basic Spectral Modeling Synthesis code in MATLAB, `http://mtg.upf.edu/sms/software/smsMatlab.zip`.

[22] J. O. Smith, Mathematics of the Discrete Fourier Transform (DFT) 2nd ed., Seattle: BookSurge, 2007.

[23] G. W. Stewart and J.-G. Sun, Matriz Perturbation Theory, London: Academic Press, 1990.

[24] P. Strobach, Square hankel SVD Subspace tracking algorithms, *Signal Processing* 57, 1-18, 1997.

[25] B. Yang, Projection Approximation Subspace Tracking, *IEEE Trans. Signal Processing* 44 (1), 95-107, 1995.

[26] U. Zölzer, DAFX: Digital Audio Effects, New York: Wiley, 2002.