UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

YAN MOURA LIMA

IDENTIFICAÇÃO AUTOMÁTICA DE ACORDES MUSICAIS

VIÇOSA
2020

YAN MOURA LIMA

IDENTIFICAÇÃO AUTOMÁTICA DE ACORDES MUSICAIS

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia II – e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Rodolpho Vilela Alves Neves.
Coorientador: Prof. M.Sc. Felipe Antunes.

VIÇOSA
2020

YAN MOURA LIMA

IDENTIFICAÇÃO AUTOMÁTICA DE ACORDES MUSICAIS

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia II e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 26 de junho de 2020.

COMISSÃO EXAMINADORA

Rodolpho Neves

Prof. Dr. Rodolpho Vilela Alves Neves - Orientador
Universidade Federal de Viçosa

Felipe Antunes

Prof. M.Sc. Felipe Antunes - Membro
Instituto Federal de Minas Gerais – Campus Ipatinga

William Caires Silva Amorim

Prof. M.Sc. William Caires Silva Amorim - Membro
Universidade Federal de Viçosa

“Eu não tenho medo do homem que praticou 10.000 chutes diferentes, mas sim do homem que praticou o mesmo chute 10.000 vezes.”

(Bruce Lee)

Dedico este trabalho, ao meu pai Paulo (in memoriam), a minha mãe Waldênia e ao meu irmão Thales.

Agradecimentos

À Deus pela presença em minha vida, à minha família pelo apoio e carinho. À Universidade Federal de Viçosa (UFV) pelas oportunidades de: cursar Engenharia Elétrica; ser tutor das disciplinas: MAT 093 e MAT 097 de Cálculo 1 e monitor das disciplinas: ELT 224 - Instalações Elétricas I, ELT 210 - Medidas Elétricas e Magnéticas, ELT 212 - Laboratório de Medidas Elétricas I e ELT 410 - Sinais e Sistemas. Também pela bolsa de iniciação científica do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e ao Heitor pela oportunidade de estágio na Pró reitoria de Administração (PAD), no setor Gerência de Projetos e Constatação de Obras (GPC).

Ao Orientador Rodolpho pelas dicas, disponibilidade para tirar dúvidas e ensinamentos, ao Coorientador Felipe pela dedicação e paciência, principalmente no momento de elaboração dos algoritmos, também pela disponibilidade de tirar as dúvidas na época em que fui monitor de ELT 210 e ELT 212, ao William pela participação na banca e por ter sido um excelente monitor na disciplina ELT 410.

Aos professores pelos ensinamentos, aos técnicos e funcionários do Departamento de Engenharia Elétrica (DEL) pela disposição em ajudar. Ao pessoal do Laboratório de Engenharia Elétrica (LEE) e ao Núcleo Interdisciplinar de Análise de Sinais (NIAS) por permitirem a realização de testes desse trabalho. Aos colegas de curso que me ajudaram no caminho do conhecimento. Ao pessoal da banda da IPVSOL pelo conhecimento musical adquirido e a todos que contribuíram para minha formação acadêmica e pessoal de forma direta ou indireta.

Resumo

A identificação de acordes musicais pode ser realizada através do método *Pitch Class Profile* (PCP) que consiste em um vetor de 12 dimensões contendo em cada posição a energia acumulada das notas musicais presentes na escala cromática. Sabendo que frequências musicais são geometricamente espaçadas e que a segmentação de um sinal pode gerar vazamento espectral, propõe-se o método *Least Squares Constant Q Transform* (LSCQT) como etapa de mapeamento de frequências e a comparação com os métodos *Discrete Fourier Transform* (DFT), *Least Squares* (LS) e *Constant Q Transform* (CQT). O trabalho foi realizado em duas partes: Teste em Músicas Artificias e Teste em Música Real. Na primeira etapa utilizaram-se os seguintes testes: Teste A, considerando a taxa de amostragem de 44,1 kHz e Tempo Médio do Acorde (TMA) de 1,8 s; Teste B, mantendo a mesma taxa de amostragem e reduzindo o TMA para 0,9 s e o Teste C, reduzindo a taxa de amostragem para 16 kHz e considerando TMA de 1,8 s. Utilizou-se 17 tipos de função de janelamento aplicadas em 100 músicas construídas artificialmente para determinar os melhores parâmetros. Na segunda etapa, Teste em Música Real, utilizaram-se os melhores parâmetros obtidos no Teste C. Analisou-se a música *Money That's What I Want-Beatles*, álbum *With The Beatles* presente no *dataset* do grupo de pesquisa LabROSA. Foi realizada a separação do áudio em conteúdos instrumental e vocal utilizando o método *Robust Principal Component Analysis* (RPCA). Nos Testes em Músicas Artificiais observou-se redução de desempenho: ao fixar a taxa de amostragem e reduzir o Tempo Médio do Acorde (TMA) e ao fixar o Tempo Médio do Acorde (TMA) e reduzir a taxa de amostragem. Para todos os Testes em Músicas Artificiais, a janela *Flat Top* apresentou os piores resultados. Para os Testes em Músicas Artificias, o método LSCQT obtém melhor Taxa de Acerto Média (TAM) nos Testes A e B. Já no Teste C, o LS é melhor. No Teste em Música Real, sugere-se que o método LSCQT seja o mais indicado para identificação de acordes. Como trabalhos futuros propõe-se para o Teste em Música Real: valores adequados para λ e G, a utilização do algoritmo *Beat Tracking* com o intuito de segmentar a música em trechos de tamanhos corretos, a utilização do algoritmo *Key Detection* para determinar o campo harmônico musical e um estudo comparativo entre métodos de correspondência de modelo e métodos de aprendizagem para classificação de acordes musicais.

Palavras-chave: *Discrete Fourier Transform*, *Least Squares*, *Constant Q Transform*, *Robust Principal Component Analysis*, *Pitch Class Profile*.

Abstract

The identification of musical chords can be performed using the Pitch Class Profile (PCP) method, which consists of a 12 dimension vector containing in each position the accumulated energy of the musical notes presents in the chromatic scale. Knowing that musical frequencies are geometrically spaced and that the segmentation of a signal can generate spectral leakage, the Least Squares Constant Q Transform (LSCQT) method is proposed as a frequency mapping step and the comparison with the Discrete Fourier Transform (DFT), Least Squares (LS) and Constant Q Transform (CQT). The work was carried out in two parts: Tests on Artificial Musics and Test on Real Music. In the first stage, the following tests were used: Test A, considering the sampling rate of 44.1 kHz and Average Chord Time (TMA) of 1.8 s; Test B, maintaining the same sampling rate and reducing the TMA to 0.9 s and Test C, reducing the sampling rate to 16 kHz and considering a TMA of 1.8 s. 17 types of window function applied to 100 artificially constructed musics were used to determine the best parameters. In the second stage, Test on Real Music, the best parameters obtained in Test C were used. The song Money That's What I Want-Beatles, album With The Beatles present in the dataset of the research group LabROSA, was analyzed. The separation of audio into instrumental and vocal content was performed using the Robust Principal Component Analysis (RPCA) method. In the Tests on Artificial Musics, a reduction in performance was observed: by setting the sample rate and reducing the Average Chord Time (TMA) and by setting the Average Chord Time (TMA) and reducing the sample rate. For all Tests on Artificial Musics, the Flat Top window presented the worst results. For the Tests on Artificial Musics, the LSCQT method obtains a better Average Hit Rate (TAM) in Tests A and B. In Test C, the LS is better. In the Test on Real Music, it is suggested that the LSCQT method is the most suitable for identifying chords. As future work it is proposed for the Test on Real Music: adequate values for λ and G, the use of the Beat Tracking algorithm in order to segment the music in sections of correct sizes, the use of the Key Detection algorithm to determine the harmonic field and a comparative study between template matching methods and learning methods for classifying musical chords.

Keywords: Discrete Fourier Transform, Least Squares, Constant Q Transform, Robust Principal Component Analysis, Pitch Class Profile.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

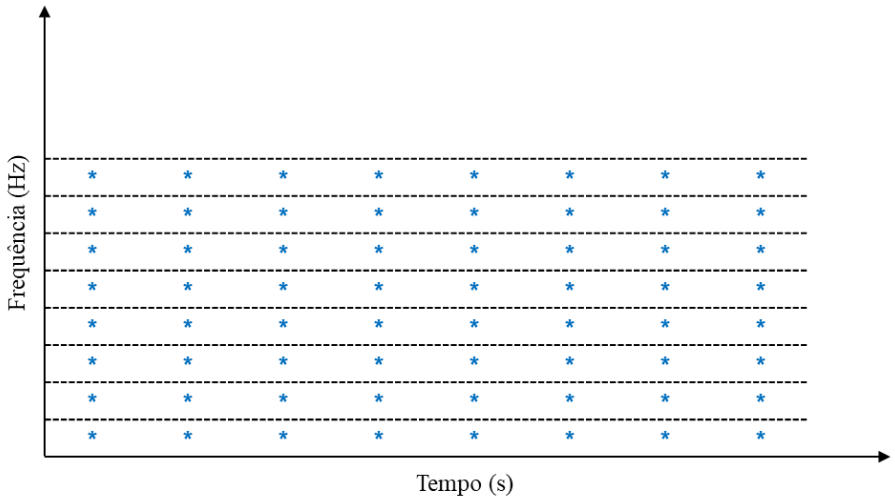
O reconhecimento de acordes é a etapa inicial para tomada de decisões quanto à mudança de tom, substituição de acordes maiores pelos seus relativos menores e a adição de acordes de preparação. A vantagem do reconhecimento automático de acordes musicais é possibilitar a identificação de sequências de acordes sem precisar aprender a música de ouvido, tarefa que pode ser difícil, dependendo do grau de conhecimento e treino de um músico. Com o avanço de métodos de processamento de sinais, como, por exemplo: mapeamento de frequências musicais (BROWN, 1991), detecção de intensidade de notas musicais (FUJISHIMA, 1999), separação do conteúdo instrumental do conteúdo vocal (HUANG, CHEN, *et al.*, 2012a) e aprendizagem de máquina (RAO, GUAN e TENG, 2016) é possível realizar o procedimento de detecção de acordes de forma cada vez mais precisa.

1.1. Mapeamento de Frequências Musicais

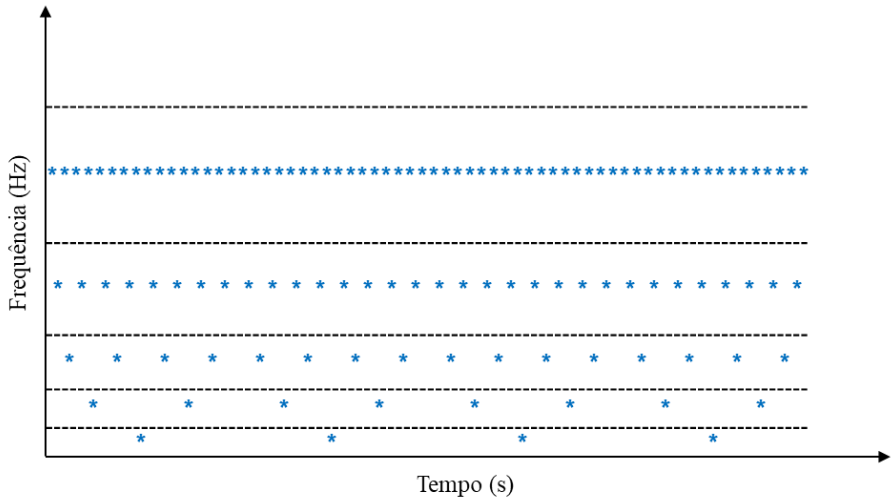
Para a extração de características de um sinal, é necessário a segmentação do áudio e o mapeamento das frequências musicais, que pode ser realizado por meio de vários métodos. Dentre os métodos existentes o *Discrete Fourier Transform* (DFT) é o mais tradicional, no entanto não mapeia de forma eficiente as frequências musicais (BROWN, 1991). Isso ocorre porque nessa transformada os *bins* estão separados por uma resolução constante, ou seja, considera-se constante a distância entre as frequências consecutivas, no entanto as frequências musicais são geometricamente espaçadas, isto é, a resolução em frequência aumenta conforme o aumento da frequência (BROWN, 1991). O método *Constant Q Transform* (CQT), é mais eficiente nesse aspecto, pois considera que o número de ciclos é sempre constante, independente da frequência, consequentemente com o aumento da frequência a resolução em frequência piora e a resolução temporal melhora (SCHÖRKHUBER e KLAPURI, 2010). A Figura 1 mostra as diferenças entre os métodos DFT e CQT quanto as resoluções em frequência e temporal, na qual os asteriscos indicam a intensidade em função da frequência e do tempo. A distância vertical entre os asteriscos consecutivos significa resolução em frequência e a horizontal, resolução temporal e quanto menor a distância entre os asteriscos consecutivos, melhor a resolução (TODISCO, DELGADO e EVANS, 2017).

Figura 1 – Comparação quanto as resoluções em frequência e temporal: (a) DFT, (b) CQT.

(a)



(b)



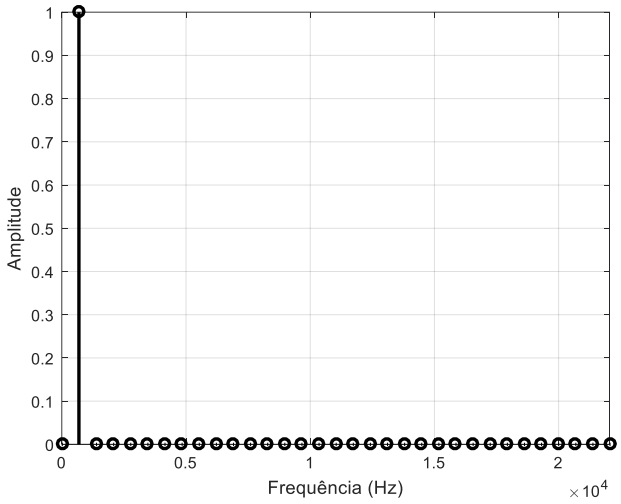
Fonte: Adaptado de TODISCO, DELGADO e EVANS, 2017.

No processo de detecção de acordes é necessário segmentar o áudio, logo pode ocorrer que o número de ciclos associado a uma determinada frequência não seja um número inteiro, com isso ocorre o vazamento espectral, comum no método DFT (JWO, WU e CHANG, 2019). O método CQT não resolve este problema, pois o número de ciclos apesar de ser constante não é exatamente um número inteiro. Define-se o número de ciclos como a razão entre a frequência considerada e a resolução, sendo que a resolução é dada pela razão entre a taxa de amostragem em Hz e o número de amostras (CHIOYE, KAY e LISKA, 2017). Como exemplo, a Figura 2 ilustra os espectros de uma senóide de amplitude unitária obtidos a partir do método DFT. No primeiro caso, considera-se 64 amostras, taxa de amostragem de 44,1 kHz e frequência

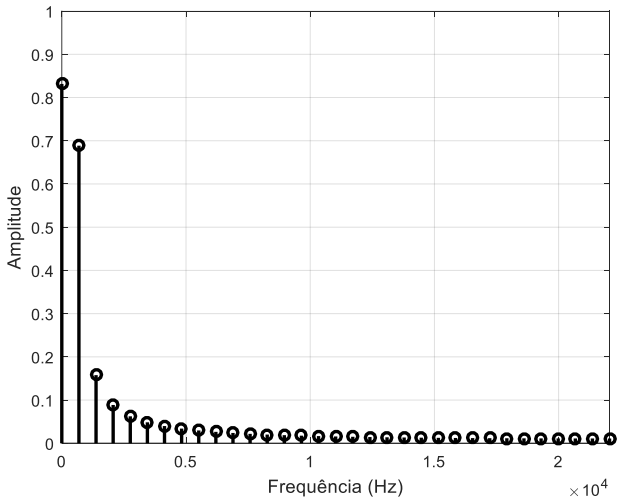
fundamental de 689,0625 Hz, logo o número de ciclos é igual a 1. Já no segundo caso, utilizam-se os mesmos parâmetros, exceto a frequência fundamental que é de 440 Hz, frequência padrão de afinação (ISO, 1975), logo o número de ciclos é 0,6385 (não inteiro), causando então um espalhamento em torno da frequência fundamental.

Figura 2 – Domínio da frequência: (a) Ausência de vazamento espectral, (b) Presença de vazamento espectral.

(a)



(b)



Fonte: Autoria própria.

Para reduzir o vazamento espectral, ou seja obter um mapeamento mais eficiente das frequências musicais seria possível através do método LS, pois minimiza a distância euclidiana entre as amostras do sinal e as amostras do modelo proposto, com o objetivo de estimar a curva que melhor caracteriza o comportamento do sinal (MARTINO, LOSITO e MASI, 2012). Para a determinação da curva, estima-se as amplitudes para cada frequência independente do fato de ocorrer um número inteiro de ciclos dentro da janela (MARTINO, LOSITO e MASI, 2012). A aproximação feita por onda senoidal é utilizada, por exemplo, para estimar os parâmetros amplitude e fase (MARTINO, LOSITO e MASI, 2012). Outras aplicações do LS incluem estimação de campo gravitacional a partir de dados coletados de satélite minimizando o efeito do vazamento espectral (GOOSSENS, 2010), recuperação de sinal original a partir da CQT (INGLE e SETHARES, 2012) e utilização do método *Non Negative Least Squares* (NNL) para identificação de acordes musicais (MAUCH e DIXON, 2010).

Além do número de ciclos presentes em cada trecho do sinal, é necessário determinar a função de janelamento que minimize o efeito do vazamento espectral, ou seja, que no domínio da frequência tenha lóbulo principal estreito e lóbulos laterais bem atenuados (LATHI, 2006). Isso sugere que ao combinar os métodos CQT e LS com funções de janelamento adequadas é possível melhorar o mapeamento das frequências musicais e assim determinar os acordes musicais com melhor precisão.

1.2 Reconhecimento de Acordes Musicais

A escala cromática musical é composta por 12 notas e a distância entre duas notas consecutivas é de um semitom (metade de um tom) (BENWARD e SAKER, 2008). A letra alfabética e o número adjacente à essa representam respectivamente o nome da nota e sua oitava correspondente, por exemplo, C4: Dó na oitava quatro, já o símbolo "#" acompanhada da letra alfabética representa uma nota sustenida, por exemplo, C#: Dó sustenido (BENWARD e SAKER, 2008) .

Um acorde musical é definido por três frequências no mínimo: tônica, terça e quinta (BENWARD e SAKER, 2008). Quando o intervalo da tônica para terça é de dois tons (quatro semitons) e o intervalo da terça para quinta é de um tom e meio (três semitons), têm-se um acorde dito maior (BENWARD e SAKER, 2008). Já quando o tamanho dos intervalos, tônica até terça e terça até quinta passam ser respectivamente um tom e meio e dois tons, têm-se um

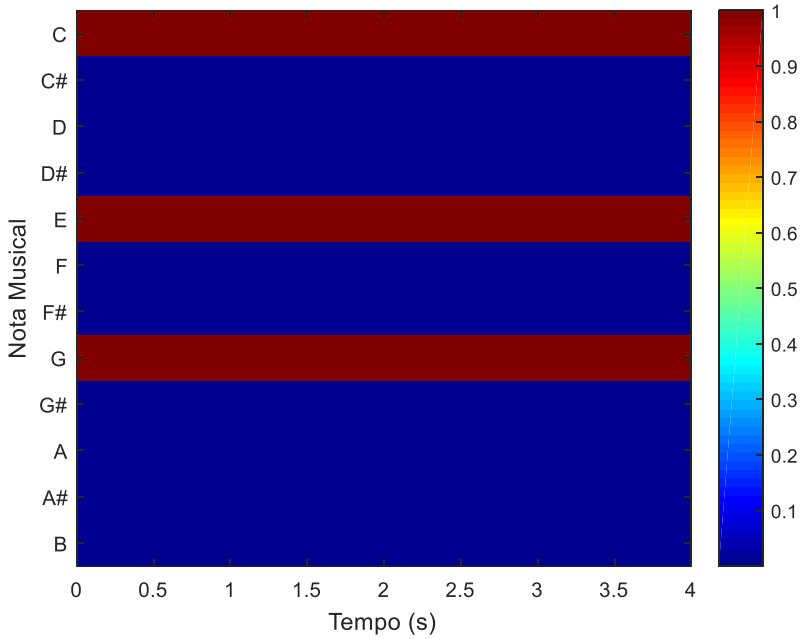
acorde dito menor (BENWARD e SAKER, 2008). Por exemplo, um acorde Dó Maior é formado pelas frequências das notas C, E e G e um acorde de Dó Menor, pelas frequências das notas C, D# e G (BENWARD e SAKER, 2008).

Para o reconhecimento de acordes musicais é necessário realizar um pré-processamento no sinal como, por exemplo, a utilização de *Robust Principal Component Analysis* (RPCA) para separação do conteúdo vocal dos instrumentos musicais (HUANG, CHEN, *et al.*, 2012a), em seguida utiliza-se procedimentos para segmentar o áudio, por exemplo *Beat Tracking* (ELLIS, 2007), e para extração das características do sinal, utiliza-se métodos como *Constant Q Transform*, CQT, que mapeia frequências considerando resolução variável com a frequência (BROWN, 1991). Posteriormente aplicam-se métodos como o *Pitch Class Profile* (PCP) para determinação das energias correspondentes a cada uma das 12 notas da escala cromática (FUJISHIMA, 1999). A última etapa consiste na classificação dos acordes musicais que podem ser realizados por métodos de correspondência de modelo (FUJISHIMA, 1999) e de aprendizagem (RAO, GUAN e TENG, 2016). Para facilitar a classificação é possível realizar a identificação do campo harmônico (tom da música) através do algoritmo *Key Detection* que extrai os acordes mais prováveis (ZENZ e RAUBER, 2007).

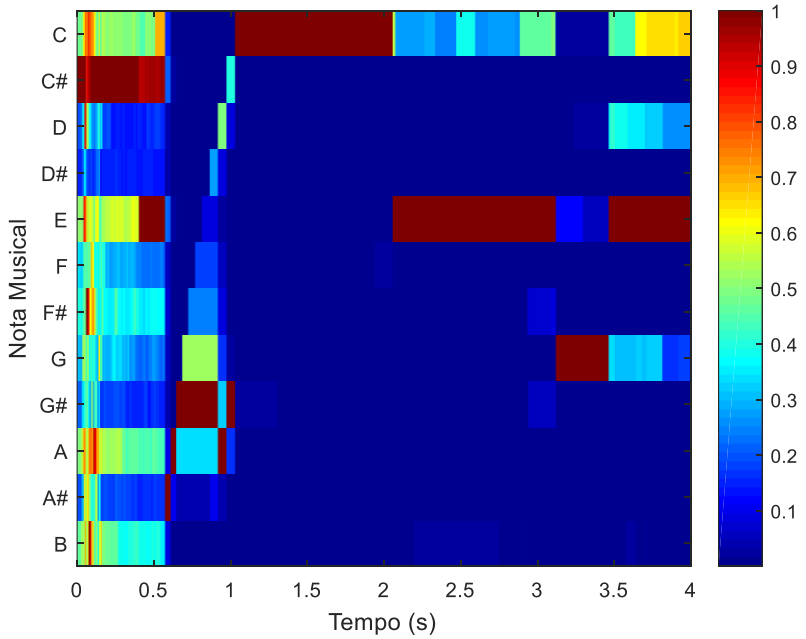
A partir das três posições do vetor PCP que apresentam maior energia é possível identificar as notas tônica, terça e quinta que compõem uma tríade (FUJISHIMA, 1999). Porém, esse método apresenta alguns problemas como: presença de frequências indesejadas causadas por harmônicas produzidas nos instrumentos musicais e confusão no reconhecimento de acordes maiores e seus relativos menores que apresentam notas em comum (LEE, 2006). Outro fator que reduz a precisão do método é a presença de voz e instrumento em um mesmo áudio (RAO, GUAN e TENG, 2016). Na Figura 3 observa-se a diferença entre um PCP ideal e real obtido a partir do acorde Dó Maior. No caso ideal, existe energia apenas nas notas musicais C, E e G que compõe o acorde e possuem mesma intensidade, duração e estão em sincronia. Já no caso real, por exemplo, gravação feita em um violão, as notas não possuem mesma energia, duração e não são tocadas ao mesmo tempo, além disso, ocorre aparecimento de harmônicas indesejadas.

Figura 3 – Acorde Dó Maior: (a) PCP ideal, (b) PCP real.

(a)



(b)



Fonte: Autoria própria.

Com o intuito de atenuar as harmônicas indesejadas e reduzir a confusão de acordes maiores e seus relativos menores, foi proposto em 2006, o método *Enhanced Pitch Class Profile* (EPCP) que utiliza o método *Harmonic Product Spectrum* (HPS) para extrair a frequência fundamental do sinal (LEE, 2006). Nesse caso, para classificação dos acordes foi utilizado o método da correlação entre os vetores PCP e um banco de dados de 24 acordes tríades (12 maiores e 12 menores).

Para reduzir a influência da voz cantada, propôs-se em 2016, o método *Enhanced Logarithmic Picth Class Profile* (ELPCP) utilizando: o método *Robust Principal Component Analysis*, RPCA, para separação da voz cantada dos instrumentos musicais, o método logarítmico para extração do PCP e para classificação de acordes o *Temporal Correlator Support Vector Machine* (TCSVM) (RAO, GUAN e TENG, 2016).

Na etapa de classificação de acordes têm-se utilizado métodos de correspondência de modelo como os métodos do vizinho mais próximo e da soma ponderada (FUJISHIMA, 1999). Possuem como vantagens não necessitar de treinamento e histórico de dados de acordes reais, isto é, gabarito contendo sequência de acordes ao longo do tempo, logo possuem baixo tempo de processamento quando comparados com os métodos de aprendizagem (RAO, GUAN e TENG, 2016), tais como: *Hidden Markov Method* (HMM) (BELLO e PICKENS, 2005), *Dynamic Bayesian Network* (DBN) (MAUCH e DIXON, 2010) e *Temporal Correlator Support Vector Machine*, TCSVM, (RAO, GUAN e TENG, 2016). No entanto, métodos de correspondência de modelo, como baseados em matriz binária de acordes, são limitados, pois consideram distâncias fixas entre as notas musicais, o que não ocorre em acordes reais (LEE, 2006).

Diante do exposto, constatam-se avanços na literatura nas etapas do processo de identificação de acordes. No entanto, ainda existem possibilidades de melhorias na etapa de mapeamento de frequências musicais. Considerando as vantagens citadas dos métodos *Least Squares*, LS e *Constant Q Transform*, CQT, propõem-se a utilização de um procedimento de extração de características denominado por *Least Squares Constant Q Transform* (LSCQT) para mapeamento de frequências musicais.

1.3 Objetivos Geral e Específicos

Este trabalho tem como objetivo geral propor o método *Least Squares Constant Q Transform*, LSCQT, como etapa de mapeamento de frequências no processo de identificação de acordes musicais e compará-lo com os métodos DFT, LS e CQT. Para isso, têm-se como objetivos específicos:

- Avaliar o efeito da redução do Tempo Médio do Acorde (TMA) fixando a taxa de amostragem utilizando Taxa de Acerto (TA);
- Avaliar o efeito da redução da taxa de amostragem fixando o Tempo Médio do Acorde (TMA) utilizando Taxa de Acerto (TA);
- Comparar os quatro métodos utilizando Taxa de Acerto (TA) e determinar os melhores parâmetros de janelamento. Para os métodos DFT e LS, parâmetro tamanho de janela em segundos e para os métodos CQT e LSCQT, parâmetro número de ciclos na janela;
- Comparar os métodos com e sem influência do RPCA.

2 Materiais e Métodos

O trabalho foi divido em duas partes principais, Testes em Músicas Artificiais e Teste em Música Real.

2.1 Testes em Músicas Artificiais

Adotou-se para as músicas artificiais 60 s de duração e o intervalo de tempo de cada acorde de 0,5 s a 4 s. Definiu-se o Tempo Médio do Acorde, TMA, como a média aritmética dos intervalos de tempo de cada acorde. Foram realizados os seguintes testes: Teste A, considerando taxa de amostragem de 44,1 kHz e TMA de 1,8 s; Teste B, mantendo a mesma taxa de amostragem e reduzindo o TMA para 0,9 s e o Teste C, reduzindo a taxa de amostragem para 16 kHz e com TMA de 1,8 s. Foram utilizados 17 tipos de função de janelamento aplicadas em 100 músicas construídas artificialmente e foram determinados os melhores parâmetros de janelamento.

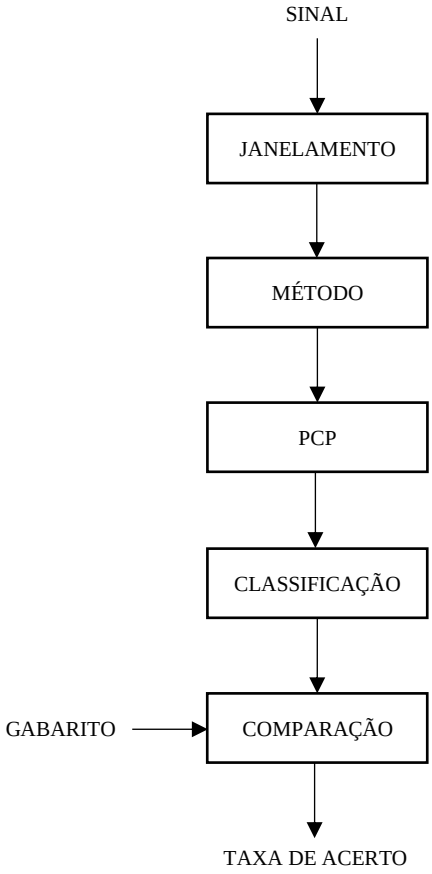
Para os métodos DFT e LS os parâmetros considerados foram: tamanho da janela que variou de 10 ms a 400 ms e tipo de função de janelamento. Já para os métodos CQT e LSCQT, considerou-se como parâmetros: o número de ciclos na janela que variou de 17 até 340 em múltiplos de 17 e o tipo de função de janelamento. Para determinar quais parâmetros de janelamento foram os melhores para cada método, definiu-se a Taxa de Acerto, TA, como a comparação do gabarito artificial referente a música artificial com o vetor de classificação obtido em cada método. A seguir para determinar qual foi o melhor método calculou-se a Taxa de Acerto Média (TAM).

As etapas utilizadas encontram-se na Figura 4. Como entrada do processo têm-se o sinal de música artificial e como saída, a Taxa de Acerto, TA. O sinal e o gabarito artificial foram determinados a partir de um banco de frequências contendo 9 linhas (oitavas) por 12 colunas (notas musicais). Na etapa de janelamento, o sinal musical foi divido em trechos de mesmo tamanho e aplicado em cada trecho uma mesma função de janelamento. Para isso, utilizou-se 17 tipos diferentes de função de janelamento. Com posse do banco de frequências, extraíram-

se as amplitudes do sinal para cada frequência utilizando os métodos DFT, LS, CQT e o LSCQT. A partir dessas amplitudes determinou-se a matriz PCP para cada método.

A classificação dos acordes foi feita encontrando a maior covariância entre o PCP e todos os acordes da matriz binária. A seguir, para cada janela determinou-se a linha (acorde) dessa matriz de classificação que possuía valor máximo. Esses valores foram armazenados em um vetor linha. Para os métodos de janelamento invariante com a frequência (DFT e LS) cada valor do vetor foi replicado na quantidade de amostras do sinal presente na janela. Já para os métodos de janelamento variante com a frequência (CQT e LSCQT) não foi preciso replicar os acordes, pois o PCP calculado para esse caso possui um número de colunas igual ao número de amostras do sinal musical. Por fim, o vetor de classificação foi comparado ponto a ponto com o vetor de gabarito artificial e foi realizado o cálculo da Taxa de Acerto, TA. O algoritmo Testes em Músicas Artificiais detalha as etapas descritas na Figura 4.

Figura 4 – Etapas principais.



Fonte: Autoria própria.

Testes em Músicas Artificiais

1: $JANELA \leftarrow \{"@barthannwin", \ldots, "@tukeywin"\}$; // Vetor de *strings* com os 17 tipos de janelas.

2: Para JJ=1:17 // Seleciona função de janelamento;

3: $janela \leftarrow JANELA\{JJ\}$; // Variável auxiliar que recebe função de janelamento.

5: Carrega a matriz binária de acordes músicas e retira de cada linha 3/12;

6: Obtém-se o banco de frequências;

7: $f_S \leftarrow 44100$; // Define-se a taxa de amostragem utilizada. Testes A e B: 44.100 Hz e Teste C: 16.000 Hz.

8: $T \leftarrow 60$; // Define-se o tempo de 60 s para cada música.

9: $L \leftarrow f_S * T$; // Define-se o tamanho do sinal.

10: $TMA \leftarrow 1{,}8$; // Define-se o Tempo Médio do Acorde (TMA). Testes A e B: 1,8 s e Teste C: 0,9 s.

11: $QA \leftarrow round(T/TMA)$; // Calcula-se a quantidade de acordes aproximada que contém no sinal.

12: **Para m=1:100** // Seleciona Música Artificial.

13: **Algoritmo 1: Determinação da Música Artificial e do Gabarito Artificial;**

14: **Para n=1:40** // Variável auxiliar para determinar tamanho da janela.

15: **Algoritmo 2: Cálculo do PCP (DFT);**

16: **Algoritmo 3: Cálculo da Taxa de Acerto (TA) (DFT);**

17: **Algoritmo 4: Cálculo do PCP (LS);**

18: **Algoritmo 5: Cálculo da Taxa de Acerto (TA) (LS);**

19: **Fim**

20: **Para Q=17:17:340** // Seleciona o número de ciclos na janela.

21: $q \leftarrow 1$; // Variável auxiliar para contar colunas da Taxa de Acerto (TA).

22: **Algoritmo 7: Cálculo do PCP (CQT);**

23: **Algoritmo 8: Cálculo da Taxa de Acerto (TA) (CQT);**

24: **Algoritmo 9: Cálculo do PCP (LSCQT);**

25: **Algoritmo 10: Cálculo da Taxa de Acerto (TA) (LSCQT);**

26: $q \leftarrow q + 1$; // Para contagem.

27: **Fim**

28: **Fim**

29: Fim

2.1.1 Determinação do Banco de Dados de Frequências Musicais

Definiu-se como referência a nota Lá de afinação padrão cuja frequência é de 440 Hz (ISO, 1975). A partir desta, determinaram-se as 11 frequências vizinhas da 5ª linha (Tabela 1), utilizando uma progressão geométrica de razão $2^{1/12}$ (ORTIZ-ECHEVERRI, RODRÍGUEZ-RESÉNDIZ e GARDUÑO-APARICIO, 2018). Para uma mesma nota musical (coluna da Tabela 1) a frequência dobra ao aumentar a oitava em uma unidade (EVEREST e POHLMANN, 2009), logo obteve-se a matriz de frequências em Hz representada na Tabela 1.

Tabela 1 – Matriz de frequências musicais em Hz.

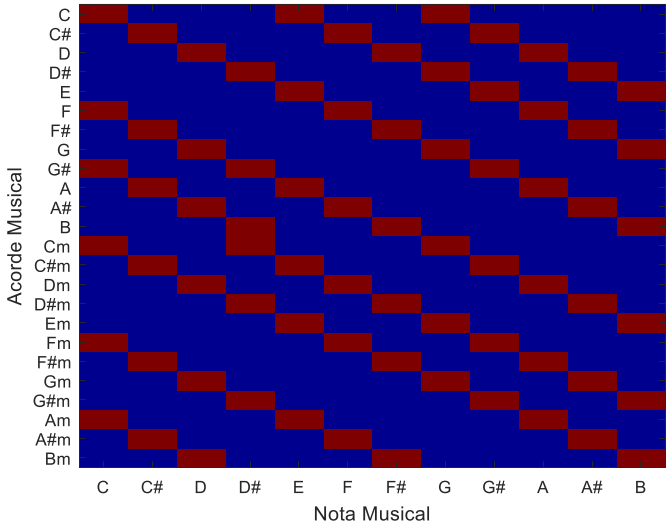
C	C#	D	D#	E	F	F#	G	G#	A	A#	B
16,352	17,324	18,354	19,445	20,602	21,827	23,125	24,5	25,957	27,5	29,135	30,868
32,703	34,648	36,708	38,891	41,203	43,654	46,249	48,999	51,913	55	58,27	61,735
65,406	69,296	73,416	77,782	82,407	87,307	92,499	97,999	103,83	110	116,54	123,47
130,81	138,59	146,83	155,56	164,81	174,61	185	196	207,65	220	233,08	246,94
261,63	277,18	293,66	311,13	329,63	349,23	369,99	392	415,3	440	466,16	493,88
523,25	554,37	587,33	622,25	659,26	698,46	739,99	783,99	830,61	880	932,33	987,77
1046,5	1108,7	1174,7	1244,5	1318,5	1396,9	1480	1568	1661,2	1760	1864,7	1975,5
2093	2217,5	2349,3	2489	2637	2793,8	2960	3136	3322,4	3520	3729,3	3951,1
4186	4434,9	4698,6	4978	5274	5587,7	5919,9	6271,9	6644,9	7040	7458,6	7902,1

Fonte: Adaptado de DELL'AVERSANA, GABBRIELLINI e AMENDOLA, 2016.

2.1.2 Determinação das Músicas Artificias e do Gabarito Artificial

Primeiramente foi construída uma matriz binária contendo 24 linhas (C: Dó Maior até Bm: Si Menor) e 12 colunas (C até B). Por exemplo, o acorde Dó Maior é representado pelo vetor (1,0,0,0,1,0,0,1,0,0,0,0), ou seja, pelas notas C, E G e o acorde Dó Menor representado por (1,0,0,1,0,0,0,1,0,0,0,0), ou seja, pelas notas C, D# e G (LEE, 2006). Na Figura 5, as hachuras vermelhas e azuis significam *bit* 1 (presença da nota) e 0 (ausência da nota) respectivamente.

Figura 5 – Matriz binária de acordes musicais: hachuras vermelhas (presença de nota) e azuis (ausência de nota).



Fonte: Autoria própria.

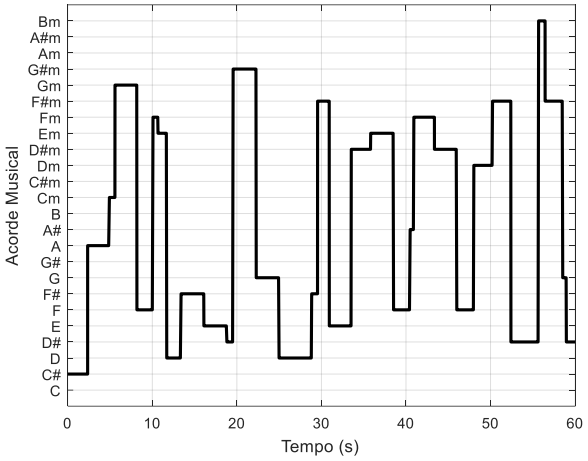
Inicialmente definiu-se o Tempo Médio do Acorde, TMA, como 1,8 s que foi posteriormente reduzido para 0,9 s. Definiu-se um tempo total de 60 s para cada música artificial. A quantidade de acordes por música foi definida pelo número inteiro mais próximo, dado pela razão entre o tempo total e o TMA.

A partir da quantidade de acordes, foi definido o tempo de cada acorde de modo que variasse de 0,5 s a 4 s, em seguida a soma desses tempos foi limitada para 60 s e foi definido o número de amostras de cada acorde supondo uma frequência de amostragem de 44,1 kHz, reduzida posteriormente para 16 kHz.

Os sinais foram formados com componentes em todas as 9 oitavas. Foram selecionadas três colunas do banco de frequências respeitando a distância das notas da matriz binária (Figura 5). As amplitudes de cada oitava foram geradas por uma distribuição uniforme entre 0 e 1 e as fases foram geradas por uma distribuição uniforme entre 0 rad e 2π rad. Em cada música foi somado um ruído cujo valor de pico correspondeu ao dobro do valor *rms* do sinal e por fim, cada sinal foi normalizado. Esse ruído representa frequências indesejáveis como, por exemplo, presença de vozes e instrumentos de percussão.

O vetor gabarito da Figura 6, foi construído concatenando um número aleatório de 1 a 24 (acorde selecionado) multiplicando pelo conjunto de amostras correspondentes a cada acorde musical. As etapas descritas encontram-se no Algoritmo 1: Determinação da Música Artificial e do Gabarito Artificial.

Figura 6 – Gabarito artificial.



Fonte 1 Autoria própria.

Algoritmo 1: Determinação da Música Artificial e do Gabarito Artificial

1: $t_A \leftarrow rand(1, QA) * 3{,}5 + 0{,}5$; // Determina-se o tempo de cada acorde que varia de 0,5 s a 4 s.

2: $t_A \leftarrow 60 * t_A / sum(t_A)$; // Condição para que as somas dos tempos não ultrapassem 60 s.

3: $N_A \leftarrow round(f_S * t_A)$; // Número de amostras de cada acorde musical.

4: $N_A(end) \leftarrow N_A(end) + L - sum(N_A)$; // Últimas amostras recebem amostras restantes.

6: $y \leftarrow [\,]$; // Prelocação do sinal musical.

7: $GABARITO \leftarrow [\,]$; // Prelocação do vetor gabarito.

8: $AS \leftarrow zeros(1, QA)$; // Prelocação do vetor que seleciona o acorde.

9: Para q=1:QA // Seleciona-se quantidade de acordes presentes na música.

10: $AS(q) \leftarrow randperm(24{,}1)$; // Seleciona aleatoriamente um acorde.

11: $f_A \leftarrow f(:, BC(AS(q), :) > 0)$; // Seleciona três colunas de 9 frequências do banco de frequências.

12: $t \leftarrow 0{:}\,1/f_s{:}\,(N_A(q) - 1)/f_s$; // Define o vetor tempo de cada acorde.

13: $y_A \leftarrow zeros(size(t))$; // Prelocação do sinal que representa o acorde.

14: **Para b=1:3** // Seleciona coluna do banco f_A de frequências.

15: **Para h=1:9** // Seleciona a oitava do banco f_A de frequências.

16: $y_A \leftarrow y_A + rand * cos(2 * \pi * f_A(h, b) * t + 2 * \pi * rand)$; // Sinal do acorde recebe três cossenos.

17: **Fim**

18: **Fim**

19: $y \leftarrow [y, y_A]$; // Concatenam-se os sinais dos acordes até formar a música completa.

20: $GABARITO \leftarrow \left[GABARITO,\ AS(q) * ones\left(1, N_A(q)\right)\right]$; // Determina-se o gabarito.

21: Fim

22: $y \leftarrow y + 2 * rms(y) * rand\left(size(y)\right)$; // Soma-se ruído ao sinal.

23: $y \leftarrow y / max(abs(y))$; // Sinal musical é normalizado.

2.1.3 Janelamento Invariante com a Frequência (DFT e LS)

Cada áudio foi dividido em trechos de 10 ms a 400 ms e em cada trecho foi aplicado uma mesma função de janelamento. Para isso, utilizou-se 17 tipos de janelas: *Bartlett-Hann*, *Bartlett*, *Blackman*, *Blackman-Harris*, *Bohman*, *Chebyshev*, *Flat Top*, *Gaussian*, *Hamming*, *Hanning*, *Kaiser*, *Nuttall and Blackman-Harris*, *Parzen and Gaussian*, *Rectangular*, *Taylor*, *Triangular* e *Tukey* (MATHWORKS, 2020a). O tamanho N de cada janela foi determinado por:

$$N = f_s \Delta t, \tag{2.1}$$

na qual f_s representa a frequência de amostragem, $\Delta t = n/100$, o tempo em segundos de cada janela e n, variável auxiliar que varia de 1 a 40. O número de janelas J foi determinado por:

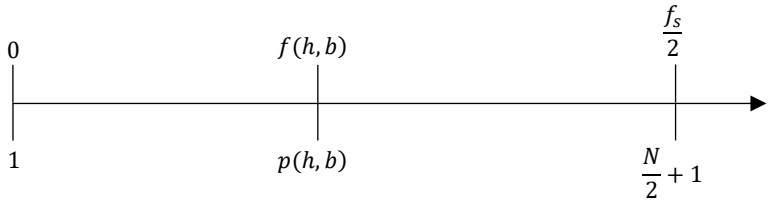
$$J = floor\left(\frac{L}{N}\right), \tag{2.2}$$

na qual L representa o número total de amostras e N o tamanho da janela. O sinal foi representado em uma matriz de N linhas e J colunas e multiplicado por uma função de janelamento.

2.1.3.1 Cálculo da DFT

A partir da Tabela 1 que contém 9 oitavas h e 12 notas musicais b, determinou-se as posições $p(h,b)$ aproximadas de todas as frequências $f(h,b)$.

Figura 7 – Relação entre amostra e frequência.



Fonte: Autoria própria.

A partir da Figura 7 é possível obter a relação em (2.3) e assim extrair cada posição em (2.4).

$$\frac{p(h,b)-1}{\frac{N}{2}+1-1}=\frac{f(h,b)-0}{\frac{f_s}{2}-0} \tag{2.3}$$

$$p(h,b)=round\left(1+\frac{Nf(h,b)}{f_s}\right) \tag{2.4}$$

A seguir utilizou-se o resultado da DFT nas posições $p(h,b)$. Para melhor desempenho computacional foi utilizada a função *Fast Fourier Transform* (FFT) (MATHWORKS, 2020b), aplicada em cada janela do sinal.

2.1.3.2 Cálculo do PCP (DFT)

Primeiramente definiu-se o PCP como uma matriz de 12 linhas e J colunas. Para cada janela J, definiu-se uma variável auxiliar que recebe as colunas do resultado da DFT. A seguir aplicou-se o resultado da DFT nas posições p do banco de frequências.

Elevando ao quadrado o valor absoluto de cada elemento da matriz DFT de 9 linhas e 12 colunas e somando em cada coluna seus 9 valores, obteve-se um vetor linha de 12 colunas que mede a energia acumulada das 9 oitavas para cada nota musical. A seguir esse vetor foi transposto e armazenado em cada coluna da matriz PCP. Esse processo foi realizado até completar todas as janelas conforme descrito abaixo:

Algoritmo 2: Cálculo do PCP (DFT)

1: $N \leftarrow f_s * (n/100)$; // Tamanho da Janela.

2: $J \leftarrow floor(L/N)$; // Número de Janelas.

3: $Y \leftarrow reshape(y(1:N*J),[N,J])$; // Redimensiona sinal para ter N linhas e J colunas.

4: W $\leftarrow window(eval(janela),N)$; // Função de janelamento.

5: $Y \leftarrow bsxfun(@times,W,Y)$; // Sinal é multiplicado ponto a ponto pela função de janelamento.

6: $X \leftarrow (2/N) * abs(fft(Y))$; // Cálculo da DFT.

7: $p \leftarrow round(1+N*(f/f_s))$; // Matriz Posição para cada frequência musical.

8: $PCP \leftarrow zeros(12,J)$; // Prelocação do PCP.

9: Para c=1:J // Janela.

10: $E \leftarrow X(:,c)$; // Varíavel auxiliar que recebe colunas da DFT.

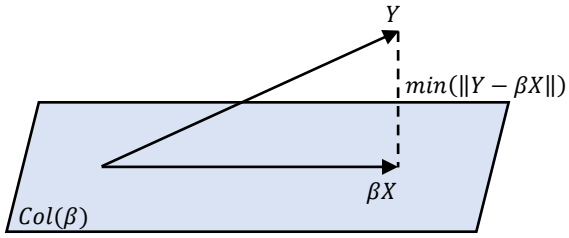
11: $PCP(:,c) \leftarrow sum(E(p).\text{^}2)^T$; // Cálculo do PCP.

12: Fim

2.1.3.3 Cálculo do LS

Considerou-se Y como a matriz que representa os dados do sinal janelado e βX a matriz que representa o modelo proposto. Conforme a Figura 8, o objetivo dos mínimos quadrados é determinar a matriz X no espaço $Col(\beta)$ que minimiza a distância euclidiana entre as matrizes Y e βX (LAY, 1999).

Figura 8 – Representação geométrica do método LS.



Fonte: Adaptado de LAY, 1999.

Para isso considerou-se o sistema dado por:

$$Y = \beta X. \tag{2.5}$$

Cada janela do sinal pode ser aproximada como uma onda senoidal, dada por:

$$y_{NJ} = A_J\, cos\big(2\pi f(h,b)t_N + \phi_J\big), \tag{2.6}$$

na qual y_{NJ} é a N - ésima amostra do sinal contido na janela J, A_J é a amplitude associada a frequência $f(h,b)$ e ϕ_J representa a fase. Expandindo (2.6) em termos de cossenos e senos tem-se:

$$y_{NJ} = A_J \cos\big(\phi_J\big) \cos(2\pi f(h,b)t_N) - A_J sen\big(\phi_J\big) sen(2\pi f(h,b)t_N). \tag{2.7}$$

Definiu-se β contendo N linhas e duas colunas, a primeira com os valores $cos(2\pi f(h,b)t_N)$ e segunda com os valores $-sen(2\pi f(h,b)t_N)$. E definiu-se X com duas linhas e J colunas contendo os valores de amplitude e fase desejados. Assim definiu-se o sistema indicado em (2.8).

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1J} \\ y_{21} & y_{22} & \cdots & y_{2J} \\ \cdots & \cdots & \cdots & \cdots \\ y_{N1} & y_{N2} & \cdots & y_{NJ} \end{bmatrix} = \begin{bmatrix} cos(2\pi f(h,b)t_1) & -sen(2\pi f(h,b)t_1) \\ cos(2\pi f(h,b)t_2) & -sen(2\pi f(h,b)t_2) \\ \cdots & \cdots \\ cos(2\pi f(h,b)t_N) & -sen(2\pi f(h,b)t_N) \end{bmatrix} \begin{bmatrix} A_1\, cos(\phi_1) & A_2\, cos(\phi_2) & \cdots & A_J\, cos\big(\phi_J\big) \\ A_1 sen(\phi_1) & A_2 sen(\phi_2) & \cdots & A_J sen\big(\phi_J\big) \end{bmatrix} \tag{2.8}$$

Determinou-se os valores da matriz X a partir de (2.9) que representa o cálculo da pseudo-inversa da matriz β multiplicado pela matriz Y do sinal janelado.

$$X = (\beta^T \beta)^{-1} \beta^T Y \tag{2.9}$$

Somando o quadrado dos elementos da primeira linha com o quadrado dos elementos da segunda linha da matriz X obteve-se as amplitudes em cada janela.

$$\left(A_J \cos(\phi_J)\right)^2 + \left(A_J \, sen(\phi_J)\right)^2 = {A_J}^2 \tag{2.10}$$

2.1.3.4 Cálculo do PCP (LS)

Primeiramente definiu-se um vetor tempo que armazena o valor em segundos das amostras N. O PCP foi definido por uma matriz de 12 linhas e J colunas. No primeiro loop escolheu-se a nota musical, no segundo loop escolheu-se a oitava. Fixando por exemplo a nota C (b=1) e sua primeira oitava (h=1) determinou-se no final as energias da nota Dó na primeira oitava em cada janela do sinal.

Na primeira coluna da matriz β armazenou-se os valores $cos(2\pi f(h,b)t)$ e na segunda coluna os valores $-sen(2\pi f(h,b)t)$. A seguir determinou-se a matriz X de duas linhas e J colunas a partir do produto entre a pseudo-inversa da matriz β e a matriz Y (que contém o sinal divido em janelas). Por fim para cada linha (nota musical) da matriz PCP acumulou-se o resultado do PCP da iteração anterior com as somas dos quadrados das componentes da primeira linha e da segunda linha da matriz X. O processo foi realizado até completar a matriz PCP e assim ter um acorde por janela conforme descrito abaixo:

Algoritmo 4: Cálculo do PCP (LS)

1: $N \leftarrow f_s * (n/100)$; // Tamanho da Janela.

2: $J \leftarrow floor(L/N)$; // Número de Janelas.

3: $Y \leftarrow reshape(y(1{:}N*J), [N, J])$; // Redimensiona sinal para ter N linhas e J colunas.

4: W $\leftarrow window(eval(janela), N)$; // Função de janelamento.

5: $Y \leftarrow bsxfun(@times, W, Y)$; // Sinal é multiplicado ponto a ponto pela função de janelamento.

6: $t \leftarrow 0{:}\,1/f_s{:}\,(N-1)/f_s$; // Vetor tempo.

7: $PCP \leftarrow zeros(12, J)$; // Prelocação do PCP.

8: **Para b=1:12** // Nota musical.

9: **Para h=1:9** // Oitava.

10: $\beta \leftarrow [cos(2*\pi*f(h,b)*t)^T, -sen(2*\pi*f(h,b)*t)^T]$; // Matriz de cossenos e senos.

11: $X \leftarrow pinv(\beta) * Y$; // Cálculo dos Mínimos Quadrados.

12: $PCP(b,:) \leftarrow PCP(b,:) + X(1,:).\hat{}2 + X(2,:).\hat{}2$; // Cálculo do PCP.

13: **Fim**

14: **Fim**

2.1.3.5 Identificação dos Acordes (DFT e LS)

Para identificação dos acordes realizou-se a covariância entre os valores das linhas da matriz BC e das colunas da matriz PCP, obtendo assim em (2.11) a matriz C de classificação, contendo 24 linhas e J colunas. $\overline{BC}$ representa a média das linhas da matriz binária BC e $\overline{PCP}$, a média das colunas da matriz PCP.

$$C = \left(BC - \overline{BC}\right)\left(PCP - \overline{PCP}\right) \tag{2.11}$$

Em cada coluna J da matriz C foi determinada a linha que possui valor máximo e esses valores máximos foram armazenados em um vetor linha de J colunas. Cada elemento do vetor foi replicado N vezes. Por fim esse vetor linha foi comparado amostra por amostra com o vetor do gabarito, atribuindo 1 para valores iguais e 0 caso contrário. A Taxa de Acerto, TA, foi calculada a partir da média dos valores binários conforme descrito abaixo:

Algoritmo 3 e 5: Cálculo da Taxa de Acerto (TA) (DFT e LS)

1: $PCP \leftarrow bsxfun(@rdivide, PCP, max(PCP))$; // Normalização das colunas da matriz PCP.

2: $PCP \leftarrow bsxfun(@minus, PCP, mean(PCP))$; // Retira-se média das colunas da matriz PCP.

3: $C \leftarrow BC * PCP$; // Cálculo da matriz de classificação.

4: $[\sim, A] \leftarrow max\ (C)$; // Determina em cada coluna da matriz de classificação a linha em que ocorre valor máximo.

5: $CLASS\_PCP \leftarrow reshape(ones(N, 1) * A, [1,\ N * J])$; // Determina o vetor de classificação.

6: $TA(m,n) \leftarrow mean(GABARITO(1{:}N*J) == CLASS\_PCP)$; // Cálculo da Taxa de Acerto.

Com posse da matriz de Taxa de Acerto, TA, que possui 100 linhas (músicas) e 40 colunas (tamanho da janela) determinou-se a média de cada coluna e a seguir determinou-se o valor máximo dessas médias e sua respectiva coluna. Isso foi realizado para determinar o valor máximo da Taxa de Acerto, TA, e seu respectivo tamanho de janela em segundos.

2.1.4 Janelamento Variante com a Frequência (CQT e LSCQT)

Novamente em cada trecho foi aplicado os 17 tipos de janelas. O número de ciclos na janela foi definido como múltiplos de 17 e foi variado de 17 até 340. Em (2.12) observa-se que o tamanho da janela $N(h,b)$ em unidade de amostra depende das frequências musicais $f(h,b)$ da Tabela 1, da taxa de amostragem f_s e do número de ciclos Q na janela.

$$N(h,b) = \frac{f_s}{f(h,b)} Q \tag{2.12}$$

E o número de janelas para cada frequência musical foi determinada por:

$$J(h,b) = floor\left(\frac{L}{N(h,b)}\right). \tag{2.13}$$

Para cada frequência musical, o sinal foi representado em uma matriz de $N(h,b)$ linhas e $J(h,b)$ colunas e multiplicado por uma função de janelamento.

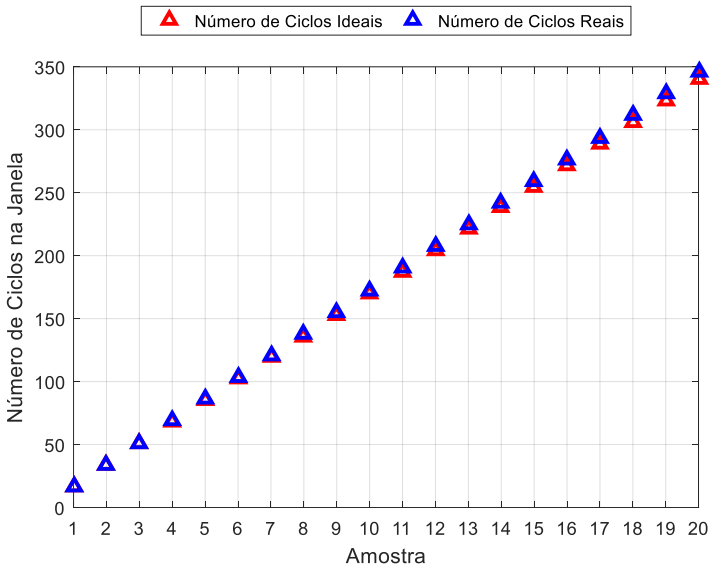
2.1.4.1 Número de Ciclos na Janela

O número de ciclos Q em (2.14) é sempre constante, independente da frequência musical adotada f_k. A resolução Δf_k depende do número de *bins* por oitava b e da frequência adotada (TODISCO, DELGADO e EVANS, 2017).

$$Q = \frac{f_k}{\Delta f_k} = \frac{f_k}{f_{k+1} - f_k} = \frac{f_k}{2^{1/b} f_k - f_k} = \frac{1}{2^{1/b} - 1} \tag{2.14}$$

Observa-se em (2.14) que para uma resolução de meio tom, o número de ciclos na janela é de aproximadamente 16,817, para um quarto de tom, 34,127, e assim sucessivamente até 345,747. Como foram adotados 20 valores inteiros de Q variando de 17 até 340 em múltiplos de 17, ocorrem erros entre os valores reais e ideais de número de ciclos na janela. A Figura 9 ilustra a comparação entre os valores reais de Q e os adotados.

Figura 9 – Número de ciclos ideais e número de ciclos reais.



Fonte: Autoria própria.

2.1.4.2 Cálculo da CQT

Com base na definição de produto escalar (LIPSCHUTZ e LIPSON, 2008), o método CQT pode ser interpretado como:

$$E = \mathrm{XY}, \tag{2.15}$$

na qual E representa o resultado do método CQT para cada janela $J(h,b)$ e $N(h,b)$ o tamanho da janela em unidade de amostra, Q o número de ciclos, $X = exp(-j2\pi Qn/N)$ o vetor linha de $N(h,b)$ colunas e Y a matriz do sinal janelado. Definindo $N = N(h,b)$ e $J = J(h,b)$ observa-se que na J - ésima janela do cálculo de E obtém-se:

$$E_J = x_0 y_{0,J} + x_1 y_{1,J} + \ldots + x_{N-1} y_{N-1,J} = \sum_{n=0}^{N-1} x_n y_{nJ} = \sum_{n=0}^{N-1} y_{nJ}\, exp\left(-j\frac{2\pi Qn}{N}\right). \tag{2.16}$$

A partir de (2.16) observa-se que os métodos CQT e DFT são parecidos, pois ambos podem ser determinados utilizando produto escalar. A Tabela 2 descreve algumas diferenças entre a transformada convencional DFT e a CQT. Observa-se que a resolução é constante no caso da transformada convencional, tornando a relação entre as frequências e *bins* linear. Já no caso do método CQT essa relação é exponencial, visto que a resolução aumenta conforme o aumento da frequência.

Tabela 2 – Comparação entre os métodos DFT e CQT.

Transformada	DFT	CQT
Frequência	$f_k = k\Delta f$	$f_k = 2^{k/b} f_{min}$
Tamanho da janela	N	$N_k = \frac{f_s Q}{f_k}$
Resolução	$\Delta f = \frac{f_s}{N}$	$\Delta f_k = \frac{f_k}{Q}$
Número de ciclos na janela	$k = \frac{f_k}{\Delta f}$	Q

Fonte: Adaptado de BROWN, 1991.

2.1.4.3 Cálculo do PCP (CQT)

A matriz PCP foi definida contendo 12 linhas (notas musicais) e L colunas (valor total de amostras do sinal). O primeiro *loop* determina a nota musical, o segundo *loop* determina a oitava em que se encontra a nota musical e o último *loop* determina em qual janela o valor da energia da nota musical será armazenado. Para cada iteração, determinam-se as amplitudes da CQT que é definido pelo produto escalar das exponencias complexas pela matriz do sinal janelado. A seguir em cada janela, a matriz PCP recebe o acúmulo de energia da iteração anterior com o valor absoluto ao quadrado da matriz E nessa mesma janela conforme descrito abaixo:

Algoritmo 6: Cálculo do PCP (CQT)

1: $N \leftarrow round(Q * (f_s./f))$; // Matriz Tamanho da Janela.

2: $J \leftarrow floor(L./N)$; // Matriz Número de Janelas.

3: $PCP \leftarrow zeros(12, L)$; // Prelocação do PCP.

4: Para b=1:12 // Nota musical.

5: **Para h=1:9** // Oitava.

6: $Y \leftarrow reshape\big(y(1{:}\,N(h,b) * J(h,b)), [N(h,b), J(h,b)]\big)$ // Redimensiona sinal para $N(h,b)$ linhas e $J(h,b)$ colunas.

7: W $\leftarrow window(eval(janela), N(h,b))$; // Função de janelamento.

8: $Y \leftarrow bsxfun(@times, W, Y)$; // Sinal é multiplicado ponto a ponto pela função de janelamento.

9: $E \leftarrow exp((-j * 2 * \pi * Q * (0{:}\,N(h,b) - 1))/N(h,b)) * Y$; // Cálculo da CQT.

10: $E \leftarrow abs(E).\hat{}2$; // Eleva ao quadrado ponto a ponto o módulo da CQT.

11: **Para c=1:J (h, b)** // Janela.

12: $PCP\big(b, (c-1) * N(h,b) + 1{:}\,c * N(h,b)\big) \leftarrow PCP\big(b, (c-1) * N(h,b) + 1{:}\,c * N(h,b)\big) + E(c)$; // Matriz PCP.

13: **Fim**

14: **Fim**

15: Fim

2.1.4.4 Cálculo do LSCQT

O modelo proposto para LSCQT foi determinado de maneira semelhante ao método LS. Definiu-se uma matriz β de $N(h,b)$ linhas e duas colunas, primeira com os valores $cos(2\pi f(h,b)t)$ e segunda com os valores $-sen(2\pi f(h,b)t)$. E definiu-se uma matriz X de duas linhas e $J(h,b)$ colunas contendo os valores desejados $A_J\, cos(\phi_J)$ e $A_J sen(\phi_J)$. A matriz X e os valores esperados E foram determinados com a mesma abordagem em (2.9) e (2.10) respectivamente.

2.1.4.5 Cálculo do PCP (LSCQT)

Primeiramente definiu-se um vetor tempo que armazena o valor em segundos das amostras $N(h,b)$. O PCP foi definido por uma matriz de 12 linhas e L colunas. O processo de estimação dos parâmetros desejados da matriz X foi semelhante ao da seção 2.1.3.4. A diferença ocorre no número de amostras $N(h,b)$ e no número de janelas $J(h,b)$ que variam conforme a frequência escolhida, assim como no cálculo do PCP pelo método CQT, conforme descrito abaixo:

Algoritmo 8: Cálculo do PCP (LSCQT)

1: $N \leftarrow round(Q * f_s./f)$; // Matriz Tamanho da Janela.

2: $J \leftarrow floor(L./N)$; // Matriz Número de Janelas.

3: $PCP \leftarrow zeros(12, L)$; // Prelocação do PCP:

4: Para b=1:12 // Nota musical.

5: **Para h=1:9** // Oitava.

6: $Y \leftarrow reshape\big(y(1{:}\,N(h,b) * J(h,b)), [N(h,b), J(h,b)]\big)$ // Redimensiona sinal para $N(h,b)$ linhas e $J(h,b)$ colunas.

7: W ← $window(eval(janela), N(h,b))$; // Função de janelamento.

8: $Y \leftarrow bsxfun(@times, W, Y)$; // Sinal é multiplicado ponto a ponto pela função de janelamento.

9: $t \leftarrow 0{:}\,1/f_s{:}\,(N(h,b)-1)/f_s$; // Vetor tempo.

10: $\beta \leftarrow [cos(2 * \pi * f(h,b) * t)^T, -sen(2 * \pi * f(h,b) * t)^T]$;// Matriz de cossenos e senos.

11: $X \leftarrow pinv(\beta) * Y$; // Cálculo dos Mínimos Quadrados.

12: $E \leftarrow X(1,:).\hat{}\,2 + X(2,:).\hat{}\,2$; // Cálculo das amplitudes.

13: **Para c=1:J (h, b)** // Janela.

14: $PCP\big(b, (c-1) * N(h,b) + 1{:}\,c * N(h,b)\big) \leftarrow PCP\big(b, (c-1) * N(h,b) + 1{:}\,c * N(h,b)\big) + E(c)$; // Matriz PCP.

15: **Fim**

16: **Fim**

17: Fim

2.1.4.6 Identificação dos Acordes (CQT e LSCQT)

Para identificação dos acordes realizou-se a covariância de maneira semelhante a seção 2.1.3.5, obtendo assim em (2.17) a matriz de classificação C, contendo 24 linhas e L colunas. $\overline{BC}$ representa a média das linhas da matriz binária BC e $\overline{PCP}$, a média das L colunas da matriz PCP.

$$C = \left(BC - \overline{BC}\right)\left(PCP - \overline{PCP}\right) \tag{2.17}$$

Em seguida foi determinada a linha que possui o valor máximo em cada coluna da matriz C. Esses valores máximos foram armazenados em um vetor linha de L colunas. Por fim esse vetor linha foi comparado amostra por amostra com o vetor do gabarito, atribuindo 1 para valores iguais e 0 caso contrário. A Taxa de Acerto, TA, foi calculada a partir da média dos valores binários conforme descrito abaixo:

Algoritmo 7 e 9: Cálculo da Taxa de Acerto (TA) (CQT e LSCQT)

1: $PCP \leftarrow abs(PCP(:,1:min\left(floor\left(L./N(:).*N(:)\right)\right)$; // Garante número mínimo de amostras para cada nota.

2: $PCP \leftarrow bsxfun(@rdivide, PCP, max(PCP))$; // Normalização das colunas da matriz PCP.

3: $PCP \leftarrow bsxfun(@minus, PCP, mean(PCP))$; // Retira-se média das colunas da matriz PCP.

4: $C \leftarrow BC * PCP$; // Cálculo da matriz de classificação.

5: $[\sim, CLASS\_PCP] \leftarrow max\ (C)$; // Determina a linha em que ocorre valor máximo.

6: $TA(m,q) \leftarrow mean(GABARITO(1:length(CLASS\_PCP) == CLASS\_PCP))$; // Cálculo da Taxa de Acerto.

Com posse da matriz de Taxa de Acerto, TA, que possui 100 linhas (músicas) e 20 colunas (número de ciclos) determinou-se a média de cada coluna e a seguir determinou-se o valor máximo dessas médias e sua respectiva coluna. Isso foi realizado para determinar o valor máximo da Taxa de Acerto, TA, e seu respectivo número de ciclos.

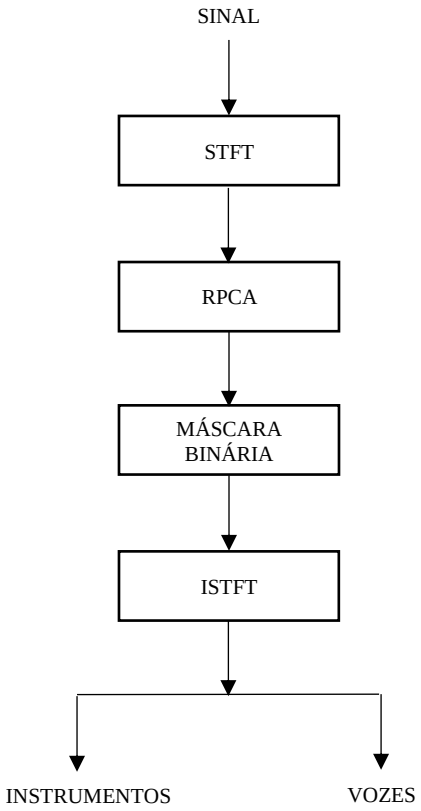
2.2 Teste em Música Real

A música analisada foi *Money That's What I Want-Beatles*, álbum *With The Beatles* presente no *dataset* do projeto *Supervised Chord Recognition for Music Audio in Matlab*, disponibilizado pelo grupo de pesquisa *Laboratory for the Recognition and Organization of Speech and Audio* (LabROSA), na pasta *beatles.zip*, código de 2009 (ELLIS, 2010). O áudio contém taxa de amostragem de 16 kHz e duração de 175,3589 s. Nesse mesmo *dataset* encontra-se um gabarito anotado por Christopher Harte (HARTE, 2007) contendo as faixas de tempo em segundos dos acordes da música. O gabarito contém duração de 168,6682 s com 93 características anotadas, incluindo presenças e ausências de acordes, logo possui Tempo Médio

do Acorde, TMA, de 1,8316 s. O tempo restante entre 168,6682 s e 175,3589 s corresponde a zona de silêncio dos espectrogramas (Figura 11). Essa zona de silêncio foi retirada do áudio no Algoritmo: Ajuste do Gabarito e do Áudio.

Para reduzir a influência da voz presente no áudio foi utilizado o algoritmo da Figura 10 (HUANG, CHEN, *et al.*, 2012a). O programa encontra-se disponível para *download* em (HUANG, CHEN, *et al.*, 2012b).

Figura 10 – Algoritmo para separação do áudio em conteúdos vocal e instrumental.



Fonte: Adaptado de HUANG, CHEN, *et al.*, 2012a.

A separação do áudio em conteúdo vocal e instrumental ocorreu primeiramente calculando o espectrograma utilizando *Short-Time Fourier Transform* (STFT). A seguir, utilizou-se o *Robust Principal Component Analysis*, RPCA (2.18) para separar o espectrograma M de tamanho m por n em duas matrizes *Low Rank* (L) e *Sparse* (S). O rank de uma matriz é definido como sendo o posto da matriz quando está na forma escalonada, ou seja, o número de linhas ou colunas não nulas após escalonamento (LAY, 1999). A matriz L é atribuída ao conteúdo instrumental que possui geralmente uma estrutura mais uniforme, já o espectrograma do conteúdo vocal é esparso, atribuído então a matriz S (HUANG, CHEN, *et al.*, 2012a).

$$\begin{cases} \min(\|L\|_* + \lambda\|S\|_1) \\ s.a \\ L + S = M \\ \lambda_k = \dfrac{k}{\sqrt{\max(m,n)}} \end{cases} \tag{2.18}$$

O valor de λ ajusta a quantidade de zeros presentes na matriz S, ou seja, quanto maior seu valor, ocorre menos interferência dos instrumentos no espectrograma das vozes, no entanto isso gera mais interferência das vozes no espectrograma dos instrumentos musicais (HUANG, CHEN, *et al.*, 2012a). Adotou-se k unitário. Para uma melhor separação utilizou-se a máscara binária (2.19).

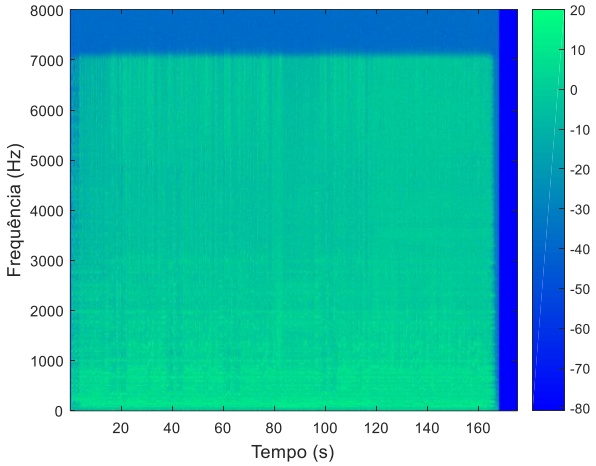
$$B(m,n) = \begin{cases} 1 & |S(m,n)| > G|L(m,n)| \\ 0 & |S(m,n)| \leq G|L(m,n)| \end{cases} \tag{2.19}$$

Na qual $B(m,n)$ é o valor da máscara binária, G representa um ganho arbitrário, $X_S(m,n) = B(m,n)M(m,n)$ e $X_L(m,n) = \big(1 - B(m,n)\big)M(m,n)$ são os pontos dos espectrogramas das vozes e dos instrumentos respectivamente. Adotou-se G unitário. Esse ganho possui função parecida com o parâmetro λ, ou seja, determina a quantidade de interferência das vozes no espectrograma dos instrumentos musicais e vice-versa (HUANG, CHEN, *et al.*, 2012a). A Figura 11 mostra a diferença entre os espectrogramas. Observa-se que no espectrograma dos instrumentos musicais as energias em dB são mais uniformes ao longo do tempo. Já no caso do espectrograma das vozes tem aparência esparsa, menos uniforme.

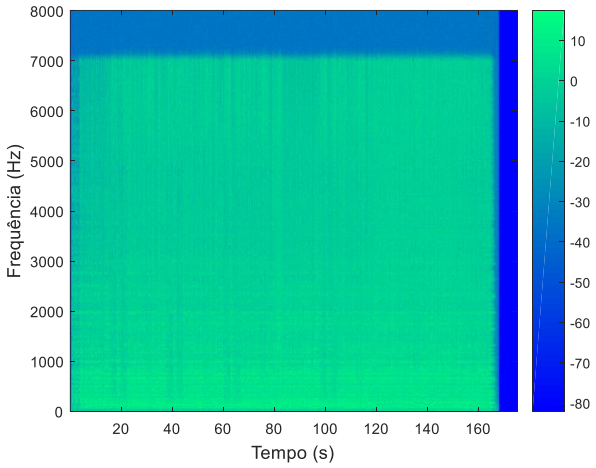
A partir dos espectrogramas do conteúdo vocal e do conteúdo instrumental, utilizou-se o método *Inverse Short-Time Fourier Transform* (ISTFT) para determinar ambos conteúdos no domínio do tempo.

Figura 11 – Espectrogramas em dB: (a) Áudio original, (b) Instrumentos musicais, (c) vozes.

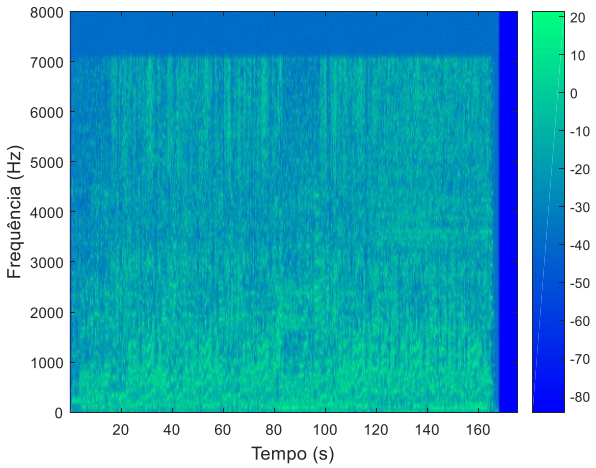
(a)



(b)



(c)



Fonte: Autoria própria.

Sabendo que os áudios original e de instrumentos musicais têm durações de 175,3589 s e o gabarito tem duração de 168,6682 s e contém acordes não tríades, foi necessário ajustar o tamanho dos mesmos utilizando o algoritmo abaixo:

Algoritmo: Ajustes dos Áudios e do Gabarito

1: $TABELA \leftarrow readtable('GABARITO.xlsx')$; // Leitura da tabela de 3 colunas (tempo inicial, tempo final e acordes).

2: $TABELA \leftarrow TABELA(2{:}end,:)$; // Para ler somente intervalos de tempos e acordes.

3: $G \leftarrow table2array(TABELA)$; // Converte tabela para vetor.

4: $[y_{ORIGINAL}, f_s] \leftarrow audioread('Money\_That\_s\_What\_I\_Want\_.mp3')$; // Leitura do áudio original.

5: $[y_{INSTRUMENTOS}, f_s] \leftarrow audioread('Money\_That\_s\_What\_I\_Want\_L.wav')$; // Leitura do áudio instrumentos.

6: $Y \leftarrow [\, y_{ORIGINAL}, y_{INSTRUMENTOS} \,]$; // Vetor que armazena áudios.

7: $AS \leftarrow G(:,3)$; // Seleciona terceira coluna de 93 acordes músicas. 0: ausência de acorde e não tríades.

8: $\Delta t \leftarrow G(:,2) - G(:,1)$; // Vetor intervalo de tempo.

9: $T \leftarrow sum(\Delta t)$; // Cálculo do tempo total do gabarito.

10: $TMA \leftarrow mean(\Delta t)$; // Cálculo do Tempo Médio do Acorde (TMA).

11: $QA \leftarrow round(T/TMA)$; // Cálculo da quantidade total de acordes (93 acordes).

12: $NA \leftarrow round(f_s * \Delta t^T)$; // Cálculo do vetor conjunto de amostras para cada acorde.

13: $GABARITO \leftarrow [\,]$; // Prélocação do vetor gabarito.

14: Para q=1:QA // Seleciona quantidade de acordes presentes na música.

15: $GABARITO \leftarrow \left[GABARITO,\ AS(q) * ones\left(1, NA(q)\right)\right]$; // Determina-se o vetor gabarito.

16: Fim

17: $binario \leftarrow zeros(1, QA)$; // Prélocação do vetor binário.

18: Para b=1:QA // Seleciona quantidade de acordes presentes na música.

19: **Se** $\boldsymbol{AS(b) \sim = 0}$ // Se acorde selecionado for não nulo, ou seja, uma tríade.

20: $binario(b) \leftarrow 1$; // Recebe 1: presença de acorde tríade.

21: **Fim**

22:Fim

23: $BS \leftarrow [\,]$; // Prélocação do vetor sinal binário

24: Para q=1:QA // Seleciona quantidade de acordes presentes na música.

25: $BS \leftarrow \left[BS,\ binario(q) * ones\left(1, N_A(q)\right)\right]$; // Determina-se o sinal binário.

26: Fim

27: $y \leftarrow Y(:,1)$; // Recebe aúdio original, ou $y \leftarrow Y(:,2)$ para o caso do áudio com conteúdo instrumental.

28: $y \leftarrow y^T$; // Transpõe vetor áudio.

29: $y \leftarrow y(1{:}size(BS,2))$; // Ajusta áudio para tamanho do sinal binário.

Com posse dos áudios e gabarito ajustados para os tamanhos corretos, realizou-se o Teste em Música Real a partir do algoritmo abaixo:

Teste em Música Real

1:Algoritmo: Ajustes do Gabarito e do Áudio;

2: Carrega a matriz binária de acordes e retira de cada linha 3/12;

3: Obtém-se o banco de frequências;

4: $L \leftarrow length(y)$; // Tamanho do sinal.

5: $JANELA \leftarrow \{"@barthannwin", \dots, "@tukeywin"\}$; // Vetor de *strings* com os 17 tipos de janelas;

6: Para m=1:1 // Um áudio de cada vez.

7: $q \leftarrow 1$; // Um número de ciclos na janela para os métodos CQT e LSCQT.

8: Para JJ=15:15 // Seleciona melhor função de janelamento do método DFT do Teste C.

9: $janela \leftarrow JANELA\{JJ\}$; // Variável auxiliar que recebe função de janelamento.

10: **Para n=18:18** // Seleciona melhor tamanho de janela para o método DFT do Teste C.

11: **Algoritmo 2: Cálculo do PCP (DFT);**

12: **Algoritmo 3: Cálculo da Taxa de Acerto (TA) (DFT);**

13: **Fim**

14:Fim

15: $JANELA \leftarrow \{"@barthannwin", \dots, "@tukeywin"\}$; // Vetor de *strings* com os 17 tipos de janelas.

16: Para JJ=15:15 // Seleciona melhor função de janelamento do método LS do Teste C.

17: $janela \leftarrow JANELA\{JJ\}$; // Variável auxiliar que recebe função de janelamento.

18: **Para n=12:12** //Seleciona melhor tamanho de janela para o método LS do Teste C.

19: **Algoritmo 4: Cálculo do PCP (LS);**

20: **Algoritmo 5: Cálculo da Taxa de Acerto (TA) (LS);**

21: **Fim**

22: Fim

23: $JANELA \leftarrow \{"@barthannwin", \dots, "@tukeywin"\}$; // Vetor de *strings* com os 17 tipos de janelas.

24: Para JJ=11:11 // Seleciona melhor função de janelamento do método CQT do Teste C.

25: $janela \leftarrow JANELA\{JJ\}$; // Variável auxiliar que recebe função de janelamento.

26: **Para Q=17:17** // Seleciona melhor Q do método CQT do Teste C.

27: **Algoritmo 6: Cálculo do PCP (CQT);**

28: **Algoritmo 7: Cálculo da Taxa de Acerto (TA) (CQT);**

29: **Fim**

30:Fim

31: $JANELA \leftarrow \{"@barthannwin", \dots, "@tukeywin"\}$; // Vetor de *strings* com os 17 tipos de janelas.

32: Para JJ=14:14 // Seleciona melhor função de janelamento do método LSCQT do Teste C.

33: $janela \leftarrow JANELA\{JJ\}$; // Variável auxiliar que recebe função de janelamento.

34: **Para Q=170:170** // Seleciona melhor Q do método LSCQT do Teste C.

35: **Algoritmo 8: Cálculo do PCP (LSCQT);**

36: **Algoritmo 9: Cálculo da Taxa de Acerto (TA) (LSCQT);**

37: **Fim**

38: Fim

39: Fim

No algoritmo Teste em Música Real, os vetores de classificação foram multiplicados ponto a ponto pelo sinal binário BS, anulando assim as sequencias de acordes não tríades, que não pertencem a classificação estabelecida na Figura 5. A seguir, as amostras nulas foram removidas dos vetores de classificação e foram comparados ponto a ponto com o gabarito. Determinou-se as taxas de acerto conforme os algoritmos abaixo:

Algoritmo 3 e 5: Cálculo da Taxa de Acerto (TA) (DFT e LS)

1: $PCP \leftarrow bsxfun(@rdivide, PCP, max(PCP))$; // Normalização das colunas da matriz PCP.

2: $PCP \leftarrow bsxfun(@minus, PCP, mean(PCP))$; // Retira-se média das colunas da matriz PCP.

3: $C \leftarrow BC * PCP$; // Cálculo da matriz de classificação.

4: $[\sim, A] \leftarrow max\ (C)$; // Determina em cada coluna da matriz de classificação a linha em que ocorre valor máximo.

5: $CLASS\_PCP \leftarrow reshape(ones(N, 1) * A, [1,\ N * J])$; // Determina o vetor de classificação.

6: $BS \leftarrow BS(1{:}\, length(CLASS\_PCP))$; // Ajusta tamanho do sinal binário para o tamanho do vetor de classificação.

7: $CLASS\_PCP \leftarrow CLASS\_PCP.* BS$; // Anula amostras dos acordes não tríades do vetor de classificação.

8: $CLASS\_PCP(BS == 0) \leftarrow [\,]$; // Remove amostras dos acordes não tríades do vetor de classificação.

9: $TA(m, n) \leftarrow mean(GABARITO == CLASS\_PCP)$; // Cálculo da Taxa de Acerto.

Algoritmo 7 e 9: Cálculo da Taxa de Acerto (TA) (CQT e LSCQT)

1: $PCP \leftarrow abs(PCP(:, 1{:}\, min\left(floor\left(L./N(:).* N(:)\right)\right))$; // Garante número mínimo de amostras para cada nota.

2: $PCP \leftarrow bsxfun(@rdivide, PCP, max(PCP))$; // Normalização das colunas da matriz PCP.

3: $PCP \leftarrow bsxfun(@minus, PCP, mean(PCP))$; // Retira-se média das colunas da matriz PCP.

4: $C \leftarrow BC * PCP$; // Cálculo da matriz de classificação.

5: $[\sim, CLASS\_PCP] \leftarrow max\ (C)$; // Determina a linha em que ocorre valor máximo.

6: $BS \leftarrow BS(1{:}\, length(CLASS\_PCP))$; // Ajusta tamanho do sinal binário para o tamanho do vetor de classificação.

7: $CLASS\_PCP \leftarrow CLASS\_PCP.* BS$; // Anula amostras dos acordes não tríades do vetor de classificação.

8: $CLASS\_PCP(BS == 0) \leftarrow [\,]$; // Remove amostras dos acordes não tríades do vetor de classificação.

9: $TA(m, q) \leftarrow mean(GABARITO == CLASS\_PCP)$; // Cálculo da Taxa de Acerto.

3 Resultados

3.1 Testes em Músicas Artificiais

3.1.1 Teste A

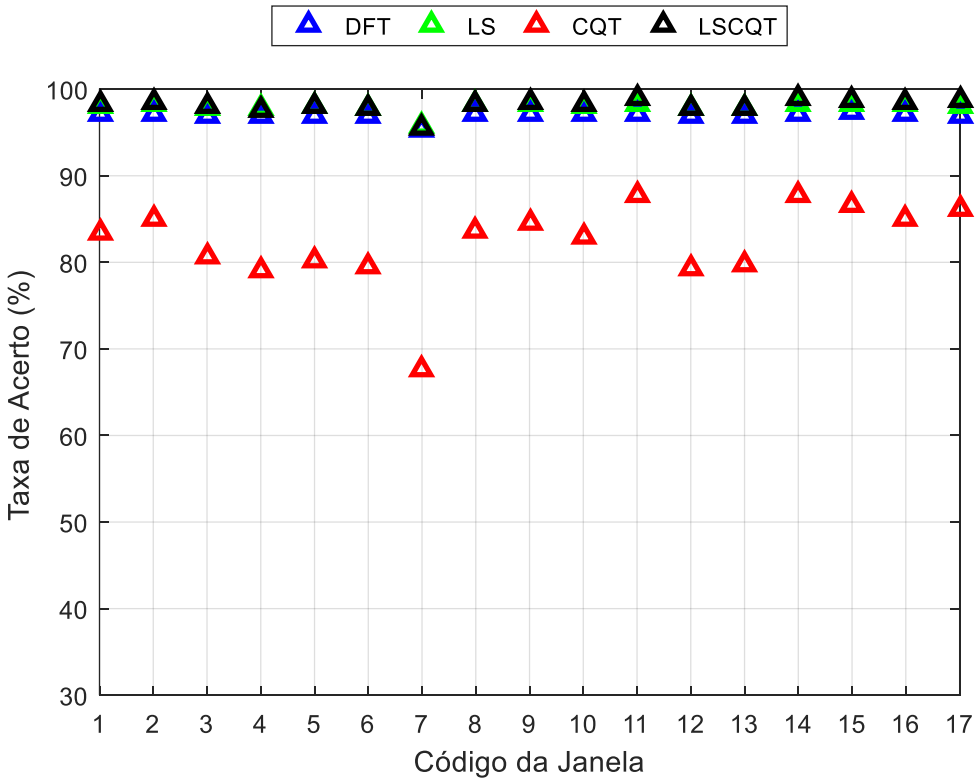
Observou-se que a melhor função de janelamento para os métodos DFT e LS, foi *Taylor*, com tamanhos de janela de 0,17 s e 0,09 s, respectivamente. Para o método CQT, destacou-se o tipo *Kaiser* considerando 17 ciclos na janela e no método LSCQT, o *Retangular*, considerando 68 ciclos na janela.

Tabela 3 – Teste A: Fs=44,1 kHz e TMA=1,8 s.

Código	Tipo de Janela	DFT		LS		CQT		LSCQT	
		Janela (s)	TA (%)	Janela (s)	TA (%)	Q	TA (%)	Q	TA (%)
1	*Bartlett-Hann*	0,17	97,09	0,11	98,15	17	83,48	102	98,34
2	*Bartlett*	0,17	97,13	0,09	98,31	17	85,06	85	98,53
3	*Blackman*	0,17	97,02	0,11	97,94	34	80,76	102	98,04
4	*Blackman-Harris*	0,19	96,92	0,13	97,75	34	79,18	119	97,68
5	*Bohman*	0,19	96,99	0,12	98,00	34	80,19	102	97,99
6	*Chebyshev*	0,17	96,94	0,13	97,82	34	79,46	102	97,78
7	*Flat Top*	0,23	95,20	0,21	95,81	51	67,53	221	95,64
8	*Gaussian*	0,19	97,12	0,1	98,27	17	83,73	85	98,39
9	*Hamming*	0,17	97,22	0,09	98,27	17	84,48	85	98,48
10	*Hanning*	0,19	97,03	0,11	98,10	17	82,97	102	98,29
11	*Kaiser*	0,17	97,11	0,1	98,18	17	87,74	68	98,96
12	*Nuttall and Blackman-Harris*	0,17	96,93	0,13	97,80	34	79,23	102	97,73
13	*Parzen and Gaussian*	0,19	96,95	0,13	97,82	34	79,81	119	97,78
14	*Rectangular*	0,17	97,10	0,08	98,17	17	87,70	68	98,96
15	*Taylor*	0,17	97,29	0,09	98,39	17	86,72	68	98,70
16	*Triangular*	0,17	97,20	0,09	98,23	17	84,95	85	98,48
17	*Tukey*	0,17	97,00	0,08	98,08	17	86,25	85	98,67
Taxa de Acerto Média (TAM)		-	96,96	-	97,95	-	82,31	-	98,14

Com relação a Taxa de Acerto, TA, observou-se que os métodos DFT, LS e LSCQT apresentaram valores superiores a 95% e pouca variação em função do janelamento. Enquanto que o método CQT, a Taxa de Acerto, TA, variou entre 67,53% a 87,74% e foi bastante influenciada pelo tipo de janelamento. Quanto ao tipo de janelamento aplicado, contatou-se que a janela 7 (*Flat Top*) obteve os piores resultados para todos os métodos.

Figura 12 – Teste A: Taxa de Acerto (%) em função do tipo de janela para cada método.



3.1.2 Teste B

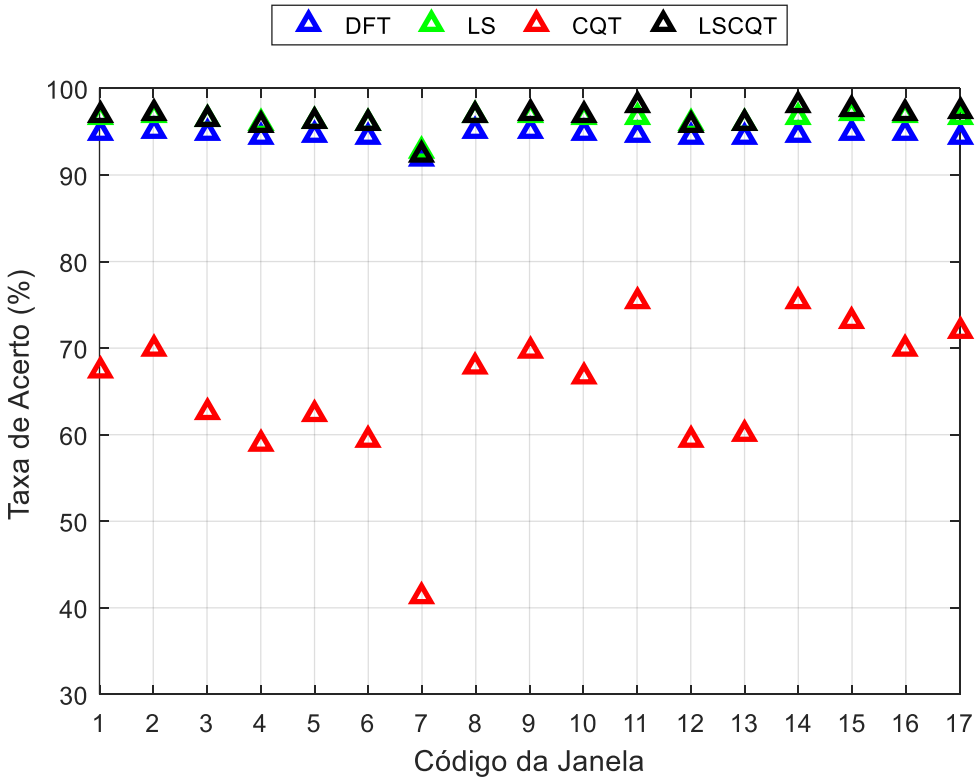
Reduzindo o Tempo Médio do Acorde, TMA, para 0,9 s, observou-se que para o método DFT a melhor função de janelamento foi *Gaussian* com tamanho de janela de 0,11 s. Novamente para o método LS, foi o tipo *Taylor* porém com tamanho de janela de 0,07 s. Também para o método CQT, foi mantido *kaiser* considerando 17 ciclos na janela e para o método LSCQT, o tipo *Rectangular* considerando 68 ciclos na janela.

Tabela 4 – Teste B: Fs=44,1 kHz e TMA=0,9 s.

Código	Tipo de Janela	DFT		LS		CQT		LSCQT	
		Janela (s)	TA (%)	Janela (s)	TA (%)	Q	TA (%)	Q	TA (%)
1	*Bartlett-Hann*	0,11	94,84	0,09	96,66	17	67,49	85	96,95
2	*Bartlett*	0,11	95,05	0,08	96,93	17	70,00	102	97,17
3	*Blackman*	0,11	94,76	0,09	96,42	17	62,61	102	96,42
4	*Blackman-Harris*	0,14	94,36	0,1	95,98	34	59,01	119	95,80
5	*Bohman*	0,11	94,63	0,09	96,28	34	62,40	102	96,28
6	*Chebyshev*	0,11	94,48	0,12	96,08	34	59,49	102	95,97
7	*Flat Top*	0,23	91,78	0,19	92,77	51	41,40	187	92,28
8	*Gaussian*	0,11	95,13	0,09	96,85	17	67,79	102	96,96
9	*Hamming*	0,11	95,10	0,08	96,91	17	69,68	85	97,18
10	*Hanning*	0,11	94,74	0,1	96,58	17	66,60	85	96,85
11	*Kaiser*	0,13	94,67	0,07	96,69	17	75,51	68	97,98
12	*Nuttall and Blackman-Harris*	0,14	94,39	0,1	96,02	34	59,29	119	95,85
13	*Parzen and Gaussian*	0,14	94,42	0,1	96,10	34	60,01	102	95,98
14	*Rectangular*	0,13	94,66	0,07	96,65	17	75,45	68	97,99
15	*Taylor*	0,12	94,93	0,07	97,03	17	73,13	85	97,55
16	*Triangular*	0,11	94,91	0,08	96,88	17	69,96	85	97,24
17	*Tukey*	0,16	94,48	0,08	96,69	17	72,09	68	97,43
Taxa de Acerto Média (TAM)		-	94,55	-	96,33		65,41	-	96,58

Considerando a Taxa de Acerto, TA, observou-se que os métodos DFT, LS e LSCQT apresentaram valores superiores a 90% e respostas similares em função do janelamento. O método CQT foi mais afetado pela redução do TMA, quando comparado ao Teste A e apresentou ampla variação de resposta em função do tipo de janela, entre 41,40% e 75,51%. Assim, foi bastante influenciado pelo tipo de janelamento. Com relação ao tipo de janela, a 7 (*Flat Top*) obteve os piores resultados para todos os métodos avaliados.

Figura 13 – Teste B: Taxa de Acerto (%) em função do tipo de janela para cada método.



3.1.3 Teste C

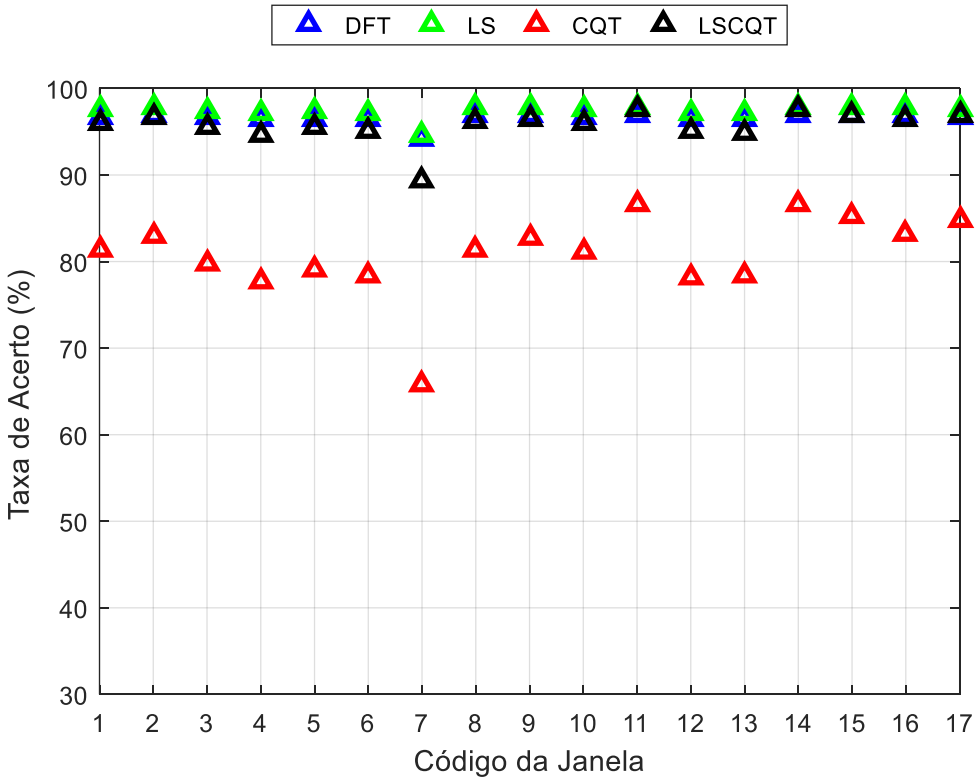
Reduzindo a taxa de amostragem para 16 kHz e mantendo TMA em 1,8 s, observou-se que para o método DFT a melhor função de janelamento foi *Taylor* com tamanho de janela de 0,18 s, para o método LS, *Taylor* com tamanho de janela de 0,12 s. Já para o método CQT, destacou-se o tipo *kaiser* considerando 17 ciclos na janela e para o método LSCQT, *Rectangular* considerando 170 ciclos na janela.

Tabela 5 – Teste C: Fs=16 kHz e TMA=1,8 s.

Código	Tipo de Janela	DFT		LS		CQT		LSCQT	
		Janela (s)	TA (%)	Janela (s)	TA (%)	Q	TA (%)	Q	TA (%)
1	*Bartlett-Hann*	0,18	96,76	0,13	97,62	34	81,39	255	96,09
2	*Bartlett*	0,18	96,96	0,12	97,81	17	83,07	221	96,58
3	*Blackman*	0,18	96,58	0,15	97,39	34	79,77	289	95,59
4	*Blackman-Harris*	0,23	96,35	0,16	97,10	34	77,71	340	94,73
5	*Bohman*	0,18	96,54	0,15	97,40	34	78,97	272	95,49
6	*Chebyshev*	0,23	96,40	0,15	97,15	34	78,31	340	95,09
7	*Flat Top*	0,34	94,28	0,34	94,62	51	65,88	340	89,28
8	*Gaussian*	0,18	96,93	0,12	97,73	17	81,43	272	96,31
9	*Hamming*	0,18	96,88	0,12	97,74	17	82,64	255	96,45
10	*Hanning*	0,19	96,69	0,13	97,53	34	81,21	255	95,97
11	*Kaiser*	0,18	96,97	0,11	97,80	17	86,59	170	97,54
12	*Nuttall and Blackman-Harris*	0,23	96,38	0,15	97,10	34	78,12	340	95,00
13	*Parzen and Gaussian*	0,22	96,41	0,16	97,18	34	78,44	340	94,94
14	*Rectangular*	0,18	96,96	0,11	97,80	17	86,51	170	97,54
15	*Taylor*	0,18	96,97	0,12	97,87	17	85,30	187	96,94
16	*Triangular*	0,18	96,92	0,13	97,81	17	83,18	204	96,45
17	*Tukey*	0,18	96,80	0,13	97,66	17	84,73	221	96,90
Taxa de Acerto Média (TAM)		-	96,58	-	97,37	-	80,78	-	95,70

Com relação a Taxa de Acerto, TA, observou-se que os métodos DFT, LS e LSCQT permaneceram superiores a 85%, e pouca variação em função do janelamento. Enquanto que o método CQT, a Taxa de Acerto, TA, variou entre 65,88% a 86,59%. Isso mostra que com relação ao Teste A, a redução da taxa de amostragem mantendo o mesmo TMA influenciou pouco nos valores de Taxa de Acerto, TA. Novamente a janela 7 (*Flat Top*) apresentou os piores resultados.

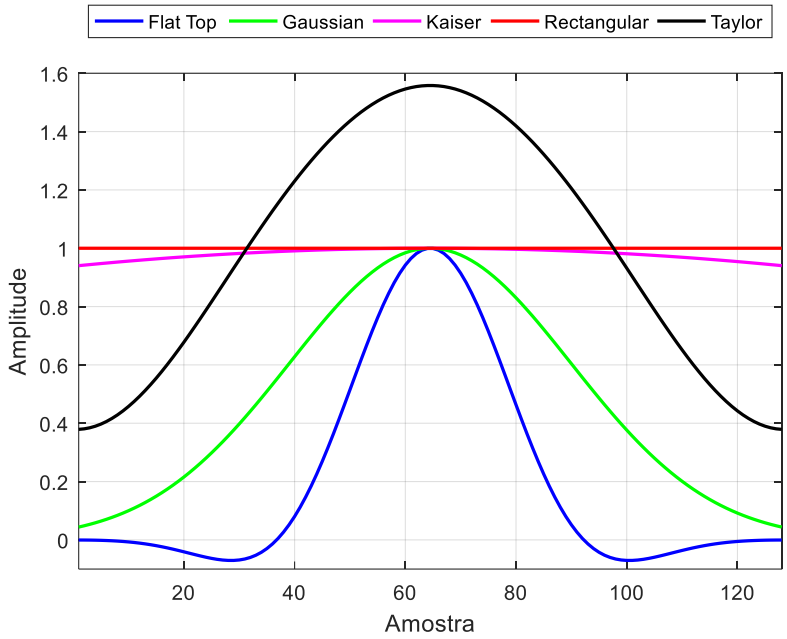
Figura 14 – Teste C: Taxa de Acerto (%) em função do tipo de janela para cada método.



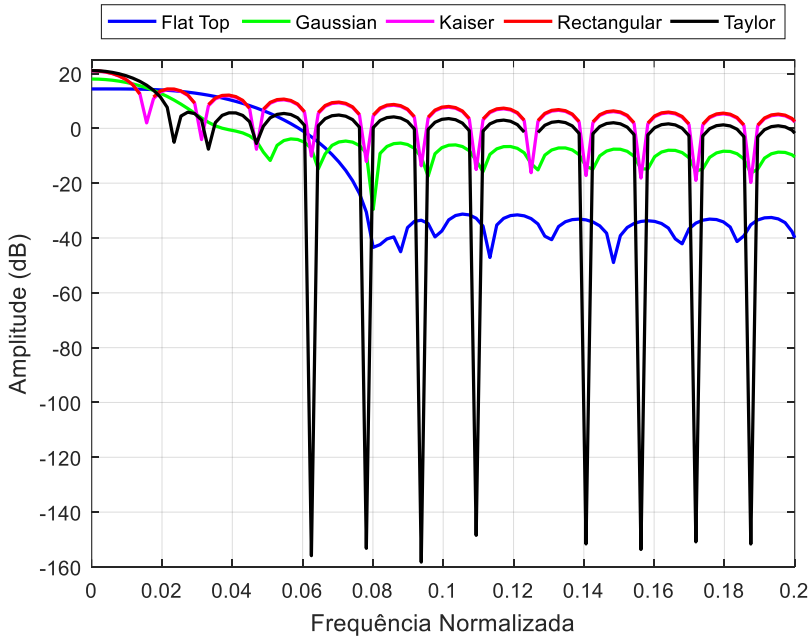
A partir da Figura 15, observa-se que no domínio do tempo a janela *Flat Top* tem a banda principal mais estreita, logo no domínio da frequência tem um lóbulo principal mais largo. Isso indica pior resolução no domínio da frequência. Além disso, tem os lóbulos laterais mais atenuados indicando que lida melhor com vazamento espectral. Porém, nos testes com músicas artificiais, as ondas senoidais possuem três tons, sendo necessária uma resolução melhor na frequência para distinguir as energias de cada nota musical, ou seja, é necessário lóbulo principal mais estreito. Observou-se que as janelas 8 (*Gaussian)*, 11 (*Kaiser*), 14 (*Rectangular*) e 15 (*Taylor*) quando comparadas com a 7 (*Flat Top*) tem lóbulos principais estreitos, logo melhor resolução na frequência.

Figura 15 – Janelas: (a) Domínio do tempo. (b) Domínio da frequência.

(a)

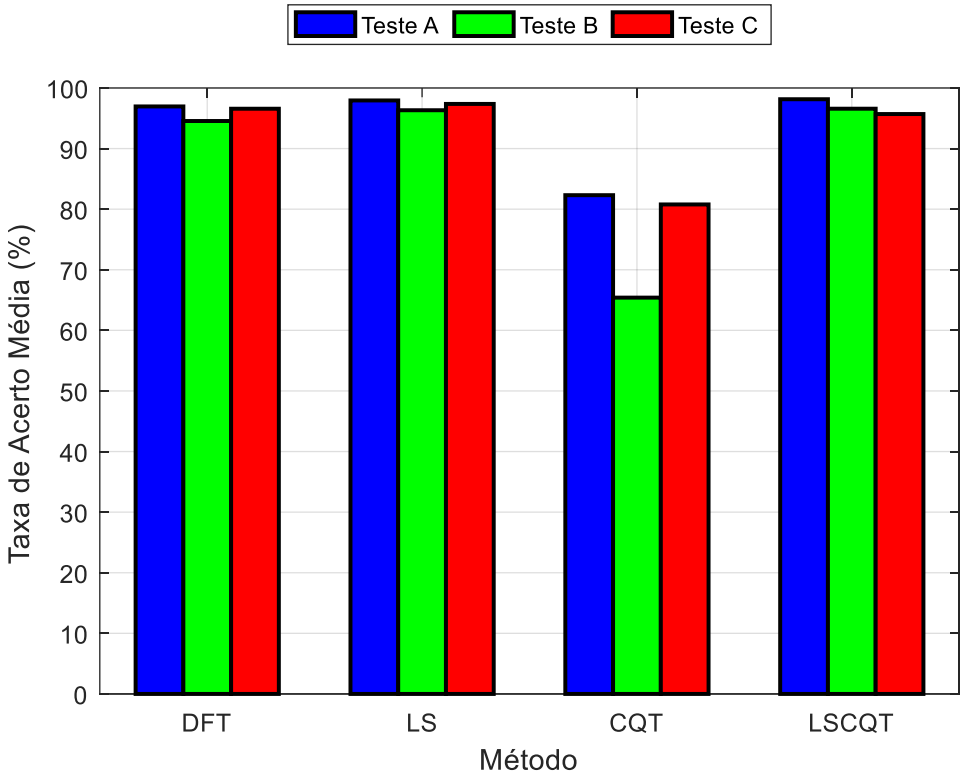


(b)



Pela Figura 16, observou-se que houve pouca diferença entre os testes considerando os métodos DFT, LS e LSCQT. Já o método CQT obteve menor Taxa de Acerto Média, TAM. Na seção 2.1.4.1 observa-se que nesse método os números de ciclos por janela foram adotados como números inteiros e que na realidade não são.

Figura 16 – Testes em músicas artificiais.

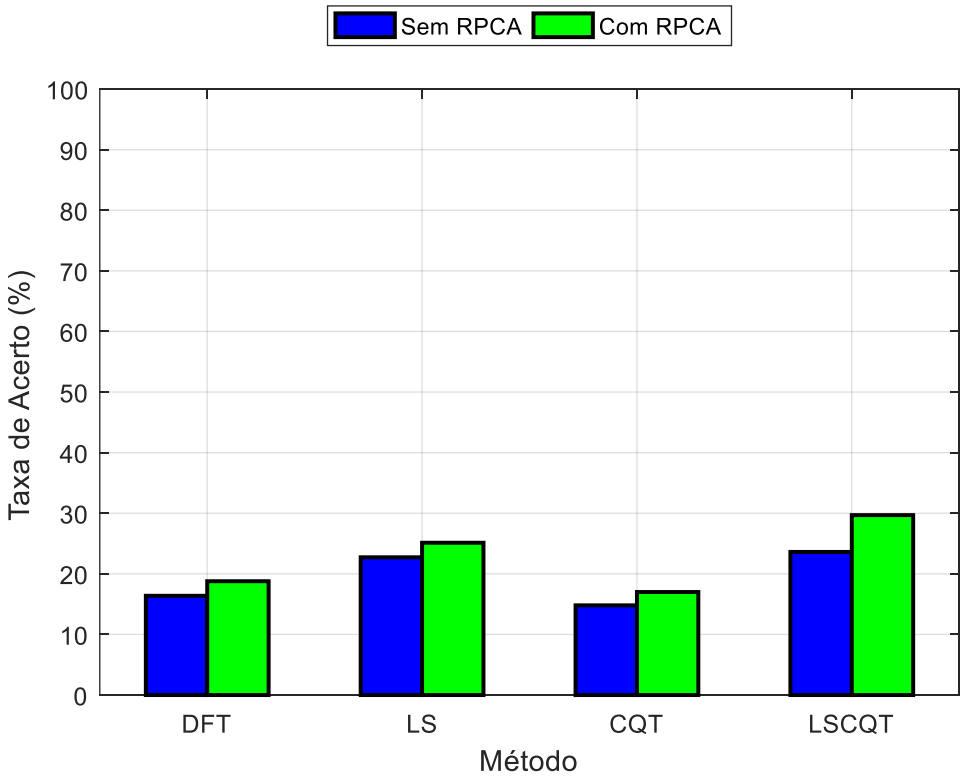


3.2 Teste em Música Real

Para os métodos DFT e LS utilizou-se a janela 15 (*Taylor*) com 0,18 s e com 0,12 s respectivamente. Já para os métodos CQT e LSCQT utilizou-se a janela 11 (*kaiser*) com 17 ciclos e a janela 14 (*Rectangular*) com 170 ciclos respectivamente.

Pela Figura 17, observou-se que ocorreu grande redução na Taxa de Acerto, TA, em cada método devido a segmentação do áudio em trechos de tamanho fixo, ou seja, diferente do intervalo real de cada acorde. Além disso, houve uma melhoria na Taxa de Acerto, TA, com a separação da voz do conteúdo instrumental, principalmente para o método LSCQT. Apesar disso, essa separação não foi perfeita, apresentando resquícios de conteúdo vocal no áudio instrumental, o que influenciou nos resultados. O método CQT foi inferior no Teste em Música Real. Para os métodos DFT, LS, CQT, LSCQT, os valores de TA, sem influência do RPCA foram respectivamente: 16,38%, 22,73%, 14,81% e 23,62%. E com influência do RPCA: 18,78%, 25,13%, 17,00% e 29,69%.

Figura 17 – Teste em música real.



4 Conclusões

Este trabalho propôs o método *Least Squares Constant Q Transform*, LSCQT, como etapa de mapeamento de frequências no processo de identificação de acordes musicais e compará-lo com os métodos DFT, LS e CQT. Para isso foram realizados Testes em Músicas Artificiais e Teste em Música Real.

Nos Testes em Músicas Artificiais observou-se redução nas taxas de acerto: ao fixar a taxa de amostragem e reduzir o Tempo Médio do Acorde, TMA, e ao fixar o TMA e reduzir a taxa de amostragem. E no Teste em Música Real verificou-se uma melhoria no desempenho de todos os métodos com a utilização do método *Robust Principal Component Analysis*, RPCA.

Para os Testes em Músicas Artificias, o método LSCQT obtém melhor Taxa de Acerto Média, TAM, nos Testes A e B. Já no Teste C, ocorre com a utilização do método LS. No Teste em Música Real sugere-se que método LSCQT seja o mais indicado para identificação de acordes.

Como trabalhos futuros propõe-se para o Teste em Música Real: valores adequados para os parâmetros λ e G, a utilização do algoritmo *Beat Tracking* com o intuito de segmentar a música em trechos de tamanhos corretos, a utilização do algoritmo *Key Detection* para determinar o campo harmônico musical e um estudo comparativo entre métodos de correspondência de modelo e métodos de aprendizagem para classificação de acordes musicais.

Referências Bibliográficas

BELLO, J. P.; PICKENS, J. **A Robust Mid-level Representation for Harmonic Content in Music Signals**. In: International Society for Music Information Retrieval Conference (ISMIR). London, UK: [s.n.]. 2005. p. 304-311.

BENWARD, B.; SAKER, M. **Music in Theory and Pratice**. 8ª. ed. New York: McGraw-Hill, v. 1, 2008.

BROWN, J. C. Calculation of a constant Q spectral transform. **The Journal of the Acoustical Society of America**, v. 89, n. 1, p. 425-434, 1991.

CHIOYE, L.; KAY, A.; LISKA, P. Fast Fourier Transforms (FFTs) and Windowing. **Texas Instruments - TI training & videos**, 2017. Disponivel em: <https://training.ti.com/ti-precision-labs-adcs-fast-fourier-transforms-ffts-and-windowing?keyMatch=FFT%20WINDOWING&tisearch=Search-EN-everything>. Acesso em: 8 dez. 2019.

DELL'AVERSANA, P.; GABBRIELLINI, G.; AMENDOLA, A. Sonification of geophysical data through time–frequency analysis: theory and applications. **Geophysical Prospecting**, v. 65, n. 1, p. 146-157, 2016.

ELLIS, D. P. W. Beat tracking by dynamic programming. **Journal of New Music Research**, v. 36, n. 1, p. 51-60, 2007.

ELLIS, D. P. W. Supervised Chord Recognition for Music Audio in Matlab. **LabROSA - Projects**, 2010. Disponivel em: <https://labrosa.ee.columbia.edu/projects/chords/>. Acesso em: 30 jul. 2019.

EVEREST, F. A..; POHLMANN, K. C. **Master Handbook of Acoustics**. 5ª. ed. [S.l.]: McGraw-Hill, 2009.

FUJISHIMA, T. **Realtime Chord Recognition of Musical Sound:** a System Using Common Lisp Music. In: International Computer Music Conference (ICMC). Beijing, China: [s.n.]. 1999. p. 464-467.

GOOSSENS, S. Applying spectral leakage corrections to gravity field determination from satellite tracking data. **Geophysical Journal International**, v. 181, n. 3, p. 1459–1472, 2010.

HARTE, C. MIR Research. **mir-rseaech.blogspot**, 2007. Disponivel em: <http://mir-research.blogspot.com/2007/09/beatles-chord-transcriptions.html>. Acesso em: 30 jul. 2019.

HUANG, P.-S. et al. **Singing-voice separation from monaural recordings using robust principal component analysis**. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. Kyoto, Japan: IEEE. 2012a. p. 57-60.

HUANG, P.-S. et al. Singing-voice separation from monaural recordings using robust principal component analysis. **GitHub**, 2012b. Disponivel em: <https://github.com/posenhuang/singingvoiceseparationrpca>. Acesso em: 26 fev. 2020.

INGLE, A. N.; SETHARES, W. A. The least-squares invertible constant-Q spectrogram and its application to phase vocoding. **The Journal of the Acoustical Society of America**, v. 132, n. 2, p. 894-903, 2012.

ISO. Acoustics - Standard tuning frequency (Standard musical pitch). **ISO 16**, 1975. Disponivel em: <https://www.iso.org/standard/3601.html>. Acesso em: 10 dez 2019.

JWO, D.-J.; WU, I.-H.; CHANG, Y. **Windowing Design and Performance Assessment for Mitigation of Spectrum Leakage**. In: E3S Web of Conferences. [S.l.]: EDP Sciences. 2019. p. 1-8.

LATHI, B. P. **Sinais e Sistemas Lineares**. 2ª. ed. Porto Alegre: Bookman, 2006.

LAY, D. C. **Álgebra Linear e suas Aplicações**. 2ª. ed. Rio de Janeiro: LTC-Livros Técnicos e Científicos , 1999.

LEE, K. **Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile**. In: International Computer Music Conference (ICMC). New Orleans, LA, USA: [s.n.]. 2006.

LIPSCHUTZ, S.; LIPSON, M. L. **Linear Algebra (Schaum's Outlines)**. 4ª. ed. [S.l.]: McGraw-Hill, 2008.

MARTINO, M.; LOSITO, R.; MASI, A. Analytical metrological characterization of three-parameter sine fit algorithm. **ISA Transactions**, v. 51, n. 2, p. 262-270, 2012.

MATHWORKS. Window. **MathWorks develops, sells, and supports MATLAB and Simulink products**, 2020a. Disponivel em: <https://www.mathworks.com/help/signal/ref/window.html>. Acesso em: 30 jul 2019.

MATHWORKS. Fast Fourier transform. **MathWorks develops, sells, and supports MATLAB and Simulink products**, 2020b. Disponivel em: <https://www.mathworks.com/help/matlab/ref/fft.html>. Acesso em: 30 jul 2019.

MAUCH, M.; DIXON, S. **Aproximate Note Transcription for the Improvement Identification of Difficult Chords**. In: International Society for Music Information Retrieval Conference (ISMIR). Utrecht, The Netherlands: [s.n.]. 2010. p. 135-140.

ORTIZ-ECHEVERRI, C. J.; RODRÍGUEZ-RESÉNDIZ, J.; GARDUÑO-APARICIO, M. An approach to STFT and CWT learning through music hands-on labs. **Computer Applications in Engineering Education**, v. 26, n. 6, p. 2026-2035, 2018.

RAO, Z.; GUAN, X.; TENG, J. Chord recognition based on temporal correlation support vector machine. **Applied Sciences**, v. 6, p. 1-14, 2016.

SCHÖRKHUBER, C.; KLAPURI, A. **Constant-Q Transform Toolbox for Music Processing**. In: 7th Sound and Music Computing Conference. Barcelona, Spain: [s.n.]. 2010. p. 3-64.

TODISCO, M.; DELGADO, H.; EVANS, N. Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. **Computer Speech & Language**, v. 45, p. 516-535, 2017.

ZENZ, V.; RAUBER, A. **Automatic Chord Detection Incorporating Beat and Key Detection**. In: IEEE International Conference on Signal Processing and Communications. Dubai, United Arab Emirates: IEEE. 2007. p. 1175-1178.