

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

WILSON LUIZ PIASECKI FILHO

**ESTUDO COMPARATIVO ENTRE OS ALGORITMOS DE  
PROCESSAMENTO DE ÁUDIO *PHASE VOCODER* E *SPECTRAL  
MODELING SYNTHESIS***

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2011

WILSON LUIZ PIASECKI FILHO

**ESTUDO COMPARATIVO ENTRE OS ALGORITMOS DE  
PROCESSAMENTO DE ÁUDIO *PHASE VOCODER* E *SPECTRAL*  
*MODELING SYNTHESIS***

Trabalho de conclusão de curso de graduação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Sistemas para Internet do Departamento Acadêmico de Informática (DAINF) da Universidade Tecnológica Federal da Paraná, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora: Prof. Dra. Leyza Baldo Dorini

Co-orientador: Prof. Dr. André Luís Vignatti

CURITIBA

2011

## FOLHA DE APROVAÇÃO

Trabalho de Diplomação

### **ESTUDO COMPARATIVO ENTRE OS ALGORITMOS DE PROCESSAMENTO DE ÁUDIO *PHASE VOCODER* E *SPECTRAL MODELING SYNTHESIS***

por

**Wilson Luiz Piasecki Filho**

Esta dissertação foi apresentada às \_\_\_\_\_ horas do dia 14 de Setembro de 2011 como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet na Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo-assinados. Após deliberação a banca considerou o trabalho \_\_\_\_\_

\_\_\_\_\_  
(aprovado, aprovado com restrições, ou reprovado)

\_\_\_\_\_  
Prof. Msc. Robson Ribeiro Linhares

\_\_\_\_\_  
Profa. Dra. Leyza Baldo Dorini

(Orientadora)

\_\_\_\_\_  
Prof. Dr. Murilo Vicente Gonçalves da Silva

\_\_\_\_\_  
Prof. Dr. André Luís Vignatti

(Co-orientador)

Visto da coordenação:

\_\_\_\_\_  
Profa. Msc. Wania Meira Matos Figueiredo  
Coordenadora do curso Tecnologia em Sistemas para  
Internet

## **AGRADECIMENTOS**

Agradeço aos meus orientadores pelo tempo, pela paciência e por todo o auxílio prestado na elaboração e na execução deste projeto, aos meus pais pelo apoio e incentivo às atividades acadêmicas, aos meus empregadores que me permitiram flexibilizar horário para cumprir com as atividades deste projeto e aos colegas do coral e do madrigal UTFPR pelo apoio e pelas risadas.

Agradeço à UTFPR, aos professores, coordenadores, técnicos administrativos e todos que tornaram meu curso possível.

Agradeço a Deus por tudo.

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>9</b>
1.1 ESTRUTURA DO TRABALHO.....	9
1.2 JUSTIFICATIVA DA ESCOLHA DO TEMA.....	10
1.3 DEFINIÇÕES.....	10
<b>2 PROCESSAMENTO DIGITAL DE SINAIS DE ÁUDIO.....</b>	<b>13</b>
2.1 DEFINIÇÕES.....	13
2.1.1 Taxa de Amostragem.....	13
2.1.2 Aliasing.....	14
2.1.3 Time-stretch.....	14
2.1.4 Pitch-shift.....	15
2.2 ALGORITMOS NO DOMÍNIO DA FREQUÊNCIA.....	15
2.2.1 Transformada de Fourier.....	15
2.2.2 Transformada Inversa de Fourier.....	15
2.2.3 Discrete Fourier Transform (DFT).....	16
2.2.4 Fast Fourier Transform (FFT).....	17
2.2.5 Short Time Fourier Transform (STFT).....	18
2.2.6 Janelas.....	18
2.2.7 Phase Vocoder.....	19
2.2.7.1 Heterodyning.....	20
2.2.7.2 Aplicação do filtro.....	20
2.2.7.3 Conversão de retangular para polar.....	20
2.2.7.4 Cálculo da fase (unwrapping).....	21
2.2.7.5 Calculo da taxa de rotação angular.....	22
2.2.7.6 Síntese.....	22
2.2.8 SMS.....	23
2.2.8.1 Cálculo da magnitude do espectro.....	23
2.2.8.2 Janela de análise.....	23
2.2.8.3 Cálculo da Transformada de Fourier.....	24
2.2.8.4 Tamanho da sobreposição.....	24
2.2.8.5 Detecção de Picos.....	24
2.2.8.6 Continuação de Picos.....	24
2.2.8.7 Guias.....	25
2.2.8.8 Representação da parte Determinística.....	25
2.2.8.9 Síntese Determinística.....	25
2.2.8.10 Processamento da Parte Estocástica.....	25
2.2.8.11 Representação da Parte Estocástica.....	26
2.2.8.12 Síntese Estocástica.....	26
<b>3 COMPARATIVO DOS ALGORITMOS.....</b>	<b>27</b>
3.1 AVALIAÇÃO QUANTITATIVA.....	27
3.1.1 Onda Quadrada.....	28
3.1.2 Onda Triangular.....	28
3.1.3 Onda Dente de Serra.....	28
3.2 AVALIAÇÃO QUALITATIVA.....	29
<b>4 RESULTADOS.....</b>	<b>33</b>
4.1 ANÁLISE QUANTITATIVA.....	33
4.2 ANÁLISE QUALITATIVA.....	36
4.2.1 Drums .....	37

4.2.2 Recit.....	37
4.2.3 Violin .....	38
4.2.4 Resultados do Questionário.....	38
<b>5 CONCLUSÃO.....</b>	<b>44</b>
<b>REFERÊNCIAS.....</b>	<b>46</b>
<b>ANEXOS.....</b>	<b>49</b>

## RESUMO

Este trabalho apresenta uma análise comparativa baseada em critérios qualitativos e quantitativos entre os algoritmos de processamento de áudio no domínio da frequência *Phase Vocoder* e *Spectral Modeling Synthesis* (SMS). A partir de ondas sintéticas, foram aplicados os algoritmos nas operações de *time-stretch* e *pitch-shift*. Inicialmente, efetuou-se a diferença entre as ondas originais e processadas, visando avaliar quantitativamente o impacto de cada algoritmo. Posteriormente, foram selecionadas amostras correspondendo a casos de interesse, tal como presença de instrumentos de percussão, por exemplo. Após a aplicação dos algoritmos, tais amostras foram apresentadas a voluntários, que avaliaram sua qualidade através de questionário. Pode-se perceber que o SMS produziu amostras com uma maior quantidade de artefatos (falhas) que o *Phase Vocoder*, recebendo uma avaliação negativa dos voluntários e também gerando ondas ruidosas ao ser aplicado nas ondas sintéticas. Como trabalhos futuros, sugere-se a utilização do SMS utilizando a sua capacidade de análise e geração de efeitos em sons mais simples.

Palavras-chave: SMS. *Phase Vocoder*. Transformada de Fourier. Processamento Digital de Áudio.

## **ABSTRACT**

*This work presents a comparative analysis based in qualitative and quantitative criteria between the frequency domain audio processing algorithms Phase Vocoder and Spectral Modeling Synthesis (SMS). The algorithms were applied on synthesized waves in time-stretch and pitch-shift operations. Initially, the difference between the original and processed waves was calculated seeking evaluate quantitatively the impact of each algorithm. Later, new samples were selected for interest cases, like percussive instruments, for example. After the application of the algorithms, each sample was shown to volunteers, which evaluated it using a questionnaire. It was notable that the SMS' samples had a larger quantity of artifacts (defects) than Phase Vocoder's, receiving a worst evaluation, also generating noisier waves when applied in the synthesized waves. For future work, it's suggested using the SMS in it's capabilities of analysis and generation of effects in simpler sounds.*

*Keywords: SMS. Phase Vocoder. Fourier Transform. Digital Audio Processing.*



## 1 INTRODUÇÃO

Na manipulação de arquivos de áudio, duas operações que podem ser efetuadas são o *time-stretch*, que possibilita alterar a duração de um som, e o *pitch-shift*, que altera suas frequências. Tais operações podem ser utilizadas por estudantes de música, estúdios e compositores, entre outros. Este trabalho tem como objetivo efetuar um breve estudo comparativo entre dois algoritmos de processamento de áudio que possuem esta funcionalidade: o *Phase Vocoder* (FLANAGAN, 1966) e o *Spectral Modeling Systems* (SMS) (SERRA, 1990).

O *Phase Vocoder* foi proposto em 1966 e é considerado adequado para se trabalhar com áudio, em especial para as operações de *time-stretch* e *pitch-shift* (LAROCHE, 1999). Contudo, é um algoritmo que apresenta algumas falhas em determinados tipos de áudio conforme Serra (1990).

O algoritmo PARSHL busca minimizar estas falhas do *Phase Vocoder* permitindo, por exemplo, a captura de parciais, sons que existem principalmente em instrumentos percussivos, mas que o *Phase Vocoder* tem dificuldade em detectar. O SMS, por sua vez, busca trabalhar com o mesmo conceito do PARSHL, porém buscando analisar o som ainda mais detalhadamente. Foi desenvolvido em 1989 e também será utilizado nas comparações deste estudo.

Os algoritmos serão comparados de forma qualitativa e quantitativa. A primeira será efetuada através da análise subjetiva, por parte de voluntários, de amostras de áudio processadas com os algoritmos. O comparativo quantitativo se dará através da síntese de ondas sintéticas nos formatos dente-de-serra, quadrada e triangular. Tais ondas serão processadas pelos algoritmos e sobrepostas às ondas originais invertidas verticalmente, então serão somados os valores de cada amostra, permitindo quantificar os artefatos criados.

### 1.1 ESTRUTURA DO TRABALHO

Este documento está dividido em:

1. Revisão da literatura para tópicos relevantes:
  1. Características do som;
  2. Características do áudio digital e do processamento de áudio digital;
  3. Transformada de Fourier;

4. Janelamento;
5. *Phase Vocoder*;
6. SMS;
2. Metodologia de comparação:
  1. Questionário;
  2. Sobreposição das ondas;
3. Resultados:
  1. Análise quantitativa;
  2. Análise qualitativa;
4. Conclusões

## 1.2 JUSTIFICATIVA DA ESCOLHA DO TEMA

Operações de *time-stretch* e *pitch-shift* permitem trabalhar com a frequência e a duração do áudio de forma independente. Estúdios utilizam estas funcionalidades a fim de alongar um determinado trecho cuja duração ficou menor, ou maior, que a esperada, corrigir afinação de notas, estudantes de música podem utilizar para reduzir a velocidade de um áudio e pode-se compor uma nova música juntando trechos com tempos diferentes, entre outras aplicações. Neste contexto, interessa a qualidade do resultado do processamento pelos algoritmos. O *Phase Vocoder* é um algoritmo adequado para trabalhar com operações de *time-stretch* e *pitch-shift* em áudio, além de permitir outros efeitos (LAROCHE, 1999). Este algoritmo foi desenvolvido em 1966 (FLANAGAN, 1966), e apresenta falhas quando na detecção de sons inarmônicos (parciais) (SERRA, 1990). A partir desta necessidade, surgiu o algoritmo PARSHL (SMITH, 1987), que busca a detecção de parciais. O PARSHL por sua vez teve como sucessor o SMS (SERRA, 1990), que busca uma representação ainda melhor e mais completa do áudio (SERRA, 1990). Este trabalho busca comparar os progressos efetuados pelo SMS com os resultados que podem ser obtidos com o *Phase Vocoder*.

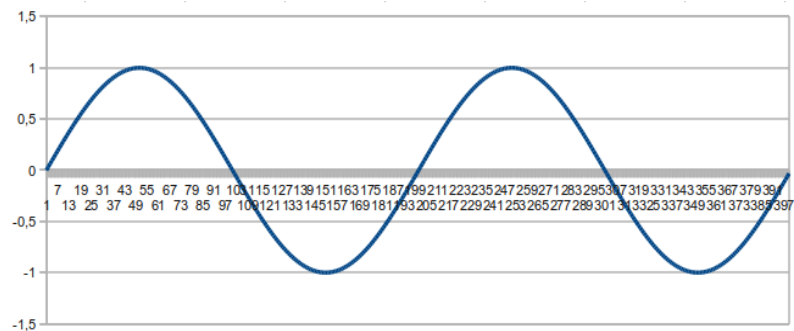
## 1.3 DEFINIÇÕES

A seguir, conceitos fundamentais de som e suas características, importantes para o entendimento deste trabalho, são brevemente discutidos. Para descrições mais detalhadas, sugere-se ao leitor as referências Castilho (2008) e Loy (2006).

- Som: Entende-se como “som” o resultado de alterações periódicas na pressão do ar nos tímpanos de um ouvinte (LOY, 2006). Um objeto que por um determinado motivo vibre, pode efetuar o deslocamento do meio (como ar ou água) produzindo a alteração de pressão, permitindo a captação e interpretação desta variação por um indivíduo. Esta variação possui propriedades físicas a serem descritas: amplitude, frequência, duração e formato da onda (LOY, 2006). Castilho (2008) troca esta última característica (formato da onda) pela característica que chama de “espectro de frequências”.

- Frequência: Por frequência entende-se a quantidade de alterações efetuadas em um meio (como o ar) durante um certo tempo. A unidade que mensura esta característica é o Hertz (Hz), que corresponde ao número de ciclos efetuados em um segundo. O ser humano consegue ouvir sons cujas frequências estão entre 17 Hz e 17.000 Hz (LOY, 2006). Sampaio (2006) afirma que a frequência audível pelo ouvido humano está entre 20 Hz e 20.000 Hz.

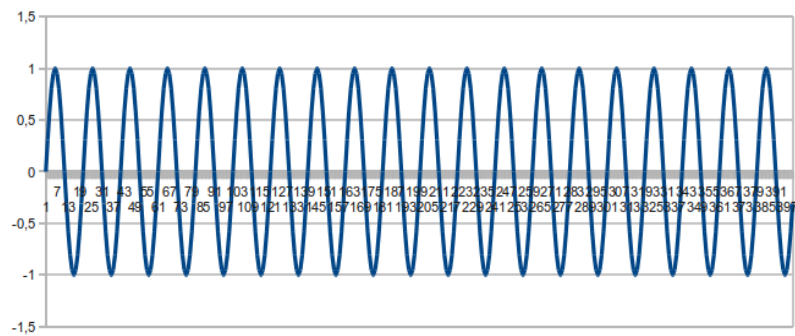
A representação de uma onda sonora com uma frequência de 40Hz, uma frequência cujo efeito percebido é de um som grave, pode ser contemplada na Figura 1:



**Figura 1: Onda de 40Hz**

Esta onda apresenta dois ciclos em 400 amostras (escala horizontal) com uma taxa de amostragem de 8000Hz, ou seja, contém 0,05 segundos.

A representação de uma onda sonora com uma frequência de 400Hz, uma frequência cujo som é mais agudo que o sinal representado na figura anterior, pode ser contemplada na Figura 2.



**Figura 2: Onda de 400Hz**

Pode-se perceber que, em uma onda cujo som é mais agudo, como a onda de 400Hz descrita na figura acima, possui maior quantidade de “ondas”, ou ciclos, no mesmo período de tempo que a onda de 40Hz e, portanto, desloca o ar mais rapidamente.

- **Duração:** Refere-se ao tempo percorrido desde o início até o término da audição de um som.
- **Amplitude:** Refere-se ao volume do som; o tamanho da onda, em sentido vertical. Sua unidade de medida é o dB (decibel).
- **Formato da onda:** O formato da onda define o timbre do som. É o que permite-se diferenciar uma nota Lá (440 Hz) produzida por um trompete, desta mesma nota produzida por um piano.

## 2 PROCESSAMENTO DIGITAL DE SINAIS DE ÁUDIO

Partindo-se das características que um sinal apresenta, é possível extrair estas informações de um sinal de áudio em um procedimento chamado de *análise*. Obtidos estes dados, pode-se alterar componentes deste sinal e obter um novo resultado. Tal processo é chamado de *síntese*.

Quando o áudio é salvo em formato digital, é obtido através de *amostragem* e, portanto, possui uma *taxa de amostragem*.

Neste trabalho serão abordadas duas técnicas para análise e síntese de sinais de áudio, *pitch-shift* e *time-stretch* utilizando dois algoritmos no domínio da frequência, *Phase Vocoder* e *SMS*.

### 2.1 DEFINIÇÕES

Esta seção apresenta definições necessárias para a compreensão de sinais digitais de áudio e seu processamento.

#### 2.1.1 Taxa de Amostragem

Um sinal de áudio capturado, por exemplo, de um microfone ou de um arquivo, possui uma taxa de amostragem. A taxa de amostragem corresponde à quantidade de amostras (dados) que foram colhidas durante o intervalo de tempo de um segundo (KOZIEROK, 2001).

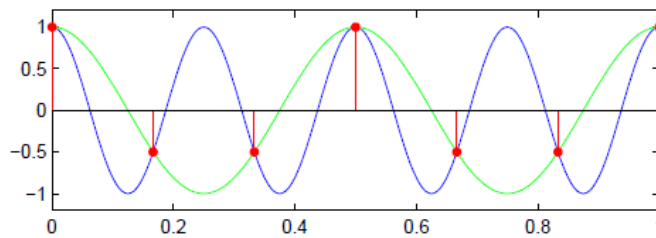
Por exemplo, se um sinal possui uma taxa de amostragem de 44.100 Hz, significa que a partir do áudio original foram colhidas 44.100 amostras no tempo de um segundo. Este procedimento é necessário dada a natureza contínua de um sinal de áudio analógico.

A frequência máxima que pode ser expressa utilizando uma taxa de amostragem  $X$  é  $X/2$ . Portanto, em um sinal de áudio cuja taxa de amostragem é de 44.100 Hz, a frequência máxima que este pode conter é de 22.050 Hz. Esta frequência máxima é chamada Frequência de Nyquist (SAMPAIO, 2006).

As ondas representadas nas Figura 1: Onda de 40Hz e Figura 2: Onda de 400Hz possuem uma taxa de amostragem de 8000Hz e 400 amostras, portanto representam 50 milissegundos de sinal.

### 2.1.2 Aliasing

Chama-se *aliasing* o processo em que frequências altas são confundidas com frequências baixas (KOTREDES, 1999). O *aliasing* ocorre quando existe uma taxa de amostragem menor do que o dobro da frequência mais alta contida em um sinal, o que acaba por interferir no espectro obtido.



**Figura 3: Aliasing**  
**Fonte: Castilho (2008, P. 24)**

Na Figura 3 é possível perceber que uma frequência alta foi obtida com amostragem insuficiente, pois considerando as amostras restantes é criada uma nova onda não inicialmente contida no sinal original, acarretando distorções no resultado.

Ao trabalhar com, por exemplo, gravação de áudio musical, a presença de uma frequência acima de metade da taxa de amostragem pode ocasionar danos ao resultado final. Para tal, os equipamentos costumam utilizar filtros passa-baixa (filtros que permitem somente frequências inferiores à frequência de corte do próprio filtro), onde são filtradas as frequências indesejadas, acima da frequência correspondente à metade da taxa de amostragem (DUNN, 1998).

### 2.1.3 Time-stretch

*Time-stretch* é uma função que permite que se aumente a duração de um sinal de áudio sem alterar a frequências das notas contidas nele (STRANDH, 2010). Se um determinado arquivo de som possui uma duração total de 10 segundos, pode-se alongá-lo em, por exemplo, 10%, fazendo com que sua duração final tenha 11 segundos. O inverso também é possível.

Esta funcionalidade deve ser executada sem efetuar nenhuma alteração na frequência do áudio original, ou seja, a redução da duração do áudio não deve alterar as notas originais.

### 2.1.4 *Pitch-shift*

A função *pitch-shift* permite que se altere a frequência do sinal de um áudio. Esta função deve funcionar de forma independente do tempo.

Uma das formas de efetuar estas alterações é através de algoritmos no domínio da frequência.

## 2.2 ALGORITMOS NO DOMÍNIO DA FREQUÊNCIA

Por algoritmos no domínio da frequência pode-se entender métodos que buscam parâmetros em sinais decompondo o sinal em suas frequências e então buscando os parâmetros que fazem parte deste (CASTILHO, 2008). Uma das técnicas utilizadas para esta finalidade é a Transformada de Fourier (*Fourier Transform*, ou FT, sigla em inglês). Esta técnica apresenta algumas variações, entre elas a Transformada de Curto Tempo de Fourier (*Short Time Fourier Transform*, ou STFT, sigla em inglês), utilizada pelo *Phase Vocoder* e pelo SMS, a Transformada Discreta de Fourier (*Discrete Fourier Transform*, ou DFT, sigla em inglês) e a Transformada Rápida de Fourier (*Fast Fourier Transform*, ou FFT, sigla em inglês).

### 2.2.1 Transformada de Fourier

A teoria de Fourier, em homenagem à Joseph Fourier (1768 – 1830), afirma que um sinal pode ser descrito como um somatório potencialmente infinito de senoides (HAYKIN, 2001). Portanto, a Transformada de Fourier permite obter o conteúdo harmônico de um sinal: o espectro de frequências que o compõe.

A Transformada de Fourier é utilizada em sinais não periódicos de tempo contínuo (HAYKIN, 2001). Para sinais periódicos utiliza-se a Série de Fourier, cuja aplicação foge ao escopo deste trabalho.

### 2.2.2 Transformada Inversa de Fourier

A Transformada Inversa de Fourier (*Inverse Fourier Transform*, IFT, na sigla em inglês) permite que um sinal no domínio da frequência seja transferido para o domínio do tempo (LYONS, 1997).

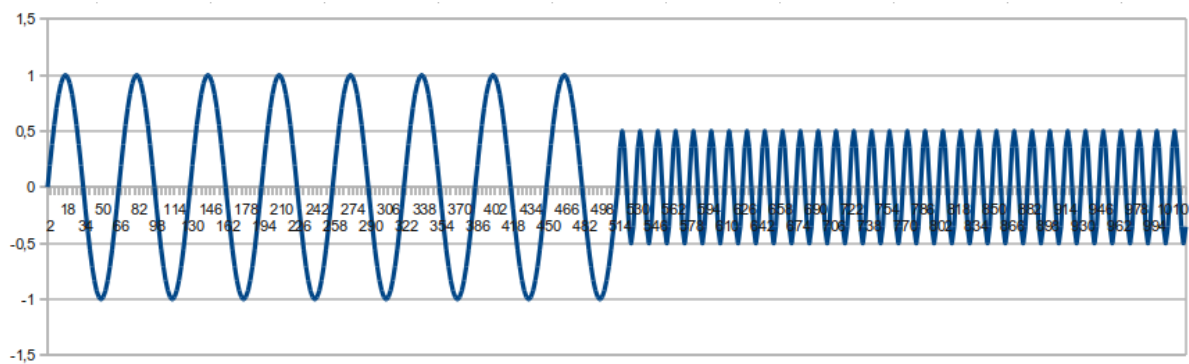
Aplicamos, por exemplo, quando deseja-se que o sinal analisado seja ressamplado.

De igual forma, existe a Transformada Inversa Discreta (*Inverse Discrete Fourier Transform*, IDFT, na sigla em inglês), Transformada Inversa Rápida (*Inverse Fast Fourier Transform*, IFFT, na sigla em inglês) e a Transformada Inversa de Curto Tempo (*Inverse Short Time Fourier Transform*, ISTFT, na sigla em inglês).

### 2.2.3 Discrete Fourier Transform (DFT)

Com o advento de computadores digitais, pioneiros nesta área desenvolveram a DFT, que permitiria o cálculo da transformada utilizando computadores digitais, trazendo o sinal do domínio do tempo para o domínio da frequência (HAYKIN, 2001).

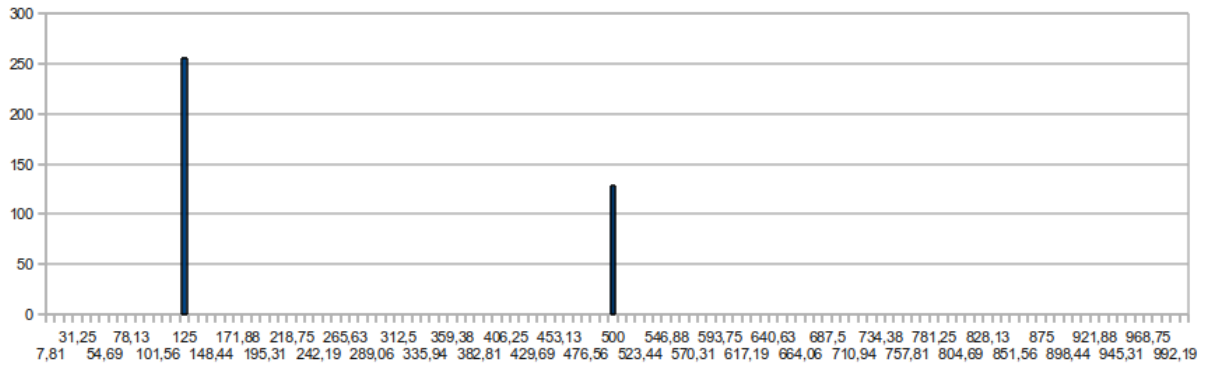
A DFT não armazena informações sobre os momentos onde estão contidas as frequências. A Figura 4, abaixo, apresenta um sinal contendo 1024 amostras e duas frequências: as primeiras 512 amostras apresentam uma frequência de 125Hz e amplitude 1, enquanto as próximas 512 amostras apresentam uma frequência de 500Hz e metade da amplitude da primeira metade do sinal.



**Figura 4: Sinal de 125Hz e 500Hz**

Ao aplicar-se, neste sinal, a Transformada de Fourier, obtém-se o espectro apresentado na Figura 5.





**Figura 5: Espectro de um sinal de 125Hz e 500Hz**

**Fonte: Produzido através da biblioteca científica em Java de Flanagan (FLANAGAN)**

Pode-se observar que o espectro apresenta valores (magnitude) nas frequências de 125Hz, cuja amplitude era maior, e na frequência de 500Hz, cuja amplitude corresponde à metade da primeira frequência. A Transformada de Fourier nos trouxe as informações contidas no sinal corretamente, porém foram perdidas as informações de tempo: não é possível saber a que momento pertence cada frequência.

Pode-se obter o mesmo resultado com um sinal de 1024 amostras contendo o somatório de ambas as senoides, ou seja, com esta informação não é possível remontar o sinal original.

A STFT, junto de operações de janelamento, busca resolver estes problemas efetuando a mesma operação em trechos menores, permitindo-se obter espectros de frequência mantendo-se as informações de tempo.

#### 2.2.4 Fast Fourier Transform (FFT)

A FFT foi descrita pela primeira vez por Cooley e Tukey (1965), e permitiu o uso da Transformada de Fourier em computadores pessoais (HAYKIN, 2001).

A DFT não é um algoritmo eficiente pois, para uma transformada de tamanho  $N$ , será necessário efetuar  $N^2$  multiplicações complexas e  $N(N-1)$  adições complexas (SAMPAIO, 2006). A FFT permite o mesmo procedimento utilizando  $\frac{N}{2} \cdot \log_2 N$  operações (SAMPAIO, 2006).

A versão mais popular do algoritmo é a *Radix-2* FFT (LYONS, 1997).

Existem outras versões mais otimizadas do algoritmo, como a *Radix-4*, *Radix-8* e *Radix-16*. A *Radix-4*, por exemplo, não fornece um desempenho em potência de dois, mas

ainda assim promove uma otimização significativa (BURRUS, 2009).

### 2.2.5 Short Time Fourier Transform (STFT)

A STFT é uma estratégia de aplicação da transformada de Fourier; trata-se da aplicação da DFT em espaços de tempo menores (CASTILHO, 2008).

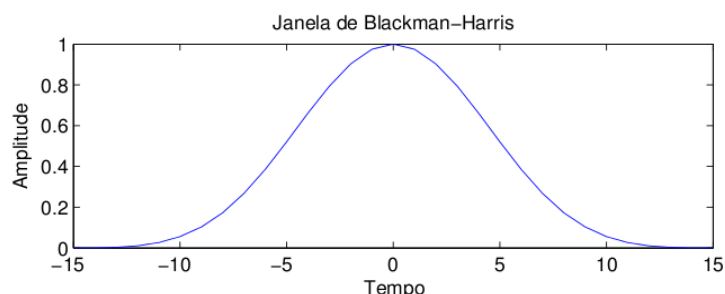
A aplicação da DFT em, um arquivo de áudio inteiro, gera um espectro complexo cujas informações temporais são perdidas. A STFT ameniza o problema reduzindo o escopo de tempo de onde as frequências estão sendo obtidas, gerando quadros, que contém informações de frequência para um determinado momento do sinal.

### 2.2.6 Janelas

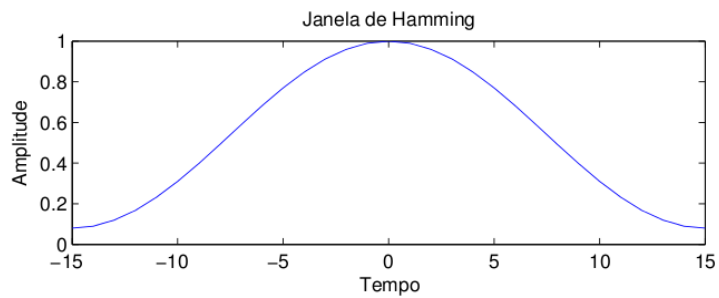
Quando aplicamos, sobre um sinal, um algoritmo como a DFT ou FFT, o algoritmo espera que o sinal exista de  $-\infty$  a  $+\infty$ , para tanto, o algoritmo supõe que o trecho de sinal se repita indefinidamente (BONFIM, 2004) (ANDRADE, 2000). Porém, caso o sinal não possua um ciclo completo, ocorrerão dessincronizações entre o início e o fim do sinal, produzindo distorções.

Para sanar este problema, aplicam-se janelas por sobre o sinal. Uma janela é um sinal com determinado formato, que é multiplicado pelo sinal onde se deseja aplicá-la. Janelas como a de Hanning produzem uma redução de amplitude no início e no final da onda (ANDRADE, 2000), tendendo a zero, permitindo que ocorram junções entre as janelas reduzindo as distorções no resultado final.

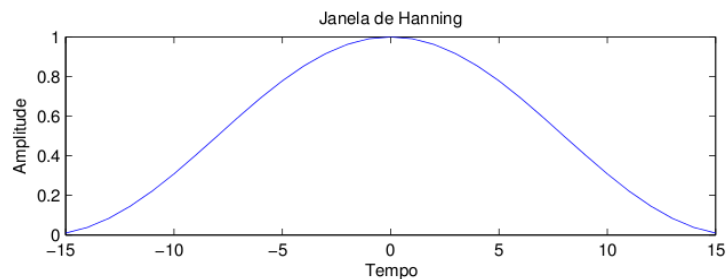
Alguns tipo de janelas são:



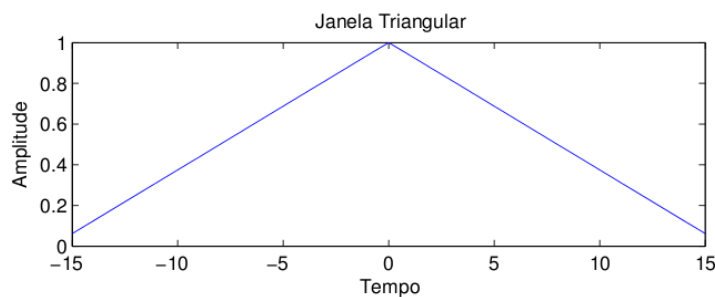
**Figura 6: Janela Blackman-Harris**  
**Fonte: Castilho (2008, p. 86)**



**Figura 7: Janela de Hamming**  
**Fonte: Castilho (2008, p. 86)**



**Figura 8: Janela de Hanning**  
**Fonte: Castilho (2008, p. 85)**



**Figura 9: Janela Triangular**  
**Fonte: Castilho (2008, p. 84)**

### 2.2.7 Phase Vocoder

O *Phase Vocoder* é um algoritmo que permite a análise e a modificação de um sinal utilizando a Transformada de Fourier ou um banco fixo de filtros passa-banda, que permitem a passagem de uma única frequência, ou de uma faixa de frequências (CASTILHO, 2008) (DOLSON, 2000).

O objetivo do *Phase Vocoder* é obter um sinal, extrair parâmetros dele, alterá-lo (ou não) e devolvê-lo. O sinal obtido é representado em um modelo que demonstra as amplitudes de cada uma das frequências em um determinado espaço de tempo. Este modelo chama-se *bin*. Trata-se de um modelo computacional para um sinal que varia de acordo com o tempo.

A facilidade de se alterar o tempo ou a frequência de um sinal, no *Phase Vocoder*, se

dá devido à transição para o modelo: se um determinado sinal, por exemplo, uma nota A (Lá) de um piano (440 Hz) for tocada, para alterar a frequência desta nota basta alterar o respectivo parâmetro do modelo. Na realidade, é necessário multiplicar todas as frequências que constam no modelo, pois instrumentos musicais não produzem somente uma única frequência, mas um conjunto delas; ocorrem uma frequência fundamental, harmônicos e parciais (JÄRVELÄINEN, 2005).

O *Phase Vocoder*, quando utilizando filtros, é composto das seguintes etapas (DOLSON, 2000):

#### 2.2.7.1 Heterodyning

O sinal é enviado para duas passagens em paralelo: em uma delas o sinal será multiplicado por uma senoide de amplitude 1.0 e frequência igual ao filtro daquela passagem, e na outra o sinal será multiplicado por uma cossenoide com as mesmas características da primeira. O resultado será dois sinais, cujas frequências serão as mesmas que a original, porém com a adição e a subtração da frequência da onda pela qual foi multiplicada. Por exemplo, um sinal de 300 Hz, que passe por este procedimento contra um filtro, cuja frequência está centrada em 50 Hz, resultará em quatro sinais, dois com frequência de 350 Hz, e com deslocamento de fase de 90°, e dois com frequência de 250 Hz, também deslocados em 90°.

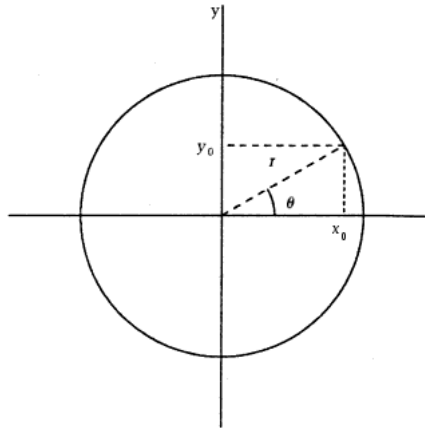
#### 2.2.7.2 Aplicação do filtro

Os sinais resultantes do processo de *heterodyning* atravessarão filtro(s) passa-baixa selecionado(s). Os sinais resultantes deverão ser somente aqueles contendo a(s) frequência(s) permitida(s) pelo(s) filtro(s).

#### 2.2.7.3 Conversão de retangular para polar

Os dois sinais resultantes corresponderão às coordenadas horizontais e verticais que podem ser plotadas em um plano cartesiano. Para obter-se o raio do círculo, utiliza-se a

equação  $r = \sqrt{x_0^2 + y_0^2}$ . Para obter-se o ângulo da reta utiliza-se  $\theta = \arctan\left(\frac{y_0}{x_0}\right)$ .

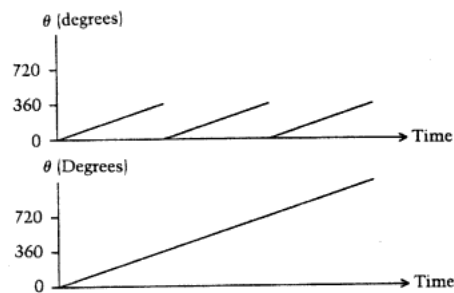


**Figura 10: Coordenadas Retangular e Polar**  
**Fonte: DOLSON (2010)**

A Figura 10 mostra a posição das coordenadas  $x$  (horizontal) e  $y$  (vertical) fornecidas e o  $r$  e  $\theta$  resultantes utilizando as equações fornecidas previamente. Com estes dois valores calculados, pode-se calcular a frequência, conforme a próxima etapa.

#### 2.2.7.4 Cálculo da fase (*unwrapping*)

Para efetuar o cálculo da frequência é necessário desencapsular a fase do sinal. Um mesmo sinal pode ter posição angular igual a, por exemplo, 0, 75, 150, 225, 300, 15, 90...; o retorno para a posição angular 15 refere-se ao momento em que terminou-se um ciclo no círculo trigonométrico e reiniciou-se a contagem. Como o cálculo de estimativa da frequência real utiliza a diferença entre as posições angulares, soma-se  $360^\circ$  a cada ciclo que termina. A representação deste cálculo pode ser visualizado na Figura 11.



**Figura 11: Desencapsulamento de fase**  
**Fonte: DOLSON (2000)**

### 2.2.7.5 Cálculo da taxa de rotação angular

Nesta etapa, calcula-se a diferença entre as posições angulares e divide-se pelo tempo, o resultado final será uma aproximação da frequência.

Cada posição será subtraída da posição anterior, por exemplo na sequência:  $(75 - 0) / T$ ,  $(150 - 75) / T$ ,  $(225 - 150) / T$ , etc. (onde  $T$  é o tempo), permitindo-se descobrir quanto tempo demora-se para completar uma revolução, sendo este valor a frequência aproximada.

### 2.2.7.6 Síntese

Nesta etapa final, utilizando os parâmetros obtidos, efetua-se uma síntese aditiva somando as saídas de todos os filtros, obtendo-se o sinal ressamplado.

Estas etapas descrevem o funcionamento mais simples do *Phase Vocoder* através de filtros. A etapa de síntese é a soma de todas as saídas dos filtros (DOLSON, 2000).

Como os filtros devem ser espaçados igualmente através do espectro desejado (normalmente o espectro correspondente ao ouvido humano) deve haver uma quantidade de filtros suficiente para que não ocorram duas frequências passando pelo mesmo filtro. Um dos problemas deste algoritmo refere-se aos parciais de um sinal; se um dos filtros permitir a passagem de mais de um parcial no mesmo sinal, estes interferirão um no outro.

Outra forma de funcionamento do *Phase Vocoder* é utilizando a Transformada de Fourier. Utilizando esta técnica, os filtros são pontos da Transformada de Fourier. Aplicando a Transformada, obtém-se um conjunto discreto de *bins*. Considera-se cada *bin* como um filtro que contempla uma determinada frequência. Após esta filtragem, repetem-se as etapas *Calcular a fase* e *Calcular a taxa de rotação angular*, conforme a implementação de filtros.

Utilizando a Transformada de Fourier obtém-se uma implementação mais flexível e rápida, onde a utilização de  $N$  filtros efetua  $N \log_2 N$  multiplicações, enquanto a utilização de um banco de filtros efetua  $N^2$  multiplicações (DOLSON, 2000).

O algoritmo do *Phase Vocoder*, utilizando filtros, pode ser descrito da seguinte forma:

1. *Heterodyning* do sinal, em duas saídas paralelas, uma multiplicada pelo seno e outra pelo cosseno;
2. Aplicação de Filtro em cada saída resultante;
3. Conversão das saídas resultantes de retangular para polar;

4. Cálculo da fase dos sinais;
5. Cálculo da taxa de rotação angular, para obtenção da frequência;

Utilizando a Transformada de Fourier, o *Phase Vocoder* pode ser descrito da seguinte forma:

1. Aplicação da Transformada de Fourier no sinal;
2. Conversão das saídas resultantes de retangular para polar;
3. Cálculo da fase do sinal;
4. Cálculo da taxa de rotação angular, para obtenção da frequência;

#### 2.2.8 SMS

O algoritmo *Spectral Modeling Synthesis* (Síntese de Modelagem Espectral, em português) foi descrito em 1989, por Xavier Serra (SERRA, 1990).

O objetivo do algoritmo é permitir a análise, transformação e síntese de um sinal através de decomposição determinística e estocástica.

Em sua parte determinística, o SMS efetua decomposição das senoides do sinal, utilizando a STFT e pesquisando por picos nas janelas de sinal. Os dados restantes são modelados como ruído branco através de filtros variáveis com o tempo (SERRA, 1990).

O SMS é composto das seguintes etapas:

##### 2.2.8.1 Cálculo da magnitude do espectro

Calcula-se a STFT do sinal. Há três parâmetros a serem considerados no cálculo da transformada: o tipo de janela de análise, cálculo da transformada de Fourier e o tamanho da sobreposição de janelas.

##### 2.2.8.2 Janela de análise

Para o SMS, o ideal é uma janela contendo um *lobe* (conjunto de *bins*) central de pequena extensão, ou seja, contendo poucos *bins* centrais, e com uma variação de amplitude grande entre o *lobe* central e os *lobes* vizinhos. Dadas as características desejáveis, a janela de Kaiser é a mais recomendada (SERRA, 1990).

### 2.2.8.3 Cálculo da Transformada de Fourier

A transformada de Fourier pode ser calculada normalmente através da DFT ou através da FFT, utilizando a estratégia de separação em quadros da STFT (SERRA, 1990). A FFT exige um quadro de tamanho múltiplo de 2, neste caso, é possível efetuar um preenchimento do quadro com zeros (*padding*), até que o quadro tenha o tamanho necessário. Este preenchimento no domínio do tempo corresponde à interpolação, no domínio da frequência.

### 2.2.8.4 Tamanho da sobreposição

O tamanho da sobreposição é a quantidade de índices que serão avançados de um quadro para outro para o processamento da STFT. A escolha do tamanho da sobreposição entre as janelas influencia na qualidade *versus* custo de processamento do algoritmo. Um bom tamanho de sobreposição é o tamanho da janela dividido pelo tamanho do *lobe* central (SERRA, 1990).

### 2.2.8.5 Detecção de Picos

A partir dos quadros obtidos pela STFT devem ser obtidos os picos principais dos quadros. Considera-se como picos principais os picos máximos do quadro e que possuem uma vizinhança de pequena magnitude.

### 2.2.8.6 Continuação de Picos

Esta etapa busca determinar para “onde” cada pico detectado na etapa anterior se encaminha em termos de frequência. A detecção de sua rota é efetuada através de “guias”, um modelo onde é visível o “caminho” percorrido pela frequência ao longo dos quadros do sinal. Faz-se necessário especificar alguns parâmetros para um melhor funcionamento deste algoritmo, portanto, é necessário conhecer de antemão algumas características do som que está sendo analisado, por exemplo, se é um instrumento de percussão, voz humana, etc., para alimentar parâmetros do algoritmo.



### 2.2.8.7 Guias

Cada guia caminhará pelos quadros traçando uma linha por entre as frequências que sejam mais próximas do valor que ela contém. Para tal, considera-se o desvio máximo que é informado como parâmetro para o SMS.

Se duas guias disputarem a frequência, a guia com a menor diferença vencerá. Se a diferença entre as frequências for maior que a tolerada, a guia passará para o estado “dormindo”. A guia pode dormir por um certo número de quadros. Caso ela continue dormindo depois deste valor, será desativada.

Caso uma frequência não seja absorvida pela rota de nenhuma guia, uma nova guia será criada para esta frequência.

Se é conhecido que o som analisado é harmônico (sem parciais, exatos em relação a uma frequência fundamental), o algoritmo buscará a frequência fundamental no quadro e, em cada avanço do quadro, buscará a mesma frequência fundamental e, encontrando-a, atualizará os valores de acordo com a série harmônica, caso contrário, serão mantidos os valores.

### 2.2.8.8 Representação da parte Determinística

Os dados resultantes do processamento usando as guias será no formato  $A_r(l), \omega_r(l)$ , onde estão representadas amplitude e frequência, respectivamente, para cada guia, ao longo de um quadro  $l$  e uma guia  $r$ .

### 2.2.8.9 Síntese Determinística

A síntese determinística pode ser alcançada utilizando as funções de amplitude  $\hat{A}(l)$  e frequências  $\omega(l)$  obtidas na etapa de análise. Para evitar cliques entre os quadros estes são interpolados, ao serem sobrepostos (SERRA, 1990).

### 2.2.8.10 Processamento da Parte Estocástica

Entende-se como componente estocástico o sinal residual simplificado obtido do sinal original.

A parte estocástica do sinal é obtida através de subtração, no domínio da frequência, da parte determinística do sinal, obtida através das etapas anteriores, do sinal original,

restando a parte estocástica.

O sinal original sem a parte determinística pode se apresentar contendo grandes variações entre as vizinhanças de cada frequência, portanto, utiliza-se um algoritmo para definir uma curva entre os pontos do espectro estocástico resultante. Algoritmos padrões são a interpolação do tipo *spline*, *least squares* (quadrados mínimos) ou aproximação por linha reta (SERRA, 1990).

#### 2.2.8.11 Representação da Parte Estocástica

A parte estocástica do sinal é representada através de “envelopes” de sinal, que podem ser vistos como filtros modeladores de frequência para um determinado quadro, ou como um filtro passa-banda, dependendo da forma como é aplicado (SERRA, 1990).

#### 2.2.8.12 Síntese Estocástica

A síntese da parte estocástica ocorre a partir da transformada inversa de Fourier, utilizando sobreposição e soma em cada um dos quadros gerados durante o processamento desta. Não há informações de fase no espectro, pois a parte estocástica é ruído e, portanto, possui sinal aleatório como fase. Assim, utiliza-se um gerador aleatório de fase para a síntese (SERRA, 1990).

O algoritmo SMS pode ser descrito nesta sequência de etapas:

1. Cálculo da magnitude do espectro
2. Detecção de picos
3. Continuação de picos
4. Representação (obtenção) da parte determinística
5. Processamento da parte estocástica (subtração determinística)
6. Representação (obtenção) da parte estocástica

### 3 COMPARATIVO DOS ALGORITMOS

Para comparar os resultados obtidos a partir do processamento das amostras de áudio com os algoritmos *Phase Vocoder* e SMS, foram consideradas abordagens de avaliação quantitativas e qualitativas.

Através da comparação pontual do resultado esperado com aquele obtido com o processamento pode-se avaliar quantitativamente quais os efeitos para cada classe de amostras. Por outro lado, a avaliação do resultado por parte de usuários (por meio de um questionário) permite uma análise subjetiva.

Estas abordagens, bem como os resultados obtidos, serão descritos a seguir.

#### 3.1 AVALIAÇÃO QUANTITATIVA

Os algoritmos foram comparados, de forma objetiva, aplicando-os em três tipos de ondas específicas: quadrada, dente de serra e triangular.

Foram sintetizadas estes três tipos de onda em duas frequências diferentes: 110 Hz e 220 Hz, ambas com 1 segundo de áudio, taxa de amostragem de 44.100 Hz e 24 bits para cada amostra, para efetuar a comparação de *pitch-shift* dos algoritmos. Foi sintetizada uma terceira onda, de 110 Hz e 2 segundos de áudio, para a comparação da funcionalidade de *time-stretch* dos algoritmos.

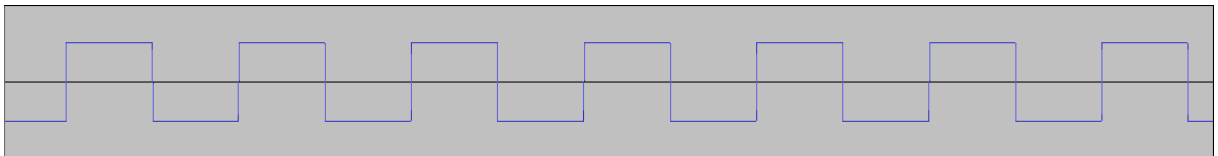
Os algoritmos *Phase Vocoder* e SMS foram aplicados na primeira onda sintetizada, de 110 Hz e 1 segundo de duração, para que a frequência fosse dobrada, para 220 Hz. As ondas de 220 Hz produzidas pelos algoritmos foram invertidas e sobrepostas à onda de 220 Hz sintetizada, sendo que o somatório das duas deveria resultar em uma onda sem sinal nenhum, somente com valores iguais à zero. As amostras do resultado desta sobreposição foram somados para produzir um “somatório de amostras”, utilizando um programa em C++ que pode ser consultado no Anexo A. Estes testes visam avaliar a capacidade do algoritmo de manter a fidelidade ao áudio original quando em sua funcionalidade de *pitch-shift*.

Para o teste de *time-stretch* os algoritmos foram aplicados na primeira onda sintetizada, de 110 Hz e 1 segundo de duração, para que sua duração fosse estendida para 2 segundos. Para a comparação foi utilizada a terceira onda sintetizada, cuja frequência é 110 Hz e duração igual 2 segundos. Novamente os resultados dos algoritmos foram sobrepostos à onda original e o resultado deveria ser um sinal somente com valores iguais à zero.

As ondas de formato triangular, dente de serra e quadrada são descritas a seguir.

### 3.1.1 Onda Quadrada

Considera-se uma onda quadrada uma onda contendo transições instantâneas entre dois níveis (WEISSTEIN, 2008). Esta onda, quando no domínio da frequência, é composta por  $n$  senoides somadas, portanto, o algoritmo deve conseguir manipular os harmônicos que a compõe a fim de manter um resultado fiel.

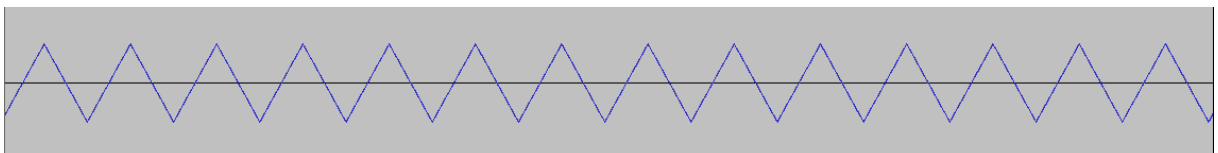


**Figura 12: Onda quadrada**

### 3.1.2 Onda Triangular

A onda triangular recebe este nome dado ao seu formato, que molda “triângulos” ao longo do ciclo completo. Varia de 1 a -1, ou outros valores, se tiver alterações em sua amplitude, e suas amostras descrevem uma linha reta de um extremo a outro.

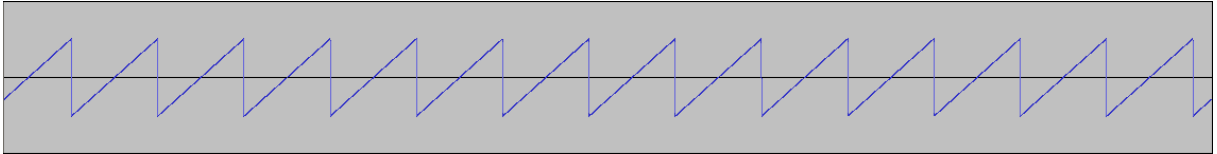
Também é composta de um conjunto grande de harmônicos, quando no domínio da frequência.



**Figura 13: Onda triangular**

### 3.1.3 Onda Dente de Serra

A onda dente de serra é formada por uma representação gráfica similar à de uma serra, com um conjunto de triângulos inclinados. O sinal efetua uma subida vertical e então apresenta uma queda abrupta. Este caminho também pode ser invertido, onde o sinal apresenta uma queda horizontal e então uma subida abrupta até o limite superior do sinal, sendo, neste caso, chamada de onda dente de serra invertida.



**Figura 14: Onda dente-de-serra**

### 3.2 AVALIAÇÃO QUALITATIVA

A avaliação qualitativa ocorreu através das análises subjetivas realizadas por voluntários, a grande maioria com experiência na área de análise de sons, após ouvir amostras de áudio originais e alteradas pelos algoritmos. Tal análise foi realizada por meio de questionários, disponibilizados em formato para web como um formulário HTML.

O voluntário teve acesso, através de links para *download*, de um pacote com todas as amostras originais e processadas pelos algoritmos. As amostras disponíveis são:

- drums: amostra com 21 segundos de duração, taxa de amostragem de 44.100Hz e amostras de 24 bits, de uma sequência de bateria e outros instrumentos de percussão;
- drums\_stretch\_P: amostra “drums” processada utilizando *Phase Vocoder* para apresentar o dobro do tempo;
- drums\_stretch\_S: amostra “drums” processada utilizando SMS para apresentar o dobro do tempo;
- recital: amostra com 8 segundos, taxa de amostragem de 88.200Hz e amostras de 24 bits, de um dueto de voz feminina e cravo, apresentando um pouco de reverberação;
- recital\_stretch\_P: amostra “recital” processada utilizando *Phase Vocoder* para apresentar o dobro do tempo;
- recital\_stretch\_S: amostra “recital” processada utilizando SMS para apresentar o dobro do tempo;
- recital\_pitch\_P: amostra “recital” processada utilizando *Phase Vocoder* para apresentar as notas elevadas em um intervalo de terça maior;
- recital\_pitch\_S: amostra “recital” processada utilizando SMS para apresentar as notas elevadas em um intervalo de terça maior;
- violin: amostra de 12 segundos, taxa de amostragem de 44.100Hz e amostras de 24 bits, contendo um trecho solo de violino em ambiente sem reverberação;

- violin\_stretch\_P: amostra “violin” processada utilizando *Phase Vocoder* para apresentar as notas elevadas em um intervalo de terça maior;
- violin\_stretch\_S: amostra “violin” processada utilizando SMS para apresentar as notas elevadas em um intervalo de terça maior;

Para cada conjunto de amostras que se desejar comparação, foi apresentada uma grade de pontuações, onde o voluntário pode graduar o nível gerado do artefato que está sendo solicitado. A grade pode ser contemplada na Figura 15.

**Escute as amostras "drums.flac" e "drums\_stretch\_P.flac" e avalie as características encontradas \***

	Muito	Pouco	Nenhum
Sobreposição: o mesmo som parece estar sendo tocado mais de uma vez com pequenas variações no tempo, pequenos atrasos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pequenos cortes: ao invés do som estar fluindo continuamente, como no original, passa a ter pequenas falhas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Chuveiro: há ruído somado ao som original	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Saturação de médios: o som parece estar sendo tocado "dentro de uma lata".	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Flutuações no áudio: o som parece estar sendo tocado "debaixo d'água".	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Encontrou alguma característica que não está na relação acima? Descreva-a:**

**Figura 15: Estrutura das perguntas para avaliação das amostras**

As amostras solicitadas para comparação foram:

- drums e drums\_stretch\_P;
- drums e drums\_stretch\_S;
- recital e recital\_stretch\_P;
- recital e recital\_stretch\_S;
- recital e recital\_pitch\_P;
- recital e recital\_pitch\_S;
- violin e violin\_pitch\_P;

– violin e violin\_pitch\_S;

O questionário apresentou os seguintes textos introdutórios, contendo links para o download das amostras, e perguntas:

“Este questionário tem por objetivo avaliar o resultado do processamento de 3 amostras de áudio utilizando dois algoritmos diferentes. Para responder as perguntas, escute os arquivos de áudio solicitados em cada questão. Sugere-se o uso de fones de ouvido circumaural (aqueles fones grandes, que cobrem toda a orelha) ou outro equipamento de áudio de qualidade. Caso não os tenha, basta descrever o equipamento que possuir.

Considere as amostras de áudio no seguinte arquivo:

<http://www.zumodrive.com/share/dXs8Mjk0N2>.

(Link alternativo para as amostras, caso o anterior dê problema:

[http://sharex.xpg.com.br/files/1543104035/amostras\\_wav.zip.html](http://sharex.xpg.com.br/files/1543104035/amostras_wav.zip.html)).

(Link alternativo para as amostras, caso o anterior também dê problema:

[http://www.4shared.com/file/T\\_WoHASD/amostras\\_wav.html](http://www.4shared.com/file/T_WoHASD/amostras_wav.html)).

- 1) Que equipamento de áudio você está utilizando agora para avaliar as amostras? \*
  - Fones de ouvido *circumaural* (fone grande, cobre toda a orelha, almofadado)
  - Fones de ouvido *earbuds* (fone pequeno, que acompanha celulares e *iPods*)
  - Fones de ouvido intra-auricular (fone que entra no canal auditivo)
  - Caixas de som para computador simples
  - Caixas de som para computador com *subwoofer*
  - Home theater* ou outro tipo de sistema grande de amplificação
  - Headset* (fone normalmente em conjunto com um microfone, similar àqueles fones de atendente de telemarketing)
  - Outro: \_\_\_\_\_
  
- 2) Você toca algum instrumento musical ou canta? \*
  - Sim  Não
  
- 3) Trabalha com edição de áudio ou em estúdio? \*
  - Sim  Não

**Time Stretch**

Esta primeira série de perguntas serve para avaliar a qualidade dos algoritmos quando efetuam a operação de "time stretch" (alongar o áudio sem alterar as frequências/notas musicais do áudio). O objetivo é verificar se efetuar esta operação de "alongar" o áudio, gerou defeitos perceptíveis.

**Pitch Shift**

Esta segunda série de perguntas serve para avaliar a qualidade dos algoritmos quando efetuam a operação de "pitch shift" (alterar as frequências/notas musicais do áudio sem torná-la mais curta ou mais longa). O objetivo é verificar se esta operação (deixar a música mais aguda) gerou defeitos perceptíveis.”



## 4 RESULTADOS

Esta seção apresenta e discute os resultados obtidos nas análises quantitativa e qualitativa.

### 4.1 ANÁLISE QUANTITATIVA

A análise quantitativa foi efetuada através do programa Audacity.

Inseriu-se os dois sinais dentro de um mesmo arquivo de áudio. O sinal original foi invertido verticalmente e a combinação resultante dos dois sinais foi exportada para um arquivo só, gerando as ondas que constam nesta seção.

Este somatório de sinais invertidos deve anular as frequências. O algoritmo que conseguir criar um sinal resultante somente com zeros (uma linha reta) terá efetuado a operação (*time-stretch* ou *pitch-shift*) com perfeição.

O resultado dos somatórios pode ser conferido nas Figuras 16, a 27. As figuras possuem um eixo vertical representando a amplitude do sinal (variando de 1 a -1) e um eixo horizontal representando o o tempo, em segundos.

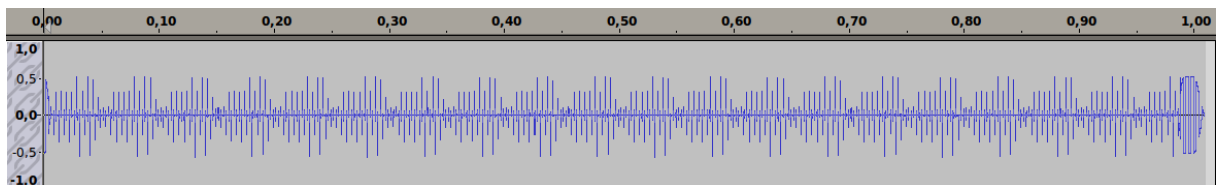


Figura 16: Somatório das ondas quadradas sintética e processada em *pitch-shift* pelo *Phase Vocoder*

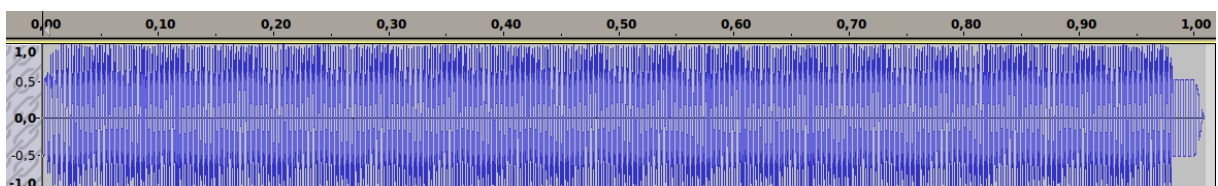


Figura 17: Somatório das ondas quadradas sintética e processada em *pitch-shift* pelo *SMS*

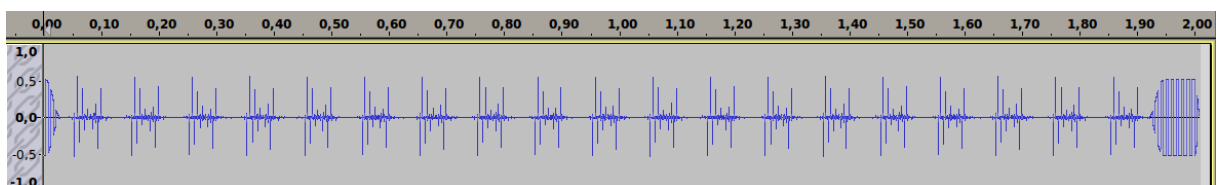


Figura 18: Somatório das ondas quadradas sintética e processada em *time-stretch* pelo *Phase Vocoder*

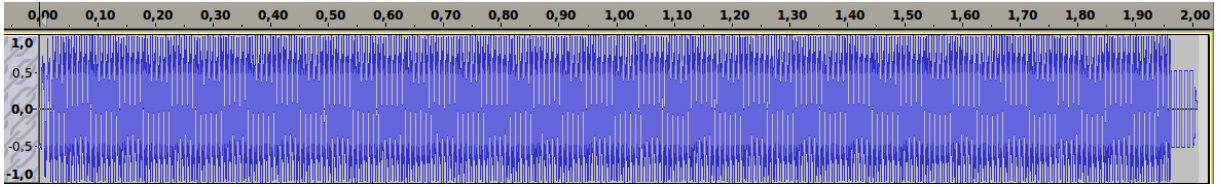


Figura 19: Somatório das ondas quadradas sintética e processada em *time-stretch* pelo SMS

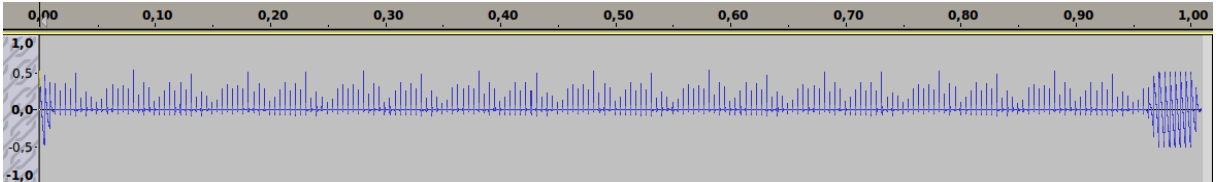


Figura 20: Somatório das ondas dente-de-serra sintética e processada em *pitch-shift* pelo *Phase Vocoder*

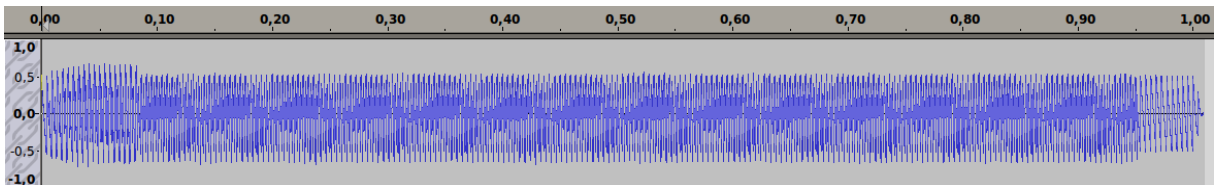


Figura 21: Somatório das ondas dente-de-serra sintética e processada em *pitch-shift* pelo SMS

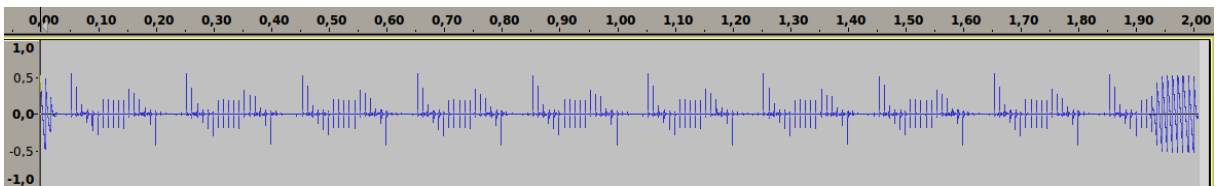


Figura 22: Somatório das ondas dente-de-serra sintética e processada em *time-stretch* pelo *Phase Vocoder*

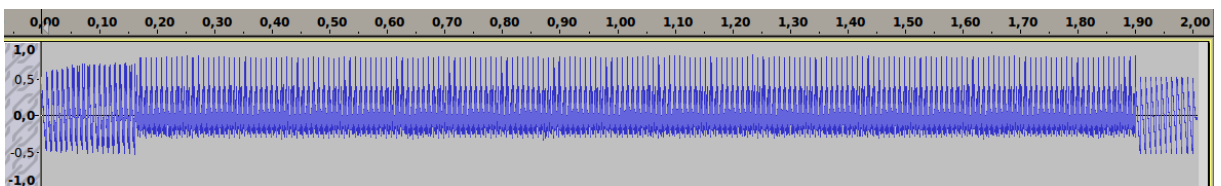


Figura 23: Somatório das ondas dente-de-serra sintética e processada em *time-stretch* pelo SMS

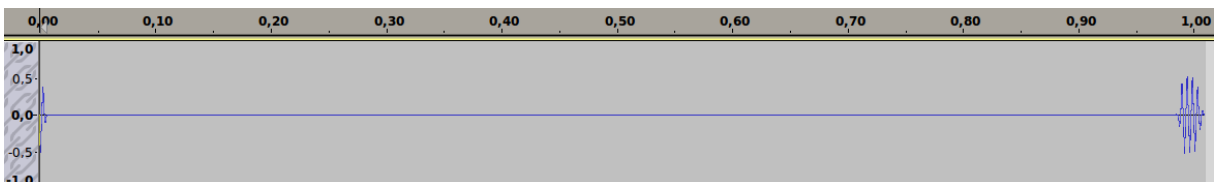


Figura 24: Somatório das ondas triangulares sintética e processada em *pitch-shift* pelo *Phase Vocoder*

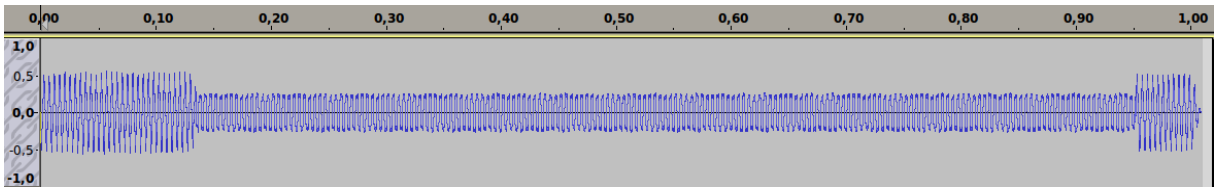


Figura 25: Somatório das ondas triangulares sintética e processada em *pitch-shift* pelo SMS

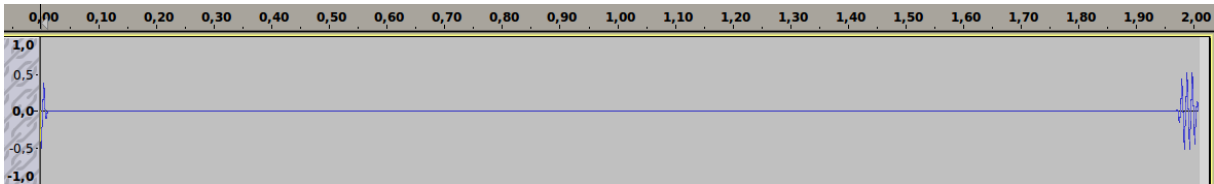


Figura 26: Somatório das ondas triangulares sintética e processada em *time-stretch* pelo *Phase Vocoder*

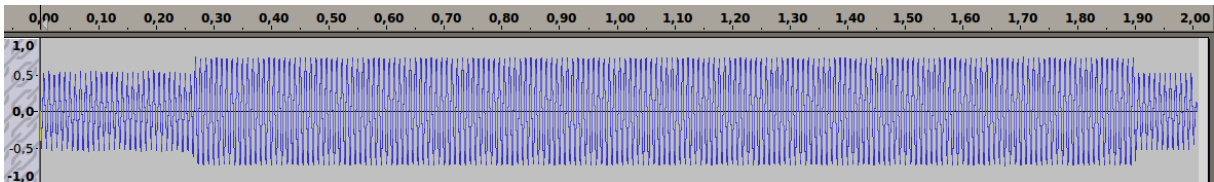


Figura 27: Somatório das ondas triangulares sintética e processada em *time-stretch* pelo SMS

Tabela 1: Somatório das amostras dos arquivos de áudio resultantes

	<b>Pitch-shift (Phase Vocoder)</b>	<b>Time-stretch (Phase Vocoder)</b>	<b>Pitch-shift (SMS)</b>	<b>Time-stretch (SMS)</b>
<b>Quadrada</b>	820,11	2222,32	22400,12	43643,27
<b>Dente-de-serra</b>	718,08	1177,20	10853,12	13143,59
<b>Triangular</b>	364,24	487,76	9142,56	35795,50

As ondas quadradas (Figuras 16, 17, 18 e 19) geraram a maior quantidade de artefatos, tanto do *Phase Vocoder* quanto do SMS. O somatório das ondas quadradas processadas pelo SMS gerou ondas com amplitude bastante alta, conforme pode ser visto na Tabela 1. O *Phase Vocoder* obteve um sinal final mais limpo, porém gerou artefatos no começo e no final do arquivo.

A onda dente-de-serra, em *time-stretch*, processada pelo SMS (Figura 23) produziu um resultado final onde a onda passou a ter mais valores positivos que negativos, assim como o *Phase Vocoder* na operação *pitch-shift* (Figura 20). O *Phase Vocoder* gerou um resultado menos ruidoso na operação *time-stretch* (Figura 22). O SMS gerou um resultado bastante ruidoso na operação de *pitch-shift* (Figura 21).

As ondas triangulares processadas pelo *Phase Vocoder* (Figuras 24 e 26) tiveram um resultado beirando à onda perfeita, onde as operações de *pitch-shift* e *time-stretch* seriam

idênticas às ondas sintetizadas. Porém foram gerados artefatos no começo e no final do arquivo. Ainda assim, manteve-se uma grande fidelidade, com um total do somatório de amostras sendo o mais baixo entre os arquivos.

O processamento do SMS em *pitch-shift* gerou também uma grande quantidade de ruído na onda triangular (Figuras 25 e 27), porém, se comparados às outras ondas processadas pelo SMS, este sinal não ficou tão poluído. Isto talvez ocorra devido à onda triangular apresentar um espectro mais simples, contendo menos harmônicos e com amplitudes menores, simplificando a sua detecção pelos algoritmos, porém a operação de *time-stretch* gerou um resultado bastante ruidoso.

Todas as ondas processadas tiveram artefatos visíveis gerados no começo e no final do arquivo.

## 4.2 ANÁLISE QUALITATIVA

Para o processamento das amostras utilizando o *Phase Vocoder* utilizou-se uma implementação em MATLAB (ELLIS, 2010). Esta implementação disponibiliza um único parâmetro para o processamento: o tamanho da janela da STFT.

A partir dos resultados obtidos com diversos tamanhos de janelas, os melhores resultados (onde foram gerados menor ou zero artefatos) foram obtidos com os seguintes parâmetros:

- drums: janela da STFT de 1024 pontos
- recit: janela da STFT de 4096 pontos
- violin: janela da STFT de 1024 pontos

Para o processamento das amostras utilizando o SMS utilizou-se a implementação do SMS disponibilizada pelo próprio grupo que desenvolve o algoritmo (MTG, 2000).

O SMS é um algoritmo complexo que permite um conjunto grande de parâmetros. Para utilizá-lo corretamente, é sugerido analisar o áudio e o conteúdo que se deseja extrair dele, como por exemplo, número de senoides, se o som é harmônico ou não, etc.. (MTG, 2000).

O uso do SMS acontece em duas etapas distintas: análise e síntese. Na etapa de análise especificam-se os parâmetros desejados e será gerado um arquivo com extensão *.sms*. Este arquivo conterá as informações analisadas do áudio. A partir deste arquivo pode-se sintetizar

um novo arquivo de áudio (onde também podem ser aplicados parâmetros).

Para o processamento das amostras foram utilizados os seguintes parâmetros, que obtiveram um melhor resultado (onde foram gerados poucos ou nenhum artefato):

#### 4.2.1 Drums

No processamento do áudio contendo sons percussivos, os comandos utilizados para a análise foram:

```
smsAnal drums.wav drums.sms -u 80 -s 3 -i 0 -p 200 -f 1
```

Onde:

-u 80: fundamental padrão, em sons inarmônicos (como percussão) este valor define o tamanho da janela de análise;

-s 3: este número é multiplicado pelo período da frequência fundamental em cada momento. O mais indicado para sons percussivos é uma janela pequena (MTG, 2000), porém, esta amostra apresentava mais instrumentos soando simultaneamente e precisou ser aumentada;

-i 0: o tipo da janela. Zero especifica o uso da janela de Hamming, adequado para sons de ataque rápido (MTG, 2000);

-p 200: número de parciais a serem detectadas;

-f 1: tipo de som, neste caso, "1" especifica que o áudio é inarmônico;

A síntese do som foi produzida utilizando o seguinte comando:

```
smsSynth drums.sms drums_SMS.wav -s 0 -g 1.9
```

Onde:

-s 0: o valor zero especifica que a síntese deve conter a parte determinística e a parte estocástica;

-g 1.9: este valor é multiplicado pelo componente estocástico. Ao efetuar-se a análise, foi perceptível uma perda de agudos, que estava sendo identificada como ruído, incrementar este valor devolveu um pouco do ataque dos instrumentos de percussão;

#### 4.2.2 Recit

No processamento da amostra contendo voz acompanhada por um cravo o comando utilizado para análise foi:

```
smsAnal recit.wav recit.sms -s 2 -i 4 -p 160 -f 1 -l 50
```

Onde:

-i 4: janela tipo “Blackman-Harris 92dB”, um tipo adequado para sons longos e estáveis (MTG, 2000);

-l 50: fundamental mais grave a ser detectada;

A síntese da amostra foi produzida utilizando o seguinte comando:

```
smsSynth recit.sms recit_SMS.wav -s 0 -g 0.5
```

-g 0.5: parte dos agudos também era perdida neste processamento, porém menos que o ataque dos instrumentos percussivos. A parte estocástica desta análise detectou uma grande quantidade de chiados, provavelmente devido à reverberação da amostra, portanto, foi necessário reduzir o componente estocástico aplicado;

#### 4.2.3 Violin

O processamento da amostra de um som único de violino foi efetuado utilizando o seguinte comando:

```
smsAnal recit_right.wav recit_right.sms -s 2 -i 4 -p 160 -f 1
-l 50
```

A síntese da amostra foi efetuada utilizando o seguinte comando:

```
smsSynth violin.sms violin_SMS.wav -s 0 -g 0.5
```

Onde:

-f 1: Apesar deste som ser harmônico, o SMS obteve um resultado melhor tratando o áudio como inarmônico.

#### 4.2.4 Resultados do Questionário

Esta seção apresenta os resultados obtidos através do questionário.

O questionário foi respondido por 11 pessoas.

Quanto ao equipamento utilizado, 18,18% dos respondentes utilizou caixas de som de notebook, 18,18% utilizou caixas de som para computador com *subwoofer*, 27,27% utilizaram caixas de computador simples, 18,18% utilizaram fones de ouvido *circumaural*, 9,09% utilizaram *earbuds* e 9,09% utilizaram *home theater* ou outro tipo de sistema grande de amplificação.

Quanto à informação se o respondente é músico (pergunta 2), 90,91% respondeu ser

músico, enquanto 9,09% respondeu “não”.

Quanto à pergunta “Trabalha com edição de áudio ou em estúdio”, 72,73% afirmou não trabalhar e 27,27% afirmou trabalhar nesta área.

Não houveram respondentes que não fossem músicos e trabalhassem em estúdio ou com edição de áudio.

Quando perguntados sobre outros artefatos gerados pelo processamento dos algoritmos, que não estivessem entre as opções disponíveis, não houve respostas para nenhum comparativo de amostras.

Os resultados dos comparativos das amostras pode ser conferido na Tabela 2.

**Tabela 2: Comparativo de resultados do questionário**

Amostra:	drums_stretch_P			Amostra:	drums_stretch_S		
	Muito	Pouco	Nenhum		Muito	Pouco	Nenhum
<b>Músicos</b>	14,00%	34,00%	52,00%	<b>Músicos</b>	28,00%	36,00%	36,00%
<b>Estúdio</b>	0,00%	40,00%	60,00%	<b>Estúdio</b>	33,33%	46,60%	20,07%
<b>Não-músicos</b>	20,00%	60,00%	20,00%	<b>Não-músicos</b>	20,00%	40,00%	40,00%
<b>Totais:</b>	<b>11,33%</b>	<b>44,67%</b>	<b>44,00%</b>	<b>Totais:</b>	<b>27,11%</b>	<b>40,87%</b>	<b>32,02%</b>

Amostra:	recital_stretch_P			Amostra:	recital_stretch_S		
	Muito	Pouco	Nenhum		Muito	Pouco	Nenhum
<b>Músicos</b>	4,00%	34,00%	62,00%	<b>Músicos</b>	12,00%	40,00%	48,00%
<b>Estúdio</b>	0,00%	20,00%	80,00%	<b>Estúdio</b>	13,30%	53,50%	33,20%
<b>Não-músicos</b>	20,00%	20,00%	60,00%	<b>Não-músicos</b>	0,00%	80,00%	20,00%
<b>Totais:</b>	<b>8,00%</b>	<b>24,67%</b>	<b>67,33%</b>	<b>Totais:</b>	<b>8,43%</b>	<b>57,83%</b>	<b>33,73%</b>

Amostra:	recital_pitch_P			Amostra:	recital_pitch_S		
	Muito	Pouco	Nenhum		Muito	Pouco	Nenhum
<b>Músicos</b>	6,00%	26,00%	68,00%	<b>Músicos</b>	14,00%	36,00%	50,00%
<b>Estúdio</b>	6,60%	33,33%	60,07%	<b>Estúdio</b>	20,00%	33,33%	46,67%
<b>Não-músicos</b>	0,00%	20,00%	80,00%	<b>Não-músicos</b>	0,00%	20,00%	80,00%
<b>Totais:</b>	<b>4,20%</b>	<b>26,44%</b>	<b>69,36%</b>	<b>Totais:</b>	<b>11,33%</b>	<b>29,78%</b>	<b>58,89%</b>

Amostra:	violin_pitch_P			Amostra:	violin_pitch_S		
	Muito	Pouco	Nenhum		Muito	Pouco	Nenhum
<b>Músicos</b>	0,00%	26,00%	74,00%	<b>Músicos</b>	10,00%	18,00%	72,00%
<b>Estúdio</b>	0,00%	26,67%	73,33%	<b>Estúdio</b>	13,30%	20,00%	66,70%
<b>Não-músicos</b>	0,00%	0,00%	100,00%	<b>Não-músicos</b>	0,00%	0,00%	100,00%
<b>Totais:</b>	<b>0,00%</b>	<b>17,56%</b>	<b>82,44%</b>	<b>Totais:</b>	<b>7,77%</b>	<b>12,67%</b>	<b>79,57%</b>

Na comparação entre as amostras “drums” e “drums\_stretch\_P”, dos respondentes que são músicos, 30% afirmou haver muita sobreposição, 50% afirmou haver poucos pequenos cortes, 10% afirmou existir muito chuveiro, 40% afirmou haver pouco chuveiro, 20% afirmou existir muita saturação de médios, 50% afirmou existir pouca saturação de médios, 10% afirmou existir muitas flutuações de áudio e 30% afirmou existir poucas flutuações no áudio.

Dos entrevistados que trabalham em estúdio, não foi detectada sobreposição, 33,33% afirmou existir pouco de pequenos cortes, 33,33% detectou pouco chuvisco, 66,67% afirmou existir pouca saturação de médios e 66,67% afirmou existir poucas flutuações no áudio.

Dos entrevistados que não são músicos, 100% afirmou existir muita sobreposição, pouco de pequenos cortes, chuvisco e saturação de médios, e nenhuma flutuação no áudio.

Na comparação entre as amostras “drums” e “drums\_stretch\_S”, dos respondentes que são músicos, 20% afirmou existir muita sobreposição e 50% afirmou existir pouca sobreposição. 10% afirmou existir muito de pequenos cortes, enquanto 50% afirmou existir pouco. 40% afirmou existir muito chuvisco, enquanto 50% afirmou existir pouco. 50% afirmou existir muita saturação de médios, enquanto 30% afirmou existir pouco. 20% afirmou existirem muitas flutuações no áudio, enquanto 30% afirmou existirem poucas.

Dos entrevistados que não são músicos, 100% afirmou existir pouca sobreposição, 100% afirmou existirem pequenos cortes, 100% afirmou existir pouco chuvisco, 100% afirmou existir muita saturação de médios, 100% afirmou não existirem flutuações no áudio.

Dos respondentes que trabalham com edição de áudio ou em estúdio, 66,67% afirmou existir pouca sobreposição, 66,67% afirmou existirem pouco de pequenos cortes, 66,67% afirmou existir muito de chuvisco, enquanto 33,33% afirmou existir pouco chuvisco, 66,67% afirmou existir muita saturação de médios, enquanto 33,33% afirmou existir pouca saturação de médios, e 33,33% afirmou existirem muitas flutuações no áudio, enquanto 33,33% identificou poucas flutuações no áudio.

Na comparação entre as amostras “recital” e “recital\_stretch\_P”, dos respondentes que são músicos, 10% afirmou existir muita sobreposição, enquanto 50% identificou pouca sobreposição. 20% identificou poucos pequenos cortes, 10% identificou pouco chuvisco, 50% afirmou existir pouca saturação de médio, 10% identificou muitas flutuações no áudio, enquanto 40% detectou pouca flutuação no áudio.

Entre os entrevistados que não são músicos, 100% identificou muita sobreposição, 100% identificou poucos pequenos cortes, não houve detecção de chuvisco, nem de saturação de médio nem de flutuações no áudio

Entre os entrevistas que trabalham em estúdio ou com edição de áudio, 33,33% afirmou ter ouvido pouca sobreposição, não foram detectados pequenos cortes, 33,33% detectou pouco chuvisco, 33,33% detectou pouca saturação de médios e 33,33% detectou poucas flutuações no áudio.

Na comparação entre as amostras “recital” e “recital\_stretch\_S”, entre os respondentes músicos, 20% declarou ter escutado muita sobreposição, enquanto 40% afirmou ter detectado



pouca sobreposição. 10% dos respondentes afirmam ter detectado muito de pequenos cortes, enquanto 50% afirmou ter ouvido pouco pequenos cortes. 10% detectou muito chuva, enquanto 40% detectou pouco chuva. 20% responderam existir muita saturação de médios, enquanto 50% detectou pouca saturação de médios. 20% afirmou existirem muitas flutuações no áudio, enquanto 20% afirmou existirem poucas.

Entre respondentes que não são músicos, 100% afirmou existir pouca sobreposição, pouco pequenos cortes, pouco chuva, pouca saturação de médios e nenhuma flutuação no áudio.

Dos respondentes que trabalham em estúdio, 66,67% afirmou ter detectado pouca sobreposição, 66,67% afirmou ter detectado pouco de pequenos cortes, 33,33% afirmou ter escutado muito chuva, enquanto 33,33% afirmou pouco chuva, 66,67% afirmou pouca saturação de médios, e 33,33% afirmou muitas flutuações no áudio, enquanto 33,33% afirmou poucas flutuações no áudio.

Na comparação entre as amostras “recital” e “recital\_pitch\_P”, dos respondentes que são músicos 10% afirmou existir muita sobreposição, enquanto 30% afirmou existir pouca, 10% afirmou existir muitos pequenos cortes, enquanto 20% afirmou existirem poucos, 20% detectou pouco chuva, 50% detectou pouca saturação de médios, 10% detectou muitas flutuações no áudio enquanto 10% detectou poucas flutuações no áudio.

Dos respondentes que não são músicos, 100% afirmou existir pouca sobreposição. Não foram detectados pequenos cortes, chuva, saturações de médio ou flutuações no áudio.

Dos entrevistados que trabalham em estúdio, 33,33% afirmou ter detectado pouca sobreposição, pouco de pequenos de cortes, pouco chuva, 66,67% detectou pouca saturação de médios e 33,33% detectou muitas flutuações no áudio.

Na comparação entre as amostras “recital” e “recital\_pitch\_S”, dos respondentes que são músicos, 20% afirmou existir muita sobreposição, enquanto 40% afirmou existir pouca sobreposição. 20% afirmou existir muito pequenos cortes, enquanto 50% afirmou existirem poucos. 10% detectou muito chuva, enquanto 30% detectou pouco. 60% detectou pouca saturação de médio. 20% detectou muita flutuação no áudio.

Dos entrevistados que não são músicos, 100% afirma ter detectado pouca sobreposição, não foram detectados pequenos cortes, chuva, saturação de médios ou flutuações no áudio.

Dos entrevistados que trabalham em estúdio ou com edição de áudio, 66,67% detectou pouca sobreposição, 33,33% detectou muitos pequenos cortes, 33,33% detectou poucos pequenos cortes, 66,67% detectou pouca saturação de médios e 66,67% detectaram muitas

flutuações no áudio.

Na comparação entre as amostras “violin” e “violin\_pitch\_P”, dos entrevistados que são músicos, 20% detectou pouca sobreposição, 10% detectou poucos pequenos cortes, 50% detectou poucos chuviscos, 40% detectou saturação de médios, 10% detectou poucas flutuações no áudio.

Dos entrevistados que não são músicos, não foram detectados sobreposição, pequenos cortes, chuvisco, saturação de médios nem flutuações no áudio.

Dos entrevistados que trabalham em estúdio ou com edição de áudio, não foram detectadas sobreposição, pequenos cortes nem flutuações no áudio, 66,67% detectou pouco chuvisco e pouca saturação de médios.

Na comparação entre as amostras “violin” e “violin\_pitch\_S”, dos entrevistados que são músicos, 10% detectou muita sobreposição, 20% detectou poucos pequenos cortes, 20% detectou muito chuvisco, 10% detectou pouco chuvisco, 40% detectou pouca saturação de médios, 20% detectou muita flutuação no áudio e 20% detectou pouca flutuação no áudio.

Dos entrevistados que não são músicos, não foram detectados sobreposição, pequenos cortes, chuvisco, saturação de médios nem flutuações no áudio.

Dos respondentes que trabalham com edição de áudio ou em estúdio, não foram detectadas sobreposição, pequenos cortes, 66,67% detectou muito chuvisco, 66,67% detectou pouca saturação de médios e 33,33% detectou poucas flutuações no áudio.

Na Tabela 2 é possível perceber que as amostras processadas pelo SMS (terminadas em “S”, à direita) tiveram uma avaliação inferior às amostras processadas pelo *Phase Vocoder* (terminadas em “P”, à esquerda).

Na primeira amostra, “drums”, contendo sons percussivos, 11,33% das opiniões acusaram muitos artefatos gerados, enquanto no SMS 27,11% das opiniões acusavam o mesmo.

A amostra “recital”, quando na operação de *time-stretch*, apresentou 57,83% de opiniões acusando de pequenos artefatos terem sido gerados no som, contra 24,67% do *Phase Vocoder*. Defeitos maiores ocorreram em proporções bem similares (8% do *Phase Vocoder* e 8,43% do SMS).

A amostra “recital”, quanto na operação de *pitch-shift*, apresentou um percentual baixo de erros grandes, no *Phase Vocoder*, com só 4,2% das opiniões atestando grandes defeitos, contra 11,33% do SMS, com quantidades similares de opiniões detectando pequenos artefatos, 26,44% do *Phase Vocoder* e 29,78% do SMS.

A amostra “violin”, processada na operação de *pitch-shift*, não detectou erros grandes

no *Phase Vocoder* e 17,56% das opiniões apontavam somente para poucos artefatos. As opiniões sobre o SMS apontam para artefatos maiores terem sido gerados, 7,77%, e menos artefatos pequenos terem sido gerados do que no *Phase Vocoder*, 12,67% contra 17,56%.

## 5 CONCLUSÃO

Sugere-se que o SMS é um algoritmo ainda em desenvolvimento. É um algoritmo bastante complexo e que, ao menos por enquanto, segundo os resultados obtidos neste trabalho, ainda não pode substituir o *Phase Vocoder*, pelo menos nas operações básicas de *time-stretch* e *pitch-shift*. Mesmo onde o algoritmo prometia suprir falhas, na área de detecção de parciais, ou seja, trabalhando com sons inarmônicos e percussivos, o SMS obteve uma avaliação inferior ao *Phase Vocoder*.

Os desenvolvedores do SMS sugerem um caso ideal de uso do SMS: que o instrumento esteja isolado e esteja num ambiente sem reverberação. Ainda que a amostra “violin” atendesse estes requisitos, sua avaliação foi mais negativa que a mesma amostra processada pelo *Phase Vocoder*.

As ondas processadas pelo SMS apresentavam um espectro mais limitado, tendo dificuldade em alcançar as frequências agudas, em média, acima de 17.000Hz, ainda que fossem utilizados os parâmetros de análise que obrigassem frequências ainda maiores serem detectadas. Quando não detectadas na parte determinística, os agudos eram detectados como ruídos, e tratados como sinal estocástico, sendo impossível definir seu espectro, pois a síntese da parte estocástica do SMS torna o sinal um ruído branco.

A frequência máxima que pode ser detectada pelo SMS é de 22.050Hz, o que deveria ser repensado pelos desenvolvedores da biblioteca, uma vez que arquivos de áudio com amostragem ainda maior podem permitir frequências mais agudas, e é interessante que estas sejam mantidas para fins de edição de áudio com mais qualidade.

O SMS aceita uma grande quantidade de parâmetros que permitem várias formas de análise do áudio. A grande quantidade de parâmetros torna o SMS um algoritmo complexo de ser utilizado. Há manuais que sugerem janelas específicas para determinados tipos de áudio; se este é mais estável ou as notas possuem um ataque maior. Esta forma de utilização, pode não ser tão clara nem tão óbvia para o público leigo.

Alguns parâmetros, como a quantidade de parciais a serem detectadas e o número de parciais que o arquivo pode conter, quando setados para valores muito altos destroem a onda do arquivo original, sendo impossível a utilização de todos os valores possíveis, segundo o manual.

O site dos desenvolvedores do SMS apresenta alguns detalhes interessantes sobre o algoritmo, como amostras mostrando a separação de uma voz da parte instrumental de uma música, ou de permitir a transformação do timbre de uma voz humana para um instrumento de

sopro ou de uma cachoeira. Estas operações produzem pequenas falhas, mas ainda assim o resultado é muito bom.

A habilidade do SMS em conseguir separar uma voz de um instrumental reside no fato de sua detecção de frequências passar por um algoritmo de “guias” que buscam “conectar” estas frequências, permitindo analisar a evolução de uma senoide. O SMS inclusive apresenta parâmetros, na sua etapa de análise, para que se especifique o quanto estas guias podem se desviar de seu caminho para se conectar com novas frequências. O *Phase Vocoder* não apresenta este recurso, simplesmente detectando as frequências que existem no áudio. Para efetuar a correlação entre eles, seria necessário um editor visual e, ainda assim, a operação estaria sendo feita manualmente.

## REFERÊNCIAS

- ANDRADE, Adriano O. et al. **Técnicas de Janelamento de Sinais**. Universidade Federal de Uberlândia. In: SEMINÁRIO DOS ESTUDANTES DE ENGENHARIA ELÉTRICA DA UFU, 3, 2000, Uberlândia, MG, Brasil, 2000. Disponível em:  
< <http://www.aoandrade.eletrica.ufu.br/Documents/T%E9cnicas%20de%20Janelamento%20de%20Sinais.pdf> >. Acesso em: 7 nov. 2010.
- BONFIM, Marlio. **Medidas Elétricas**. Disponível em:  
< <http://www.eletr.ufpr.br/marlio/medidas/apostila/apostila2c.pdf> >. Acesso em: 7 nov. 2010.
- BURRUS, Sidney C. et al. **The Cooley-Tukey Fast Fourier Transform Algorithm**. 2009. Disponível em: < <http://cnx.org/content/m16334/latest/> >. Acesso em 26 jan. 2011.
- CASTILHO, Saulo. **Análise e Ressíntese de Sinais Musicais**. 2008. 144 f. Dissertação (Pós Graduação em Matemática e Computação Científica) – Universidade Federal de Santa Catarina. Florianópolis, 2008. Disponível em: < [http://mtm.ufsc.br/pos/Saulo\\_Castilho.pdf](http://mtm.ufsc.br/pos/Saulo_Castilho.pdf) >. Acesso em: 26 jul. 2010.
- COOLEY, J.W., and TUKEY, J.W. **An algorithm for the Machine Calculation of Complex Fourier Series**. Mathematics of Computation, abr. 1965. Disponível em:  
< [www.utdallas.edu/~cpb021000/EE%204361/Great%20DSP%20Papers/Cooley%20FFT.pdf](http://www.utdallas.edu/~cpb021000/EE%204361/Great%20DSP%20Papers/Cooley%20FFT.pdf) >. Acesso em: 2011 jan. 12.
- DOLSON, Mark. **The Phase Vocoder: A Tutorial**. 2000. Acessado em: 24 out 2010. Disponível em: < <http://www.panix.com/~jens/pvoc-dolson.par> >.
- DUNN, Julian. **Anti-alias and anti-image filtering**: The benefits of 96kHz sampling rate formats for those who cannot hear above 20kHz. In: AES CONVENTION, 104, 1998, Amsterdam. Disponível em: < [www.nanophon.com/audio/antialia.pdf](http://www.nanophon.com/audio/antialia.pdf) >. Acesso em: 19 ago. 2011.
- ELLIS, Dan. P. **A Phase Vocoder in MATLAB**. Acesso em: 30 jul 2011. Disponível em:  
< <http://labrosa.ee.columbia.edu/matlab/pvoc/> >

FLANAGAN, James L. et al. Phase Vocoder. The Bell System Technical Journal. 9 nov. 1966. Disponível em: < [www.ee.columbia.edu/~dpwe/e6820/papers/FlanG66.pdf](http://www.ee.columbia.edu/~dpwe/e6820/papers/FlanG66.pdf) >. Acesso em: 16 ago. 2011.

FLANAGAN, M. T. **Michael Thomas Flanagan's Java Scientific Library**. Disponível em: < <http://www.ee.ucl.ac.uk/~mflanaga/java/FourierTransform.html> >. Acesso em: 25 jun. 2011.

HAYKIN, Simon, VEEN, Barry V. **Sinais e Sistemas**. Porto Alegre: Bookman, 2001.

JÄRVELÄINEN, Hanna I.; KARJALAINEN, Matti. **Importance of Inharmonicity in the Acoustic Guitar**. In: INTERNATIONAL COMPUTER MUSIC CONFERENCE. Barcelona, 2005. Disponível em: < [www.acoustics.hut.fi/~mak/PUB/InharmJarvelainen2005.pdf](http://www.acoustics.hut.fi/~mak/PUB/InharmJarvelainen2005.pdf) >. Acesso em: 30 out. 2010.

KOTREDES, Robert M. **Anti-aliasing of Images Using Line Sampling**. Disponível em: < [web.mit.edu/rsi/www/pdfs/papers/99/rkotrede.pdf](http://web.mit.edu/rsi/www/pdfs/papers/99/rkotrede.pdf) >. Acesso em: 18 ago. 2011.

KOZIEROK, Charles M. **Sample Rate**. Disponível em: < <http://www.pcguide.com/art/soundSampleRate-c.html> > Acesso em: 23 ago. 2011.

LAROCHE, Jean. DOLSON, Mark. **New Phase-Vocoder Techniques for pitch-shifting, harmonizing and other exotic effects**. In: IEEE WORKSHOP ON APPLICATIONS OF SIGNAL PROCESSING TO AUDIO AND ACOUSTICS , 1999, New York. Disponível em: < <http://www.ee.columbia.edu/~dpwe/papers/LaroD99-pvoc.pdf> >. Acesso em: 15 ago. 2010.

LOY, Gareth. **Musimathics**. Vol 1. The MIT Press, 2006. Londres. Inglaterra.

LYONS, Richard G. **Understanding Digital Signal Processing**. Prentice Hall, 1997.

MTG - Music Technology Group. **SMS Software Manual**. 2000. Disponível em: < <http://peabody.sapp.org/class/st2/link/SMS-software-manual.pdf> >. Acesso em: 3 ago. 2011.

SAMPAIO, Rubens; CATALDO, Edson; BRANDÃO, Alexandre de S. **Análise e**

**Processamento de Sinais (Notas em Matemática Aplicada)**. Rio de Janeiro: SBMAC, 2006. Disponível em: < <http://www.sbmac.org.br/boletim/arquivos2006/minicurso-22.pdf> >. Acesso em: 15 ago. 2011.

SERRA, Xavier, SMITH III, Julius O. **PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation**. In: PROCEEDINGS OF THE INTERNATIONAL COMPUTER MUSIC CONFERENCE. Tokyo, 1987. Disponível em: < <https://ccrma.stanford.edu/~jos/parshl/> >. Acesso em: 15 ago. 2010.

SERRA, Xavier. **Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition**. Computer Music Journal, vol 14, no 4. 1990. Massachusetts Institute of Technology. Disponível em: < <http://mtg.upf.edu/node/304> >. Acesso em: 14 maio 2011.

SMITH, Julius O. SERRA, Xavier. **PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation**. In: INTERNATIONAL COMPUTER MUSIC CONFERENCE, 1987, Tokyo. Disponível em: < <https://ccrma.stanford.edu/~jos/parshl/> >. Acesso em: 15 ago. 2010.

SOVIERZOSKI, Miguel. **Notas de Aula do curso de Engenharia da Computação da Universidade Tecnológica Federal do Paraná**. 2010. 2o. Semestre de 2010.

STRANDH, Erik. **Polyphonic Pitch Modification Using Phase Vocoder Techniques**. 2010. 32 f. Dissertação. Lunds Universitet. NatFak, 2010. Disponível em: < [http://fileadmin.cs.lth.se/cs/Personal/Pierre\\_Nugues/memoires/erik/polyphonic\\_pitch\\_modification.pdf](http://fileadmin.cs.lth.se/cs/Personal/Pierre_Nugues/memoires/erik/polyphonic_pitch_modification.pdf) >. Acesso em: 26 jul. 2010.

WEISSTEIN, Eric W. **Square Wave**. 2008. Disponível em: < <http://mathworld.wolfram.com/SquareWave.html> >. Acesso em: 20 jun. 2011



## ANEXOS

ANEXO A – Código-fonte do programa utilizado para calcular a soma das amostras dos arquivos contendo a sobreposição das ondas sintéticas. Este código utiliza a linguagem de programação C++ e a biblioteca libsndfile.

```
#include <iostream>
#include <stdio.h>
#include <cmath>
#include <sndfile.h>

using namespace std;

int main(int argc, char *argv[])
{
    // nome do diretório
    string dir = "/media/CHRONICLES/ondas/merge/";

    // arquivos de áudio sobrepostos
    string files[] = {
        "sawtooth_220hz_pvoc_merge.wav", "sawtooth_220hz_sms_merge.wav",
        "sawtooth_2s_pvoc_merge.wav", "sawtooth_2s_sms_merge.wav",
        "square_220hz_pvoc_merge.wav", "square_220hz_sms_merge.wav",
        "square_2s_pvoc_merge.wav", "square_2s_sms_merge.wav",
        "triangle_220hz_pvoc_merge.wav", "triangle_220hz_sms_merge.wav",
        "triangle_2s_pvoc_merge.wav", "triangle_2s_sms_merge.wav"};

    for (int i = 0 ; i < sizeof(files); i++)
    {
        string fileName = dir+files[i];
        char* fullDir = (char*) fileName.c_str();

        // informações do arquivo de áudio
        SF_INFO infileInfo;

        SNDFILE *infile = sf_open(fullDir, SFM_READ, &infileInfo);

        double audio[infileInfo.frames];
        int samples = sf_read_double(infile, audio, infileInfo.frames);
    }
}
```

```
double somatorioAmostras = 0;
for (int j = 0; j < samples; j++)
{
    double amostra = fabs( audio[j] );
    somatorioAmostras += amostra;
}

string out = "Somatório das amostras " + files[i] + ": %.3f\n";
printf( ((char*)out.c_str()), somatorioAmostras );
}

return 0;
}
```