

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica
Departamento de Semicondutores,
Instrumentação e Fotônica

Transformações sonoras através de
operações timbrais

Este exemplar corresponde à redação final da tese
defendida por Jose Eduardo Fornari
Novo Jr e aprovada pela Comissão
Julgadora em _____
Orientador

Por : José Eduardo Fornari Novo Junior
Orientador: Furio Damiani

Dissertação submetida como requisito
parcial para a obtenção do título de Mestre em
Engenharia Elétrica

Campinas, junho de 1995

210113

@

UNIDADE	BC		
N.º CHAMADA:	UNICAMP		
	N859t		
V.	Ex.		
TOMBO B.	30.685		
PROC.	284197		
C	<input type="checkbox"/>	D	<input checked="" type="checkbox"/>
PREÇO R.B.	11,00		
DATA	29/05/1997		
N.º CPD			

CM-00098430-0

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

N859t

Novo Junior, José Eduardo Fornari
Transformações sonoras através de operações
timbrais / José Eduardo Fornari Novo Junior. --Campinas,
SP: [s.n.], 1995.

Orientador: Furio Damiani.
Dissertação (mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica.

1. Música - Acústica e física. 2. Engenharia acústica.
3. Processamento de som por computador. 4. Som -
Técnicas digitais. 5. Análise harmônica. 6. Fourier,
Transformações de. I. Damiani, Furio. II. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica
III. Título.



Dedico este trabalho a :

Sylvia Mantovani Novo
José Eduardo Fornari Novo
Luciana Beatriz Chagas

Agradecimentos:

Agradeço ao Professor Dr. **Furio Damiani** pela orientação deste trabalho.

Ao amigo **Jaime Manuel Ribeiro E. Santo**.

Ao Dr. **Jonatas Manzolli**, do NICS, Núcleo Interdisciplinar de Comunicações Sonoras, da UNICAMP.

Ao Professor Dr. **Peter Jürgen Tatsch**, do DSIF.

Ao Professor Dr. **Rafael Santos Mendes**, do DCA.

Ao **Marcelo Jara**, colega da pós-graduação do DSIF.

À **Maria Raquel Mendes Brito**, da secretaria do DSIF.

Ao **Departamento de Telemática**, pela utilização do seu laboratório.

Ao estúdio de gravação **Dimas**.

Resumo:

O som é abordado, pelo menos, por três ramos da ciências: a Acústica, a Fisiologia (da audição humana) e a Psicoacústica. A intersecção deste conhecimento, em linhas gerais, define: “o que”, “por quê” e “como” escutamos. Dado isto, torna-se possível dimensionar um sistema de processamento de sinais de áudio que represente e manipule um som analógico (contínuo no tempo) com níveis de ruídos e perdas abaixo dos limites da percepção auditiva normal, que é chamado de “som de alta fidelidade” (HF). Devido ao crescente aumento da capacidade de memória e velocidade dos processadores digitais de sinais (DSPs), atualmente tornou-se factível projetar um sistema DSP para áudio (ADSP), de alta fidelidade e que trabalhe em tempo-real, ou seja, um sistema ADSP cujo tempo de retardo entre o som de entrada e o de saída não seja percebido pela audição humana.

A Psicoacústica trata das percepções subjetivas da audição humana em relação às grandezas da Acústica; como o *loudness* (a percepção subjetiva da intensidade sonora), o *pitch* (a percepção subjetiva da frequência sonora) e o *quality*, ou timbre (a percepção subjetiva das similaridades sonoras). No caso do timbre, sabe-se que este pode ser representado como: a variação contínua no tempo das intensidades dos harmônicos do espectro do som. Um sistema de ADSP HF que transforme o espectro em tempo-real, através de operações que possam ser continuamente re-parametrizados em tempo-real, pode manipular o timbre sonoro. O projeto de um sistema deste tipo é o objetivo deste trabalho.

Este sistema trabalha com sons de entrada e saída analógicos. No entanto, o processamento é digital. Desse modo existe a necessidade de pré-processamento (filtragem e controle de nível) e conversão A/D e D/A adequados para garantir o desempenho HF do sistema. Os espectros do som discretizado de entrada são obtidos através do algoritmo rápido da transformada discreta de Fourier, a FFT. Cada saída da FFT equivale a um espectro discreto de um intervalo de tempo do som. Utilizando-se intervalos de tempo suficientemente curtos para não serem notados pela percepção auditiva, cada saída da FFT representará um espectro “instantâneo” do som. Cada espectro é manipulado por um operador timbral, com um dado parâmetro, e transformado novamente num trecho de som através da IFFT (FFT inversa) antes da próxima saída da FFT. Assim tem-se a manipulação do timbre do som.

A parametrização *on-line* das operações timbrais pode ser feita por controladores externos e/ou informações do passado do som (espectros anteriores ao que está sendo manipulado num dado instante). Assim os espectros obtidos em cada FFT são armazenados numa pilha de vetores que, para um instante de tempo, pode ser vista como uma matriz. Desta matriz retira-se toda a informação sobre o som, ou na forma gráfica, representando a superfície sonora (superfície contínua que mostra a variação da envoltória do espectro no tempo), ou através da extração de parâmetros para as operações timbrais. Assim é possível transformar e observar a transformação do som em tempo-real.

Neste trabalho foi desenvolvido o estudo e projeto para um sistema de síntese timbral controlada pelo som analógico de entrada. Foram feitas simulações no MATLAB 4.0 para cada passo do processamento do som, tais como: o pré-processamento e a conversão A/D e D/A, a obtenção do espectro discreto pela FFT e o retorno ao domínio do tempo pela IFFT, o armazenamento dos espectros na forma de pilha de vetores, as operações timbrais e os resultados alcançados em termos gráficos e auditivos (no padrão u-law). No final do trabalho é proposto um projeto para o sistema, utilizando o DSP 96002 (IEEE floating-point, dual-port processor) da MOTOROLA.

Índice

Agradecimentos.....	III
Resumo.....	IV
Índice.....	V
Capítulo 1: Introdução.....	1-1
1.1 Processamentos de audio.....	1-3
1.2 As transformações sonoras pelas operações timbrais.....	1-7
1.3 Organização da tese.....	1-8
Capítulo 2: Definição e representação do som	2-1
2.1 A percepção sonora humana.....	2-1
2.2 O processamento digital de audio.....	2-5
Capítulo 3: Anteprojeto do sistema.....	3-1
3.1 A superfície sonora.(S.S.).....	3-1
3.2 A matriz harmônica (M.H.).....	3-2
3.3 O operador timbral (O.T.).....	3-3
3.4 O sistema de transformação sonora.....	3-5
Capítulo 4: Simulação do sistema em software.....	4-1
4.1 A simulação da montagem da MH.....	4-1
4.2 A simulação das operações timbrais.....	4-2
4.3 Os resultados da simulação de MH.....	4-4
4.4 Os resultados da simulação das operações timbrais.....	4-10
Capítulo 5: Projeto do sistema.....	
5.1 Do som analógico de entrada ao operador timbral.....	

5.2 O processamento das operações timbrais.....	
5.3 Do operador timbral ao som analógico de saída.....	
5.4 Visão geral do sistema.....	
5.5 Projeto do sistema utilizando o DSP 96002.....	
5.6 Considerações finais.....	
Conclusões e comentários.....	
Apêndice 1: A DFT e a FFT.....	
Apêndice 2: Os programas da simulação em software.....	

Lista de figuras

- 1.1 A evolução temporal de um som complexo.
- 1.2 Som emitido por um violoncelo.
- 1.3 Diagrama esquemático de um vocoder.
- 1.4 Diagrama de um circuito eletrônico para síntese aditiva.
- 1.5 Síntese subtrativa.
- 1.6 Exemplo de síntese FM com dois osciladores.
- 1.7 Exemplo de distribuição dos *samples* por região de frequência da interface controladora.
- 1.8 Distorção de um sinal de entrada.
- 1.9 Diagrama esquemático do sistema de transformação sonora por operações timbrais.

- 2.1 Envoltória de um som de clarinete.
- 2.2 As curvas isofônicas de Fletcher-Munson..
- 2.3 Percepção humana da intensidade sonora.
- 2.4 Sensibilidade, em $\Delta f / f$, em função da frequência f .
- 2.5 O *pitch* correspondente às notas de três oitavas da escala musical.
- 2.6 Digitalização de $s(t)$ pela modulação (produto) com a função $m(t)$.
- 2.7 Processo de amostragem de $a(t)$ visto no domínio da frequência.
- 2.8 Janelamento de uma seqüência de N pontos.
- 2.9 Entrelaçamento das seqüências de entrada, de 50%.
- 2.10 Diagrama esquemático do algoritmo *butterfly* para cálculo da FFT.

- 3.1 A representação da superfície sonora, SS.
- 3.2 Volume limite máximo da percepção auditiva humana.
- 3.3 A representação da SS discreta pela MH.
- 3.4 Estrutura básica do operador timbral, OT.
- 3.5 Ciclo de operação do OT.
- 3.6 Exemplo de estrutura em hardware de um operador timbral
- 3.7 Estrutura esquemática do sistema de transformação sonora.
- 3.8 Organização do sistema de transformação sonora.

- 4.1 Simulação da montagem da MH.
- 4.2 Simulação das operações timbrais.
- 4.3 Som de violoncelo.
- 4.4 Som do violoncelo, envoltória no tempo.
- 4.5 Análise do som de violoncelo.
- 4.6 Superfície sonora de uma senóide de 1280 Hz.
- 4.7 Efeito de aliasing para uma senóide.
- 4.8 Efeito de aliasing para três senóides.
- 4.9 O efeito de *frequency leakage* em uma senóide.
- 4.10 *Frequency leakage*, para $bin=128$ Hz.
- 4.11 *Frequency leakage* de som com 10 harmônicos.
- 4.12 Som de oboé.
- 4.13 Som janelado, sem entrelaçamento.
- 4.14 Som janelado, com entrelaçamento de 25%.
- 4.15 Som janelado, com entrelaçamento de 50%.
- 4.16 Som de apito de trem.
- 4.17 Som modificado de apito de trem.
- 4.18 Som de violino.
- 4.19 Som modificado de violino.
- 4.20 Som de violoncelo.
- 4.21 Modificação da envoltória da SS do som de violoncelo.

- 4.22 Som modificado de violoncelo.
- 4.23 Som de trombone.
- 4.24 Som modificado de trombone.
- 4.25 Som modificado de trombone.
- 4.26 Som de clarinete.
- 4.27 Som modificado de clarinete.
- 4.28 Trecho de *Aleluia*, de Händel.
- 4.29 Preserva apenas os três maiores harmônicos.
- 4.30 Som de flauta com harmônico fundamental $f_0=880$ Hz.
- 4.31 Som de flauta. Deslocamento do pitch original, de 880 Hz para 220Hz.

- 5.1 Parte inicial do sistema.
- 5.2 Pré-processamento do sinal de entrada e conversão A/D.
- 5.3 Montagem do vetor de entrada da FFT
- 5.4 Cálculo da FFT e do vetor controlador.
- 5.5 Estrutura básica do operador timbral.
- 5.6 Um ciclo de operação do operador timbral.
- 5.7 O cálculo da IFFT e a montagem do vetor do som modificado.
- 5.8 O pós-processamento e a saída do som analógico transformado.
- 5.9 Diagrama esquemático dos estágios do sistema.
- 5.10 Diagrama de tempos do processamento do sistema.
- 5.11 Diagrama de tempos do processamento do sistema, para 2 processadores.
- 5.12 Diagrama de tempos do processamento do sistema evitando-se transferências de dados desnecessárias.
- 5.13 Diagrama do sistema com dois DSPs.
- 5.14 Mapeamento dos harmônicos no eixo da frequência em função da percepção.

Capítulo 1: Introdução

O som é constituído por vibrações mecânicas que se propagam num meio material, na forma de compressões e expansões sucessivas às quais chamamos de sinal sonoro. O sinal sonoro pode ser convertido num sinal elétrico $s(t)$ que o pode representar fielmente. Do mesmo modo, $s(t)$ pode ser convertido em um som.

A acústica define três grandezas principais para tratar o som: a intensidade sonora (em decibéis), a frequência (em Hertz), e a duração (em segundos). A fisiologia da audição determina os limites para a percepção e resolução dessas grandezas acústicas. A psicoacústica estuda a percepção dessas grandezas pela audição humana, através da qual um som pode ser reconhecido e discriminado.

Considera-se como a componente sonora mais simples, a oscilação da pressão cuja representação temporal é expressa por uma função senoidal do tipo $A \cdot \sin(\omega t + \phi)$. Sua representação no domínio da frequência é um número complexo cuja magnitude é o valor da intensidade sonora A , cuja fase é o valor de ϕ e cuja frequência f está associada à velocidade angular ω por $f = \omega / 2\pi$. Essa componente é representada no eixo da frequência por uma magnitude A na frequência f .

Utilizando-se a teoria matemática desenvolvida por Fourier é possível, dentro de certas condições, representar sinais do tipo $s(t)$ por uma somatória de funções ortogonais, como o são as funções seno e coseno. Para um $s(t)$ qualquer, a somatória (ou série) tem um número infinito de termos. Admitindo-se um certo valor de erro pode-se truncar esta série, determinando-se então o número finito de funções ortogonais que deverão ser usadas para representar $s(t)$. Cada uma delas é um harmônico de intensidade A_n e frequência f_n . O conjunto das componentes de frequência de um sinal $s(t)$ constituem seu espectro de frequência. Na figura 1.1 vê-se um exemplo de um som e suas componentes em frequência obtidas através de filtragem passa-faixa.

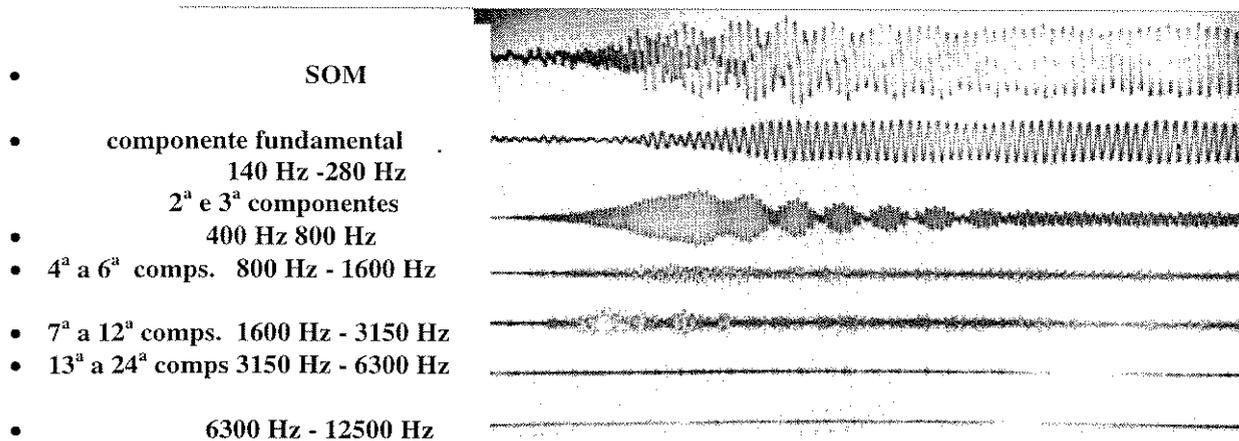


Fig. 1.1 A evolução temporal de um som complexo (orgão de tubo) e de suas componentes em frequência [32]. As faixas de frequência indicam os limites do filtro utilizado para obter as componentes (cps = Hz). Os *partials* correspondem aos harmônicos da frequência fundamental.

Assim, pode-se representar $s(t)$ através dos seus harmônicos componentes. A representação gráfica da evolução dos harmônicos no tempo é uma superfície cujas dimensões são amplitude, frequência e tempo, a qual denominaremos de superfície sonora (SS). A SS permite visualizar as características do som que caracterizam o seu timbre.

A teoria da amostragem [9] indica que se pode representar um sinal contínuo $s(t)$ por uma sequência de amostras discretas $s(n) = s(t) |_{t=n.T}$; desde que a taxa de amostragem $F_s = 1/T$ seja

maior que duas vezes a maior frequência componente de $s(t)$, f_H . A amostragem é feita por um conversor analógico-digital (A/D). Reciprocamente, $s(n)$ pode ser convertido em $s(t)$ por um conversor digital-analógico (D/A). Um sinal de áudio real tem sempre um espectro de frequência limitado.

Portanto, é possível representar a superfície sonora por uma matriz cujos elementos são as intensidades do conjunto discreto de harmônicos para os instantes discretos escolhidos. A matriz harmônica MH é a representação discreta da superfície sonora.

Os harmônicos de $s(n)$ podem ser obtidos pela transformada discreta de Fourier, a DFT que fornece um conjunto finito e discreto de harmônicos. O algoritmo utilizado neste trabalho para o cálculo da DFT é a transformada rápida de Fourier, FFT (ver apêndice 1). A FFT requer $N=2^v$ amostras de $s(t)$ para gerar um espectro discreto com $N/2$ pontos.

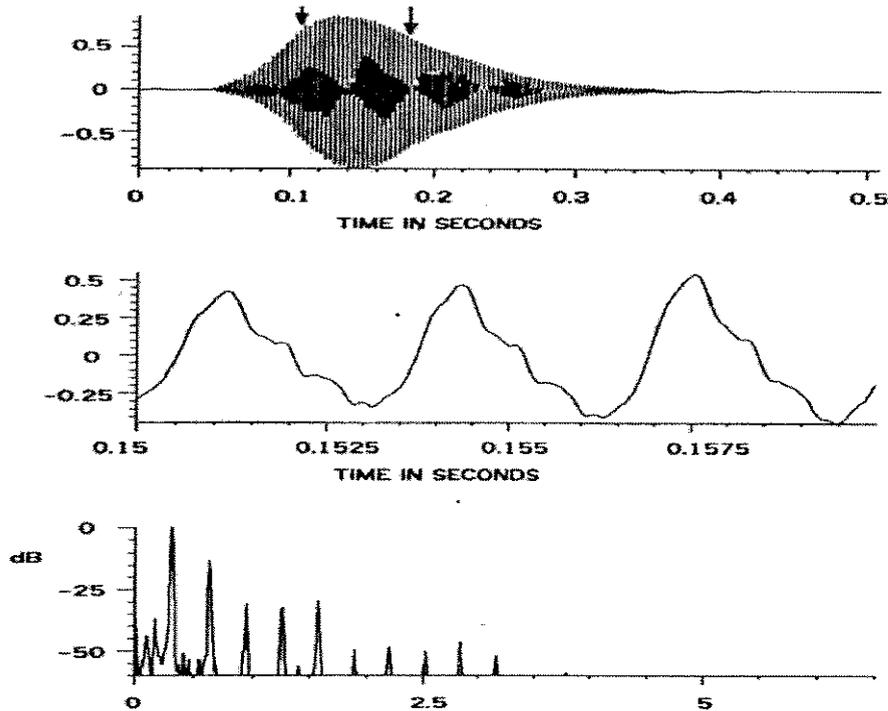


Fig. 1.2 Som emitido por um violoncelo. (a) envoltória temporal. (b) detalhe do som no tempo. (c) harmônicos obtidos pela DFT de um trecho de 80 ms [Moorer].

Um processamento de sinais de áudio pode manipular a SS correspondente a $s(t)$. No processamento analógico, manipula-se o espectro da $s(t)$ continuamente. Como consequência tem-se duas SS, uma representando o $s(t)$ original, e a outra, o $s(t)$ modificado, com um retardo devido ao tempo de processamento do sinal.

O processamento digital de sinais de áudio modifica a SS correspondente ao $s(n)$. No processamento em tempo real, manipula-se o espectro discreto de $s(n)$ (tendo $N/2$ pontos), obtido pela FFT a partir de N pontos de $s(n)$, correspondentes a um intervalo de tempo N/F_s . Assim tem-se duas SS, representadas por duas MH, correspondentes ao $s(n)$ original e ao $s(n)$ modificado. O processamento digital de sinais de áudio tem por sigla ADSP (*Audio Digital Signal Processing*).

O objetivo deste trabalho é o estudo, dimensionamento e projeto de um sistema ADSP para a geração de sons obtidos por transformações timbrais em tempo real sobre um sinal de áudio de controle. O sinal de áudio pode provir de qualquer tipo de fonte. A transformação timbral é feita por uma série de operações parametrizáveis sobre os espectros do sinal de áudio de controle. Os parâmetros das operações timbrais podem ser modificados em tempo real.

1.1 Processamentos de áudio

O som usualmente é manipulado por processamento analógico ou digital, podendo ou não ser em tempo real. Estas classes de processamento apresentam vantagens e desvantagens.

O processamento analógico é normalmente bastante rápido e costuma ser em tempo real, no entanto não tem a flexibilidade da programação digital. O processamento digital tem a flexibilidade da programação, mas pode requerer operação em tempo não-real, quando envolver um grande número de cálculos. O desenvolvimento de processadores digitais muito rápidos a um custo relativamente baixo, tornou possível a construção de equipamentos ADSP que trabalham em tempo real.

Quando o processamento não é executado em tempo real, há a necessidade de armazenamento do som por gravadores analógicos ou digitais, memória RAM, disquetes, etc. Esta etapa adicional pode introduzir modificações não desejadas no sinal original (distorções harmônicas, ruídos, etc.) além de encarecimento no custo.

Chamaremos de **síntese** sonora ao processamento de áudio visando a criação de novos sons a partir de elementos sonoros e/ou não sonoros; será chamado de **análise** o processamento para a medida das características de um dado som. Um exemplo de aparelho para a análise sonora é o analisador de espectro. Ele pode ser constituído por um banco de filtros passa-faixa de banda estreita que cobrem uma região do espectro de frequência do som. A saída de cada filtro está relacionada a uma componente em frequência do espectro. Estas componentes podem ser vistas numa saída gráfica, como um diagrama de colunas, onde cada coluna representa a magnitude de uma componente em frequência do som analisado.

Um aparelho musical que utiliza um analisador de espectro é o *vocoder* [21]. Conforme visto na figura 1.3, trata-se de um banco de filtros passa faixa (cada filtro centrado em f_n) acoplado a um banco de osciladores (cada oscilador com frequência f_n). O *vocoder* permite analisar o espectro de $s(t)$ pela análise da saída $s'(t)$.

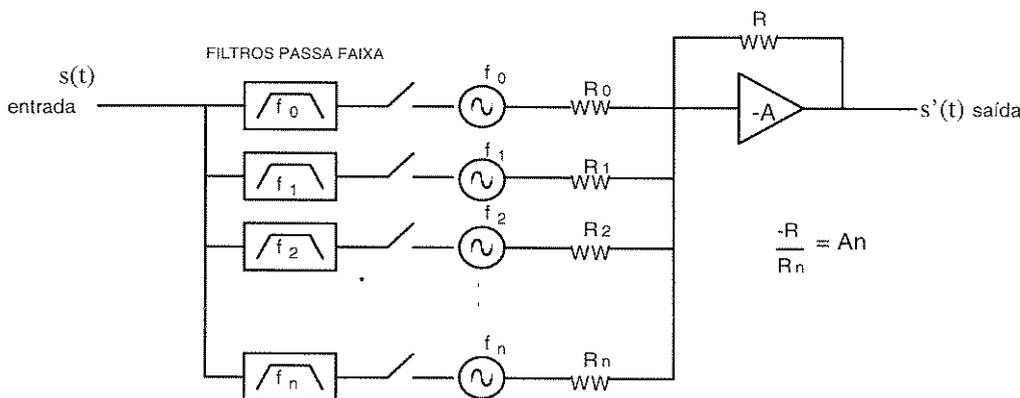


Fig. 1.3 Diagrama esquemático de um vocoder.

Numa implementação digital, o banco de filtros foi substituído pelo cálculo sequencial da FFT de $s(n)$, para pequenos trechos de som, a FFT *short-time*. Esta viabilizou o vocoder digital em tempo real, usando o DSP-56000 da Motorola [19]. O equipamento trabalha com sons de entrada cujo espectro de frequência está limitado em 8kHz e permite a sua análise em tempo real.

Pode-se também utilizar técnicas de filtragem para a análise da envoltória do espectro de um som de entrada. Desse modo, as principais características deste espectro podem ser observadas sem levar em conta todas as suas componentes em frequência.

Síntese: Dividiremos aqui as técnicas de síntese sonora em: sínteses lineares, não-lineares e por transformação sonora.

Sínteses lineares: Um sinal harmônico pode ser gerado por um oscilador eletrônico e pode ser expresso por:

$$s(t) = A \cdot \text{sen}(2\pi \cdot t + \phi). \quad (1.1)$$

Podem-se sintetizar sons pela adição de harmônicos [21]. Esta técnica é conhecida como síntese aditiva. Conforme é mostrado na figura 1.4, pode-se criar um novo som somando os sinais de vários osciladores senoidais e controlando sua amplitude A , frequência f e fase ϕ , podem-se obter sons cujo espectro de frequência é formado pelos harmônicos gerados pelos n osciladores

$$s(t) = \sum_n A_n \cdot \text{sen}(2\pi \cdot t_n \cdot t + \phi_n). \quad (1.2)$$

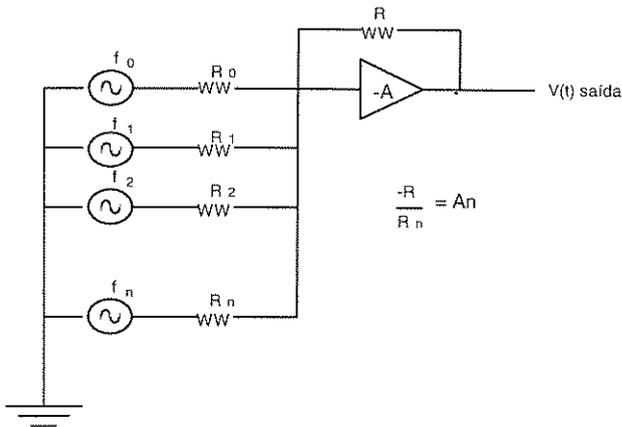


Fig. 1.4 Diagrama de um circuito eletrônico para síntese aditiva.

A síntese subtrativa gera um som a partir de uma fonte sonora cujo espectro tem mais harmônicos que o som sintetizado de saída. Na figura 1.5 tem-se o diagrama da síntese subtrativa. Nela a fonte sonora geradora (cujos parâmetros são: extensão e envoltória da amplitude e fase espectro de frequências) é filtrada por um banco de filtros também parametrizável (família e tipo de filtro, decaimento...). Deste modo pode-se obter o espectro de frequência desejado na saída. A fonte sonora normalmente utilizada é um gerador de ruído, que possui um espectro bastante rico em harmônicos.

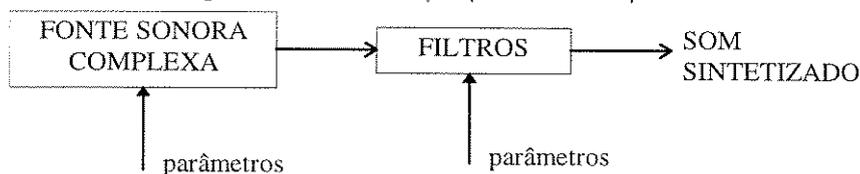


Fig. 1.5 Síntese subtrativa.

Sínteses não-lineares: Diferentemente das técnicas de síntese linear, as não-lineares permitem a geração de novos harmônicos. Um desses processos é a síntese por modulação da frequência, ou síntese FM. Nesta, a frequência de um oscilador é modulada por outro. Assim, a frequência resultante não é fixa, variando de acordo com o sinal modulador. Tem-se um espectro variante ao longo do tempo. Como exemplo (ver fig. 1.6), no caso de uma modulação senoidal de um sinal senoidal, obtêm-se:

$$s(t) = A_1 \cdot \text{sen}(2 \cdot \pi \cdot fc + A_0 \cdot \text{sen}(2 \cdot \pi \cdot fm)) \quad (1.3)$$

Na síntese FM, um pequeno número de osciladores pode sintetizar sons com um espectro bastante rico e dinâmico.

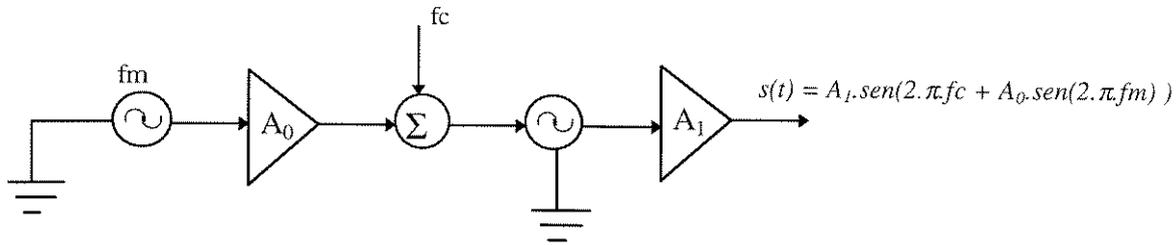


Fig. 1.6 Exemplo de síntese FM com dois osciladores

As sínteses por transformações sonoras são os processamentos sonoros lineares e não-lineares que geram um novo som com base num som de entrada. As transformações sonoras podem ser em tempo real ou não. Atualmente a maioria dos equipamentos de síntese por transformação sonora são digitais, ou ADSP. Existem vários tipos que englobam um repertório de transformações em um único módulo. São equipamentos comuns em estúdios de gravação, ou em performances sonoras.

- *Sampler*: É feita uma transformação sonora com um ADSP em tempo não-real para a preparação de amostras de um som natural, *samples*, que serão utilizadas posteriormente em síntese. Os *samples* são costumeiramente de instrumentos musicais acústicos. O som digitalizado é separado e editado em trechos representativos de sua envoltória temporal para posteriormente serem acessados em tempo real por uma interface controladora MIDI (*Musical Interface Digital Instrument*) ou um computador. Os processos de edição sonora mais utilizados são: compressão/expansão da amplitude, modificação da duração temporal, *loopings* (repetição de uma amostra), modificações na envoltória temporal.

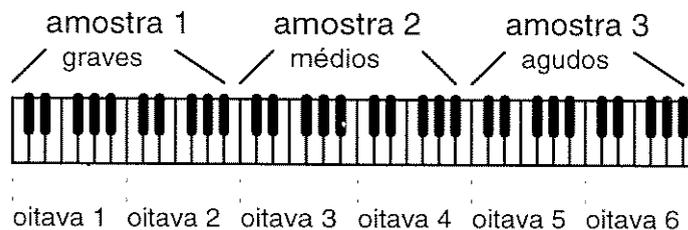


Fig. 1.7 Exemplo de distribuição dos *samples* por região de frequência da interface controladora

Geralmente o processamento de som no *sampler* é feito como no exemplo da figura 1.7, onde cada amostra representa um estado característico do timbre do instrumento musical que está sendo simulado (sons graves ou agudos, fortes ou fracos, *legatos* ou *staccatos*). Em seguida cada amostra é comprimida/expandida no tempo e na intensidade para que possa representar diversas notas na escala musical. O padrão de representação digital utilizado pelos *samplers* atualmente é o mesmo do *Compact Disk*, ou seja, resolução (comprimento da palavra binária que representa a amplitude de cada amostra) de 16 bits e taxa de amostragem de 44,1 KHz.

- Transformações na duração temporal: Modificam a duração temporal do som. A seguir listamos alguns equipamentos que executam transformações deste tipo:
 - *Delay*: insere um atraso entre o som de entrada e a saída.
 - Reverberador: simula o efeito de reverberação sonora, ou seja, as múltiplas repetições do som.
 - Câmara de eco: simula o efeito do eco, para vários tempos de reflexão.
- Transformações no espectro da frequência: Atuam primordialmente no espectro de frequência do som. Abaixo descrevemos as mais importantes.
 - Filtros: Removem uma ou várias regiões do espectro sonoro. Desse modo podem-se eliminar ruídos indesejáveis que estejam situados numa região do espectro, bem como intensificar componentes
 - Deslocador espectral: Conhecido como *shift-delay*, desloca o espectro de frequência do som, o que corresponde a tornar o som mais grave ou mais agudo, sem alterar sua duração.

- *Chorus*: Cria o efeito de várias vozes em uníssono quando a entrada é apenas uma voz, ou som. A sensação das vozes em uníssono é dada pelas pequenas diferenças existentes entre cada voz. Existem duas implementações para o *chorus*. Uma é realizada por um banco de *delays*. A outra por um banco de filtros que variam aleatoriamente a amplitude e a frequência de regiões do espectro do som de entrada. Esse segundo tipo é o mais utilizado e usa também uma manipulação na intensidade.
- Transformações na intensidade: atuam sobre a intensidade sonora; eis algumas:
 - Prolongamento: mantém constante a intensidade sonora a partir de um momento qualquer da envoltória temporal de um som, prolongando indefinidamente sua duração. É utilizado em guitarras elétricas, para que o som de cada nota perdure até a ocorrência da próxima nota.
 - Equalização: manipulação da amplitude de regiões específicas do espectro efetuado por um banco de filtros com atenuação controlada.
 - Compressão: limitação dos níveis máximo e mínimo da intensidade do sinal sonoro. É utilizada para evitar variações excessivas, como no caso de gravações sonoras.
 - Modificação da envoltória: alteração da envoltória temporal do sinal sonoro nos tempos de *attack*, *decay*, *sustain* e *release* (ver glossário).
- Transformações mistas: são aquelas que atuam conjuntamente em duas ou mais grandezas sonoras simultaneamente. Dentro desta classe de transformações pode-se destacar:
 - Distorção: utiliza a amplificação não-linear do sinal sonoro para gerar efeitos sonoros. O processo não-linear de amplificação gera novos harmônicos na saída.

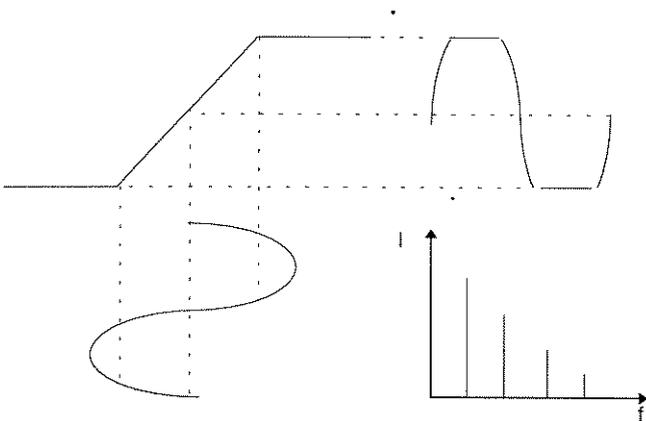


Fig. 1.8 Distorção. O sinal de entrada é modificado pela relação não linear de amplificação. O sinal senoidal de entrada tem seus picos achatados, gerando uma série de harmônicos no sinal de saída.

As transformações sonoras que utilizam processamento digital em tempo real são as mais poderosas, pois possuem: rapidez de processamento, flexibilidade de programação e observabilidade em tempo real. Muitos equipamentos possuem estas características, como os *delays*, *chorus* e *reverbers*, mas seu nível de manipulação do sinal de entrada é baixo, estando costumeiramente dimensionados para realizar um tipo de transformação única e fixa do som de entrada.

1.2 As transformações sonoras pelas operações timbrais

As operações timbrais fazem a transformação do espectro de um som. O espectro é transformado pela manipulação da amplitude, frequência e duração temporal de seus harmônicos.

O desenvolvimento desse equipamento abre uma nova categoria de manipulação do timbre, de maneira acessível e interativa. A possibilidade da audição em tempo real dos resultados sonoros permite o aprendizado intuitivo da manipulação do timbre pelo usuário do sistema.

O sistema que proposto neste trabalho é mostrado esquematicamente na Fig 1.8. Suas partes principais são:

- pré-processamento e conversão A/D,
- cálculos da FFT e do vetor das magnitudes do espectro de frequência,
- operador timbral,
- interface de entrada para os parâmetros do operador timbral,
- matriz harmônica,
- cálculo da IFFT (operação inversa da FFT) e conversão D/A.

As operações timbrais são especificadas por um controle externo, que pode ser tanto uma interface controladora quanto um algoritmo de programação executado por um computador. As transformações sonoras ocorrem no operador timbral. O operador timbral recebe a instrução de operação timbral, decodifica-a, executa as operações correspondentes sobre o espectro e le e/ou atualiza a matriz harmônica.

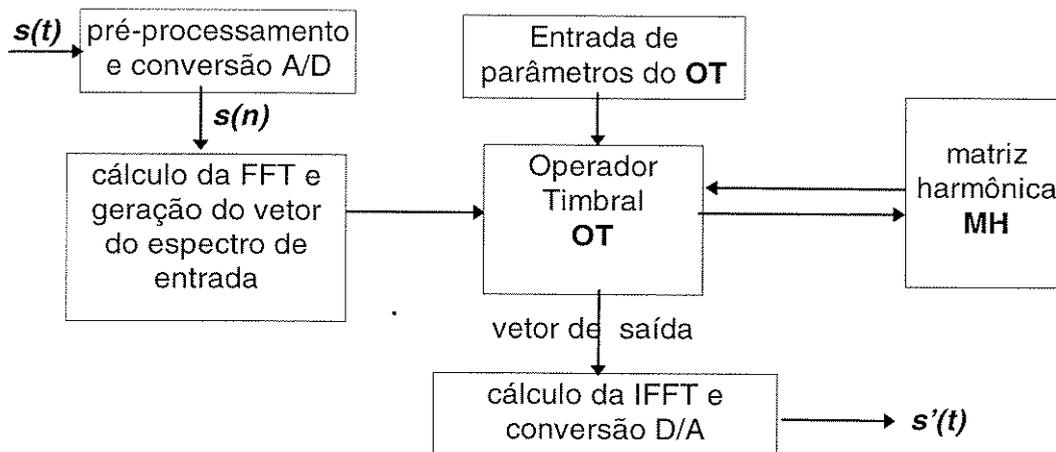


Fig. 1.9 Diagrama esquemático do sistema de transformação sonora por operações timbrais, onde $s(t)$ é o sinal acústico de entrada, $s(n)$ é o sinal amostrado na taxa F_s , $s'(t)$ é o sinal de áudio de saída.

Este sistema se mostra diferente dos outros equipamentos de transformação sonora em tempo real porque:

- permite a elaboração de operações timbrais quaisquer, que podem ser programadas;
- permite a modificação dos parâmetros dessas operações em tempo real, ou seja, permite a manipulação interativa do timbre.
- Utiliza o sinal acústico de entrada como um sinal de controle; o operador timbral gera a saída a partir da SS do sinal de entrada e não apenas modificando seu timbre.
- Inclui operadores cuja função depende do *passado* do som gerado, criando uma gama de novas possibilidades de transformação.

1.3 Organização da tese

Este trabalho está organizado em cinco capítulos e dois apêndices.

O **primeiro capítulo** fornece uma introdução aos processamentos utilizados na música contemporânea. Dentro desse universo é inserido o sistema de transformação sonora por operações timbrais, sendo apresentadas suas principais características

O **segundo capítulo** trata das definições do som, apresentando-se suas características básicas, os limites da sua percepção e o reconhecimento de suas grandezas pela audição humana. Também são abordadas a representação digital do som e a sua representação no domínio da frequência, conceitos fundamentais para este trabalho.

No **terceiro capítulo** são fundamentados os conceitos de: superfície sonora, matriz harmônica, operação timbral e de como o som pode ser representado para visualizar os elementos componentes de seu timbre. No final do capítulo apresentam-se e discutem-se genericamente as partes de um sistema de transformação sonora por operações timbrais.

No **quarto capítulo**, são mostrados os principais resultados das simulações feitas por computador do sistema proposto. Mostram-se as representações gráficas da superfície sonora para diferentes condições de processamento do som e os resultados de diversas operações timbrais desenvolvidas.

No **quinto capítulo** é feito o dimensionamento e o projeto do sistema. São discutidos os limites, características e condições de contorno para sua realização. É proposta uma implementação utilizando o DSP 96002 da Motorola.

O **apêndice 1** trata dos cálculos e propriedades da DFT e da FFT, sendo também abordadas algumas suas características como as famílias de janelamento, taxa de entrelaçamento do sinal e análise de ruído.

O **apêndice 2** traz os programas feitos para a simulação do sistema e operações timbrais desenvolvidos para o software: MatLab 4.0 (versão para Unix).

Capítulo 2. Definição e representação do som

2.1 A percepção sonora humana

O som representado por $s(t)$ é um sinal temporal. Na sua envoltória podem-se observar os tempos de: *attack*, *decay*, *sustain*, *release*, (ver glossário) como pode ser visto na Fig. 2.1.

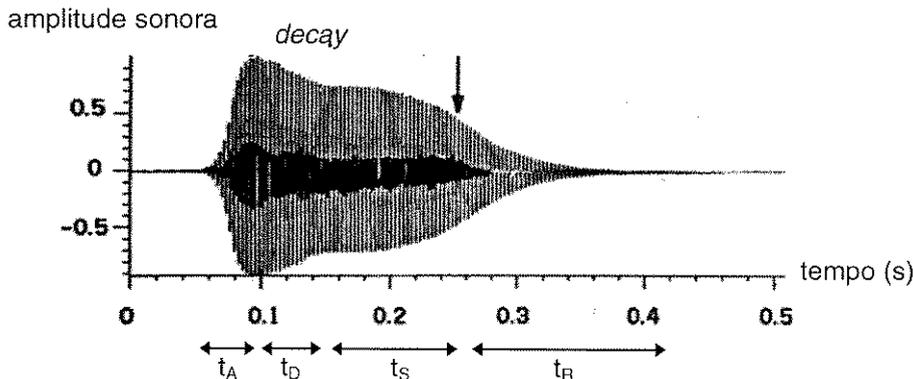


Fig. 2.1 Envoltória de um som de clarinete [22].

O som, como qualquer sinal contínuo e periódico, pode ser representado por um somatório infinita de funções ortogonais [4]. Uma família de funções ortogonais entre si são os senos e cossenos. A transformada de Fourier permite representar funções do tipo $s(t)$ através de funções seno e coseno. Ela pressupõe $s(t)$ como uma função periódica. Se $s(t)$ for uma função não-periódica considera-se $s(t)$ como tendo período infinito.

A definição de transformada de Fourier para uma função $s(t)$ é :

$$S(f) = \int_{-\infty}^{+\infty} s(t) \cdot e^{-j2\pi \cdot f \cdot t} \cdot dt \quad (2.1)$$

$$\text{onde: } e^{-j2\pi \cdot f \cdot t} = \cos(2\pi \cdot f \cdot t) + j \cdot \text{sen}(2\pi \cdot f \cdot t).$$

A transformada inversa de Fourier é dada por:

$$s(t) = \int_{-\infty}^{+\infty} S(f) \cdot e^{j2\pi \cdot f \cdot t} \cdot df \quad (2.2)$$

Dada uma função $s(t)$ contínua e complexa, sua transformada de Fourier, $S(f)$, é uma função contínua e complexa. $S(f)$ é chamada de função do espectro da frequência de $s(t)$. Cada componente complexo $e^{j2\pi \cdot f \cdot t}$ desse espectro é um harmônico de $s(t)$ e possui magnitude e fase.

A informação sonora relevante para a caracterização do timbre está na magnitude de seus harmônicos componentes [27]. Esta é fornecida pela transformada de Fourier de $s(t)$, $S(f)$. Tanto $s(t)$ quanto $S(f)$ são funções reais e desse modo a transformada de Fourier representa $s(t)$ no domínio da frequência.

A percepção auditiva da intensidade sonora

A intensidade sonora é a integral das intensidades das componentes de frequência de um som. No caso da representação de Fourier, é igual à somatória das magnitudes de seus componentes harmônicos.

A percepção humana da intensidade sonora tem uma extensão muito ampla. Para representá-la utiliza-se uma escala logarítmica, em decibels, dB.

A intensidade sonora pode ser dada pela sua densidade de potência em W/m^2 ou pela pressão sonora em Pascals (N/m^2). A escala cuja unidade é Pascals tem por sigla SPL (*sound pressure level*). O nível de referência de intensidade sonora adotado para zero dB é $I = 10^{-12} W/m^2$, correspondente à pressão de $P=20 \mu N/m^2$.

$$[dB] = 10 \cdot \log_{10} \left[\frac{I_1}{I_2} \right] \quad I \text{ em } (W \cdot m^{-2}) \quad (2.3)$$

$$[dB] = 20 \cdot \log_{10} \left[\frac{P_1}{P_2} \right] \quad P \text{ em } (N \cdot m^{-2})$$

A Fig. 2.2 mostra a região audível do espectro de frequência sonora. Observa-se que o ouvido pode perceber sons com densidade de potência entre $10^{-12} W/m^2$ e $10^{-1} W/m^2$ correspondentes aos limiares da percepção (onde começa-se a perceber sons) e o limiar da dor.

A sensibilidade à *variação* da intensidade é inversamente proporcional a intensidade absoluta do som. Assim, quanto mais se aumenta o nível absoluto do som ouvido, mais se diminui a sensibilidade à sua variação de intensidade. Dentro da melhor faixa da percepção (aproximadamente entre 500 e 4.000 Hz.) e em condições de quietude, pode-se perceber variações de intensidade de cerca de ± 1 dB,

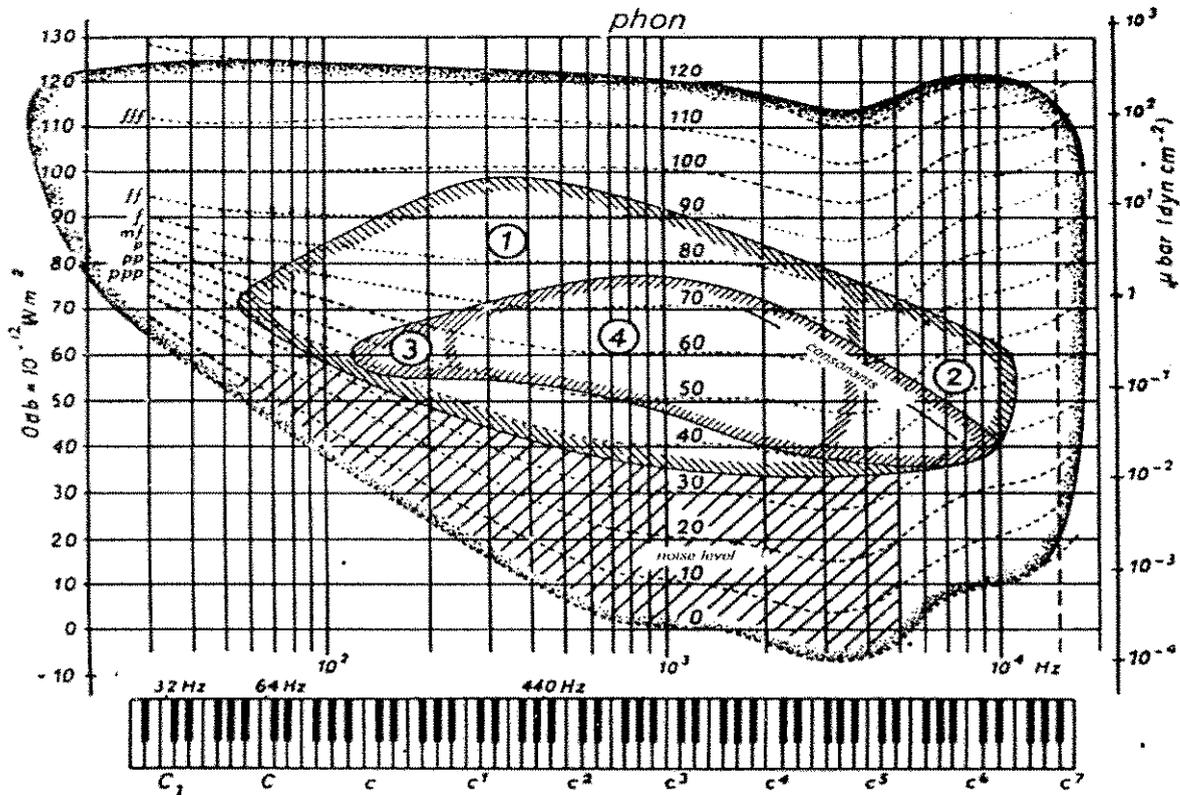


Fig. 2.2 As curvas isofônicas de Fletcher-Munson. (1) instrumentos musicais, (2) harmônicos da voz humana, (3) harmônicas fundamentais da voz humana. (4) componentes sonoras formantes das vogais [32].

correspondente a cerca de 25% da variação da intensidade do som [28]. A percepção da intensidade sonora depende da frequência; uma curva isofônica é aquela para a qual a sensação de intensidade sonora é a mesma. A unidade da percepção auditiva humana (*loudness*) da intensidade sonora é o fon (*phon*) e equivale ao decibel na frequência de 1KHz. As curvas representativas da percepção auditiva humana foram obtidas experimentalmente por Fletcher e Munson [1] e adotados pela *American Standard Association* em 1936..

Observando-se a variação da intensidade das curvas isofônicas em função da frequência, vê-se que a percepção auditiva é bastante sensível na região entre 500 Hz a 4 KHz, tornando-se menos sensível a medida que se desloca para os extremos dos limites da audição.

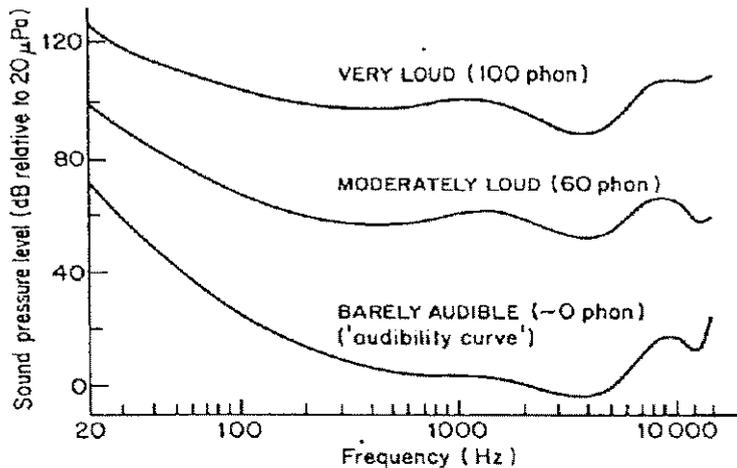


Fig. 2.3 Percepção humana da intensidade sonora. [28]

A percepção auditiva à intensidade sonora, ou *loudness*, é chamada pelos músicos de *intensidade*. Para não criar confusão, utilizaremos o termo intensidade significando sempre intensidade sonora. A loudness possui uma escala subjetiva, indo desde o *pianissimo*, *ppp* (aprox. 30 fons) até o *fortissimo*, *fff* (aprox. 110 fons). Na notação musical a escala completa é denotada pelas siglas: *ppp*, *pp*, *p*, *mp*, *mf*, *f*, *ff*, *fff*. As variações de intensidade são indicadas por *crescendos* e *decrescendos*, as mudanças abruptas por *súbito piano* ou *súbito forte*.

A percepção auditiva da frequência

A percepção de sons numa região relativamente grande do espectro de frequência é característica dos mamíferos. São comumente aceitos como limites da percepção humana em frequência (mínimo e máximo) a região do espectro entre 20 e 20.000 Hz. Um piano produz sons com frequência entre 27,5 Hz (A_0) e 4.186 Hz (C_8) [10], a voz humana entre 80 Hz. (baixos) e 1 KHz (sopranos) [8].

A faixa audível do espectro se reduz com a idade do indivíduo. Entre indivíduos de audição normal, crianças podem escutar até acima de 20 KHz., adolescentes e jovens adultos até 16 KHz. pessoas muito idosas podem ficar com percepção diminuída para 5KHz [28].

O ouvido humano possui uma grande sensibilidade à variação na frequência do som. Foram realizadas medidas da sensibilidade à variação da frequência com público não treinado [8]. A sensibilidade à variação de frequência, $\Delta f/f$ atinge um máximo de 0,3%, aproximadamente 1/20 de semitom, entre 500 e 4.000 Hz. Essa sensibilidade é essencial para o entendimento da fala humana, por este motivo, ela é maior na região do espectro correspondente a melhor percepção à variação da intensidade.

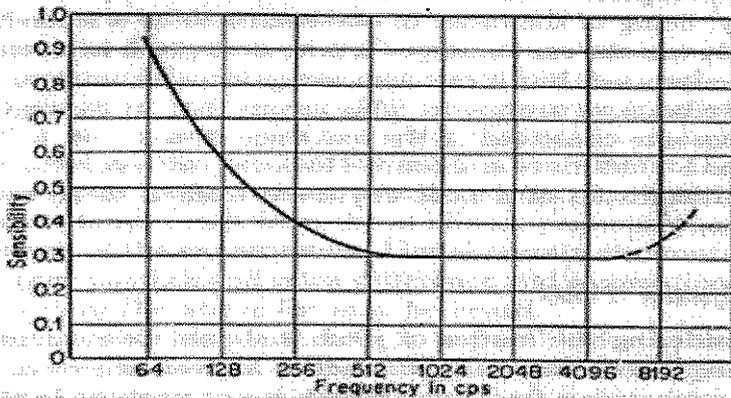


Fig. 2.4 Sensibilidade à variação de frequência $\Delta f / f$, em função da frequência f . [8].

A percepção auditiva da frequência fundamental de um som (o primeiro harmônico de um som) é chamada de *pitch*. A sensibilidade ao *pitch* pode ser aumentada através de treinamento, por isso os músicos costumam perceber variações de frequência mais sutis que o público em geral (por exemplo: um afinador de piano pode perceber até 1/50 de semitom).

Em música o *pitch* é representado pela escala musical. A escala comumente utilizada para instrumentos de teclas (piano, sintetizadores eletrônicos) é a escala temperada cromática. Ela é dividida em 12 semitons, cada semitom equivalendo a um intervalo de frequência de $2^{1/12} \cdot f \cong 1,059463 \cdot f$, ou aproximadamente 6% da frequência f da nota anterior. Cada intervalo é um semitom. O conjunto de 12 semitons equivale ao intervalo em frequência de $2^{12/12} \cdot f = 2 \cdot f$, de uma oitava.

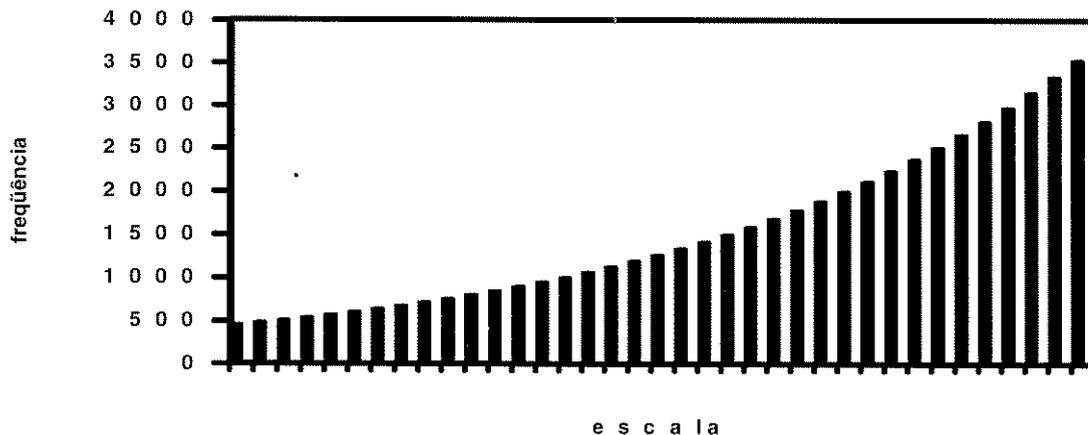


Fig. 2.5 O *pitch* correspondente às notas de três oitavas da escala musical cromática temperada, de A_4 (440 Hz) até A_7 (3520 Hz).

Na região de maior sensibilidade, a audição humana pode discriminar variações de frequência da ordem de 1/20 de semitom, o que corresponderia a uma escala temperada onde cada oitava teria $(12 \cdot 20) = 240$ notas.

Percepção auditiva da duração de um som:

Existe um intervalo de tempo limite abaixo do qual dois ou mais estímulos sonoros são percebidos pela audição humana como um único estímulo. Ele é chamado de **persistência auditiva** e equivale aproximadamente a 50 ms [28]

Considerando-se um som constituído por um trem de impulsos cujo período T (intervalo de tempo entre dois pulsos consecutivos) pode ser variável observa-se que para $T > 50\text{ms}$ escuta-se cada impulso separadamente. Para $T < 50\text{ms}$ passa-se a escutar a frequência correspondente à $f = 1/T$.

A persistência auditiva portanto estabelece a fronteira na qual a audição deixa de perceber intervalos de tempos e passa a perceber frequências. Existe uma relação entre a intensidade da componente do som e seu intervalo de persistência auditiva [8]. No entanto o valor de 50ms é comumente aceito como padrão para qualquer intensidade sonora.

O intervalo da persistência auditiva estabelece a especificação do máximo retardo de processamento sonoro em tempo real. Considera-se a percepção de duração do som linear.

Existe um intervalo de tempo para a interpretação (reconhecimento) de um som pelo cérebro, que é feita pela associação das características do som de entrada com padrões sonoros contidos na memória. Pode-se considerar este atraso de associação, como o intervalo de tempo para o reconhecimento de um timbre [28].

O valor do atraso de associação não é fixo, sabendo-se que varia de acordo com parâmetros comportamentais, como a familiaridade do som ouvido. Pode-se notar esse fenômeno quando se escuta duas vezes um trecho musical desconhecido. Na primeira audição ainda não se conhecem os padrões sonoros e o tempo de associação é grande, fazendo com que outros detalhes da música não sejam percebidos. Na segunda vez, já tendo associado diversos padrões sonoros a percepção é mais rica em detalhes. Na música alguns princípios foram intuitivamente desenvolvidos pelos compositores, visando contornar ou utilizar o atraso de associação.

Timbre

Num sentido genérico, o timbre depende de vários parâmetros do som, incluindo a envoltória do espectro do som e suas mudanças no tempo, as flutuações periódicas da amplitude ou da frequência fundamental ou mesmo se o som é melódico ou percussivo [27]. Considerando um instante de sustentação sonora (*steady-state*) num senso muito estrito, o timbre é: aquele atributo da sensação auditiva pelo qual um indivíduo pode julgar se dois sons complexos e sustentados, com o mesmo *loudness* e *pitch* são diferentes entre si [American Standards Association].

A definição de timbre é subjetiva e permite uma grande quantidade de interpretações. Por outro lado, a sua representação pode ser adequadamente estabelecida.

O timbre pode ser representado como a informação contida na variação temporal da envoltória do seu espectro de frequência, ou seja, dos seus harmônicos componentes. Experimentalmente verificou-se que a fase dos harmônicos do espectro sonoro de um som exerce pouca influência sobre o reconhecimento de um timbre [27]. Deste modo, pode-se representar o timbre apenas pela variação temporal das magnitudes dos harmônicos componentes de um som, ou seja, pela evolução temporal de sua superfície sonora.

O timbre pode ser visto através das abordagens macroscópica e microscópica do som. Na abordagem macroscópica, analisa-se a envoltória temporal do som com os tempos e formas do *attack*, *decay*, *sustain* e *release*. A forma dessa envoltória é necessária para se caracterizar um timbre mas não é suficiente. Na abordagem microscópica analisa-se a evolução temporal do som com mais detalhe, sua forma de onda. Podem-se estabelecer as componentes harmônicas que compõem essa forma de onda, por exemplo, com a transformada de Fourier. A análise harmônica é necessária para a caracterização do timbre mas não é suficiente.

Considerando-se as duas abordagens simultaneamente pode-se analisar a evolução temporal das componentes harmônicas do som, caracterizando seu timbre.

2.2 O processamento digital de áudio

O processamento digital do som deve representar $s(t)$ com precisão suficiente. O padrão de digitalização sonora adotado atualmente para *Compact Disc* é de $n = 16$ bits de resolução e taxa de amostragem $F_s = 44,1\text{ KHz}$ [6].

Antes de amostrar o sinal de entrada $s(t)$, é feito um pré processamento, consistindo da filtragem passa-faixa, para limitar o sinal à região de frequência de interesse e do controle de ganho, para limitar o sinal a uma faixa de amplitudes dentro dos limites máximo e mínimo da resolução do conversor A/D, ou

seja 2^n níveis, sendo n a resolução do conversor A/D. A amostragem do sinal deve ser feita numa taxa de pelo menos 2 vezes a frequência do harmônico mais alto existente no sinal $s(t)$ [9]

O som pode ser representado por uma função $s(t)$, onde o tempo é uma variável independente e contínua. A representação discreta do som é dada por sua amostragem em intervalos uniformes de tempo T .

$$s(nT) = s(t)|_{t=n.T}, \quad n \in \mathbb{N} \quad (2.4)$$

onde: $n = 0, 1, 2, 3, \dots$

A taxa de amostragem é a quantidade de amostras por segundo que sua representação digital possui. Seu valor é dado por $F_s = T^{-1}$ Hz. O sinal amostrado $s(n)$ é obtido pelo produto ou modulação entre o sinal analógico $s(t)$ e um trem de pulsos de período T $m(t)$, como na Fig. 2.6

$$m(t) = \sum_{n=-\infty}^{+\infty} \delta(t - n.T) \quad (2.5)$$

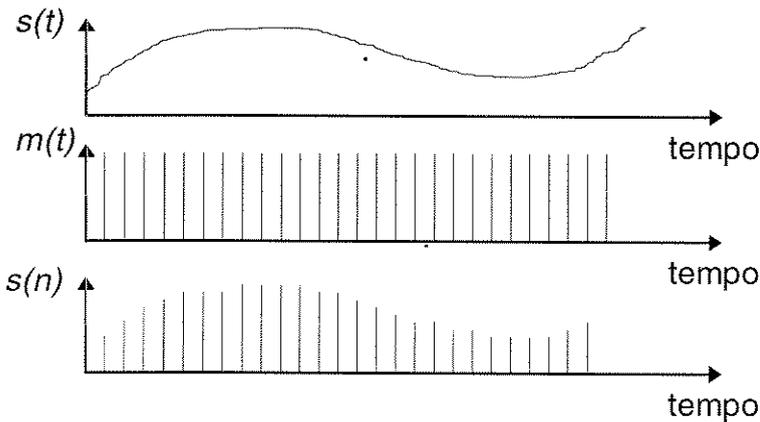


Fig. 2.6 Digitalização de $s(t)$ pela modulação (produto) com a função $m(t)$.

Como $s(t)$ é filtrado antes da amostragem, o seu espectro está limitado a uma faixa de frequências. A função no domínio da frequência, correspondente ao trem de pulsos $m(t)$ é $M(f)$, um trem de pulsos em frequência, espaçados de $F_s = T^{-1}$ Hz, como se pode ver na Fig. 2.7 (a).

A representação de $s(n)$ no domínio da frequência (Fig. 2.7 (b) e (c)) é dada pelo convolução do espectro do sinal analógico de entrada, $A(f) * M(f)$,

Chama-se de *aliasing* à superposição do espectro de $s(n)$, quando $F_s < 2.f_H$, como se pode apreciar na Fig. 2.7 (c), as frequências mais altas se superpõem às mais baixas do espectro, resultando em ruído e distorção.

O processo de conversão analógico-digital possui uma série de condições que devem ser satisfeitas para se obter uma boa amostragem [24].

As mais importantes são:

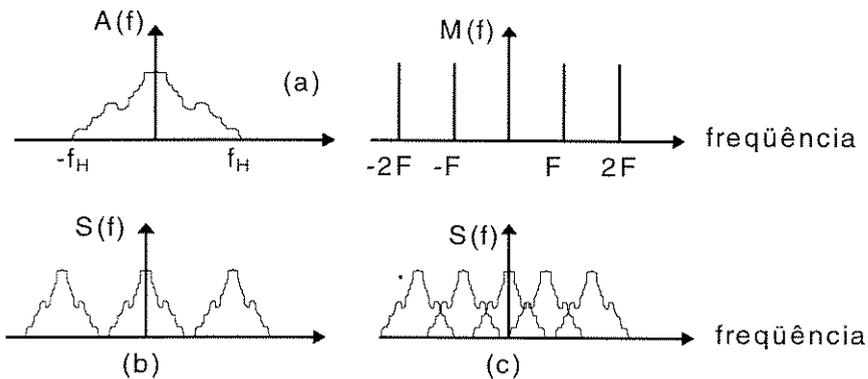


Fig. 2.7 Processo de amostragem de $a(t)$ visto no domínio da frequência. (a) Espectros de: $s(t)$ e de $m(t)$. (b) Espectro de $s(n)$, com $F_s=2.f_H$. (c) Espectro de $s(n)$, com $F_s < 2.f_H$ (*aliasing*).

- Velocidade de conversão:** O processo de conversão A/D atua sobre $s(t)$, limitado pela filtragem à faixa de frequência entre f_L e f_H . Apesar do som poder conter outras frequências, a audição humana só percebe aquelas entre 20 Hz e 20 KHz. Pode-se então limitar o sinal de entrada a esta faixa, sem comprometer a sua qualidade em relação à audição humana. Para se evitar *aliasing*, a taxa de amostragem F_s deve ser maior que $2.f_H$. Portanto, o conversor A/D tem que possuir uma velocidade de conversão tal que sua taxa de amostragem seja $F_s \geq 40000$ amostras/s. O mesmo vale para o conversor D/A, só que a sua velocidade da conversão deve ser igual à taxa de amostragem, ou seja $F_s' = 2.f_H$ amostras/s. Se a taxa de amostragem do conversor D/A for $F_s' \neq F_s$, $s'(t)$ terá seu espectro deslocado em frequência. A taxa de amostragem padrão CD é $F_s = 44,1$ KHz.
- Resolução:** A resolução de um conversor (A/D ou D/A) é dada pelo número de bits da palavra binária utilizada para representar a amplitude do sinal amostrado. Na amostragem linear cada amostra é representada por $n=b+1$ bits. Sendo b bits da representação da amplitude e 1 bit (LSB) usado para o sinal. O número de bits e conseqüentemente, a resolução do conversor, estão relacionados à sua faixa dinâmica em decibéis. Em termos da relação sinal/ruído do sinal amostrado a resolução do conversor está relacionada a amplitude deste sinal em decibéis pela expressão $SQNR=20.\log_{10}2^n$ dB. Assim, um conversor linear de resolução $n=16$ bits permite gerar $2^{16} = 65.536$ níveis (-32.768 a 32.767) de amplitude, suficientes para representar uma região de intensidades de som de 96 dB. Viu-se anteriormente que a audição humana percebe sons (dentro da faixa de frequência de maior sensibilidade) cuja intensidade está aproximadamente entre 0 e 120 dB. Aplicando a expressão dada obtém-se uma resolução $n=20$ bits, ou $2^{20}=1.048.576$ níveis. Apesar de atualmente se utilizar resolução de $n = 16$ bits observa-se que o ideal é uma resolução de 20 bits para a conversão linear.
- Linearidade:** Um conversor A/D ideal espaça igualmente os 2^n níveis das amostras do sinal de entrada. Isto é chamado de *linearidade* do conversor. Na prática, os conversores A/D não espaçam os níveis exatamente. Este erro é chamado de *erro diferencial de linearidade* [6]. Se o erro diferencial exceder ao valor de um nível inteiro, a amplitude amostrada subsequente passa a apresentar um desvio de um nível sobre a amplitude do sinal original, o que implica em ruído. Outro erro de linearidade é o *erro integral*. Ao contrário do erro diferencial, este é causado pelo conversor D/A. A saída do conversor apresenta um desvio de amplitude em relação ao sinal amostrado. Isto acontece em conseqüência da não-linearidade da amplificação interna do conversor D/A para gerar o sinal de saída. Normalmente este erro é desprezível se comparado ao erro diferencial. Para a medição da linearidade do processo de conversão, divide-se a amplitude do sinal de entrada em $N-1$ intervalos iguais, onde N é o número total de níveis. Calculam-se os níveis máximo e mínimo da saída do conversor e comparam-se estes valores com os níveis máximo e mínimo da entrada. Não-linearidades abaixo de metade do valor representado pelo bit menos significativo (LSB) da resolução do conversor são consideradas toleráveis [6].
- Precisão:** A precisão se baseia nos extremos da amplitude do sinal representado pelo conversor. Para um conversor A/D com resolução de n bits e entrada entre $-V$ e $+V$, os níveis de representação

estarão entre: $-V[(1 - 2^{(1-n)})]$ e $+V.(1 - 2^{(1-n)})]$, correspondentes à precisão do conversor. Nota-se que quanto maior a resolução do conversor melhor sua precisão. A precisão de um conversor também está relacionada ao erro de offset (surgimento de uma componente contínua no sinal de saída do conversor) e ao erro de controle de ganho (desvios nos níveis máximo e mínimo apresentados no sinal de saída do conversor). No entanto esses erros são solucionáveis e não comprometem a fidelidade sonora.

- **Tempo de ajuste:** Quando a entrada digital de um conversor D/A muda de nível, a sua saída leva um certo tempo para responder à essa mudança, o tempo de ajuste (*settling time*). Os valores típicos estão abaixo de 30 μ s para grandes variações da saída (10V a +10V) e entre 2 a 5 μ s para variações em torno de 0,1V. No entanto os conversores D/A geram ruídos nas mudanças abruptas de nível do sinal amostrado. Esse ruído é chamado de *glitch*, e pode ser reduzido através de um filtro passa-baixas na saída do conversor D/A.

O processo de conversão analógico-digital apresenta dois tipos principais de ruído [9], o ruído de quantização e o ruído de saturação, descritos abaixo.

- **Ruído de quantização da conversão A/D:** É o resultado da representação da amplitude do sinal contínuo por um número fixo de níveis digitais, com valores: mínimo e máximo pré-definidos. Este tipo de ruído aparece como um sinal (ruído) adicionado ao sinal analógico de saída, do tipo: $[x(nT)]_Q = x(nT) + e(nT)$. O erro $e(nT)$ é chamado de erro de arredondamento da amplitude do sinal analógico de entrada, que está representado por 2^n níveis. O erro de quantização aparece na forma de um ruído branco, decorrelacionado ao sinal analógico de entrada. A relação sinal/ruído de quantização $SQNR=20.\log_{10}2^n$ expressa em decibéis quanto a intensidade do ruído de quantização está abaixo da intensidade do sinal de entrada.
- **Ruído de saturação da conversão A/D:** É o ruído gerado quando a entrada ultrapassa os níveis máximo ou mínimo do conversor A/D. O ruído de saturação depende: da resolução do conversor e do nível RMS de entrada do sinal. O controle de ganho é fundamental para evitar esse tipo de ruído.

O espectro da frequência de $s(n)$, é obtido pela transformada discreta de Fourier uma variação da transformada de Fourier para funções discretas. A seguir descrevemos seu algoritmo..

O som amostrado é dado por $s(n)$, onde $n = 1, 2, \dots, N$ pontos reais. Estes pontos são obtidos a uma taxa de amostragem F_s . A representação de $s(n)$ no domínio da frequência é dada por $S(k)$, onde $k=1, 2, \dots, N$ pontos complexos. A transformada discreta de Fourier, ou DFT, é definida por:

$$S(k) = \frac{1}{N} \sum_{n=0}^{N-1} s(n).W_N^{nk} \quad (2.6)$$

A sua inversa, IDFT, por:

$$s(n) = \sum_{k=0}^{N-1} S(k).W_N^{-nk} \quad (2.7)$$

$$\text{onde : } W_N^{nk} = e^{-j.2.\pi.n.k/N} = \cos(2.\pi.n.k/N) + j.\text{sen}(2.\pi.n.k/N).$$

A diferença entre a DFT e a sua inversa, a IDFT, é apenas no fator de escala (1/N) e no sinal do expoente de W_N^{nk} . Tanto $s(n)$ quanto $S(k)$ podem assumir valores complexos. Pode-se representar a entrada e a saída da DFT como vetores complexos, de dimensão N. Se o vetor de entrada for real, como é o caso do amostragem sonora, a saída da DFT é simétrica, em relação ao centro do vetor. Os primeiros N/2 pontos complexos do vetor de saída se repetem, ordenados inversamente, nos N/2 pontos seguintes. Estes N/2 pontos representam os harmônicos do vetores de som da entrada, de N pontos reais. Como usamos apenas a magnitude dos harmônicos calcula-se a magnitude dos N/2 pontos do vetor de saída. Em suma: para um vetor de entrada de N pontos reais, a saída da DFT é um vetor com N/2 pontos reais.

O cálculo do espectro de frequência pela DFT, possui uma série de características que influenciam nos seus resultados. São elas: a seletividade de frequência (*frequency selectivity*), a distribuição espectral (*spectral leakage*) e as perdas em escalonamento (*scalloping loss*) [9]. A maioria dessas características é inerente do cálculo da DFT e são consequência do espectro de saída da DFT se apresentar discretizado no eixo das frequências na forma de um vetor complexo de N pontos.

O espectro discreto tem os seus pontos espaçados em intervalos de $f_k = k.Fs/N$, onde $k = 1, 2, \dots, N$. O passo de frequência é chamado de *bin* = Fs/N , e estabelece a chamada *frequency selectivity*. Um harmônico componente de $s(n)$, contido no vetor de entrada, que tiver uma frequência $f \neq f_k$ gerará na saída da DFT não um único harmônico, mas uma série de harmônicos de menor magnitude que o harmônico original e com frequências f_k vizinhas à frequência original deste harmônico, criando a *spectral leakage* do sinal. Na prática ela sempre ocorre pois as frequências dos harmônicos nunca são exatamente iguais às frequências f_k . Os harmônicos criados pelo *spectral leakage* tem menor magnitude do que o harmônico original. Este efeito é chamado de *scalloping loss*, e representa perda na energia do espectro representado pela DFT. Nota-se que quanto maior for o número de frequências f_k , ou seja, o número de pontos N do vetor, menor serão as perdas da saída da DFT.

Por outro lado, do mesmo modo que a transformada contínua de Fourier, a DFT também apresenta um compromisso entre o número de amostras da entrada e a resolução em frequência. Desejamos observar o espectro instantâneo do som amostrado. Quando o vetor de entrada da DFT tiver um grande número N de amostras o espectro será representado por N/2 pontos com boa resolução. No entanto, o tempo correspondente a essa amostragem pode ser maior do que o máximo que podemos usar. Do mesmo modo, se for utilizado um pequeno N, a resolução do espectro poderá ser insuficiente. Este fato é discutido adiante.

Uma técnica usada para minimizar as perdas características da saída da DFT é o janelamento da função de entrada. Ela basicamente consiste em multiplicar o vetor de entrada por um outro vetor cujos elementos extremos tendem a zero. Uma função janela, $a(n)$, é aplicada à entrada do seguinte modo:

$$S_j(k) = \sum_{n=0}^{N-1} j(n) \cdot s(n) \cdot W_N^{nk} \quad (2.6)$$

As funções de janelamento são usadas especialmente para evitar o ruído causado pelo *spectral leakage*. A função de janelamento força os extremos da seqüência, contida no vetor de N pontos de entrada a assumir valores próximos de zero. Assim a DFT sempre interpreta os N pontos de entrada como a representação de um sinal periódico. O janelamento representa uma convolução das seqüências $s(k)$ e $j(k)$, no tempo, ou seja, uma filtragem de $s(t)$ por $j(k)$. Existe uma série de famílias de funções de janelamento que são utilizadas de acordo com as especificações e necessidades de saída do sinal.

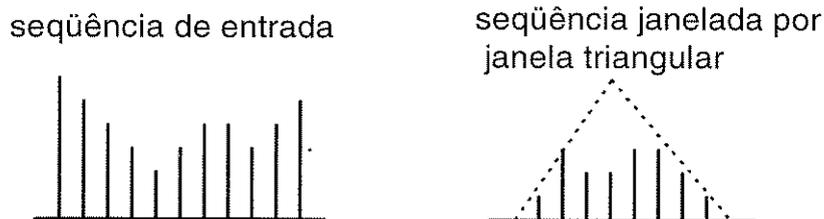


Fig. 2.8 Janelamento de uma seqüência de N pontos

O janelamento implica na perda de informação do sinal nos pontos extremos da seqüência de N pontos de entrada. Isto pode ser contornado através da técnica de entrelaçamento (*overlapping*) da seqüência de vetores de saída da DFT. As seqüências de N pontos de entrada janelada da DFT podem ser entrelaçadas de modo a minimizar a perda nos extremos do janelamento. O entrelaçamento pode ser feito em qualquer taxa de porcentagem, embora seja necessário pelo menos 50% de entrelaçamento para minimizar as perdas causadas no janelamento(ver Fig. 2.9)

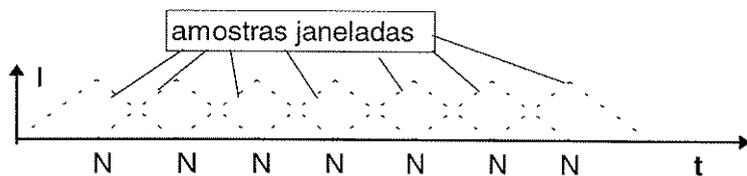


Fig. 2.9 Entrelaçamento das seqüências de entrada, de 50%.

A utilização do janelamento e entrelaçamento do sinal implica redundância de DFTs, pois os pontos do vetor de entrada da DFT são parcialmente repetidos no cálculo da DFT seguinte. Por outro lado, este constitui um método eficiente de se evitar os ruídos causados na reconstituição do sinal analógico, ou seja, no cálculo da IDFT. Neste estágio, o vetor de entrada, que contém a representação de um espectro e transformado em um vetor que representa um trecho (intervalo de tempo) do sinal amostrado no tempo. Na sucessão de cálculos da IDFT vê-se que os vetores de saída, postos lado a lado possuem algumas vezes grandes descontinuidades, que constituem ruído do sinal. A maneira de se evitá-lo é através da técnica de janelamento e entrelaçamento dos vetores de saída da IDFT.

Observando a somatória que constitui o cálculo da DFT para N pontos de entrada, vê-se que ela necessita de N^2 operações complexas para calcular o vetor de saída, de N pontos. No caso de se utilizar grandes valores de N os cálculos da DFT pode se tornar imensos, inviabilizando o processamento da próxima DFT em tempo hábil. Existe no entanto uma série de algoritmos desenvolvidos para se calcular a DFT, que demandam uma quantidade de cálculos bem menor. Estes são chamados de algoritmos rápidos da DFT, ou transformada rápida de Fourier, conhecidos como FFT. Estes algoritmos reduzem o número de operações complexas para o cálculo da DFT, da ordem de N^2 para $N \cdot \log_2 N$. Isso é feito através de duas propriedades da DFT, a simetria e a periodicidade.

- 1) simetria : $W_N^{k(N-n)} = (W_N^{kn})^*$
 2) periodicidade : $W_N^{k(N+n)} = W_N^{n(N+k)}$

A FFT se baseia na decomposição da DFT original de N pontos em duas DFTs de N/2 pontos. Cada uma dessas novas DFTs é novamente decomposta em outras duas, de N/4 pontos e assim sucessivamente. Portanto é necessário que o vetor de entrada tenha um número de pontos tal que $N = 2^v$. Neste caso, tem-se $v = \log_2 N$ possíveis fragmentações do vetor de entrada original, de N pontos. O algoritmo básico do cálculo da FFT chama-se *butterfly* e é dado por:

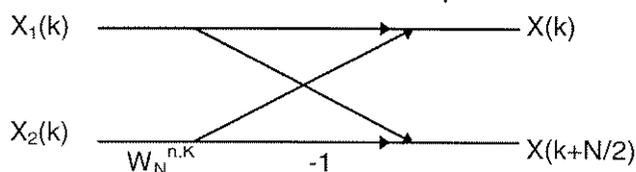


Fig. 2.10 Diagrama esquemático do algoritmo *butterfly* para cálculo da FFT [9].

A fragmentação do vetor de N pontos pode tanto ser feita no sentido da entrada para a saída, conhecida como decimação no tempo (*decimation-in-time*), quanto no sentido inverso ou decimação na frequência (*decimation-in-frequency*). Ambos são similares, diferindo apenas no sentido em que são realizados os estágios do cálculo da FFT. Nota-se que o número de cálculos requeridos para a FFT cresce proporcionalmente ao logaritmo base 2 do número de pontos N. Disto conclui-se que quanto maior o valor de N, proporcionalmente menor será o crescimento do número de cálculos. Para efeito de comparação, tem-se o número de cálculos requeridos na DFT e na FFT, para uma dada taxa de amostragem do vetor de entrada, ou seja, um vetor de entrada a cada N/F_s segundos:

$$\begin{array}{ll} \text{DFT: MCPS} = N \cdot F_s & \text{(multiplicações complexas por segundo)} \\ \text{DFT: ACPS} = (N - 1) \cdot F_s & \text{(adições complexas por segundo)} \end{array}$$

DFT: OCPS = $(2.N - 1).F_s$ (operações complexas por segundo)

FFT: MCPS = $(F_s/2).log_2N$

FFT: ACPS = $F_s.log_2N$

FFT: OCPS = $(3/2).F_s.log_2N$

A FFT realiza $5.F_s.log_2N$ cálculos reais por segundo. Sabendo-se que $N=2^v$ o número de cálculos reais da FFT fica sendo $5.F_s.v$, ou seja, a cada vez que o número de pontos N dobra de tamanho, o aumento do número de cálculos é de apenas $5.F_s$. Uma visão mais detalhada das definições e propriedades da DFT e da FFT é dada no apêndice 1.

Capítulo 3. Anteprojeto do sistema

3.1 A superfície sonora (SS)

O som pode ser representado nos domínios do tempo e da frequência. No domínio do tempo, além dos valores instantâneos da sua forma de onda, pode-se observar a envoltória, que indica a evolução temporal da energia média propagada e que é caracterizada pelos tempos de *attack*, *decay*, *sustain* e *release*, como foi visto na Fig 2.1 do capítulo anterior. No domínio da frequência pode-se observar a envoltória do seu espectro de frequência, contendo a informação da magnitude de seus componentes.

Essas informações são importantes para a representação do timbre. Cada uma delas evidencia um aspecto do som. A superfície sonora, SS, permite representar a evolução temporal das magnitudes das componentes de frequência de um som.

O conjunto dos componentes de frequência num mesmo instante de tempo forma o espectro instantâneo de um som. Em princípio, esse espectro é contínuo (formado por infinitos harmônicos) tornando a SS contínua.

A seção da SS por um plano perpendicular ao eixo do tempo, ou plano harmônico, mostra o espectro de frequência daquele instante. A seção da SS por um plano perpendicular ao eixo da frequência ou plano morfológico mostra a evolução temporal daquela frequência, conforme é visto na figura 3.1:

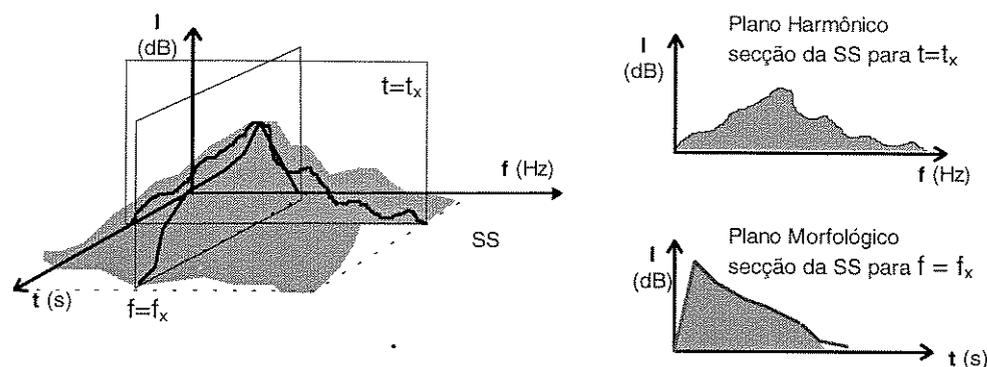


Fig. 3.1 A representação da superfície sonora SS e dos planos harmônico e morfológico.

A cada instante o som possui um espectro de frequência. Cada componente do espectro possui uma magnitude e uma fase. Foi visto que na percepção do timbre a fase dos harmônicos exerce pouca influência. Portanto a SS pode ser representada apenas pelas magnitudes dos harmônicos dentro dos limites da percepção auditiva humana. Conforme é observado na figura 3.2, estes limites dimensionam um volume máximo da SS dentro do qual a magnitude dos harmônicos do espectro do som devem ser representados, ou seja; amplitude entre 0 dB e 120 dB, frequência entre 20 Hz e 20 KHz, intervalo de tempo abaixo de 50ms.

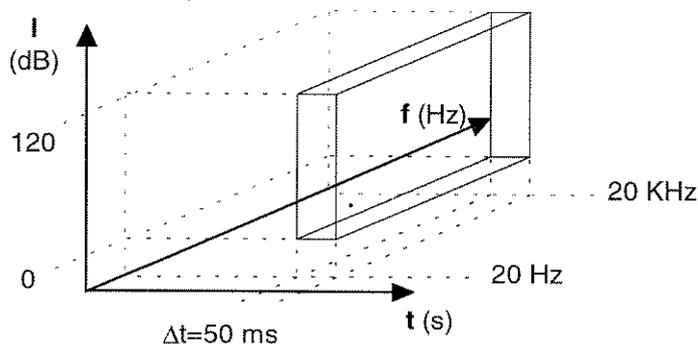


Fig. 3.2 Volume limite máximo da percepção auditiva humana.

Dentro desses limites é possível utilizar procedimentos aproximados de análise e de síntese, plenamente satisfatórios para o ouvinte humano.

3.2 A matriz harmônica (MH)

Como foi visto no capítulo anterior, o algoritmo da FFT mapeia um vetor de N pontos complexos no domínio do tempo para um novo vetor complexo de N pontos, no domínio da frequência. Se o vetor de entrada do algoritmo da FFT for real, a sua saída será simétrica, ou seja, a cada N amostras obtém-se um espectro com $N/2$. Cada ponto do espectro representa uma faixa de frequência $bin=Fs/N$.

Cada espectro com $N/2$ pontos da FFT corresponde a uma linha da MH, ou **vetor harmônico**. Cada coluna da MH corresponde à envoltória temporal de um harmônico de $s(n)$, chamada de **vetor morfológico**. Os vetores harmônicos e morfológicos correspondem aos planos harmônicos e morfológicos da SS.

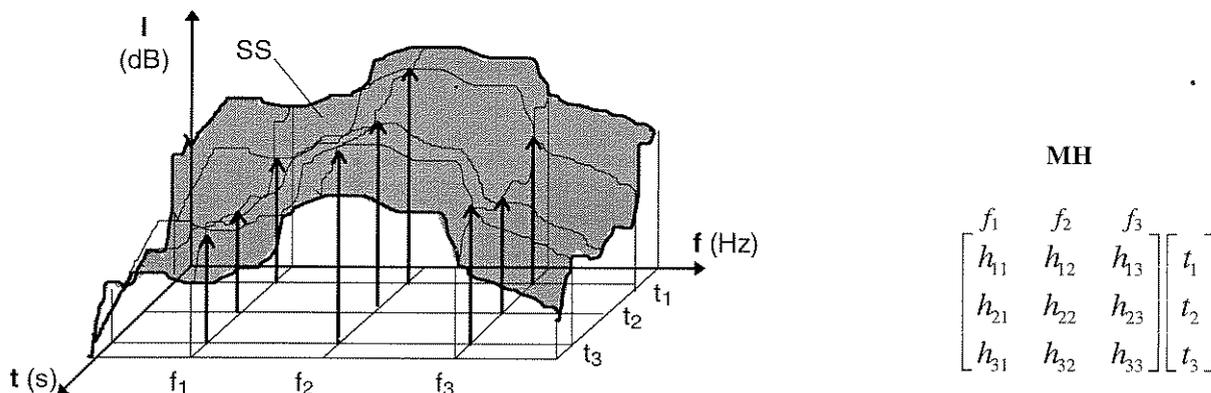


Fig. 3.3 A representação da SS discreta pela MH. A linha $[h_{11} \ h_{12} \ h_{13}]$ é o vetor harmônico para o instante t_1 . A coluna $[h_{11} \ h_{21} \ h_{31}]$ é o vetor morfológico para o harmônico de frequência f_1 .

Os elementos da MH estão espaçados em frequência por intervalos $bin=Fs/N$ e no tempo por $jt=N/Fs$. O valor numérico desses elementos é representado digitalmente com $m=2^n$ níveis discretos. Isto corresponde a uma variação não igualmente espaçada em dB, de $20 \cdot \log(m_{i+1}/m_i)$, ficando 6,02 dB entre o primeiro (0 dB) e o segundo nível, 3,52 dB entre o segundo e o terceiro, 2,49 dB entre o terceiro e o quarto, etc...

Os elementos dos vetores harmônicos devem estar espaçados em frequência por intervalos menores ou iguais às variações da frequência perceptíveis pela audição humana; adotamos o valor referente à menor variação perceptível na faixa de melhor audibilidade (500 Hz - 4 KHz), obtendo-se o valor de $bin = 0,3\%$ de 500 Hz, ou 1,5 Hz.

O espaçamento entre os níveis de intensidade deveria ser suficiente para representar a menor variação perceptível, ou seja, ± 1 dB [28]. Com a especificação usada para a resolução, este espaçamento máximo é conseguida a partir do 10^9 nível, de 20 dB onde a variação entre o nível 9 e 10 é de 0,91 dB.

Esses espaçamentos uniformes criam um limite abaixo do qual a especificação não é atendida e acima do qual é excessiva. Um modo de evitar isto é adotar espaçamentos não-lineares na representação da MH.

3.3 O operador timbral (OT)

O operador timbral (OT), gera o timbre de saída através de operações sobre os elementos dos vetores harmônicos de entrada. Cada vetor harmônico é transformado por uma ou mais operações vetoriais que compõem a operação timbral.

O sistema de transformação sonora trabalha em tempo real e o retardo entre a entrada e saída deve estar abaixo da persistência auditiva. Assim o operador timbral deve realizar a sua função dentro de um intervalo de tempo tal que, somado aos demais intervalos de processamento do sistema seja inferior a 50ms.

Cada operação timbral é parametrizada em tempo-real. Ela pode ser combinacional, controlada pelo vetor do espectro do som de entrada, pode ser seqüencial, controlada pelos elementos da MH, ou ambas.

O operador timbral executa as operações timbrais usando três entradas de dados:

- o vetor do espectro do som de entrada,
- a MH e
- o controle externo.

A primeira fornece o espectro do sinal de entrada, controlador do sistema, a segunda fornece informações sobre o passado das transformações sonoras realizadas, a terceira fornece as instruções de operação timbral e seus parâmetros. A Fig. 3.4 mostra as três entradas de dados no OT.

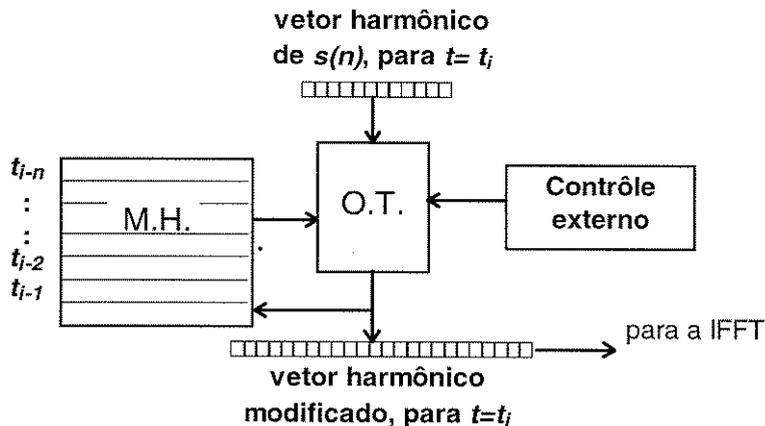


Fig. 3-4 Estrutura esquemática do operador timbral, OT.

Um ciclo do OT equivale à compilação e execução das operações timbrais. Na figura abaixo é exemplificado o processo de um ciclo de operação do OT.

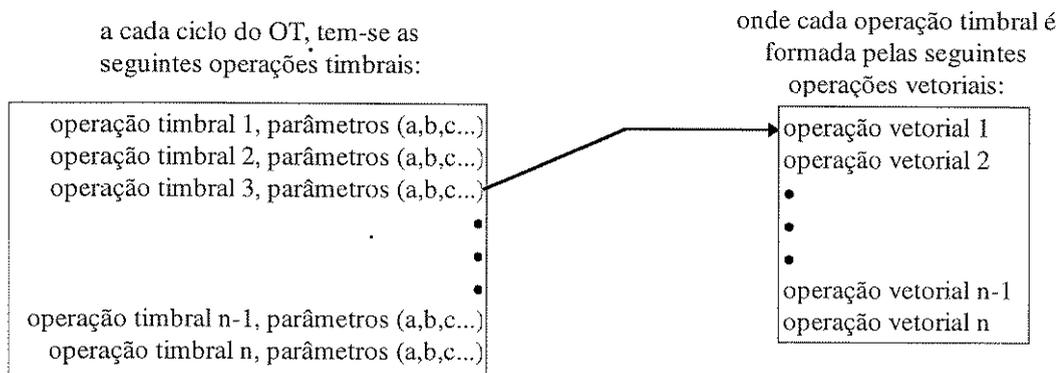


Fig. 3-5 Ciclo de operação do OT.

Uma possibilidade de implementação das operações vetoriais que compõem as operações timbrais é através de circuitos (hardware) para diminuir o tempo de processamento do ciclo do OT. Um exemplo é mostrado na Fig. 3.6 Observa-se que todas as operações timbrais possíveis são feitas unicamente por três operadores vetoriais: produto, deslocamento e soma.

A parametrização das operações timbrais realizadas por estes operadores vetoriais é feita pelo controle externo com o uso de vetores auxiliares.

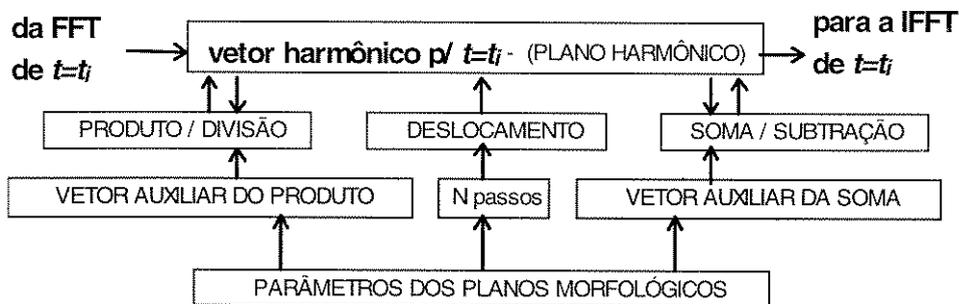


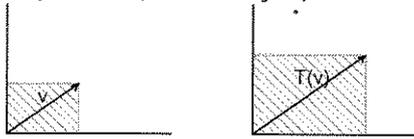
Fig. 3-6 Exemplo de estrutura em hardware de um operador timbral

Uma outra possibilidade de implementação das operações timbrais é através da programação das operações vetoriais em processadores digitais de sinais (DSP), na forma de instruções parametrizáveis. Existem DSPs suficientemente rápidos para a realização das operações timbrais em um ciclo de OT. Pode-se organizar um repertório de operações vetoriais que interpretem uma grande quantidade de operações timbrais. A sua vantagem sobre a implementação em hardware é a flexibilidade de programação das operações vetoriais.

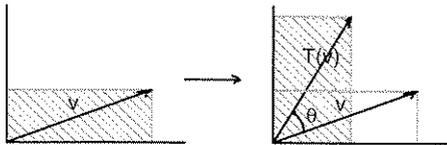
3.4 O sistema de transformação sonora

As transformações são operações matemáticas que realizam o mapeamento de conjuntos de elementos, os vetores, de um espaço vetorial R^n em outro espaço vetorial R^m . Elas podem ser lineares e não-lineares. Uma transformação linear satisfaz a condição $F(u+v) = F(u) + F(v)$, e $F(k.v) = k.F(v)$ [4]. As transformações em R^2 tratam de transformações entre vetores contidos em um plano. No caso das transformações sonoras lidamos com transformações no plano harmônico. Alguns exemplos de transformações no plano harmônico são:

Expansão (ou contração) uniforme: $T(x,y) = \alpha.(x,y)$:



Rotação de um ângulo θ : $T(x,y) = [(x.\cos\theta - y.\sen\theta), (x.\sen\theta + y.\cos\theta)]$:



Transformações destes tipos são feitas em sistemas de processamento digital por transformações nos elementos de vetores que representam um plano amostrado. O sistema aqui tratado manipula os vetores harmônicos a cada ciclo do OT.

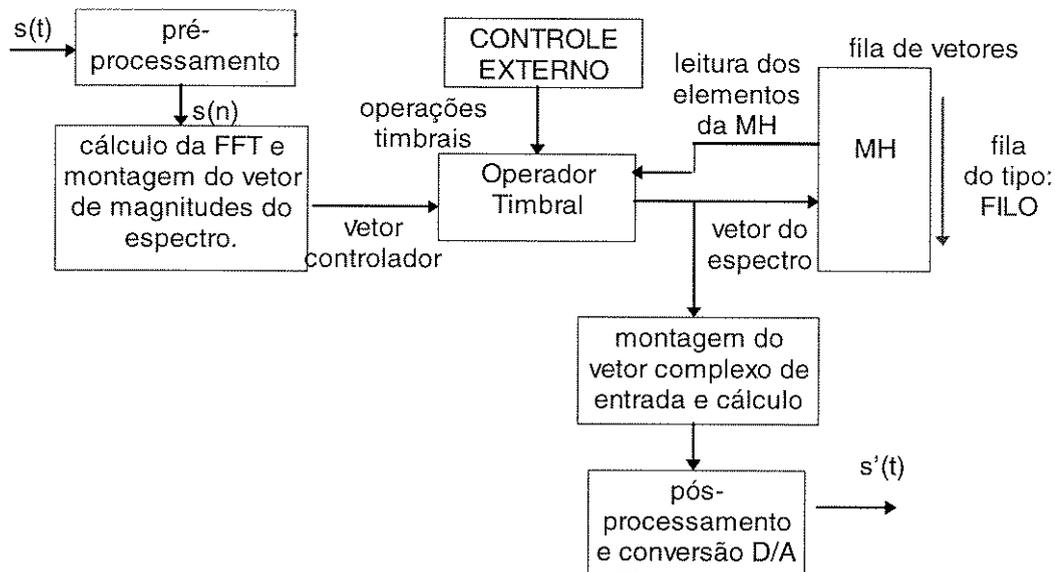


Fig. 3-7 Estrutura esquemática do sistema de transformação sonora.

O sistema de transformações sonoras completo necessita realizar outros processamentos que demandam grande quantidade de cálculos. A estrutura geral deste sistema de transformação sonora é mostrada na Fig. 3.7 e suas partes são descritas a seguir.

- **Pré-processamento do sinal de entrada e conversão A/D:** Chamou-se de $s(t)$ a função no tempo do sinal de audio de entrada. Este deve inicialmente ser limitado na frequência e na amplitude, em regiões adequadas aos limites da percepção auditiva dessas grandezas. A seguir o som é discretizado no tempo e na amplitude, pelo conversor A/D. A discretização no tempo é feita numa dada taxa de amostragem, F_s , e a discretização na amplitude, pela resolução do conversor A/D. Tem-se o sinal amostrado $s(n)$. Os pontos de $s(n)$ são agrupados em vetores de N pontos, para o cálculo da FFT
- **Cálculo da magnitude do espectro:** Executa a FFT sobre N amostras e calcula as magnitudes dos $N/2$ elementos de saída, gerando o vetor controlador “ r ” do operador timbral.
- **O operador timbral:** Executa as operações timbrais usando como entrada: r , MH e o controle externo a cada ciclo de operação do OT.

Cálculo do vetor de som de saída: Monta o vetor de entrada para a IFFT e a executa. O algoritmo da IFFT é o mesmo da FFT apenas acrescentado da multiplicação de todos os elementos da saída deste pelo valor de N , e do deslocamento de seus elementos de $N/2$. A saída deste módulo é o vetor do som transformado, real, pois sua entrada é simétrica

- **Pós-processamento e conversão D/A:** O vetor do som de saída é janelado e entrelaçado para evitar descontinuidades na envoltória de $s'(t)$. A taxa utilizada no entrelaçamento é 50%, pois se trata do ponto ótimo entre o melhor resultado com a menor redundância de cálculos. Em seguida o vetor é convertido em sinal de audio pelo conversor D/A e tratado por um filtro passa baixas de 20 Khz.

O sistema foi dividido em três partes: entrada, operação do timbre e saída. Cada uma dessas partes é formada pelos módulos descritos, e cada módulo por fases, como pode ser visto na figura 3.8.

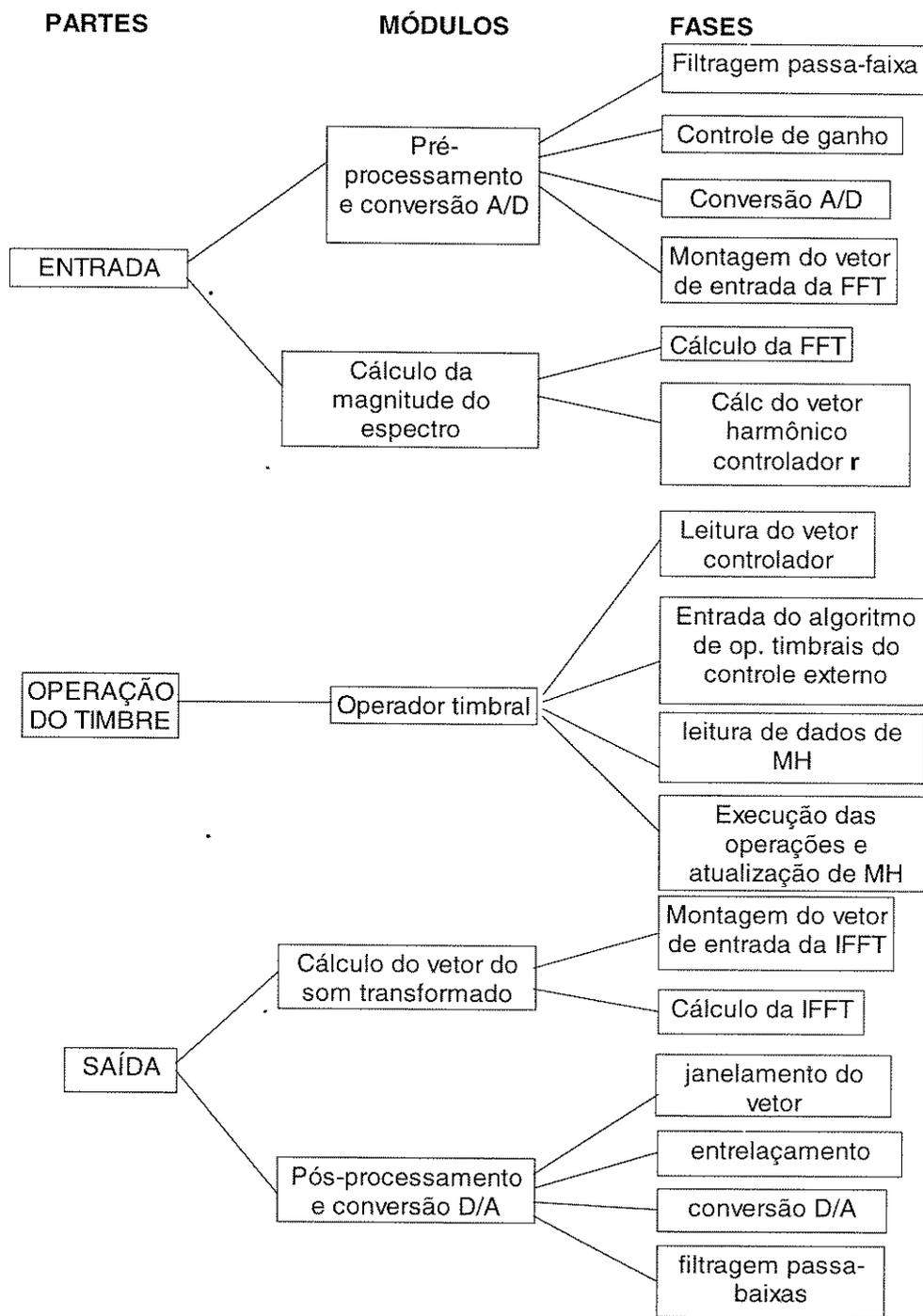


Fig. 3-8 Organização do sistema de transformação sonora.

Capítulo 4. Simulação do sistema em software

O ambiente de simulação do MATLAB 4.0 foi utilizado para o desenvolvimento de programas de simulação das transformações sonoras. Uma vez que se trata de um software, a simulação das operações timbrais no MATLAB não é em tempo-real. Nele foram desenvolvidos dois conjuntos de programas. O primeiro conjunto, chamado de MH, simula o pré-processamento do som e a montagem de sua MH. O segundo conjunto, chamado de OT, simula as operações timbrais sobre uma dada MH.

4.1 A simulação da montagem da MH

Foram elaborados alguns programas para simularem, desde a entrada $s(t)$ até a montagem da SS, representada pela MH. Estes programas permitem também a simulação do pré-processamento do som analógico, ou seja: sua conversão A/D, para um determinado valor de taxa de amostragem, o seu janelamento, com um determinado percentual de entrelaçamento das janelas, e o número de pontos de entrada e saída no algoritmo da FFT.

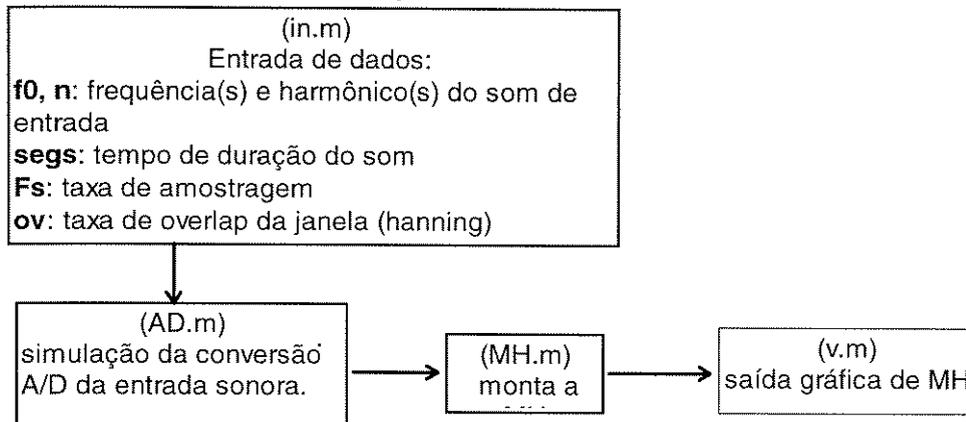


Fig. 4.1 Simulação da montagem da MH.

O sinal de entrada é dado por uma somatória de funções senoidais discretas. Pode-se entrar com uma série de valores de frequências fundamentais. Sobre elas, pode-se escolher quantos harmônicos naturais desejar. Entende-se por harmônicos naturais, a série infinita e decrescente em magnitude de harmônicos, sempre com frequência multipla de 3/2 da frequência do harmônico antecessor. O resultado desse somatório é dado num vetor com $N = F_s \cdot (\text{duração do som})$ pontos. Assim, simula-se o processo de conversão A/D para um som analógico, para um dado valor de F_s . Depois disso, o vetor é janelado em grupos de N pontos, por uma função janela específica. De todas as famílias de janelas do matlab, usou-se a *Hanning*, que é uma sequência do tipo: $w[n] = 0,5 \cdot (1 - \cos(2 \cdot \pi \cdot n / (N + 1)))$, para $n = 1, \dots, N$. Usou-se esta janela porque demonstrou ser a mais adequada nos testes elaborados. As janelas estão entrelaçadas entre si, numa dada taxa. Isto é conhecido como *overlapping*, que implica num processamento redundante do sinal. O entrelaçamento demonstrou melhor eficiência em 50%. Cada janela é então colocada na entrada da FFT, que apresenta como saída N_{fft} pontos complexos. É calculada a magnitude de cada ponto complexo, de onde se obtém a magnitude do espectro de um instante (1/bin s) do som. Esse calculo é repetido para todas as janelas que compõem o vetor do sinal de entrada, em ordem sequencial. Cada vetor de N_{fft} pontos, com o valor da magnitude dos pontos de saída da FFT, é uma linha de MH. Desse modo, a MH é calculada.

4.2 A simulação das operações timbrais

As operações timbrais foram simuladas por esse conjunto de programas. Os valores da taxa de amostragem ($F_s=8.192$ Hz.), do número de pontos de entrada e saída da FFT (256 pontos) e conseqüentemente o valor do bin $F_s/N = 32$ Hz. são fixos. Desse modo, cada espectro representa a magnitude dos harmônicos em $N/2 = 128$ pontos que equivalem a $1/\text{bin} = 31\text{ms}$ de som.

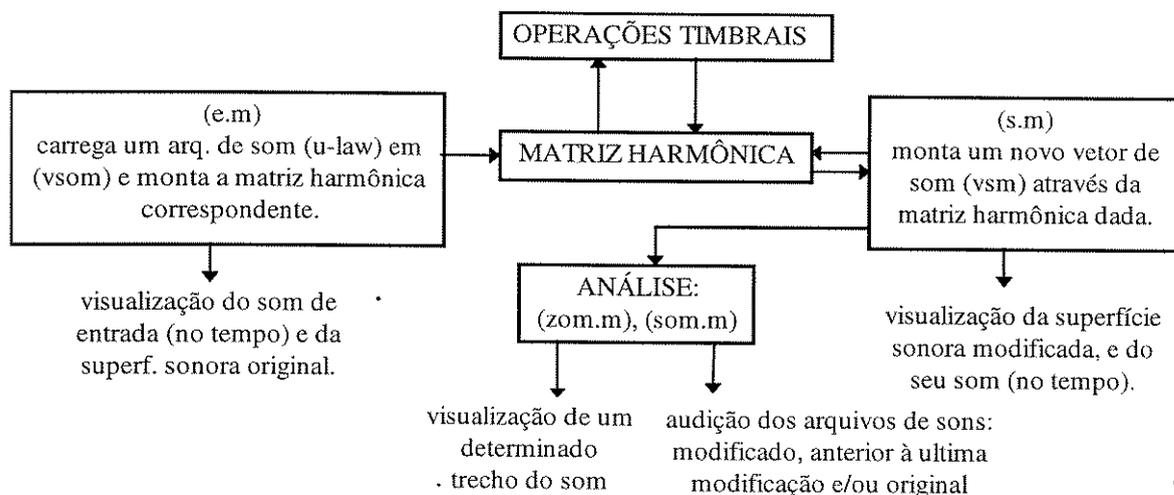


Fig. 4.2 Simulação das operações timbrais.

No programa de inicialização e.m, um trecho de som amostrado no padrão *u-law* ($F_s=8192\text{Hz}$, resolução de 8 bits) é escolhido e armazenado na forma de um vetor de $F_s \cdot \text{segs}$ pontos, onde segs é a entrada da duração do trecho de som analisado, em segundos. O vetor de som é então particionado e janelado, de 256 em 256 pontos, onde $256/F_s=31\text{ms}$ de som. Foram escolhidas janelas de 256 pontos por que cada uma delas deve representar um intervalo de som $< 50\text{ms}$. Conforme o valor escolhido da taxa de *overlap*, cada janela é composta com um grau de $ov \cdot 256$ pontos de redundância, com as janelas vizinhas, onde *ov* está entre 0 e 1. Para cada janela é calculado a FFT resultando 256 valores complexos e simétricos, em relação ao ponto central do vetor de saída. Dessa saída preservam-se os primeiros 128 pontos. Cada vetor harmônico de 128 pontos reais, representando a magnitude do espectro do som, forma uma linha da MH, e representa um instante de tempo da SS. Assim tem-se a formação de duas matrizes: a primeira é a matriz complexa MH que representa o som e será transformada nas operações timbrais, a segunda é MH que representa a evolução temporal da dos valores harmônicos do som e pode ser visualizada na forma de um gráfico, como a da SS.

No programa de finalização s.m tem-se como entrada MH, onde cada linha é formada por 256 pontos complexos e simétricos. Essa matriz tem $F_s \cdot \text{segs}$ linhas, cada uma representando um instante do espectro do som. Para cada linha da MH, calcula-se seqüencialmente a sua IFFT. A saída de cada IFFT será 256 pontos reais, representando cerca de 31ms de som janelado. De acordo com o valor da taxa de entrelaçamento *ov*, as saídas da IFFT são somadas, nos seus $ov \cdot 256$ pontos extremos.

Desenvolvemos dois programas para ajudar na análise do som transformado. São eles: zom.m que permite visualizar um pequeno trecho do som modificado, juntamente com o seu espectro; e som.m que permite a audição do som original e transformado. A utilização do programa zom.m é mostrada abaixo para um som emitido por uma nota contínua de violoncelo.

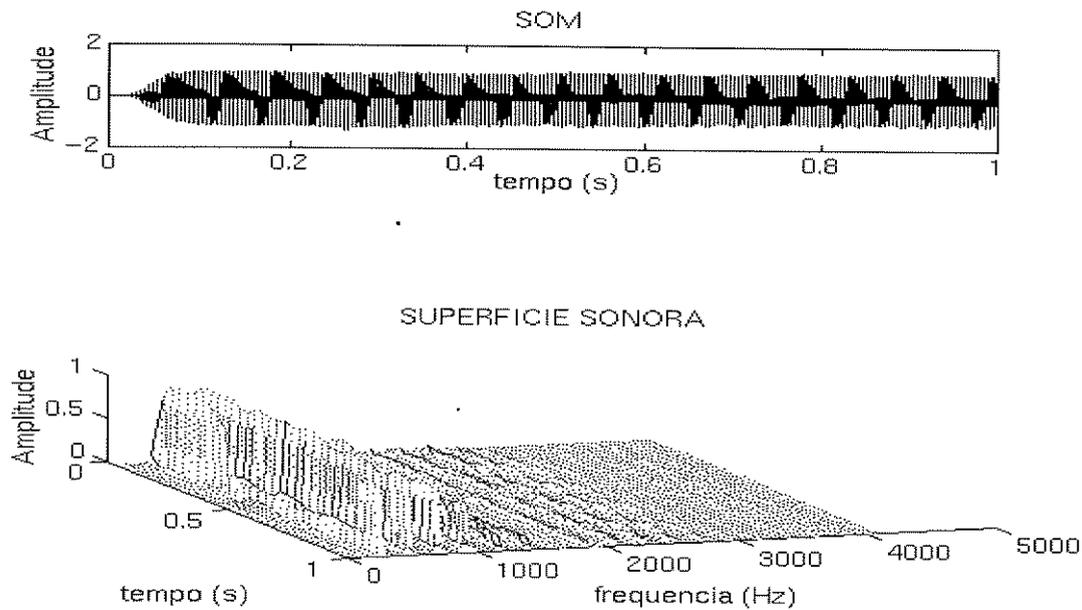


Fig. 4.3 Som de violoncelo. (a) s(t). (b) SS.

Como o programa *zom.m*, escolheu-se um trecho do som, como na figura 4.3(a). Na figura 4.3(b) é mostrado a SS de um trecho para visualização, aproximadamente entre os pontos 2.400 e 3.300 do arquivo de som original de 8.192 pontos.

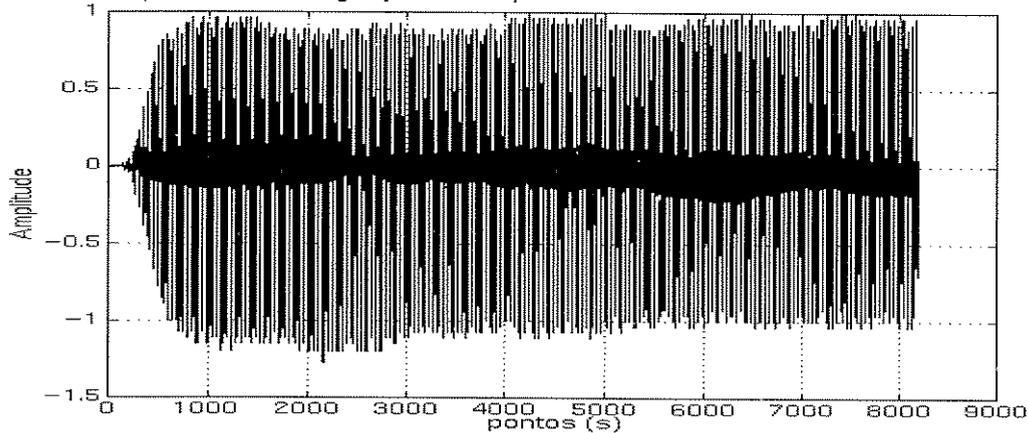


Fig. 4.4 Som de violoncelo. Envolvória de amplitude.

Na figura 4.4 a seguir vê-se o som de um violoncelo. Observa-se os padrões quase-periódicos do som. O intervalo de 900 pontos analisado equivale aproximadamente a 110 ms. Também é apresentado o espectro desse trecho de som. Observa-se que cerca de 11 harmônicos principais estão presentes no trecho escolhido.

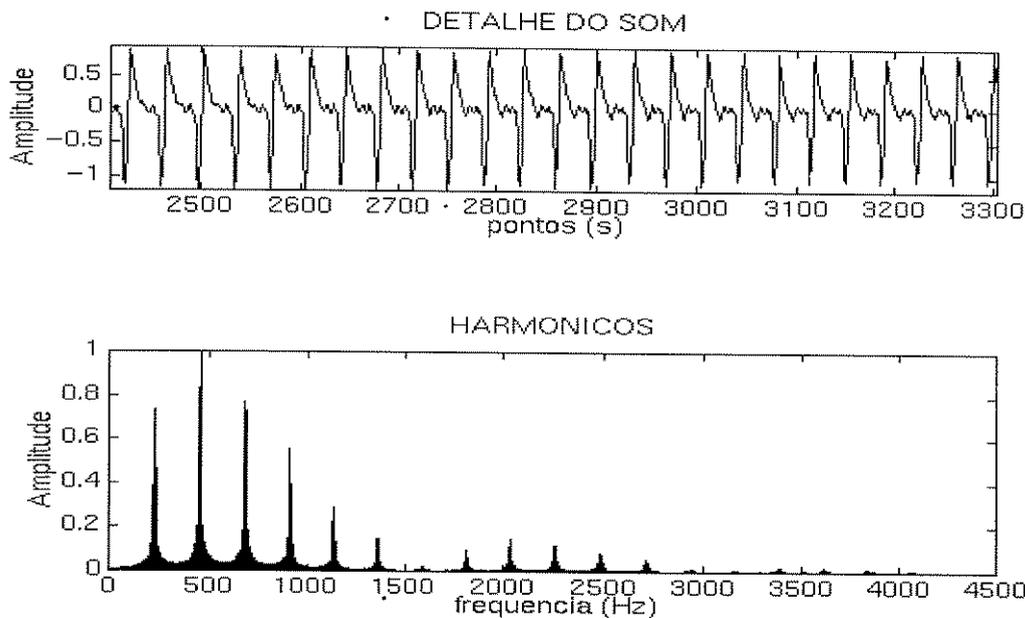


Fig. 4.5 Som de violoncelo. (a) Trecho de cerca de 110ms podendo-se ver a forma de onda. (b) Espectro em frequência do trecho visto em (a).

As operações timbrais foram feitas por vários programas que utilizam operações vetoriais básicas sobre a MH, como: adição, subtração, produto, divisão, inversão e deslocamento dos seus elementos. Através dessas operações criamos programas que simulam operações timbrais mais sofisticadas, como as que são exemplificadas a seguir.

4.3 Os resultados das simulações de MH

Foram realizadas simulações da montagem da MH para diferentes valores da taxa de amostragem F_s , número de pontos por espectro discreto N , resolução do eixo da frequência Δf e janelamento para diferentes taxas de entrelaçamento ov .

A figura 4.6 mostra a SS correspondente à MH montada para $F_s = 4.096$ Hz e $N = 128$ pontos. Cada espectro corresponde a um intervalo de tempo $\Delta t = 31$ ms. O sinal de entrada é uma senóide de frequência $f_0 = 1280$ Hz, multipla da resolução do eixo da frequência $bin = 32$ Hz.

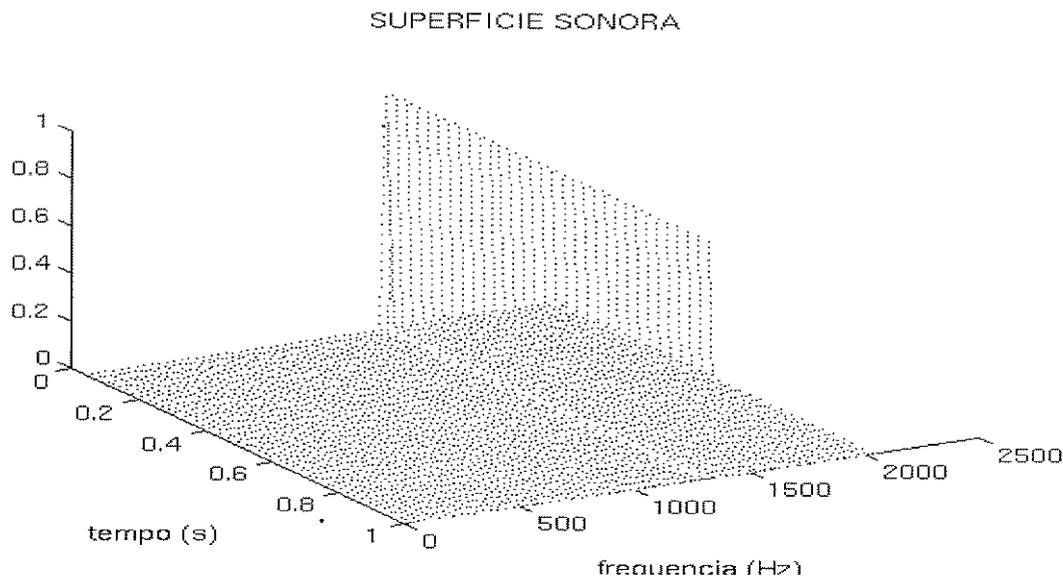


Fig. 4.6 SS de uma senóide de $f_0=1280$ Hz. $F_s = 4.096$ Hz.

Note que a senóide cuja freqüência é múltiplo de *bin* é adequadamente representada pela SS. Não se observa qualquer tipo de espalhamento do espectro (*leakage*).

Na figura 4.7 observa-se uma senóide de freqüência $f_0=1280$ Hz, da figura 4.6, mais 10 harmônicos distanciados entre si na razão de $(3/2)*f$ e com magnitude decrescente. $F_s = 2048$ Hz.

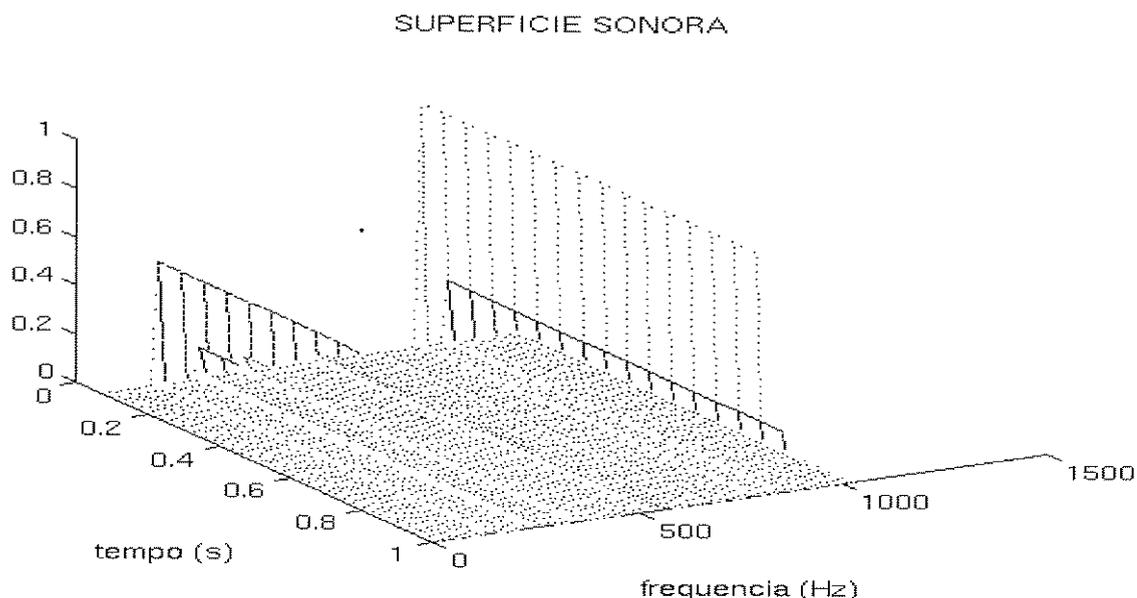


Fig. 4.7 Efeito de *aliasing* para senóide com $f_0=1280$ Hz mais 10 harmônicos com $f_i=(3/2)*f_{i-1}$. $F_s = 2.048$ Hz

Na figura 4.7 observa-se o efeito de *aliasing* sobre o sinal de entrada uma vez que todos os harmônicos componentes do sinal estão acima de $F_s/2$ Hz.

Na figura 4.8 são mostradas três senóides cujas freqüências são: 1.280 Hz, 2.560 Hz e 3.840 Hz para $F_s=4.096$ Hz.

SUPERFICIE SONORA

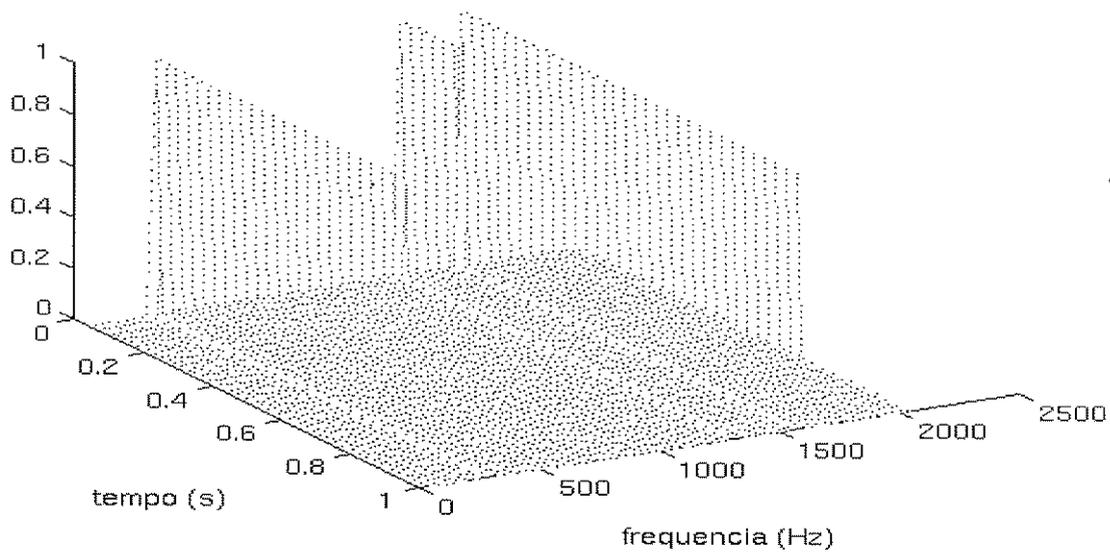


Fig. 4.8 Efeito de *aliasing* para três senóides com frequências: $f_0= 1.280$ Hz, $f_1= 2.560$ Hz e $f_2= 3.840$ Hz. $F_s=4.096$ Hz.

Na figura 4.8 o efeito do *aliasing* é evidenciado pelo deslocamento das senóides em relação ao eixo de freqüência delas. Apesar de não haver *leakage* pois as frequências das senóides são múltiplas de *bin* nenhuma delas senóides está representada no ponto correspondente a sua frequência original.

Na figura 4.9 vê-se a SS de um sinal senoidal cuja frequência $f_0=1296$ Hz não é múltipla de *bin*. Os múltiplos de $bin = 32$ mais próximos são 1.280 Hz e 1.312 Hz. A senóide de entrada tem frequência $f = 1.296$ Hz.

SUPERFICIE SONORA

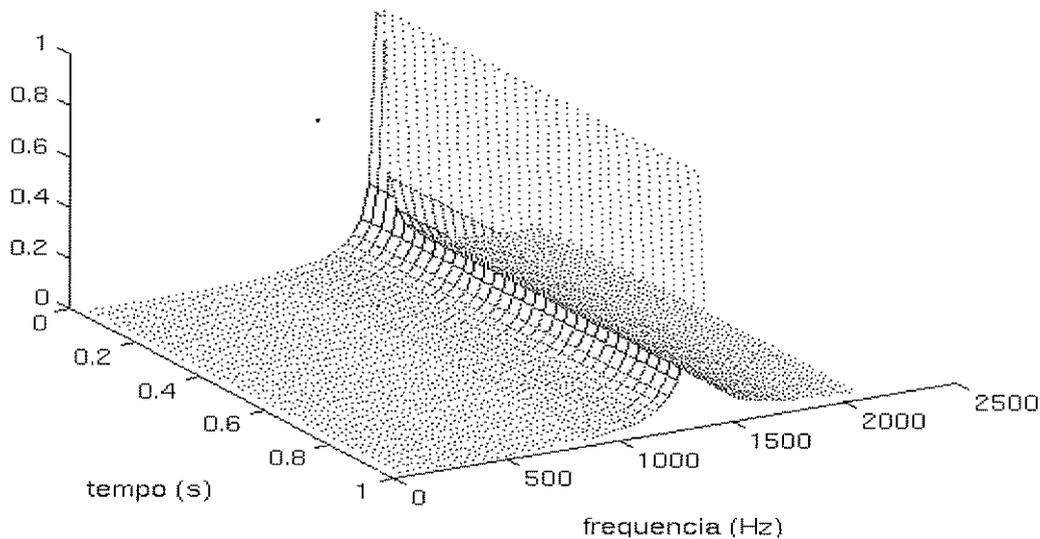


Fig. 4.9 O efeito de *frequency leakage* em uma senóide de $f_0=272$ Hz num eixo de frequência com resolução $bin=32$ Hz.

Na figura 4.9 observa-se o espalhamento do espectro na frequência, ou *frequency leakage*. Como a SS está normalizada não se percebe o efeito do *leakage* na redução da amplitude original do harmônico.

Na figura 4.9 foram utilizados $N = 32$ pontos e $F_s = 1.024$ Hz. O valor da senóide de entrada é $f = 272$, exatamente entre os pontos 256 e 288 Hz.

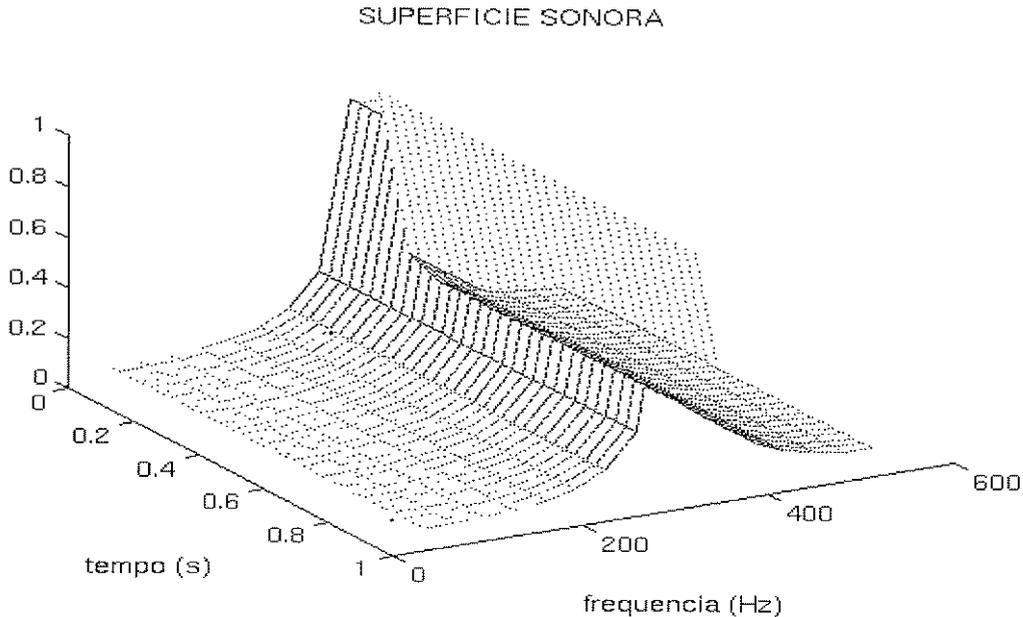


Fig. 4.10 *Frequency leakage* num eixo de frequência cuja resolução é $bin=128$ Hz.

A figura 4.10 mostra que com uma frequência de entrada mais baixa pode-se observar melhor o efeito do *leakage*. Apesar do gráfico da SS estar mostrando uma curva contínua de inclinação aproximadamente exponencial nas laterais da frequência da senóide de entrada, o *leakage* gera novos harmônicos, na região vizinha à frequência da senóide. Este é um dos principais problemas enfrentados na representação da SS pela MH. Por melhor que seja a sua resolução *bin*, sempre haverá algum *leakage* quando a frequência da senóide analisada não for múltipla de *bin*.

O efeito de *leakage* pode ocorrer (como aliás na maioria dos casos) com os harmônicos superiores do som, mesmo que a sua fundamental seja múltiplo de *bin*. Na figura 4.11, usou-se como entrada um sinal com o harmônico fundamental $f = 640$ Hz seguido por 10 harmônicos naturais (do mesmo modo que utilizado anteriormente). A taxa de amostragem é $F_s = 4.096$ Hz, $N = 32$ pontos e $bin = 128$ Hz.

SUPERFICIE SONORA

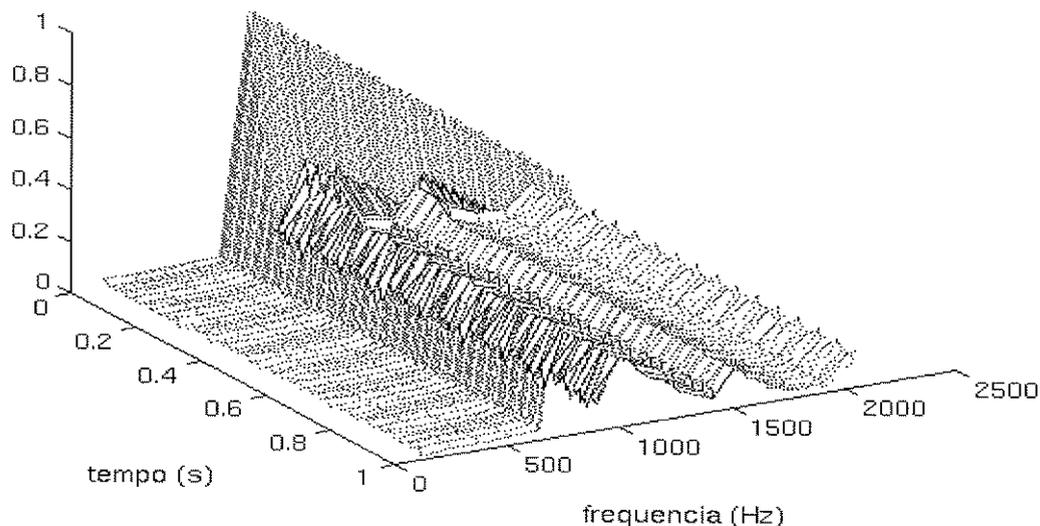


Fig. 4.11 *Frequency leakage* de som com 10 harmônicos onde $f_0=640$ Hz . $F_s = 4.096$ Hz. $bin = 128$ Hz.

Observa-se na figura 4.11 que apesar da fundamental não provocar *leakage* todos os seus harmônicos o ocasionam. O resultado é a formação de novos harmônicos entre os harmônicos originais.

Nesta simulação e nas seguintes mostra-se o efeito do janelamento e entrelaçamento (*overlap*) do sinal no tempo. Utiliza-se esta técnica para se evitar o ruído *glitch* gerado quando uma seqüência de valores amostrados possui fase diferente de zero, ou seja, seus pontos extremos não tendem a zero. A DFT interpreta a seqüência como periódica gerando ruído. A técnica mais utilizada para evita-lo é o janelamento, em especial a janela *Hanning*. Na simulação seguinte referente à figura 4.13 utilizou-se: $F_s = 8192$ Hz, $N = 256$ pontos, $bin = 32$ Hz. O som utilizado é uma nota de oboé. Na figura 4.12 é mostrado o som original.

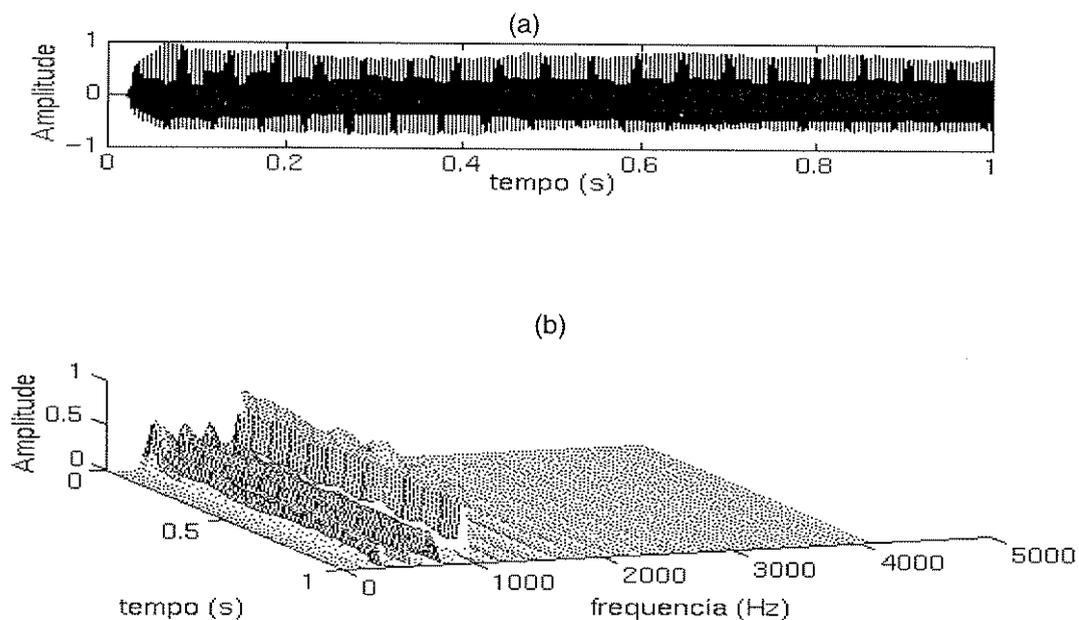


Fig. 4.12 Som de oboé. (a) sinal de audio $s(t)$. (b) SS correspondente.

A seguir, na figura 4.13, aplicou-se a janela *Hanning* sobre cada vetor harmônico, representando um plano harmônico da SS. Observa-se na figura 4.13 o efeito do janelamento na envoltória de amplitude. O janelamento por si só distorce o sinal no tempo. A maneira de se evitar essa distorção é através do entrelaçamento dos espectros.

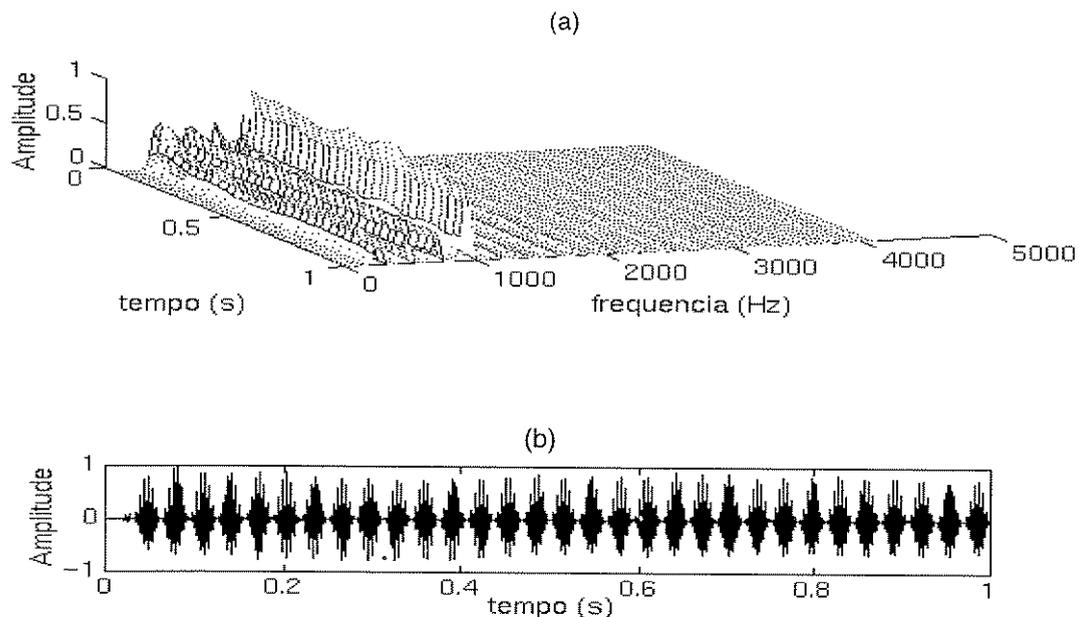


Fig. 4.13 Som janelado, sem entrelaçamento. (a) SS do som janelado. (b) $s(t)$ janelado.

Na figura 4.14 tem-se o resultado do entrelaçamento dos espectros discretos na razão de 25%.

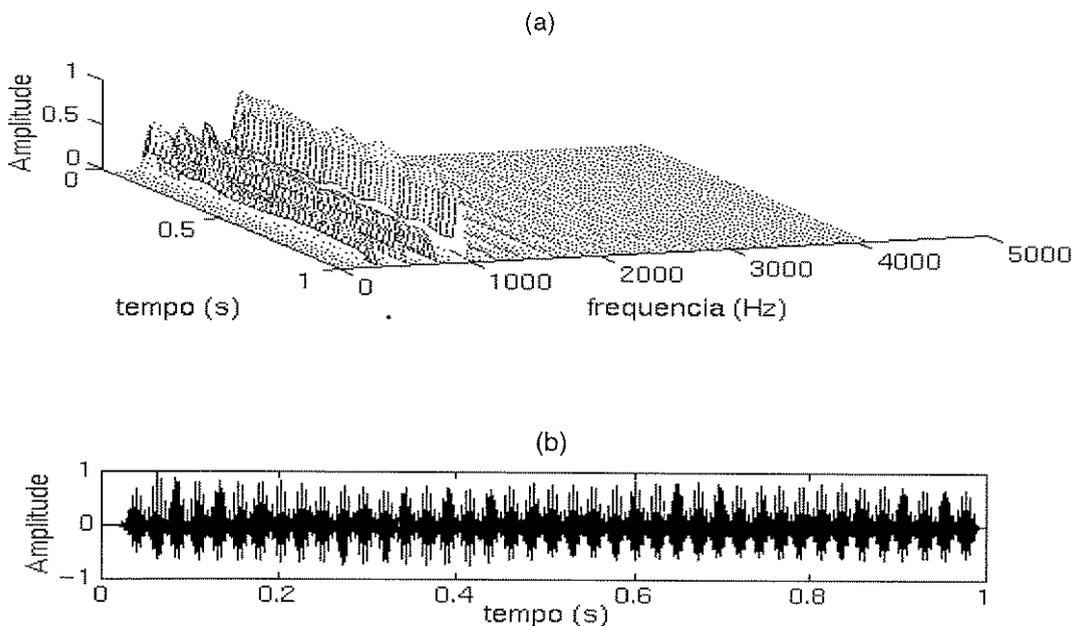


Fig. 4.14 Som de oboé com janelamento e entrelaçamento de 25%. (a) SS. (b) $s(t)$.

Na figura 4.15 vê-se o resultado do entrelaçamento dos espectros discretos na razão de 50%. O sinal não apresenta diferenças significativas com o sinal sonoro original.

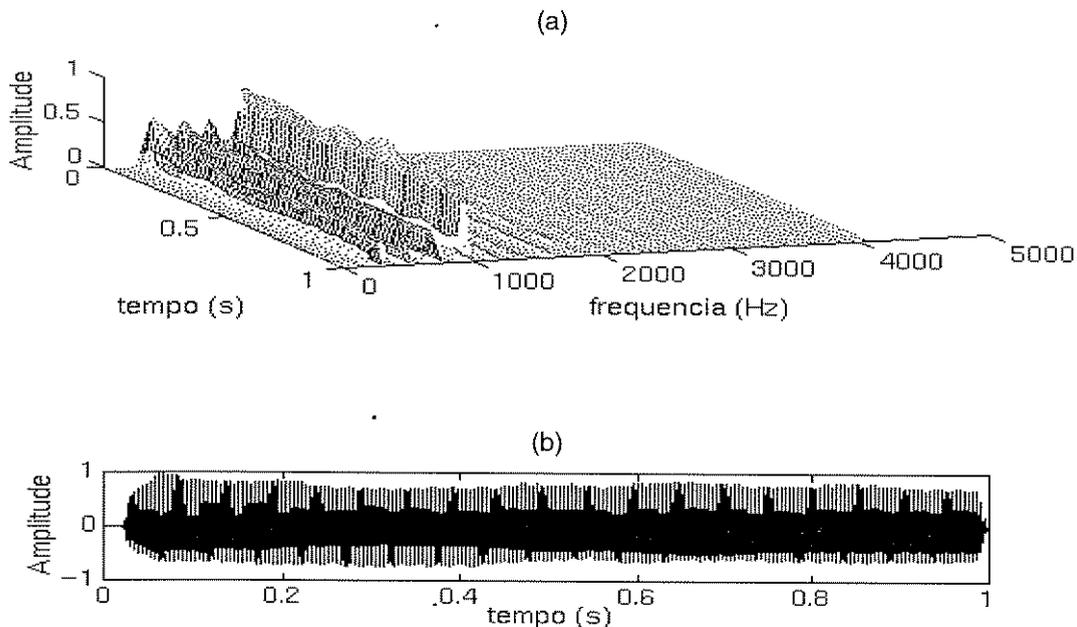


Fig. 4.15 Som janelado, com entrelaçamento de 50%.(a) SS. (b) s(t).

4.4 Os resultados da simulação das operações timbrais

A seguir são mostrados exemplos gráficos da simulação de operações timbrais. Cada operação foi aplicada sobre um arquivo de som original, que é mostrado inicialmente. Em cada figura vê-se: o arquivo completo, na forma de envoltória no tempo, a representação de sua SS, onde se observa a variação dos harmônicos componentes do espectro de cada som. Nota-se a observabilidade aos detalhes do som, oferecida pela SS. Nela pode-se ver o resultado de cada transformação sonora, de forma mais detalhada. Os arquivos de som utilizados nos exemplos são todos de duração inicial de 1 segundo e não apresentam qualquer tipo de janelamento, ou entrelaçamento (*overlap*).

- **Compressão dos níveis máximo e mínimo da magnitude dos elementos da MH:** Foi utilizado o operador *cp.m*. O som inicial é o de um apito de trem, porque este apresenta bastante variação de amplitude.

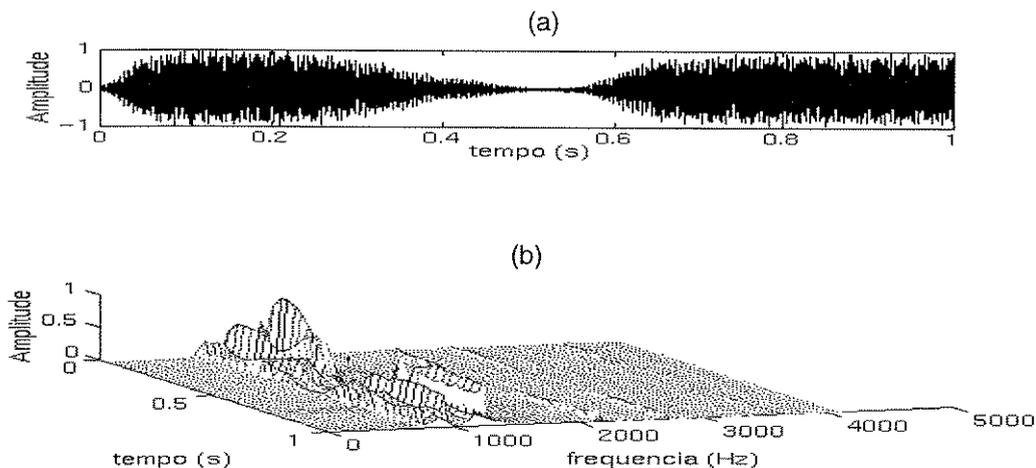


Fig. 4.16 Som de apito de trem.(a) s(t). (b) SS.

O operador *cp.m* permite a parametrização da taxa de compressão, entre 0 e 100%. Utilizou-se a taxa de compressão de 90%. Nota-se na *SS* modificada o crescimento da magnitude dos harmônicos anteriormente menores. Desse modo o espectro do som passa a ter mais harmônicos “*significativos*”. O resultado pode ser visto na envoltória da amplitude da figura 4.17, nele observando-se o efeito do acréscimo de harmônicos de maior frequência.

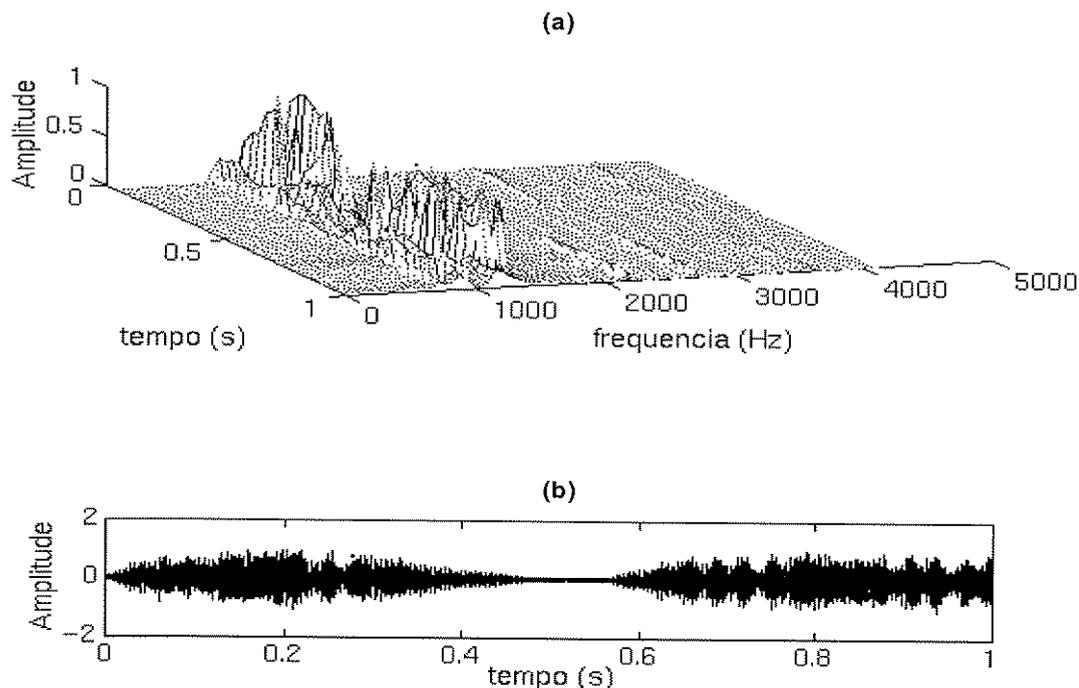


Fig. 4.17 Som de apito de trem. Compressão da magnitude dos harmônicos na taxa de 90%. (a) *SS*'. (b) *s'(t)*.

- **Expansão (ou compressão) da magnitude do espectro do som:** Utilizou-se aqui o operador *ec.m*. Este operador permite comprimir ou expandir o espectro de um som, na razão de 1 para 2. Isto é feito através do distanciamento em frequência dos harmônicos do espectro.
O som utilizado é o de um violino. Observa-se na figura 4.18 a riqueza de harmônicos presentes em seu espectro.

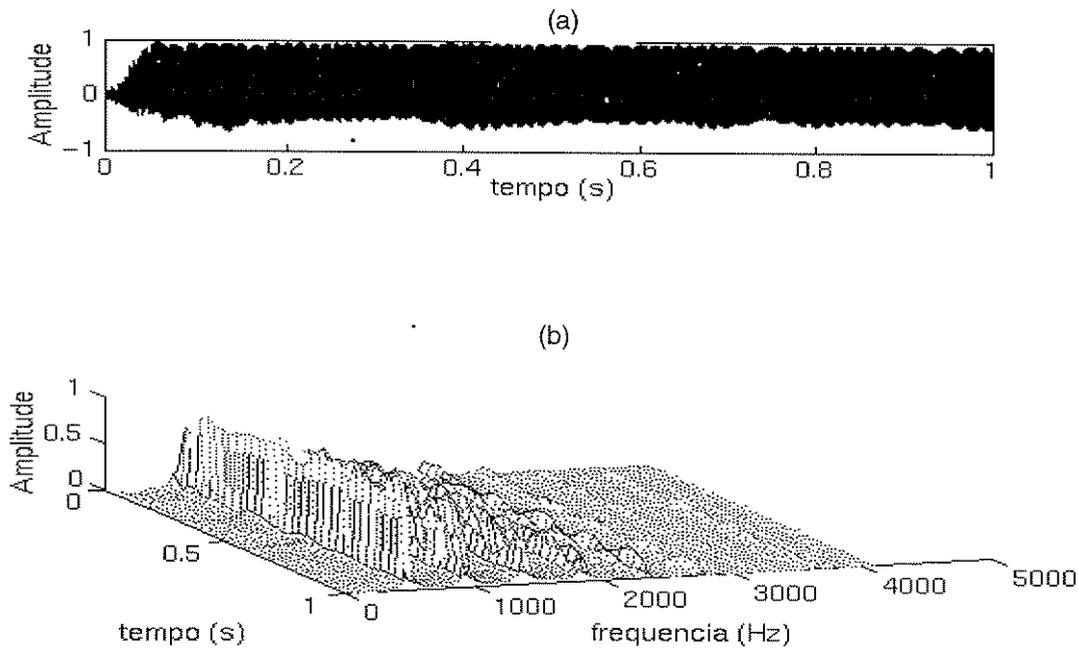


Fig. 4.18 Som de violino (a) $s(t)$. (b) SS.

Na figura 4.19 observa-se que o som original é transformado pelo operador timbral, parametrizado para expandir o espectro, na razão de 1 para 2. Nota-se na SS modificada o espalhamento dos harmônicos, para regiões mais graves e mais agudas. A envoltória do som, na figura 4.19 mostra a distorção provocada por este operador.

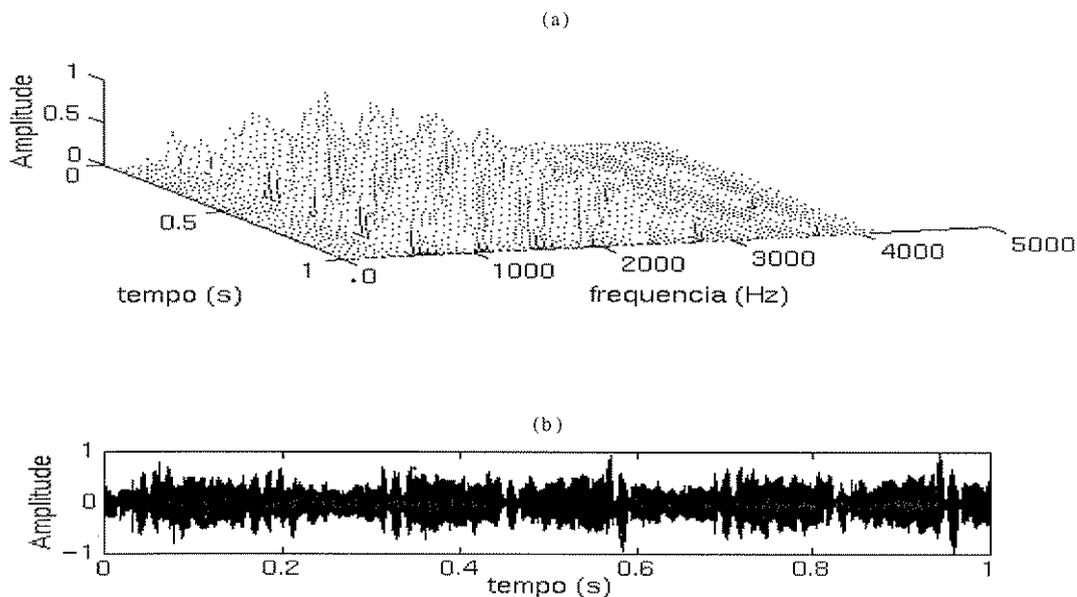


Fig. 4.19 Som de violino. Expansão do espectro do som (2:1). (a) SS' . (b) $s'(t)$.

- **Modificação da envoltória dos planos morfológicos da SS:** Na figura 4.20 observa-se a SS do som de um violoncelo.

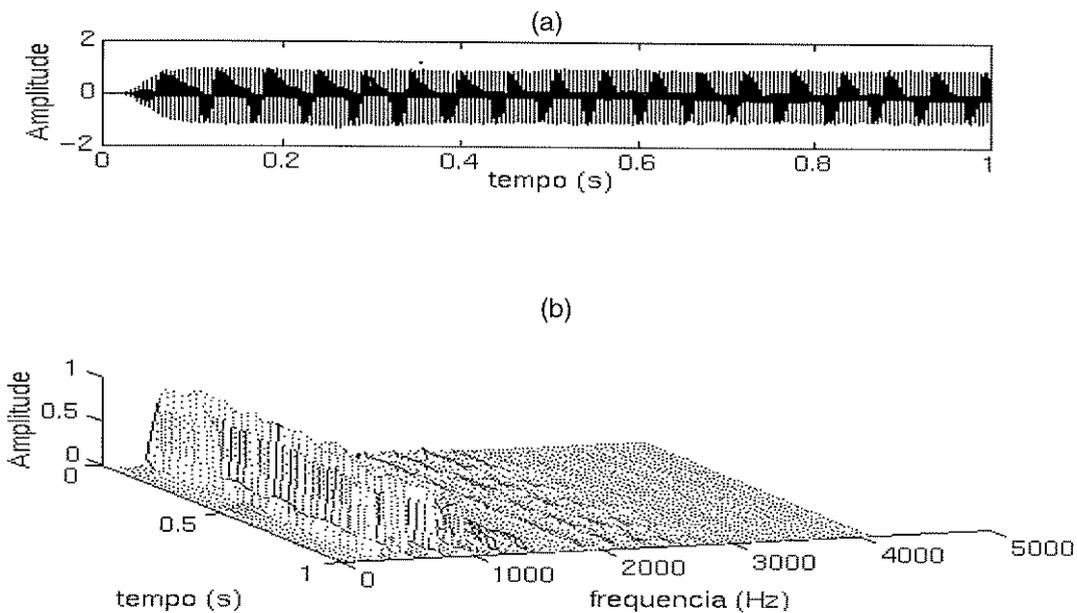


Fig. 4.20 Som de violoncelo. (a) $s(t)$. (b) SS.

Na figura 4.20 vê-se a SS. Sobre a $s(t)$ da figura 4.20(a) foi aplicada uma nova envoltória, mostrada na figura 4.20(b).

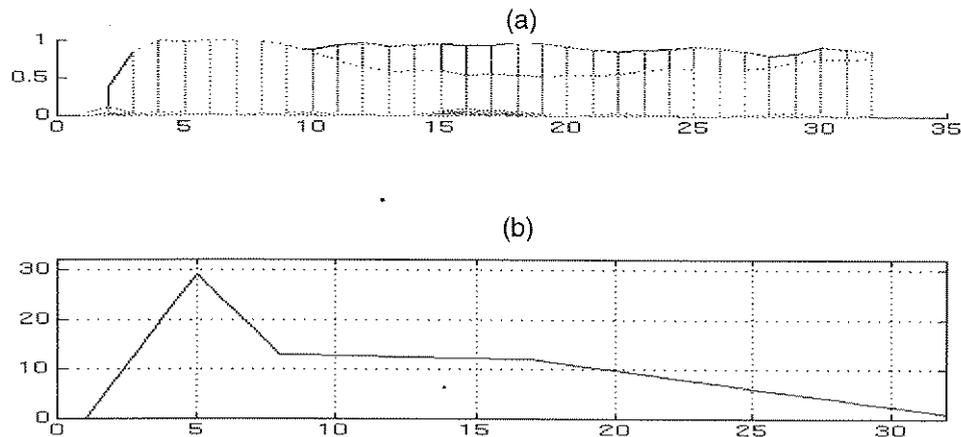


Fig. 4.21 Modificação da envoltória dos planos morfológicos da SS. (a) Planos morfológicos. (b) Forma da suas novas envoltórias.

O resultado é visto na figura 4.21 através da SS modificada. A composição dos harmônicos do espectro permanece inalterada.

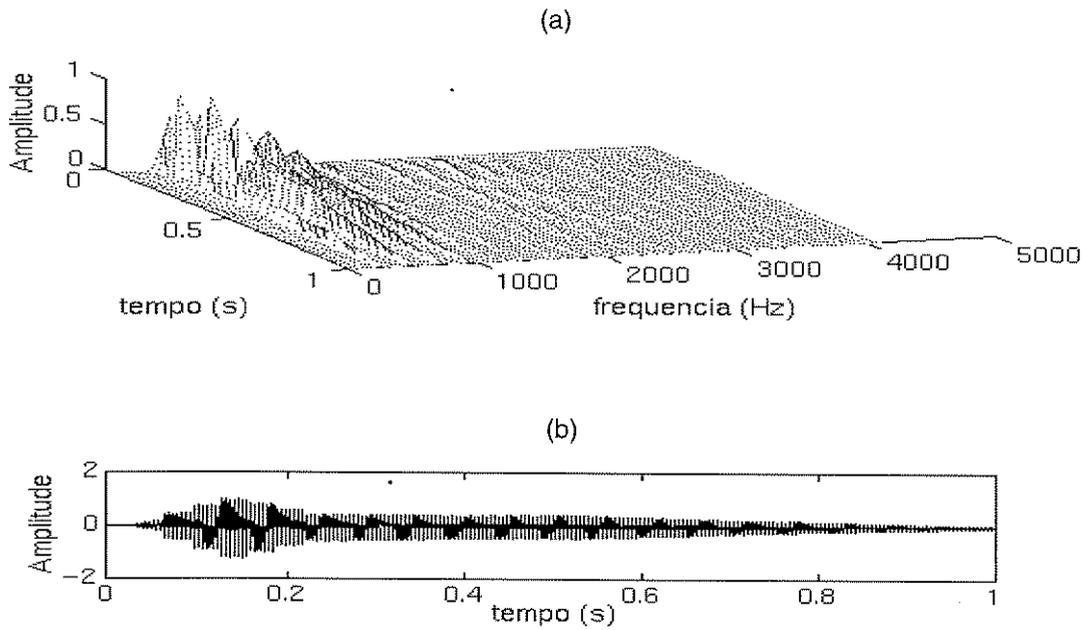


Fig. 4.22 Som de violoncelo. Aplicação de nova envoltória para os planos morfológicos. (a) SS' . (b) $s'(t)$.

- **Modificação da envoltória dos planos harmônicos:** Para demonstrar a modificação da envoltória espectral, utilizou-se o som de um trombone, mostrado na figura 4.23.

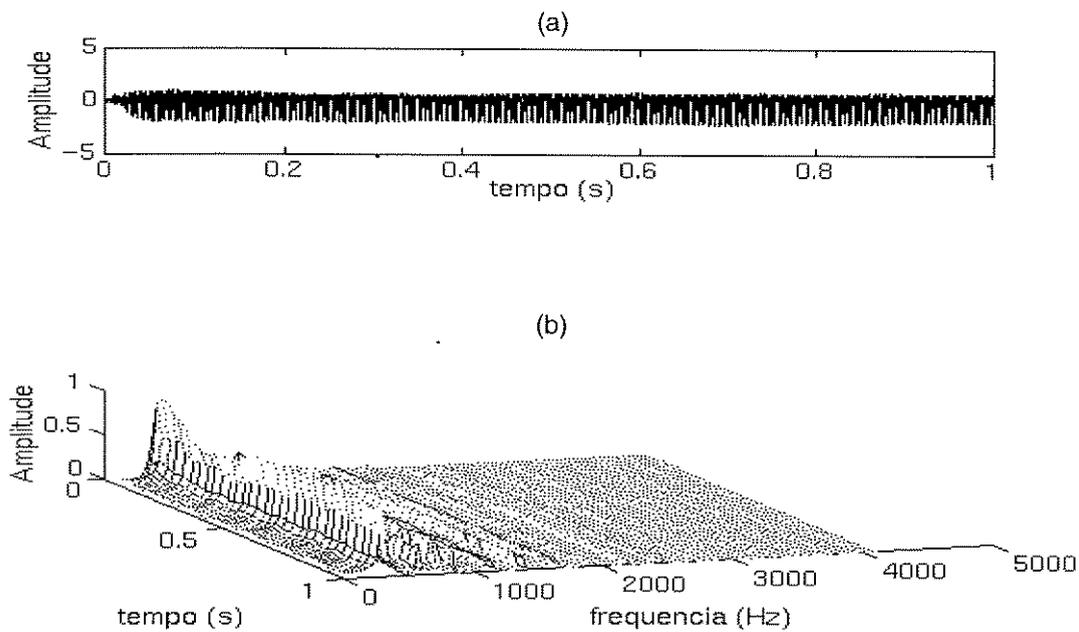


Fig. 4.23 Som de trombone. (a) $s(t)$. (b) SS .

Na figura 4.24 mostra-se na nova envoltória dos planos espectrais o harmônico fundamental do som original. A modificação de suas envoltórias equivale à filtragem de $s(t)$, no caso equivalente à um filtro passa alta.

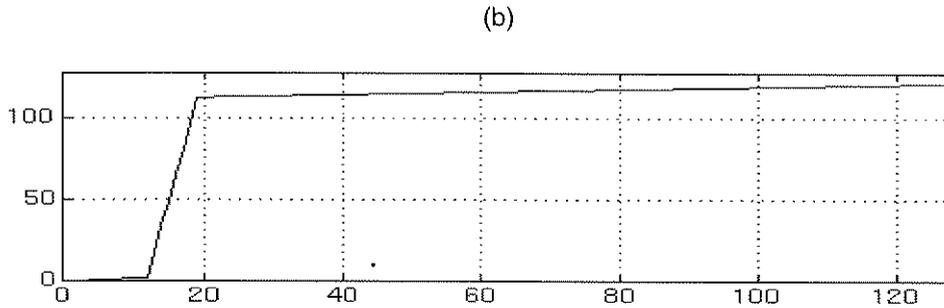
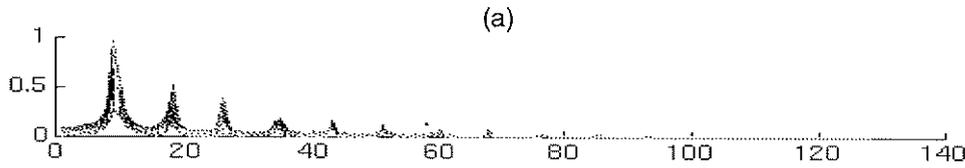


Fig. 4.24 Som de trombone. Modificação da envoltória dos planos harmônicos. (a) Planos harmônicos. (b) Forma de suas novas envoltórias.

O resultado é observado na figura 4.25 onde observa-se a ausência do harmônico fundamental original. O restante dos harmônicos permanece inalterado. Uma vez que se pode redesenhar a envoltória, tem-se um sistema de equalização do som bastante flexível. Reparametrizando o operador para cada plano harmônico, tem-se um sistema de equalização que opera modificações na envoltória espectral da SS de forma programável.

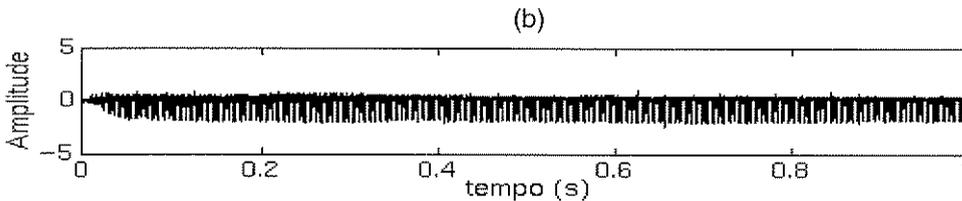
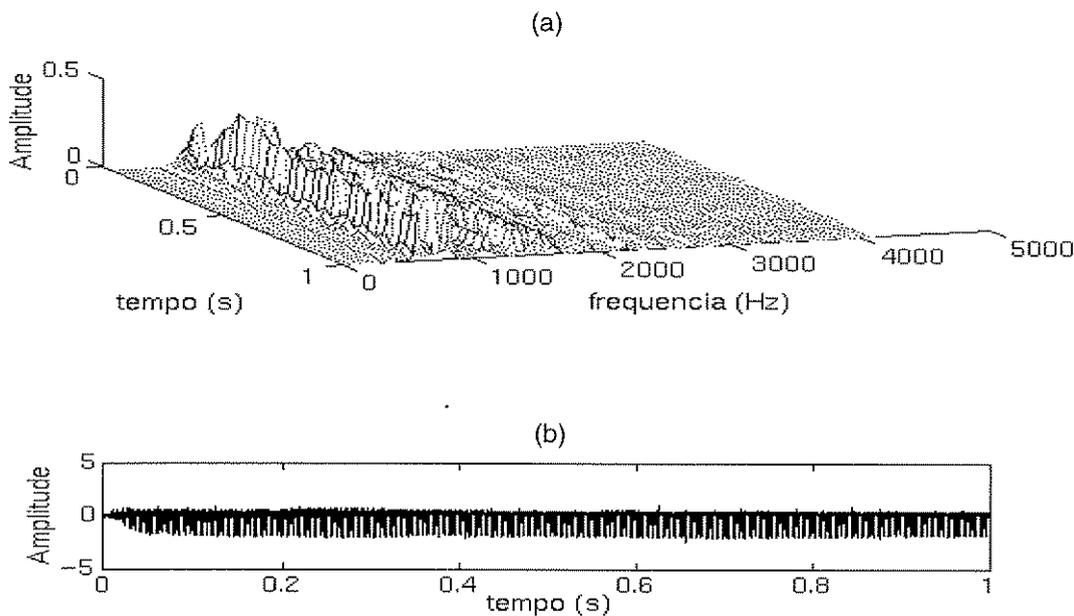


Fig. 4.25 Som de trombone. Aplicação da envoltória espectral da figura 4.24(b). (a) SS' . (b) $s'(t)$.

- **Redução do número de harmônicos do espectro:** Este operador preserva apenas os “ n ” harmônicos de maior magnitude. O restante dos harmônicos componentes do espectro original é suprimido. Foi utilizado o som de um clarinete. Na figura 4.26(b) pode-se observar a predominância de um harmônico fundamental, aproximadamente de frequência igual a 440 Hz.

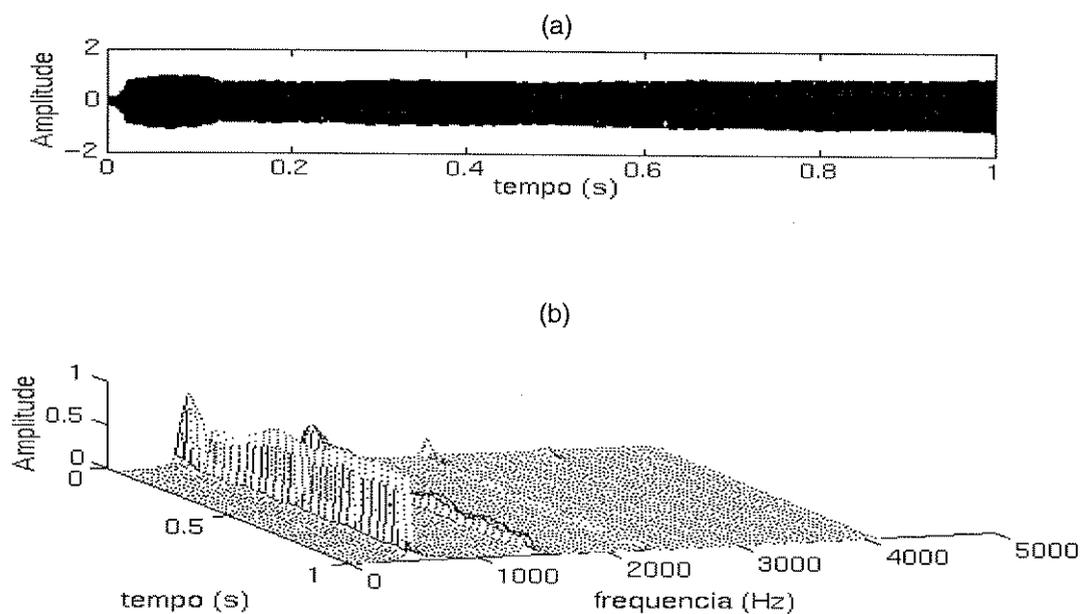


Fig. 4.26 Som de um clarinete. (a) $s(t)$. (b) SS .

O operador foi parametrizado para conservar apenas 1 harmônico. Na figura 4.27(a) vê-se apenas o harmônico fundamental. Nota-se também que a envoltória de amplitude, figura 4.27(b), permanece igual à original.

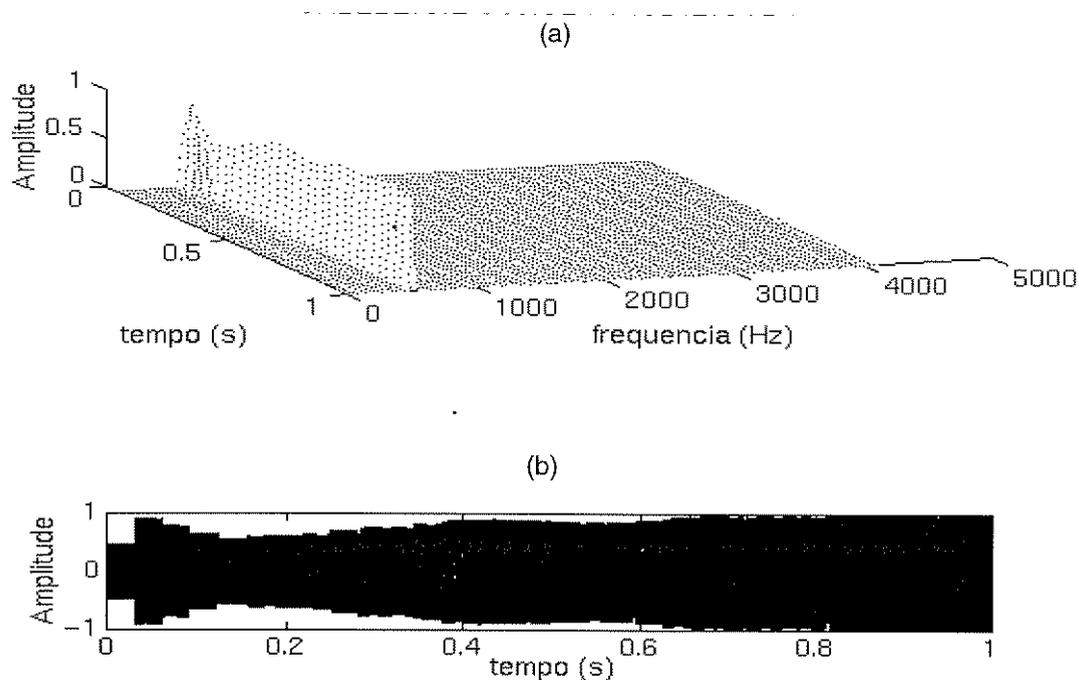


Fig. 4.27 Som de clarinete. (a) SS' . (b) $s'(t)$.

Na figura 4.28 mostra o primeiro segundo do trecho da peça musical (oratório) *Aleluia*, de Händel. Neste primeiro segundo o coral, acompanhado pela orquestra, canta "a-leluia, a-". Este som foi utilizado pela sua riqueza espectral, uma vez que toda a orquestra e todo o coral atuam ao mesmo tempo.

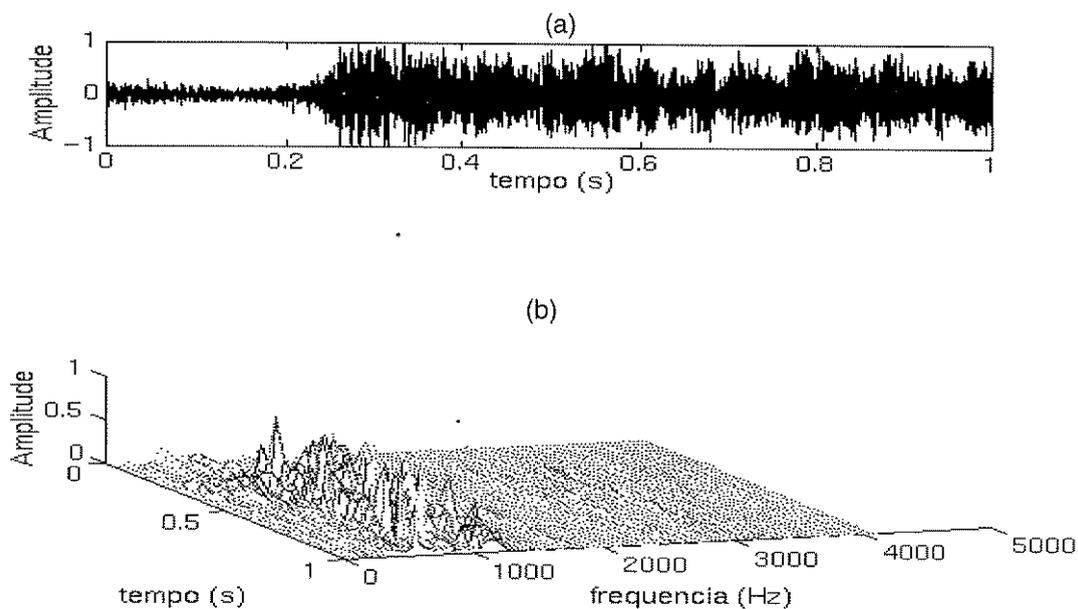


Fig. 4.28 Trecho de *Aleluia*, de Händel. (a) $s(t)$. (b) SS .

O operador foi parametrizado afim de preservar apenas os 3 harmônicos de maior amplitude. Apesar da modificação não ser muito visível na figura 4.29(a), nota-se a modificação em $s'(t)$. Nesta pode-se observar o efeito da diminuição do número de harmônicos.

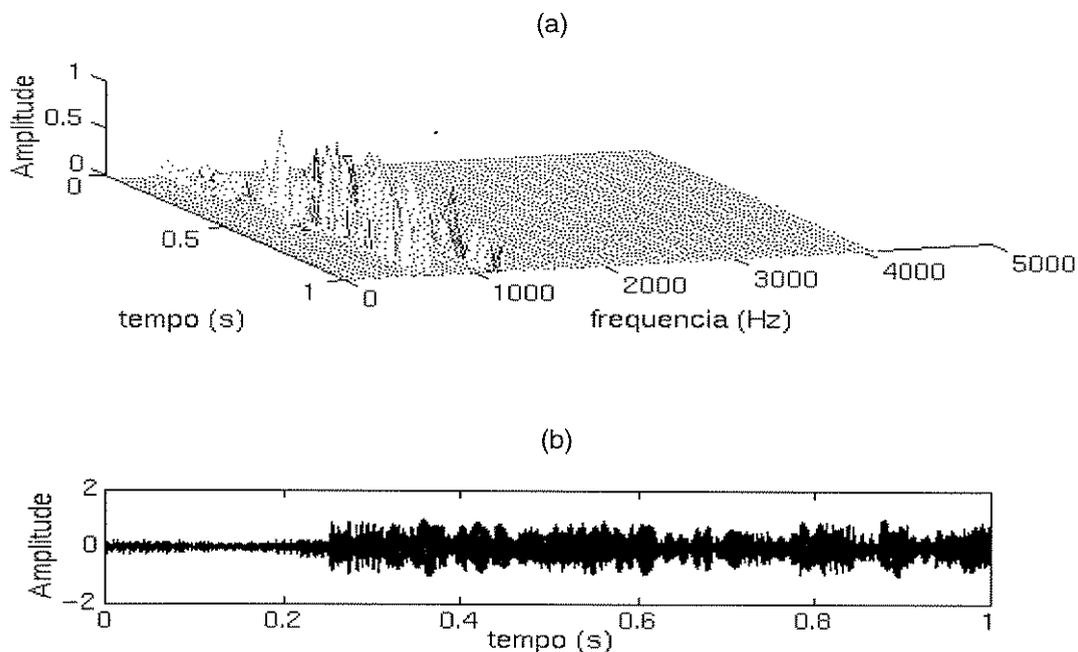


Fig. 4.29 Preserva apenas os três maiores harmônicos. (a) SS' . (b) $s'(t)$.

- **Modificação do *pitch* do som:** O operador aqui utilizado desloca os harmônicos no eixo da frequência. O deslocamento não é circular, portanto no som utilizado os harmônicos da extremidade do espectro são suprimidos. Na figura 4.30 vê-se o som emitido por uma flauta.

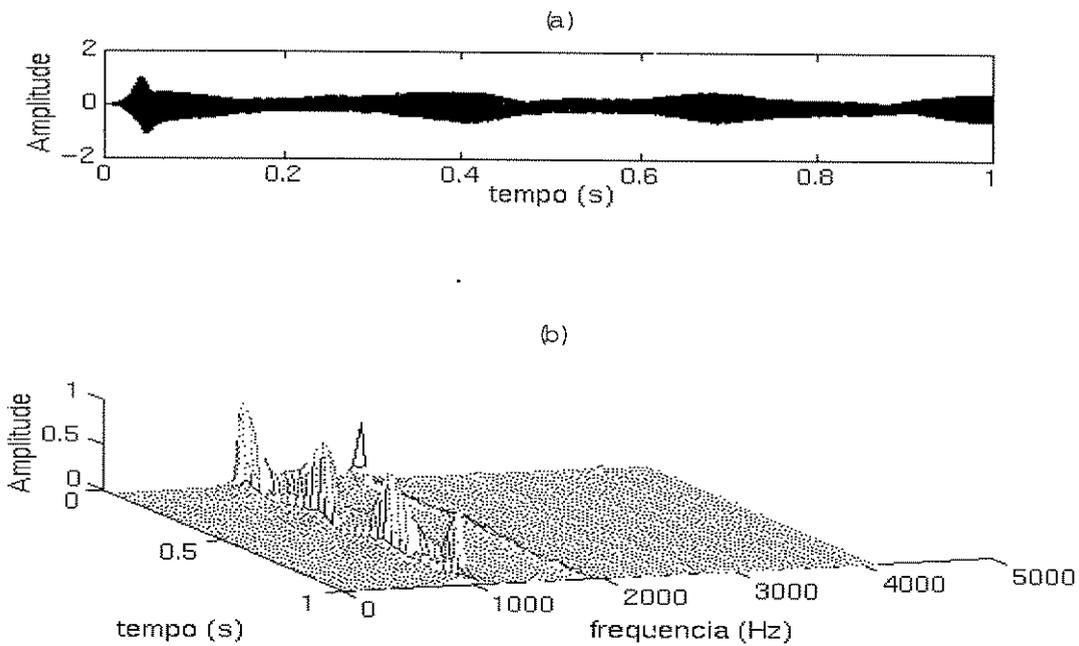


Fig. 4.30 Som de flauta com harmônico fundamental $f_0 = 880$ Hz. (a) $s(t)$. (b) SS.

O operador foi parametrizado para deslocar o espectro duas oitavas abaixo, equivalente ao harmônico fundamental indo para $f_0 = 220$ Hz. Observa-se que, com excessão ao deslocamento, todo o espectro permanece inalterado, especialmente em relação ao seus planos morfológicos. O resultado é a modificação do *pitch* sem a modificação da envoltória do tempo. O som torna-se duas oitavas mais grave, porém com a mesma duração no tempo.

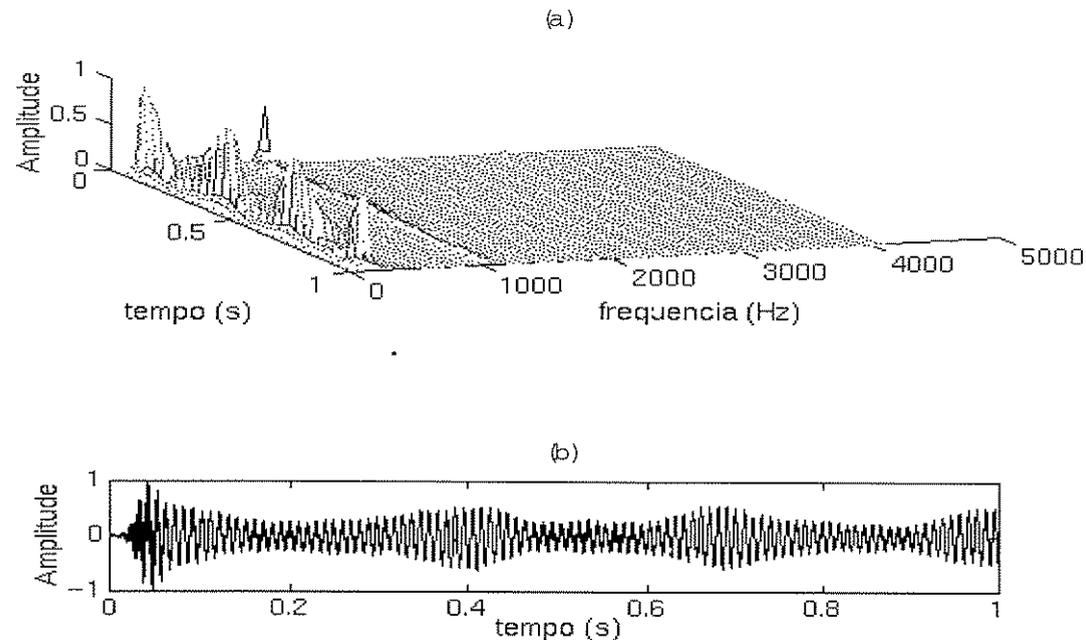


Fig. 4.31 Som de flauta. Deslocamento do *pitch* original, de 880 Hz para 220Hz. (a) SS' . (b) $s'(t)$.

Capítulo 5. Projeto do sistema

Este capítulo aborda o dimensionamento e projeto de um sistema digital de transformação sonora através de operações timbrais. Como foi mostrado no terceiro capítulo, o sistema possui cinco módulos de processamento. Estes podem ser agrupados em: entrada do som, operação timbral e saída. Neste capítulo, cada uma dessas três partes é tratada separadamente, e dividida nos seus estágios componentes.

Observou-se que o sistema deve trabalhar em tempo-real, para executar as operações timbrais e promover a imediata audição dos resultados pelo usuário.

Portanto o tempo de execução é de grande importância neste projeto. No capítulo anterior foram mostradas as simulações feitas em software do sistema e que contribuem para o dimensionamento do sistema.

Nas simulações utilizou-se a entrada e a saída com o mesmo padrão de representação digital. O equipamento foi simulado para operar transformações sonoras no som de entrada. Neste projeto a entrada será usada para o controle das operações timbrais. Deste modo diminuem-se as especificações do primeiro módulo, aumenta-se o tempo disponível (e a capacidade) para as operações timbrais e preserva-se a possibilidade de operação em tempo-real. Pode-se dizer que o sistema projetado deste modo estabelece uma nova geração de instrumentos musicais eletrônicos.

Os módulos 1 e 5 não apresentam grandes restrições ao projeto do sistema. Dessa forma é feito apenas o seu dimensionamento.

Ao final do capítulo faremos uma série de considerações gerais sobre o dimensionamento do sistema.

5.1 A entrada do sinal de audio e o operador timbral.

Da entrada do som analógico até a operação de seu timbre, existe uma série de partes de processamento. Dividiu-se esta primeira parte de processamento em dois módulos. Cada um desses módulos é dividido em fases de processamento.

Assim, na primeira parte do sistema tem-se os dois módulos:(1) pré-processamento, conversão A/D e montagem do vetor de entrada da FFT,(2) cálculos: da FFT e da magnitude dos harmônicos.

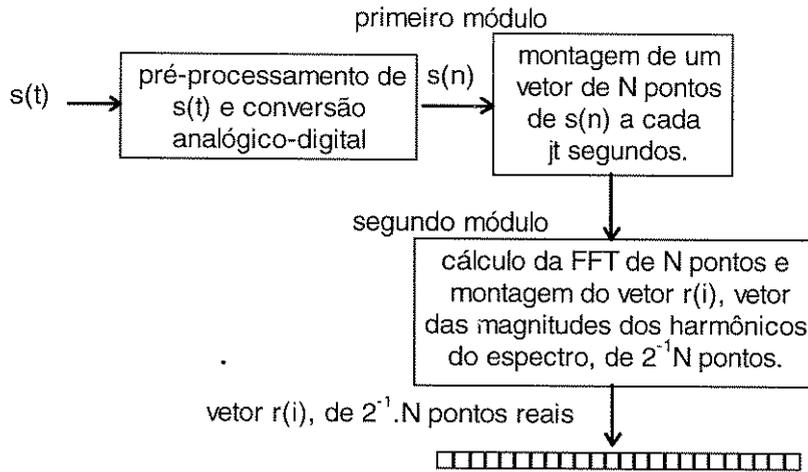


Fig. 5.1 Parte inicial do sistema.

No caso do primeiro módulo, tem-se diversas fases. No pré-processamento do sinal tem-se duas fases: pré-filtragem passa-faixa e controle de ganho. Na conversão A/D tem-se apenas a fase que representa o conversor A/D, que transforma $s(t)$ em $s(n)$, a montagem do vetor de entrada da FFT é a última fase do primeiro módulo. Estas quatro fases iniciais compõem o primeiro módulo da primeira parte do sistema.

Convenções do primeiro módulo: pré-processamento, conversão A/D e montagem do vetor:

- $s(t)$: sinal analógico de entrada
- $s(n)$: som amostrado no tempo.
- f_H e f_L : a frequência mais alta e a mais baixa existentes no espectro de $s(t)$, respectivamente.
- V_H e V_L : os limites de tensão (máximo e mínimo) da entrada do conversor analógico-digital.
- n : resolução do conversor A/D, número de bits que representa cada ponto de $s(n)$.
- F_s : taxa de amostragem do conversor A/D
- jt : intervalo de tempo correspondentes a N amostras de $s(n)$.

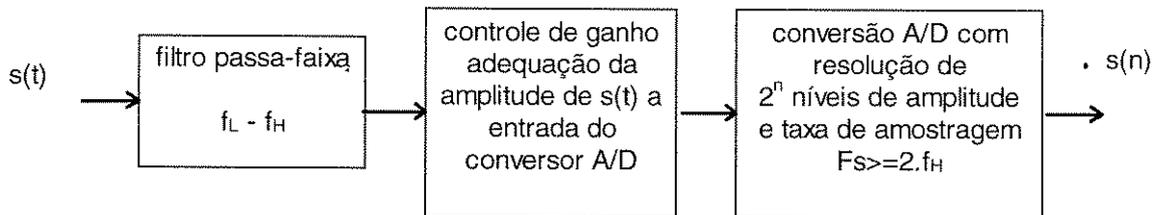


Fig. 5.2 Pré-processamento do sinal de entrada e conversão A/D.

A primeira e a segunda fase preparam $s(t)$ para a próxima fase (conversão A/D). Sua função é limitar a variação do sinal, no domínio da frequência e na amplitude. Após estas duas fases, o sinal está pronto para ser amostrado. Para a terceira fase (conversor A/D) tem-se a quantização de $s(t)$ na amplitude e no tempo. Em relação ao tempo tem-se o **teorema da amostragem**, que estabelece que o sinal $s(t)$, após a primeira fase (filtragem), deve ter o harmônico componente de maior frequência, f_H , respeitando: $F_s \geq 2 \cdot f_H$. Afim de se evitar *aliasing* do sinal (demonstrado no capítulo anterior) deve-se então limitar as componentes em frequência. Em relação a amplitude tem-se que ao final do processo a amplitude de cada uma das amostras estará contida dentro de 2^n níveis de amplitude linearmente espaçados, equivalentes a uma relação sinal/ruído: $SQNR = 20 \cdot \log_{10}(2^n)$ dB.

O sinal correspondente à entrada, representado por $s(t)$, é limitado em banda de frequência, por um filtro passa-faixa, limitado em nível de amplitude, por um controlador de ganho e quantizado no tempo

e na amplitude, por um conversor A/D. Assim obtém-se $s(n)$. Na quarta fase os valores de $s(n)$ são agrupados em vetores de N em N valores, similar a uma nova taxa de amostragem, cujo período de tempo de amostragem é jt .

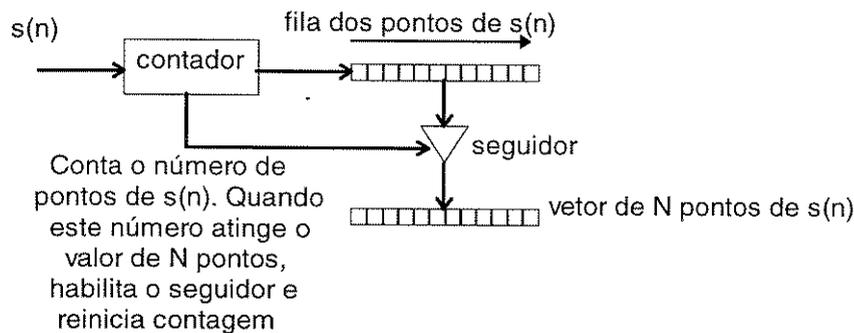


Fig. 5.3 Montagem do vetor de entrada da FFT

A saída da FFT, tem $N/2$ valores igualmente espaçados entre 0 até $F_s/2$ Hz. Assim o espaçamento entre os valores é igual a $F_s/2/N/2 = F_s/N$. Esse valor é chamado de *bin*. Do mesmo modo, a cada F_s valores, tem-se 1 segundo de som amostrado pelo conversor A/D, então N valores equivalem a N/F_s segundos de som. esse valor é chamado aqui de jt . Nota-se que $jt = 1/bin$. Portanto, jt é o intervalo de tempo correspondente a um vetor de N valores, amostrados em F_s Hz, entrada da FFT.

Os valores de $s(n)$ são alocados em um vetor a cada jt segundos, para a entrada da FFT. Este intervalo de tempo deve ser suficientemente pequeno para que a representação desses N valores no domínio da frequência seja um espectro cuja variação de sua envoltória é imperceptível para a audição. A cada intervalo de tempo correspondente a jt tem-se um vetor de N valores de $s(n)$, na entrada da FFT. O segundo módulo da primeira parte é composto pelo cálculo da FFT e cálculo do vetor r . Para o segundo módulo tem-se as seguintes convenções:

- N: número de amostras do vetor de $s(n)$, entrada da FFT.
- bin: intervalo entre os harmônicos de saída da FFT.
- jt : intervalo de tempo equivalente a N valores de entrada da FFT. $jt = 1/bin$.
- $r(i)$: vetor das magnitudes dos valores (complexos) de saída da FFT. $i = 1, 2, 3, \dots, 2^{11} \cdot N$

O vetor r das magnitudes da saída complexa da FFT, do tipo: $a(i) = j.b(i)$, é calculado com base na fórmula: $r(i) = (a(i)^2 + b(i)^2)^{1/2}$. Um vetor de N valores de $s(n)$ é posto na entrada da FFT a cada jt segundos. A saída da FFT será um vetor de N elementos complexos. Desta saída, calcula-se a magnitude dos $2^{11}N$ primeiros valores e monta-se o vetor $r(i)$. Isto porque, como a entrada da FFT é um vetor de números reais, a sua saída é simétrica (ver apêndice 1).

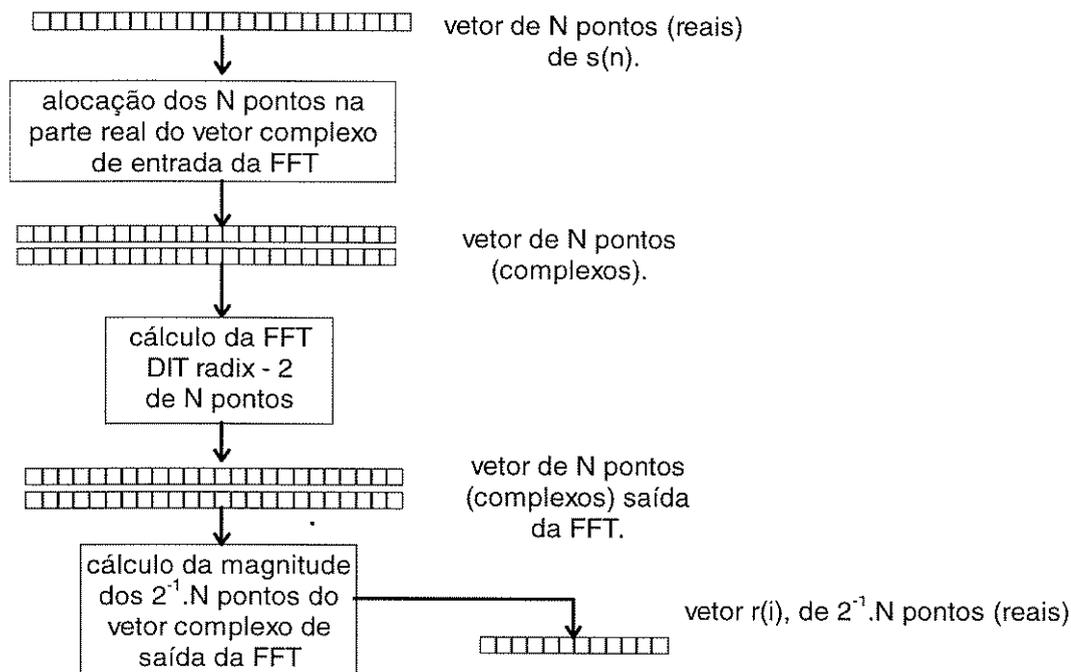


Fig. 5.4 Cálculo da FFT e do vetor controlador.

Assim a magnitude do espectro discreto dos N valores de $s(n)$ é representado pelos $2^{1/2}N$ valores reais de $r(i)$. O nome dado a este vetor (da saída do segundo módulo) é vetor controlador porque considerou-se neste projeto que a entrada deve exercer um papel de controle sobre as operações timbrais. O som de entrada não é aquele que vai necessariamente ter o seu timbre operado, mas vai controlar as operações timbrais sobre um novo espectro. Isto diminui em muito o processamento exigido para a primeira parte do sistema, que torna viável a possibilidade do tempo-real.

O tempo de processamento da FFT de N valores e o cálculo da magnitude de $2^{1/2}N$ valores é o principal tempo de atraso da parte de entrada do sistema, sendo os outros atrasos praticamente insignificantes. O atraso total desta primeira parte do sistema, ou seja, desde a entrada de $s(t)$ até a saída de $r(i)$, é dado por τ_1 .

5.2 O processamento das operações timbrais

A segunda parte do sistema é composto basicamente por três elementos: operador timbral, controle externo e matriz harmônica. Estes compõem o terceiro módulo do sistema. As operações timbrais são executadas pelo OT que recebe três entradas: o vetor $r(i)$, também chamado de vetor controlador; a entrada de dados do controle externo com a lista de operações timbrais, e/ou seus parâmetros; e leitura dos dados da matriz harmônica, caso seja solicitado pela operação timbral. Neste módulo tem-se as seguintes convenções:

- ve: vetor do espectro do som com $2^{1/2} \cdot N'$ valores reais.
- ot($p_1 \dots p_m$): operação timbral, e seus parâmetros "p".
- h: número máximo de ot por ciclo de operação.
- ov[ve]: operação vetorial (transformação) executada em ve.
- q: número máximo de ov[ve] por ciclo.

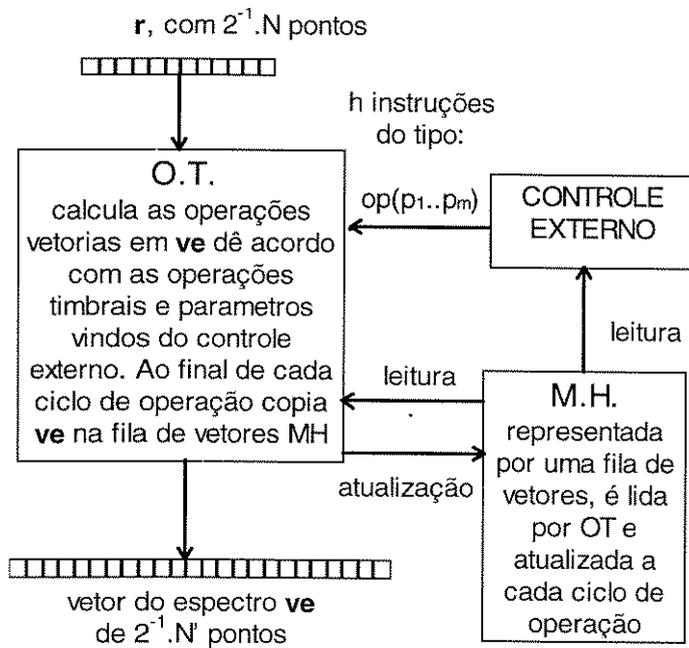


Fig. 5.5 Estrutura básica do operador timbral.

O operador timbral recebe, a cada ciclo de operação, as entradas: $r(i)$ e $ot(p_1, p_2, \dots, p_m)$. Com base nesses dados o OT executa recursivamente a lista dos "q" $ov[ve]$, do tipo:

$$ov_q[ov_{q-1}[ov_{q-2}[\dots ov_1[ve]\dots]]].$$

Se for solicitado por uma instrução ot , o OT lê dados de $MH(i,j)$. Após estes cálculos o OT atualiza MH, ou seja, copia ve na sua entrada. Assim se encerra um ciclo de operação de OT. Para um ciclo de instrução do OT, pode-se estabelecer o seguinte algoritmo:

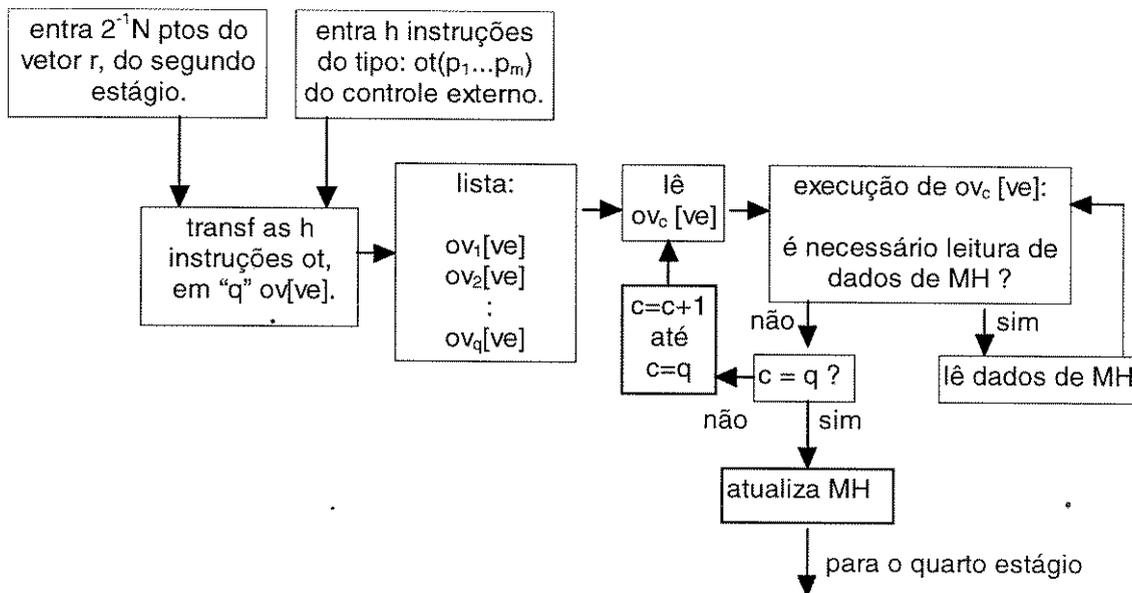


Fig. 5.6 Um ciclo de operação do operador timbral.

Observa-se que alguns dos processos descritos acima podem ser executados em paralelo, tais como: entrada de $r(i)$ e $ot(p_1...p_m)$, e execução dos $ov[ve]$. Isto incita a possibilidade de utilização de multiprocessamento do OT. O ciclo de instrução tem uma duração fixa. Isto implica num limite de operações timbrais por ciclo. Caso não haja nenhuma entrada em OT durante um ciclo, mesmo assim deve-se manter o tempo do ciclo, como um intervalo de atraso constante, que este módulo acrescenta ao atraso total de processamento do sistema.

A matriz harmônica, como foi vista, é uma fila de vetores, do tipo *last in first out*, que a cada ciclo é deslocada de uma linha para baixo. Em cada ciclo o CONTROLE tem acesso de leitura, para qualquer um dos elementos de MH. Desse modo o operador timbral pode fazer operações com base no passado do som transformado pelo sistema. Um fato importante do acesso a MH pelo CONTROLE é a possibilidade de observação da superfície sonora, *on-line*.

Diferente do vetor controlador, o vetor do espectro deve possuir uma excelente representação do espectro do som. Isso porque é nele que está contida toda a informação do espectro do som de saída. Portanto ve tem um grande número de valores, suficientes para representar os harmônicos do som dentro da faixa audível, ou seja, até 20.000 Hz. e com uma boa resolução no eixo da frequência, que corresponde a um valor de bin abaixo de 1,5 Hz. (ver capítulo 2).

O tempo total de processamento dessa parte coincide com o ciclo de operação do OT. Como foi dito, este é fixo. Seu valor é representado por τ_2 .

5.3 O operador timbral e a saída analógica.

A terceira parte do sistema possui dois módulos, respectivamente: o quarto e quinto módulo do sistema. O quarto módulo é o cálculo do vetor do som transformado. O quinto módulo é o pós-processamento e conversão D/A. Para esses dois últimos módulos, são dadas as seguintes convenções.

Quarto módulo:

vst: vetor do som transformado, de N' valores.

Quinto módulo:

Fs': taxa de amostragem de saída.

ov: taxa de entrelaçamento das seqüências de saída do sistema.

f_H' : limite máximo em freqüência do espectro de $s'(t)$.

$s'(t)$: som analógico de saída do sistema.

N' : valores utilizados de vst , em cada janelamento.

O quarto módulo possui duas fases de processamento: montagem do vetor de entrada da IFFT e cálculo da IFFT.

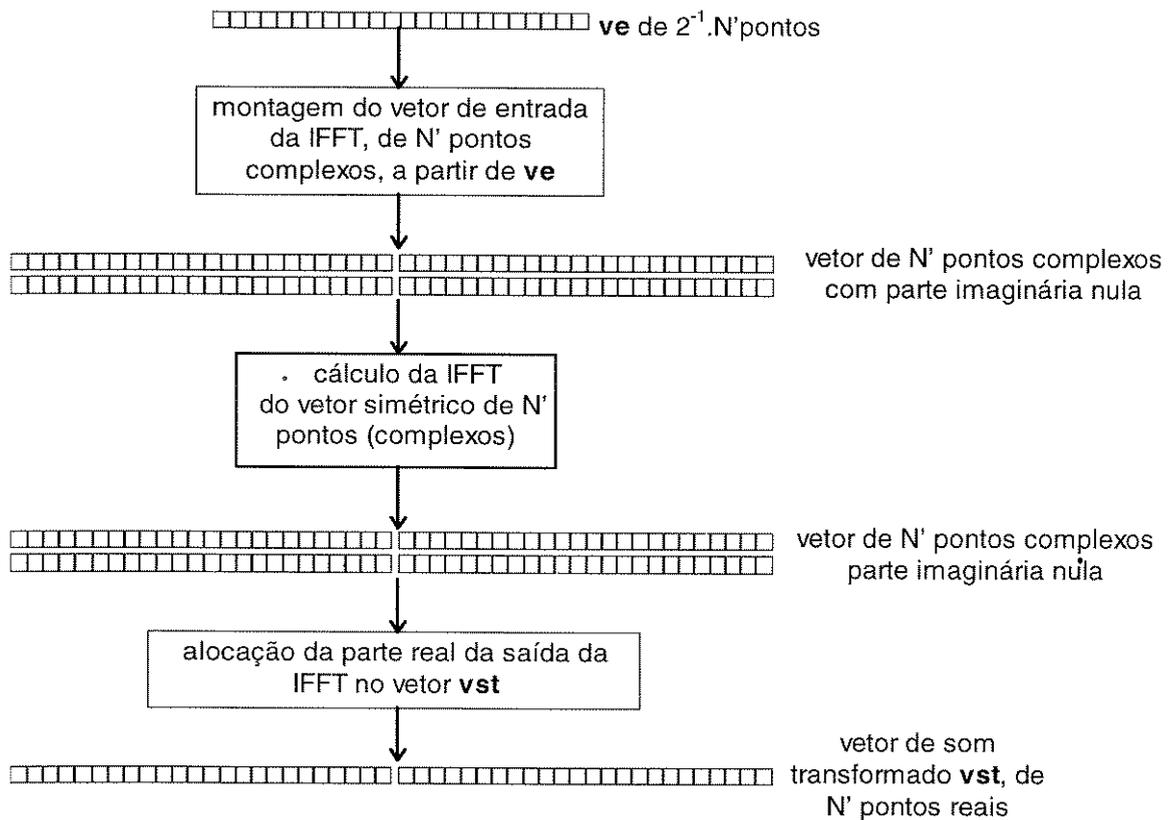


Fig. 5.7 O cálculo da IFFT e a montagem do vetor do som transformado.

A primeira fase do quarto módulo, ou seja, a montagem do vetor de entrada para IFFT, monta um vetor de N' valores complexos a partir dos dados dos $2^{-1} \cdot N'$ valores reais de ve , saída do terceiro módulo. Este novo vetor é simétrico em relação ao seu ponto mediano e possui uma parte imaginária nula. É como se os harmônicos representados por ve tivessem todos fases nulas e magnitudes consequentemente iguais a parte real de ve . Esse vetor deve ser simétrico afim de gerar uma saída da IFFT de números reais, ou seja, também com a parte imaginária nula. A segunda fase, cálculo da IFFT, aplica um algoritmo bastante parecido com o da FFT. A principal diferença, que implica em mais processamento, é a divisão de todos os valores do vetor real de saída de N' valores pela constante N' (fator de escala). Nota-se que tanto a saída da FFT quanto a da IFFT não foram normalizadas, pois está

se processando trechos de som que são posteriormente remontados. A normalização implicaria numa distorção (achatamento do nível dos harmônicos) durante o processo de transformação sonora.

Observou-se no segundo módulo que no algoritmo da FFT (e conseqüentemente o da IFFT) o valor do intervalo de tempo, respectivo aos N valores da entrada, no domínio do tempo, é o inverso do valor da resolução da saída, no domínio da freqüência. Isto foi resumido na expressão: $bin=(Fs/N)=(1/jt)$. Como ve possui um grande número de valores (N grande) e uma grande resolução na freqüência (bin pequeno) o seu valor equivalente ao intervalo de tempo que o vst representa é grande (jt grande). No entanto é imperativo que o mesmo intervalo de tempo de som que entra no sistema seja o mesmo intervalo de tempo de som que saí, para que se tenha operação em tempo-real. Uma maneira de contornar este problema é apresentada no quinto módulo.

O quinto e último módulo do sistema é formado pelas quatro fases: janelamento do vetor do som transformado, entrelaçamento, conversão D/A e filtragem do som analógico de saída.

Como foi dito acima, vst possui um grande comprimento, que eqüivale a uma grande duração no tempo de som. Durante toda essa duração de vst, tem-se o mesmo espectro representativo, dado por ve. É de se esperar então que o som contido em vst, durante toda a sua duração no tempo apresente os mesmos harmônicos componentes, nas mesmas magnitudes e nas mesmas posições no eixo da freqüência, em suma um som complexo porém contínuo (sem nenhuma variação). Desse modo qualquer trecho no tempo, de toda a duração de vst, tem o mesmo timbre. Não há informação timbral nova ao longo de sua duração. Assim pode-se desprezar toda a duração excessiva de vst, separando apenas um número suficiente de valores para que sua duração no tempo eqüivalha a duração do trecho de som de entrada do sistema. Este trecho de N'' valores de vst é então janelado através da multiplicação ponto a ponto com outro vetor de mesmo comprimento. Este outro vetor possui armazenada a seqüência de uma

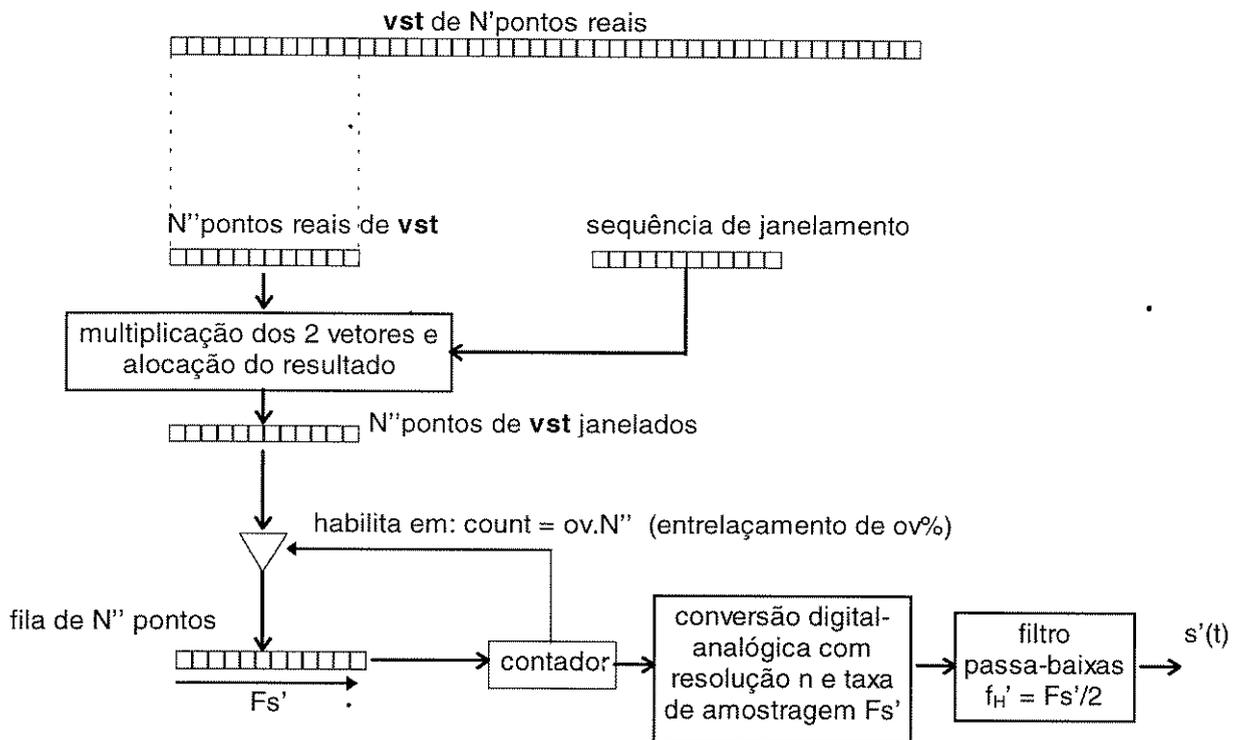


Fig. 5.8 O pós-processamento e a saída do som analógico transformado.

função de janelamento. Nas simulações feitas no capítulo quatro utilizou-se a janela *Hanning*, por apresentar melhor resultados no entrelaçamento dos vetores, que é o passo seguinte. Essa janela de N'' valores de vst é entrelaçada com as suas janelas vizinhas no tempo através da soma ponto a ponto entre estas e transferência desses valores a uma fila que os leva ao conversor D/A, numa taxa de Fs'

valores/segundo. A taxa F_s' está relacionada ao comprimento de ve . O último ponto de ve representa o último harmônico (de maior frequência) que o som de saída pode conter. Assim, como se viu no teorema da amostragem, F_s' deve ser o dobro do valor do último harmônico, representado pelo comprimento de ve , ou $[ve]$. Assim $[ve]=F_s/2$ e $[ve]$ corresponde a $f_H \geq 20.000$ Hz. Estes valores são devidamente dimensionados adiante.

O atraso total para a terceira parte do sistema é basicamente dado pelos cálculos da montagem do vetor de entrada e pelos cálculos da IFFT. O valor total do atraso desta parte é representado por τ_3 .

5.4 Visão geral do sistema.

O sistema recebe como entrada o som analógico que é fracionado em pequenos trechos e transformado em vetores de envoltórias de espectros. Estes controlam operações timbrais que dão origem a um novo e bem maior espectro. Este espectro, com grande quantidade de harmônicos é novamente transformado em som e posto na saída do sistema. Tudo com um atraso de tempo abaixo do que pode ser percebido pela audição. Uma vez que existe som constantemente entrando e som constantemente saindo, os processamentos de cada parte ocorrem simultaneamente, para diferentes trechos do som. Assim o atraso total do sistema não pode ser simplesmente a somatória do atraso de cada uma das três partes τ_1 τ_2 e τ_3 . Tem-se também que considerar que vários dos cálculos executados na segunda parte ocorrem paralelamente. O atraso total de processamento é chamado aqui de **atp**. A seguir tem-se uma ampla discussão sobre o **atp**, visto que para um sistema que trabalha em tempo-real,

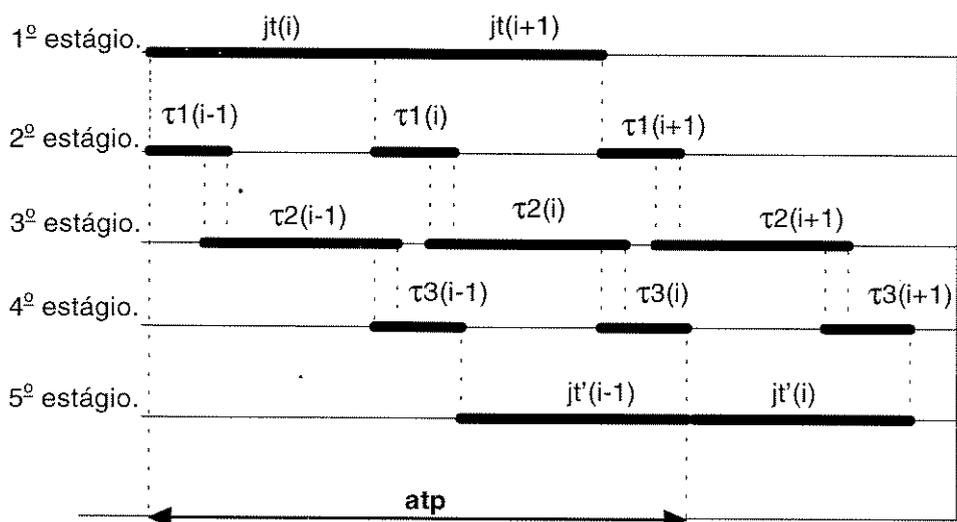


Fig. 5.9 Diagrama de tempos do processamento do sistema.

este é um fator fundamental.

Observa-se que os maiores processamentos deste sistema estão relacionados aos cálculos da FFT, da IFFT e com as operações timbrais. Existe então a necessidade de utilização de processadores digitais de sinal que processem estas partes simultaneamente, e em tempo hábil. Estes processadores são chamados de DSPs e podem ser programados para a execução de tarefas tais como as tratadas aqui. Considera-se a seguir o diagrama dos tempos de processamento de cada módulo, levando em conta a sua execução por um DSP.

O sistema possui 5 módulos. O tempos de processamento mais importantes são os relacionados aos DSPs. Assim tem-se o seguinte diagrama de tempos:

As linhas grossas representam o tempo de processamento total de uma seqüência de valores de $s(n)$, em cada módulo. No primeiro módulo, tem-se a conversão A/D de $s(t)$ numa dada taxa de amostragem F_s . Cada seqüência de N valores de $s(n)$ representa um vetor de entrada para o módulo seguinte, onde a FFT é calculada. Este vetor é chamado de janela e equivale no tempo a jT segundos. No segundo módulo tem-se o cálculo da FFT e o cálculo da magnitude da primeira metade dos valores de saída da FFT, ou seja, $2^{-1} \cdot N$ valores. Este será dado na forma de um vetor de elementos reais, cada um deles representando a magnitude de um harmônico componente do $s(n)$. O tempo necessário para todo o processamento deste módulo, ou seja: transferência do vetor janela para a entrada da FFT, cálculo da FFT, cálculo da magnitude dos $2^{-1} \cdot N$ valores da FFT (vetor r) e sua transferência para o próximo módulo; equivale a τ_1 . O terceiro módulo equivale ao cálculo das operações timbrais sobre os vetores dos espectros. O tempo τ_2 é a somatória dos tempos de transferências de dados (da entrada e para a saída), leitura de dados da MH e cálculo do conjunto de operações vetoriais sobre o vetor do espectro, que juntas compõem a operação timbral. Vale lembrar que $\tau_2 = (\Delta op + \Delta w)$, onde: Δop = tempo da operação timbral (variável) e Δw = tempo de espera (variável), no qual a soma de ambas mantêm τ_2 fixo. O quarto módulo representa a transferência de dados e o cálculo da IFFT do vetor do espectro. O seu tempo total de processamento equivale a τ_3 . A intercessão entre os tempos de processamento dos módulos vizinhos representa o tempo de transferência de dados (vetor) entre esses estágios. Sem levar em conta os

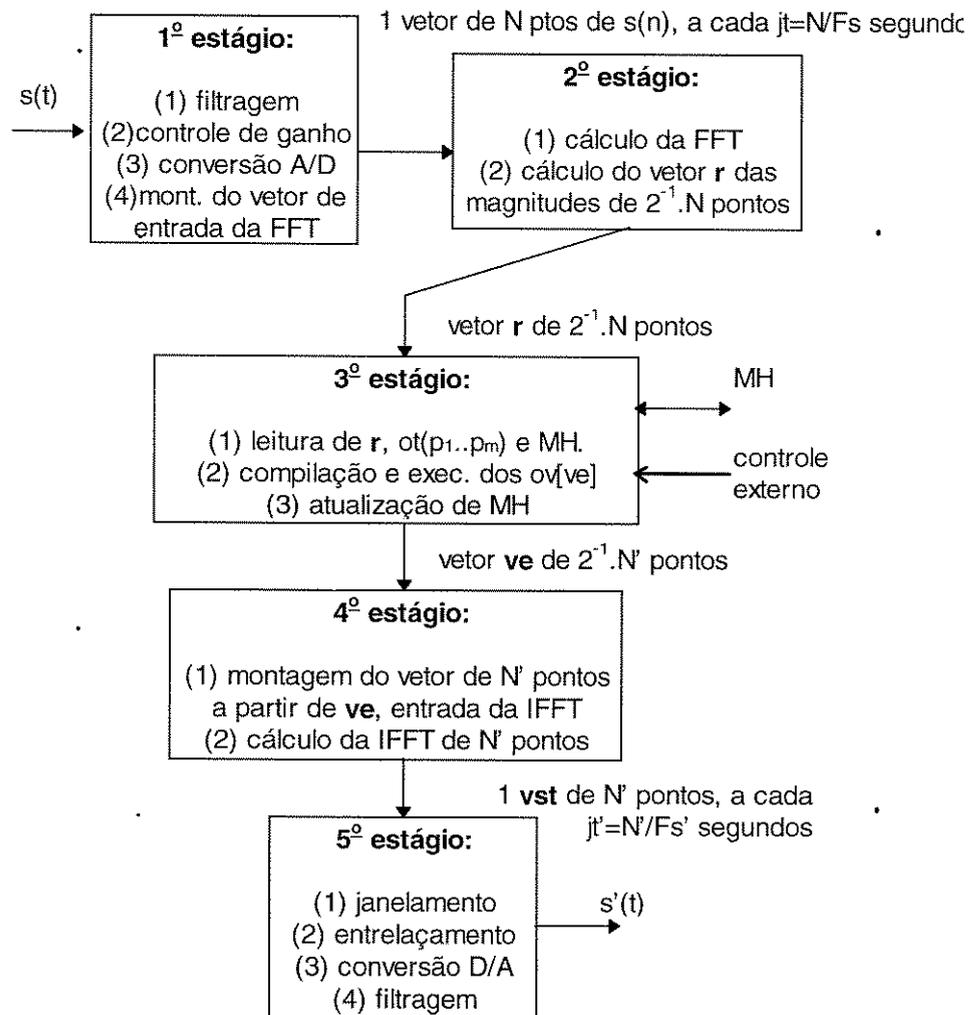


Fig. 5.10 Diagrama esquemático dos módulos do sistema.

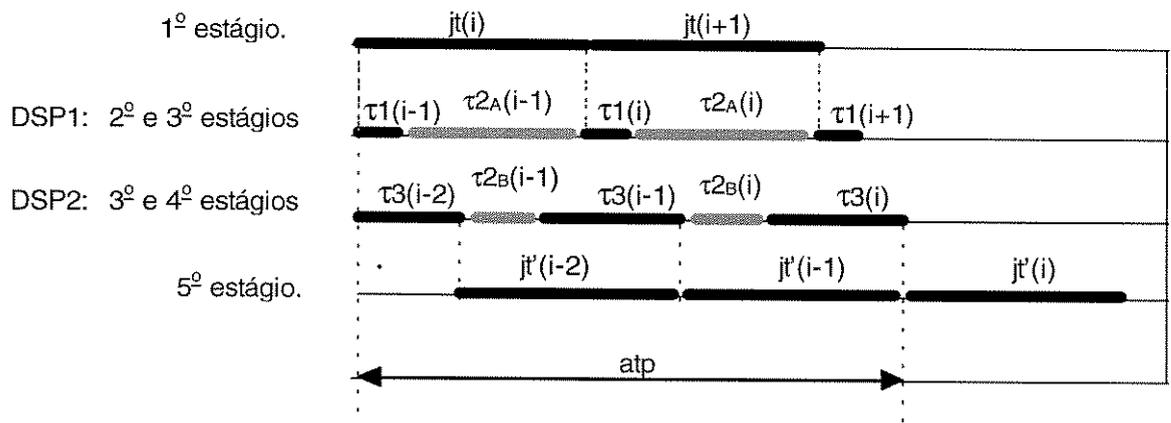
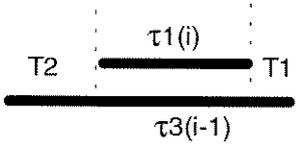


Fig. 5.11 Diagrama de tempos do processamento do sistema, para 2 processadores.

tempos de transferências de dados, o atraso total de processamento: $atp = (jt + \tau1 + \tau2 + \tau3 + jt')$. Para que o sistema seja em tempo-real, $atp \leq 50ms$. No diagrama de tempos observa-se que existem momentos em que $\tau1$, $\tau2$ e $\tau3$ ocorrem simultaneamente. No entanto, vê-se que existem tempos ociosos, especialmente para os módulos do cálculo da FFT e IFFT, ou 2 e 4, apesar desses cálculos acontecerem simultaneamente, durante alguns intervalos de tempo. Se o tempo ocioso entre os cálculos do 2º e 4º módulo for utilizado para os cálculos do 3º módulo então elimina-se os tempos ociosos nos DSPs bem como reduz-se de 3 para 2 o número de processadores.

Os cálculos referentes ao 3º módulo (OT) são feitos simultaneamente pelos dois processadores, onde $\tau2_A = \tau2_B$. No início de $\tau2_A$ (antes de $\tau2_B$) tem-se a transferência de dados do 2º para o 3º módulo (FFT para OT), cujo tempo é representado por T1. No final de $\tau2_A$ (depois de $\tau2_B$) tem-se a transferência de dados do 3º para o 4º módulo (OT para IFFT), ou T2. Como a quantidade de dados transferência de dados do OT para a IFFT é maior do que da FFT para o OT, tem-se que $T2 > T1$.



O tempo total de atraso do sistema para este caso é: $atp = (jt + \tau1 + T2 + jt')$. Como, para este projeto $jt = jt'$, então: $atp = 2.jt + \tau1 + T2$. O tempo de processamento do 3º módulo torna-se: $\tau2 = \tau2_B$, com o processamento simultâneo dos dois DSPs. Observa-se que um fator de aumento do atp é dado pela transferência de dados de um módulo para outro, ou seja T1 e T2. Como o sistema trata com vetores de grandes dimensões, estes atrasos podem impedir o processamento em tempo-real. Uma maneira de se evitar esses atrasos é evitar a transferência de dados, ou seja o seu deslocamento desnecessário. Sem estes atrasos o diagrama dos tempos do sistema fica:

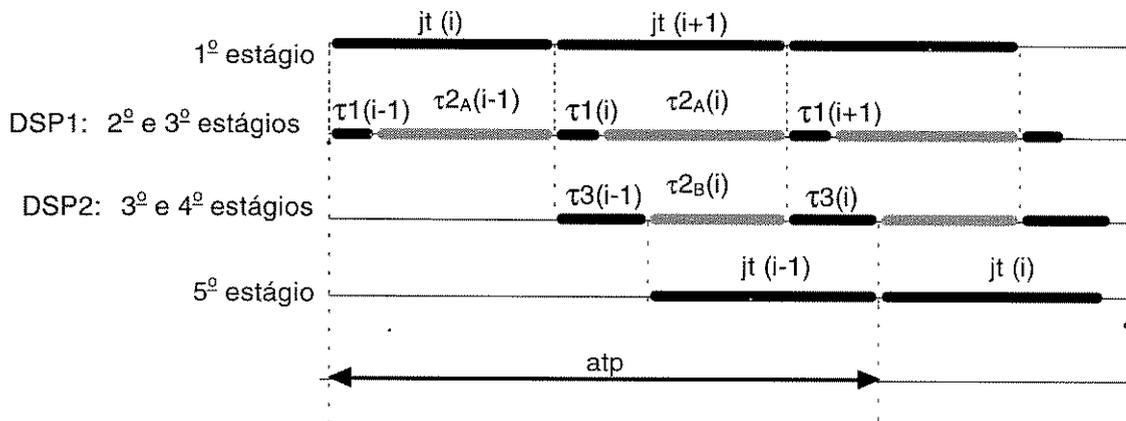


Fig. 5.12 Diagrama de tempos do processamento do sistema evitando-se transferências de dados desnecessárias.

Deste modo o valor de atp fica sendo: $atp = (jt + \tau_3 + jt) = (2 \cdot jt + \tau_3)$. O cálculo do quarto módulo passa a ser construtivo. No entanto, como é visto adiante, este pode ser executado em intervalo suficientemente pequeno para existir a possibilidade de tempo-real do sistema. A duração das janelas de entrada e saída de som, jt e jt' , são iguais, uma vez que o sistema completo deve trabalhar sincronamente.

Com base no que foi analisado até aqui, pode-se estabelecer o seguinte dimensionamento dos parâmetros do sistema:

1º módulo: Filtragem passa-faixa: $f_L = 20$ a $f_H = 8.000$ Hz.
 Controle de ganho: de acordo com a entrada do conversor A/D.
 Conversor A/D: taxa de amostragem: $F_s = 2^{14} = 16.384$ Hz.
 resolução: $n=16$ bits (equivalente a SQNR=96 dB).
 Vetor de entrada da FFT: $N = 2^8 = 256$ valores, equivalendo a:
 $jt = (N/F_s) = 1/64 = 15,6$ ms.

2º módulo: FFT de $N = 256$ valores.
 Cálculo dos $2^{-1} \cdot N$ valores do vetor de magnitudes $r(i)$
 $[r] = 2^{-1} \cdot N = 128$ valores.
 $bin = 64$ Hz.

3º módulo: Dimensionamento do vetor do espectro ve , de $2^{-1} N'$ valores:

Considerando-se os conceitos tratados no capítulo 2, tem-se que:

- (1) O maior valor de freqüência representado por ve , equivalente ao seu último ponto, deve respeitar a: $f_H' \geq 20$ K.Hz
- (2) A taxa de amostragem do conversor D/A deve ser $F_s' = 2 \cdot f_H'$
- (3) O comprimento de ve deve ser do tipo: $2^{-1} N' = 2^v$.
- (4) A resolução do eixo da freqüência deve ser $bin' = F_s'/N' \leq 1,5$ Hz.

Chegou-se aos seguintes resultados:

$F_s' = 44.096$ Hz onde $f_H' = 22.048$ Hz
 $N' = 2^{15} = 32.768$ valores
 $bin' = F_s'/N' = 1,34$ Hz.
 $[ve] = 2^{-1} \cdot N' = 16.384$ valores.

4º módulo: Montagem do vetor de $N' = 2^{15} = 32.768$ valores
 Cálculo da IFFT de $N' = 32.768$ valores.

comprimento do vetor do som transformado:
[vst] = N' = 32.768 valores

5º módulo: Separação de N'' valores de vst:

Considerando-se que vst possui N' valores e a taxa de amostragem de saída é Fs', então tem-se que vst representa no tempo a duração de $jt' = N'/Fs' = 0,74$ segundos. Comparando com a duração dos vetores de entrada jt, tem-se que $jt' = 47,55 \cdot jt$. Como a duração dos vetores de entrada e saída tem que ser iguais, separa-se apenas os N'' primeiros valores de vst, onde $N'' = N'/47,55$. Assim, tem-se um novo vetor no tempo de **N'' = 689 valores**.

Janelamento: utiliza-se a função de janelamento *Hanning* (ver apêndice 1).
Entrelaçamento: os vetores de N'' valores são entrelaçados numa taxa de 50%.
Conversor D/A: **Fs' = 44.096 Hz**.
Filtragem passa-baixa: **f_H = 20.000 Hz**.

As fases de cada módulo envolvendo grande quantidade de processamento são distribuídas como tarefas para os dois processadores utilizados: DSP1 e DSP2. Assim os módulos 2, 3 e 4, são realizados conforme observado no diagrama dos tempos anterior, ou seja, módulo 2, por DSP1; módulo 4, por DSP2 e módulo 3 por ambos, simultaneamente (a maior parte do tempo). Com base nos valores que foram dimensionados acima, tem-se uma primeira aproximação dos tempos de processamento do sistema:

Segundo módulo: DSP1:

- (1) Cálculo de FFT de N = 256 valores.
- (2) Cálculo de $r(i) = (a(i)^2 + b(i)^2)^{1/2}$, para $2^{-1} \cdot N = 128$ valores.

Tempo de processamento: τ_1

Transferência de dados:

- (1) 128 valores de r(i) para cálculos do terceiro módulo
- Tempo de processamento: T1
- (2) $2^{-1} \cdot N' = 16.384$ valores de **ve** para os cálculos do quarto módulo, feito pelo DSP2:
- Tempo de processamento: T2

Quarto módulo: DSP2:

- (1) Cálculo do vetor de N' = 32.768 valores, montado a partir dos valores de **ve**, através da repetição simétrica dos seus elementos (transferência de N' valores). Alocação na parte real do vetor complexo. A parte imaginária é um vetor de N' valores nulos.
- (2) Cálculo da IFFT de N' valores.

Tempo de processamento: τ_3

Terceiro módulo: DSP1 e DSP2.

Apenas o DSP1:

- (1) Leitura das h instruções do controle externo: $ot(p_1 \dots p_m)$.
- (2) Execução dos $ov[ve]$, a partir de r(i), dos h $ot(p_1 \dots p_m)$ e dos dados da MH (caso seja solicitado por uma ot).
- (3) Elaboração da lista das q operações vetoriais ov sobre **ve**.

Tempo de processamento: τ_{2A}

Apenas o DSP2:

- (1) Execução seqüencial das q operações vetoriais $ov[ve]$.
- (2) Atualização de MH com o novo ve (transferência de $2^{21} \cdot N' = 16.384$ valores) .

Tempo de processamento: τ_{2B}

Observa-se que não há transferência da lista de ov para o DSP2, que a executa. Isto porque a execução (leitura) dessa lista é feita em paralelo com o DSP1.

Tempo total de processamento do terceiro módulo: $\tau_2 = \tau_{2B}$.

5.5 Projeto do sistema utilizando o DSP-96002

Para efeito de projeto utiliza-se como processador o DSP 96002 da MOTOROLA. Este possui as seguintes características gerais [Motorola]:

- 1) Ponto-flutuante, em duas precisões: simples (SP) com 32 bits (8 de expoente, 24 de mantissa) ou estendida (SEP) com 44 bits (11 de expoente e 32 de mantissa).
- 2) 16,5 Mips p/ clock = 33MHz.
- 3) 49,5 MFLOPS máximos p/ clock = 33MHz.
- 4) Multiplicação paralela de 32 x 32 bits, em 1 ciclo.
- 5) 2 memórias RAM independentes, para dados, de 512 x 32 bits.
- 6) 2 memórias ROM independentes, para dados, de 1024 x 32 bits.
- 7) permite expansão externa de memória de dados para 2 de 2^{32} x 32 bits.
- 8) memória RAM para programa, de 1024 x 32 bits.
- 9) permite expansão externa de memória de programa para 2^{32} x 32 bits
- 10) duas portas idênticas para expansão externa de memória.
- 11) duas interfaces paralelas de 32 bits, para MPU/DMA.
- 12) dois controladores de canal DMA.

Para este processador optou-se por utilizar uma estrutura de memória externa compartilhada, na montagem do sistema.

Consultando a tabela das análises de performance do DSP 96002 [Sohie], tem-se os seguintes valores de número de ciclos por instrução (IC) para as seguintes operações, considerando-se 16,5 milhões de instruções / segundo (MIPS) num clock = 33 M.Hz.:

(1) **O cálculo da FFT.** Existem vários programas para o cálculo da FFT, desenvolvidos para este DSP. No *bulletin board* da Motorola, Dr.Bub, encontram-se diversos algoritmos desenvolvidos para o DSP 56000/1, que são compatíveis com o DSP 96002. De um modo geral, o cálculo de uma FFT de 1024 valores complexos é feito em cerca de 20.931 ciclos de instrução. A execução de uma *butterfly* da FFT, com entrada complexa tem $IC = 4 \cdot N$. O número de cálculos para a FFT (e IFFT) cresce de acordo com $5 \cdot Fs \cdot \log_2 N$, Assim tem-se:

para $N = 256$ valores complexos, o número de cálculos é 40.Fs, onde $IC = 16.744$

para $N = 1.024$ valores complexos, o número de cálculos é 50.Fs, onde $IC = 20.931$

para $N = 32.768$ valores complexos, o número de cálculos é 75.Fs, onde $IC = 31.396$

No caso da IFFT, o algoritmo da *butterfly* é praticamente o mesmo (diferindo apenas o sinal do expoente de W). A diferença importante para estes cálculos é que os valores do vetor de saída do algoritmo da FFT modificado para calcular a IFFT devem ser divididos por $1/N$, que aumenta o tempo de processamento da IFFT.

(2) **Operações vetoriais.** Para operações de soma e multiplicação entre os valores de vetores reais de N valores, do tipo: $v=v+s$, $v=v*s+v$, $IC = 2 \cdot N$. Para valores complexos, $IC = 4 \cdot N$.

(3) **Transferência de dados.** Para a transferência de N valores reais de uma posição de memória para outra, tem-se: $IC = 4.N$.

O intervalo de tempo para o processamento de cada uma dessas operações é obtido fazendo-se: $IC / MIPS$. Assim, tem-se o tempo de processamento para cada tarefa executada pelos DSPs.

Segundo módulo: DSP1:

Do primeiro módulo tem-se uma seqüência de vetores reais de N valores alocados na memória X. Na memória Y tem-se um vetor nulo, de N valores, que é a parte imaginária do vetor complexo de entrada da FFT. O resultado da FFT é alocado na mesma posição de X, mas em posição diferente de Y, afim de se preservar o vetor que representa parte imaginária nula, para o próximo vetor de entrada. O cálculo de $r(i)$ é executado e seu resultado é alocado novamente na mesma posição em X.

(1) Cálculo da FFT de N = 256 valores.

Montagem do vetor complexo de N valores, entrada da FFT.

transferência de N valores para X: $4.N$

Cálculo da FFT de N valores: $1,015 \text{ ms}$

total: $4.N + 1,015\text{ms} = 1,08\text{ms}$

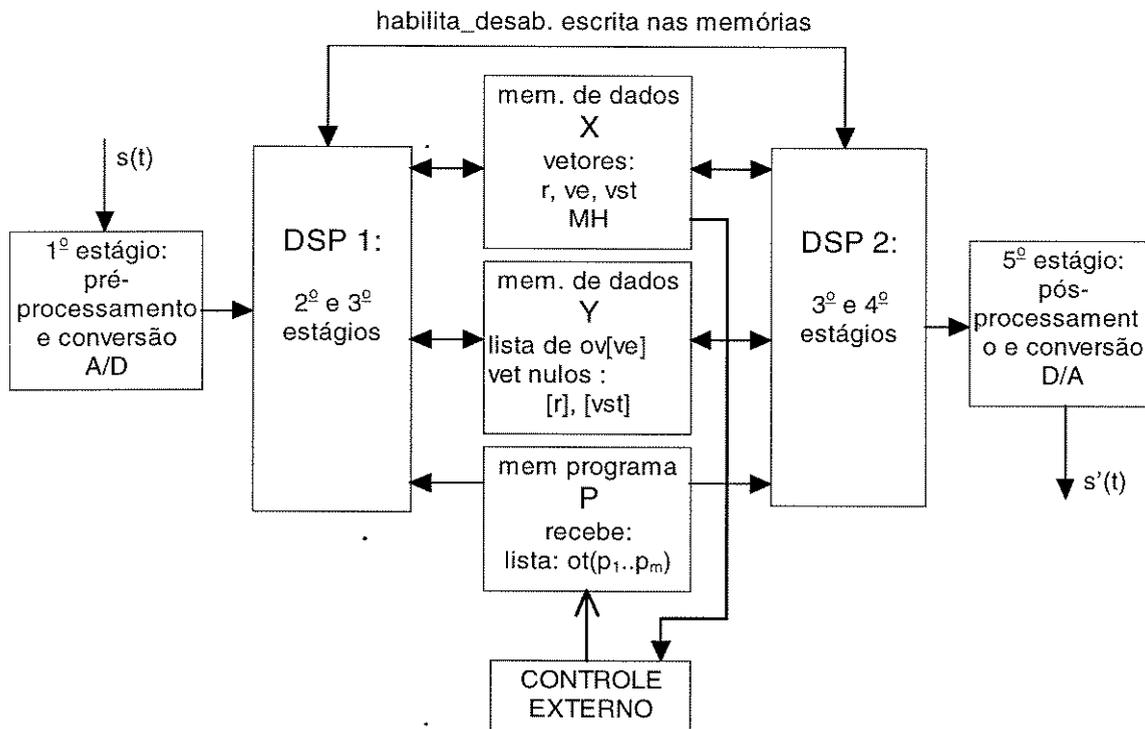


Fig. 5.13 Diagrama do sistema com dois DSPs.

(2) Cálculo do vetor $r(i)$, de $2^{-1}N = 128$ valores

Para cada ponto complexo: $x(i) + j.y(i)$, tem-se: $r(i) = (x(i)^2 + y(i)^2)^{1/2}$, onde $i = 1, 2, \dots, 2^{-1}N$

- a) $x(i) = x(i)*x(i)$ $2.N$
- b) $y(i) = y(i)*y(i)$ $2.N$
- c) $x(i) = x(i)+y(i)$ $2.N$

$$d) x(i) = (x(i))^{1/2}$$

12.N

total: 18.N = 0,14ms

Saída: $2^{-1}N$ valores de $r(i)$ em X.

$$\tau_1 = 1,14\text{ms} + 0,14\text{ms} = \mathbf{1,23\text{ ms}}$$

Quarto módulo: DSP2:

O quarto módulo inicialmente monta o vetor de entrada da IFFT. Como anteriormente, a parte real está na memória X, e a parte imaginária está em Y. O processamento da montagem implica basicamente na transferência de dados dos valores de ve , para criar o novo vetor simétrico, de N' valores. A seguir vem o cálculo da IFFT

1) Transferência de $2^{-1}.N' = 16.384$ valores de ve para montar vetor simétrico

$$4.(N'/2) = 65.536$$

total: 3,97 ms

2) Cálculo da IFFT de N' valores:

cálculo de FFT de N' valores

$$31.396$$

multiplicação dos N' valores por cte escalar ($1/N'$)

$$2.N' = 65.536$$

total: 5,87 ms

$$\tau_3 = \mathbf{9,84\text{ ms}}$$

Terceiro módulo: DSP1 e DSP2:

O terceiro módulo realiza o processamento do operador timbral do sistema. Seu tempo de processamento é complementar ao tempo de processamento dos outros módulos (segundo e quarto) uma vez que o processamento total de todos esses três módulos é feito por dois DSPs. Observando-se o diagrama de tempos do sistema, dado na figura 5.12, tem-se que os tempos de processamento para o terceiro módulo ficam sendo:

Para o DSP1, $\tau_{2A} = \mathbf{14,3\text{ ms}}$

IC = 235.950 ciclos de instruções.

Para o DSP2, $\tau_{2B} = \tau_2 = \mathbf{5,76\text{ ms}}$

IC = 95.040 ciclos de instruções.

O processamento do terceiro módulo é dado, basicamente, pelas tarefas descritas nos itens abaixo:

- (a) leitura dos $h\ ot(p_1, \dots, p_m)$ do controle externo .
- (b) execução da lista de $ot()$ em "q" operações $ov[ve]$
- (c) execução dos $ov[ve]$, onde $[ve] = 16.384$ valores.
- (d) atualização de MH

Considerando que as h instruções de operação timbral são enviadas à memória de programa P pelo controle externo, não existe o tempo de processamento do item (a). Também, se as operações $ov[ve]$ forem feitas sobre a posição da memória referente à entrada da fila de vetores MH, então não se tem o tempo de processamento do item (d). O tempo de transferência de dados para a leitura de MH não é aqui tratado pois depende da instrução $ot()$ e pode variar muito em seu processamento (indo desde a transferência de apenas um ponto, até todo um vetor de $[ve] = 16.384$ valores). Os tempos de processamento relevantes nos cálculos do terceiro módulo são os referidos nos itens (b) e (c). Convencionou-se anteriormente que o DSP1 faz os cálculos do item (b), em paralelo com o DSP2, que

calcula o item (c). Assumindo operações vetoriais que necessitem de IC próximos a 2.n, (do tipo de operação vetorial encontrada na tabela de análise de performance do DSP) tem-se a possibilidade de uma lista de operações vetoriais $ov[ve]$ de $q = 3$ $ov[ve]$, somente para o DSP2. Utilizando o processamento do DSP1 em paralelo, esse número sobe, pelo menos, para $q = 6$ $ov[ve]$, a cada ciclo de operação do terceiro módulo.

O processamento total do sistema:

Considerando-se que o tempo da janela de entrada é $jt = N/Fs = 256/16.384 = 15,6$ ms, e que a janela de saída é forçada a ter esse mesmo valor. Tem-se que o atraso total de processamento do sistema é:

Atraso total de processamento: $atp = (2.jt + \tau_3) = 41$ ms.

Foi visto na figura 5.12 que para esta montagem, com dois DSPs de memórias compartilhadas, tem-se as seguintes relações:

Intervalo de tempo da janela: $jt = (\tau_1 + \tau_{2A}) = (\tau_3 + \tau_{2B})$.

O intervalo τ_{2A} e τ_{2B} são complementares aos valores de jt , τ_1 e τ_3 . De acordo com o tempo ocioso do processamento do segundo e do quarto módulo, foi feito os cálculos para o processamento do terceiro módulo. Uma grande vantagem deste é a possibilidade do processamento paralelo dos dois DSPs que está disponível. No atual módulo é impossível precisar as possibilidades de realizações de operações timbrais em tempo hábil de processamento. No entanto, observou-se que esta possibilidade é perfeitamente factível. A implementação futura deste sistema dirá qual a real extensão do número de operações timbrais possíveis de serem realizadas por este sistema, em tempo-real.

Não foram levados em consideração os tempos de processamento do primeiro e quinto módulo. Isto porque considera-se que estes estágios possuem um tempo de atraso muito abaixo daqueles encontrados nos módulos intermediária do sistema. No entanto, no quinto módulo tem-se o janelamento e entrelaçamento dos vetores de $N'' = 689$ valores. Estes são dados respectivamente por uma multiplicação e uma soma de 689 valores. Em termos de processamento com um DSP 96002, este corresponderia a IC = $2.N'' + 2.N'' = 2.756$, cujo tempo de processamento seria: 0,16 ms. Como o sistema está com $atp = 41$ ms (uma folga de 9 ms para o processamento total ainda ser considerado em tempo-real) pode-se perfeitamente criar uma nova estrutura de alocação de tempos de processamento que preveja o processamento do quinto módulo.

5.6 Considerações finais

O sistema foi dimensionado para gerar timbres através do processamento digital de espectros discretos, representados por vetores. A representação desses espectros nos vetores é feita de modo linear, ou seja, linearmente espaçados na frequência e com níveis de amplitude também linearmente distanciados entre si. Considerando a percepção humana, faremos algumas considerações em relação ao dimensionamento dos eixos de frequência e amplitude:

Para se representar devidamente uma oitava de variação de frequência, dentro do intervalo de 500 a 4.000 Hz devem-se utilizar 240 valores espaçados na taxa de $2^{1/240} \cdot f_N$. Tanto abaixo de 500 Hz, quanto acima de 4.000 Hz, a percepção auditiva decai. No entanto, pode-se considerar o decaimento de sensibilidade apenas para as baixas frequências, para poder representar devidamente os harmônicos na

freqüência, com o menor número de valores. Um possível mapeamento dos harmônicos no eixo da freqüência é mostrado abaixo:

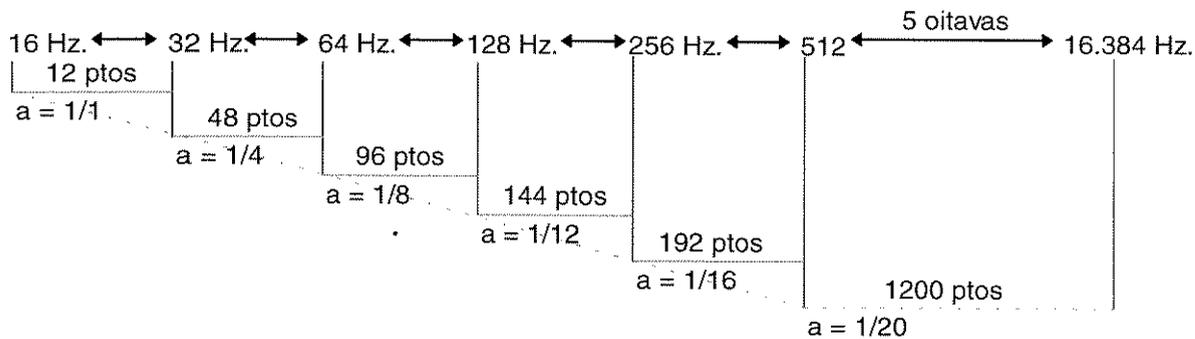


Fig. 5.14 Mapeamento dos harmônicos no eixo da freqüência em função da percepção auditiva humana.

As primeiras 5 oitavas são mapeadas de acordo com um índice crescente de sensibilidade $s = 2^{(a/12)}$ onde $2^{(1/12)}$ equivale a 1 semitom. Para a melhor sensibilidade (entre 500 e 4.000 Hz) $a = 20$ e $s = 2^{1/240}$, que equivale a $1/20$ de semitom. Tem-se assim um total de 1692 valores representando as 10 oitavas entre 16 e 16384 Hz. Este mapeamento pode ser feito por uma tabela pré-definida, que aloque os harmônicos. Os $(2^i \cdot N)$ valores de saída da FFT que representam o a magnitude do espectro estão contidos no vetor $r(i)$ e são linearmente espaçados na freqüência, em bin Hz. $r(i) = i \cdot bin$. O eixo de freqüência subjetiva é logarítmico, do tipo: $(2^{1/12})^{k/a} \cdot f_L$, onde f_L depende de a . A tabela de mapeamento é elaborada calculando-se os valores de l e k , para os quais $i \cdot bin \equiv (2^{1/12})^{k/a} \cdot f_L$. Isto é particularmente interessante para a elaboração de algumas operações timbrais, tais como aquelas que manipulam o *pitch*.

O conversor análogo-digital representa a magnitude de $s(t)$ em $s(n)$, com uma resolução de n bits. Considerando a representação PCM (*Pulse Code Modulation*) tem-se 2^n níveis de intensidade sonora, linearmente espaçados que representam níveis de som dentro de um intervalo dado por $20 \cdot \log_{10} 2^n$ dB.

Pelas curvas de Fletcher-Munson, vê-se no entanto que a percepção da intensidade sonora depende da freqüência do som. Na faixa de maior sensibilidade, a audição percebe sons de intensidade entre 0 e 120 dB. Isto equivale a conversão A/D com resolução $n = 20$ bits.

Foi visto que o ouvido percebe a variação de intensidade sonora de cerca de 10% da intensidade absoluta. que equivale á variação de intensidade de $10 \cdot \log_{10}(1,1) \approx 0,4$ dB.

Atualmente é considerado suficiente a utilização de 16 bits, equivalentes a $SQNR = 96$ dB, para uma representação razoável da amplitude sonora [Chamberlin].

A escala mais adequada para o eixo das magnitudes dos harmônicos do som em *phons*, a unidade da percepção subjetiva do som. Isto pode ser feito através da utilização dos dados contidos nas curvas isofônicas, citadas anteriormente.

Os níveis de amplitude deixariam de ser linearmente espaçados em 2^n e passariam a representar a diferença entre cada nível como 10% de variação da intensidade sonora.

O projeto de operadores timbrais deve levar em conta o caráter não-linear da percepção auditiva humana, da frequência e da amplitude, para realizarem suas transformações sonoras. O mesmo vale para a parametrização das operações timbrais pelo controle externo, que também devem ocorrer em tempo-real.

Conclusões e comentários:

Do projeto:

O objetivo deste trabalho foi estudar, dimensionar e projetar de um novo sistema de processamento digital para áudio (ADSP), de manipulação do timbre, em tempo-real, e com intervalo de tempo de retardo de processamento abaixo da percepção auditiva humana. Este sistema foi projetado para que as transformações sonoras fossem feitas “baseadas no” (e não “sobre o”) som de entrada.

O espectro do som de entrada é usado como controlador das transformações sonoras. O som de entrada é limitado em 8 KHz, amostrado em 16 bits à taxa de $F_s = 16.384$ Hz. Tem-se um vetor controlador (da magnitude do espectro do som de entrada) de 128 pontos a cada 15,6 ms. A sua resolução para o eixo da frequência é 64 Hz.

O som de saída é dado no padrão CD (*Compact Disc*), ou seja resolução de 16 bits, taxa de amostragem aproximadamente igual a 44,1 KHz, banda de frequência maior que: 20 a 20.000 Hz e resolução do eixo da frequência igual a 1,34 Hz.

As operações timbrais foram representadas como: $ot(p_1...p_m)$, onde ot é a operação timbral propriamente dita e p_m o seu m -ésimo parâmetro. Cada ot é interpretada pelo OT como uma série de operações vetoriais, chamadas de $ov[ve]$, onde ve é o vetor da magnitude do espectro do som de saída, de 16.384 pontos.

Cada ot é interpretado pelo OT como uma série de $ov[ve]$, executadas em um ciclo de operação do OT. As operações timbrais são acessadas e parametrizadas pelo controle externo.

O operador timbral, ou OT, realiza a transformação do som da saída, baseado no espectro do som de entrada. A transformação sonora é dada pela operação timbral acessada pelo controle externo. O OT realiza as seguintes tarefas à cada ciclo de operação: (1) leitura da $ot(p_1...p_m)$ do controle externo. (2) compilação e execução da série de $ov[ve]$, (3) atualização de MH. Caso seja solicitado pela operação timbral, o OT lê dados da MH, que no projeto do sistema é uma fila de vetores ve , do tipo FILO (*First In Last Out*).

Neste projeto o OT é constituído por um intervalo de tempo do processamento de dois DSP 96002. Estimou-se o tempo de duração de um ciclo de operação do OT em cerca de 14,3 ms, nos quais os últimos 5,76 ms são de processamento simultâneo nos dois DSPs. Isto equivale a $IC = 330.990$ instruções executadas por ambos DSPs, para $clock=33MHz$.

O projeto do sistema utilizando dois DSP 96002, com memórias compartilhadas, trabalha em tempo-real, com um retardo entre a entrada e a saída de 41 ms.

Das definições do sistema:

O som é a matéria-prima manipulada pelo sistema. Foi visto que ele é constituído por vibrações mecânicas que se propagam num meio material, na forma de oscilações de pressão aproximadamente periódicas, ao longo do tempo. Ele pode ser convertido em sinal elétrico representado por $s(t)$, chamado de sinal de áudio. Este por sua vez pode ser amostrado e representado por uma seqüência de números discretos, $s(n)$. O sistema processa as transformações do som de entrada através de cálculos em $s(n)$.

O timbre foi definido como: o conjunto de características sonoras perceptíveis isoladamente pela audição humana, em qualquer uma de suas três grandezas acústicas, que possibilita o reconhecimento e a discriminação de um dado som. Ele é representado na superfície sonora, cuja representação discretizada é a matriz harmônica, ou MH.

A operação timbral manipula o timbre, através da manipulação da superfície sonora. No sistema ADSP, tempo-real que foi projetado, a manipulação do timbre é feita através da manipulação dos 16.384 elementos de **ve** a cada ciclo de operação do OT (14,3 ms).

Da simulação do projeto:

O software MATLAB 4.0 foi utilizado para o desenvolvimento de programas de simulação das transformações sonoras através de operações timbrais. Uma vez que se trata de um software, estas operações timbrais não ocorrem em tempo-real.

A simulação do sistema foi feita através de programas que fazem o papel das operações timbrais, e dessa maneira, modificam os elementos contidos na MH, ao contrário do sistema real, onde as operações ocorrem em tempo-real, portanto estas manipulam a envoltória do espectro do som e armazenam o resultado na MH.

O sistema simulado realiza transformações sonoras num som de entrada, ou seja, o som de saída é o som de entrada transformado pelas operações timbrais. No sistema projetado as transformações sonoras são baseadas num som de entrada, ou seja, O som de saída tem o seu timbre elaborado a partir das informações do timbre do som de entrada.

Das possibilidades de utilização do sistema projetado:

Optou-se por projetar o sistema de transformação sonora com a manipulação do timbre baseada no espectro do som de entrada para: (1) diminuir o processamento necessário aos cálculos da magnitude do espectro da entrada, dedicando este aos cálculos do OT e da IFFT. (2) conseguir um sistema de transformação sonora em tempo-real e retardo entre a entrada e a saída abaixo de 50ms, que opere com apenas dois DSP 96002.

A entrada serve para o controle do sistema. Ela é dada em um vetor de 128 pontos, a cada 15,6 ms, com resolução do eixo da frequência de 64 Hz.

A saída tem a possibilidade de representação de 16.384 harmônicos, espaçados de 1,34 Hz, entre 0 e 22.048 Hz.

A utilização do espectro do som de entrada para o controle das transformações sonoras aparenta ser razoável. O timbre do som de saída pode ter mais de 16.000 harmônicos, que podem ser manipulados a cada 41ms (abaixo da percepção auditiva) por mais de 300.000 instruções dos DSP-96002 (em clock=33MHz).

Esta manipulação é feita com base nos 128 pontos do vetor controlador. Lembrando; um piano, que é um instrumento que possui uma das maiores extensões em representação de frequências, gera sons fundamentais entre 27 e 3.480 Hz, em 88 notas. O vetor controlador possui a representação de um som de entrada entre 20 a 8.000 Hz, em 128 pontos.

O sistema projetado para manipulação timbral baseada no espectro de um som de entrada é então um instrumento musical. Pode-se dizer, o primeiro de uma nova geração de instrumentos musicais eletrônicos. O seu controle pode ser qualquer som (com pelo menos parte do seu espectro dentro de 20 - 8 KHz). O tipo da manipulação do timbre do som da saída pode ser programado e parametrizado pelo controle externo, que é um computador. Pode-se também utilizar uma interface de controle das operações timbrais e/ou seus parâmetros. Em suma, este instrumento tem a transformação sonora e o seu controle generalizados e abertos.

Pode-se citar algumas das possibilidades (até agora vislumbradas) de utilização deste instrumento:

Composição timbral:

O instrumento permite criar intuitivamente novos timbres, através de qualquer controle escolhido pelo usuário, desde o som de um instrumento musical (como um piano) até qualquer tipo de som considerado a princípio como não-musical.

Modelamento virtual do timbre:

Através de interfaces para realidade virtual acopladas ao computador do controle externo, responsável pelo acesso e parametrização das operações timbrais, pode-se “modelar” uma transformação sonora em um som a princípio não-controlado (ex: um som gravado). Uma vez obtendo a transformação sonora desejada, esta pode ser arquivada na forma de uma nova operação timbral, e adicionada ao repertório das outras **ot** já existentes.

Performances artísticas:

Especifica-se um tipo de som para o controle do sistema (ex: voz, percussões, um instrumento musical...) e as operações timbrais desejadas. A partir disto pode-se “tocar” este instrumento, da maneira que o artista preferir. Isto porque ele trabalha em tempo-real, com um retardo inferior a 50ms.

Tratamentos e análises sonoras:

Extendendo-se o dimensionamento do espectro do som de entrada (igualando-o ao padrão da saída), este instrumento pode ser preparado para fazer análises e tratamentos sonoros inteligentes (com parametrização condicionada à um algoritmo do computador do controle externo). Entre as possibilidades, pode-se citar : correções de *pitch*, equalizações de acordo com o tipo de som, detecções de padrões sonoros, etc. Graças ao tempo-real e a programação da parametrização dos **ots**, estes processos podem ser feitos menos tempo e com menos dificuldade. No entanto, passa a existir a necessidade de um terceiro DSP 96002, devido ao aumento de cálculos do processamento da entrada.

Das limitações do projeto: .

Tanto as simulações do sistema em software quanto o projeto do sistema foram feitos sobre a premissa de uma superfície sonora representante dos padrões característicos do timbre, cujos eixos de amplitude, frequência e tempo são lineares. No entanto, sabe-se que a percepção auditiva, para a intensidade e a frequência do som, não são lineares.

As operações timbrais a serem programadas no sistema projetado devem levar em conta a natureza da audição humana para estas grandezas. A resolução e os limites máximo e mínimo de cada eixo da superfície sonora devem corresponder aos padrões subjetivos da percepção auditiva humana.

Modelamento do eixo da frequência

O sistema projetado realiza as operações timbrais através de sucessivas operações vetoriais sobre **ve**. A representação do espectro é feita pela FFT. Foi visto que a FFT mapeia vetores de R^n para R^n , e o espaçamento dos harmônicos no eixo da frequência é linear. Seria útil para este sistema a elaboração de novo algoritmo da FFT, com: (1) mapeamento de R^m para R^n , onde $m \neq n$, (2) modelamento do espaçamento dos pontos de saída.

Modelamento do eixo da amplitude

A amplitude dos harmônicos deveria ser representada de acordo com as curvas isofônicas de Fletcher-Munson, afim de adequá-las à natureza não-linear da percepção auditiva à intensidade sonora.

Adendo aos comentários:

CONVERSÃO SIGMA-DELTA

TRANSFORMADA WAVELET

Apêndice 1: A DFT e a FFT.

1 As propriedades da DFT [Weaver]

Dada uma sequência $x(n)$, de N pontos no tempo, pode-se calcular a sua DFT. Este cálculo requer N^2 adições e multiplicações complexas. Para grandes valores de N , o quantidade de operações complexas torna o cálculo da DFT inviável. As propriedades da transformada discreta de Fourier permitem a redução significativa dos cálculos complexos para sequências que satisfaçam determinadas condições especificadas. Através delas pode-se também manipular adequadamente o espectro, $X(k)$, do sinal de entrada. Estas propriedades serão mostradas abaixo sem as suas provas matemáticas, que podem ser encontradas nas referências bibliográficas [16].

Linearidade: Se $DFT[x(n)]=X(k)$ e $DFT[g(n)]=G(k)$ e se considerarmos uma nova sequência $h(n)$ onde $h(n) = a.x(n) + b.g(n)$, onde a e b são constantes, então:

$$DFT[h(n)]=H(k)=a\{X(k)\}+b\{G(k)\} \quad (1)$$

Deslocamento circular da DFT: Se $DFT[x(n)]=X(k)$, então, para $m \in [0, N-1]$:

$$DFT[x(n+m)]=X(k).W_N^{-mk} \quad (2)$$

$$DFT[x(n).W_N^{n.m}]=X(k+m) \quad (3)$$

Periodicidade: Para o par $x(n)$ e $X(k)$, ambos com N pontos, tem-se :

$$x(n)=x(n+N) \quad (4)$$

$$X(k)=X(k+N) \quad (5)$$

Diferença: Considerando o par $x(n)$ e $X(k)$, onde a diferença para ambos é, respectivamente :

$$\begin{aligned} X(k) &= X(k+1) - X(k) \\ x(n) &= x(n+1) - x(n) \end{aligned} \quad \text{então:}$$

$$X(k) = DFT[x(n). (W_N^n - 1)] \quad (6)$$

$$x(n) = DFT[X(k). (W_N^n - 1)] \quad (7)$$

Sequências pares e ímpares: Considerando o par $x(n)$ e $X(k)$, tem-se :

Se $x(n)$ for uma função *par*, o que implica em $x(n)=x(-n)$, então $X(k)$ também será *par*.
Se $x(n)$ for uma função *ímpar*, ou seja, $x(n)=-x(-n)$, então $X(k)$ será *ímpar*.

Sequências reais: Assumindo o par $x(n)$ e $X(k)$, tem-se :

$$\text{Se } x(n) \text{ é real, então : } X(k)=X^*(-k)=X^*(N-k) \quad (8)$$

$$\text{Se } X(k) \text{ é real, então : } x(n)=x^*(-n)=x^*(N-n) \quad (9)$$

Sequências imaginárias: Assumindo o par $x(n)$ e $X(k)$, tem-se :

$$\text{Se } x(n) \text{ é imaginário, então : } X(k)=-X^*(-k) \quad (110)$$

$$\text{Se } X(k) \text{ é imaginário, então : } x(n)=-x^*(-n) \quad (11)$$

Relações entre sequências reais/imaginárias e pares/ímpares: Assumindo o par $x(n)$ e $X(k)$, tem-se :

Se $x(n)$ é real e par, então $X(k)$ é real e par.

Se $x(n)$ é real e ímpar, então $X(k)$ é imaginário e ímpar.

Se $x(n)$ é imaginário e par, então $X(k)$ é imaginário e par.

Se $x(n)$ é imaginário e ímpar, então $X(k)$ é real e ímpar.

Convolução: Assumindo duas sequências de N pontos, $x(n)$ e $g(n)$, e as suas respectivas transformadas $X(k)$ e $G(k)$, tem-se que a *convolução* entre $x(n)$ e $g(n)$, é definida como :

$$\begin{aligned} x(n)*g(n) &= \sum_{m=0}^{N-1} x(m).g(n-m) = \\ &= \sum_{m=0}^{N-1} g(m).x(n-m) \end{aligned} \quad , \quad n \in [0, N-1] \quad (12)$$

Possui as seguintes propriedades :

$$DFT[x(n)*g(k)] = N.X(k).G(k) \quad (13)$$

$$DFT[x(n).g(k)] = X(k)*G(k) \quad (14)$$

Cálculo (simultâneo) da DFT, para duas sequências reais - [deFatta]: Considere duas *sequências reais* : $x_1(n)$ e $x_2(n)$ e suas transformadas $X_1(k)$ e $X_2(k)$. Combinando essas duas sequências reais em uma única *sequência complexa* $x(n)$ onde : $x(n) = x_1(n) + i.x_2(n)$, tem-se :

$DFT[x(n)] = X(k) = X_1(k) + i.X_2(k)$, de onde tem-se :

$$X_1(k) = \frac{[X(k) + X^*(N-k)]}{2} \quad (15)$$

$$X_2(k) = \frac{[X(k) - X^*(N-k)]}{2.i} \quad (16)$$

2 O cálculo da FFT [Weaver]

A FFT é o algoritmo rápido para o cálculo da DFT. Portanto todas as propriedades anteriormente vistas valem para as seqüência de entrada e saída da FFT. Existem vários tipos de algoritmos FFT que via de regra se equivalem entre si. Aqui será apresentado o mais comum deles, ou seja: o algoritmo FFT radix-2, com decimação no tempo.

Decimação-no-tempo Uma vez que N é um número composto potência de 2, separa-se a seqüência original, $x(k)$ em duas novas seqüências de $N/2$ pontos, sendo a primeira de *números pares* e a segunda de *números ímpares*:

$$X(k) = \sum_{r=0}^{(N/2)-1} x(2r) \cdot W_N^{(2r)k} + \sum_{r=0}^{(N/2)-1} x(2r+1) \cdot W_N^{(2r+1)k} \quad (17)$$

$$X(k) = \sum_{r=0}^{(N/2)-1} x(2r) \cdot (W_N)^{2rk} + W_N^k \cdot \sum_{r=0}^{(N/2)-1} x(2r+1) \cdot (W_N)^{2rk} \quad (18)$$

onde :

$$W_N^2 = e^{-2 \cdot j \cdot (2\pi/N)} = e^{-j \cdot 2 \cdot \pi / (N/2)} = W_{N/2} \quad (19)$$

Substituindo $W_N^2 = W_{N/2}$, a DFT original passa a ser representada por duas DFTs de $N/2$ pontos. A primeira, $G(k)$, formada pelos $N/2$ pontos pares da DFT original e a segunda, $H(k)$, pelos seus $N/2$ pontos ímpares:

$$X(k) = G(k) + W_N^k \cdot H(k) \quad (20)$$

Aplicando da propriedade da periodicidade, para uma DFT original de 4 pontos, tem-se :

$$\begin{aligned} X(0) &= G(0) + W_N^0 \cdot H(0) \\ X(1) &= G(1) + W_N^1 \cdot H(1) \\ X(2) &= G(2) + W_N^2 \cdot H(2) \\ X(3) &= G(3) + W_N^3 \cdot H(3) \end{aligned}$$

Fig. 1 Fragmentação de uma seqüência $X(k)$ de 4 pontos em duas seqüências. A primeira de números pares, $G(k)$ e a segunda de ímpares, $H(k)$. Ambas de 2 pontos.

Se $G(k)$ e $H(k)$ também forem seqüências compostas, aplica-se o mesmo processo anterior :

$$G(k) = \sum_{l=0}^{(N/4)-1} g(2l) \cdot W_N^{lk} + W_{N/2}^k \cdot \sum_{l=0}^{(N/4)-1} g(2l+1) \cdot W_{N/4}^{lk} \quad (21)$$

$$H(k) = \sum_{l=0}^{(N/4)-1} h(2l) \cdot W_{N/4}^{lk} + \sum_{l=0}^{(N/4)-1} h(2l+1) \cdot W_{N/4}^{lk} \quad (22)$$

Para uma sequência $x(k)$ com $N = 2^v$ pontos, o processo da fragmentação, ou divisão da DFT em duas novas DFTs se repete v vezes, onde $v = \log_2 N$. Em cada um desses v estágios, ocorrem N operações complexas. Dessa maneira, o número final de cálculos complexos de todo o processo da FFT é igual a $N \cdot \log_2 N$.

Exemplificando, para uma sequência $x(k)$ de 8 pontos, tem-se inicialmente a fragmentação da DFT original em duas DFTs de 4 pontos:

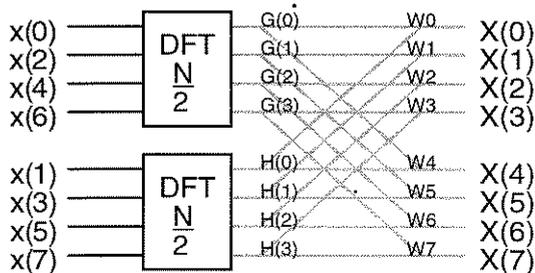


Fig. 2 Fragmentação da DFT de uma sequência $x(k)$ de N pontos em duas DFTs de $N/2$ pontos, pelo método da decimação-no-tempo

Cada uma das DFTs de $N/2$ pontos é novamente fragmentada em outras duas DFTs de $N/4$ pontos:

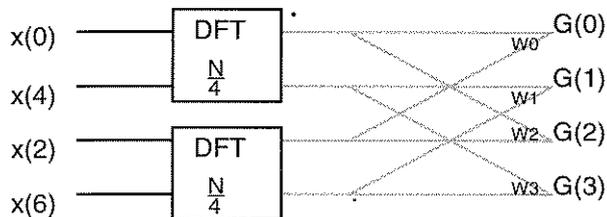


Fig. 3 Fragmentação da DFT de $N/2$ pontos em duas DFTs de $N/4$ pontos, pelo método da decimação-no-tempo

Fazendo o diagrama completo da fragmentação no tempo da sequência de 8 pontos, tem-se:

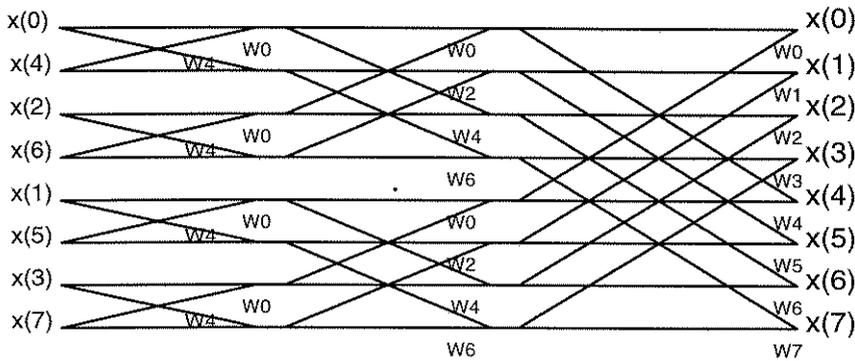


Fig. 4 Resultado da substituição da Fig. 3 na Fig. 2

Contando cada um dos ramos do diagrama acima, observa-se que cada estágio possui N operações complexas. Existem $\log_2 N$ estágios, deste modo tem-se um total de $N \cdot \log_2 N$ operações complexas em todo o processo. Este tipo de fragmentação no tempo, onde a DFT original é sucessivamente dividida em duas DFTs, é conhecido como FFT radix-2.

É possível implementar outros algoritmos de FFT que dividem a DFT original em 4 (radix-4), ou 8 (radix-8) DFTs de $N/4$, ou $N/8$ pontos. Estes algoritmos foram desenvolvidos porque apresentam melhor adaptação em algumas implementações de hardware. No entanto a FFT radix-2 é o algoritmo que proporciona a maior fragmentação possível da DFT original.

Cálculo In-place

Para cada um dos $v = \log_2 N$ estágios, ocorrem operações complexas sobre um vetor $X_m(l)$, onde $m = 1, 2, \dots, v$ estágios e $l = 0, 1, \dots, N-1$ pontos. Os valores de $X_{m+1}(l)$ dependem apenas das operações entre W_N^r e $X_m(l)$:

Pode-se montar um diagrama relacionando $X_{m+1}(l)$ e $X_m(l)$:

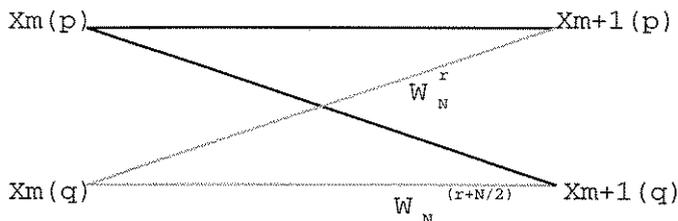


Fig. 5 Gráfico de fluxo típico da butterfly.

Em termos de equações :

$$\begin{aligned} X_{m+1}(p) &= X_m(p) + W_N^r \cdot X_m(q) \\ X_{m+1}(q) &= X_m(p) + W_N^{r+N/2} \cdot X_m(q) \end{aligned} \quad (23)$$

O diagrama dessas equações é conhecido como butterfly. Pode-se ainda considerar que :

$$W_N^{N/2} = e^{-j(2\pi/N) \cdot N/2} = e^{-j\pi}$$

Tem-se o diagrama a seguir:

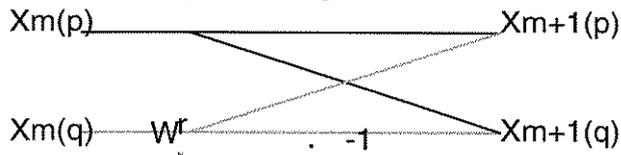


Fig. 6 Gráfico de fluxo butterfly simplificado, requerendo apenas uma multiplicação complexa.

O cálculo dos N pontos de $X(k)$ em um estágio dependem apenas dos valores dos N pontos de $X(k)$ do estágio anterior. Esse tipo de organização é chamado de cálculo *In-place*.

Sua vantagem é poder utilizar apenas dois registradores de armazenamento. Um para o estágio atual dos cálculos e outro para o anterior.

Quando os cálculos são concluídos em um estágio específico, os valores atuais dos N pontos de $X(k)$ são carregados no registrador anterior e os cálculos do próximo estágio se iniciam armazenando os valores no registrador atual. Isso ocorre sucessivamente para todos os v estágios. O cálculo *in-place* reduz o número de multiplicações complexas de $N \cdot \log_2 N$ para $(N/2) \cdot \log_2 N$ [11].

O gráfico-de-fluxo de uma DFT de 8 pontos, com butterfly, fica :

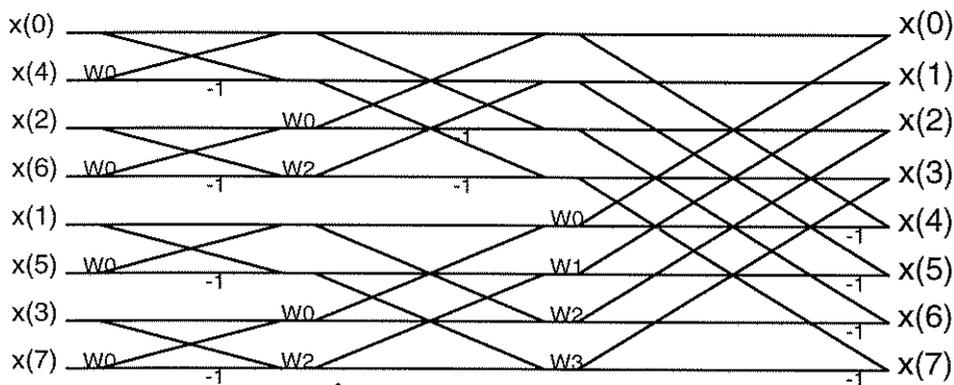


Fig. 7 Gráfico-de-fluxo do cálculo da DFT de uma sequência de 8 pontos utilizando butterfly

Ordenação Bit-reverso

Observa-se na figura 1.6 que a sequência de entrada não se apresenta ordenada. A maneira que a sequência inicial deve ser ordenada chama-se ordenação em bit-reverso.

Para obter esta ordenação, deve-se *espelhar* o índice n da sequência de entrada $x(n)$ em sua representação binária.

Exemplificando, para a sequência de entrada $x(k)$ de 8 pontos, a representação binária do índice terá 3 dígitos. Assim:

$$\begin{aligned} X(000) &= x(000) \\ X(001) &= x(100) \\ X(010) &= x(010) \\ X(011) &= x(110) \\ X(100) &= x(001) \\ X(101) &= x(101) \\ X(110) &= x(011) \\ X(111) &= x(111) \end{aligned}$$

Fig. 8 Exemplo de ordenação em bit-reverso de uma sequência de 8 pontos através do espelhamento do valor binário dos índices.

Desse modo ocorre uma inversão dos bits que representam o índice, ou o ponto da sequência de entrada $x(k)$. Os bits mais significativos, na ordenação normal, tornam-se os menos significativos em bit-reverso, e vice-versa. Na decimação-no-tempo, este é o primeiro passo do algoritmo da FFT.

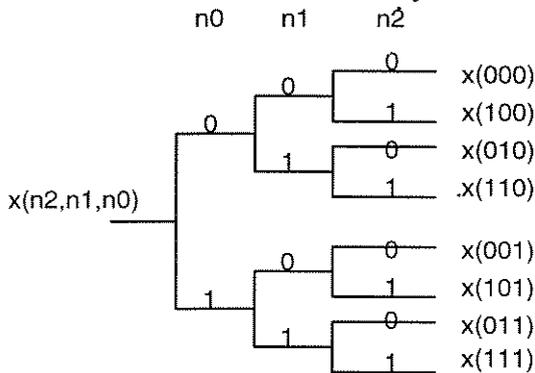


Fig. 9 Diagrama em árvore da ordenação em bit-reverso

Calcular uma DFT de 8 pontos através do algoritmo da FFT tem $\log_2 8 = 3$ estágios, cada um com 8 multiplicações e 8 adições. Em geral, o número de cálculos complexos da FFT é de $N \cdot \log_2 N$ ao invés de N^2 do cálculo direto da DFT. Como exemplo, se considerarmos $N=1.000$ pontos, a FFT demanda cerca de 100 vezes menos cálculos que o método direto, e esta taxa aumenta quanto mais cresce o valor de N . Sabe-se também que o erro de arredondamento reduz na FFT numa taxa de $\log_2 N/N$. Existem algumas restrições quanto ao uso da FFT. Em primeiro lugar, não é economicamente viável o uso da FFT para um número pequeno de pontos. Em segundo, a FFT não pode ser implementada para software, em tempo-real, a menos que se use recursos especiais de hardware [Beauchamp]. Com o método direto da DFT :

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk}$$

, pode-se trazer um ponto de $x(n)$, multiplicá-lo por W_N^{nk} , fazer a somatória para os n pontos e achar um único valor de $X(k)$. Não se pode fazer o mesmo com a FFT. Para que esta seja possível, tem-se que trabalhar com valores de N que sejam potência de 2. Quando se está trabalhando com valores pré-fixados de N , que não são potências de 2, deve-se simular um novo valor de N , N' , maior que N e que seja potência de 2. Em N' vão todos os pontos de N e o restante é completado com zeros. Levando-se em conta todos os aspectos citados, conclui-se que a FFT só é economicamente viável quando $N > 50$ pontos [17]. Fora esse caso, o método direto da DFT é preferido.

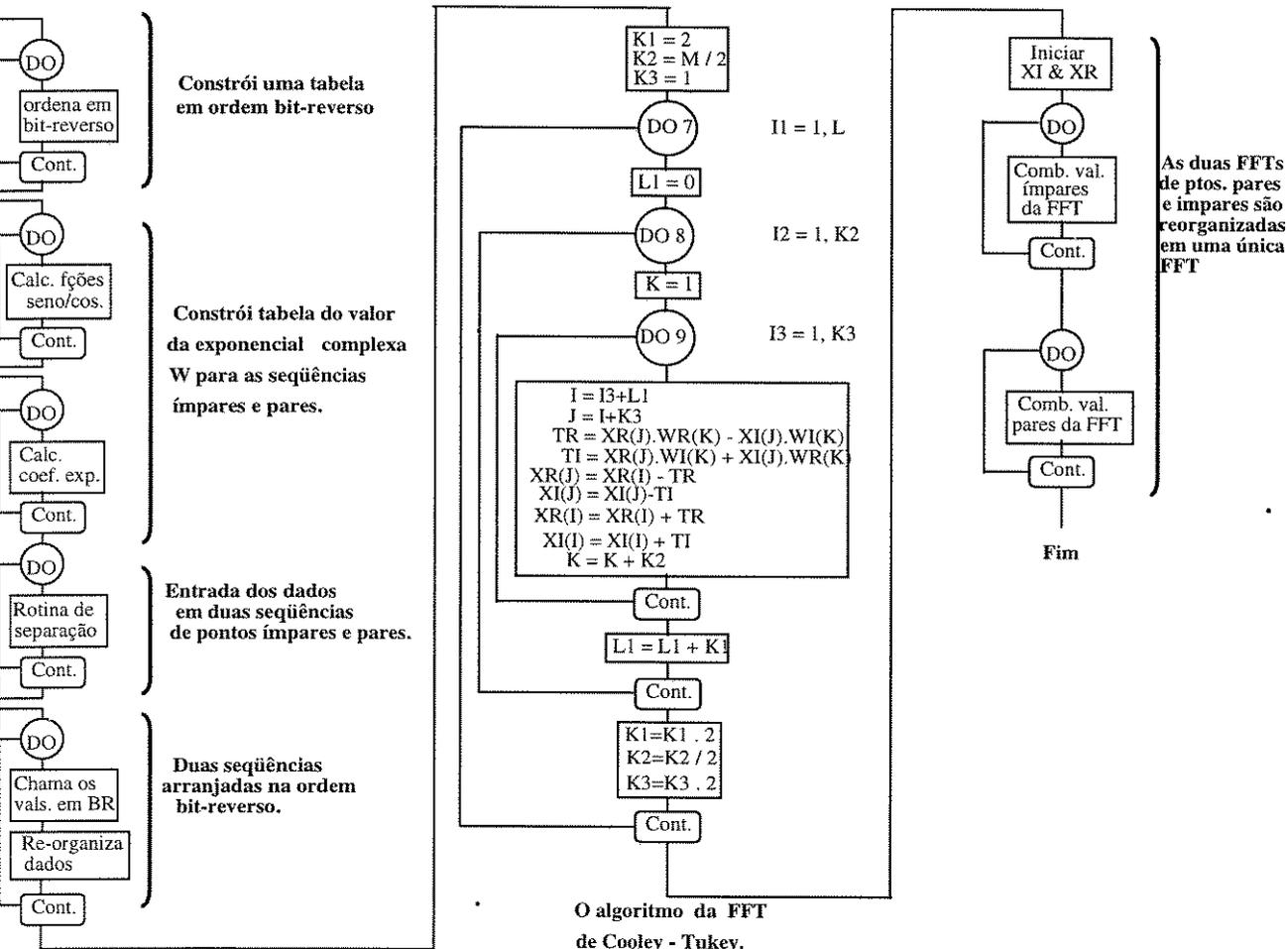


Fig. 10 Diagrama esquemático para o cálculo da FFT

A entrada complexa do algoritmo da FFT pode ser modificado para trabalhar com entradas reais[Beauchamp], nesse caso a seqüência de entrada de N pontos $x(n)$ será dividida em duas seqüências de N/2 pontos $x_1(n)$ e $x_2(n)$. Uma delas será posta no lugar da seqüência real, a outra, como a parte imaginária da primeira seqüência, ou seja: $x(n) = x_1(n) + i.x_2(n)$. A seqüência complexa de saída da FFT será: $X(k) = X_1(k) + i.X_2(k)$, onde: $X_1(k)=0,5. [X(k)+X(N-k)]$ e $X_2(k)=0,5i. [X(k)-X(N-k)]$. Quando a entrada da FFT é uma seqüência real, a sua saída é complexa e simétrica. Utilizando este artifício, tem-se duas seqüências $X_1(k)$ e $X_2(k)$ de N/2 pontos, também complexas e simétricas.

2 O janelamento e overlap da entrada da FFT [deFatta]

A sequência de entrada da FFT é formada por N pontos reais que representam um intervalo de tempo do som que foi amostrado, com taxa de amostragem F_s . Esse intervalo de tempo é equivalente a N/F_s segundos. O algoritmo da FFT irá encarar essa entrada como um período de uma função (quantizada) periódica. Se essa sequência de entrada não tiver seus pontos extremos tendendo ao zero, o fato do algoritmo da FFT encará-la como periódica induzirá numa descontinuidade, que gera um ruído a cada N/F_s s. Esse ruído é conhecido como glitch.

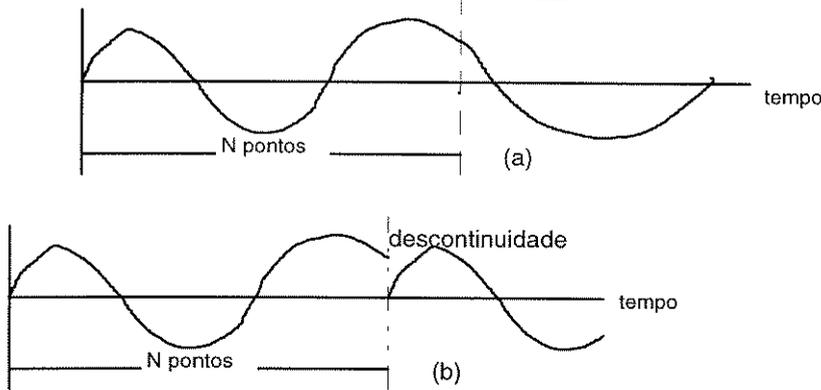


Fig. 11 Ruído causado pela descontinuidade da sequência de entrada.

Este ruído pode ser eliminado através do janelamento. O janelamento irá modular a sequência de entrada. Desse modo os pontos extremos da sequência sempre irão tender a zero, onde não haverá mais nenhuma descontinuidade.

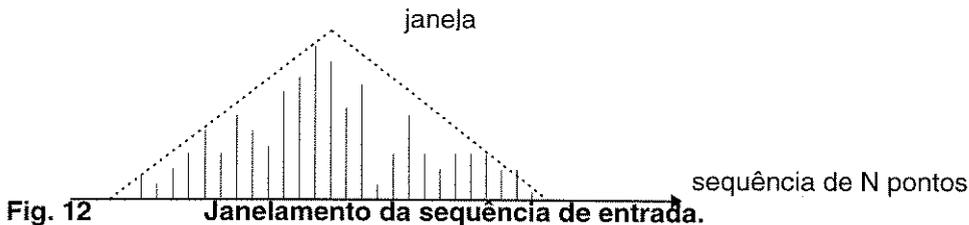


Fig. 12 Janelamento da sequência de entrada.

Existem uma série de famílias de funções de janelamento[Eden], as mais importantes são listadas abaixo:

Função triangular A função triangular é definida por :

$$a(n) = \begin{cases} 2n/N & n = 0,1,\dots,N/2 \\ a(N-n) & n = (N/2)+1,(N/2)+2,\dots,N-1 \end{cases} \quad (24)$$

Com o janelamento triangular da entrada, a largura da raia principal do espectro é dobrada. A queda para a primeira raia é de -26dB. As próximas raias decaem aproximadamente -12 dB por oitava. O scalloping loss que ocorre devido ao cruzamento dos fk é de -1,82dB. $ENBW = 1,33$.

Função Hann (Hanning): Esta janela é definida como :

$$a(n) = 0,5 \cdot [1 - \cos(2 \cdot \pi \cdot n / N)], \quad n=0, 1, 2, \dots, N-1 \quad (25)$$

Trata-se de uma janela da família que utiliza a função $\cos^a(\theta)$, com $a=2$. A queda de magnitude para a primeira raia é -32 dB. O scalloping loss é de -1,42dB. ENBW = 1,5.

Função Hamming: Essa janela é similar à anterior, apenas modificando os coeficientes da expressão do cosseno:

$$a(n) = 0,54 - 0,46 \cdot \cos(2 \cdot \pi \cdot n / N) \quad n=0, 1, 2, \dots, N-1 \quad (26)$$

O decaimento da primeira raia é de -43dB. As próximas raias decaem de -6 dB por oitava. SL= 1,78 dB. ENBW= 1,36.

Função Blackman-Harris: As janela Hann e Hamming são exemplos de funções da família do $\cos^a(\theta)$. Pode-se expandir essa família com uma somatória de cossenos :

$$a(n) = \sum_{m=0}^M (-1)^m \cdot b(m) \cdot \cos(2 \cdot \pi \cdot n \cdot m / N), \quad n=0, 1, 2, \dots, M \quad (27)$$

onde a soma dos $b(m)$ é igual à unidade.

A janela Blackman utiliza este conceito. Ela é formada por 3 termos $b(m)$: $b(0)=0,42$, $b(1)=0,50$, $b(2)=0,08$. Harris usou este método para criar uma família de janelas que apresentam um mínimo de nível de raias laterais. A função da família de janelas Blackman-Harris é :

$$a(n) = a_0 - a_1 \cdot \cos\left(\frac{2 \cdot \pi}{N-1} n\right) + a_2 \cdot \cos\left(\frac{2 \cdot \pi}{N-1} 2 \cdot n\right) - a_3 \cdot \cos\left(\frac{2 \cdot \pi}{N-1} 3 \cdot n\right) \quad (28)$$

para : $n=0, 1, 2, \dots, N-1$

Função Kaiser-Bessel: Os coeficientes dessa janela são :

$$a(n) = \frac{I_0(\pi\beta)}{I_0(\pi\alpha)} \quad 0 \leq |n| \leq N/2$$

$$\beta = \alpha [1 - (2n/N)^2]^{0,5}$$

$$I_0(x) = \sum_{k=0}^{\infty} \left[\frac{(x/2)^k}{k!} \right]^2 \quad (29)$$

Uma das maneiras de se determinar o família de janela a ser utilizada é através da largura da banda do ruído equivalente (ENBW) [DeFatta]. Segundo os experimentos de Harris, com análise de harmônicos, afim de avaliar o desempenho dos tipos de janelas, chegou-se a conclusão de que as melhores, no que tange a prover uma melhor seletividade de frequência foram as janelas das famílias

Blackman-Harris e Kaiser-Besel. As K.B. são preferíveis pois os seus coeficientes são mais facilmente gerados num computador.

Correlação de entrelaçamento (overlap correlation): Quando a janela é aplicada na entrada da FFT, os pontos próximos dos extremos da janela tem seus ganhos reduzidos. Isto reduz o glitch mas também cria vazios entre as saídas da FFT, de N em N pontos.

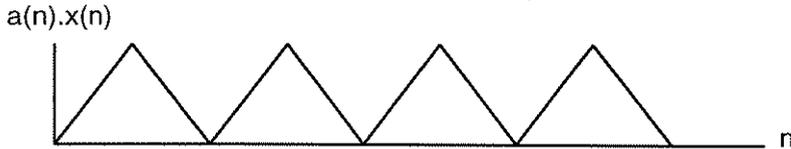


Fig. 11 Janelamento triangular sem o entrelaçamento (overlap).

Isto pode ser evitado se forem feitas FFTs adicionais entre as FFTs anteriores.

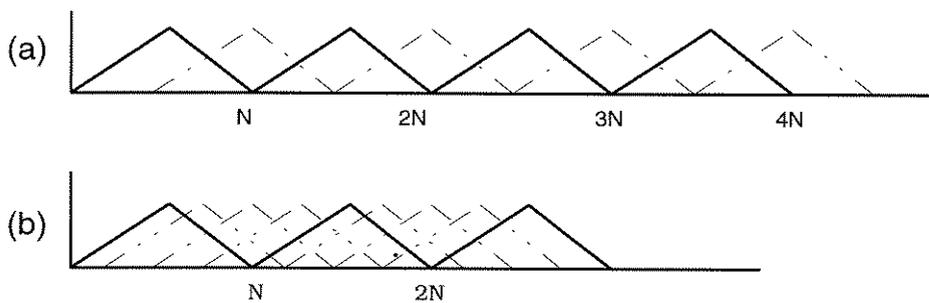


Fig. 12 Janelamento da entrada com : (a) 50% de overlap. (b) 75% de overlap.

Para quantificar do overlap, em relação ao processamento redundante de dados, tem-se a expressão abaixo. Trata-se de um coeficiente de correlação \$c(r)\$ entre as sucessivas FFTs, onde \$r\$ é a fração de overlap aplicado ao sinal janelado e \$a(n)\$, a função janela,.

$$c(r) = \frac{\sum_{n=0}^{N-1} a(n) \cdot a[n + (1-r)N]}{\sum_{n=0}^{N-1} a^2(n)} \quad (30)$$

Geralmente utiliza-se uma taxa de overlap entre 50% e 75% que apresenta 90% de correlação para a maioria das famílias de janelas.

4 Análise de erros da FFT - [deFatta]:

A sequência de saída da FFT possui \$N\$ pontos complexos, do tipo: \$a+i.b\$, que representam a magnitude \$r = (a^2 + b^2)^{1/2}\$ e a fase \$\varphi = \arctg(b/a)\$ dos harmônicos do som. No entanto, existem apenas \$N\$ pontos na saída da FFT para representar um número de harmônicos por intervalo de som frequentemente maior que \$N\$. A distância entre dois pontos da saída da FFT equivale a um intervalo de frequência bin = \$F_s/N\$. Isto é chamado de seletividade da frequência (Frequency Selectivity). Os harmônicos cuja frequência não coincide com um múltiplo inteiro de bin, é representado pelos pontos vizinhos, próximos ao valor de frequência original do harmônico. Isto é conhecido como espalhamento espectral (spectral leakage). Esse fenômeno induz o aparecimento de harmônicos extra, que não fazem parte do som original. Quando a entrada da FFT possui um harmônico cuja frequência se encontra exatamente entre dois pontos do eixo da frequência, ou \$(k+0,5) \cdot F_s/N\$ tem-se a maior redução da magnitude da

representação do harmônico, na saída da FFT. A magnitude do harmônico só permanece sem perda se sua frequência coincidir exatamente com um dos pontos do eixo da frequência. Essa perda em magnitude é chamada de Scalloping Loss.

O cálculo da FFT incorre numa série de erros e ruídos, devido ao comprimento finito dos registradores que armazenam o resultado das operações complexas. As duas maneiras utilizadas para a representação dos resultados das operações complexas pelos registradores (de comprimento finito) são : representação em *ponto-fixo* e *ponto-flutuante*. Na aritmética de ponto-fixo os valores são armazenados em registradores de $b+1$ bits, sendo b bits para o valor do sinal e 1 bit para o sinal. Para a aritmética de ponto-flutuante, aloca-se 1 bit de sinal da mantissa, 1 bit de sinal do expoente, m bits para expoente e b bits para a mantissa. O DR máx. para FFT pto-fixo é :

$$DR_{FIP} = 6,02.b_r - 3,01.\mu - 6,99 \quad (31)$$

O DR máx. para FFT pto-flutuante, é :

$$DR_{FLP} = 6,02.b_r - 10,1.\log_{10} \mu - 3,77 \quad (32)$$

onde: $b_r=23$ bits para aritmética de ponto-fixo, ou 18 bits da mantissa, na aritmética de ponto-flutuante.
 μ =número de estágios do cálculo da FFT, onde $N=2^\mu$.

Para o pto.-flutuante aloca-se 1 bit de sinal da mantissa, 1 bit de sinal do expoente e 4 bits para expoente. A comparação do DR das duas FFTs é dado na tabela abaixo. Pode-se ver que somente quando uma FFT possui um número de pontos $N \geq 1024$ é que a aritmética de pto.-flutuante torna-se mais apropriada.

TABELA 1

COMPARAÇÃO DA PERFORMANCE ENTRE FFT DE PTO-FIXO E FLUTUANTE.

N	DR PONTO-FIXO	DR PONTO-FLUTUANTE
256	107,39	103,10
512	104,38	102,59
1024	101,37	102,13
2048	98,36	101,72
4096	95,35	101,34
8192	92,34	100,99

Apêndice 2: Os programas da simulação do sistema, em software

Foram desenvolvidos dois grupos de programas para a simulação do funcionamento básico deste sistema. O primeiro grupo chama-se MH e simula: a entrada do som, sua amostragem no tempo, seu janelamento e overlap, o cálculo dos espectros e a montagem final da matriz harmônica. O segundo grupo de programas chama-se OT e simula os operadores timbrais e as suas respectivas transformações sonoras. Como a simulação não pode ser feita em tempo-real, as transformações sonoras através dos operadores são feitas sobre toda a matriz harmônica. Na realidade, para o sistema funcionando em tempo-real, as operações timbrais são feitas sobre cada espectro, porém com parâmetros fornecidos pela MH. Os programas de simulação estão organizados em sub-diretórios, do seguinte modo:

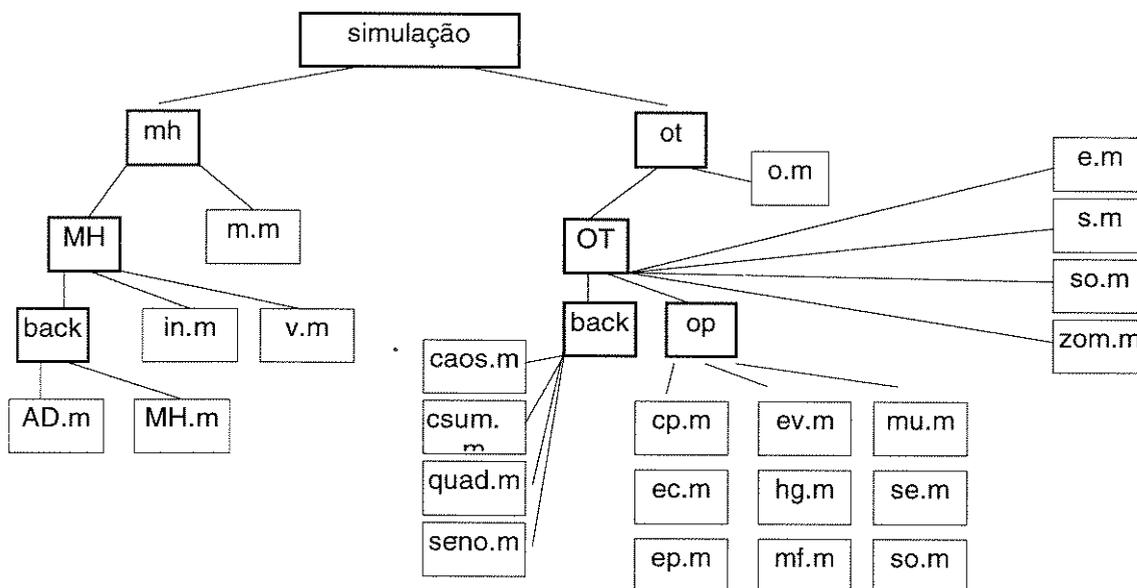


Fig. 1 Estrutura da simulação do sistema em software.

1. Grupo MH:

O grupo MH possui cinco programas básicos: **m.m** que inicia a simulação da montagem da matriz harmônica, **AD.m** que gera um vetor formado por uma somatória de senóides, simbolizando um som analógico de t segundos de duração, e o amostra com taxa de amostragem F_s . **MH.m** faz o janelamento, o overlap e a série de FFTs com número de pontos igual ou inferior a 50ms. **in.m** que fornece valores tais como: harmônicos presentes no som, taxa de amostragem, taxa de overlap, número de pontos da FFT **v.m** mostra graficamente a MH resultante. O grupo OT possui três sub-grupos: entrada e saída do som amostrado; análise e audição do som (original e transformado) e os operadores timbrais.

m.m:

```
clear;  
cd MH;  
mm=0;
```

```

while mm~=3,
mm=menu('M.H.', 'entrada', 'graficos', 'EXIT');
if mm==1 in;end;
if mm==2 v;end;
end;
cd ..;
%clear;

```

in.m:

```

%-----
%VARIAVEIS DO MH
%Nfft      numero de pontos de cada FFT
%Nfft >= Fs/1.5,
%bin >= Fs/Nfft

%VARIAVEIS DO AD
%f0        freq. fundamental do sinal em y f0<=(Fs/2)*(3/2)^(1-n) harmonicos
%n         numero de harmonicos de y, por f0
%Fs        frequ^encia de amostragem do sinal
%segs     qtos segundos de som amostrado y, segs >= (3/2)*Ncel/Fs
%ov        overlap da janela ( 0 - 1 )

%-----
'SINAL DE ENTRADA ' % dados p/ o AD.m
segs=input('tempo de duracao do som, em segundos ');
f0=input('freq. das senoides do som ');
n=input('numero de harmonicos naturais, por senoide ');
Fs=input('taxa de amostragem do sinal Fs=');

'PRE-PROCESSAMENTO '
'bin = Fs/Nfft' % dados p/ MH.m
Nfft=input('interv.tempo, por linha, (em ms) ');Nfft=Nfft*Fs/1000;
jan=menu('JANELAMENTO?', 'sim', 'nao');
ov=input('taxa de overlap ( 0 - 1 ) ');

cd back;
MH;
cd ..;

%clear matriz Nfft i j ans m;

%save mh; % salva todas as variaveis da execucao do programa
%quit;

```

v.m:

```

' MOSTRA MH E PERMITE VISUALIZAR UMA REGIAO ESPECIFICA'
'dimensao de MH'
size(mh)
fh=f0(length(f0))*(3/2)^(n-1);
figure;
tc=(1:length(yc))*(fh/length(yc));t=(1:length(y))/Fs;
subplot(2,1,1), plot(tc,yc), title('SOM ANALOGICO, um periodo');
xlabel('tempo (s)');ylabel('Amplitude');
subplot(2,1,2), bar(t,y), title('SOM AMOSTRADO');xlabel('tempo (s)'); ylabel('Amplitude');

dim=size(mh);t1=(1:dim(1))*(segs/dim(1));f1=(1:dim(2))*(Fs/(2*dim(2)));
figure;mesh(f1,t1,mh);
ylabel('tempo (s)');xlabel('frequencia (Hz)');title('SUPERFICIE SONORA');
view(30,-37);
roda=0;while roda~=2,
roda=menu('RODAR MH ? ', 'sim', 'nao');
if roda==1
roda=input('graus de rotacao do eixo da amplitude ');
view(30+roda,-37);
end;
end;

%clear mhr q wir jr q w ir t1 f1 dim ov segs Fs roda;

```

AD.m:

```

'conversao A/D'

% cria e amostra um sinal

clear y yc;

```

```

R=500;
T=1/Fs;
tempo=round(Fs*segs);
%fh=f0(length(f0))^(3/2)^(n-1); % frequencia mais alta do sinal
for l=1:tempo,y(l)=0;end;
for l=1:R,yc(l)=0;end;
J_c=length(f0);K_c=n;

for J=1:length(f0),
for l=1:tempo, % sinal amostrado: y
for K=1:n,
y(l)=y(l)+(1/2)^K*sin(2*pi*((l*T)-1)*(f0(J))^(3/2)^(K-1));
end;
end;
for l=1:R, % sinal analogico: yc
for K=1:n,
yc(l)=yc(l)+ ((1/2)^K*K_c)*sin(2*pi*((l/R)-1)*(f0(J_c))/f0(1))^(3/2)^(K_c-1);
end;
if K_c>=2 K_c = K_c -1;end;
end;
if J_c>=2 J_c = J_c -1;end;
end;
y=y/max(y);
yc=yc/max(yc);
clear l K K_c J J_c T tempo fh R ;

```

MH.m:

'MONTAGEM DE MH'

```

AD; % SIMULACAO DA AMOSTRAGEM DO SOM
%
if ov==0 lc=length(y)/Nfft; % CALCULA O NUMERO DE LINHAS DE MH
else lc=((length(y)-Nfft)/(Nfft*(1-ov)));
end;lc=round(lc);
%
c=0;matriz=zeros(lc,Nfft); % MONTA Y NA FORMA DE MATRIZ
for i=1:lc,
for j=1:Nfft,
c=c+1;if c>length(y) c=length(y);end;
matriz(i,j)=y(c);
end;

c=c-(Nfft*ov);
end;
%
tr=hanning(Nfft); c=0;
for i=1:lc,
if jan==1 % JANELAMENTO DAS LINHAS
for j=1:Nfft,
matriz(i,j)=matriz(i,j)*tr(j);
end;
end;

c=0;
end;
clear tr;
%
for i=1:lc,
for j=1:Nfft,
espec(j)=matriz(i,j);
end;
espec=fft(espec);
for j=1:Nfft,
matriz(i,j)=espec(j);
end;
for j=1:(Nfft/2),
mh(i,j)=abs(espec(j));
end;

end;

mh=mh/max(max(mh));
%
%clear i j k lc espec out c matriz;

```

2. Grupo OT:

O grupo OT gera uma matriz harmônica para qualquer arquivo de entrada, no padrão u-law, com $F_s=8.192$ Hz, $N_{fft}=256$, $bin=32$ Hz e $jt=31$ ms. Sobre essa MH aplica-se programas de simulação de operações timbrais. Os programas desenvolvidos são: **o.m** que inicia a simulação **e.m** entrada do arquivo de som e montagem da MH original **s.m** saída, montagem do arquivo de som modificado a partir da MH modificada **som.m** saída de audio do som original e modificado **zom.m** visualização de um trecho do som, que se deseja observar os detalhes da envoltória do tempo e harmônicos componentes **caos.m** gera arquivo de som caótico **csum.m** conversão da saída no padrão MATLAB para o padrão u-law **quad.m** gera som de onda quadrada **seno.m** gera som composto por n senoides **cp.m** operador que comprime as magnitudes dos harmônicos de MH **ec.m** expande/comprime as linhas da MH **ep.m** elimina regiões de pausas do som **ev.m** modifica envoltória da superfície sonora **hg.m** multiplica/divide espectro por um determinado harmônico **mf.m** deslocamento do espectro no eixo da frequência **mu.m** multiplicação de espectros distanciados no tempo **se.m** preserva apenas os n maiores harmônicos **so.m** soma dois espectros distanciados no tempo.

o.m:

```
OTclear;
menu('SIMULACAO DAS OPERACOES TIMBRAIS','ARQUIVOS DE SOM NO PADRAO U-LAW','Fs=8192','8 bits de resolucao','Nfft=256 pontos','bin=31 ms');
clear;
cd OT;
in=0;esc_oper=0;
while in~=6,
in=menu('O.T.s','entrada','saida','som','zoom','operacoes','EXIT');
if in==1 e;end;
if in==2 s;end;
if in==3 som;end;
if in==4 zom;end;
if in==5 & exist('mh')==1 esc_oper=1;figure; e; end;
end;
cd ..;
```

OT

e.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% ENTRADA
% converte: Y -> MH e mostra as representacoes graficas: de Y como o som
% amostrado no tempo, das magnitudes de MH', como a representacao da
% superficie sonora.
%
% VARIAVEIS:
% jc numero de linhas, correspondente ao tempo (s).
% i,j,q,c contadores para a montagem de vetores e matrizes
% vsom vetor do som original, de 100*jc pontos
% z matriz de amostras do vetor vsom
% mh matriz harmonica MH, com elementos complexos
% mh1 matriz harmonica, com a magnitude (real) de mh
% e vetor que armazena o espectro para montar mh e mh1
% zoomc
% vsb, mhb,mh1b copias de: vsom, mh e mh1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if esc_oper==0

'ENTRADA'

zoomc=0;qp=0;Ncel=256;Fs=8192;
sqt=0.01; % porcentagem de eliminacao de harmonicos
jnl=hanning(Ncel); %harming,hanning,blackman,bartlett,triang...

jch=menu('janelamento','sim','nao');
ov=input('taxa de overlap? (0 - 1) ');if ov==" ov=0;end;
segs=input(' segs de som? ');if segs==" segs=.1;end;
all=Fs*segs;al=round(all);
jc=(all-Ncel)/(Ncel*(1-ov));j=round(jc)+1;

vsom=menu('ARQUIVOS DE SOM:','caos','chirp','gong','handel','laughter','splat','train','seno','quadrada','SPARCfile ');vsom=vsom-1;ph=Ncel;

cd Back;
if vsom == 0
caos;
elseif vsom == 1
load chirp;
elseif vsom == 2
```

```

        load gong;
    elseif vsom == 3
        load handel;
    elseif vsom == 4
        load laughter;
    elseif vsom == 5
        load splat;
    elseif vsom == 6
        load train;
    elseif vsom == 7
        seno;
    elseif vsom == 8
        quad;
    elseif vsom == 9
        arqname=input('NOME DO ARQUIVO ? ','s');y=auread(arqname);
        clear arqname;
    end;
cd ..;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:all, vsom(1,i)=y(i); end;
vsom=vsom/max(vsom);
c=0;
for j=1:jc,
    for i=1:ph,
        c=c+1;
        if c>length(vsom) celula(i,j)=0;
        else celula(i,j)=vsom(1,c);
        end;
    end;
c=c-(Ncel*ov);c=round(c);
end;
t=(1:all)/Fs;
subplot(4,1,1), plot(t,vsom), title('SOM ');
xlabel('tempo (s)');ylabel('Amplitude');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:jc,
for i=1:ph,espec(i)=celula(i,j);end;
if jch==1 for i=1:ph,espec(i)=espec(i)*jnl(i);end;end; %JANELAMENTO
espec=fft(espec);
for q=1:ph,
mh(q,j)=espec(q);
end;
end;clear espec;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sqmax=max(mh);
for j=1:jc,for i=1:ph,
%if abs(mh(i,j))<=max(abs(sqmax*sqrt(i))) mh(i,j)=0;end; %elimina ruidos de mh
end;end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:jc,for q=1:(ph/2), mh1(q,j)=abs(mh(q,j)); end;end;
mh1=mh1/abs(max(sqmax));
%
% monta a matriz harmonica normalizada e c/ magn.linear
crr=256*(1-ov)^c/length(vsom);
t1=(1:jc)*segs*crr/jc; f1=1:(ph/2)*Fs/ph; % eixos de tempo e freq.
subplot(2,1,2), mesh(t1,f1,mh1), title('SUPERFICIE SONORA');
view(70,50);
xlabel('tempo (s)'); ylabel('frequencia (Hz)');zlabel('Amplitude');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mhb=mh; mh1b=mh1; vsb=vsom;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all c celula i j jc jch jnl y MENU_VARIABLE Ncel q qp sqrt sqmax t;

operadores=0;
cd op;

while operadores--=10,

operadores=menu('OPERACOES TIMBRAIS','compress.nivel','expans./contr.','elimin.pausas','envoltoria','harm.guia','modif.pitch','elim.harmonicos','soma de ss','produto de
ss','EXIT');

    if operadores==1 cp;end;
    if operadores==2 ec;end;
    if operadores==3 ep;end;
    if operadores==4 ev;end;
    if operadores==5 hg;end;
    if operadores==6 mf;end;
    if operadores==7 se;end;
    if operadores==8 so;end;
    if operadores==9 mu;end;

end;
clear operadores;
cd ..;

```

s.m:

```
cd Back;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               SAIDA
%   converte MH -> Y e mostra as representacoes graficas: das magnitudes da
%   MH' modificada, como a representacao da superficie sonora, e do Y' como o % som
%   modificado, amostrado no tempo.
%
%   VARIAVEIS
%   jc,ph, i, j, c, z, mh
%   mh1      matriz das magnitudes de mh
%   mhbb     copia de mh
%   ask      condicional para qual som sera' apresentado
%   vsm      vetor do som modificado
%   zoomc    habilitacao de zoom
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mhbb=mh;zoomc=0;
if exist('vsm')==1 vant=vsm; else vant=vsom; end;
ask=menu('SAIDA','original','transformada');
    if ask==1 mh=mhb; cd ..;
    else mh=mhbb; zoomc=1;
    end;
[ph,jc]=size(mh);clear mh1;
for j=1:jc,for i=1:(ph/2),mh1(i,j)=(abs(mh(i,j)));end;end;
mh1=mh1/abs(max(max(mh1)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[ph,jc]=size(mh);
t1=(1:jc)*segs*crf/jc; f1=(1:(ph/2))*Fs/ph;
figure;
subplot(2,1,1), mesh(t1,f1,mh1), title ('SUPERFICIE SONORA MODIFICADA');
view(70,50);
xlabel('tempo (s)');ylabel('frequencia (Hz)');zlabel('Amplitude');
if ask==1
    title('SUPERFICIE SONORA ORIGINAL'); end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear vsm;for i=1:length(vsom), vsm(i)=0;end;
for i=1:jc,
    for j=1:ph,espec(j)=mh(j,i);end;
    espec=real(fft(espec));
    for j=1:ph,out(i,j)=espec(j);end;
end;
c=0;
for i=1:jc,
    for j=1:ph,
        c=c+1;if c>length(vsm) c=c-1;end;
        vsm(c)=vsm(c)+out(i,j);
    end;
c=c-(ph*ov);c=round(c);
end;
vsm=vsm/max(vsm);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=(1:length(vsm))/8192;
subplot(4,1,4), plot(t,vsm), title('SOM MODIFICADO');
xlabel('tempo (s)');ylabel('Amplitude');
if ask==1
    title('SOM ORIGINAL'); end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%clear ask c espec t f1 i j;
cd ..;
```

som.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               ROTINA DE SOM
%   permite escutar o som representado por Y, nas opcoes: som original, som
%   modificado e som anterior a ultima modificacao.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ask=menu('AUDICAO DO SOM','original','modificado','anterior');
if ask == 1
    sound(vsom);sc=vsom;
elseif ask == 2
    sound(vsm);sc=vsm;
elseif ask == 3
    sound(vant);sc=vant;
end;
ask=input('REPETICOES ? ');
for i=1:ask,
    sound(sc);
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear vait;
```

zom.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           ZOOM
%   apresenta a representacao grafica do Y e o usuario escolhe um intervalo
%   entre dois pontos de Y. Este intervalo e'apresentado novamente na sua
%   representacao no tempo, e tambem o seu espectro de frequencia.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if zoomc ~= 1 break; end;
figure:plot(vsm),title('SOM MODIFICADO');
grid:xlabel('pontos (s)');ylabel('Amplitude');
[xz,yz]=ginput(2);
        if xz(1)<=0, xz(1)=1; end;
xz(1)=round(xz(1)); xz(2)=round(xz(2));
clear vzoom; for i=1:xz(2), vzoom(i)=0; end;
for q=1:(xz(2)-xz(1)),
        vzoom(q+xz(1)-1)=vsm(q+xz(1)-1);
end;
ez=abs(fft(vzoom));
t1z=(xz(1):xz(2))/Fs;
subplot(3,1,1),
plot(vzoom), title('DETALHE DO SOM');
axis([xz(1) xz(2) min(vzoom) max(vzoom)]);
xlabel('pontos (s)');ylabel('Amplitude');
for i=1:xz(2)/2, e1z(i)=ez(i); end;
e1z=e1z/max(e1z);
f1z=(1:length(e1z))*(Fs/length(e1z))/2;
subplot(2,1,2), bar(f1z,e1z), title('HARMONICOS');
xlabel('frequencia (Hz)');ylabel('Amplitude');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear vzoom xz yz ez t1z f1z e1z;
axis('normal');
```

Back:

seno.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%           SENOS
%   gera vetor Y composto por uma ou mais senoides.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fs=8192;clear y;
senq=input('QUANTOS SENOS ? ');

for i=1:all, y(i)=0; ysen(i)=0;end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for q=1:senq,
        sfrq=input('FREQUENCIA (0 - 4000) ? '); sfrq=sfrq/8.192;
        smag=input('AMPLITUDE (0 - 1) ? ');
                if smag==" smag=1; end;
        sph=input('FASE (0 - 360) ? ');
                if sph==" sph=0; end;

        for i=1:all,
                ysen(i)=smag*sin(2*pi*(i-1)*sfrq/1000+(sph*0.0175));
        end;
        y=y+ysen;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear senq sfrq smag sph ysen;
```

csum.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   CONVERSAO DE SOM E ARMAZENAGEM
%   converte arq.som u-law, para padrao MATLAB (vetor Y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cau=input('LER(1) OU ESCREVER(2) ARQUIVO .au ? ');
cname=input('nome do arquivo ? ','s');
if cau==1
        y=auread(cname);
end;
if cau==2
        if exist('vsm') auwrite(vsm,cname);else auwrite(vsom,cname);end;
```

end;

op:

op

cp.m:

%%%

```
%
%           COMPRESSAO
%           diminui a diferencas de intensidade entre os harmonicos do
%           espectro do som. Pode ser usado para valorizar harmonicos que
%           estao com intensidade muito baixa, em relacao ao maior
%           harmonico.
```

%%%

'COMPRESSAO DO NIVEL DE INTENSIDADE'

```
[ph,jc]=size(mh);
[tM,pM]=max(max(mh));
[tm,pm]=min(min(mh));
comp=input('TAXA DE COMPRESSAO (0-1) ? ');
mid=tM-tm;

for j=1:jc,for i=1:ph,
    if mh(i,j)>pm
        if mh(i,j)>mid mh(i,j)=mh(i,j)-(comp*(mh(i,j)-mid)); end;

        if mh(i,j)<mid mh(i,j)=mh(i,j)+(comp*(mid-mh(i,j)));end;
    end;
end;end;
[tm,pm]=min(min(mh));
for j=1:jc,for i=1:ph,mh(i,j)=mh(i,j)-tm;end;end;
```

%%%

clear comp tM pM tm pm mid;

ec.m:

%%%

```
%           EXPANSAO/COMPRESSAO
%           manipulando as colunas, este operador altera a duracao do som no tempo,
%           sem alterar as suas frequencias. Manipulando as linhas afasta/aproxima
%           os harmonicos no eixo da frequencia. O efeito nesse caso e'de migracao,
%           ou distanciamento dos harmonicos, no eixo da frequencia.
```

%%%

'EXPANSAO/COMPRESSAO DE MH'

```
[ph,jc]=size(mh);
cdim=menu('EXPANSAO/COMPRESSAO','no tempo','na frequencia');
cec=menu('E/C','expansao','compressao');
%if cdim ~= 1 | cdim ~= 2 break; end;
%if cec ~=1 | cec ~= 2 break; end;
clear vace;clear mhm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

[ph,jc]=size(mh);

```
if cdim==1 % E/C no tempo
    for i=1:ph,
        for j=1:jc,
            vace(j)=mh(i,j);
            end;
        if cec==1 % EXPANSAO
```

```
c=0;
for q=1:jc,
c=2*q;
vacee(c)=vace(q);
vacee(c-1)=vace(q);
end;clear c;
clear vace;vace=vacee;clear vacee;
end;
```

if cec==2 % COMPRESSAO

```
c=0;
for q=1:2:length(vace),
c=c+1;
```

```

        vacec(c)=vace(q);
        end;
        for q=1:length(vacec),vace(q)=vacec(q);end;
        for q=length(vacec):length(vace),vace(q)=0;end;
        clear c;clear vace;vace=vacec;clear vacec;
        end;
        for j=1:length(vace),
        mhm(i,j)=vace(j);
        end;
    end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if cdim==2          % E/C na frequencia
    for j=1:jc,
        for i=1:ph,
            vace(i)=mh(i,j);          % monta vetor auxiliar para E/C
            end;
        if cec ==1          % EXPANSAO
            c=0;
            for q=1:ph/2,
                c=2*q;
                vacee(c)=vace(q);
                vacee(c-1)=0;
                end;
            clear c;
            vacee(2*ph)=0;
            for q=1:ph,vacee(2*ph-q)=vacee(q);end;
            clear c1;clear c;clear vace;vace=vacee;clear vacee;
            end;
        if cec==2          % COMPRESSAO
            for q=1:ph/2,
                vacec(q)=vace(2*q);
                end;
            c=length(vacec);
            for q=1:c,
                vacec(2*q-q)=vacec(q);
                end;
            clear c vace;vace=vacec;clear vacec;
            end;
            for i=1:length(vace),
                mhm(i,j)=vace(i);
                end;
            end;
        end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear mh;mh=mhm;clear mhm;

```

ep.m:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%          ELIMINADOR DE PAUSAS
%
%          elimina as regioes de silencio, ou intensidade de som, abaixo de
%          um nivel de intensidade escolhido. O som resultante tem menor duracao
%          no tempo que o som original.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'ELIMINADOR DE PAUSAS'

[ph,jc]=size(mh);
[tM,pM]=max(max(mh));[tm,pm]=min(min(mh));
mid=(tM-tm)/2;
cel=input('NIVEL DE ELIMINACAO (0-1) ? ');
c=0;
mhm=zeros(ph,jc);
for j=1:jc,
    for i=1:ph,vaux(i)=mh(i,j);end;
    [tMn,pMn]=max(vaux);
    if abs(tMn)-(cel*abs(mid))
        c=c+1;
        for i=1:ph,mhm(i,c)=vaux(i);end;
    end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear mh;mh=mhm;clear mhm;
clear vaux c tM tm pM pm mid tMn pMn;

```

ev.m:

%%%

% ENVOLTORIA
% em relacao `as linhas da MH', o efeito e`de um filtro, cujas
% caracteristicas podem ser "desenhadas" sobre o espectro do som (regioes
% de passagem e de corte, curvas de decaimento...). Em relacao `as
% colunas, permite "redenhar" a envoltoria do som, no tempo (ataque,
% release, sustentacao e decaimento).
%

%%%

'MODIFICADOR DE ENVOLTORIA'

```
[ph,jc]=size(mh);
envc=menu('MODIFICAR A ENVOLTORIA','do tempo','da frequencia');
if envc==1 envc=jc;
elseif envc==2 envc=ph/2;
else break; end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;stop=0;clear envolt;
for i=1:envc,envolt(i)=0;end;
subplot(4,1,1);
if envc==ph/2 mesh(mh1),view(90,0),title('FREQUENCIA');end;
if envc==jc mesh(mh1),view(0,0),title('TEMPO');end;
subplot(2,1,2),grid,axis([0 envc 0 envc]),title('ENVOLTORIA');
ai=1;bi=1;
while stop == 0,
[a,b]=ginput(1);if a<0 | b<0 [a,b]=ginput(1);end;
a=round(a);b=round(b);
alfa=(b-bi)/(a-ai);
for i=ai:(a+1),
envolt(i+1)=envolt(i)+alfa;
if i>=envc a=envc;stop=1;end;
end;
ai=a;bi=b;
plot(envolt);grid,axis([0 envc 0 envc]);
end;
envolt=envolt/max(envolt);
clear c alfa beta a b;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if envc==jc
mhm=zeros(ph,jc); % envoltoria no tempo
for j=1:jc,
for i=1:ph,vpm(i)=mh(i,j);end;
vpm=envolt(i)*vpm;
for i=1:ph,mhm(i,j)=vpm(i);end;
end;end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if envc==ph/2
mhm=zeros(ph,jc); % envoltoria na frequencia
for i=1:ph/2,
for j=1:jc,vpm(j)=mh(i,j);end;
vpm=envolt(i)*vpm;
for j=1:jc,mhm(i,j)=vpm(j);end;
end;
for i=((ph/2)+1):ph,
for j=1:jc,vpm(j)=mh(i,j);end;
vpm=envolt(i-ph/2)*vpm;
for j=1:jc,mhm(i,j)=vpm(j);end;
end;
end;
mhm=mhm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear mhm vpm envolt;
```

hg.m:

%%%

% HARMONICO-GUIA
% condiconia a intensidade do som `a evolucao da`magnitudo de um unico
% harmonico, no tempo.
%

%%%

```
'HARMONICO-GUIA '
Fs=8192;
[ph,jc]=size(mh);
fref=input('frequencia do harmonico (0- 4096 Hz) ? ');
fref=fix((fref*ph)/(2*Fs));
mhm=zeros(ph,jc);
```

```

ask=menu('OPERACAO NOS ESPECTRO','multiplicar','dividir');
if ask==1
    for j=1:nc,
        for i=1:ph, vpm(i)=mh(i,j);end;
        [tm,pm]=max(vpm);
        while fref==0, fref=input('escolha outro ponto de frequencia ');end;
        if fref==("")
            freqref=abs(vpm(pm));elseif freqref==0 freqref=1;
        else
            freqref=fref;end;
        vpm=vpm*freqref;
        for i=1:ph,mhm(i,j)=vpm(i);end;
        end;
else
    for j=1:nc,
        for i=1:ph, vpm(i)=mh(i,j);end;
        [tm,pm]=max(vpm);
        while fref==0, fref=input('escolha outro ponto de frequencia ');end;
        if fref==("")
            freqref=abs(vpm(pm));elseif freqref==0 freqref=1;
        else
            freqref=fref;end;
        vpm=vpm/freqref;
        for i=1:ph,mhm(i,j)=vpm(i);end;
        end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear mh;mh=mhm;clear mhm;
clear vpm fref freqref tm pm ctm ask;

```

mf.m:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% MANIPULACAO DA FREQUENCIA
%
% altera a frequencia do som (para o grave, ou agudo) sem alterar a sua
% duracao no tempo.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'MANIPULACAO DA FREQUENCIA'
[ph,nc]=size(mh);
mhm=zeros(ph,nc);
desloc=input('ptos do deslocamento ? '); % deslocamento
if desloc>ph | desloc<-ph break; end;
for j=1:nc,
    for i=1:ph/2,
        vph(i)=mh(i,j); %monta vetor auxiliar vph
    end;
    if desloc>0
        for q=1:(desloc-1),
            c=ph/2;
            for i=1:(ph-1)/2,
                vph(c)=vph(c-1);
                c=c-1;
            end;
            vph(1)=0;
        end;
    end;clear c;
    if desloc<0
        for q=1:(abs(desloc)+1),
            for i=1:(ph-1)/2,
                vph(i)=vph(i+1);
            end;
            vph(ph/2)=0;
        end;
    end;
    if desloc == 0 % inversao
        c=ph/2;clear ivp;
        for i=1:ph/2,
            ivp(i)=vph(c);
            c=c-1;
        end;clear vph;
        for i=1:ph/2,vph(i)=ivp(i);end;
    end;clear ivp;
    c=ph;
    for i=1:ph/2,
        mhm(i+1,j)=vph(i);
        mhm(c,i)=conj(vph(i));
        c=c-1;
    end;
end;

```

```

mh=mhm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
clear mhm vph;
```

mu.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Multiplicacao da mh modificada com a original
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
[ph,jc]=size(mh);
```

```
for j=1:jc,for i=1:ph,
    mh(i,j)=mh(i,j)*mhb(i,j);
end;end;
```

se.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% SIMPLIFICADOR ESPECTRAL
```

```
%
% elimina todos os harmonicos do som abaixo de uma certa intensidade
% sonora.
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
'SIMPLIFICADOR ESPECTRAL'
```

```
[ph,jc]=size(mh);
qp=input('QUANTOS PICOS/AMOSTRA ? '); clear vp;
mhm=zeros(ph,jc); %monta mhm nula
for j=1:jc;
    for i=1:ph/2, vp(i)=0;end; % cria vetor nulo
    for i=1:ph/2, vaux(i)=mh(i,j);end; % carrega espectro
    for q=1:qp,
        [tm,pm]=max(vaux);
        vp(pm)=tm;
        vaux(pm)=0;
    end;
    c=ph;
    for i=1:ph/2,
        mhm(i+1,j)=vp(i);
        mhm(c,i)=conj(vp(i));
        c=c-1;
    end;
end;
```

```
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear mh,mh=mhm;
clear mhm qp tm pm vaux vp g c;
```

so.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% SOMA DE SS modificada com a original
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
mh=mh+mhb;
```

Glossário:

- **atak, ataque** - (ver “som no tempo, envoltória”).
- **acústica, grandezas** - grandezas que caracterizam um som: intensidade, em decibéis, frequência, em Hertz, duração no tempo, em segundos.
- **altura** - frequência do harmônico fundamental de um som, *pitch*.
- **amostrado, som** - representação discreta de $s(t)$, simbolizado por $s(n)$.
- **amostragem, teoria da** - um sinal $s(t)$ é devidamente amostrado se a sua taxa de amostragem for $F_s \geq 2 \cdot f_H$, onde f_H é a frequência do seu harmônico de maior frequência.
- **análise** - processamento de áudio visando a medição das características de um dado som.
- **audição, limites da** - limites da percepção auditiva humana. Intensidade de 0 a 120 dB (na frequência de 1KHz); frequência de 20 a 20 KHz; persistência auditiva 50ms.
- **audio, sinal de** - representação elétrica proporcional à variação de pressão correspondente ao som, simbolizado por $s(t)$.
- **ADSP** - (*Audio Digital Signal Processing*). Processamento Digital de Sinais de Áudio.
- **conversão A/D e D/A** - converte o sinal de áudio $s(t)$ em uma sequência de amostras do tipo $s(n)$ em uma dada taxa de amostragem “ F_s ” e numa dada resolução “ b ”; ou vice-versa.
- **DFT** - (*Discrete Fourier Transformation*). Transformada Discreta de Fourier. Cálculo da transformada de Fourier para N pontos amostrados de uma função $s(n)$, com N^2 operações complexas.
- **DSP** - (*Digital Signal Processing*). Processamento Digital de Sinais.
- **decay, decaimento** - (ver “som no tempo, envoltória”).
- **espectro, espectro de frequência** - o conjunto das componentes de frequência de um sinal.
- **fon** - unidade de percepção auditiva da intensidade sonora, equivalente a 1dB na frequência de 1KHz.
- **frequência, componente de** - oscilação de pressão cuja representação temporal é expressa por uma função senoidal, do tipo $A \cdot \sin(\omega t + \phi)$. Sua representação no domínio da frequência é um número complexo, cuja magnitude é o valor de sua intensidade sonora “ A ”, a fase é o valor de “ ϕ ” e a frequência “ f ” está associada a velocidade angular ω , ou seja, $f = \omega / 2\pi$. Uma componente é representada no eixo da frequência por uma magnitude A e frequência f .
- **FFT** - (*Fast Fourier Transform*). transformada rápida de Fourier. Algoritmo eficiente do cálculo da DFT, para sequências de $N = 2^V$ pontos, com $N \cdot \log_2 2^V$ operações complexas.
- **FFT short-time** - FFT que trabalha com vetor de entrada que representa um trecho de som cujo intervalo de duração está abaixo da persistência auditiva (50 ms).

- **Fourier, transformada de.** - representação de uma função do tipo $s(t)$ por uma somatória infinita de senos e cossenos.
- **harmônico** - representação de uma função senoidal da somatória que representa $s(t)$, pela transformada de Fourier. Um harmônico possui amplitude, magnitude e fase.
- **harmônico, plano** - secção da superfície sonora paralela ao eixo da amplitude, em um dado valor de tempo constante.
- **harmônico, vetor** - representação discreta do plano harmônico.
- **loudness** - intensidade sonora percebida pela audição, em fons.
- **matriz harmônica** - representação discretizada da superfície sonora cujos elementos são as magnitudes de harmônicos nos instantes discretos de tempo.
- **MH** - (ver matriz harmônica).
- **morfológico, plano** - secção da superfície sonora paralela ao eixo da amplitude, em um dado valor de frequência constante.
- **morfológico, vetor** - representação discreta do plano morfológico.
- **operador timbral** - estágio do sistema de processamento sonoro onde são executadas as operações timbrais.
- **OT** - (ver operador timbral).
- **operação timbral** - operação parametrizável que modifica o timbre através de operações vetoriais sobre o vetor do espectro de um som.
- **operação vetorial** - unidade estrutural da operação timbral.
- **persistência auditiva** - dois estímulos sonoros separados por intervalo de tempo menor que o tempo da percepção auditiva são percebidos pela audição como um único estímulo.
- **pitch** - percepção auditiva da frequência de um som, em Hertz.
- **processamento, retardo** - tempo de atraso entre o sinal de entrada e o de saída em um sistema de processamento.
- **quality** - mesmo que timbre.
- **síntese** - processamento de áudio visando a criação de novos sons a partir de elementos sonoros e/ou não sonoros.
- **som no tempo, envoltória** - a envoltória do som é comumente dividida em quatro fases ou intervalos de tempo sequenciais, respectivamente: *attack*, *decay*, *sustain*, *release*.
- **superfície sonora** - representação gráfica da evolução dos harmônicos no tempo cujas dimensões são amplitude, frequência e tempo, que correspondem às três grandezas acústicas do som. A superfície sonora (SS) permite visualizar as características do som que constituem o seu timbre.

- **sustain, sustentação** - (ver “som no tempo, envoltória”).
- **SS** - (ver superfície sonora).
- **s(n)** - (ver som amostrado).
- **s(t)** - (ver sinal de audio).
- **release** - (ver “som no tempo, envoltória”).
- **resolução** - número de bits que compõem cada amostra de s(t) feita pelo conv. A/D.
- **taxa de amostragem** - taxa de amostras por segundo feita pelo conversor A/D
- **tempo-real** - propriedade de um sistema no qual o processamento da entrada à saída é feito continuamente ao longo do tempo, sem paradas.
- **timbre** - conjunto de características sonoras percebidas pela audição que permitem o reconhecimento de um dado som.

Bibliografia

- [1] Alles G. Harold. *Music Synthesis Using Real Digital Techniques*. Proceedings of IEEE, vol. 68, n.4, april. 1980.
- [2] Beauchamp, Kenneth George. *Digital methods for signal analysis*. George Allen & Unwin LTD. University press, Cambridge. 1979.
- [3] Blahut, Richard E. *Fast algorithms for digital signal processing*. Addison-Wesley. Publishing company, Inc. 1985.
- [4] Boldrini, José Luis. *Algebra Linear*. 3.ed. cap. 5. São Paulo : Harper & Row do Brasil. 1980.
- [5] Carvalho, Benjamin A. *Acústica aplicada à arquitetura*. Biblioteca técnica Freitas Bastos. Rio de Janeiro. 1967.
- [6] Chamberlin, Hal. *Musical applications of microprocessors*, cap. 2. 2ed. Indianapolis, Hayden Books. 1985.
- [7] Cliford, A. Pickover. *Representation of melody patterns using topographic spectral distribution functions*. Computer music journal, vol.10, n.3, Fall. 1986.
- [8] Culver, C. A. *Musical Acoustics*. 4°ed. cap. 5. McGraw-Hill. 1968.
- [9] DeFatta, David J. *Digital signal processing: cap. 6 e 9. A system design approach*. David J. DeFatta, Joseph G. Lucas, William S. Hodgkiss. Singapore. John Willy & Sons Inc. 1988.
- [10] Dodge, Charles. *Computer music*. cap. 3 e 4. Schirmer books.1985.

- [11] Eden, A.W.M. van den. *Discrete-time signal processing: an introduction*. A.W.M. van den Enden, N.A.M. Verhoeckx. Great Britain: Prentice-Hall International.(U.K.). 1989.
- [12] Fletcher, H. *Speech and Hearing*. D. Van Nostrand Company Inc., New York. 1929.
- [13] Hewlett Packard. *Acoustics Handbook. Application note 100*
- [14] Helmholtz, H. von. *On the sensation of tone*. trans. A. J. Ellis, Longmans. Green & Co., Inc., New York. 1912.
- [15] Hill, Fredrick J. *Digital Systems*. Fredrick J. Hill and Gerald R. Peterson. third edition. John Wiley & Sons. 1987.
- [16] Hilburn, John L. *Manual of active filter design*. 2ed. cap.2. McGraw-Hill Book Company. 1983.
- [17] Johnson, Ian. *Measured Tones*. British Library Cataloguing in Publication Data. IOP Publishing Ltd. 1989.
- [18] Matheus, Max V. *Current directions in computer music*. cap. 9. Massachusetts Institute of Technology. 1989.
- [19] McGee, W. F. *A real-time logarithmic-frequency phase vocoder*. Computer Music Journal. 1991.
- [20] Motorola. *DSP-96002 IEEE Floating-Point Dual-Port Processor User's Manual*. MOTOROLA INC. 1989.
- [21] Moore, Richard. *Applications on computer music*. cap.3.
- [22] Moorer, James Anderson. *Signal processing aspects of computer music: a survey*. Proceedings of the IEEE, vol. 65, n° 8. august 1977
- [23] Oppenheim, Alan V. *Digital signal processing*. A.V. Oppenheim, R.W. Schafer. Englewood Cliffs, N.J: Prentice-Hall. 1975.
- [24] Oppenheim, Alan V. *Application of digital signal processing*. cap. 2. Englewood Cliffs, N.J. Prentice-Hall, 1978.
- [25] Oppenheim, Alan V. *Discrete-time signal processing*. . A.V. Oppenheim, R.W. Schafer. Englewood Cliffs, N.J: Prentice-Hall. 1989.
- [26] Parker. Sybil P. *Acoustics source book*. McGraw-Hill. San Francisco. 1987.
- [27] Plomp, Reiner. *Aspects of the tone sensation, A psychophysiological study*. Academic Press. London, New York, San Francisco. A subsidiary of Hancourt Brace Favonovich Publisher. 1976
- [28] Rosemberg, Martin E. *Sound and Hearing. Studies in Biology n,145*. cap. 2. Afd Sonologie. 1982.
- [29] Sohie, Guy R.L. *A Digital Signal Processor with IEEE Floating-Point Arithmetic*. Guy R. L. Sôhie and Kevin L. Kloker. IEEE MICRO. December 1988.
- [30] Smith, Ralph J. *Ciscuits devices systems*. 2 ed. cap 9. John Wiley & Sons. 1971.
- [31] Weaver, H. Joseph. *Theory of discrete and continous Fourier analysis*. A Wily-Interscience publication. John Wiley & Sons Inc. 1989.

[32] Winckel, Fritz. *Music, sound and sensation*. Dover Publications, Inc. 1967.