

Franco Beltrame Runza

Análise de Sinais Biológicos Utilizando *Wavelets*

Dissertação de Mestrado apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

Área de Concentração:
Engenharia de Sistemas

Orientador:
Prof. Dr. Fuad Kassab Jr.

São Paulo
Junho de 2001

Agradecimentos

Agradeço a todos que tiveram a paciência de ficar ao meu lado durante a realização deste trabalho:

Aos meus amigos do Rotaract de toda parte do Brasil, por terem entendido que se encontrar pouco não apaga a lembrança e a amizade;

Aos meus companheiros, colegas e chefes do Instituto Ludwig, por me terem permitido conciliar trabalho e pesquisa;

A todo o pessoal do Laboratório de Automação e Controle, pelo ambiente maravilhoso que me proporcionaram — um abraço especial ao Anselmo e ao Ricardo;

Ao pessoal do Laboratório de Investigação Médica e ao Dr. Lopasso em especial, pelo imenso trabalho de preparar os animais e coletar os dados;

Aos dois Marcelos, o Lipas e o Fernandes, sem cujo trabalho nada seria realizado;

Ao professor Gastão, pelo incentivo constante desde o início;

À Regina, por todo o amor e compreensão;

À minha família, por todo o apoio e carinho neste tempo todo;

Com todas as desculpas, obrigados e votos de que possamos estar sempre juntos e com sucesso, agradeço do fundo do meu coração especialmente ao meu orientador, Dr. Fuad Kassab Jr., que continuou acreditando em mim e no meu trabalho mesmo nos piores momentos.

Acima de tudo, agradeço à pessoa a quem tenho plena consciência de dever tudo que sou hoje: minha mãe.

Não é o fim, nem mesmo o início do fim, mas o fim do começo.

Winston Churchill

Sumário

1	Introdução	10
1.1	Motilidade do sistema digestivo	11
1.2	Objetivos do projeto	13
1.3	Estrutura da dissertação	14
2	Considerações Teóricas	16
2.1	A transformada de Fourier	17
2.2	Propagação entre sinais	18
2.3	A transformada <i>Wavelet</i>	20
2.4	Outros métodos de análise em tempo-freqüência	24
3	Apresentação dos Dados	25
3.1	Considerações sobre os dados	27
3.1.1	Qualidade dos dados	27
3.1.2	Quantidade e diversidade dos dados	27
3.2	Dados disponíveis	28
4	Resultados	30
4.1	Métodos utilizados	30
4.1.1	Método semi-automatizado	32
4.1.2	Método manual	32

<i>SUMÁRIO</i>	4
4.2 Cálculo computacional da transformada <i>Wavelet</i>	34
4.3 Cálculo da frequência basal do sinal	35
4.4 Apresentação dos resultados obtidos	35
5 Discussão	49
5.1 Análise de precisão dos resultados	50
6 Conclusões	52
6.1 Proposta de continuidade	53
A Listagem dos programas utilizados	61
A.1 fbasal.m	61
A.2 prepara.m	63
A.3 show_ratos.m	63
A.4 show_wave.m	64
A.5 tr2vel.m	65
A.6 trechos2wave.m	67
A.7 ve_sinal.m	69
A.8 wavyze.m	70
A.9 waveplan.m	73
A.10 wavevel.m	74
A.11 wv_matr.m	76

Lista de Figuras

2.1	Exemplos de <i>Wavelets</i>	22
2.2	<i>Wavelet</i> Submetida à Dilatação e Translação	23
3.1	Equipamento de Aquisição de Dados	26
4.1	Transformada <i>Wavelet</i> Típica de um <i>Spike Burst</i>	33
4.2	Seqüência de Canais Mostrando Propagação num Trecho	37
4.3	Transformada <i>Wavelet</i> do Trecho da Figura 4.2	38
4.4	Trecho com Transiente	39
4.5	Transformada <i>Wavelet</i> do Trecho da Figura 4.4	40
4.6	Densidade Espectral de Energia para um dos Trechos	41
4.7	Evolução das Velocidades para o Animal 063	45
4.8	Evolução das Velocidades para o Animal 065	45
4.9	Evolução das Velocidades para o Animal 066	46
4.10	Evolução das Velocidades para o Animal 067	46
4.11	Evolução das Velocidades para o Animal 068	47
4.12	Evolução das Velocidades para o Animal 070	47
4.13	Evolução das Velocidades para o Animal 071	48

Lista de Tabelas

3.1	Leituras Realizadas nos Ratos Normais	29
4.1	Valores da Velocidade e Freqüência Basal – Animais 063 a 067	43
4.2	Valores da Velocidade e Freqüência Basal – Animais 068 a 071	44

Lista de Símbolos

A/D	Analógico / Digital
ARMA	Auto-Regressive Moving Average
$c_{a,b}$	Coefficientes <i>wavelet</i>
$\mathcal{C}(t)_{u,y}$	Correlação cruzada entre $u(t)$ e $y(t)$
DC	Corrente contínua
$f(t)$	Função real no tempo
$\hat{f}(\xi)$	Transformada de Fourier da função $f(t)$
$g(t)$	Função da janela de convolução
GI	Gastro-Intestinal
i	vetor complexo ($i = \sqrt{-1}$)
n	número inteiro
t	tempo
$u(t)$	sinal de entrada do canal anterior
$y(t)$	sinal de entrada do canal posterior
ξ	freqüência
$\varphi_{a,b}(t)$	<i>Wavelet</i> de escala a e translação b

Resumo

A análise de sinais mioelétricos provenientes do tubo gastro-intestinal de animais de laboratório (ratos), conseguidos por meio de eletrodos cronicamente implantados, é peça-chave no entendimento das desordens associadas ao sistema digestivo.

Esta análise enfrenta consideráveis dificuldades quando realizadas por métodos clássicos, em especial os baseados na transformada de Fourier. A interação de várias componentes mioelétricas torna muito complicado e trabalhoso o acompanhamento destes sinais ao longo do tubo digestivo e a obtenção de parâmetros típicos como a velocidade de propagação entre eletrodos.

Estuda-se aqui uma alternativa mais nova e promissora: a transformada *Wavelet*. Utilizando esta ferramenta matemática, torna-se possível obter uma melhor resolução tempo-freqüencial dos sinais estudados, permitindo encontrar padrões referentes à propagação do sinal mesmo em leituras ruidosas e compostas de várias freqüências.

Foram analisados 82 leituras de 9 animais normais do Laboratório de Investigação Médica da Faculdade de Medicina da USP, sendo possível determinar dois parâmetros: a velocidade de propagação média entre eletrodos (cerca de 1.2 cm/s) e as componentes principais da freqüência basal (0.63 e 0.65 Hz).

Abstract

The analysis of myoelectric signals from the gastro-intestinal tube of laboratory animals (mice), recorded by chronically implanted electrodes, is a key stone in understanding the disorders associated to the digestive system.

This analysis meets considerable difficulties when done by classical methods, specially those based in the Fourier transform. The many myoelectric components interactions makes the following of these signals along the digestive tract and the retrieval of typical parameters (such as the propagation velocity between electrodes) a very complicated and laborious task.

Here is studied a newer and more promising alternative: the *Wavelet* transform. Using this mathematical tool, it becomes possible to obtain a better time-frequency resolution of the studied signals, allowing to find patterns related to the signal propagation even in noisy and multifrequencial readings.

82 readings from 9 normal animals belonging to the Medical Investigation Laboratory of the Medicine Faculty of University of São Paulo were analyzed, becoming feasible to determine two parameters: the mean propagation velocity between electrodes (about 1.2 cm/s) and the main components of the basal frequency (0.63 e 0.65 Hz).

Capítulo 1

Introdução

A análise de sinais por métodos clássicos (Transformada de Fourier, métodos baseados na função de transferência do sistema) baseia-se fundamentalmente na presunção do sinal ser quase-estacionário, ou seja, na hipótese de seus valores médios, variâncias e outros parâmetros estatísticos não variarem ao longo do tempo de observação dos mesmos [1, 2], ou variarem de uma maneira predizível por meio de um modelo estatístico simples, como por exemplo o modelo auto-regressivo de médias móveis (*Auto-Regressive Moving Average* – ARMA).

Além disso, estes métodos levam em conta o sinal considerado como um todo, sem resolução temporal que permita distinguir entre fenômenos de curta e de longa duração durante o tempo de observação. Mesmo os métodos baseados nos modelos mais flexíveis consideram o sinal como visto a partir de uma única escala, ou seja, não considerando a diferença de escala dos vários fenômenos que podem estar contidos no sinal.

Porém, em sinais provenientes de sistemas biológicos, é praticamente certa a ocorrência de uma grande variedade de frequências naturais do sistema, devido às diferenças entre as velocidades das interações dos componentes

deste sistema biológico. Assim, os métodos clássicos introduzem quase sempre efeitos no sentido de mascarar os fenômenos de frequências mais altas (no caso de examinarem-se as frequências mais baixas), modelando-os como ruído, ou contaminá-los com as componentes que variam mais lentamente, no caso de examinarem-se as frequências mais altas [3, 4].

Durante os últimos anos, têm sido consideradas alternativas para circundar ou minimizar estes efeitos, como transformadas que permitam obter uma melhor resolução tanto na escala de tempo como na de frequência, notadamente a chamada transformada *wavelet*. Esta ferramenta matemática tem propriedades (enunciadas na seção 2.3) que permitem obter características localizadas do sinal no tempo, conseguindo-se assim separar componentes de longa duração daquelas de curta duração.

A seção 1.1 mostra como os sinais de interesse para este trabalho (obtidos do complexo digestivo de ratos) possuem características que tornam desejável a utilização de novos métodos, como a transformada *wavelet*, em sua análise.

1.1 Motilidade do sistema digestivo

A análise de sinais do trato gastrointestinal (GI) de animais de experimentação tem como principal motivação o estudo da motilidade do aparelho digestivo [5]. Este estudo vem sido realizado em colaboração com a Faculdade de Medicina da Universidade de São Paulo [6], onde estão sendo realizadas as aquisições dos dados utilizados neste trabalho.

Em resumo, o estudo da motilidade do trato GI visa a um melhor entendimento dos processos digestivos e seus mecanismos de controle mioelétrico, de modo a ser possível a compreensão das várias anormalidades que ocorrem quando da realização de intervenções cirúrgicas em pacientes humanos. A ca-

pacidade de prever e duplicar este controle seria o primeiro passo na obtenção de uma solução para os diversos problemas digestivos que os submetidos a estas cirurgias apresentam.

Os animais utilizados nos experimentos são ratos machos Wistar, tendo sido implantados cirurgicamente eletrodos em seu tubo digestivo. Estes eletrodos permitem o registro dos sinais mioelétricos da camada seromuscular do tubo digestivo do animal *in vivo*. Estes registros são realizados através de uma placa de aquisição de dados (conversor A/D) ligada a um microcomputador, conforme mostrados na figura 3.1.

Os registros têm sido realizados com os animais em várias situações de alimentação, bem como em animais modificados por preparação cirúrgica ou indução química de sintomas. Os tipos de animais com registros realizados pretendem incluir:

- normais (grupo de controle);
- submetidos à gastrectomia com reconstrução por gastroenterostomia associada à obliteração da alça;
- submetidos à gastrectomia com instalação de cateter ileal para injeção de lípidos;
- submetidos à derivação do duodeno por meio de Y de Roux;
- submetidos à preparação de van Stiegman;
- com diabetes *mellitus* induzida por meio de administração intraperitoneal de estreptozotocina;
- tratados com cisaprida via oral e
- diabéticos.

Os registros foram realizados com vários animais em cada situação e com vários períodos após a realização dos procedimentos cirúrgicos. A descrição completa de cada registro encontra-se no capítulo 3.

Os registros têm sido analisados por meio das já citadas ferramentas clássicas para Processamento Digital de Sinais e Identificação de Parâmetros. Estas ferramentas foram reunidas, para facilitar a análise, num programa especialmente desenvolvido para este projeto, chamado **Analyzer** [7], que é utilizado dentro do ambiente de processamento numérico MATLAB [8, 9].

Os resultados obtidos até agora experimentalmente [10] e na literatura [11] sugerem que a atividade mioelétrica na porção do trato GI estudada não é aleatória, mas sim caracterizada por duas componentes principais: uma *onda basal* quase sempre presente (composta tipicamente por duas frequências) e *pulsos* (períodos de maior atividade mioelétrica, também chamados de *spike bursts*). Estes pulsos parecem se propagar na direção do alimento sendo digerido, dependendo da situação de alimentação do animal, sugerindo portanto estarem ligados às contrações do processo digestivo.

Utilizando-se o programa **Analyzer**, tentou-se caracterizar a velocidade de propagação destes pulsos ao longo da porção estudada do trato, bem como sua distribuição temporal, repetição e características.

1.2 Objetivos do projeto

Deseja-se obter métodos automatizados ou pelo menos semi-automatizados para a detecção desta propagação a partir dos registros efetuados, com o objetivo de conseguir um modelamento mais preciso da atividade mioelétrica.

Considerando-se a natureza dos sinais mioelétricos estudados, busca-se neste trabalho uma análise dos sinais que leve em conta a natureza tem-

poral das variações no sinal. Como foi citado, a transformada *wavelet* tem propriedades tais que supõe-se ser possível caracterizar melhor a propagação dos sinais, pois este consiste justamente de sinais de longa duração e baixa frequência (onda basal) concomitantemente a outros de curta duração e alta frequência (pulsos).

Além disso, busca-se uma caracterização dos sinais determinantes do movimento gástrico, com sua associação com os *spike bursts* observados.

Por fim, espera-se que este trabalho sirva como ponto de partida para, a longo prazo, ser possível modelar e sintetizar o controle mioelétrico do trato GI, resultando num alívio dos sintomas percebidos pelos paciente humanos submetidos a intervenções cirúrgicas semelhantes às estudadas.

1.3 Estrutura da dissertação

No capítulo 2 estudam-se sucintamente as bases teóricas das ferramentas matemáticas utilizadas neste trabalho: as transformadas de Fourier e *Wavelet* e a correlação cruzada. Não se espera de maneira alguma esgotar estes assuntos, mas sim prover os conceitos necessários à compreensão do trabalho.

No capítulo 3 são apresentados os dados experimentais utilizados neste trabalho: os métodos e materiais necessários à sua coleta, sua quantidade e qualidade e algumas considerações sobre os mesmos.

No capítulo 4 revelam-se os métodos desenvolvidos para a análise dos dados experimentais e os resultados alcançados, especialmente as estimativas da velocidade de propagação dos sinais e frequência basal.

No capítulo 5 discutem-se os resultados obtidos através das ferramentas matemáticas, incluindo-se aí uma análise da validade estatística destes resultados.

No capítulo 6 conclui-se então a análise, em vista das discussões realizadas e resultados obtidos.

O apêndice A traz a lista dos programas desenvolvidos ao longo do trabalho, juntamente com seu código fonte para ambiente MATLAB.

Capítulo 2

Considerações Teóricas

Tradicionalmente, ao analisar um sinal (seja qual for a sua origem), os primeiros parâmetros de interesse são aqueles que consideram o sinal como um todo: suas frequências componentes, sua energia, relação sinal-ruído, etc. Este parâmetros são importantes, por exemplo, no próprio planejamento inicial do experimento e construção física de dispositivos para o mesmo, como filtros, amplificadores e condicionadores de sinal.

Desta visão do sinal nasceu uma das primeiras e mais importantes, até hoje, ferramentas para análise de sinais: a transformada de Fourier. Detalham-se as propriedades desta transformada na seção 2.1.

Uma das mais úteis características desta transformada é proporcionar parâmetros, como distribuição de energia, relação sinal/ruído e frequências componentes sobre o sinal como um todo, pois toda a duração do sinal contribui para seus coeficientes. Assim, a transformada de Fourier é naturalmente indicada para os chamados sinais *quase-estacionários*, cujos parâmetros se modificam muito pouco ao longo da duração do sinal.

Porém, existe a situação onde o sinal não se comporta do mesmo modo ao longo de sua duração, sinais que possuem características *transientes*. Neste

caso, a transformada de Fourier não é suficiente: torna-se necessário recorrer a ferramentas capazes de localizar os parâmetros de interesse em relação à duração do sinal. Estas ferramentas matemáticas permitem então uma análise em *tempo-freqüência*, combinando a capacidade de análise da transformada de Fourier com a possibilidade de detectar eventos não-estacionários no sinal.

A representante mais evidente deste tipo de análise é a escolhida como ferramenta principal neste trabalho, a transformada *wavelet*. Detalham-se suas características e propriedades na seção 2.3.

Neste trabalho também tem-se muito interesse não só na análise de sinais isolados como em sua relação temporal com outros sinais. As ferramentas usuais para esta comparação são a correlação cruzada e o atraso na função de transferência do sinal, estudadas brevemente na seção 2.2.

2.1 A transformada de Fourier

A transformada unidimensional de Fourier de um sinal temporal é dada por [1]

$$\hat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-it\xi} f(t) dt, \quad (2.1)$$

onde ξ denota cada freqüência em que $f(t)$ é decomposta.

Examinando-se (2.1), nota-se que para cada freqüência em que $f(t)$ é decomposta, a função original contribui ao longo de toda a sua duração; logo, a transformada resultante, embora informe as freqüências presentes no sinal analisado, não dá qualquer informação a respeito da localização temporal destas freqüências e nem mesmo de sua duração. Na verdade, não é facilmente recuperável esta informação sobre localização temporal de determinadas freqüências a partir da transformada de Fourier do sinal. Portanto, fenômenos localizados no tempo e não presentes em toda a duração do sinal,

como pulsos ou transientes, não são detectáveis por inspeção dos valores desta transformada.

A fim de acrescentar-se resolução temporal à transformada de Fourier, pode-se adicionar uma janela de convolução $g(t)$, de modo que primeiro convolui-se $f(t)$ com $g(t)$ e depois calcula-se a transformada de Fourier do sinal resultante. Deste modo, constrói-se uma função que mapeia $f(t)$ em duas variáveis (frequência ξ e tempo t_0),

$$\hat{f}_{win}(\xi, t_0) = \int_{-\infty}^{\infty} e^{-it\xi} f(t)g(t - t_0) dt, \quad (2.2)$$

considerando-se que $g(t)$ efetivamente seja bem localizada no tempo e na frequência. A transformada resultante mostra uma distribuição das frequências de $f(t)$ ao redor do tempo t_0 . Tomando-se o módulo $|\hat{f}_{win}(\xi, t_0)|$, o que se obtém é o chamado *espectograma* de f , que é uma das técnicas mais comuns para obter-se uma estimativa da localização tempo-frequência de um sinal.

Mesmo assim, nota-se em (2.2) que este método ainda é limitado pela escolha da janela $g(t)$, já que esta tem sua largura fixa e portanto deve haver um compromisso em sua capacidade de análise: se $g(t)$ for muito estreita, perderá em localização frequencial; se for muito larga, perderá em localização temporal.

2.2 Propagação entre sinais

Um dos objetivos deste trabalho é encontrar um mesmo sinal se propagando entre dois ou mais canais de aquisição de dados. Dois métodos comuns e bastante empregados para a detecção da propagação entre sinais são:

- a estimação baseada na função de transferência do sinal e
- a correlação cruzada.

A função de transferência entre dois sinais com um atraso θ e uma função de primeira ordem pode ser modelada como

$$\frac{Y(s)}{U(s)} = e^{\theta s} \frac{k}{s + a}, \quad (2.3)$$

onde s é a variável de Laplace e $U(s)$ e $Y(s)$ denotam respectivamente as transformadas de Laplace dos sinais temporais dos canais anterior e posterior em termos de orientação espacial. Ao calcular numericamente o valor do atraso θ , fazendo a substituição usual $s = i\xi$ por toda a linha de frequência ξ , nota-se que este método leva em consideração todas as frequências do sinal e não apenas aquelas correspondentes aos *spike bursts*.

Para tentar sanar estas deficiências, utiliza-se neste trabalho uma ferramenta matemática relativamente nova para analisar o sinal obtido nos registros: a transformada *wavelet*. Esta ferramenta já tem sido extensivamente utilizada para a caracterização de sinais biológicos [11, 12, 13, 14, 15, 16, 17, 18, 19, 20] como os utilizados neste trabalho.

Complementarmente a esta transformada, utiliza-se outra técnica de detectar padrões de propagação: a técnica da correlação cruzada, obtida ao calcular-se em cada ponto o produto interno do primeiro sinal com a inversão no tempo do segundo sinal. Se considerarmos os sinais de entrada $u(t)$ e $y(t)$ como os canais anterior e posterior e denotando-se como \mathcal{C} a correlação cruzada, teremos

$$\mathcal{C}(\tau)_{u,y} = \int_{-\infty}^{\infty} u(t)y(\tau + t)dt. \quad (2.4)$$

Nota-se que pode-se obter a correlação cruzada computacionalmente ao multiplicar-se a transformada de Fourier de um sinal pela do outro, fazendo-se então a transformada inversa de Fourier do sinal resultante para obter-se a correlação cruzada entre os dois sinais.

O valor da correlação cruzada será tanto maior em cada ponto quanto mais semelhantes forem os sinais nas redondezas daquele ponto, funcionando como uma métrica de similaridade entre os sinais. Por este motivo, considera-se seu uso apropriado para o fim desejado de detectar-se sinais semelhantes como os que são buscados neste trabalho, já que pode-se utilizá-la para comparar padrões quaisquer de entrada, inclusive os provenientes da transformada *wavelet*.

2.3 A transformada *Wavelet*

A transformada *wavelet*, similarmente à transformada de Fourier, é um operador capaz de decompor um sinal ou função $f(t)$ em componentes freqüenciais (os chamados coeficientes *wavelet* do sinal) e de recuperá-lo a partir destes coeficientes. A maior diferença entre esta transformada e a de Fourier é que cada coeficiente é obtido analisando-se a função original com uma função adequada à resolução das características de $f(t)$: detalhes são analisados utilizando-se funções bem localizadas no tempo, enquanto variações lentas são analisadas utilizando-se funções com suporte maior: é a chamada *análise multi-resolução* do sinal. Além disso, todas estas funções são obtidas através de uma estrutura algorítmica simples, realizada por dilatação e translação de uma função original $\varphi(t)$, chamada de *wavelet*-mãe [3, 21, 22].

As *wavelets* podem portanto ser utilizadas como bases para análises de sinais, em especial sinais com transitórios de curta duração em relação à duração total do sinal ou aqueles em que existe uma variação muito grande das freqüências envolvidas.

É possível construir uma grande variedade de bases de *wavelets*, incluindo bases ortonormais e bases redundantes (*frames*), tendo muitos tipos de

wavelets-mãe sido estudadas até o momento: com suporte finito no tempo ou na frequência, com vários momentos nulos (isto é, com $\int_{-\infty}^{\infty} \varphi(t)t^n dt = 0$ para algum $n > 0$ inteiro). Na literatura encontra-se ampla referência a estas construções [1, 23, 24, 25, 26, 27].

As *wavelets* ortonormais possuem ainda as seguintes propriedades:

$$\int_{-\infty}^{\infty} \varphi(t) dt = 0; \quad \int_{-\infty}^{\infty} |\varphi(t)|^2 dt = 1. \quad (2.5)$$

A figura 2.1 mostra algumas *wavelets* comumente utilizadas.

Como citado, as bases de *wavelets* são geradas através de uma estrutura algorítmica muito simples do tipo

$$\varphi_{a,b}(t) = \frac{1}{\sqrt{a}} \varphi\left(\frac{t-b}{a}\right), \quad (2.6)$$

parametrizadas em a e b . Deste modo, é possível a localização tanto frequencial (relacionada ao parâmetro a , que controla o nível de detalhamento ou *zoom*) como temporal (ditada pelo parâmetro b) do sinal. Toda a base é gerada a partir de uma única função $\varphi(t)$. Na figura 2.2 mostra-se como fica a forma de uma *wavelet* Symmlet para alguns valores de a e b (mostrados na figura como (a, b)).

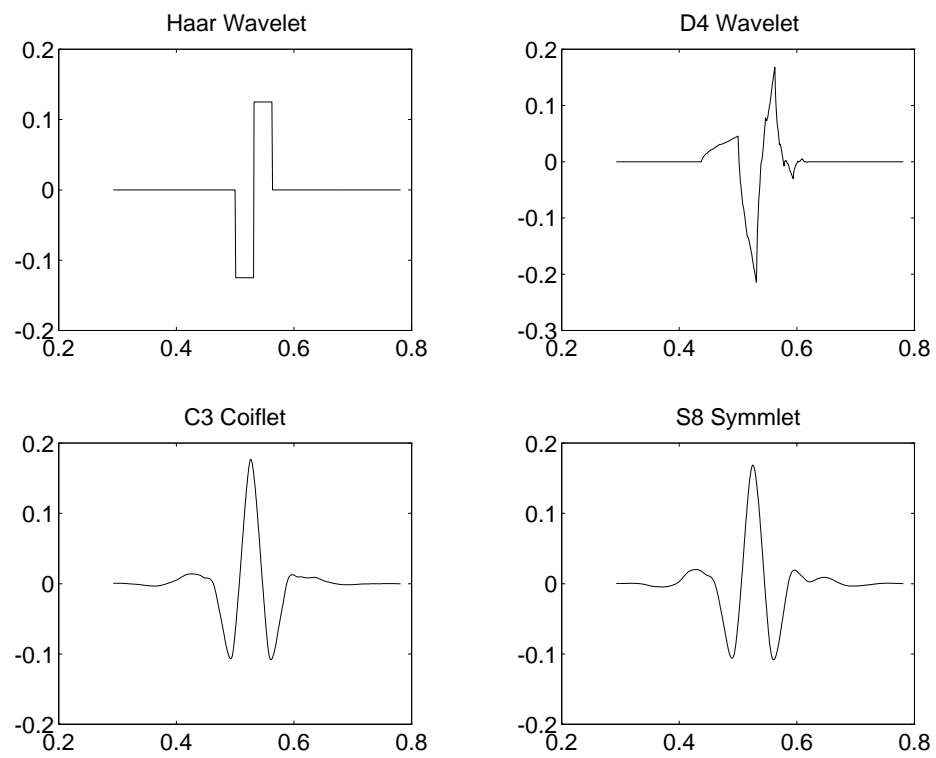
Após a definição da base a ser utilizada, uma função qualquer pode ser decomposta através das fórmulas

$$f(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c_{a,b} \varphi_{a,b}(t) da db \quad (2.7)$$

onde

$$c_{a,b} = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \overline{\varphi}\left(\frac{t-b}{a}\right) dt, \quad (2.8)$$

o traço denotando o conjugado da função φ .

Figura 2.1: Exemplos de *Wavelets*

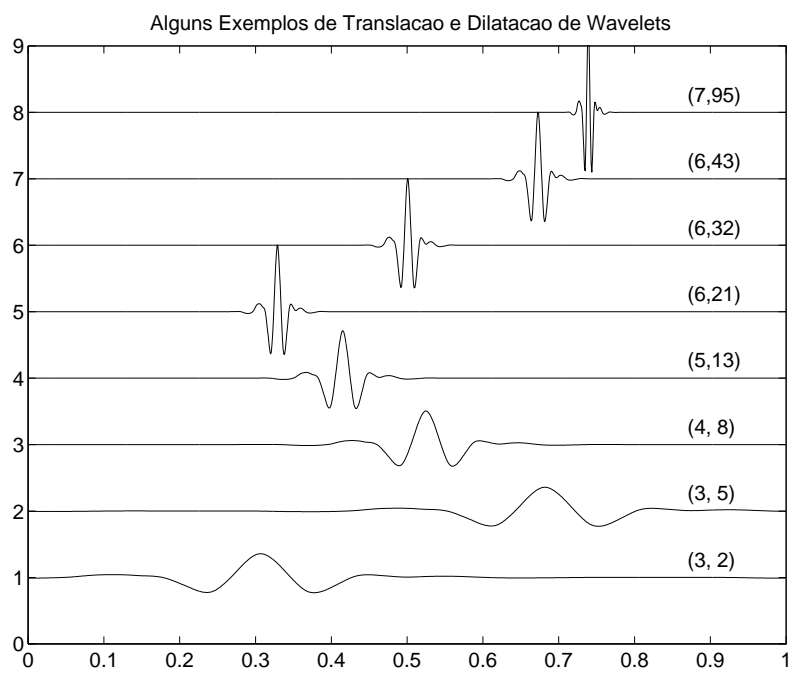


Figura 2.2: *Wavelet* Submetida à Dilatação e Translação

2.4 Outros métodos de análise em tempo-freqüência

As *wavelets* constituem apenas um de vários métodos de análise de sinais em tempo-freqüência [28, 29, 30, 31, 32, 33]. O próprio espectograma e as bases redundantes de *wavelets* (*frames*) constituem outros métodos de análise.

Frames são bases de *wavelets* que não são linearmente independentes. No entanto, estas bases permitem uma decomposição extremamente simples da função original e sua reconstrução de uma maneira estável numericamente. As *frames* permitem inclusive (como as *wavelets*) que o sinal seja comprimido consideravelmente apenas eliminando-se coeficientes abaixo de um determinado valor; a informação que resta é suficiente para a reconstrução, no pior caso afetando apenas os pontos onde os coeficientes foram suprimidos. (No caso da transformada de Fourier, suprimir um coeficiente qualquer afeta toda a duração do sinal.)

Para os sinais de interesse neste trabalho, utiliza-se a decomposição por *wavelets* para obter uma “assinatura” do sinal nos pontos de interesse (*spike bursts*) e em seguida procurá-la no restante das transformadas *wavelet* do sinal.

Capítulo 3

Apresentação dos Dados

Os dados utilizados neste trabalho foram obtidos ao longo de mais de 3 anos (novembro de 1995 a fevereiro de 1999) pelo Laboratório de Investigação Médica (LIM07) da Faculdade de Medicina da USP.

Os animais utilizados nos experimentos são ratos machos Wistar, tendo sido implantados cirurgicamente eletrodos em seu tubo digestivo a distâncias fixas de 1cm a partir do piloro. Estes eletrodos permitem o registro dos sinais mioelétricos da camada seromuscular do tubo digestivo. Estes registros são realizados através de uma placa de aquisição de dados (conversor A/D) ligada a um microcomputador, como se pode ver na figura 3.1.

O condicionador de dados tem como funções:

- atuar como um filtro passa-banda, atenuando frequências acima da metade da taxa de aquisição, para que não haja o efeito de *aliasing*, conforme critério de Nyquist;
- eliminar a componente DC do sinal, de modo a aproximar a média do sinal a ser analisado de zero.

Para evitar em parte a contaminação do sinal com ruídos ou interferências

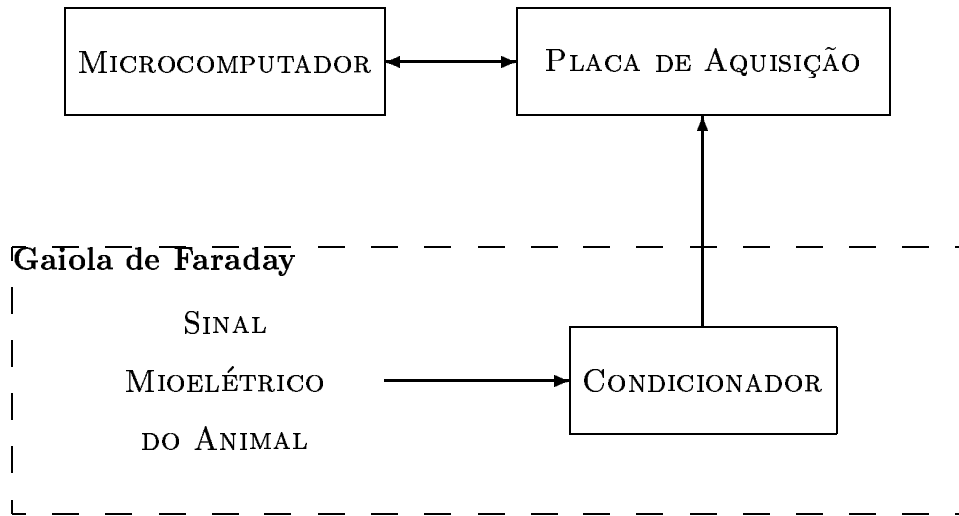


Figura 3.1: Equipamento de Aquisição de Dados

externos, tanto o animal como o condicionador são envoltos por uma gaiola de Faraday.

As leituras foram realizadas inicialmente com uma taxa de aquisição de 50 Hz. Esta taxa foi posteriormente aumentada para 100 Hz, para que fosse possível uma melhor resolução temporal do sinal. Todos os sinais utilizados neste trabalho foram adquiridos com a taxa de 100 Hz. O tempo de cada leitura foi em média de 20 minutos, resultando em arquivos com cerca de 120 mil valores de sinal para cada canal. O total de canais (número de eletrodos) foi definido como 7, sendo o sétimo canal ligado à terra.

Após cada leitura, realizada na Faculdade de Medicina da USP, os dados eram transferidos para o Laboratório de Automação e Controle do Departamento de Engenharia de Telecomunicações e Controle (PTC) da Escola Politécnica da USP, onde realizava-se a conversão do arquivo no formato gerado pela placa de aquisição de dados (formato TEM) para o formato utilizado pelo ambiente MATLAB. Esta conversão é realizada pelo programa

`loadtem` [6, 8, 34], desenvolvido durante o projeto.

3.1 Considerações sobre os dados

A qualidade e quantidade dos dados recolhidos foram determinantes nos resultados conseguidos neste trabalho, conforme mostra-se em mais detalhes no capítulo 4. Explicam-se aqui as características e limitações inerentes a esses dados.

3.1.1 Qualidade dos dados

Mesmo com todo o aparato empregado para garantir uma aquisição do sinal livre de problemas intrínsecos às condições do experimento, os sinais possuíam um nível de ruído considerável, proveniente da própria serosa gastrointestinal e outras fontes externas não passíveis de controle por ocasião do experimento. Além disso, a despeito do grande cuidado na preparação cirúrgica dos animais e na aquisição dos dados, os eletrodos não tinham uma inserção idêntica na camada seromuscular. Em alguns casos não foi possível evitar que alguns eletrodos se desprendessem completamente da serosa ou apresentassem falhas em alguns trechos da aquisição. Estas interrupções eram seguidas por um forte transiente devido ao condicionador DC, típico da resposta ao degrau de voltagem que se apresentava na entrada, tornando o sinal naquele trecho inútil do ponto de vista da análise.

3.1.2 Quantidade e diversidade dos dados

Ao iniciar este trabalho, esperava-se poder analisar vários grupos de animais, desde o grupo normal (que serviria como controle) até aqueles submetidos a vários tipos de intervenções cirúrgicas. Porém, nesta altura, estes dados

simplesmente não estão disponíveis em quantidade suficiente para realizar esta comparação. Portanto, não foi possível realizar todas as comparações propostas na seção 1.1, devido à falta destes dados.

Considerando-se este fato, optou-se por restringir a análise ao grupo normal, com vistas a conseguir-se parâmetros suficientes para automatizar a análise dos demais grupos de animais, assim que estes dados estejam disponíveis. Esta análise será realizada e publicada em trabalhos futuros, para que o objetivo inicial deste trabalho não se perca.

3.2 Dados disponíveis

Conforme as restrições citadas na seção anterior, foram utilizados 82 arquivos de dados (leituras) pertencentes a 9 animais do grupo normal para análise. A tabela 3.1 lista os arquivos de leitura disponíveis por roedor. Cada animal é identificado por um número próprio do laboratório.

Tabela 3.1: Leituras Realizadas nos Ratos Normais

Animal	Leituras realizadas	Total
063	001, 002, 003, 004, 005, 006, 007	7
064	001, 002, 003	3
065	002, 003, 004, 005, 006, 007, 008, 009, 011, 012	10
066	001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012	12
067	001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012	12
068	001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012	12
069	002, 003, 004	3
070	001, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012	11
071	001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012	12

Capítulo 4

Resultados

Os resultados obtidos são de dois tipos:

- o cálculo da velocidade de propagação dos *spike bursts* e frequência basal em cada leitura e
- a análise da mudança deste valor com a passagem do tempo a partir da primeira leitura de cada animal.

Deve-se enfatizar que ao início do projeto não se tinha praticamente nenhuma bibliografia prévia sobre os valores típicos e/ou esperados da velocidade de propagação dos sinal no tubo digestivo, pelo que se explicam os métodos dependentes de intervenção do operador que foram utilizados.

4.1 Métodos utilizados

No decorrer do trabalho foram utilizados dois métodos para a detecção dos *spike bursts*: um método semi-automatizado e um método misto entre manual e automático. Ambos os métodos dependiam de uma escolha prévia do operador dos trechos a serem analisados, os quais (esperava-se) continham

exemplos de atividade do trato GI. Estes métodos serão detalhados nas seções seguintes.

Em resumo, o primeiro método objetivava detectar os *bursts* a partir de uma escolha inicial do operador num dos canais de aquisição de dados e encontrar os sinais correspondentes nos demais canais, utilizando como métrica a correlação cruzada entre a transformada do sinal escolhido pelo operador e as transformadas do mesmo trecho nos outros canais de aquisição. A velocidade de propagação do sinal naquele trecho era então calculada a partir das coordenadas encontradas.

Já no segundo método eram realizadas manualmente a verificação da propagação dos sinais entre os diversos canais e o cálculo da velocidade no trecho escolhido. Este procedimento permitiu encontrar-se por inspeção uma velocidade média e um padrão médio da transformada do *burst* para estes dados.

O motivo inicial desta dualidade foi tentar comparar a detecção automatizada com a manual. Porém, logo ficou claro que o primeiro método, embora fosse mais interessante do ponto de vista dos objetivos esperados, não era capaz de detectar adequadamente os sinais desejados. Na verdade, embora os resultados iniciais fossem promissores, logo ficou claro que a detecção automatizada era falha. As possibilidades envolvidas nestes problemas encontram-se no capítulo 5.

Apresentam-se aqui os dois métodos pois espera-se que futuramente, de posse dos resultados obtidos com o segundo método, seja possível ajustar o primeiro método de modo a automatizar a detecção tanto do trecho a ser analisado como dos próprios *spike bursts*.

4.1.1 Método semi-automatizado

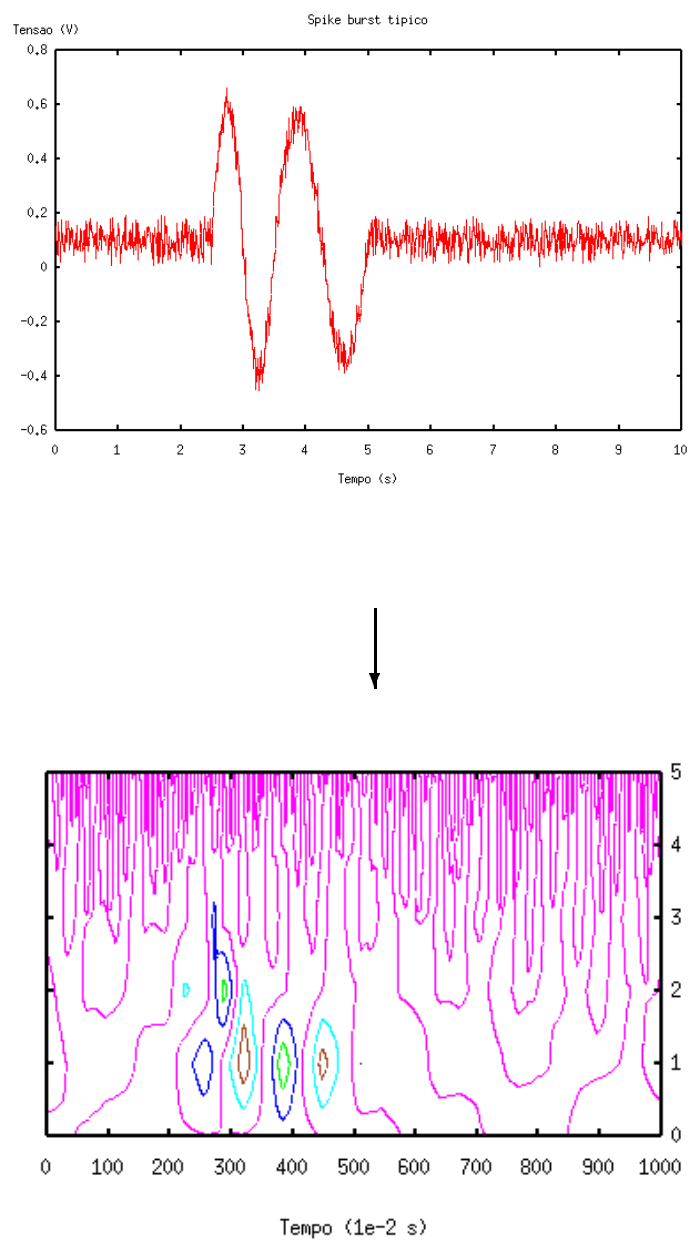
O método semi-automatizado utilizado para calcular a velocidade de propagação consiste, em resumo, dos seguintes passos:

1. Escolhe-se um *spike burst* observável por inspeção de algum dos canais de aquisição de dados;
2. em seguida, utilizando-se o programa **wavelyze** (seção A.8), os sinais de cada canal são transformados em seu plano *wavelet* e os limites temporais e frequenciais do padrão do *spike burst* são delimitados (figura 4.1);
3. o programa então tenta detectar automaticamente o mesmo padrão nos demais canais (que podem ser restringidos durante a análise para evitar-se os canais cujos eletrodos apresentem algum problema), conseguindo-se os tempos de atraso referentes a cada canal;
4. finalmente, a velocidade de propagação do sinal é calculada por uma regressão linear utilizando-se os tempos do item anterior.

Os vários valores da velocidade obtidos para cada leitura do sinal serão então plotados como função do tempo decorrido desde a primeira leitura realizada no animal, para verificar-se qual a influência do reestabelecimento do padrão digestivo do animal pós preparação cirúrgica na ocorrência e características dos *spike bursts*.

4.1.2 Método manual

O método manual utilizado para calcular a velocidade de propagação consiste quase que nos mesmos passos básicos do método semi-automatizado:

Figura 4.1: Transformada *Wavelet* Típica de um *Spike Burst*

1. Escolhe-se um *spike burst* observável por inspeção de algum dos canais de aquisição de dados, utilizando-se o programa **show_rato** (seção A.3);
2. adiciona-se o trecho onde aparece o *spike burst* a um arquivo de dados;
3. em seguida, utilizando-se o programa **trechos2wave** (seção A.6), os sinais de cada canal são transformados utilizando-se *wavelets*;
4. utilizando-se o programa **show_wave** (seção A.4), visualiza-se a transformada simultaneamente em todos os canais de aquisição e adicionam-se os valores das coordenadas temporais dos padrões identificados a um segundo arquivo de dados;
5. finalmente, a velocidade de propagação do sinal é calculada por uma regressão linear pelo programa **tr2vel** (seção A.5) utilizando-se os tempos do item anterior.

4.2 Cálculo computacional da transformada *Wavelet*

Hoje em dia, já estão disponíveis muitos pacotes de programas e bibliotecas computacionais utilizados para o cálculo da transformada *wavelet* e suas análises correlatas [35, 36, 37, 38]. Um destes pacotes é a própria biblioteca de funções para *wavelets* desenvolvido para ambiente MATLAB pela Mathworks [39]. Outro é a biblioteca de funções WAVELAB [40] para ambiente MATLAB, desenvolvido na Universidade de Stanford e que pode ser utilizado para fins não-comerciais sem custo.

Além disso, existem muitos programas em várias linguagens como C, C++ e Java, disponíveis tanto com como sem custo na Internet [41, 42, 43,

44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58].

No início deste projeto não estava ainda disponível o pacote da Mathworks, portanto optamos pelo uso do pacote WVELAB, ainda que não tão avançado como o primeiro.

Utilizando-se estas funções, torna-se bastante simples implementar computacionalmente uma transformada *wavelet* para os sinais unidimensionais utilizados. Estes programas estão listados no apêndice A.

4.3 Cálculo da frequência basal do sinal

Uma vez escolhidos os trechos com atividade mioéletrica, torna-se simples obter um valor para a frequência basal naquele trecho, através da análise da densidade espectral de potência.

Um estimador da densidade espectral de potência de um sinal na frequência ξ é dada por

$$\mathcal{P}(\xi) = \hat{f}(\xi) * \overline{\hat{f}}(\xi), \quad (4.1)$$

onde \hat{f} é a transformada de Fourier do sinal e o traço denota o conjugado da função.

Basta então verificar onde ocorrem os máximos locais desta densidade para obterem-se as frequências dominantes do sinal. Dentro do ambiente MATLAB, tornou-se muito simples implementar esta função no programa **fbasal** (seção A.1).

4.4 Apresentação dos resultados obtidos

Como já foi dito, o método semi-automatizado mostrou-se incapaz de prover resultados confiáveis para os valores procurados de velocidade de propagação

entre canais. Assim, foi utilizado o método manual, ainda que muito mais trabalhoso. Assim, cada um dos 82 arquivos foi analisado conforme o método manual, resultando na escolha de 345 trechos. Estes trechos foram então verificados para a ocorrência de padrões coerentes de *wavelets*. A figura 4.2 mostra o sinal temporal num dos trechos escolhidos, para 4 dos 6 canais analisados (os demais não eram aproveitáveis); a figura 4.3 mostra a transformada destes canais para o plano *wavelet*.

Por segurança, sempre eram escolhidos mais de três trechos por arquivo, pois alguns destes trechos apresentavam na verdade padrões de *wavelets* típicos de um transiente, com todos os canais apresentando um padrão de impulso no mesmo ponto temporal. A figura 4.4 mostra o sinal temporal num destes casos; a figura 4.5 mostra a transformada *wavelet* deste transiente. Nestes casos o trecho não pôde ser aproveitado para análise.

Descontados os trechos escolhidos erroneamente, foram analisados no total 314 trechos.

Após esse desconto, foi calculada a densidade espectral de potência para cada um dos trechos a fim de obter a frequência basal para o mesmo. Verificou-se que na grande maioria dos casos eram discerníveis duas frequências próximas como dominantes no sinal. A figura 4.6 mostra uma destas análises. Indica-se com uma seta (\nearrow) a localização dos picos.

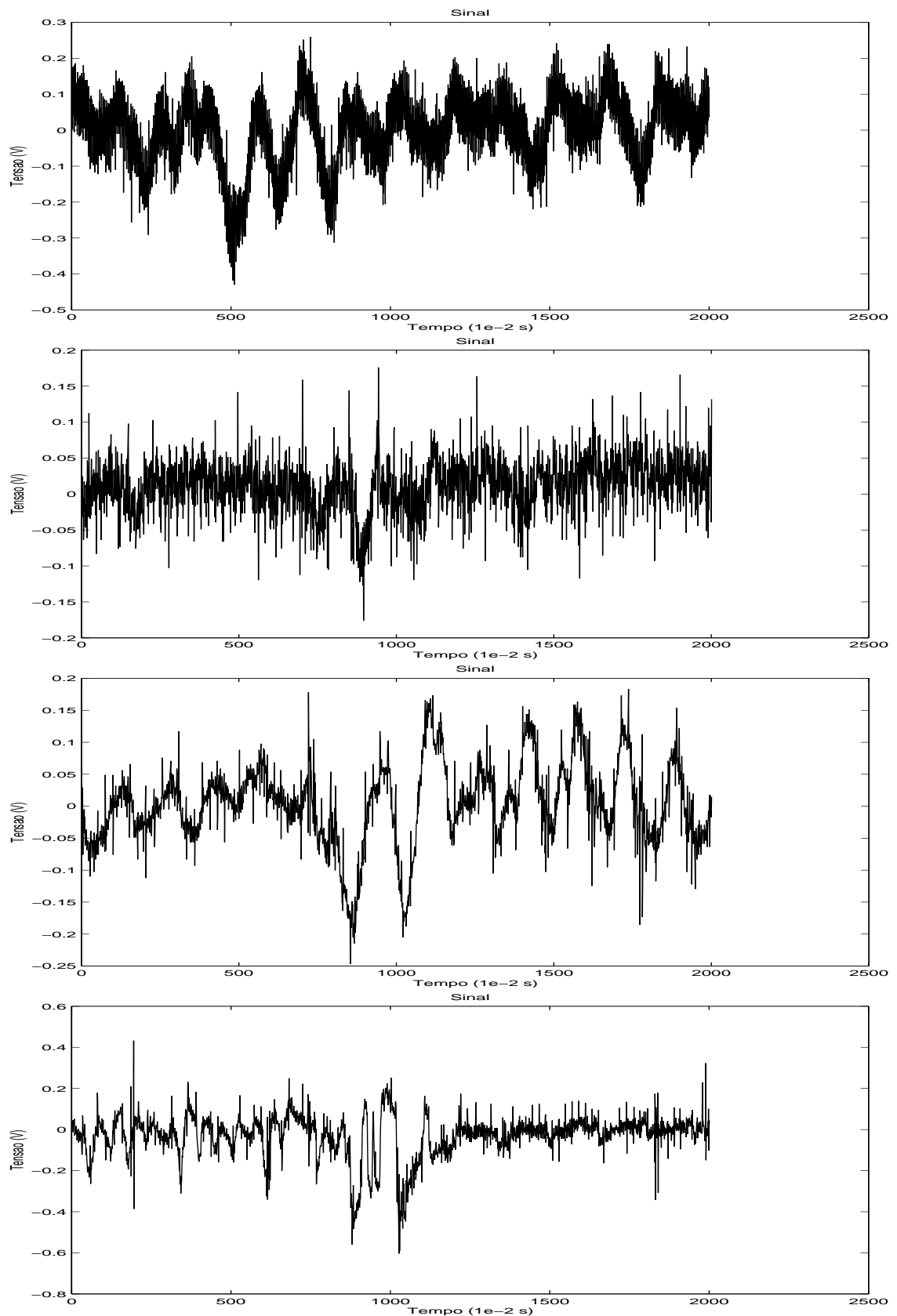


Figura 4.2: Seqüência de Canais Mostrando Propagação num Trecho

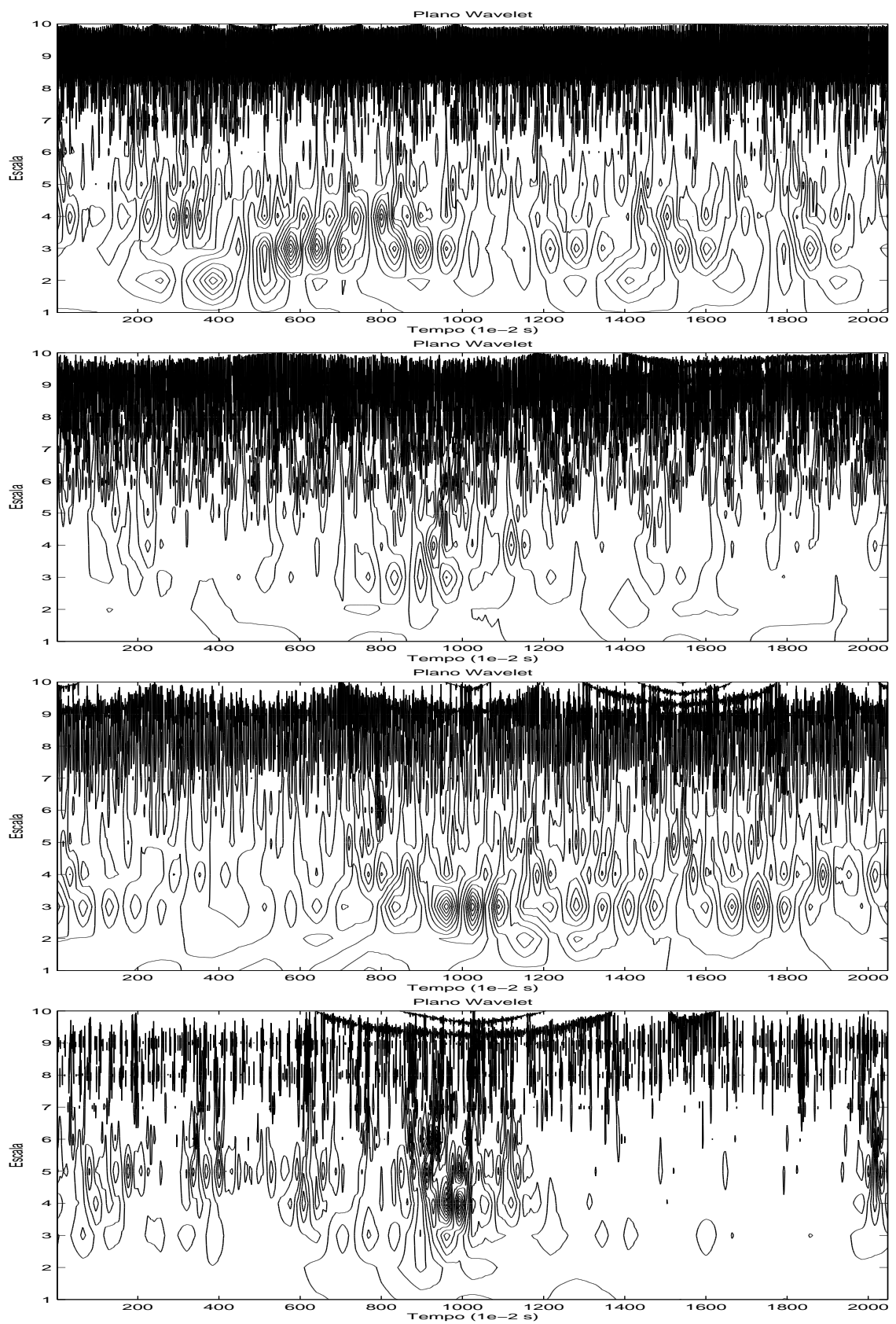


Figura 4.3: Transformada *Wavelet* do Trecho da Figura 4.2

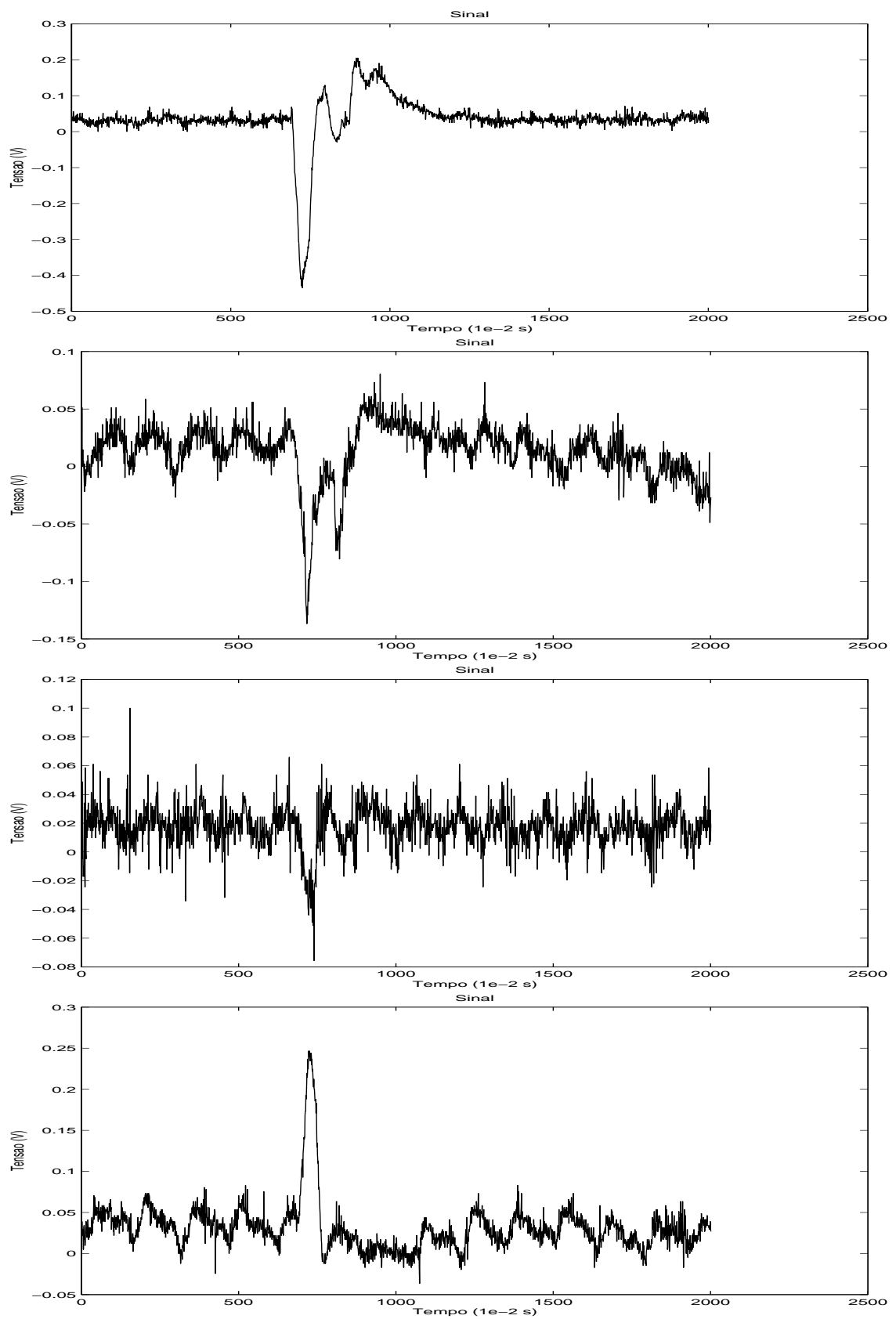


Figura 4.4: Trecho com Transiente

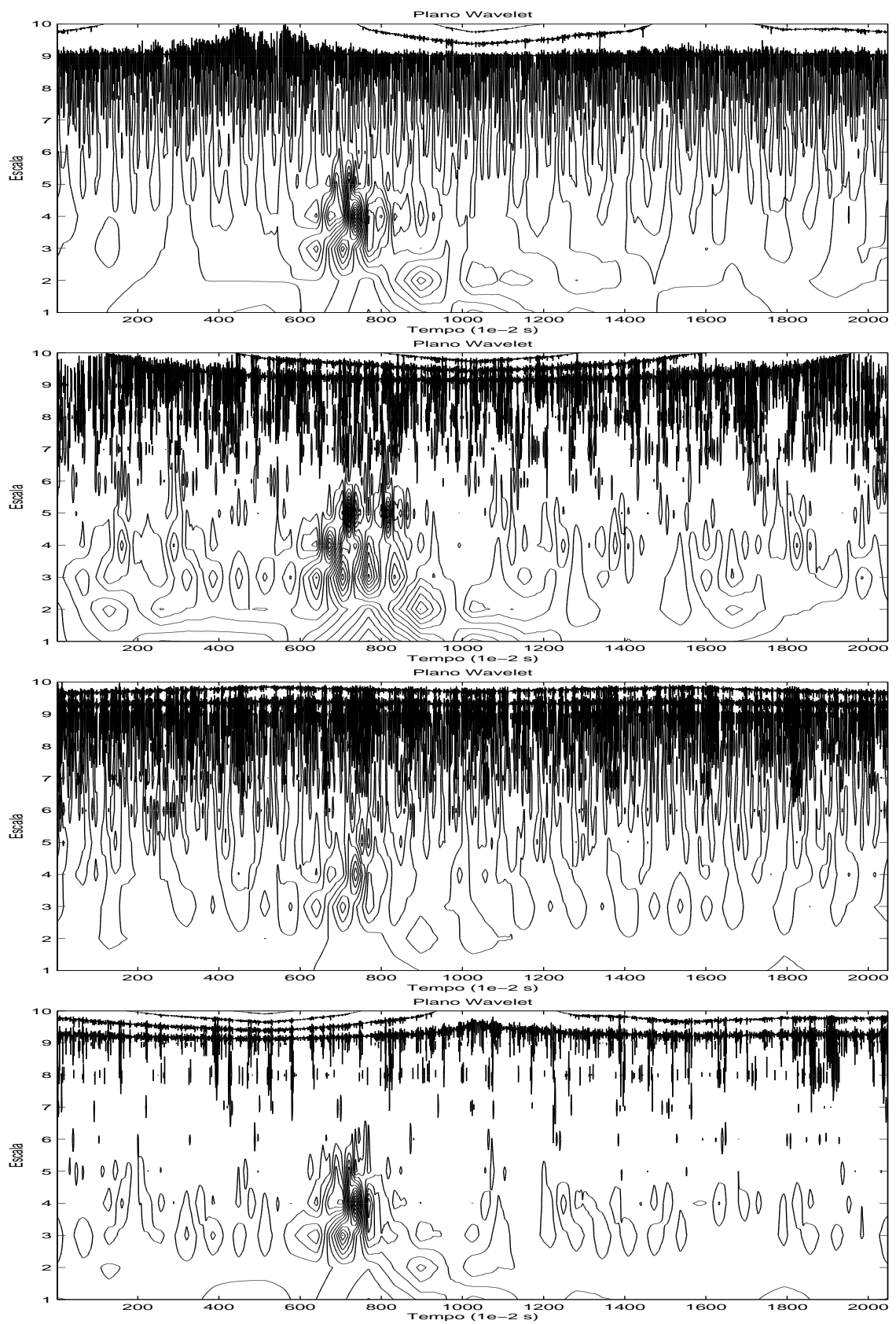


Figura 4.5: Transformada *Wavelet* do Trecho da Figura 4.4

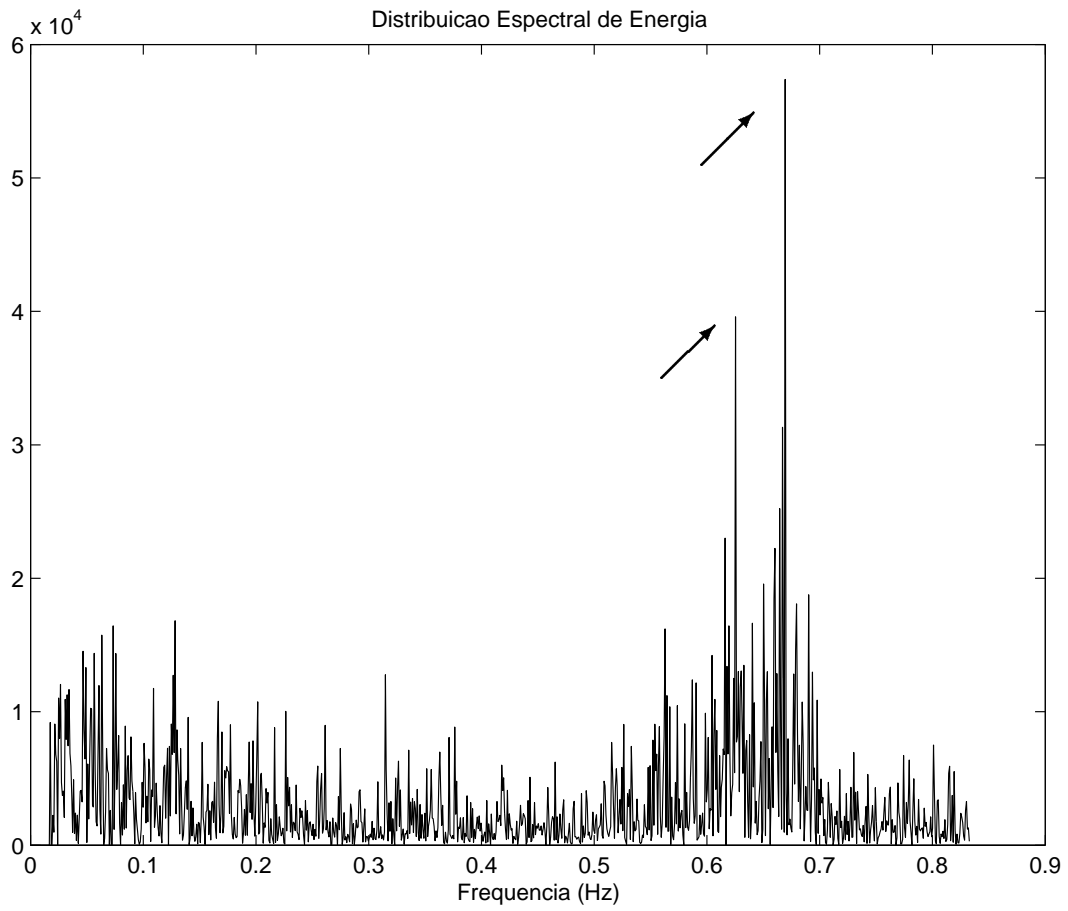


Figura 4.6: Densidade Espectral de Energia para um dos Trechos

As tabelas 4.1 e 4.2 resumem todo este trabalho, mostrando para cada leitura e animal a velocidade média e frequência basal encontradas. A média global encontrada para a velocidade foi de 1.2 ± 0.3 cm/s e para a frequência basal foi de 0.63 ± 0.02 Hz para a menor e de 0.65 ± 0.02 Hz para a maior.

Mostra-se em seguida (figuras 4.7 até 4.13) a evolução dos valores em relação às leituras para cada um dos animais.

Como se pode notar, a passagem do tempo entre as leituras não parece influenciar a medida da velocidade de propagação dos sinais, não sendo possível retirar qualquer conclusão desta evolução.

Tabela 4.1: Valores da Velocidade e Frequência Basal – Animais 063 a 067

Animal	Leitura	Velocidade Média (cm/s)	Frequências Basais (Hz)	
063	001	0.69	0.63 / 0.67	
	002	1.05	0.68 / 0.69	
	003	1.01	0.66 / 0.67	
	004	0.77	0.68 / 0.70	
	005	1.18	0.64 / 0.65	
	006	1.64	0.65 / 0.68	
	007	0.79	0.65 / 0.66	
064	001	1.19	0.66 / 0.69	
	002	0.90	0.65 / 0.67	
	003	1.29	0.70 / 0.71	
065	002	1.48	0.66 / 0.68	
	003	1.03	0.65 / 0.65	
	004	1.31	0.62 / 0.64	
	005	0.88	0.69 / 0.69	
	006	1.62	0.65 / 0.67	
	007	1.08	0.62 / 0.63	
	008	1.37	0.61 / 0.62	
	009	2.47	0.65 / 0.66	
	011	1.32	0.60 / 0.61	
	012	1.06	0.60 / 0.60	
	066	001	0.79	0.65 / 0.66
		002	1.24	0.63 / 0.64
003		1.39	0.63 / 0.64	
004		1.03	0.61 / 0.62	
005		1.61	0.63 / 0.64	
006		1.12	0.61 / 0.68	
007		1.30	0.62 / 0.63	
008		0.67	0.63 / 0.65	
009		0.79	0.62 / 0.63	
010		1.02	0.61 / 0.65	
011		0.71	0.60 / 0.62	
012		1.18	0.60 / 0.62	
067	001	1.05	0.62 / 0.63	
	002	0.96	0.61 / 0.64	
	003	0.80	0.62 / 0.65	
	004	1.50	0.64 / 0.70	
	005	1.06	0.65 / 0.62	
	006	0.63	0.63 / 0.66	
	007	1.05	0.61 / 0.63	
	008	1.25	0.62 / 0.63	
	009	0.50	0.61 / 0.64	
	010	1.70	0.60 / 0.65	
	011	1.02	0.60 / 0.63	
	012	1.15	0.62 / 0.63	

Tabela 4.2: Valores da Velocidade e Freqüência Basal – Animais 068 a 071

Animal	Leitura	Velocidade Média (cm/s)	Freqüência Basal (Hz)
068	001	1.11	0.63 / 0.65
	002	2.55	0.66 / 0.67
	003	2.00	0.62 / 0.63
	004	0.61	0.61 / 0.64
	005	1.40	0.60 / 0.62
	006	1.31	0.63 / 0.63
	007	1.15	0.60 / 0.64
	008	1.01	0.61 / 0.64
	009	1.16	0.65 / 0.67
	010	1.02	0.63 / 0.65
	011	1.53	0.60 / 0.60
	012	0.63	0.64 / 0.65
069	002	1.62	0.63 / 0.64
	003	0.82	0.62 / 0.64
	004	0.90	0.61 / 0.62
070	001	0.98	0.65 / 0.66
	003	1.36	0.62 / 0.63
	004	0.72	0.60 / 0.62
	005	1.12	0.64 / 0.68
	006	0.97	0.60 / 0.65
	007	1.04	0.64 / 0.66
	008	1.16	0.62 / 0.69
	009	1.47	0.63 / 0.66
	010	1.18	0.64 / 0.66
	011	2.10	0.60 / 0.62
	012	1.52	0.65 / 0.67
	071	001	1.23
002		1.05	0.62 / 0.65
003		1.41	0.61 / 0.64
004		1.14	0.65 / 0.66
005		1.11	0.66 / 0.68
006		1.05	0.61 / 0.63
007		1.43	0.65 / 0.66
008		1.25	0.62 / 0.63
009		0.89	0.61 / 0.63
010		1.32	0.61 / 0.62
011		0.75	0.65 / 0.66
012		1.18	0.65 / 0.66

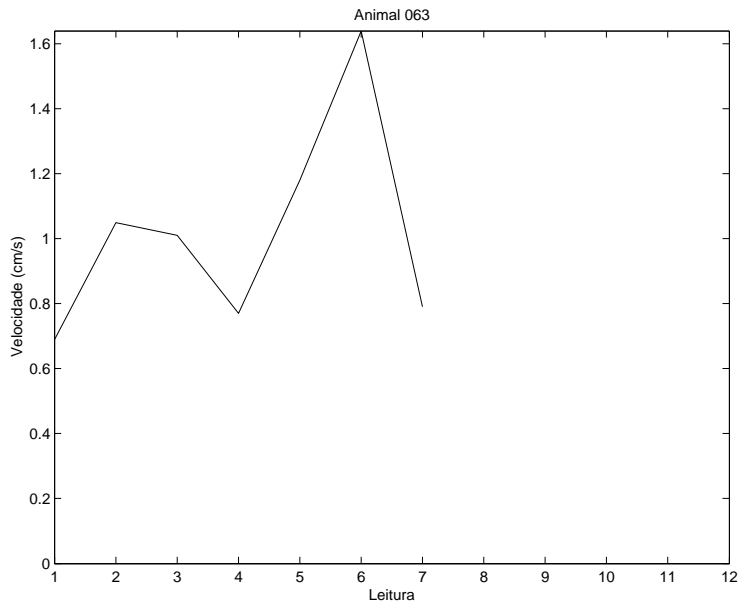


Figura 4.7: Evolução das Velocidades para o Animal 063

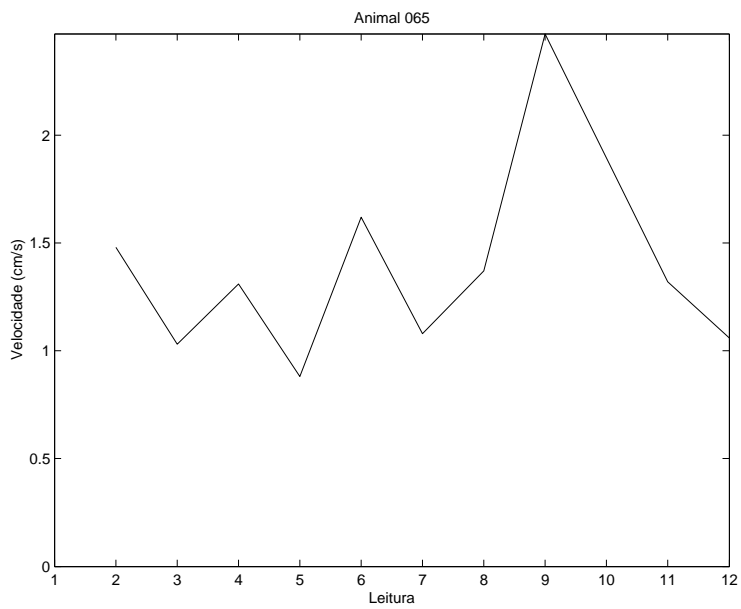


Figura 4.8: Evolução das Velocidades para o Animal 065

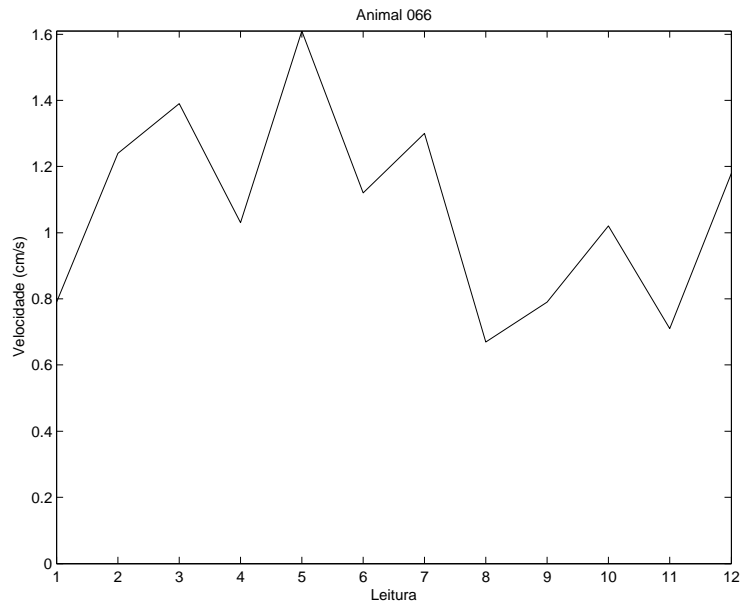


Figura 4.9: Evolução das Velocidades para o Animal 066

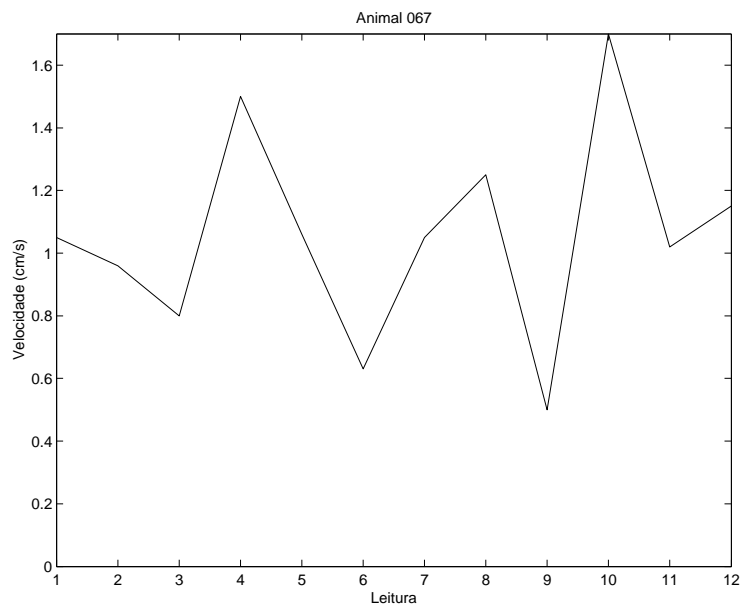


Figura 4.10: Evolução das Velocidades para o Animal 067

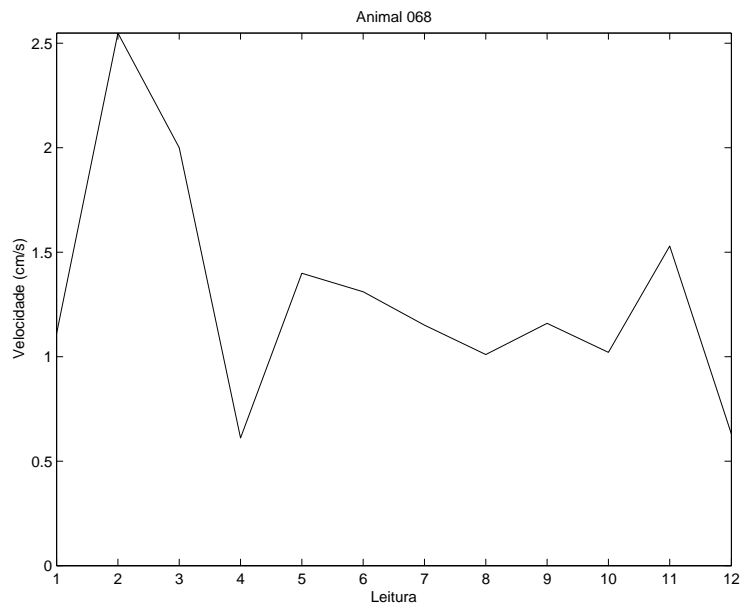


Figura 4.11: Evolução das Velocidades para o Animal 068

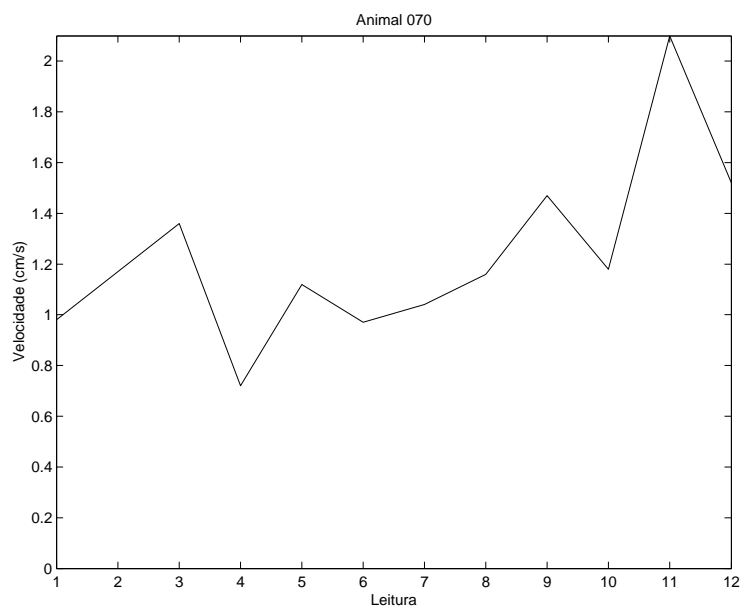


Figura 4.12: Evolução das Velocidades para o Animal 070

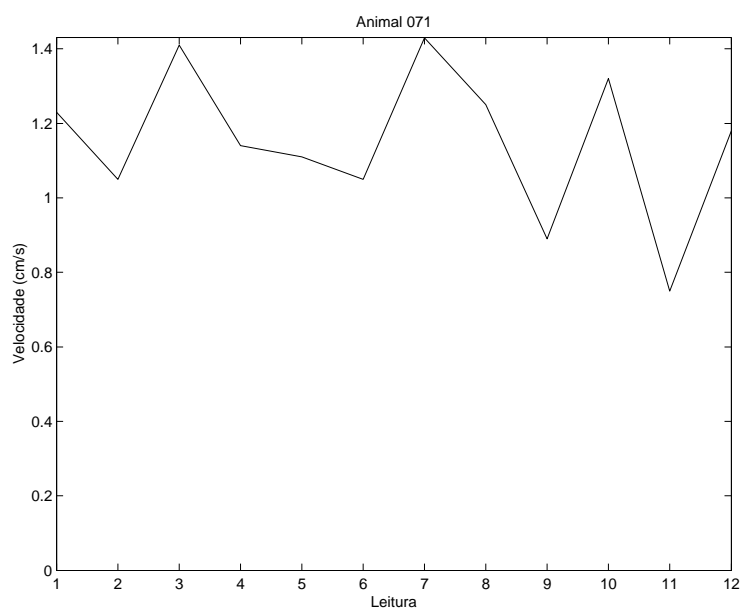


Figura 4.13: Evolução das Velocidades para o Animal 071

Capítulo 5

Discussão

O primeiro ponto a ser debatido é obviamente o porquê da análise semi-automatizada não ter conseguido detectar os padrões procurados, tornando necessário o recurso de analisar-se manualmente os arquivos de dados.

Embora não se tenha ainda uma hipótese satisfatória, a causa provável para este resultado encontra-se na relação sinal-ruído dos arquivos analisados. De fato, mesmo depois de muita experiência em analisar este tipo de sinal, é fácil observar que existem muitos tipos de ruídos que podem desestabilizar a procura através de correlação cruzada, como os erros na escolha dos trechos a serem analisados demonstram.

Porém, embora a detecção automática tenha falhado, os resultados encontrados sugerem que a utilização de *wavelets* para determinação das propriedades de propagação dos sinais no trato GI é uma alternativa viável e até, como sugerem alguns estudos recentes [15, 16], superior à determinação destes mesmos padrões pelos métodos clássicos utilizando função de transferência.

Realmente, embora com um desvio-padrão alto (cerca de 25% do valor), foi-se capaz de sistematicamente encontrar os padrões dos *spike bursts* na

maioria dos trechos analisados e conseguir-se uma estimativa de 1.2 cm/s para a velocidade de propagação. Pesquisas na literatura não trouxeram outras estimativas estabelecidas desta velocidade de propagação, portanto torna-se difícil a comparação com valores conhecidos, embora acredite-se que é uma estimativa bastante realista.

Um dos fatores que pode ser melhorado é a *wavelet* específica escolhida para a transformada. Em todo este trabalho utilizamos uma *wavelet* de Daubechies de ordem 6, seguindo a recomendação de parâmetros constante no pacote WAVELAB. Entretanto, como citado [21, 30, 31, 32], existem literalmente milhares de famílias de *wavelets* disponíveis, e poder-se-ia considerar o estudo de outros tipos de *wavelet* na tentativa de obter um melhor resultado, utilizando por exemplo *wavelets* não simétricas para detecção de bordas.

Quanto à frequência basal, o valor encontrado de 0.63 / 0.65 Hz é perfeitamente compatível com o valor normalmente atribuído a este parâmetro [16]. O fato mais interessante foi a identificação do *doublet* de frequências envolvido na onda basal, o que normalmente não é citado, atribuindo-se apenas um valor a esta frequência.

5.1 Análise de precisão dos resultados

Com certeza, os maiores fatores de erro na análise em si estão relacionados com a precisão com a qual consegue-se identificar os padrões no plano *wavelet* e suas respectivas posições. Utilizando-se um *grid* para identificar estas posições, avaliou-se que a resolução exequível é de cerca de 2.5% da escala utilizada (50 pontos quando o intervalo é de 2048 pontos). Essa era a escala onde era melhor visualizada a propagação dos sinais no plano *wavelet*,

permitindo a análise de cada trecho.

Considerando-se que o valor do desvio-padrão encontrado é substancialmente maior do que isso, estando em 25% do valor encontrado, pode-se supor que os erros experimentais ao adquirir os dados são o fator predominante na variabilidade dos resultados, sendo que a análise não introduz, por si só, erros significativos.

Quanto à frequência basal, o pequeno desvio padrão (3%) dos valores entre todas as amostras e, além disso, a concordância com a literatura, sugere que este parâmetro está corretamente avaliado.

Em termos da análise estatística do intervalo de confiança [59], pode-se afirmar com 99% de confiança que a média da velocidade de propagação das demais leituras, quando forem analisadas, estará dentro do intervalo 1.09 e 1.31 cm/s. Estes valores, dentro do desvio-padrão das amostras, sugerem que o método utilizado é estatisticamente consistente.

Para a frequência basal, este intervalo, também com confiança de 99%, é de 0.624 a 0.635 para a frequência menor e de 0.644 a 0.655 Hz para a frequência maior.

Capítulo 6

Conclusões

Após a análise de 314 trechos de 82 arquivos de leitura, foi encontrada uma média geral da velocidade de propagação de 1.2 ± 0.3 cm/s e das componentes principais da frequência basal de 0.63 ± 0.02 Hz e 0.65 ± 0.02 Hz, com uma confiança estatística de 99% ($P < 0.01$) dos valores encontrarem-se dentro do desvio-padrão.

Assim, o tipo de análise proposta foi capaz de localizar a propagação dos impulsos mioelétricos na camada seromuscular, permitindo uma análise alternativa aos métodos clássicos de determinar-se as características de motilidade do trato GI.

Entretanto, como já foi mencionado, é necessário encontrar-se métodos automatizados para realizar este trabalho, pois a análise manual dos arquivos, como foi feita, não só é extremamente trabalhosa como dependente do conhecimento e experiência do operador que a está realizando.

6.1 Proposta de continuidade

Após os resultados obtidos neste trabalho, são os seguintes os caminhos apontados como continuação natural para futuras explorações do assunto:

- corrigir e melhorar o método semi-automatizado, utilizando os valores encontrados de velocidade como parâmetro e/ou utilizar outros métodos de detecção de padrões (redes neurais, por exemplo) a fim de comparar os resultados obtidos com a correlação cruzada;
- continuar a série de experiências para os outros tipos de perturbação digestivas que se propunha estudar quando do início do trabalho para determinar como estas modificações do trato GI alteram os valores encontrados;
- melhorar significativamente os algoritmos computacionais utilizados, já que cada análise pelo método semi-automatizado durava cerca de 1h30min numa estação de trabalho UltraSparc 5, passando algumas das funções mais numericamente intensivas para uma linguagem de baixo nível como C ou Java;
- identificar séries complexas dos *spike bursts* como determinantes de outros fenômenos associados à motilidade do trato GI.

Referências

- [1] I. Daubechies, *Ten Lectures on Wavelets*. SIAM, 1992.
- [2] J. S. Walker, “Fourier analysis and wavelet analysis,” *Notices of the AMS*, vol. 44, pp. 658–670, July 1997.
- [3] P. A. Morettin, *Ondas e Ondaletas*. EDUSP, 1999.
- [4] C. K. Chui, *An Introduction to Wavelets*. Academic Press, 1992.
- [5] J. Chen and R. McCallum, “Clinical applications of electrogastrography,” *American Journal of Gastroenterology*, vol. 88, pp. 1324–1336, September 1993.
- [6] F. P. Lopasso, F. Kassab Jr., and R. Marques, *Projeto de Motilidade do Aparelho Digestivo*. Relatório Final apresentado à FAPESP, Protocolo 5463/96: Laboratório de Automação e Controle da Escola Politécnica da USP, Julho 1996.
- [7] F. P. Lopasso, F. Kassab Jr., and R. Marques, “Analyzer: Manual do usuário,” [6].
- [8] The MathWorks, Inc., *Matlab User’s Guide*, 1994.
- [9] M. M. de Assumpção Filho, ed., *MATLAB Versão do Estudante: Guia do Usuário*. Makron Books, 1997.

- [10] F. P. Lopasso, F. Kassab Jr., R. Marques, A. Bonizzia, A. A. S. Ponce, A. A. Laudanna, and J. P. E. Costa, "Myoelectric record of digestive tract of rats with chronic implanted electrodes," *Anais da Academia Brasileira de Ciências*, vol. 68, pp. 260–261, 1996.
- [11] Z. Lin and J. Chen, "Advances in time frequency analysis of biomedical signals," *Critical Reviews in Biomedical Engineering*, vol. 24, no. 1, 1996.
- [12] A. Figliola and E. Serrano, "Analysis of physiological time series using wavelet transforms," *IEEE Engineering in Medicine and Biology*, pp. 74–79, June 1997.
- [13] R. Coifman and M. V. Wickerhauser, "Adapted wavelet de-noising for medical signals and images," *IEEE Engineering in Medicine and Biology*, vol. 14, pp. 578–586, September 1995.
- [14] M. A. Westenberg and J. Roerdink, "X-ray volume rendering through two-stage splatting," *Machine Graphics and Vision*, vol. 9, pp. 307–314, May 2000.
- [15] A. Akin and H. H. Sun, "Time-frequency methods for detecting spike activity of stomach," *Med Biol Eng Comput*, vol. 37, pp. 381–90, May 1999.
- [16] W. Qiao, H. H. Sun, W. Y. Chey, and K. Y. Lee, "Continuous wavelet analysis as an aid in the representation and interpretation of electro-gastrographic signals," *Ann Biomed Eng*, vol. 26, pp. 1072–1081, Nov 1998.
- [17] P. J. Durka and K. J. Blinowska, "Matching pursuit parametrization of sleep spindles," in *Abstracts of 18th Annual International Conference*

- of the IEEE EMBS*, November 1996. Laboratory of Medical Physics, Warsaw University.
- [18] L. G. Weiss, “The application of wavelet transforms to blood flow velocimetry,” in *Wavelets in Medicine and Biology* (A. Aldroubi and M. Unser, eds.), ch. 21, CRC Press, 1996.
- [19] P. J. Durka, *Time-frequency Analyses of EEG*. PhD thesis, Institute of Experimental Physics, Department of Physics, Warsaw University, August 1996.
- [20] A. R. Tate, *Pattern Recognition Analysis of In Vivo Magnetic Resonance Spectra*. PhD thesis, University of Sussex, September 1996.
- [21] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [22] B. Jawerth and W. Sweldens, “An overview of wavelet based multiresolution analyses,” tech. rep., Department of Mathematics, University of South Carolina, 1993.
- [23] Y. Meyer, *Wavelets—Algorithms and Applications*. SIAM, 1992.
- [24] W. Sweldens, “Wavelets and the lifting scheme: A 5 minute tour,” tech. rep., Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995.
- [25] W. Sweldens and P. Schröder, “Building your own wavelets at home,” tech. rep., Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995.
- [26] K. C. McGill and C. Taswell, “Wavelet transform algorithms for finite-duration discrete-time signals,” in *Progress in Wavelet Analysis and Ap-*

- plications* (Y. Meyer and S. Roques, eds.), pp. 221–224, Editions Frontières, 1993. Proceedings of the International Conference on Wavelets and Applications, Toulouse France, June 1992.
- [27] C. Taswell, “Algorithms for the generation of Daubechies orthogonal least asymmetric wavelets and the computation of their Holder regularity,” tech. rep., Scientific Computing and Computational Mathematics, Stanford University, Aug. 1995.
- [28] H. N. Kritikos, “Signal processing in the time-frequency domain.” Notas de Aula para Curso de Pós-graduação ministrado na EPUSP, 1997.
- [29] C. Taswell, “The what, how, and why of wavelet shrinkage denoising,” Tech. Rep. CT-1998-09, Computational Toolsmiths, Aug. 1998.
- [30] A. Graps, “An introduction to Wavelets,” *IEEE Computational Science and Engineering*, vol. 2, no. 2, 1995.
- [31] W. Sweldens and R. Piessens, “Wavelet sampling techniques,” in *Proceedings of the Joint Statistical Meetings*, August 1993. submitted.
- [32] B. Deng, B. Jawerth, , G. Peters, and W. Sweldens, “Wavelet probing for compression based segmentation,” in *Proceedings of SPIE Conference*, July 1993. submitted.
- [33] W. Sweldens, “Wavelets: What next?,” tech. rep., AT&T Bell Laboratories, 1996.
- [34] B. W. Kernighan and D. M. Ritchie, *C - A Linguagem de Programação*. EDISA, 1986.
- [35] C. Taswell, “MATLAB wavelet toolbox,” *Wavelet Digest*, vol. 2, July 1993. Topic #7.

- [36] C. Taswell and K. C. McGill, “Algorithm 735: Wavelet transform algorithms for finite-duration discrete-time signals,” *ACM Transactions on Mathematical Software*, vol. 20, pp. 398–412, Sept. 1994.
- [37] C. Taswell, “Wavelet transform convolution versions for fast wavelet based numerical algorithms,” in *Abstracts of Talks at Conference on Wavelets, Relations with Operators and Applications* (X. Dai, ed.), p. 27, July 1996. available from xdai@uncc.edu.
- [38] C. Taswell, “Wavelet transform compression of functional magnetic resonance image sequences,” in *Proceedings of the IASTED International Conference on Signal and Image Processing*, pp. 725–728, Oct. 1998. Paper #281-162.
- [39] “Matlab Wavelet Toolbox.” <http://www.mathworks.com/products/wavelet/>.
- [40] J. Buckheit *et al.*, *Wavelab Reference Manual*, December 1995.
- [41] “Image Wavelet compressor.” <http://www.barrt.ru/parshukov>.
- [42] “University of California Berkeley Wavelet Group.” <http://gabor.eecs.berkeley.edu/>.
- [43] “California Institute of Technology Multiresolution Modelling Group.” <http://www.multires.caltech.edu/>.
- [44] “Lawrence Livermore National Laboratories Wavelet Page.” <http://www.llnl.gov/das/wavelet/wavelet.html>.
- [45] “Bell Labs Wavelet Group.” <http://cm.bell-labs.com/cm/ms/who/jelena/Wavelet/>.

- [46] “Massachusetts Institute of Technology: Department of Mathematics.”
<http://www-math.mit.edu/~gs/>.
- [47] “Wavelet resources.” <http://www.amara.com/current/wavelet.html>.
- [48] “Mathsoft wavelet resources.” <http://www.mathsoft.com/-wavelets.html>.
- [49] “Wavelet Digest.” <http://www.wavelet.org/wavelet/index.html>.
- [50] “Courses in applied wavelet analysis.” <http://www.wavelets.com/>.
- [51] “Practical guide to wavelet analysis.” <http://paos.colorado.edu/research/wavelets/>.
- [52] “Wavelet NetCare.” <http://www.math.wustl.edu/wavelet/>.
- [53] “Bibliographies on wavelets.” <http://iinwww.ira.uka.de/bibliography/-Theory/Wavelets/>.
- [54] “FBI fingerprint compression standard.” <http://www.c3.lanl.gov/~brislawn/FBI/FBI.html>.
- [55] “Discovering wavelets.” <http://www.gvsu.edu/mathstat/wavelets.htm>.
- [56] “Wavelet tutorial.” <http://www.public.iastate.edu/~rpolikar/-WAVELETS/waveletindex.html>.
- [57] “Yale wavelet resources.” <http://www.math.yale.edu/wavelets/>.
- [58] “A really friendly guide to wavelets.” <http://perso.wanadoo.fr/polyvalens/clemens/wavelets/wavelets.html>.
- [59] E. S. Berquó, J. Souza, and S. Gotlieb, *Bioestatística*. EPU, 1981.

- [60] L. Lamport, *L^AT_EX - A Document Preparation System*. Addison Wesley, 1994.
- [61] M. Goosens, F. Mittelbach, and A. Samarin, *The L^AT_EX Companion*. Addison Wesley, 1993.

Apêndice A

Listagem dos programas utilizados

Os programas listados a seguir foram desenvolvidos para uso em ambiente MATLAB, utilizando a biblioteca de funções WAVELAB.

A.1 fbasal.m

```
%% FBASAL   Calcula e mostra frequencia basal de cada trecho

% Arquivos com os trechos
load ratos/trechos.txt;
oldrato=0;
oldt=0;

disp(['Tenho ' int2str(size(trechos,1)) ' trechos a analisar.']);
for arq = 1:size(trechos,1);
```

```
rato = trechos(arq,1);
t     = trechos(arq,2);
nome  = sprintf("rato%03d/n%03dt%03d", rato, rato, t);

%% Carrega o arquivo correspondente ao trecho, se necessario
if rato ~= oldrato | t ~= oldt,
disp(['Carregando ' nome ]);
eval(['load ratos/' nome]);
end;

%% Pega as duas frequencias preponderantes
    primax=[];
    secmax=[];
for i=1:6;
    fb=fft(Y(trechos(arq,3) * 1000:trechos(arq,4) * 1000,i));
    spect= fb .* conj(fb);

    %% Elimina media
    spect(1:5) = 0;
    [x primax(i)] = max(spect);
    spect((primax(i)-2):(primax(i)+2)) = 0;
    [x secmax(i)] = max(spect);
end;
disp([ (median(primax) ./ (size(spect) / 100))   median(secmax) ./ (size(spect)
oldrato = rato;
oldt     = t;
end;
```

A.2 prepara.m

```
function prepara(arquivo);  
% Carrega arquivo  
clear global Y T;  
clear CHNAME CHNUM EV Y T;  
global Y T;  
eval(['load /oca/biomat/marcelof/Novos/todos/' arquivo]);
```

A.3 show_ratios.m

```
%% SHOW_RATIOS Carrega e mostra os arquivos dos ratos, com pausas  
%% para determinar os trechos a serem analisados.  
  
incr=3000;  
for arq = 1:size(arquivos,1);  
  
%% Carrega o arquivo e coloca na variavel Y  
disp(['Carregando ' arquivos(arq,:)]);  
eval(['load ' arquivos(arq,:) '.txt']);  
eval(['Y = ' arquivos(arq,9:20) '; clear ' arquivos(arq,9:20) ';'']);  
  
%% Mostra o arquivo com pausas  
loops = floor(size(Y,1) / incr);  
for steps=0:loops-1;  
for i=1:6;  
figure(i);  
start= (steps * incr) + 1;
```



```
stop = (steps * incr) + incr;
axis([ start, stop, -0.3, 0.3 ]);
plot( start:stop, Y( start:stop, i ) );
end;
input(['Mostrando de ' num2str(start) ' ate ' num2str(stop) ' ;']);
end;
clear Y;
end;
```

A.4 show_wave.m

```
%% SHOW_WAVE Carrega e mostra os trechos transformados por Wavelet, com pausas

clear;
axis();
title('');

% Arquivo com os trechos
load tab_wave_tr.txt;

for arq = 1:size(tab_wave_tr,1);

%% Carrega os trechos e mostra
rato = tab_wave_tr(arq,2);
t     = tab_wave_tr(arq,3);
base = sprintf('rato%03d/n%03dt%03dtr%d', rato, rato, t, arq);
b     = size(base,2);
```

```
for i=1:6;
eval(['load ' base 'ca' int2str(i) '.txt' ]);
eval(['sinal = ' base(9:b) 'ca' int2str(i) ';'']);
axis([ 0 max(size(sinal)) 0 7 ]);
gset size 1.2,1.2;
figure(i);
contour(sinal',15);
end;
input(['Carregando trecho ' int2str(arq) ]);
end;
axis();
title('');
```

A.5 tr2vel.m

```
%% TR2VEL Calcula velocidades baseadas nos trechos de cada rato

% Arquivos com os trechos
clear;
load tab_wave_tr.dat;
oldrato=0;
oldt=0;
count=0;
velocidades=[];
counter=0;
```

```
for arq = 1:size(tab_wave_tr,1);

rato = tab_wave_tr(arq,2);
t     = tab_wave_tr(arq,3);

%% Carrega o arquivo correspondente ao trecho, se necessario
if rato ~= oldrato | t ~= oldt,

    if oldrato ~= 0 & oldt ~= 0,
        disp([' Velocidade media: ' num2str(mean(vel)) ' cm/s'])
        counter = counter + 1;
        velocidades(counter) = mean(vel);
    end;

    disp(['Analisando rato ' int2str(rato) ', arquivo ' int2str(t)]);
    vel=[];
    count=0;
end;

ind = 0;
for i = 4:9;
    if tab_wave_tr(arq,i) ~= 0,
        ind = ind + 1;
        time(ind) = i-3;
        pos(ind) = tab_wave_tr(arq,i);
    end;
end;
```

```
if ind ~= 0,
    p = polyfit(time, pos, 1);
    count = count + 1;
    vel(count) = p(1) / 100;
end;

oldrato = rato;
oldt     = t;
end;

disp([' Velocidade media: ' num2str(mean(vel)) ' cm/s'])
counter = counter + 1;
velocidades(counter) = mean(vel);

disp('');
disp(['Velocidade media global: ' num2str(mean(velocidades)) ' cm/s'])
```

A.6 trechos2wave.m

```
%% TRECHOS2WAVE Calcula transformada wavelet dos trechos escolhidos dos
%% arquivos dos ratos e converte em arquivos de texto
%% para poder visualizar no octave.

% Arquivos com os trechos
load ratos/trechos.txt;
oldrato=0;
```

```
oldt=0;
L=2;
qmf=MakeONFilter('Daubechies',6);

disp(['Tenho ' int2str(size(trechos,1)) ' trechos a analisar.']);
for arq = 1:size(trechos,1);

rato = trechos(arq,1);
t     = trechos(arq,2);
nome  = sprintf("rato%03d/n%03dt%03d", rato, rato, t);

%% Carrega o arquivo correspondente ao trecho, se necessario
if rato ~= oldrato | t ~= oldt,
disp(['Carregando ' nome ]);
eval(['load ratos/' nome]);
end;

%% Faz com que o sinal tenha tamanho diadico
start = trechos(arq,3) * 1000;
fim   = trechos(arq,4) * 1000;
lsinal = 2 .^ ceil( log2(fim-start) );
stop = start + lsinal - 1;
disp(['   Analisando de ' int2str(start) ' ate ' int2str(stop) ]);

%% Calculo do Plano de Fase Wavelet para cada canal
for canal=1:6;
sinal=(Y( start:stop, canal)');
end;
```

```
W = FWT_PO(sinal,L,qmf);
Wgraf = ww_matr(W,L,0,qmf,'Contour');
eval(['save ratos/' nome 'tr' num2str(arq) 'ca' num2str(canal) '.txt Wgraf -asci
end;

oldrato = rato;
oldt     = t;
end;
```

A.7 ve_sinal.m

```
function ve_sinal(lim1,lim2);
%% Mostra o sinal original separado por canal

global Y T
b=min(size(Y));

figure(1);
clf;
for a=1:b
subplot(b,1,a);
plot(T(lim1:lim2),Y(lim1:lim2,a));
title(['Canal ',int2str(a)]);
end;
```

A.8 wavelyze.m

```
function velocidade=wavelyze();
%      WAVELYZE      Script para rodar uma analise completa do sinal.
%      lim1 e lim2 sao os limites a serem analisados.

global Y T;

% Parametros do Sinal
freq=1/( T(2) - T(1) );
torig=length(T);

% Prepara o sinal com tamanho diadico
% Se for "perto" do maximo, aumenta com zeros, senao corta
% Nao se supoe operacoes na parte zerada !!
sY = round(log2(max(size(Y))));
if( (2^sY) > max(size(Y)) )
Y = [ Y; zeros( ((2^sY) - max(size(Y))), min(size(Y)) ) ];
else
Y = Y(1:(2^sY),:);
end
T= (1:2^sY) ./ freq;

% Mostra o sinal dentro dos limites (todos os canais)
% '+' dobra o tamanho, '-' reduz 'a metade, <num> muda ponto de inicio
refaz = 1 ./ freq; faixa=2^sY;
while( refaz ~= 0 )
if(refaz == '+')
```

```
faixa = faixa * 2;
elseif(refaz == '-')
faixa = faixa / 2;
elseif(refaz == '*')
faixa = faixa / 4;
elseif(refaz == '/')
faixa = faixa / 16;
else
lim1 = refaz * freq;
end;
lim2 = lim1 + faixa - 1;
ve_sinal(lim1,lim2);
refaz= input('Continuar? (0=sim, <num> muda lim1, + dobra faixa, - divide por 2)
end;

% Escolhe canais para analise (itera se nao ficar bom) e mostra plano de fase wa
refaz = 1;
while( refaz == 1 )
canais = input('Quais os canais da analise ? (entre [ ]) ');
canais=sort(canais);

% A matriz Wab e' na verdade [ Wab(canal(1)) Wab(canal(2)) ... ]
Wab = waveplan(canais,lim1,lim2);
refaz= input('Ficou bom ? (0 = sim, 1 = nao)' );
end;

% Imprime as figuras
```



```
%%impr = input('Imprimir sinal ? ([] = nao) ');
%%if ( ~isempty(impr) )
%% figure(1);
%% eval(['print -dcdjmono sig' lim1 '.jet']);
%%end;

%%impr = input('Imprimir plano wavelet ? ([] = nao) ');
%%if ( ~isempty(impr) )
%% figure(2);
%% eval(['print -dcdeskjet wav' lim1 '.jet']);
%%end;

% Calcula propagacao, escolhendo os limites no plano de fase wavelet
refaz = 1;
while( refaz == 1 )
disp('Escolha de limites: (tecle enter para os anteriores) ');

zanzo = input(' Aminimo ? '); if ( ~isempty(zanzo)) Alow= zanzo; end;
zanzo = input(' Amaximo ? '); if ( ~isempty(zanzo)) Ahigh= zanzo; end;
zanzo = input(' Bminimo ? '); if ( ~isempty(zanzo)) Bleft= zanzo; end;
zanzo = input(' Bmaximo ? '); if ( ~isempty(zanzo)) Brght= zanzo; end;

Aspan = Alow:Ahigh;
Bspan = Bleft:Brght;

velocidade = wavevel2(Wab,canais,Aspan,Bspan) .* freq;
disp('Velocidade de propagacao :');
```

```
disp(velocidade);  
refaz= input('Ficou bom ? (0 = sim, 1 = nao)' );  
end;
```

A.9 waveplan.m

```
function Wabttotal=waveplan(canais,lim1,lim2);  
% waveplan gera plano de fase wavelet para varios canais.  
  
global Y T;  
b=max(size(canais));  
  
Wabttotal = [];  
figure(2);  
clf; ca=0;  
  
qmf=MakeONFilter('Daubechies',6);  
for a=canais;  
ca=ca+1;  
sinal=(Y(lim1:lim2,a)');  
subplot(b,1,ca);  
  
% Calculo do Plano de Fase Wavelet - Sinal Unidimensional  
L=2;  
W = FWT_PO(sinal,L,qmf);  
J = log2(max(size(W)));
```

```

%%%%% Calculo Novo - usa rotina modificada do wavelab
Wab = wv_matr(W,L,0,qmf,'Contour');
Wabttotal = [ Wabttotal Wab ];

%%%%% Calculo Anterior - utilizava uma matriz com os coeficientes das wavelets
%%%%% Wab = zeros(J-1,2^(J-1));
%%%%% for level = L:J-1;
%%%%% faixa = 2^(J-1-level);
%%%%% for t = 0 : ((2^level)-1)
%%%%% passo = t*faixa+1;
%%%%% Wab(level,passo:(passo+faixa-1)) = ...
%%%%% ones(1,faixa) * W(dyad2ix(level,t));
%%%%% end;
%%%%% end;
%%%%% % plot
%%%%% tmp=(lim2-lim1+1)/2;
%%%%% contour(abs(Wab));
%%%%% title(['Plano Wavelet do canal ',int2str(a)]);
%%%%% xlabel('Tempo/2 (s)'); ylabel('Nivel de detalhe');

end;

```

A.10 wavevel.m

```

function velocidade=wavevel(Wab,canais,Aspan,Bspan);
% wavevel procura e calcula correlacao e mostra valor da velocidade de propagacao

```

```
%%% Tamanho de cada matriz Wab
%%% A Wab passada e' na verdade [ Wab_canal(1) | Wab_canal(2) | ... ]
szcWab = size(Wab);
numchannel = max(size(canais));
tWab = szcWab(2)/numchannel;

%%% Calcula velocidade de propagacao
jc=0;
for a1 = canais;
    for a2=fliplr(canais);
if (a1 == a2)
    %% Decide onde esta' o maximo para o proximo canal-base
    %% E' a ultima operacao para o canal
    Bspan=Bspan+tempo(jc);
end;

if (a1 < a2);
    %% pega matrizes a serem comparadas
    indaux = ( (find( canais == a1) - 1) * tWab ) +1;
    inddep = ( (find( canais == a2) - 1) * tWab ) +1;
    Maux = Wab(:, indaux:(tWab+indaux-1) );
    Mdep = Wab(:, inddep:(tWab+inddep-1) );

    %% acha maximo da correlacao entre as matrizes
    Mant=zeros(size(Maux));
    Mant(Aspan,Bspan) = Maux(Aspan,Bspan);
```

```
    Comp = xcorr2(Mdep,Mant);
    jc=jc+1;
    sComp = size(Comp);
    distancia(jc) = a2-a1;
    aux1 = max(Comp);
    [aux2 tm] = max(aux1);
    tempo(jc) = tm - ( (sComp(2)+1)/2 );
end;
    end;
end;

aux = polyfit(distancia,tempo,1);
velocidade = aux(1);
```

A.11 `wv_matr.m`

```
function display_array = wv_matr(wc,L,scal,qmf,type)

    wavecoef = ShapeAsRow(wc);
    if scal==0,
    scal = 1. / max(abs(wavecoef));
    end

    [n,J] = dyadlength(wavecoef);
    num_levels = J - L + 1;
    display_array = zeros(num_levels,n);
```

```
count = 1;
w = zeros(1,n);
w(1:2^(L)) = wavecoef(1:2^(L));
display_array( count, : ) = scal .* IWT_PO(w,L,qmf);

for j=(J-1):-1:L
    count = count + 1;
    w = zeros(1,n);
    w(dyad(j)) = wavecoef(dyad(j));
    display_array( count, : ) = scal .* IWT_PO(w,j,qmf);
end

display_array = flipud(display_array);
```