

PROCESSAMENTO EM TEMPO-REAL
DE SINAIS DOPPLER DE FLUXO SANGUÍNEO

TESE SUBMETIDA À UNIVERSIDADE DO ALGARVE
PARA O GRAU DE DOUTOR
NA ÁREA DAS CIÊNCIAS DA COMPUTAÇÃO

Maio 2004

Por

Maria Margarida da Cruz Silva Andrade Madeira e Carvalho de Moura

Área Departamental de Engenharia Electrónica e Computação
Faculdade de Ciências e Tecnologia

Conteúdo

Resumo	15
Abstract	17
Agradecimentos	19
Pensamento	21
1 Introdução	23
1.1 Perspectiva Histórica	25
1.2 Organização da Tese	27
1.3 Contribuições	29
2 Enquadramento	33
2.1 Introdução	33
2.2 Sistemas Clínicos de Ultrassom Doppler	33
2.2.1 Instrumentação Doppler	34
2.2.2 Fluxo Sanguíneo	35
2.2.3 Análise de Fluxo Sanguíneo	36
2.2.4 Indicadores Clínicos de Fluxo Sanguíneo	38
2.3 Estimação Espectral	41
2.3.1 Short Time Fourier Transform	42
2.3.2 AR - Modified Covariance	43
2.3.3 Distribuição Tempo-Frequência de Choi-Williams	45
2.3.4 Distribuição Tempo-Frequência de Bessel	50
2.3.5 Sinal <i>Analítico</i>	51
2.3.6 Parâmetros Espectrais	56
2.4 Sistemas Paralelos	61

2.4.1	Processadores	61
2.4.2	Modelos de Comunicação	65
2.4.3	Abordagens à Programação Paralela	66
2.5	Avaliação de Desempenho	66
2.5.1	Desempenho Estatístico	67
2.5.2	Desempenho Computacional	67
2.5.3	Critério Custo/Benefício	73
2.6	Aquisição e Visualização de Dados	74
2.7	Conclusão	75
3	Planeamento do Sistema de Processamento	77
3.1	Introdução	77
3.2	Descrição do Contexto da Aplicação	78
3.2.1	Caracterização de Fluxo Sanguíneo	78
3.2.2	Validação e Ensaio Clínico	79
3.2.3	Apoio à Investigação e Desenvolvimento	80
3.3	Plataforma de Hardware	82
3.4	Modelo Conceptual	82
3.4.1	Aquisição de Dados	83
3.4.2	Estimação Espectral	84
3.4.3	Visualização	84
3.4.4	Avaliação de Qualidade	85
3.5	Requisitos Funcionais e Não Funcionais	86
3.6	Arquitecturas de Implementação	88
3.7	Considerações sobre as Linguagens de Implementação	89
3.8	Utilização da Língua Portuguesa	91
3.9	Conclusão	91
4	Subsistemas de Aquisição e Visualização de Dados	93
4.1	Introdução	93
4.2	Fases Intermédias do Sistema de Aquisição de Dados	94
4.2.1	Dados Simulados	94
4.3	Diversas Origens dos Dados para Processamento Diferido	96
4.3.1	Hospital Distrital de Faro	96
4.3.2	Universidade de Florença - Itália	97
4.3.3	Universidade de Leicester - Reino Unido	97

4.3.4	Universidade Nacional Autónoma do México	97
4.3.5	Geradores de Sinais	98
4.4	Formato dos Dados	98
4.4.1	Dados Simulados	99
4.4.2	Dados Reais	99
4.4.3	Dados Processados	99
4.5	Conversores	100
4.6	Aquisição de Dados Reais em Tempo-Real	105
4.7	Visualização	106
4.7.1	Apresentação Simples	106
4.7.2	Espectro Médio de Sinais Simulados	108
4.7.3	Espectro Médio de Sinais Clínicos	108
4.8	Conclusão	109
5	Subsistema A de Estimação Espectral	111
5.1	Introdução	111
5.2	Uso de Plataformas Aditivas	111
5.2.1	Arquitecturas de Implementação	112
5.2.2	Algoritmos	113
5.2.3	Implementações na Plataforma Aditiva A	114
5.2.4	Implementações na Plataforma Aditiva B	136
5.2.5	Implementação na Plataforma Aditiva C	142
5.2.6	Outra Implementação	143
5.3	Conclusão	144
6	Subsistema B de Estimação Espectral	145
6.1	Introdução	145
6.2	Formas Alternativas dos Estimadores	146
6.2.1	Impacto da Representação Numérica	147
6.3	Implementação Sequencial	149
6.3.1	STFT	149
6.3.2	Sinal <i>Analítico</i>	158
6.3.3	AR - Modified Covariance	163
6.3.4	Algoritmos da DTF de Choi-Williams	181
6.3.5	Algoritmo da DTF de Bessel	200
6.4	Conclusão	210

7	Subsistema de Avaliação de Qualidade	213
7.1	Introdução	213
7.2	Métodos de Avaliação de Desempenho	214
7.2.1	Avaliação Custo/Benefício	216
7.3	Conclusão	226
8	Comentários Finais	229
8.1	Introdução	229
8.2	Estimadores Espectrais	229
8.2.1	Indicadores Clínicos e Parâmetros Espectrais	230
8.2.2	Comparação do Desempenho dos Métodos de Estimação Espectral	232
8.2.3	Informação Clínica	233
8.2.4	Sumário	234
8.3	Estratégias no Desenvolvimento do Sistema	235
8.3.1	Desenho do Sistema	235
8.3.2	Implementações	236
8.3.3	Obstáculos e Consequentes Decisões	240
8.3.4	Outras Opções	241
8.3.5	Sumário	243
8.4	Considerações finais	244
8.5	Trabalho Futuro	249
	Bibliografia	251
A	Formatos de Ficheiros Estrangeiros	261
A.1	Sinais Simulados	261
A.2	TIM	261
A.3	WAV	262
A.4	SON	263
A.5	ALG	264
A.6	DOP	265
A.7	EMB	265
A.8	MIX	266
B	Parametrização dos Estimadores Espectrais	269

Lista de Tabelas

4.1	Ficheiro de descrição dos parâmetros dos sinais.	95
4.2	Formato do cabeçalho dos ficheiros de sinais simulados.	100
4.3	Formato do cabeçalho dos ficheiros de sinais reais.	101
4.4	Formato do cabeçalho dos ficheiros de espectrogramas.	102
5.1	Tempos de cálculo do estimador AR-MC no processador i860. . .	116
5.2	Efeito da optimização efectuada pelo compilador no algoritmo do estimador AR-MC executado pelo processador i860.	117
5.3	Tempos (em ms.) da implementação heterogénea do estimador AR-MC na estimação de quatro segmentos de 10 ms..	122
5.4	Avaliação em Matlab do algoritmo e núcleo da distribuição CW. .	124
5.5	Tempo total do estimador CWD num único T8.	125
5.6	Implementação do estimador CWD num único C40.	126
5.7	Tempos de execução (em ms.) da implementação do algoritmo da CWD num único processador C40. O tempo para todo o sinal é uma estimativa.	127
5.8	Variação de tempos de cálculo exibida pelo processador C40. . .	128
5.9	Variação de tempos de cálculo exibida pelo processador i860. . .	128
5.10	Tempos de transferência (em ms.) de dados na arquitectura heterogénea.	129
5.11	Tempos de execução do estimador AR-MC (quarta ordem) no processador C40.	136
5.12	Tempos de transferência de dados medidos na plataforma aditiva B.	141
5.13	Tempos calculados de transferência de dados na plataforma aditiva B para os estimadores AR-MC e CWD.	141
5.14	Tempos de execução (em ms.) do estimador AR-MC na plataforma aditiva C.	142

6.1	Carga (em número de segmentos) variando a taxa de sobreposição de segmentos.	147
6.2	Carga (em número de pontos) variando a taxa de sobreposição de segmentos.	148
6.3	Impacto no tempo de execução (em segundos) da representação de dados adoptada no desempenho da AR-MC de quarta ordem. . .	148
6.4	Impacto no tempo de execução (em segundos) da representação de dados adoptada no desempenho da CWD ($M = 30, \sigma = 5$).	149
6.5	Tempos de execução (em segundos) dos algoritmos sequenciais da STFT.	151
6.6	Tempos de execução (em segundos) do cálculo e armazenamento em memória partilhada da matriz de transformação do sinal <i>analítico</i>	160
6.7	Tempos de execução (em segundos) para o cálculo do sinal <i>analítico</i> por convolução com um filtro.	161
6.8	Síntese dos tempos de execução (em segundos) das abordagens ao cálculo do sinal <i>analítico</i>	161
6.9	Tempos de execução dos algoritmos sequenciais do estimador AR-MC.	178
6.10	Tempos de cálculo, em milisegundos, de alguns núcleos da CWD.	191
6.11	Tempos de cálculo, em milisegundos, dos produtos cruzados da implementação da CWD.	194
6.12	Tempos de execução, em segundos, do estimador CWD de núcleos modificados.	196
6.13	Distinção entre os computadores utilizados.	201
7.1	Benefício na estimação da largura de banda de sinais de válvula aórtica com SNR de 10 dB.	218
7.2	Desempenho do estimador STFT na estimação dos parâmetros f_m e bw , em termos de rmse (Hz).	222
7.3	Desempenho do estimador AR-MC na estimação dos parâmetros f_m e bw , em termos de rmse (Hz).	223
7.4	Desempenho do estimador CWD na estimação dos parâmetros f_m e bw , em termos de rmse (Hz).	223
7.5	Custo (em segundos) das implementações do estimador AR-MC.	224

Lista de Figuras

2.1	Espectro de potências Doppler obtido pela estimação de um sinal simulado de fluxo sanguíneo arterial.	37
2.2	Alguns dos indicadores observáveis num espectrograma.	38
2.3	Alguns dos elementos quantitativos retirados das curvas características de fluxo sanguíneo da artéria carótida primitiva normal.	39
2.4	Ilustração de uma curva de fluxo sanguíneo característica da artéria carótida primitiva.	40
2.5	Segmentação por ciclos cardíacos de uma curva de frequência média usando o algoritmo FPS.	59
2.6	Determinação da frequência máxima pelo método geométrico modificado.	61
2.7	Imagem recolhida durante um exame à artéria carótida.	75
3.1	Sistema de caracterização de fluxo sanguíneo	79
3.2	Representação do sistema para ensaios clínicos.	80
3.3	Representação do sistema para estudo da implementação de métodos de processamento de sinais biomédicos.	81
3.4	Principais blocos do sistema computacional	83
3.5	Custo/Benefício em relação à FFT.	85
3.6	Representação gráfica dos principais blocos do sistema analisador de fluxo sanguíneo	86
4.1	Conversor de dados simulados	103
4.2	Conversor de dados reais	104
4.3	Conversor de dados processados	105
4.4	Aquisição de dados para estimação espectral	106
4.5	Definição da opção de visualização da estimação.	107
4.6	Espectrograma simples	107

5.1	Plataforma aditiva A.	112
5.2	Plataforma aditiva B.	113
5.3	Plataforma aditiva C.	113
5.4	Processo de criação de uma aplicação para processador i860. . . .	116
5.5	Tempos de execução, em ms., do algoritmo do estimador AR-MC no processador i860. Linha horizontal a 10 ms. correspondente ao limiar de tempo-real.	117
5.6	Regularidade dos tempos de execução do algoritmo do estimador AR-MC no processador i860.	118
5.7	Gradiente de execução do algoritmo do estimador AR-MC no processador i860.	119
5.8	Percentagem de redução dos tempos de execução conseguida utilizando as opções de optimização do compilador para o estimador AR-MC no processador i860.	120
5.9	Aceleração do algoritmo do estimador AR-MC no processador i860.	121
5.10	Tempos médios por segmento de dados do estimador AR-MC na plataforma heterogénea.	123
5.11	Efeito das opções de optimização do compilador <i>pgcc</i> nos tempos de comunicação entre o T8-i860.	130
5.12	Tempos de comunicação dos segmentos de dados utilizados pelo estimador AR-MC entre a máquina hospedeira Sun e o processador raiz.	131
5.13	Gradiente de comunicação de dados utilizados pelo estimador AR-MC entre a máquina hospedeira Sun e o processador raiz.	132
5.14	Tempos de comunicação dos segmentos de dados utilizados pelo estimador CWD entre a máquina hospedeira Sun e o processador raiz.	133
5.15	Tempos de comunicação, por segmento, dos dados utilizados pelo estimador CWD entre a máquina hospedeira Sun e o processador raiz.	134
5.16	Tempos de comunicação, por elemento, dos dados utilizados pelo estimador CWD entre a máquina hospedeira Sun e o processador raiz.	135
5.17	Implementação melhorada do estimador AR-MC na Plataforma Aditiva B.	137

5.18	Regularidade da implementação do estimador AR-MC na Plataforma Aditiva B.	138
5.19	Detalhe da implementação melhorada do estimador AR-MC na Plataforma Aditiva B.	139
5.20	Regularidade da implementação do bloco RMC do estimador AR-MC na Plataforma Aditiva B.	140
5.21	Regularidade da implementação do estimador AR-MC na Plataforma Aditiva C.	143
6.1	Tempos de execução dos algoritmos sequenciais da FFT na máquina <i>arquimedes</i>	151
6.2	Tempos de execução do algoritmo de Kay na máquina <i>arquimedes</i>	152
6.3	Tempos de execução médios por elemento de dados do algoritmo de Kay na máquina <i>arquimedes</i>	153
6.4	Tempos de execução médios por segmento de dados do algoritmo de Kay na máquina <i>arquimedes</i>	154
6.5	Tempos de execução do algoritmo <i>fftw</i> na máquina <i>arquimedes</i>	155
6.6	Tempos de execução médios por elemento de dados do algoritmo <i>fftw</i> na máquina <i>arquimedes</i>	156
6.7	Tempos de execução médios por segmento de dados do algoritmo <i>fftw</i> na máquina <i>arquimedes</i>	157
6.8	Definição das opções de execução do estimador STFT.	162
6.9	Coeficientes de distância zero.	165
6.10	Coeficientes de distância um.	167
6.11	Coeficientes de distância dois.	168
6.12	Coeficientes de distância três.	168
6.13	Coeficiente de distância quatro.	169
6.14	Coeficientes de distância zero.	170
6.15	Coeficientes de distância um.	172
6.16	Comparação entre os algoritmos de cálculo da matriz de Covariância Modificada aplicados a dados reais.	175
6.17	Razão entre os algoritmos simplificados de cálculo da matriz de Covariância Modificada aplicados a dados reais.	176
6.18	Operações de vírgula flutuante dos algoritmos de cálculo da matriz de Covariância Modificada aplicados a dados complexos.	177

6.19	Melhoria nos tempos de execução obtida com o algoritmo simplificado face aos obtidos com o algoritmo proposto por Kay aplicado a dados reais e complexos.	179
6.20	Definição das opções de execução do estimador AR-MC.	180
6.21	Vista de topo de um núcleo standard da CWD.	182
6.22	Perspectiva de um núcleo standard da CWD.	183
6.23	Vista de topo de um núcleo RI da CWD.	185
6.24	Perspectiva de um núcleo RI da CWD.	186
6.25	Vista de topo de um núcleo AF da CWD.	187
6.26	Perspectiva de um núcleo AF da CWD.	188
6.27	Vista de topo de um núcleo RIAF da CWD.	189
6.28	Perspectiva de um núcleo RIAF da CWD.	190
6.29	Função de autocorrelação no cálculo dos produtos cruzados da distribuição CWD	193
6.30	Diagrama da implementação do estimador CWD	195
6.31	Tempos de execução da implementação do estimador CWD para segmentos de 64, 128, 256 e 512 pontos com 0, 25, 50 e 75% de sobreposição.	197
6.32	Definição das opções de execução do estimador CWD.	199
6.33	Tempos de execução da distribuição Bessel, nos computadores <i>arquimedes</i> , <i>coral</i> e <i>ticoteco</i> , para janelas não sobrepostas de 256 pontos, variando o tipo de janela utilizada.	202
6.34	Tempos de execução da distribuição Bessel, nos computadores <i>arquimedes</i> , <i>coral</i> e <i>ticoteco</i> , para janelas não sobrepostas de 64, 128, 256 e 512 pontos.	203
6.35	Tempos de execução da distribuição Bessel, nos computadores <i>arquimedes</i> , <i>coral</i> e <i>ticoteco</i> , para segmentos de 512 pontos com 0, 50, 75 e 87,5% de sobreposição. Linha horizontal representa o limite de tempo-real (33,61 s).	204
6.36	Tempos de execução da distribuição Bessel, nos computadores <i>arquimedes</i> , <i>coral</i> e <i>ticoteco</i> , considerando o cálculo do sinal <i>analítico</i> de segmentos não sobrepostos de 64, 128, 256 e 512 pontos. Linha horizontal representa o limite de tempo-real (33,61 s).	205

6.37	Tempos de execução da distribuição Bessel, no computador <i>ar-quimedes</i> , variando a dimensão do segmento e do parâmetro α do núcleo. Linha horizontal representa o limite de tempo-real (33,61 s).	206
6.38	Tempos de execução da distribuição Bessel, nos computadores <i>coral</i> e <i>ticoteco</i> , variando a dimensão do segmento e do parâmetro α do núcleo. Linha horizontal representa o limite de tempo-real (33,61 s).	207
6.39	Definição das opções de execução do estimador BD.	209
7.1	Custo das implementações dos métodos AR-MC e CWD em distintas plataformas.	218
7.2	Benefício normalizado da CWD com núcleos STD, RI, AF e RIAF na estimação de sinais de válvula aórtica.	220
7.3	Custo/benefício, custo e benefício normalizados da CWD com núcleo STD na estimação de sinais de válvula aórtica.	221
7.4	Custo/Benefício das implementações do estimador AR-MC.	225

Resumo

O objectivo fundamental do trabalho aqui descrito é o desenvolvimento de um sistema computacional em tempo-real, baseado num computador pessoal, que forneça informação quantitativa precisa, por forma a complementar o diagnóstico clínico de anomalias cardiovasculares.

Para este fim, o sistema modular de processamento de sinais de ultra-som de fluxo sanguíneo concebido considera quatro subsistemas nucleares: aquisição de dados, estimação espectral, visualização de espectros e avaliação de desempenho. A modularidade do sistema permite a sua utilização com diversos tipos de sinais de ultra-som, não se restringindo aos de fluxo sanguíneo, podendo servir objectivos de aplicação industrial ou médica.

O subsistema de aquisição de dados considera módulos para a utilização de dados provenientes de uma sonda de ultrassom Doppler e a importação de dados de programas de simulação e de outras origens. A cada origem dos dados a importar corresponde um módulo de conversão para o formato único adoptado pelos restantes módulos do sistema.

A especificação do subsistema de visualização de espectros prevê o desenvolvimento de módulos cuja funcionalidade é definida em função da utilização dada ao sistema. Assim, foi desenvolvido um módulo que permite a visualização dos espectros instantâneos em tempo-real.

No desenvolvimento do subsistema de estimação espectral considerou-se a implementação de quatro métodos: a transformada de Fourier de tempo curto (STFT), o método auto regressivo de Covariância Modificada (AR-MC), a distribuição tempo-frequência de Choi-Williams (CWD) e a distribuição tempo-frequência de Bessel (BD). Foram feitos estudos comparativos da implementação dos métodos AR-MC e CWD em plataformas aditivas heterogéneas visando a determinação da melhor solução para a estimação de parâmetros de sinais de fluxo sanguíneo obtidos por ultra-som. Os elementos de processamento das plataformas

aditivas consideradas foram INMOS T8, Texas Instruments TMS320C40, Intel i860, ADSP20162 (SHARC) e uma FPGA. Usando processadores de propósito geral, foram estudadas as implementações dos quatro estimadores. As implementações produzidas não foram dirigidas para o processamento de um único tipo de fluxo sanguíneo, podendo ser utilizadas para outros tipos de fluxo e mesmo para outros fins. Na maioria dos casos, o tempo de execução das implementações é inferior ao limite de tempo-real, possibilitando ainda a incorporação de outros métodos/técnicas complementares.

O subsistema de avaliação de desempenho considera as vertentes de qualidade estatística e computacional, articuladas pela variante ao critério Custo/Benefício desenvolvida para permitir avaliar o desempenho do sistema independentemente do método de estimação ou da base computacional utilizada.

Foi desenvolvido um interface gráfico com o utilizador que integra módulos dos subsistemas de aquisição de dados, estimação espectral e visualização, conseguindo-se o protótipo de um sistema amigável que pode ser utilizado na caracterização de fluxo sanguíneo e no apoio à investigação em processamento de sinal.

Abstract

The ultimate goal of the work described here is the development of a real time computational system, based on a personal computer, that produces accurate quantitative information to complement the clinical diagnose of cardiovascular diseases.

Envisaging this, a modular system was conceived, contemplating four main subsystems: data acquisition, spectral estimation, spectra visualisation and performance evaluation. A strong adherence to systems' modularity during design and development process guarantees that, by not being restricted to blood flow signals, the system can fulfil industrial or medical application objectives.

The data acquisition subsystem encompasses several modules to enable several data sources, such as an ultrasound probe or the import of data from simulation programs or other real signals from off-line sources. For each data source corresponds a conversion module that translates the foreign data format to a common data format to be used by the remaining system modules.

Spectra visualisation subsystem specification is defined by the system targeted application. For each particular functionality desired, a separate module can be built. In the present work, a module that allows the real-time visualisation of instantaneous spectra was included.

The spectral estimation subsystem development comprised the implementation study of four methods: Short Time Fourier Transform (STFT), the Auto Regressive Modified Covariance method (AR-MC), the Choi-Williams time-frequency distribution (CWD) and the Bessel time-frequency distribution.

Comparative studies of AR-MC and CWD implementations were conducted using additive platforms, aiming the identification of the best solution for the estimation of some spectral parameters of ultrasound blood flow signals. The processing elements used in the platforms included INMOS T8, Texas Instruments TMS320C40, Intel i860; two other processing solutions were also evaluated,

namely ADSP20162 (SHARC) and a FPGA with a TMS320C40.

General purpose processors were used in the implementation studies of the four methods. These implementations were not targeted towards the estimation of an unique kind of blood flow, being suitable to other kinds of flow or even other purposes. In most cases, execution times achieved with these implementations is below real-time requirements, still allowing the addition of other, complementary, methods/techniques.

The performance evaluation subsystem takes into account both the statistical and computational quality, geared by the developed variant of Cost/Benefit criterion, allowing the evaluation of system performance, regardless of the estimation method or computational base used.

A graphical user interface was developed, integrating the data acquisition, spectral estimation and visualisation subsystems, into a friendly-system prototype that can be used for blood flow characterisation or for signal processing research support.

Agradecimentos

Gostaria de deixar expresso o meu reconhecimento

à Universidade do Algarve, na pessoa do Magnífico Reitor Professor Doutor Adriano Lopes Gomes Pimpão, pela autorização oportuna de desenvolver estes estudos e pela dispensa de serviço docente que tornou possível a sua prossecução e pelos meios disponibilizados para a sua realização;

ao Programa de Desenvolvimento Educativo para Portugal – PRODEP III, Medida 5, Acção 5.3;

ao CRUP/Tratado de Windsor pelo apoio financeiro às deslocações, no âmbito das Acções Integradas Luso-Britânicas, à The Sheffield University, Reino Unido;

ao Department of Automatic Control & Systems Engineering da The Sheffield University, Reino Unido, onde fui tão bem recebida nas visitas de trabalho que efectuei, pelo Professor Peter Fleming, Dr. Osman Tohki, Daniela Ramos-Hernandez e à família Fonseca, Carlos e Viviane, que me acolheram;

ao CYTED – Programa de Cooperación Científica y Tecnológica – por ter apoiado a criação da RITUL – Rede Ibero-Americana de Tecnologias de Ultrassom – fomentando a colaboração com vários grupos de investigação com interesses comuns nesta área;

à UNESCO, pelo apoio financeiro ao projecto MAGIAS – Metodos Avanzados de Generación de Imágenes Acústicas;

ao Serviço de Cardiologia do Hospital Distrital de Faro, pelo interesse, disponibilidade e colaboração.

Gostaria de agradecer

à minha orientadora, Prof. Doutora Maria da Graça Ruano, pelo conhecimento oferecido, pela amizade, confiança, apoio e incentivo, guiando-me sempre para um pouco mais longe;

à Eng^a Ana Isabel Leiria pela amizade, disponibilidade e conversa criativa e

inspiradora;

ao Dr. Gago Leiria, ao Dr. Sangiva Cacodcar, pela atenção, interesse, conselhos e opiniões;

ao Professor Evans, Dr. Francesco Guidi, pelos sinais reais e ao Dr. Fábian Garcia Nocetti e Dr. Eduardo Moreno, pela sonda ultra-sónica; ao Dr. Wagner Coelho pelos esclarecimentos e bons conselhos;

aos amigos e colegas, pela boa disposição e bom conselho, Luis, Viviane, Carlos, Pedro, Helder e Filomena;

ao Luis, aos meus pais e irmãos, às minhas tias e a toda a família, pela aceitação da minha ausência e pelo apoio emocional incondicional

A todos com quem lidei, que tornaram grata a lembrança destes tempos, o meu Muito Obrigada.

Pensamento

“ (...) que fazemos nós nesta vida que nos é dada, neste destino que temos de cumprir?”

Podemos, até, não saber responder sobre o que fazemos e que destino é o nosso porque, na realidade, apenas conseguimos efectuar a leitura do nosso destino quando já passou algum tempo sobre a nossa vida e olhamos para trás. Estou convencida de que apenas no momento da morte é que se faz o balanço da vida, pois até lá tudo se encontra, ainda, em evolução.

O que nos é possível é, por um lado, o esforço da compreensão e, por outro, o esforço feito para que algo tenha melhorado à nossa volta com a nossa passagem.

Acho que vale a pena tentar proceder assim (...).”

Yvette Centeno *in* Diálogos Filosóficos e Alquímicos (Mendanha, 1996)

Capítulo 1

Introdução

O processamento de sinais biomédicos é uma das áreas em que se tem registado um progressivo aumento de utilização da tecnologia de informação, sendo muitos dos componentes físicos substituídos por programas computacionais. O aumento da capacidade dos computadores e a diversidade e vulgarização de utilização de sistemas embebidos tem facilitado a utilização de programas em substituição de sistemas de processamento de sinal analógicos ou digitais, anteriormente implementados em circuitos dedicados.

Centrando a atenção nos sistemas de diagnóstico, os mais comuns à data deste texto são de Ressonância Magnética (MRI), de Tomografia Assistida por Computador (TAC) e ecográficos. Os sistemas ecográficos, de menores dimensões e menor custo, têm sido adoptados como forma preferencial de diagnóstico em diversas especialidades, podendo ser destacadas a Cardiologia, a Neurologia, a Nefrologia e a Obstetrícia.

A determinação do diagnóstico, após o exame, acarreta vários procedimentos de análise da informação recolhida, procedimentos morosos e, considerando que são efectuados por pessoal clínico especializado sob supervisão médica, dispendiosos. A implementação computacional de alguns desses procedimentos e a utilização de mecanismos complementares que permitissem aumentar a precisão de estimação de alguns parâmetros clínicos essenciais ao diagnóstico seria benéfica, podendo vir a traduzir-se numa possibilidade efectiva de reduzir o tempo e custo de diagnóstico e contribuir para facilitar o rastreio das camadas populacionais mais sujeitas a determinados factores de risco e a implementação de medidas correctivas que evitassem a progressão de possíveis patologias.

Para tal, e considerando a perspectiva de evolução tecnológica no âmbito

dos computadores pessoais, sistemas operativos mais utilizados e processadores digitais de sinal, pretendeu-se definir um sistema de processamento de sinais biomédicos que possa complementar, ou em certos tipos de diagnóstico, competir com, de forma genérica, alguns dos sistemas clínicos de diagnóstico.

O volume de dados e conseqüente volume de processamento em causa, ainda que considerando a capacidade dos actuais sistemas, são, na sua grande maioria, bastante significativos. Considerando individualmente cada uma das tarefas implementadas computacionalmente necessárias à análise da informação clínica em que se baseia um diagnóstico, poucas são as muito complexas ou demasiado demoradas. Considerando o total de tarefas, e o escasso tempo de um técnico clínico face às solicitações a que é sujeito, seria ideal que a totalidade das tarefas implementadas computacionalmente pudesse ser realizada instantaneamente. Concorrendo para a satisfação desse ideal, é necessário recorrer a técnicas de implementação de algoritmos rápidos.

Este trabalho, originalmente subordinado ao tema *Processamento em tempo-real de sinais biomédicos em arquitecturas de processamento paralelo*, tem como objectivo a criação de um sistema de informação complementar para diagnóstico clínico por ultra-som, que forneça, em tempo-real, atempadamente os indicadores clínicos necessários à formulação de um diagnóstico.

Considerando que as doenças cardiovasculares são ainda, em Portugal, a maior causa de morte, e uma das maiores causas de invalidez em que a pós-vida dos indivíduos afectados e seus familiares fica muito condicionada, foi dado ênfase aos actuais equipamentos de ecografia cardiovascular. O modo Doppler pulsado, utilizado quando se pretende uma avaliação detalhada de uma pequena região, é muito relevante para a quantificação de uma patologia e obtenção de dados para elaboração da história clínica do paciente, com vista à aplicação de terapia preventiva.

Assim, e evitando comprometer a generalidade do sistema, pretende-se um sistema modular, versátil e de baixo custo, que permita manipular os dados tais como os obtidos de um equipamento de ultra-som cardiovascular funcionando em modo Doppler pulsado, produzindo os indicadores mais relevantes ao diagnóstico clínico.

Os trabalhos efectuados na prossecução destes objectivos são aqui descritos, subordinados ao tema *Processamento em tempo-real de sinais Doppler de fluxo sanguíneo*, título do trabalho desenvolvido.

1.1 Perspectiva Histórica

À data de início dos trabalhos aqui relatados, em 1998, citando o plano de trabalhos, pretendia-se “desenvolver um sistema computacional, baseado num computador pessoal, o qual, incluindo arquitecturas de processamento paralelo, homogéneas ou heterogéneas (utilizando nomeadamente *transputers* e processadores digitais de sinal (DSP’s)), permita executar em tempo-real uma estimação quantitativa precisa dos factores clínicos relevantes à detecção de estenoses.”

O desenvolvimento tecnológico e cooperações com o Hospital Distrital de Faro (HDF), nomeadamente com o seu Serviço de Cardiologia, e com outros grupos de investigação, originaram solicitações que motivaram o alargamento dos objectivos e metodologias iniciais. Acresce ainda que, no grupo de trabalho, para além do processamento de elevado desempenho, é investigado o processamento (análise e síntese) de sinais de fluxo sanguíneo. As suas necessidades das investigações ocorridas e respectivos resultados contribuíram para a frequente reavaliação dos objectivos do presente trabalho.

Sendo natural que tal alargamento de objectivos aconteça num trabalho, a evolução tecnológica ocorrida durante a execução deste projecto foi de tal forma acentuada que se introduz esta perspectiva histórica para permitir situar o leitor no contexto em que o trabalho foi sendo desenvolvido. Nesse sentido, a par de citações do plano de trabalhos inicial, serão tecidos alguns comentários sobre as alterações efectuadas quanto aos objectivos e metodologia do trabalho.

Pretendia “ (...) obter-se um sistema que permita estimar espectros de fluxo sanguíneo alternativamente aos sistemas Doppler ultra-sónicos actualmente existentes; este sistema será composto por uma unidade de estimação espectral, outra de aquisição de dados e outra de visualização de espectros. Qualquer uma das unidades constituintes do sistema deve individualmente, e, como um todo, cumprir níveis de desempenho computacional por forma a garantir o diagnóstico preciso de casos estenóticos em tempo-real.” Estudos anteriores do grupo de investigação visavam a determinação de estenoses em artéria carótida primitiva. Ao Serviço de Cardiologia do HDF interessavam outros objectos de estudo, válvulas. Ao diagnóstico de estenoses em artérias ou vasos, alia-se a estenose valvular. No âmbito de uma cooperação com o *Medical Physics Department* da Universidade de Leicester, Reino Unido, a partir de 2000, os sinais obtidos por Doppler transcraniano começam a ser analisados pelo grupo de trabalho no âmbito de metodologias eficientes de processamento de sinal, visando a caracterização do fluxo sanguíneo

em artéria cerebral média, a detecção de embolias e a sua classificação.

“Para tal, far-se-á uma selecção de alguns algoritmos identificados como ‘bons’ estimadores de fluxo sanguíneo, e desenvolver-se-ão os mesmos em formas capazes de processamento paralelo. Serão implementados em arquitecturas de processamento rápido, tanto homogéneas (integrando somente *transputers* ou somente DSP’s) como heterogéneas, sendo feito um estudo comparativo do desempenho dessas mesmas arquitecturas.” Tendo o grupo de trabalho identificado dois métodos para a estimação de sinais de fluxo sanguíneo da artéria carótida primitiva, cujos parâmetros haviam sido estudados e definidos por forma a apresentarem melhor qualidade estatística da estimação espectral, a implementação desses métodos é testada em arquitecturas paralelas aditivas. Para realizar esses estudos sobre a implementação nessas arquitecturas recorreu-se à colaboração com outras instituições, uma vez que, na Universidade do Algarve, existiam alguns tipos de DSP’s mas não placas que permitissem a sua ligação em rede. Assim, ou por deslocações à Universidade de Sheffield, Reino Unido, em alguns casos, ou em outro, enviando o código para teste no *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas* (IIMAS) da Universidade Nacional Autónoma do México, foram obtidos resultados com outras arquitecturas.

Alguns elementos do grupo de trabalho que se dedicam à investigação na área de processamento de sinal foram desenvolvendo e melhoraram progressivamente, outros estimadores e geradores de sinais de fluxo sanguíneo em outros vasos. Eram então contemplados sinais não estacionários e com ruído, a artéria cerebral média, bem como os sinais de fluxo em válvula aórtica e mitral, com vista à caracterização do fluxo sanguíneo, tanto em condições normais como anómalas. Os elevados tempos de processamento associados à busca dos parâmetros óptimos dos métodos para cada caso-estudo motivam a utilização das implementações dos algoritmos, entretanto também realizadas em máquinas de propósito geral, bastante mais rápidas que os processadores dedicados de que se dispunha. É neste enquadramento, e mantendo sempre presente o objectivo global do grupo de trabalho, isto é, a concretização de sistemas mais precisos mas económicos de apoio ao diagnóstico clínico de fluxo sanguíneo, que surgem as “formas alternativas dos estimadores”, designação empregue na presente tese. Já numa fase final do trabalho é incluído um outro estimador, referido por alguns investigadores como adequado à estimação de sinais de fluxo sanguíneo em artéria carótida primitiva, mas sistematicamente preterido devido ao peso computacional que implicava,

mesmo no caso de cálculo *offline*.

“Seguidamente, proceder-se-á ao desenvolvimento dum módulo de tratamento de dados. Este módulo incluirá duas componentes: um sistema de aquisição de dados do ecógrafo para posterior processamento no computador pessoal, e, outro sistema resultante da adaptação/desenvolvimento de software, para geração dos espectrogramas resultantes da aplicação dos métodos de estimação espectral ao sinal de fluxo sanguíneo. Em ambas as situações, o factor tempo de execução deve ser minimizado, utilizando para tal dispositivos de processamento rápido e/ou arquitecturas de processamento paralelo.” Por impossibilidade financeira de aquisição de um produto comercial que permitisse a transferência de dados de um ecógrafo do tipo existente no HDF, foi criado um módulo computacional de conversão de dados. A utilização de dados clínicos, necessária ao teste e validação do sistema global, foi sendo suprida por instituições de investigação colaborantes em projectos internacionais, instituições essas dispendo de modos de armazenamento de dados clínicos distintos, o que conduziu à implementação de módulos conversores de dados alternativos, a integrar no módulo de aquisição de dados do sistema global. De referir também, que na fase final do trabalho (2002), na sequência do plano de trabalhos de um projecto de investigação internacional MAGIAS (Métodos Avanzados de Generación de Imágenes Acusticas, projecto UNESCO: 2201.09, 3304.17, 3307.22, 3311.10)) em que o grupo está envolvido, foi obtido um protótipo laboratorial de um transdutor ultra-sónico de Doppler contínuo, através do qual se pode proceder à aquisição de sinais de fluxo sanguíneo de um indivíduo directamente para um computador através da sua placa de som. Uma vez mais, o sistema global sofreu actualizações por forma a contemplar este tipo de processamento.

1.2 Organização da Tese

Neste capítulo introdutório, tendo sido apresentados motivação, objectivos e uma perspectiva histórica, é descrita a organização da tese e são enunciadas as principais contribuições conseguidas.

O segundo capítulo deste relatório apresenta como enquadramento uma revisão bibliográfica sucinta dos temas limítrofes deste estudo, nomeadamente, os sistemas clínicos de ultrassom Doppler, alguns dos métodos de estimação espectral

mais utilizados na quantificação de parâmetros do fluxo sanguíneo, conceitos gerais sobre sistemas paralelos e avaliação de desempenho, terminando com algumas observações sobre a aquisição e visualização de dados.

O terceiro capítulo descreve a metodologia e a justificação das opções tomadas na concepção do sistema de processamento de sinais de ultra-som a implementar para o caso particular da análise de fluxo sanguíneo da artéria carótida – caso-estudo fundamental no desenvolvimento deste trabalho. O sistema de processamento foi dividido, na sua concepção, nos subsistemas de entrada, processamento, visualização e avaliação.

O quarto capítulo contempla uma descrição generalista da concepção e implementação dos subsistemas de aquisição e visualização de dados. O subsistema de aquisição de dados considera os módulos de entrada de dados nas suas diferentes formas alternativas, nomeadamente sinais simulados, sinais clínicos para processamento em tempo diferido e sinais clínicos adquiridos directamente do transdutor e processados em tempo-real. A descrição respeitante ao subsistema de visualização incide sobre a concepção de um módulo, nomeadamente no que respeita à apresentação de espectros de fluxo sanguíneo e pós-processamento desses espectros.

O quinto capítulo relata parte dos esforços envidados no processamento em tempo-real dos estimadores espectrais considerados e descreve os resultados obtidos numa fase inicial do projecto, variando processadores e modelos de computação paralela recorrendo à adição de processadores dedicados.

Os estudos efectuados sobre estimadores espectrais e resultados obtidos, numa fase posterior, utilizando computadores de propósito geral, visando o processamento em tempo-real dos estimadores, surgem descritos no sexto capítulo.

No sétimo capítulo é descrito o subsistema implementado para a avaliação da qualidade estatística e computacional dos métodos de estimação espectral descritos nos capítulos anteriores. Algumas das diferentes implementações dos estimadores espectrais descritas no quinto e sexto capítulos são analisadas visando o cumprimento dos objectivos primordiais de um possível utilizador do sistema, ou seja, por exemplo, melhor par algoritmo-arquitectura, menor tempo de processamento ou maior acuidade de cálculo das velocidades sanguíneas.

O oitavo capítulo discute inicialmente o desenvolvimento e integração do sistema. Tendo anteriormente sido apresentados os diferentes subsistemas com especial ênfase para o caso particular da análise do fluxo em artéria carótida primitiva,

neste capítulo discutem-se os requisitos necessários para a adaptação do sistema por forma a incluir no mesmo a análise e síntese do fluxo sanguíneo noutros vasos sanguíneos e avaliação de outras anomalias de fluxo. No final do capítulo, e a título conclusivo, são tecidas considerações finais sumariando as conclusões deste trabalho e apresentadas algumas linhas de acção conducentes à melhoria do actual sistema analisador de fluxo sanguíneo.

1.3 Contribuições

As contribuições inovadoras do trabalho subjacente à presente tese são sinteticamente as abaixo apontadas. De referir que se encontram em elaboração artigos relatando componentes do trabalho até então não divulgadas à comunidade científica.

- A concepção da estrutura modular do sistema de processamento de sinais, e, a implementação de um protótipo do sistema (Ruano, Moura, Leiria, Garcia & Moreno, 2003). A idealização do sistema considerou-o centrado no subsistema de processamento de sinais, em articulação com os outros subsistemas, de acordo com o tipo de utilização final do sistema global. O protótipo foi implementado integrando a melhor das abordagens ensaiadas para cada um dos subsistemas. Permitindo o protótipo actuações independentes (clínica ou de investigação) e simultâneas, possibilitou-se uma ferramenta vital de aproximação da vertente de investigação à de utilização final, promovendo-se desta forma a transferência de tecnologia.
- A implementação eficiente do algoritmo do método AR - Modified Covariance e a quantificação do impacto da utilização do referido algoritmo para o processamento em tempo-real dos sinais de fluxo sanguíneo dos elementos de processamento *Inmos transputers*, *Texas Instruments TMS320C40*, *Intel i860*, *Analog Devices ADSP 20162* e uma aproximação híbrida *Xilinx FPGA* e *Texas Instruments TMS320C40* (Madeira, Tokhi & Ruano, 1997; Madeira, Tokhi & Ruano, 2000; Madeira, Beltran, González, Nocetti, Tokhi & Ruano, 1998; Madeira, Bellis, Marnane & Ruano, 1998; Madeira, Bellis, Beltran, González, Nocetti, Marnane, Tokhi & Ruano, 1999). Nestes trabalhos investigava-se a adequação entre a plataforma de *hardware* e o algoritmo, tendo sido avaliada a utilização de diversas plataformas aditivas

homogêneas, heterogêneas ou híbridas.

- A implementação eficiente do algoritmo da distribuição tempo-frequência de Choi-Williams e a avaliação quantitativa da adequação das plataformas aditivas homogêneas e heterogêneas usando os elementos de processamento *Inmos transputers*, *Texas Instruments TMS320C40*, *Intel i860* (Madeira, Tokhi & Ruano, 1999).
- O desenvolvimento dos algoritmos de implementação das distribuições tempo-frequência de Choi-Williams para os núcleos modificados *Reduced Interference*, *Alias-Free* e *Reduced Interference Alias-Free* (Leiria, Moura & Ruano, 2001).
- O desenvolvimento do algoritmo de implementação do algoritmo da distribuição tempo-frequência Bessel (Moura, Leiria & Ruano, 2002). A implementação da distribuição tempo-frequência Bessel em computadores pessoais foi avaliada para uma gama de parâmetros aplicável na estimação de diferentes sinais de fluxo sanguíneo, permitindo caracterizar a utilização dos recursos computacionais pelo algoritmo. Na maioria dos casos testados conseguiu-se a implementação em tempo-real deste estimador, sistematicamente excluído pelos investigadores devido à carga computacional que acarretava.
- A implementação do cálculo computacional do sinal analítico da mesma frequência, interpolado ou decimado. A elaboração de algoritmos de cálculo do sinal analítico integrando a transformada discreta de Fourier, uma matriz de transformação, a transformada discreta de Hartley e a convolução com um filtro que aproxima a transformada de Hilbert, de parte ou da totalidade do ciclo cardíaco. Esta última opção metodológica havia sido sempre preterida pelos investigadores devido à elevada carga computacional associada (publicação a elaborar).
- Adaptação dos algoritmos dos estimadores espectrais estudados por forma a poderem ser aplicados a sinais de fluxo sanguíneo de fluxo directo e reverso.
- A optimização dos algoritmos dos estimadores espectrais reformulados visando a optimização do tempo de execução, e, a generalidade de aplicação. A melhoria do tempo de execução é uma contribuição significativa para a

aplicação clínica do sistema em tempo-real e para a investigação em processamento de sinal quando nesta se pretende estudar um vasta gama de sinais. A reformulação dos algoritmos permite a aplicação dos estimadores noutras áreas científicas (publicação em execução).

- A criação de uma nova versão do algoritmo simplificado de implementação do método AR - Modified Covariance para contemplação de dados pertencentes ao domínio complexo. Em alguns sinais de fluxo sanguíneo (nomeadamente em artéria femoral) é requerida a utilização do sinal complexo como indicador das componentes de fluxo directo e reverso. O sinal complexo é também requerido pelo algoritmo quando do sinal de fluxo sanguíneo se calcula o correspondente sinal analítico. Desta forma, generalizou-se o algoritmo para utilizações em outras áreas de aplicação que não a biomédica (publicação elaborada a submeter).
- A reformulação do critério Custo/Benefício para considerar uma nova versão de custo (Leiria, Madeira & Ruano, 1999; Leiria, Moura & Ruano, 2001). A alteração proposta permite a avaliação do subsistema de estimação espectral de uma forma normalizada e independente da arquitectura de implementação.

A referir ainda a participação no desenvolvimento de:

- A conjugação das novas implementações dos métodos e a avaliação do benefício da estimação na obtenção de formas automáticas de escolha dos parâmetros dos estimadores espectrais para estimação adaptativa de sinal de fluxo sanguíneo em artéria carótida (Moura, Leiria & Ruano, 2003).

Tendo uma versão das novas implementações dos diferentes métodos sido partilhada com o grupo de investigação do IIMAS (UNAM, México) foi integrada num sistema para a análise clínica testado em laboratório com médicos do Hospital Hermanos Almejeiras (La Habana, Cuba) e durante uma intervenção cirúrgica (*bypass*) no mesmo hospital em Março e Abril de 2004, respectivamente.

Capítulo 2

Enquadramento

2.1 Introdução

Neste capítulo serão apresentados, de forma muito sucinta, alguns fundamentos deste estudo. Esta descrição assumirá a forma de uma revisão bibliográfica que permitirá situar o âmbito do trabalho aqui apresentado.

Introduzir-se-ão de forma geral os sistemas clínicos de diagnóstico de fluxo sanguíneo por ultra-som Doppler contínuo e pulsado; apresentar-se-ão alguns estudos que incidiram sobre os métodos de estimação espectral de sinais de fluxo sanguíneo, focando, sinteticamente, as conclusões atingidas bem como vantagens e desvantagens que condicionam a sua adopção; sobre os sistemas paralelos, serão apresentadas as características de alguns processadores e modelos de computação paralela; a introdução à visualização de espectros de fluxo sanguíneo e processamento de imagem referirá standards e ferramentas gráficas com relevância para a extracção de informação clínica; a apresentação das métricas de desempenho reproduzirá as tradicional e actualmente utilizadas na avaliação de desempenho de sistemas paralelos e sistemas tempo-real.

Finalmente concluir-se-à assinalando os pontos mais relevantes da matéria exposta.

2.2 Sistemas Clínicos de Ultrason Doppler

As ondas ultra-sónicas são ondas de compressão do meio de propagação geradas por uma fonte de som a vibrar a frequências superiores a 20 kHz. Em aplicações médicas a frequência de ultra-som utilizada situa-se normalmente entre 2 MHz e

10 MHz, conforme a resolução espacial e a profundidade pretendidas (Evans & McDicken, 2000).

O ultra-som é reflectido sempre que existe uma descontinuidade no meio de propagação, isto é, pela superfície de contacto entre dois meios com índices de refração diferentes de forma análoga à da luz. Caso as ondas sejam reflectidas por um objecto em movimento verifica-se uma alteração na frequência. Este fenómeno é designado por efeito Doppler e a diferença entre a frequência do sinal emitido e do sinal recebido é designada por frequência Doppler.

Das várias aplicações médicas de ultra-som Doppler destacam-se a instrumentação de imagiologia ou de medição. Em ambas, verifica-se a insonorização do corpo e são recolhidos e analisados os ecos devolvidos. Na instrumentação de imagiologia é enfatizado o cálculo das coordenadas dos elementos que ecoaram o ultra-som enquanto que a instrumentação de medida privilegia a análise da frequência do ultra-som retornado.

2.2.1 Instrumentação Doppler

Os sistemas Doppler podem ser divididos em duas categorias: onda contínua ou onda pulsada.

As unidades de onda contínua emitem e recebem ultra-som continuamente, o que é normalmente conseguido com dois transdutores. A área amostrada é a intercepção dos feixes emitido e recebido e é normalmente designada por amostra. O feixe de ultra-som, reflectido e refractado pelas partículas constituintes da amostra, apresenta diferenças em relação ao sinal inicialmente transmitido, diferenças essas associadas às características da amostra. Os sistemas de onda pulsada, com um único elemento transdutor, intercalam a emissão com a recepção. A electrónica envolvida é significativamente mais complexa, aumentando o custo destes equipamentos. Os sistemas pulsados têm vindo a ser preferidos aos contínuos por permitirem definir a localização da amostra.

Nos sistemas contínuos, o eco recebido traduz todas as partículas reflectoras existentes entre o transdutor e a profundidade máxima de penetração do feixe de ultra-som. Nos sistemas pulsados, o eco é recebido em determinados intervalos de tempo após a emissão de um feixe de ultra-som, intervalo esse correspondente ao tempo de penetração e retorno do feixe, permitindo limitar a amostra e localizar a profundidade dos elementos reflectores.

Pode-se ainda classificar os sistemas Doppler em função do objectivo que

visam cumprir em sistemas de detecção de velocidade, sistemas *Duplex*, sistemas de detecção de perfis ou ainda sistemas de visualização de velocidade¹.

Os sistemas de detecção de velocidade, muito simples, permitem a verificação da existência ou inexistência de fluxo e a monitorização de alterações desse fluxo.

Os sistemas *Duplex* combinam um sistema pulso-eco *B-scan* e um sistema Doppler. Um sistema *B-scan* apresenta imagens bidimensionais que traduzem a profundidade (em função do atraso no tempo) e a posição do reflector (em função da orientação do feixe que originou o eco), sendo a amplitude do eco traduzida pelo brilho, permitindo definir a orientação do feixe Doppler e a profundidade dos ecos pretendidos.

Os sistemas de detecção de perfis são normalmente sistemas de Doppler pulsado multi-canal, que permitem a recolha de ecos de várias profundidades. O principal inconveniente destes sistemas advém da complexidade electrónica e associado custo.

Os sistemas de visualização de velocidade baseiam-se na produção de uma imagem planar ou bidimensional criada por varrimento da área em análise. Esse varrimento pode ser manual, mecânico ou produzido recorrendo a múltiplos transdutores manipulados electronicamente de forma a ser conseguida a focagem.

É actualmente comum que os sistemas Doppler considerem a detecção da direcção do movimento. A abordagem mais frequente consiste na inclusão de circuitos adicionais que implementam a técnica designada por detecção de quadratura de fase².

2.2.2 Fluxo Sanguíneo

A análise do efeito Doppler provocado por um fluxo sanguíneo, nomeadamente a observação da velocidade do sangue, permite a percepção de fluxo anómalo e, possivelmente, da causa da anomalia.

O fluxo sanguíneo é um fluido heterogéneo não Newtoniano (Evans & McDicken, 2000), consistindo numa suspensão de células (na sua maioria glóbulos vermelhos ou eritrócitos) num meio designado plasma.

O fluxo sanguíneo pode ser considerado estável. Aplica-se a designação de fluxo estável quando a velocidade e pressão em qualquer ponto da corrente se

¹do inglês, *velocity imaging systems*

²do inglês, *quadrature phase detection*

mantém constante no tempo. Em presença de fluxo turbulento pode ser verificada a existência de movimentos de partículas transversais ao eixo do vaso, com variações significativas na velocidade e direcção.

Essencialmente é o perfil de velocidades do fluxo sanguíneo, arterial ou venoso, que determina o sinal Doppler obtido.

No caso de fluxo sanguíneo arterial, é necessário ter em atenção que: o fluxo sanguíneo arterial é pulsátil, podendo ser considerado como a soma de vários componentes de fluxo estável. Há dois tipos de velocidades a ter em consideração num fluxo arterial: a velocidade do sangue e a velocidade de propagação da onda de fluxo sanguíneo. A alteração do diâmetro das artérias durante o ciclo cardíaco, introduz uma oscilação de pressão, alterando o perfil de velocidade e, conseqüentemente, condicionando o sinal Doppler recebido. O sinal Doppler é também afectado pelo movimento da parede arterial, sendo a refração de ultra-som causada não só pelo fluxo sanguíneo como também pelo vaso sanguíneo.

A direcção do fluxo sanguíneo pode ser clinicamente importante, quer em situações normais como anómalas. Por exemplo, em artérias periféricas – como a femoral – a direcção do fluxo normal varia durante o ciclo cardíaco; em casos anómalos, pode verificar-se uma alteração da direcção do fluxo sanguíneo por se ter estabelecido um percurso alternativo de circulação sanguínea.

2.2.3 Análise de Fluxo Sanguíneo

Os objectos de análise são múltiplos e, dependendo do elemento a considerar, uma medida pode reflectir, entre outros factores, a direcção ou perfil de velocidades, o fluxo sanguíneo, pressão, área ou frequência. O resultado produzido pode ser um som, uma imagem, um espectro, ou informação discreta da amplitude e frequência dos ecos recebidos.

O desvio na frequência de ultra-som produzido por uma partícula em movimento, também designada por frequência Doppler (f_d) (Evans & McDicken, 2000), é dada por

$$f_d = f_t - f_r = (2 f_t v \cos \theta) / c \quad (2.1)$$

em que f_t e f_r são, respectivamente, as frequências das ondas transmitida e recebida, v a velocidade da partícula reflectora, θ o ângulo entre a direcção do feixe de ultra-som e a direcção do movimento da partícula e c a velocidade de propagação do ultra-som no meio. Assume-se normalmente na análise de fluxo

sanguíneo que a velocidade de propagação do ultra-som, em tecidos moles, é de 1540 ms^{-1} .

Note-se na equação 2.1 que a frequência Doppler é proporcional à velocidade. Em condições ideais, as potências de um espectro Doppler numa gama de frequências são proporcionais ao volume de partículas que se movimentam a uma velocidade correspondente a essa gama de frequências.

Dependendo do equipamento, o sinal de frequência Doppler pode consistir num sinal complexo contendo informação direccional do fluxo sanguíneo ou pode ser separado, por processamento, em dois canais de fluxos directo e reverso. Esse processamento pode ser analógico ou digital, podendo neste caso, ser considerado na estimação espectral.

Uma estimativa da distribuição das velocidades de um fluxo sanguíneo é patente no espectro de potência Doppler, que pode ser obtido pela aplicação de métodos de estimação espectral.

A representação da variação temporal da forma do espectro de potências Doppler é designada espectrograma e é ilustrada na Figura 2.1.

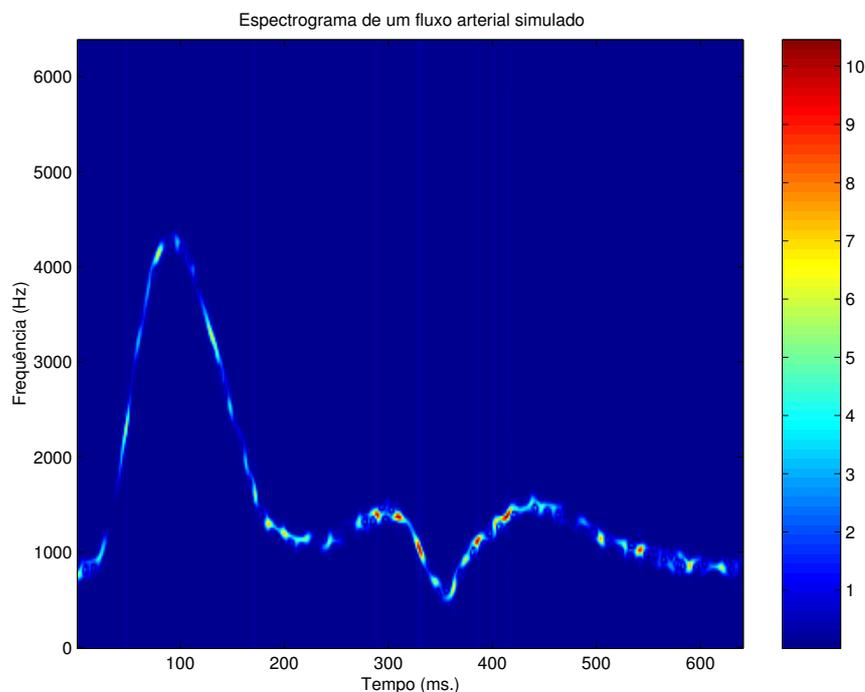


Figura 2.1: Espectro de potências Doppler obtido pela estimação de um sinal simulado de fluxo sanguíneo arterial.

O tempo e frequência são representados nos eixos horizontal e vertical, respectivamente, sendo a potência representada por cores ou níveis de cinzento. Assim, uma linha do espectrograma representa o perfil de velocidades presentes no volume amostrado num determinado instante de tempo.

O pós-processamento do espectrograma (Evans & McDicken, 2000) consiste normalmente na produção de médias ou aplicação de filtros 2D. A representação em espectrograma é bastante significativa para a avaliação qualitativa do fluxo em análise e detecção de anomalias. No entanto, o volume de informação apresentada é tal que é necessária a extração de algumas características antes que sejam retiradas conclusões quantitativas. A determinação de curvas características pode ser conseguida por métodos de processamento analógicos ou digitais, podendo estes últimos ser baseados em técnicas de processamento de sinal – numericamente a partir do espectro de potências – ou processamento de imagem – por determinação dos contornos do espectrograma – ou por métodos de reconhecimento de padrões.

As curvas características relevantes variam com o objecto analisado incluindo frequentemente as curvas de frequência média, frequência máxima ou largura de banda (ver Figura 2.2). A frequência média e a largura de banda, observáveis

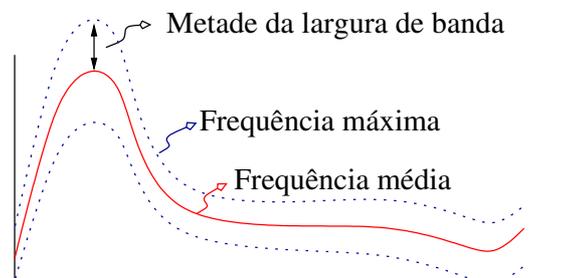


Figura 2.2: Alguns dos indicadores observáveis num espectrograma.

na representação em espectrograma, são dois indicadores fundamentais na determinação de anomalias do fluxo sanguíneo em vasos e válvulas, proporcionais a, respectivamente, velocidade média e turbulência.

2.2.4 Indicadores Clínicos de Fluxo Sanguíneo

A análise e conseqüente diagnóstico de fluxo sanguíneo por Doppler pulsado é feita através da determinação de parâmetros quantitativos obtidos a partir da forma de onda das curvas características, podendo variar com o objecto analisado e com

o seu estado – normal ou indicativo de anomalias.

A Figura 2.3 ilustra alguns dos elementos retirados da curva de frequência média onde S e D correspondem, respectivamente, ao valor máximo em sístole e ao mínimo valor em diástole, M é a altura média da curva e V_{max} indica o valor em frequência em que a velocidade atinge o valor máximo. Esses e outros elementos são utilizados para a determinação dos parâmetros clínicos (Evans & McDicken, 2000) da artéria carótida primitiva ou outros fluxos arteriais que a seguir se descrevem.

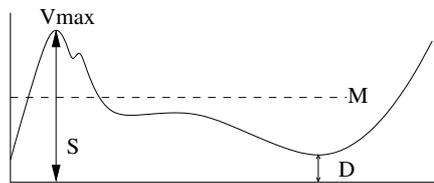


Figura 2.3: Alguns dos elementos quantitativos retirados das curvas características de fluxo sanguíneo da artéria carótida primitiva normal.

Índice de pulsatilidade (PI) permite uma descrição objectiva e quantitativa da característica pulsátil do sangue exibida pela forma da onda de um espectro de potência Doppler.

$$PI = \frac{(S - D)}{M} \quad (2.2)$$

onde S e D são, respectivamente, a máxima e mínima altura do espectro e M a altura média ao longo do ciclo cardíaco. Este indicador é influenciado por vários factores entres os quais estenose *proximal* e *distal* ou ritmo cardíaco entre outros. Este indicador reflecte a resistência da parede dos vasos sanguíneos.

Índice de resistência de Porcelot (RI) indica resistência circulatória e é definido por

$$RI = \frac{(S - D)}{S} \quad (2.3)$$

onde S é a altura máxima do espectro e D é a altura no fim da diástole. Este índice é usado na detecção de alterações do espectro em várias condições patológicas como são exemplos estenose da carótida interna ou hemorragia intra ventricular.

Rácio A/B indica a relação entre a altura dos dois picos da onda correspondente à artéria carótida primitiva, relação essa que varia com a idade e doença. A Figura 2.4 apresenta uma forma de onda da artéria carótida primitiva e os dois picos sistólicos usados na determinação do rácio A/B .

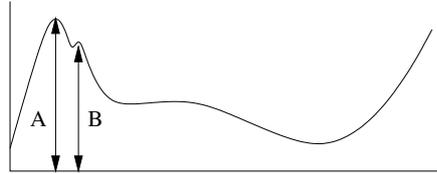


Figura 2.4: Ilustração de uma curva de fluxo sanguíneo característica da artéria carótida primitiva.

Índice altura-largura combina a pulsatilidade do fluxo arterial com a relação temporal do período de sístole no ciclo cardíaco.

$$HWI = PI \frac{T}{t_s} \quad (2.4)$$

onde t_s é a duração do pico sistólico.

Índice de alargamento espectral traduz a distribuição de velocidades num determinado instante. Na presença de anomalias verifica-se um alargamento do espectro no pico sistólico da artéria carótida interna. Esse índice é obtido por

$$SBI = k \frac{f_{max} - f_{min}}{f_{max} + f_{min}} \quad (2.5)$$

em que f_{max} e f_{min} são, respectivamente, os valores de frequência máxima e mínima no pico sistólico e k uma constante experimental de valor 0.47 (Evans & McDicken, 2000).

No caso do fluxo na artéria carótida primitiva, a análise baseia-se na velocidade média e alargamento espectral. Quando são examinadas válvulas cardíacas os indicadores mais relevantes são baseados nas áreas das válvulas ou no declive da curva de frequência máxima. Existem casos em que alguns dos parâmetros de um mesmo indicador são obtidos por Doppler contínuo e outros por Doppler pulsado, ou mesmo por pós-processamento do espectrograma. A detecção de embolias baseia-se na determinação da potência do sinal (amplitude, traduzida pela cor)

mas a classificação da natureza do êmbolo baseia-se na velocidade média e na potência (Leiria, 2004).

2.3 Estimação Espectral

A determinação do espectro de potência Doppler e a extração das curvas características de forma a possibilitar o cálculo de parâmetros clínicos é possível através do processamento digital dos sinais temporais de frequência Doppler.

O método tradicional de análise do espectro de Doppler consiste na utilização de algoritmos baseados na transformada rápida de Fourier (*Fast Fourier Transform* (FFT)) para a obtenção do espectro de fluxo sanguíneo.

Tratando-se o sinal Doppler de um sinal discreto obtido de forma contínua, teoricamente de tamanho infinito, a aplicação de qualquer método de estimação espectral é feita a segmentos do sinal. Devido à natureza não estacionária do sinal de fluxo sanguíneo a aplicação de algoritmos de FFT requer a sua segmentação por aplicação de janelas temporais. Atendendo às características não estacionárias do sinal Doppler, as janelas temporais a aplicar devem ter curta duração (aproximadamente 10 ms. (Cardoso, Ruano & Fish, 1996a)) por forma a atenuar a não estacionaridade no segmento do sinal em análise, assumindo o método a designação de Short Time Fourier Transform (STFT). A aplicação destas janelas, seguida da aplicação de transformadas de Fourier, induz componentes de frequência não integrantes do sinal original, as quais alterarão a estimação da frequência média e largura de banda do espectro (Ruano, 1992; Proakis & Manolakis, 1988).

Uma estratégia frequentemente adoptada para aumentar a resolução temporal dos estimadores espectrais e diminuir a variância das estimativas (Vaitkus & Cobbold, 1988) consiste na utilização de janelas deslizantes ³. Esta estratégia é adoptada com diversos estimadores conforme referido em (Boashash & Black, 1987; Cardoso, 1998; Leiria, 2000).

Um estudo de métodos alternativos que conferissem maior precisão à estimação de sinais de fluxo sanguíneo incidiu sobre métodos paramétricos e o método da variância mínima (Ruano, 1992). O método de variância mínima foi desclassificado nesse estudo comparativo atendendo à pouca fiabilidade na completa execução computacional do método. Esse estudo considerou métodos

³Tradução livre da expressão inglesa *running windows*

paramétricos, classificáveis em três categorias – auto regressivos (AR), médias móveis (MA) e auto regressivos de médias móveis (ARMA) – identificando o método AR Modified Covariance (MC) de quarta ordem como o mais adequado.

A característica de não estacionaridade do sinais de fluxo sanguíneo motivou outro estudo que incidiu sobre a aplicabilidade de distribuições tempo-frequência (*Time Frequency Distributions* (TFD)) da classe de Cohen (Cardoso, 1998). Esse estudo considerou as TFD de Bessel, de Wigner-Ville e Choi-Williams, identificando esta última como a mais adequada.

Assim, foram seleccionados como casos de estudo para o processamento em tempo-real, os métodos de estimação AR-MC e TDF de Choi-Williams, analisados em detalhe nos capítulos 5 e 6. A qualidade estatística da distribuição tempo-frequência de Bessel referida por vários autores (Guo, Durand & Lee, 1994a; Cardoso, Ruano & Fish, 1996a; Cardoso, Ruano & Fish, 1996b; Cardoso, 1998) motivou o estudo da implementação deste estimador na fase final do estudo referente às plataformas não aditivas. A STFT será usada como termo de comparação na avaliação dos métodos de estimação por ser a tradicionalmente usada na instrumentação ultra-sónica comercial.

2.3.1 Short Time Fourier Transform

A transformada discreta de Fourier (*Discrete Fourier Transform* (DFT)) é dada por (Brigham, 1988):

$$X(n) = \sum_{k=0}^{N-1} x_0(k) e^{-j2\pi nk/N} \quad n = 0, 1, \dots, N-1 \quad (2.6)$$

onde $x_0(k)$ é o sinal temporal de dimensão N e $X(n)$ o sinal transformado para o domínio da frequência.

A transformada rápida de Fourier (*Fast Fourier Transform* (FFT)) consiste num algoritmo computacional, publicado em 1965 por Cooley e Tukey, que reduz a complexidade algorítmica do cálculo da DFT. Devido às inúmeras aplicações da DFT em vários ramos da ciência, a implementação da FFT encontra-se extensivamente estudada, frequentemente fazendo parte das bibliotecas de funções fornecidas pelos fabricantes de hardware ou software de processamento de sinal.

A STFT evita a questão da não estacionaridade do sinal estimando a potência espectral no tempo t da transformada de Fourier do sinal multiplicado por uma

função de janela $w(t)$ centrada no instante t . A dimensão da janela deve ser adequada de forma a que o sinal possa ser considerado estacionário em cada um dos segmentos em que é dividido. Assim sendo

$$STFT(t, \omega) = \left| \int e^{-j\omega t'} x(t') w(t' - t) dt' \right|^2 \quad (2.7)$$

2.3.2 AR - Modified Covariance

O estimador espectral AR Modified Covariance (MC) (Kay, 1988) permite a estimação da densidade espectral de potência, $P_{MC}(f_n)$, de um sinal discreto no tempo por

$$\hat{P}_{MC}(f_n) = \frac{\hat{\sigma}^2}{|A(f_n)|^2} = \frac{\hat{\sigma}^2}{\left| 1 + \sum_{k=1}^p a[k] z^{-k} \Big|_{z=e^{j2\pi f_n}} \right|^2} \quad (2.8)$$

onde $\hat{\sigma}^2$ representa a estimativa da variância do ruído branco do sinal e $A(f_n)$ constitui uma função polinomial de ordem p das estimativas $a[k]$ dos parâmetros auto regressivos do modelo.

A estimativa da variância do ruído branco do sinal é calculada por

$$\hat{\sigma}^2 = c_{xx}[0, 0] + \sum_{k=1}^p \hat{a}[k] \cdot c_{xx}[0, k] \quad (2.9)$$

onde as estimativas $\hat{a}[k] (k = 1, \dots, p)$ são obtidas pela resolução do sistema de equações lineares de covariância modificada

$$\begin{bmatrix} c_{xx}[1, 1] & c_{xx}[1, 2] & \dots & c_{xx}[1, p] \\ c_{xx}[2, 1] & c_{xx}[2, 2] & \dots & c_{xx}[2, p] \\ \vdots & \vdots & \ddots & \vdots \\ c_{xx}[p, 1] & c_{xx}[p, 2] & \dots & c_{xx}[p, p] \end{bmatrix} \begin{bmatrix} \hat{a}[1] \\ \hat{a}[2] \\ \vdots \\ \hat{a}[p] \end{bmatrix} = - \begin{bmatrix} c_{xx}[1, 0] \\ c_{xx}[2, 0] \\ \vdots \\ c_{xx}[p, 0] \end{bmatrix} \quad (2.10)$$

onde cada elemento $c_{xx}(i, j)$ da matriz de covariância e do vector do lado direito da Equação 2.10 (com $j = 0$) é obtido por

$$c_{xx}[i, j] = \frac{1}{2(N-p)} \cdot (c1_{xx}[i, j] + c2_{xx}[i, j]) \quad (2.11)$$

onde

$$c1_{xx}[i, j] = \sum_{n=p}^{N-1} x^*[n-i] \cdot x[n-j] \quad (2.12)$$

e

$$c2_{xx}[i, j] = \sum_{n=0}^{N-1-p} x[n+i].x^*[n+j]. \quad (2.13)$$

em que $x[n]$ e N significam o sinal discreto a considerar e o comprimento do segmento de dados, respectivamente.

2.3.2.1 Estudos Anteriores

A complexidade algorítmica do estimador AR - Modified Covariance e os requisitos computacionais de uma implementação em tempo-real justificaram várias abordagens ao problema.

A identificação deste estimador para a aplicação a sinais de fluxo sanguíneo e abordagem detalhada pode ser encontrada em (Ruano, 1992). Esse estudo inclui simplificações algorítmicas e implementações deste estimador espectral sobre um mesmo tipo de elementos de processamento (*transputers* T805 a 20 MHz) e usando OCCAM, uma linguagem de programação específica desse processador.

É referida em (Ruano, García-Nocetti, Fish & Fleming, 1993) a possibilidade de implementação desse estimador em tempo-real, ainda que isso não tivesse sido conseguido para os segmentos de dados de maior comprimento. Nessa implementação foram estudadas algumas alternativas de paralelização do algoritmo e estudadas implementações em diversas topologias, nomeadamente *pipeline processor farm* e *processor farm* em árvore ternária. Os melhores resultados foram obtidos com uma aplicação *processor farm* numa topologia em árvore ternária.

Na continuação desse estudo, é conseguida uma melhoria nos resultados (Ruano, García-Nocetti, Fish & Fleming, 1992) na estimação concorrente de vários segmentos de dados (paralelismo de dados).

Outro grupo de investigação, sobre a utilização clínica deste algoritmo, estudou a aplicabilidade de algoritmos genéticos (González & García-Nocetti, 1994) à estimação do sinal Doppler de fluxo sanguíneo, visando a redução da complexidade algorítmica. O algoritmo genético mostra-se mais apropriado para valores da ordem do modelo mais elevados. Numa implementação paralela (González, García-Nocetti & Vásquez, 1995) do algoritmo genético o tempo-real não foi ainda obtido. Um compromisso baseado no tempo médio de cálculo de vários segmentos apresenta melhores resultados, se for aceitável alguma latência. Considerando a aplicação clínica os tempos obtidos não respeitam os requisitos de tempo-real.

É referido o desenvolvimento de um nó heterogéneo composto por *transputers* e DSP (TMS320C30) (García-Nocetti, González & Flores, 1994). Uma parte muito significativa desse desenvolvimento diz respeito à comunicação inter-processador, utilizando um *link adapter* e *hardware* dedicado. Esta implementação não atingia tempo-real para a dimensão mais elevada dos segmentos de dados.

Esse nó heterogéneo é integrado numa arquitectura de *transputers* (García-Nocetti, Flores & González, 1995) e é desenvolvida uma ferramenta de *software* para a integração do nó na plataforma, constituindo uma arquitectura heterogénea. O estudo da implementação do estimador espectral AR - Modified Covariance na arquitectura conduz à conclusão de que o uso da arquitectura heterogénea é eficiente, permitindo reduzir o número de *transputers* necessários pela exploração das capacidades de processamento rápido oferecidas pela DSP no nó heterogéneo.

A abordagem seguida por (Ruano, García-Nocetti, Fish & Fleming, 1992) parece ser a mais indicada para a implementação em tempo-real do estimador tendo sido objecto de estudo mais aprofundado (Madeira, 1997) utilizando arquitecturas de processamento paralelo homogéneas compostas por *transputers* e DSP's. As arquitecturas compostas por DSP's permitem cumprir os requisitos de tempo-real. No entanto, é atribuída à plataforma que suporta as arquitecturas ensaiadas a responsabilidade por uma degradação dos tempos totais de execução, nomeadamente através da adição de elevados tempos de comunicação com a máquina hospedeira, comunicação essa assegurada por um *transputer*.

A utilização de janelas sobrepostas com o estimador AR-MC, nem sempre considerada, eleva a carga computacional do estimador.

Das conclusões e considerações acima e ressaltando a relevância da implementação adoptando paralelismo de dados impõe-se como necessária uma análise mais extensa da implementação do estimador AR-MC em diferentes arquitecturas, usando DSP's e outros processadores, recorrendo a técnicas de processamento paralelo.

2.3.3 Distribuição Tempo-Frequência de Choi-Williams

A distribuição exponencial (Choi & Williams, 1989), vulgarmente designada de Choi-Williams (CWD) é um dos métodos de estimação espectral adequados a sinais não estacionários e é frequentemente enquadrada numa gama de outras distribuições tempo-frequência designadas como pertencendo à classe de Cohen.

Uma TFD da classe de Cohen (Cohen, 1989) pode ser descrita como uma transformada de Fourier da função generalizada de autocorrelação

$$TFD(t, \omega) = \int_t^{+\infty} R'_x(t, \tau) e^{-j\omega\tau} \partial\tau \quad (2.14)$$

sendo

$$R'_x(t, \tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} R_x(t, \tau) \Psi(t - \mu, \tau) \partial\mu \quad (2.15)$$

$R_x(t, \tau)$ é a função instantânea da autocorrelação do sinal $x(t)$ e é dada por

$$R_x(t, \tau) = x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) \quad (2.16)$$

e $\Psi(t - \mu, \tau)$ é o núcleo do domínio da autocorrelação, definido por

$$\Psi(t, \tau) = \int_{-\infty}^{+\infty} \Phi(\theta, \tau) e^{-j\theta t} \partial\theta \quad (2.17)$$

sendo $\Phi(\theta, \tau)$ o núcleo que caracteriza a TFD.

A CWD apresenta um núcleo exponencial $\Phi(\theta, \tau) = e^{\theta^2\tau^2/\sigma}$. A função generalizada de autocorrelação é

$$R'_x(t, \tau) = \int_{\mu} \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{(-\frac{(\mu-\tau)^2}{4\tau/\sigma})} x\left(\mu + \frac{\tau}{2}\right) x^*\left(\mu - \frac{\tau}{2}\right) \partial\mu \quad (2.18)$$

onde $\sigma > 0$ é um factor de escala que controla a suavidade da distribuição, isto é, quanto maior for σ , mais picos apresenta a distribuição e maiores serão os termos cruzados (Choi & Williams, 1989).

A forma discreta da CWD, designada DCWD, é dada por

$$DCWD(n, k) = 2 \sum_{\tau=-L}^L W_N(\tau) e^{-j2\pi k\tau/N} \left[\sum_{\mu=-m}^{+m} W_M(\mu) \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(n + \mu + \tau) x^*(n + \mu - \tau) \right] \quad (2.19)$$

onde $W_N(\tau)$ é uma janela simétrica com valores nulos fora do intervalo $-L \leq \tau \leq L$ e $W_M(\mu)$ é uma janela rectangular com valor unitário no intervalo $-m \leq \mu \leq m$. O formato e tamanho desta janela determinará a resolução na frequência.

O valor de m indica o intervalo dentro do qual a função de autocorrelação será estimada.

A formalização discreta da CWD exhibe uma periodicidade em π que se pode traduzir em *aliasing* quando o estimador é aplicado a sinais reais. A abordagem mais comum consiste na aplicação do estimador não ao sinal $x(t)$ mas sim ao sinal *analítico* de uma realização do sinal real; como alternativa poder-se-à aumentar a frequência de amostragem para o dobro da frequência de Nyquist (Boashash & Black, 1987; Choi & Williams, 1989; Cardoso, 1998). Diferentes abordagens à constituição de representações analíticas do sinal a estimar serão apresentadas na secção 2.3.5.

Outras alternativas ao núcleo proposto por (Choi & Williams, 1989) são descritas na secção 2.3.3.2.

2.3.3.1 Estudos Anteriores

A carga computacional de uma distribuição tempo-frequência é de tal forma significativamente superior à do método padrão baseado na transformada de Fourier que visa substituir, que Choi e Williams ao proporem a distribuição exponencial (Choi & Williams, 1989) incluem uma definição para sinais discretos visando a eficiência computacional.

É também proposto um algoritmo para a implementação em tempo-real da CWD usando processadores paralelos (Barry, 1992). O equipamento então usado, um Zenith 386 com coprocessador matemático e frequência de relógio de 16 MHz não permite avaliar o desempenho do algoritmo então proposto.

Um estudo sobre a aplicabilidade de algumas distribuições tempo-frequência, entre as quais a TFD de Choi-Williams, à estimação de sinais de ultra-som Doppler de fluxo sanguíneo na artéria femoral (Guo, Durand & Lee, 1994a) compara o desempenho estatístico e o custo computacional. A descrição pormenorizada das implementações testadas não é apresentada, sendo apenas comparados os tempos de execução relativos à criação do espectrograma. A melhor distribuição, Bessel, apresenta tempos de execução dez vezes superiores, enquanto que para a CWD os tempos de execução são apenas cerca de quatro vezes superiores aos do espectrograma.

A aplicação das TFD à análise de sinais ultra-som Doppler da artéria carótida é descrita em (Cardoso, 1998). Os valores óptimos da STFT, Wigner-Ville, Choi-Williams e Bessel são inicialmente estudados (Cardoso, Ruano & Fish, 1996a),

sendo a CWD eleita, pelo critério Custo/Benefício, como a melhor distribuição (Cardoso, Ruano & Fish, 1996b). Em (Cardoso, 1998) é também relatado o estudo da implementação em tempo-real, usando técnicas de processamento paralelo, da CWD, para aplicação a dados complexos ou reais e algumas simplificações algorítmicas possíveis em um ou outro caso. Especial atenção é dada à autocorrelação do sinal e ao cálculo do espectro. São descritos os algoritmos em *pipeline* e apresentadas diferentes estratégias de distribuição de carga. Duas dessas estratégias são concretizadas (Cardoso, Fish & Ruano, 1996) numa arquitectura de *transputers* com o máximo de quatro elementos de processamento. O desempenho das aplicações em OCCAM é avaliado, sendo demonstrado que uma das abordagens descritas permite a estimação de segmentos reais em tempo-real, ou seja, um segmento pode ser estimado em tempo inferior ao de amostragem. Os tempos de inicialização, bem como os tempos de transferência de dados entre a plataforma de *transputers* e a máquina hospedeira não são referidos no estudo.

Os tempos de transferência de dados, como foi verificado em (Madeira, 1997), podem comprometer a estimação em tempo-real. Assim, e considerando muito relevantes as abordagens propostas em (Cardoso, 1998), impõe-se estender o estudo a outras arquitecturas de processamento.

A utilização do sinal *analítico* conforme utilizada em (Cardoso, 1998) altera a carga computacional do algoritmo e impossibilita as simplificações algorítmicas possíveis nos casos em que os dados são reais.

Acresce que os relativamente bons resultados da CWD descritos em (Guo, Durand & Lee, 1994a) justificam que o estudo da implementação em tempo-real seja estendido de forma a considerar outros valores para além daqueles que proporcionam os melhores resultados para a artéria carótida. Pela mesma razão, é incluído o estudo da implementação de distribuições tempo-frequência de Choi Williams de núcleo modificado.

2.3.3.2 TFD de Choi-Williams com Núcleos Modificados

O núcleo confere a uma distribuição tempo-frequência as suas propriedades. A utilização de núcleos modificados permite contornar algumas limitações da distribuição ou dotá-la de propriedades desejáveis (Jeong & Williams, 1992b; Jeong & Williams, 1992a; Leiria, Moura & Ruano, 2001; Cloutier, Chen & Durand, 2001).

As *Reduced Interference Distributions* (Jeong & Williams, 1992b) são uma classe de distribuições tempo-frequência que garantem as desejáveis propriedades

de suporte no tempo e na frequência.

A forma discreta da *Reduced Interference Choi-Williams Distribution* (RI-CWD) é dada por

$$RI - CWD(n, k) = 2 \sum_{\tau=-L}^L W_N(\tau) e^{-j2\pi k\tau/N} \left[\sum_{\mu=(-\tau-|\tau|)/2}^{(-\tau+|\tau|)/2} \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(n + \mu + \tau) x^*(n + \mu) \right] \quad (2.20)$$

onde $W_N(\tau)$ é uma janela simétrica com valores nulos fora do intervalo $-L \leq \tau \leq L$ e σ é o factor de escala do núcleo.

A utilização da formulação discreta da RI-CWD também pode dar origem a *aliasing*. O método vulgarmente utilizado para contornar esta limitação consiste na utilização da forma analítica do sinal a estimar.

A adopção de um outro núcleo modificado, conforme (Jeong & Williams, 1992a), permite obviar esta limitação, dando origem a uma distribuição tempo-frequência cuja formulação discreta não provoca *aliasing* no sinal. A forma discreta da *Alias Free Choi-Williams Distribution* (AF-CWD) é dada por

$$AF - CWD(n, k) = 2 \sum_{\tau=-L}^L W_N(\tau) e^{-j2\pi k\tau/N} \left[\sum_{\mu=-m}^{+m} \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{(2\mu+\tau)^2}{16\tau^2/\sigma}} x(n + \mu + \tau) x^*(n + \mu) \right] \quad (2.21)$$

onde m determina o intervalo dentro do qual a função de autocorrelação será estimada, σ é um factor de escala do núcleo e $W_N(\tau)$ é uma janela simétrica com valores nulos fora dos intervalos $-L \leq \tau \leq L$.

Pode-se ainda considerar uma distribuição que reuna as características de interferência reduzida e sem *aliasing*, a *Reduced Interference Alias Free Choi-Williams Distribution* (RIAF-CWD) cuja formulação discreta é dada por

$$RIAF - CWD(n, k) = 2 \sum_{\tau=-L}^L W_N(\tau) e^{-j2\pi k\tau/N} \left[\sum_{\mu=(-\tau-|\tau|)/2}^{(-\tau+|\tau|)/2} \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{(2\mu+\tau)^2}{16\tau^2/\sigma}} x(n + \mu + \tau) x^*(n + \mu) \right] \quad (2.22)$$

onde $W(\tau)$ é uma janela simétrica com valores nulos fora do intervalo $-L \leq \tau \leq L$ e σ é um factor de escala do núcleo.

2.3.4 Distribuição Tempo-Frequência de Bessel

A distribuição Bessel (BD) (Guo, Durand & Lee, 1994b), uma das distribuições tempo-frequência da classe de Cohen, tem sido referida como um estimador eficiente em várias áreas de aplicação e particularmente no processamento de sinais biomédicos.

A sua formulação é semelhante à da distribuição de Choi-Williams, sendo o núcleo da distribuição distinto. Para a distribuição Bessel, o núcleo é definido como

$$\Phi(\xi, \tau) = \frac{J_1(2\pi\alpha\xi\tau)}{\pi\alpha\xi\tau} \quad (2.23)$$

onde J_1 é uma função de Bessel de primeira ordem de grau um, ξ e τ representam, respectivamente, as variáveis frequência e tempo, e $\alpha > 0$ um factor de escala.

A forma discreta da BD, (Guo, Durand & Lee, 1994b), designada DBD, de um sinal complexo discreto $x(n)$ é dada por

$$DBD_f(n, k) = 2 \sum_{\tau=-\infty}^{+\infty} W_N(\tau) e^{-j2\pi k\tau/N} \left[\sum_{\mu=-2\alpha|\tau|}^{+2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(n + \mu + \tau) x^*(n + \mu - \tau) \right] \quad (2.24)$$

com

$$\left[\sum_{\mu=-2\alpha|\tau|}^{+2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(n + \mu + \tau) x^*(n + \mu - \tau) \right]_{\tau=0} = x(n)x^*(n) \quad (2.25)$$

onde n e k correspondem, respectivamente, às variáveis discretas tempo e frequência e $W_N(\tau)$ é uma janela simétrica de valores nulos fora do intervalo $[-N/2, N/2]$.

Tal como a DCWD, esta formalização discreta pode dar origem a *aliasing*. Quando um sinal real é usado, verifica-se vulgarmente uma das seguintes alternativas: 1) a frequência de amostragem é elevada para além do dobro da frequência de Nyquist, ou, 2) usa-se a forma analítica do sinal (Guo, Durand & Lee, 1994a; Guo, Durand & Lee, 1994b). A primeira opção é facilmente implementável, com a contrapartida de elevar para o dobro a carga computacional. Diferentes abordagens

à constituição de representações analíticas do sinal a estimar serão apresentadas na secção 2.3.5.

Caso o sinal em análise seja complexo, contendo assim informação relevante nas componentes real e imaginária, o uso da distribuição Bessel com um núcleo Alias-Free poderá produzir melhores resultados (Guo, Durand & Lee, 1994b).

2.3.4.1 Estudos Anteriores

Alguns estudos já referidos sobre a distribuição de Choi-Williams, nomeadamente na estimação de sinais biomédicos, referem também a distribuição Bessel. Esta distribuição é referida como muito útil para a análise tempo-frequência de sinais Doppler de fluxo sanguíneo em (Cardoso, Ruano & Fish, 1996a; Guo, Durand & Lee, 1994a) e de sons do coração em (Guo, Durand & Lee, 1994b). Nos dois primeiros casos a carga do algoritmo foi motivo para a selecção de outro estimador. No entanto, o desempenho estatístico deste estimador motivou que o seu estudo fosse incluído neste trabalho, ainda que após a fase inicial do estudo das implementações nas plataformas aditivas.

Alguns pontos de índole prática, associados ao tipo de sinais em estudo, e que devem ainda ser considerados, serão detalhados na descrição do algoritmo implementado.

2.3.5 Sinal *Analítico*

O cálculo da representação complexa de um sinal real – sinal *analítico* – surgiu inicialmente na implementação das distribuições tempo-frequência da classe de Cohen, nomeadamente Choi-Williams e Bessel. O procedimento de cálculo do sinal *analítico*, segundo o procedimento de Gabor (Cohen, 1995), recorre ao cálculo da transformada de Hilbert. De referir que a transformada de Hilbert é também utilizada na separação das componentes de fluxo sanguíneo directo e reverso.

Assim, foram revistas algumas abordagens a aproximações à transformada de Hilbert e ao sinal *analítico*.

2.3.5.1 Formulação Discreta do Sinal *Analítico*

O sinal analítico é um sinal de banda lateral única ⁴ obtido pelo método de fase, baseado no conjugado harmónico (transformada de Hilbert).

⁴em inglês, *single side band*

O sinal analítico X_a é um sinal contínuo complexo definido por

$$X_a(t) = x(t) + j H\{x(t)\} \quad (2.26)$$

em que a componente real é a réplica do sinal real e a componente imaginária é a transformada de Hilbert do sinal original.

Embora a analiticidade no domínio discreto não possa ser verificada pode, no entanto, ser entendida com a sequência discreta $s_a(nT)$ tal que

$$s_a(nT) = s_a(t)|_{t=nT} \quad (2.27)$$

onde $T = 1/f_s$, f_s a frequência Nyquist de amostragem e $s_a(t)$ um sinal *analítico* complexo (Picone, Prezas, Hartwell & LoCicero, 1988).

A forma analítica de um sinal discreto pode ser obtida quer no domínio da frequência – recorrendo a transformadas discretas de Fourier ou de Hartley, – quer no domínio do tempo – pela convolução com filtros que aproximam a transformada de Hilbert – conforme se expõe de forma resumida.

2.3.5.2 Domínio da Frequência via DFT

A forma analítica de um sinal discreto pode ser obtida aplicando uma DFT, duplicando as frequências positivas e anulando as frequências negativas e, finalmente, aplicando a transformada inversa.

Três formulações são descritas em (Marple Jr., 1999) para a obtenção de um sinal *analítico* com a mesma frequência do sinal que lhe deu origem - designado *standard* - e para a obtenção de sinais decimados ou interpolados. A aplicação do sinal *analítico standard* a N pontos do sinal original produz um sinal com a mesma frequência cujos pontos são obtidos:

- calculando os N pontos da transformada discreta de Fourier $X[m]$
- formando a transformada do sinal *analítico* com N pontos de maneira que

$$Z[m] = \begin{cases} X[0] & \text{para } m = 0 \\ 2 X[m] & \text{para } 1 \leq m \leq \frac{N}{2} - 1 \\ X[N/2] & \text{para } m = N/2 \\ 0 & \text{para } \frac{N}{2} + 1 \leq m \leq N - 1 \end{cases} \quad (2.28)$$

- calculando a transformada inversa de Fourier, também com N pontos, aplicando o factor de escala de $1/N$.

A aplicação do sinal *analítico decimado* produz um sinal complexo de frequência f_s/T que, assumindo que N é par, pode ser obtido:

- calculando os N pontos da transformada discreta de Fourier $X[m]$
- formando a transformada do sinal *analítico* com $N/2$ pontos de maneira que

$$Z[m] = \begin{cases} X[0] + X[N/2] & \text{para } m = 0 \\ 2X[m] & \text{para } 1 \leq m \leq \frac{N}{2} - 1 \end{cases} \quad (2.29)$$

- calculando a transformada inversa de Fourier com $N/2$ pontos e aplicando um factor de escala de $1/2$.

Assumindo também que N é par e M é o factor de interpolação, o sinal *analítico interpolado* pode ser obtido:

- calculando os N pontos da transformada discreta de Fourier $X[m]$
- formando a transformada do sinal *analítico* com NM pontos tal que

$$Z[m] = \begin{cases} X[0] & \text{para } m = 0 \\ 2X[m] & \text{para } 1 \leq m \leq \frac{N}{2} - 1 \\ X[N/2] & \text{para } m = \frac{N}{2} \\ 0 & \text{para } \frac{N}{2} + 1 \leq m \leq NM - 1 \end{cases} \quad (2.30)$$

- calculando a transformada inversa de Fourier com NM pontos e aplicando um factor de escala de M .

2.3.5.3 Domínio da Frequência via DHaT

A transformada de Hartley (HaT) é uma transformada inteira real e é matematicamente equivalente à transformada de Fourier aplicada a dados reais. As principais vantagens na sua aplicação computacional residem em não requerer aritmética complexa e ser quase recíproca considerando que o par transformada directa e inversa difere apenas num factor de escala (Hahn, 1996).

A transformada de Hartley discreta (denotada DHaT) de uma sequência discreta $u(i)$ é definida por

$$DHaT(k) = \sum_{i=0}^{N-1} u(i)cas(w) \quad w = 2\pi ik/N \quad (2.31)$$

onde $cas(w) = \cos(w) + \sin(w)$. A transformada inversa tem a forma

$$u(i) = \frac{1}{N} \sum_{k=0}^{N-1} DHaT(k)cas(w) \quad w = 2\pi ik/N \quad (2.32)$$

A derivação da transformada de Hilbert pode ser conseguida pela transformada inversa de Hartley (Hahn, 1996) da fórmula

$$v(i) = \sum_{k=0}^{N-1} sgn(k)sgn(N/2 - k)Ha(N - k)cas(w) \quad (2.33)$$

ou eventualmente

$$v(i) = \sum_{k=1}^{N-1} sgn(N/2 - k)Ha(N - k)cas(w) \quad (2.34)$$

onde $Ha(k)$ é $DHaT(u(i))$. A função *signum* (sgn) pode ser entendida como uma distribuição *signum* definida como

$$sgn(f) = \begin{cases} 1 & \text{para } f > 0 \\ 0 & \text{para } f = 0 \\ -1 & \text{para } f < 0 \end{cases}$$

DFT via DHaT

É possível obter a transformada discreta de Fourier $F(v)$ de um sinal real $u(i)$ recorrendo à transformada discreta de Hartley (Bracewell, 2000). Separando a sequência par $E(v)$ e ímpar $O(v)$ da transformada discreta de Hartley $H(v)$ tal que,

$$E(v) = \frac{H(v) + H(N - v)}{2}$$

$$O(v) = \frac{H(v) - H(N - v)}{2}$$

obtem-se

$$F(v) = E(v) - iO(v) \quad (2.35)$$

O cálculo da potência do espectro dispensa a reconstituição do sinal complexo, podendo ser obtida por (Bracewell, 2000)

$$Z^2 = \frac{H(v)^2 + H(-v)^2}{2} \quad (2.36)$$

2.3.5.4 Domínio do Tempo por Filtragem

O sinal complexo discreto $s_a(nT)$ corresponde a (Picone, Prezas, Hartwell & LoCicero, 1988)

$$s_a(nT) = s(nT) + j\hat{s}(nT) \quad (2.37)$$

onde $\hat{s}(nT)$ é obtido a partir do sinal discreto $s(nT)$,

$$\hat{s}(nT) = \sum_{m=-\infty}^{+\infty} s((n-m)T) h(m) \quad (2.38)$$

pela convolução com o filtro $h(m)$, que aproxima a transformada de Hilbert no domínio discreto e é geralmente definida por

$$h(m) = \begin{cases} \frac{\sin^2((\pi m)/2)}{(\pi m)/2} & m \neq 0 \\ 0 & m = 0 \end{cases} \quad (2.39)$$

A dimensão do filtro referida na bibliografia varia entre 79 pontos (Boashash & Black, 1987) e 81 pontos (Picone, Prezas, Hartwell & LoCicero, 1988).

Mais recentemente é referido um filtro FIR com dois pontos que aproxima a transformada de Hilbert (Moura & Monteiro, 2002).

2.3.5.5 Separação de Fluxo Direccional

A transformada de Hilbert é também utilizada na separação das componentes direccionais do fluxo sanguíneo de um sinal de frequência Doppler complexo quando tal não foi feito de forma analógica.

Quando essa separação do fluxo direccional é feita por processamento digital de sinal, assume-se que o sinal Doppler pode ser modelado por um sinal complexo

(*analítico*) expresso por (Vaitkus & Cobbold, 1988)

$$\tilde{X}(t) = X_f(t) + j X_q(t) \quad (2.40)$$

em que $X_f(t)$ e $X_q(t)$ são, respectivamente, as componentes de banda base em fase e em quadratura. $X_d(t)$ e $X_r(t)$, respectivamente as componentes de fluxo directo e reverso, podem ser obtidas por

$$\begin{aligned} X_d(t) &= X_f(t) + H\{X_q(t)\} \\ X_r(t) &= X_f(t) - H\{X_q(t)\} \end{aligned} \quad (2.41)$$

em que $H\{\cdot\}$ denota a transformada de Hilbert.

Das diversas formas de extrair as componentes de fluxo direccional referidas em (Evans & McDicken, 2000) reparam-se a técnica de filtro de fase⁵ e a transformada complexa de Fourier.

A técnica de filtro de fase simplifica a expressão apresentada na Eq. 2.41 em que as componentes direccionais são obtidas por

$$\begin{aligned} X_d(t) &= D\{X_q(t)\} + H\{X_f(t)\} \\ X_r(t) &= D\{X_q(t)\} - H\{X_f(t)\} \end{aligned} \quad (2.42)$$

em que $D\{\cdot\}$ denota um atraso temporal equivalente à duração do cálculo da transformada de Hilbert. Esta abordagem é referida em (García, Moreno, Solano, Berragán, Sotomayor & Fuentes, 2002).

A utilização da transformada complexa de Fourier consiste em atribuir as componentes em fase e em quadratura às partes real e imaginária do sinal a estimar, verificando-se que a parte positiva do espectro se refere ao sinal de fluxo directo e a parte negativa ao fluxo reverso.

2.3.6 Parâmetros Espectrais

O cálculo de parâmetros relevantes ao diagnóstico clínico é feito tendo por base índices apurados nas curvas características extraídas do espectro de potência Doppler. As curvas características expõem a variação temporal de grandezas, sendo

⁵do inglês, *phasing-filter technique*

as mais frequentemente usadas a frequência média, largura de banda e frequência máxima.

Vários factores determinam variações fisiológicas do tempo entre batimentos cardíacos. Para minimizar a influência de tais variações nos indicadores clínicos é normal considerar curvas características médias de vários ciclos cardíacos. Para tal é necessário segmentar um registo cardíaco contínuo. A segmentação por ciclo cardíaco é por vezes conseguida com o auxílio de um sinal de electrocardiograma sendo o início de cada ciclo determinado pela detecção do pico R.

Como nem sempre esse sinal está disponível é frequente o recurso a técnicas de cálculo da média de sinais com deslocamentos de fase ⁶(Kitney & Giddens, 1983; Challis & Kitney, 1990). Estes métodos são computacionalmente exigentes porque consideram o recurso iterativo a correlações cruzadas para o alinhamento dos espectros. Acresce ainda que, em indivíduos que não sofram de arritmias, a duração da sístole é normalmente pouco variável, verificando-se que as principais variações ocorrem na diástole.

O algoritmo *Foot Pulse Seek* (FPS) proposto em (Evans, 1988) segmenta em ciclos cardíacos um registo contínuo pela análise da curva da frequência máxima. O início da aceleração sistólica é determinado em oito passos:

1. É criada uma curva suavizada pela média móvel de cada cinco pontos;
2. Essa curva é dividida em secções de N pontos, sendo assinalado o máximo de cada secção;
3. Os máximos assinalados podem ser verdadeiros máximos locais ou podem estar no limite de uma secção. Os valores no limite da secção são comparados com o valor da curva na secção adjacente, sendo ignorados os falsos máximos;
4. é verificado se existe mais do que um valor máximo na vizinhança de N pontos de cada máximo, sendo ignorados os de menor amplitude. Posto isto, é esperado que estejam assinalados apenas os valores de maior amplitude.
5. É criada uma curva correspondente ao diferencial da curva suavizada pela aplicação da ponderação $(-2, -1, 0, 1, 2)$ a cada grupo de cinco pontos.

⁶do inglês, *phase shift averaging*

6. São assinalados os valores máximos da curva diferencial que ocorrem até M pontos antes dos máximos determinados sobre a curva anterior. Estes pontos correspondem aos pontos de maior variação (gradiente) da curva de frequência máxima.
7. É criada a curva com o segundo diferencial aplicando a mesma ponderação à curva do primeiro diferencial. Os valores máximos desta curva que ocorram até oito pontos antes dos valores máximos do primeiro diferencial são também assinalados. Estes valores correspondem à maior variação de gradiente, ou seja, o início da aceleração sistólica;

Embora não associado à determinação do início do ciclo cardíaco, é recomendado um último passo, corresponde à aplicação de um filtro passa baixo simples $(1, 2, 1)$ à curva de frequência máxima para reduzir os efeitos do ruído nos indicadores clínicos que dependem dos valores máximos e mínimos registados em cada ciclo.

Os parâmetros de aplicação do algoritmo FPS dependem da resolução temporal do método aplicado para a determinação da curva inicial, sendo aconselhável que o valor de N compreenda 40% a 90% do ciclo cardíaco e M inferior a 10%.

Os diferentes passos da aplicação deste algoritmo a uma curva de frequência média de um sinal de fluxo sanguíneo arterial são ilustrados na fig.2.5.

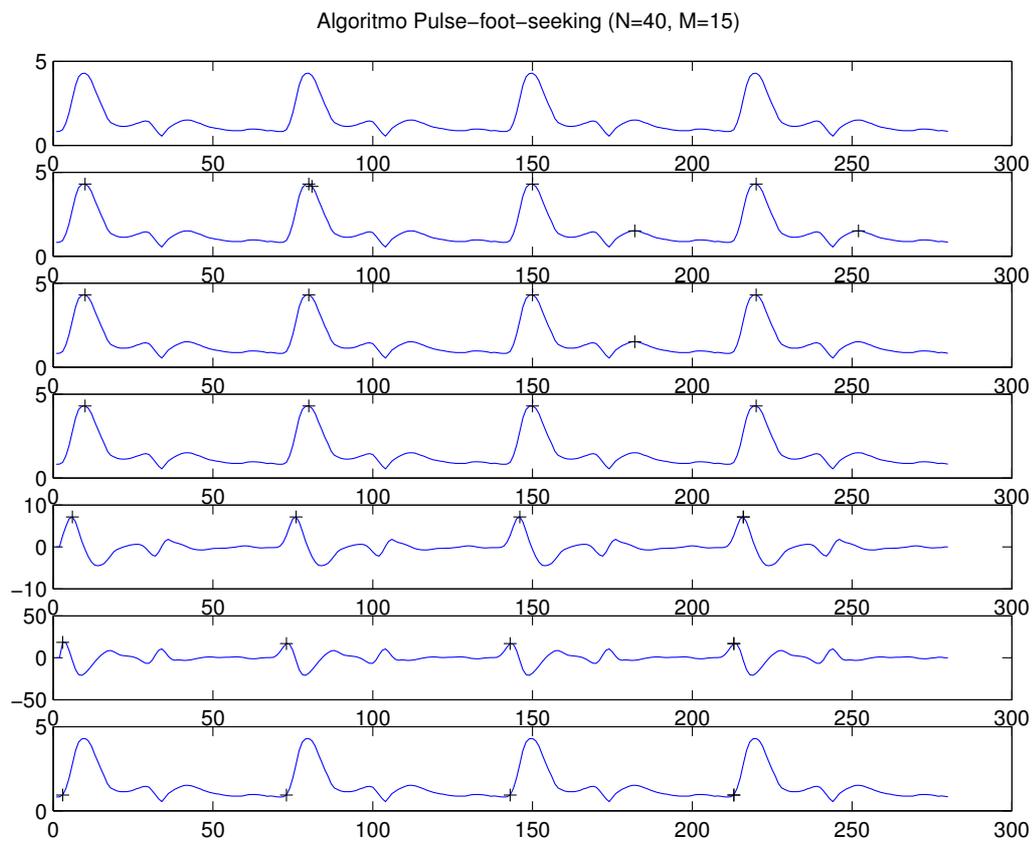


Figura 2.5: Segmentação por ciclos cardíacos de uma curva de frequência média usando o algoritmo FPS.

2.3.6.1 Freqüência Média

A curva característica mais comum é a da freqüência média que, ponderada pela intensidade do sinal em cada ponto, pode ser estimada como

$$\hat{f}_m(n) = \frac{\sum_{k=0}^{N-1} f_k \hat{P}(n,k)}{\sum_{k=0}^{N-1} \hat{P}(n,k)} \quad (2.43)$$

onde $\hat{P}(n,k)$ é a estimativa da potência espectral Doppler no instante temporal n à freqüência k e $f(k)$ a freqüência instantânea (Evans & McDicken, 2000).

2.3.6.2 Largura de Banda

A estimativa da largura de banda obtida com base na fórmula 2.43 é dada por

$$\hat{b} = \sqrt{\frac{\sum_{k=0}^{N-1} (\hat{f}_m(n) - f_k)^2 \hat{P}(n,k)}{\sum_{k=0}^{N-1} \hat{P}(n,k)}} \quad (2.44)$$

2.3.6.3 Freqüência Máxima

A freqüência máxima instantânea corresponde idealmente à maior velocidade presente num fluxo sanguíneo. No entanto, o alargamento espectral intrínseco e o ruído alteram os valores de freqüência máxima.

Vários métodos de estimar a freqüência máxima são descritos em (Evans & McDicken, 2000), cuja adequação pode variar com a razão sinal-ruído (*Signal-to-Noise Ratio* (SNR)), com o alargamento espectral intrínseco do método aplicado, como também da forma do espectro de potências Doppler. O método geométrico modificado (MGM) destacado por ser de uso geral e bastante robusto, foi escolhido. Este método foi proposto por (Moraes, Aydin & Evans, 1995).

Seja $\phi(f)$ o integral do espectro de potências Doppler estimadas no intervalo de freqüências $[f_L, f_H]$ definido por

$$\phi(f) = \int_{f_L}^{f_H} \hat{P}(f) df \quad (2.45)$$

e R uma linha que contém os valores máximo e mínimo de $\phi(f)$. A frequência máxima corresponde ao ponto de maior altura vertical entre as linhas $\phi(f)$ e R , conforme ilustrado na Figura 2.6.

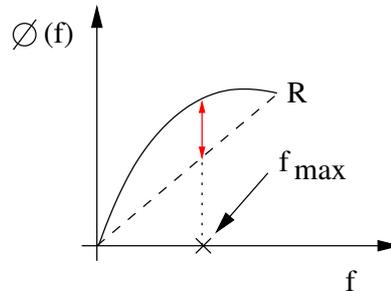


Figura 2.6: Determinação da frequência máxima pelo método geométrico modificado.

2.4 Sistemas Paralelos

O constante desejo de chegar mais longe e mais rápido expressa-se nos sistemas computacionais em solicitações para produzir mais resultados, mais depressa e com maior fiabilidade. As técnicas de processamento paralelo surgiram para servir estes objectivos utilizando vários processadores ou microcomputadores, do mesmo tipo ou de tipos diferentes, interligados entre si, que cooperam no suporte da carga computacional do sistema. Algumas dessas técnicas, anteriormente usadas apenas em processamento paralelo, são aplicadas a processadores de uso genérico e esbate-se a distinção entre sistemas paralelos e distribuídos.

Os modelos de comunicação inter-processador mais comuns são a passagem de mensagens e acesso a uma área de memória partilhada pelos vários processadores.

As abordagens mais comuns ao processamento paralelo são a programação concorrente, a adição de processadores ou micro-computadores – quer *transputers* quer processadores digitais de sinal (*Digital Signal Processors* (DSPs) – a uma máquina hospedeira, normalmente de uso genérico ou a utilização de placas com vários processadores idênticos – *Symmetric Multi-Processor* (SMP).

2.4.1 Processadores

A partir de meados da década de 80 regista-se um forte desenvolvimento das técnicas de processamento paralelo utilizando processadores específicos, de que

são exemplo os *transputers* e as DSPs. O baixo custo aliado à rápida evolução do mercado dos computadores pessoais favoreceu a utilização de processadores de uso genérico em sistemas distribuídos e baterias de processadores⁷. Quer se trate de processamento paralelo, distribuído ou embebido⁸, fabricantes de *hardware* e *software* apresentam um alargado leque de produtos que incluem a integração de processadores em placas desenhadas para aplicações de processamento de som, imagem, vídeo e, compiladores, linguagens de programação, ambientes integrados de desenvolvimento e *debuggers*.

Descrevem-se de seguida as características fundamentais dos processadores utilizados neste trabalho, sejam de uso genérico ou não.

2.4.1.1 *Transputers*

A palavra *transputer* surge da contracção de *transistor* e *computer*, traduzindo a ideia da miniaturização de um computador que se liga com os seus pares através de alguns fios eléctricos (Galletly, 1990).

O *transputer* utilizado foi o INMOS T805 (INMOS Limited, n.d.), um processador de 32 *bit*, com velocidade de relógio de 25 MHz, uma performance de 20 MIPS (*million instructions per second*), 4 kbytes de memória RAM e capacidade para 4,3 MFLOPS (*million of floating point operations per second*). Cada um destes *transputer* contém quatro canais de comunicação série para transmissão de dados entre processadores, cuja velocidade é de 20 Mbit/s. Cada porta suporta dois canais unidireccionais. A comunicação entre processadores é síncrona e faz-se a uma taxa de transmissão unidireccional de cerca de 1,7 Mbyte/s e bidireccional, a uma taxa de 2,3 Mbyte/s. As quatro portas de comunicação em série são independentes umas das outras e do processador, permitindo o envio de mensagens enquanto a unidade de processamento executa as instruções. Além disso, apresenta suporte *hardware* multitarefa, ou seja, existe um algoritmo de escalonamento microcodificado que dá ao processador a possibilidade de partilhar a execução de processos distintos, permitindo-lhe a atribuição de dois níveis de prioridade aos processos.

Os *transputers* não dispõem directamente de dispositivos que permitam a entrada e saída de dados, recorrendo para esse efeito ao computador hospedeiro.

⁷Em inglês, *cluster* significa grupo coeso foi aqui traduzido livremente por bateria de processadores

⁸Em inglês *embedded*

Numa rede de *transputers*, um deles, ligado à máquina hospedeira, designado por *root*, apresenta uma configuração tipo cliente-servidor. O *transputer root* (cliente) envia pedidos de transmissão de dados a um programa (servidor) em execução no computador hospedeiro.

Os *transputers* dispõem de uma linguagem de programação específica, o OCCAM, ou podendo ser programados em C ou Assembler.

2.4.1.2 Processador Digital de Sinal TMS320C4x

Os dispositivos da família TMS320C4x (Texas Instruments, 1991) são processadores digitais de sinal de vírgula flutuante de 32 *bit*. Os dispositivos desta família C4x integram com o processador central, uma unidade de vírgula flutuante, um controlador de acesso directo à memória e portas de comunicação.

A capacidade de processamento do TMS320C40 (C40) pode alcançar cerca de 30 MIPS, 60 MFLOPS; a sua taxa de transmissão máxima é de 384 *Mbyte/s*. O C40 tem 2 *kbyte* de memória RAM e *cache* de 128 palavras. Possui dois *buses* que permitem endereçamento de um espaço de memória único com 4 *gigaword*. O C40 pode tratar em cada ciclo de relógio mais de 4 palavras de 32 *bit* e transfere informação acedendo directamente à memória. Todos os *buses* internos podem transferir quatro palavras em paralelo, dispondo de sete fontes de memória de dados.

Este processador dispõe de 6 portas paralelas para a comunicação entre processadores. As comunicações são assíncronas e unidireccionais. A sua taxa de transmissão assíncrona atinge um máximo de 28 *Mbyte/s*.

Este tipo de processadores são incluídos num computador pessoal, conferindo-lhe as características de uma máquina paralela.

2.4.1.3 Processador Vectorial i860

O processamento vectorial utiliza técnicas de *pipeline* de forma a realizar operações sobre vectores de n elementos de forma mais eficiente que sobre n elementos escalares.

O processador i860 da Intel é um processador digital de sinal de vírgula flutuante de 64 *bit* (Kohn & Margulis, 1989). Este processador RISC, equipado com *caches* de dados (8 *Mbyte*) e de instruções (4 *Mbyte*), cuja frequência de relógio é 40 MHz pode atingir 40 MIPS e 80 MFLOPS. O processador utilizado neste trabalho está contido numa TTM110 i860 TRAM que inclui um *transputer* da

classe T8 a 25 MHz com 16 *Mbyte* de memória partilhada (Transtech Parallel Systems Ltd., 1992).

Este processador beneficia também da existência de um compilador dedicado *Portland Group i860 C compiler (pgcc)*. Este compilador, que implementa as directivas standard ANSI x2.159-1989, é também um otimizador, sendo capaz de efectuar optimizações locais, globais, de ciclos e outras, dependentes da arquitectura, como vectorização automática, *software pipelining*, e integração de procedimentos quando aplicáveis.

2.4.1.4 Processador Digital de Sinal SHARC ADSP20162

O processador da Analog Devices ADSP-20162 (Alex Parallel Computers, Inc., 1996) é um processador de 32 *bit* optimizado para aplicações de elevado desempenho em processamento digital de sinal, apresentando um pico de desempenho de 50 MIPS .

A configuração base forma um sistema completo, com memória SRAM *dual-ported* de 4 *Mbit* e periféricos de I/O integrados devido a um *bus* dedicado. Dispõe de mais 4 *buses* independentes para dados duplos, instruções, e entrada e saída de dados. Além disso tem ligações de *crossbar switch* para a memória. Este processador representa um novo padrão de integração para processadores digitais de sinal, apresentando as características de um DSP de vírgula flutuante e incluindo uma interface com o processador *host*, controlador, porta série, portas de comunicação e conexão em *bus* partilhado para multiprocessamento *glueless*.

O ADSP-2106x é caracterizado por 6 portas série síncronas que permitem fazer uma interface a uma variedade de periféricos digitais e de sinal misto. A sua taxa de transmissão máxima é de 40 *Mbyte/s*. A independência das funções de transmissão e recepção permitem uma grande flexibilidade para comunicações em rede.

Este tipo de processadores, como os *transputers* e processadores digitais de sinal, são adicionados a computadores pessoais conferindo-lhes as características de uma máquina paralela.

2.4.1.5 Processador Genérico Intel Pentium II

O Intel Pentium II é um processador de 32 *bit*, com 32 *kbyte* de *cache* L1, 521 *kbyte* de *cache* L2 *half speed*, *bus* duplo independente – *Dual Independent Bus* – e suporte *MMX* com *superpipelining* (Intel Corporation, 2001). A *cache* L1 um

bloco de memória interno ao microprocessador enquanto a *cache* L2 é um bloco de memória adicional de acesso mais rápido que a memória principal. O *bus* duplo independente permite que o processador aceda em simultâneo à memória principal e à *cache* L2. O suporte MMX refere-se a um conjunto de instruções multimédia suportadas directamente pelo processador, conferindo-lhe algumas das capacidades dos processadores digitais de som e imagem. *Superpipelining* refere-se à capacidade de execução concorrente de instruções. Estes processadores Pentium podem executar pelo menos seis instruções em simultâneo.

Dois destes processadores estão inseridos numa placa mãe *ASUS* constituindo uma máquina paralela (denominada *ticoteco*) sob o sistema operativo GNU Linux distribuído pela Debian.

O custo desta solução é relativamente baixo se comparado com os custos envolvidos na utilização de DSPs. A utilização genérica e alargada de processadores Intel propicia a existência de uma vasta gama de produtos de suporte ao desenvolvimento e *software* de aplicação que apoia a portabilidade de um produto final.

2.4.1.6 Outros Processadores Genéricos

Foram utilizadas alguns computadores pessoais equipados com processadores de propósito geral como hospedeiros das plataformas paralelas e para ensaios particulares. Destacam-se dois desses computadores, que serão directamente referidos:

arquimedes um Pentium MMX a 133 MHz equipado com 32 MB de RAM.

coral um Pentium III a 500 MHz equipado com 64 MB de RAM.

O sistema operativo usado nos dois computadores era também GNU Linux distribuído pela Debian.

2.4.2 Modelos de Comunicação

Nos sistemas paralelos a carga computacional é distribuída por vários processadores que cooperam entre si para efectuar determinada tarefa, concorrendo entre si na partilha de recursos comuns. A comunicação em sistemas paralelos serve dois propósitos fundamentais: a comunicação entre processos e a sincronização de processos. A comunicação entre processadores de um sistema paralelo é normalmente feita segundo um dos modelos:

Passagem de Mensagens Neste modelo, uma mensagem originada por um processador é enviada através da rede de interligação a um processador vizinho que a aceita e reage ao seu conteúdo, retransmitindo-a ou não. A carga adicional imposta pela gestão de comunicações pode condicionar o desempenho global do sistema.

Memória Partilhada O acesso a uma área de memória partilhada, seja ela física ou lógica, é normalmente a forma preferida de comunicação inter-processador. Em sistemas multiprocessador, a concorrência pelo acesso ao *bus* pode provocar uma diminuição no desempenho do sistema.

2.4.3 Abordagens à Programação Paralela

A utilização de vários processadores que cooperam entre si para efectuar determinada tarefa é normalmente feita seguindo uma das abordagens:

Programação Concorrente que pode ser considerada a forma abstracta do processamento paralelo. Servida por linguagens e/ou sistemas operativos vocacionados para o efeito, a programação concorrente realiza-se como processamento paralelo (caso existam mais do que um processador), ou como multitarefa.

Adição de Processadores normalmente de propósito específico, como é o caso dos processadores digitais de sinal, a um sistema hospedeiro de uso genérico, é uma opção muito frequente para acelerar determinados tipos de computação (como é exemplo a adição gráfica a um computador pessoal).

Multiprocessadores Simétricos são sistemas com vários processadores idênticos na placa principal (*motherboard*) do computador com acesso a uma área fisicamente una de memória partilhada. Esta forma de cooperação implica que qualquer dos processadores pode assumir qualquer processo.

2.5 Avaliação de Desempenho

A avaliação de desempenho está naturalmente associada às expectativas existentes, quer em relação à eficácia pretendida, quer em relação à eficiência desejada. No caso presente, num sistema analisador de fluxo sanguíneo, pretende-se que a

análise seja eficaz e que a eficiência de implementação dos métodos de análise adoptados convide à sua utilização.

Visando estas duas vertentes do desempenho, designa-se por desempenho estatístico a avaliação da qualidade com que um método de estimação espectral determina um ou mais parâmetros seleccionados do sinal Doppler pulsado; e designa-se por desempenho computacional a avaliação de um sistema, computador e programas, na produção de resultados respeitando requisitos frequentemente temporais.

2.5.1 Desempenho Estatístico

Considerando que o estimador ideal, óptimo para o caso em estudo, é aquele que produz resultados mais precisos, o desempenho estatístico pode ser encarado como o inverso do erro de estimação.

A determinação do erro de um estimador pressupõe a sua aplicação a um conjunto de sinais cujas características sejam ou possam ser conhecidas. Assim, o teste de um estimador pode ser efectuado com sinais simulados ou com sinais cujas características possam ser aferidas por outros métodos sendo possível determinar o desempenho estatístico desse método. O indicador do desempenho estatístico escolhido foi a raiz quadrada do erro quadrático médio por ser uma medida vulgarmente referida em comunicações científicas, favorecendo assim o termo de comparação com outros trabalhos.

No caso que motiva este trabalho, atendendo à utilização de mais do que um parâmetro espectral e ao objectivo de medir quão fiável é um método de estimação espectral, recorre-se a uma ponderação associada ao erro quadrático médio de cada parâmetro em estudo, ponderação essa representativa da importância desse parâmetro em particular na avaliação global do desempenho estatístico.

2.5.2 Desempenho Computacional

A definição de desempenho está fortemente associada aos requisitos da aplicação em causa. No que respeita às aplicações paralelas e em tempo-real, tradicionalmente, uma implementação era avaliada segundo os critérios tempo de execução, *ratio* R/C , aceleração (*speedup*), eficiência e escalabilidade.

Um critério menos frequente, usado neste trabalho, é o tempo médio por elemento que se designa por *gradiente* (g), $g = T(n_1)/n_1$. O gradiente pode ser

aplicado a tempos de execução ou de comunicação, sendo esperado que, num algoritmo regular, um aumento de carga expresso por n_1 implique um aumento proporcional nos tempos expressos por $T(n_1)$. Quando a proporcionalidade esperada se não verifique poder-se-á inferir que algo terá mudado na relação entre algoritmo e arquitectura. Essa mudança pode ser devida a vários factores, sendo os mais frequentes o esgotar de espaço de memória local para armazenamento de dados, passando a ser necessário o uso de memória virtual (“swapping”) ou a capacidade de armazenamento temporário (“buffering”) de uma ligação ter sido ultrapassada.

Em sistemas homogéneos, onde todos os processadores utilizados para processamento são idênticos, a análise dos valores obtidos destas métricas indica a forma como o algoritmo reage perante configurações alternativas, permitindo optar pela melhor solução, ou encaminhando à implementação de outro tipo de configurações.

Em sistemas heterogéneos são frequentemente aplicadas as métricas associadas ao estudo da escalabilidade de sistemas, nomeadamente *isoefficiência*, *isomemória*, *isotempo* e *isovelocidade*, que permitem não só avaliar quantitativamente a qualidade da implementação como também fornecem indicadores da adequação do algoritmo à plataforma de implementação e permitem prever a resposta do sistema ao aumento da dimensão do problema.

A qualificação tempo-real empregue em sentido lato indica que a taxa de produção do resultado do processamento respeita, em termos médios, um limite temporal considerado aceitável. A definição de tempo-real em sentido lato admite a existência de latência no sistema que se traduz num período de tempo em que existe entrada de dados e não existe produção de resultados. A qualificação de tempo-real em sentido estrito significa que o processamento jamais ultrapassa o limite temporal, definido como um requisito do sistema. Associado a esta definição surge frequentemente *data throughput*, a frequência de processamento de dados, que deve ser maior que a frequência de amostragem. A frequência de processamento de dados é o inverso do gradiente. Alguns indicadores são especialmente relevantes nos sistemas em tempo-real como seja a previsibilidade, fiabilidade e tolerância a falhas (Burns & Wellings, 1997).

Considerando que, para um sistema em tempo-real, o tempo de execução é talvez a faceta do sistema em relação à qual mais expectativas foram criadas, deverá, nesses casos, merecer uma atenção especial (Sahni & Thanvantri, 1996). O

impacto no tempo de execução de várias características físicas do sistema empregue propiciam a utilização de outras métricas. A propósito refira-se o recurso ao conceito máquina virtual (Tokhi, Ramos-Hernandez, Chambers & Hossain, 1997) permite a comparação de diversos sistemas reflectindo a adequação entre algoritmo e sistema empregue, mantendo a desejada abstracção das características físicas do sistema.

2.5.2.1 Métricas Genéricas

Numa aplicação de processamento de dados a métrica de desempenho mais adequada talvez seja *data throughput*, ou seja, o número de elementos de informação atómicos processados por unidade de tempo; da mesma forma, numa aplicação gráfica talvez seja o número de píxeis correctamente iluminados por segundo (Ruano & Madeira, 1999).

Outra possibilidade interessante é a da *capacidade de resposta*, traduzindo o intervalo de tempo entre o registo de um acontecimento e a resposta do sistema a esse acontecimento. Actualmente, muitas aplicações exigem sistemas embebidos, onde o processador é parte de um sistema maior. Geralmente o objectivo é responder em tempo-real a um acontecimento que ocorra quer no sistema maior quer no ambiente exterior. O significado de *tempo-real* neste caso pode ser encarado como a resposta a determinado acontecimento ocorrendo antes do próximo registo desse acontecimento, significando neste caso que a frequência de produção de resultados é equivalente (e não necessariamente superior) à frequência de aquisição de dados.

Uma terceira métrica de desempenho, *instruction throughput* (taxa de tratamento de instruções), muito utilizada comercialmente, apresenta pouco valor neste trabalho. *Instruction throughput* refere o número de instruções máquina atómicas que podem ser executadas por unidade de tempo. É necessário ter em mente que este tipo de métrica visa analisar o desempenho físico do sistema em condições muito particulares. Para que se possam retirar quaisquer considerações seria necessário analisar qual o teste de desempenho (*benchmark*) efectuado.

2.5.2.2 Tempo de Execução

O tempo de execução é calculado em função da dimensão do problema N , do número de processadores P envolvido, do número de tarefas U e de outras características do algoritmo e do hardware utilizado.

Este tempo de execução pode ser obtido de duas formas: como sendo a maior soma dos tempos de cálculo, comunicação e espera num determinado processador ou como sendo a soma de todos estes tempos em todos os elementos de processamento dividido pelo número de processadores N .

Designa-se por *tempo de cálculo* os períodos de tempo consumidos em cálculo, os quais dependem da dimensão do problema.

O *tempo de comunicação* consiste no período de tempo que as tarefas gastam no envio e recepção de mensagens. Existem dois tipos distintos de comunicação: a inter-processador e a intra-processador. No primeiro caso, as tarefas que comunicam encontram-se em unidades de processamento distintas, no segundo caso, ambas tarefas se encontram no mesmo processador.

Numa arquitectura ideal o tempo de comunicação inter-processador pode ser traduzido por dois parâmetros, o *tempo de arranque*, t_a , que corresponde ao tempo necessário para iniciar a comunicação, e o *tempo de transferência* de uma palavra, t_w . Desta forma o tempo consumido na transferência da mensagem, t_{msg} , pode exprimir-se por

$$t_{msg} = t_a + t_w L \quad (2.46)$$

sendo L o número de palavras contidas na mensagem.

Os tempos de cálculo e de comunicação são definidos explicitamente num algoritmo paralelo. Porém o período de espera (*idle*) não é fácil de determinar, uma vez que depende da ordem pela qual as operações são executadas. Um período de espera num processador pode dever-se à falta de condições para a execução de cálculos ou comunicação. No caso dos cálculos, esta falta de condições pode dever-se a uma má distribuição de carga pelos processadores, o que pode ser obviado pela aplicação de técnicas de distribuição de carga (*load-balancing*). No caso de espera devida a comunicação, a situação normalmente ocorre quando um processador remoto ainda não terminou as operações que lhe estavam destinadas, logo não pode estabelecer a comunicação. Em qualquer dos casos de falha de condições, a minimização do tempo de espera é procurada ocupando o processador com outro tipo de operações.

O tempo de execução nem sempre é a melhor forma de avaliar o que condiciona o desempenho dum sistema paralelo. Uma vez que este tempo é função da dimensão do problema, devem apresentar-se os tempos normalizados quando se efectuam comparações com outras implementações. Como factores de normalização pode-se referir a carga computacional, representada pela dimensão N do

problema, ou características físicas como a frequência do processador. A vantagem da métrica tempo de execução ocorre quando se pretende avaliar sistemas de tempo-real, onde, em geral, tempo de execução é um dos factores determinante do sucesso da implementação.

2.5.2.3 Aceleração

Outro indicador utilizado na avaliação dum sistema paralelo é o ganho obtido no desempenho de uma determinada paralelização face à implementação sequencial.

Aceleração (S) é definida como a razão entre o tempo de execução dum problema num único processador t_1 e o tempo necessário na resolução desse mesmo problema em p processadores idênticos, t_p :

$$S = \frac{t_1}{t_p} \quad (2.47)$$

Considerando a existência de várias implementações de um algoritmo sequencial, deve ser utilizado na comparação o algoritmo sequencial mais rápido de entre os conhecidos ou desenvolvidos. Idealmente, o valor da aceleração deve ser igual ao número de processadores utilizados na implementação. Na prática, o seu valor é inferior, pois os processadores passam por estados inactivos (*idle*) causados por conflitos de acesso a memória, atrasos de comunicações, ou ineficiência algorítmica.

2.5.2.4 Eficiência

Apenas um sistema ideal com p processadores pode ter um valor de aceleração igual a p . Na prática, o comportamento ideal nunca é atingido durante a execução de um algoritmo paralelo devido ao sistema não dedicar 100% do seu tempo aos cálculos. A medida de eficiência vem permitir caracterizar a forma como um algoritmo dispõe dos recursos computacionais, não sendo directamente dependente da dimensão do sistema.

Define-se eficiência como sendo a fracção de tempo que os processadores realizam trabalho útil, ou seja, o quociente entre aceleração e o número de processadores:

$$E = \frac{S}{p} \quad (2.48)$$

Assim, a eficiência terá valor unitário quando um sistema computacional com

p processadores for p vezes mais rápido. Se for algo mais lento, a eficiência será proporcionalmente menor.

A partilha de carga computacional exige cooperação o que, por sua vez, exige comunicação. Por vezes a comunicação num sistema paralelo é encarada como uma sobrecarga, uma vez que na maioria dos sistemas actualmente disponíveis, a velocidade de processamento é muito superior à taxa de transmissão suportada pelas ligações entre processadores. Mas apenas a comunicação que não é intrínseca ao problema poderá constituir sobrecarga. O requinte do desenho de um algoritmo paralelo consiste em reduzir a comunicação ao absolutamente essencial. Qualquer processador desenhado para constituir um elemento de processamento de um sistema paralelo deve suportar comunicação inter-processador, ou seja, suportar *links*. O conceito de eficiência exige a utilização plena de todos os recursos, quer de processamento quer de comunicação. O fulcro da questão reside na determinação do processamento e comunicações necessárias e conseguir um compromisso com as capacidades de processamento e de comunicação disponíveis. Este é o problema fundamental no desenho de um sistema paralelo.

2.5.2.5 Escalabilidade

Em sistemas paralelos, a capacidade de computação é associada ao número p de processadores utilizados. Um sistema diz-se escalar no intervalo compreendido entre p_1 e p_2 (quantidade mínima e máxima de processadores utilizados) se nesse intervalo a eficiência do sistema se mantiver constante.

$$E_p \rightarrow \text{const} \quad (p_1 < p < p_2) \quad (2.49)$$

Em suma, a escalabilidade (*scalability*) refere-se à propriedade de um sistema manter a eficiência quando o volume de computação aumenta.

Considerando um sistema com um processador assume-se uma eficiência de 100%, ou seja, a utilização de um único processador é máxima. Frequentemente a zona de interesse tem início imediatamente após $p = 1$.

A questão reside em saber quantos processadores poderão ser empregues antes que o sistema se torne ineficiente de forma inaceitável.

Existe uma causa principal para a dificuldade de obtenção na prática de um sistema escalar: o aumento do domínio implica um aumento das necessidades de comunicação de acordo com o algoritmo em causa. Com o crescimento do

número de processadores, a capacidade de comunicação aumenta de acordo com a forma como os processadores estão interligados. O problema de balancear necessidade e capacidade de comunicação com que anteriormente nos deparámos na maximização da eficiência é acrescido caso se pretenda que o sistema seja também escalar.

É de referir a existência de algumas formas padronizadas de tentar manter a escalabilidade dum sistema através do controlo de outras métricas usadas na avaliação do desempenho, como sejam isoeffiência, isomemória, isotempo e iso-velocidade.

2.5.3 Critério Custo/Benefício

A avaliação das implementações dos métodos de estimação de sinal de fluxo sanguíneo em tempo-real baseou-se na aplicação do critério Custo/Benefício à estimação da frequência média e largura de banda, na sua forma inicialmente proposta (Ruano & Fish, 1992). Este critério pondera o custo computacional (função da complexidade algorítmica) de um método face ao benefício estatístico (função inversa da raiz quadrada do erro quadrático médio) permitindo quantificar cada solução composta por método/algoritmo, pela função

$$c(i) = \frac{\text{custo}(O_{(n)})}{\text{benefício}(W_b, \text{rmse}_{f_m}, \text{rmse}_{b_w})} \quad (2.50)$$

onde $i = 1, \dots, M$ representa o método analisado, $O_{(n)}$ é a complexidade algorítmica, rmse_{f_m} e rmse_{b_w} são as raízes quadradas dos erros quadráticos médios obtidos na estimação de, respectivamente, frequência média e largura de banda, e W_b um vector de factores de ponderação dos parâmetros de benefício (cuja soma, por convenção, deve totalizar 1).

A função benefício é dada por

$$\begin{aligned} \text{benefício}(W_b, \text{rmse}_{f_m}, \text{rmse}_{b_w}) &= \\ &= (w_{f_m} \text{rmse}_{f_m} + w_{b_w} \text{rmse}_{b_w})^{-1} \end{aligned} \quad (2.51)$$

A raiz quadrada do erro médio quadrático (rmse) é obtida por

$$\text{rmse} = \sqrt{\text{viés}^2 + \text{variância}} \quad (2.52)$$

onde

$$viés = \frac{1}{m} \sum_{n=0}^{m-1} (E[\hat{p}a(n) - pa(n)]) \quad (2.53)$$

$$variância = \frac{1}{m} \sum_{n=0}^{m-1} (E([\hat{p}a(n) - \bar{p}a(n)]^2)) \quad (2.54)$$

sendo $E[\cdot]$ o operador valor esperado, $pa(n)$ o real valor do parâmetro (neste caso, normalmente frequência média ou largura de banda) em estudo no instante de tempo n , $\hat{p}a(n)$ o valor estimado do parâmetro no mesmo instante de tempo, $\bar{p}a(n)$ o valor médio, nesse instante de tempo, do parâmetro estimado ao longo de vários ciclos e m o número de espectros calculados em cada ciclo cardíaco.

Face à diversidade de processadores, de velocidades e arquitecturas distintas, o sucesso de uma implementação depende não só do algoritmo utilizado mas também da sua adequação à arquitectura (Tokhi & Hossain, 1995). Inúmeras métricas podem ser aplicadas na avaliação de desempenho de uma implementação paralela e, conseqüentemente, como medida de custo computacional. Considerando que o objectivo último é obter uma implementação em tempo-real, o tempo efectivamente decorrido na aplicação de um método pode ser utilizado como medida de custo computacional. Este tema será retomado no capítulo 7.

2.6 Aquisição e Visualização de Dados

A aquisição de dados de um equipamento clínico de um equipamento de ultra-som Doppler não é um processo suficientemente documentado. À semelhança do que acontece na instrumentação ultra-sónica existente no Serviço de Cardiologia do Hospital Distrital de Faro, a maioria dos instrumentos ultra-sónicos só permitem a recolha de dados numa forma já processada, quer seja audível – gravação de fita do sonograma –, ou espectrogramas em formato gráfico.

O diagnóstico clínico assenta na análise do espectrograma e parâmetros considerados relevantes para o caso em estudo.

Os equipamentos clínicos de ultra-som Doppler, para além de apresentarem o espectrograma numa escala de cores *standard* ou definida pelo clínico e fornecerem alguns dos parâmetros relevantes, permitem ainda alguns procedimentos visando a determinação não automática de outros dos parâmetros. Esses procedimentos podem incluir a revisão ou paragem de imagens bem como a exportação de imagens para outros dispositivos onde serão sujeitas a tratamento posterior.

A título ilustrativo, apresenta-se a Figura 2.7 recolhida durante um exame da artéria carótida, resultado da impressão, a preto e branco, de uma imagem visualizada a cores no monitor do ecógrafo. À esquerda, é visualizada a bifurcação da artéria carótida primitiva nas artérias carótidas interna e externa; à direita, é apresentado o espectrograma da artéria carótida externa juntamente com alguns indicadores.

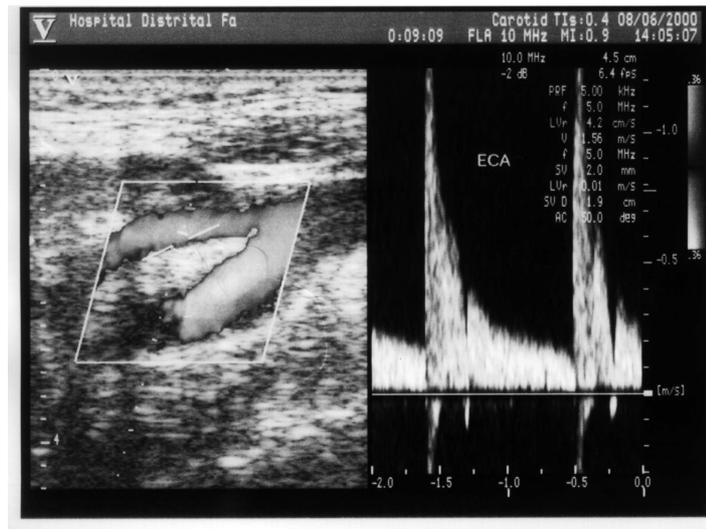


Figura 2.7: Imagem recolhida durante um exame à artéria carótida.

Assim, para além da apresentação de espectros, será necessário contemplar a interface gráfica com o utilizador (GUI⁹).

2.7 Conclusão

A título introdutório foi apresentado o enquadramento do processamento em tempo-real de sinais de fluxo sanguíneo obtidos por ultra-som Doppler em arquiteturas de processamento paralelo.

Foram revistos alguns princípios dos sistemas clínicos de diagnóstico de fluxo sanguíneo por ultra-som Doppler contínuo e pulsado bem como os métodos de estimação espectral a aplicar para a extracção de indicadores que propiciem o diagnóstico.

Para a utilização efectiva desses métodos de estimação espectral é necessária

⁹do inglês, *Graphical User Interface*

a a sua implementação em tempo-real. Acresce ainda que a obtenção de informação relevante para o diagnóstico clínico em tempo útil exige uma capacidade de computação significativa, sendo desejável que a determinação das curvas características de fluxo sanguíneo relevantes para a qualificação do estado clínico possa também ser obtida em tempo-real.

Sendo proposta, desde início, a utilização de técnicas de processamento paralelo para atingir o objectivo tempo-real foi apresentada uma visão geral dos sistemas paralelos e métricas de avaliação de desempenho dos mesmos sistemas. O critério de Custo/Benefício, a usar na determinação da melhor implementação, foi alvo de uma descrição pormenorizada.

Assim, e numa fase anterior ao estudo da implementação em tempo-real dos métodos de estimação, procurou-se articular o objectivo global do trabalho com as diferentes possibilidades de implementação, concebendo-se um sistema tempo-real de processamento de sinais de fluxo sanguíneo, tema do próximo capítulo.

Capítulo 3

Planeamento do Sistema de Processamento

3.1 Introdução

Tendo sido apresentado o enquadramento deste estudo, este capítulo descreve a concepção e planeamento de um sistema de processamento de sinais biomédicos. Considerando as necessidades dos utilizadores finais, médicos e investigadores de processamento de sinal, pretende-se um sistema de processamento de sinais que seja um sistema aberto, que possa contemplar o tratamento de diferentes sinais e evoluir para um sistema de suporte à investigação em processamento de sinais biomédicos.

A análise e definição de requisitos foi e é dificultada pelo facto de se tratar de um sistema novo. Optou-se por esboçar o documento de requisitos de software, identificando os subsistemas principais e, pela utilização de prototipagem evolutiva (Ruano, 2000; Somerville, 1992), desenvolver cada um deles para posterior integração. A adopção desta estratégia de planeamento e desenvolvimento favorece a produção de um sistema que se pretende flexível, aberto, modular, portátil e de baixo custo.

A dimensão do sistema justifica e encoraja a utilização híbrida das diferentes técnicas de prototipagem, nomeadamente a utilização de linguagens de muito alto nível, a criação de componentes reutilizáveis bem como desenvolvimento por reutilização, e prototipagem da interface com o utilizador.

A descrição do sistema que aqui se apresenta é já o resultado de várias iterações de prototipagem e refinamento de especificações, após teste em situações diversas

(Ruano, Moura, Leiria, Garcia & Moreno, 2003).

Inicialmente concebido para a análise de fluxo sanguíneo em artéria carótida primitiva, a multiplicidade dos casos considerados (vasos sanguíneos e anomalias detectáveis) exige uma visão global numa perspectiva funcional. No capítulo 8 apresenta-se de forma sumária a articulação entre o contexto da aplicação e do desenho do sistema em função do alargamento das especificações.

3.2 Descrição do Contexto da Aplicação

A parte fundamental do sistema de processamento de sinais de fluxo sanguíneo obtidos por equipamentos de ultra-som Doppler pulsado é realizado pela implementação, em tempo-real, de um método de estimação espectral seleccionado. Essa implementação, para ser eficaz, deve ser feita numa arquitectura adequada ao algoritmo do estimador seleccionado e que tenha capacidade computacional para cumprir os requisitos de tempo-real.

Considerando um sistema completo, para além da estimação do espectro, é necessário: incluir indicadores clínicos para a análise do fluxo sanguíneo; contemplar a aquisição de dados para a arquitectura de processamento escolhida e prever as necessidades de apresentação gráfica de resultados.

Embora este estudo deva considerar primordialmente o processamento de sinais de fluxo sanguíneo da artéria carótida primitiva, o mesmo sistema poderá contemplar outros sinais de fluxo sanguíneo se for desenhado de forma modular e aberta ou, talvez, os métodos de processamento de sinal aplicados possam ser adaptados ou substituídos para estimação de outros sinais biomédicos.

A análise preliminar do sistema a desenvolver revelou distintas utilizações do sistema de processamento de sinais biomédicos, ainda que restringindo a sua aplicação a sinais Doppler de ultra-som pulsado da artéria carótida primitiva.

3.2.1 Caracterização de Fluxo Sanguíneo

O método convencional de processamento de sinais de fluxo sanguíneo baseado na transformada de Fourier é utilizado na instrumentação ecográfica de ultra-som Doppler pulsado. Vários estudos indiciam que outros métodos permitem melhorar a qualidade de determinação dos parâmetros clínicos sobre os quais se baseia o diagnóstico. Ora esse diagnóstico é conseguido através de um exame cujo desenrolar depende fortemente dos indicadores que vão sendo recolhidos. Assim,

para que esses outros métodos possam ser usados é necessário que a produção dos indicadores seja feita em tempo-real.

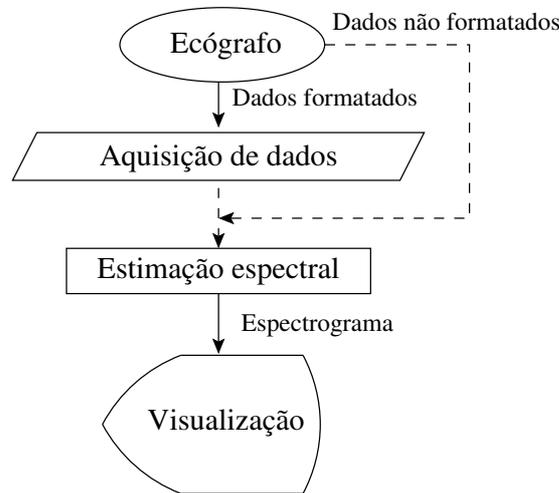


Figura 3.1: Sistema de caracterização de fluxo sanguíneo

A Figura 3.1 representa as componentes fundamentais do sistema de caracterização de fluxo sanguíneo baseado em sinais de ultra-som Doppler. O tipo de sinal a utilizar depende do ecógrafo. Caso seja possível, dever-se-ão utilizar dados *raw*, isto é, dados não tratados, representativos do desvio em frequência – Doppler *shift* – impresso ao sinal pelo movimento do sangue. A aquisição de dados visa reconstituir o sinal temporal do desvio em frequência caso os dados disponibilizados tenham sido sujeitos a tratamento prévio ou a uma acção de formatação. O processo de estimação espectral produz, a partir do sinal temporal, espectros de potências Doppler. Dependendo do método de estimação utilizado, em fase anterior à apresentação do espectrograma, poderá ser necessário, ou não, proceder ao cálculo do espectro médio de um número variável de espectros (Guo, Durand & Lee, 1994a). A visualização dos espectrogramas é acompanhada da apresentação de alguns indicadores como a frequência média e a largura de banda do espectro.

3.2.2 Validação e Ensaio Clínico

A validação de um método de determinação dos parâmetros clínicos sobre os quais se baseia o diagnóstico é necessariamente uma tarefa que exige competência clínica porque não é possível comparar os resultados numéricos e certificar a qualidade das medidas obtidas.

Outro factor crítico do sucesso da adopção de um novo método de determinação de parâmetros clínicos reside na eventual necessidade de considerar alterações nos valores limites que definem condições anómalas ou patológicas. É necessário permitir o ensaio do sistema, pela comparação de resultados obtidos com os dois métodos.

Um sistema computacional de processamento de sinais biomédicos para permitir diminuir o intervalo de tempo de implementação do resultado de investigações na área e a utilização clínica desses resultados deve:

- fomentar a confiança clínica pelo envolvimento de médicos na investigação de métodos de processamento de sinais biomédicos e pela utilização paralela dos métodos em causa;
- explorar a conveniência de utilização, associada ao tempo e procedimentos necessários à produção de resultados.

A Figura 3.2 ilustra como o sistema de processamento de sinais poderá ser utilizado em ensaios clínicos, sendo o S o método *standard* baseado na transformada de Fourier e o E o método em estudo. O processamento do sinal temporal

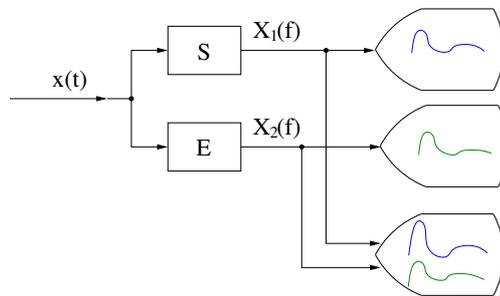


Figura 3.2: Representação do sistema para ensaios clínicos.

$x(t)$ em paralelo por ambos os métodos permite a obtenção de espectros apresentados separadamente ou em conjunto, fornecendo a base para a validação dos métodos e o ensaio do sistema em ambiente clínico ou hospitalar, numa fase posterior ao exame do paciente. Desta forma, visa-se contribuir para a mudança do procedimento usado durante o exame conducente ao diagnóstico.

3.2.3 Apoio à Investigação e Desenvolvimento

A metodologia de estudo de alternativas ao processamento de sinais faz-se normalmente usando, numa fase inicial, sistemas de simulação. No caso de sinais de fluxo

sanguíneo esses simuladores podem ser modelos físicos ou métodos numéricos. Em qualquer dos casos pretende-se uma avaliação quantitativa de acuidade dos métodos de processamento, avaliação essa determinante da escolha do método a empregar. Os métodos numéricos, de menor custo e passíveis de implementação computacional, são vulgarmente preferidos. No caso presente a escolha é motivada pelo facto de a investigação de métodos numéricos de simulação e estimação espectral de sinais de fluxo sanguíneo ser uma das áreas de trabalho desenvolvidas no laboratório onde este trabalho foi executado.

A Figura 3.3 ilustra um sistema para estudo da implementação de métodos de estimação de sinais de fluxo sanguíneo, centrado no sistema de processamento.

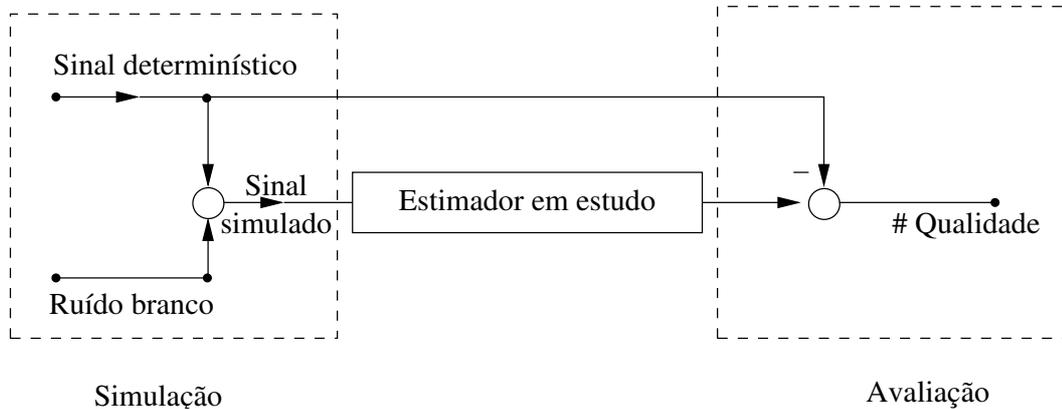


Figura 3.3: Representação do sistema para estudo da implementação de métodos de processamento de sinais biomédicos.

O núcleo de investigação em processamento de sinal do laboratório onde este trabalho foi desenvolvido recorre a vários métodos para a criação de sinais simulados. O método numérico de simulação mais utilizado, e que se pretende integrar com o sistema de processamento de sinais biomédicos, modela um sinal de ruído branco, através de filtros construídos com base no sinal determinístico, dando origem a sinais simulados. É com base nas características determinísticas destes sinais simulados que se procede à comparação de diferentes métodos de estimação. Cumpre saber qual dos métodos melhor identifica ou reconstitui as características dos sinais teóricos, tarefa desempenhada na avaliação. Normalmente apoiada no critério Custo/Benefício, a determinação da qualidade da estimação é função inversa do erro produzido.

A inclusão de um sistema de simulação e avaliação da qualidade dos métodos

empregues poderia também contribuir para fortalecer a confiança clínica por facilitar a intervenção do médicos na investigação desenvolvida e integrar também o sistema de suporte a estudos clínicos. O estudo da aplicabilidade de um método é normalmente um processo longo e iterativo, sendo frequentemente necessária a afinação de parâmetros, que pode ser significativamente encurtado por um sistema de avaliação automática.

Alguns estudos clínicos visam a determinação de causalidade ou interdependência de sintomatologia. O sistema de simulação e avaliação poderia ser gradualmente suprido com as ferramentas necessárias a alguns estudos. Como exemplo cite-se a correlação entre sinais ou indicadores ou a determinação automática de curvas teóricas para a simulação de diferentes casos clínicos de distintos sinais biomédicos.

Assim, para apoiar a investigação e desenvolvimento em processamento de sinal, pretende-se, numa primeira fase, integrar o sistema de simulação e avaliação, podendo, numa fase posterior, ser incluído no sistema final.

3.3 Plataforma de Hardware

A escolha do sistema computacional bem como o sistema executivo ou operativo a utilizar condicionam o sistema a implementar. Essa escolha, que é normalmente uma das fases iniciais da definição de requisitos de um sistema, é um dos objectivos deste estudo.

Os processadores disponíveis para testes e eventuais candidatos a integrarem o sistema de processamento em tempo-real são *transputers*, C40, Sharcs, um sistema com dois processadores Intel Pentium II, um PowerPC G3, um Pentium III ou, quando tecnologicamente possível, combinações dos anteriores.

A divisão das tarefas pelo *hardware* disponível será determinada pelo desempenho obtido nos diferentes processadores. A adição de processadores ou a utilização do sistema com dois processadores surgirá da necessidade de maior capacidade de processamento.

3.4 Modelo Conceptual

São quatro os subsistemas identificados – aquisição de dados, estimação espectral, visualização e avaliação de qualidade – cuja natureza varia consoante a finalidade

pretendida: utilização em tempo-real, investigação em processamento de sinais, ferramenta complementar de diagnóstico e ensaios clínicos.

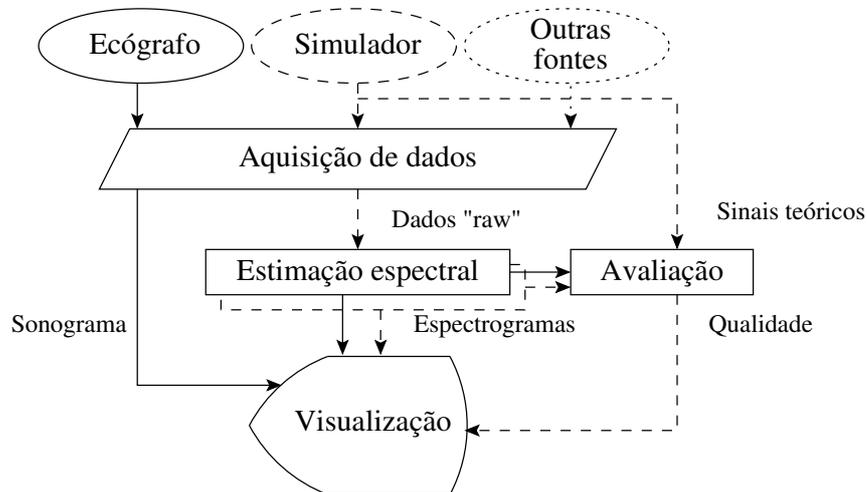


Figura 3.4: Principais blocos do sistema computacional

3.4.1 Aquisição de Dados

Este subsistema contempla três origens de dados: directamente provenientes do equipamento ecográfico, recolhidos utilizando equipamentos auxiliares e do simulador.

Para a ferramenta complementar de diagnóstico seria desejável que a aplicação de métodos de processamento de sinal fosse feita a dados não processados. No entanto, a diversidade de sistemas proprietários dificulta esta tarefa. Alguns sistemas ecográficos disponibilizam os dados em formato digital e existem sondas que, através da utilização de cartas de conversão de sinais analógico/digital, podem ser ligadas a um computador.

Noutros sistemas os dados podem ser gravados num equipamento auxiliar áudio ou vídeo, de forma analógica. Por vezes, disponibilizam ainda as imagens pós-processadas sobre as quais é possível aplicar procedimentos automáticos de extracção de indicadores. Para estes sistemas é necessário considerar um sistema de conversão de dados que deverá ser definido caso a caso.

Os dados simulados protagonizam a situação mais simples, encontram-se numa sequência de ficheiros, gravados em formato ASCII por conveniência de utilização em outros sistemas. Surge portanto a necessidade de conversão para um formato

binário de forma a diminuir o tempo de leitura dos dados no subsistema de processamento.

Dados de outras fontes poderão ser considerados deste que seja criado uma interface com o sistema de processamento.

3.4.2 Estimação Espectral

Tal como se verifica em outras áreas de aplicação de processamento digital de sinal, a referência na estimação espectral para os sinais de ultra-som Doppler pulsado de fluxo sanguíneo baseia-se na transformada de Fourier. Este é um dos estimadores a considerar no sistema. Nesta fase de desenvolvimento do sistema, devem-se ainda considerar os estimadores espectrais AR-MC e CWD.

3.4.3 Visualização

O resultado da aplicação dos métodos de estimação espectral é a densidade espectral de potência do sinal Doppler, vulgarmente designada por espectro, sendo a apresentação mais directa desse resultado feita sob a forma de espectrograma colorido.

Como ferramenta complementar de diagnóstico, a visualização de informação deve ser feita de forma a propiciar a decisão, dando maior relevo ao que é mais determinante em cada caso, como é exemplo a determinação quantitativa de alguns indicadores clínicos de diagnóstico. Tendo em mente que os indicadores clínicos podem variar não só em função do objecto de análise como também da patologia a investigar, deverá ser possível ir integrando e validando alguns mecanismos de informação para diagnóstico.

O estudo de casos e a validação clínica de métodos exigem a visualização, simultânea ou alternada, do resultado da aplicação de diferentes métodos.

Na investigação em processamento de sinais, é relevante a qualidade da estimação espectral sendo interessante comparar indicadores de desempenho estatístico.

Em resumo, perspectivam-se diferentes tipos de sistemas de visualização:

- apresentação simples de espectrograma(s);
- apresentação de espectrograma e indicadores definidos pelo utilizador;

- apresentação da qualidade dos métodos empregues na fase de estudo e adaptação do sistema existente à sua aplicação real.

3.4.4 Avaliação de Qualidade

A avaliação da qualidade está fortemente associada aos métodos em causa e aos objectivos/expectativas dos utilizadores. Caso o sistema venha a ser usado em ambiente clínico deve fazer-se inicialmente uma avaliação da capacidade de produção de informação complementar de apoio ao diagnóstico clínico. Contudo, não sendo tal do âmbito deste trabalho, a atenção será focada em duas vertentes, a estatística e a computacional. O critério Custo/Benefício (cf. secção 2.5.3) é considerado um bom ponto de partida na avaliação da implementação sem prejuízo de ser complementado com outras métricas. Como grande vantagem realce-se o facto de permitir quantificar o benefício estatístico obtido pela aplicação de um estimador e o custo da implementação, servindo assim as duas principais vertentes da avaliação de qualidade.

Considerando a tradicional utilização de métodos baseados na transformada de Fourier nos equipamentos comerciais, a FFT pode ser encarada como uma referência na escolha de um método de estimação.

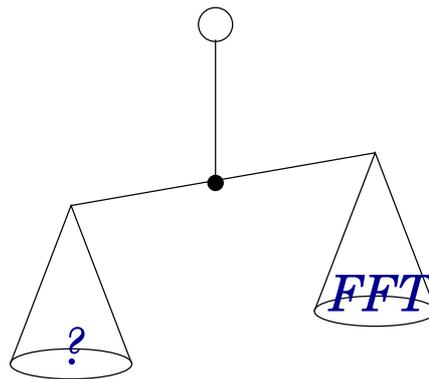


Figura 3.5: Custo/Benefício em relação à FFT.

Outros factores a considerar do ponto de vista computacional podem incluir o tempo de execução ou o número mínimo de elementos de processamento para uma execução em tempo-real.

3.5 Requisitos Funcionais e Não Funcionais

Considerem-se separadamente os quatro subsistemas Aquisição de Dados, Estimacção Espectral, Visualização e Avaliação.

Módulos alternativos integrarão cada um dos subsistemas, de forma a contemplar as diversas utilizações pretendidas, conforme representado de forma simplificada na Figura 3.6.

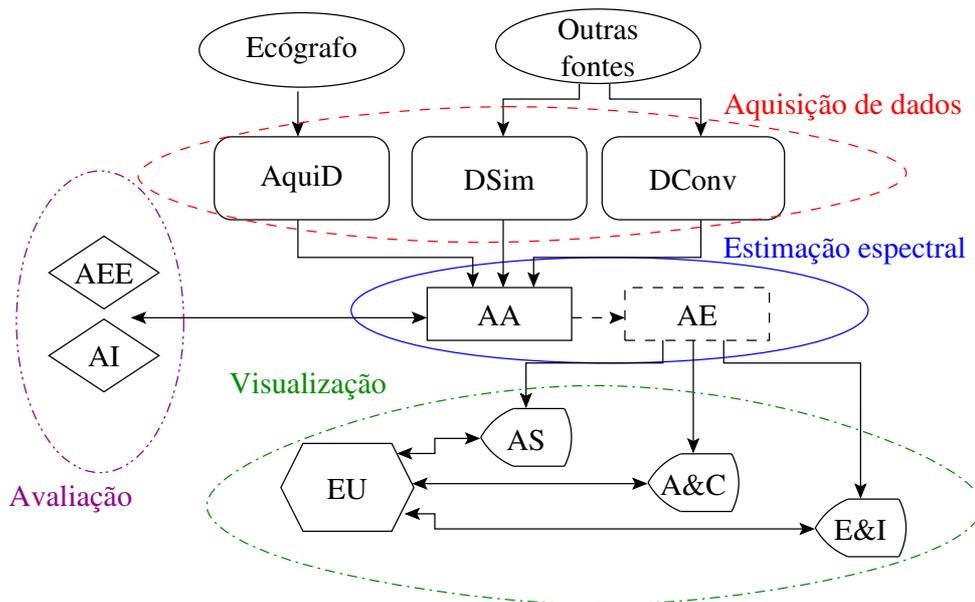


Figura 3.6: Representação gráfica dos principais blocos do sistema analisador de fluxo sanguíneo

Aquisição de Dados – O sinal de fluxo sanguíneo é lido para o sistema. Será usado um dos módulos:

AquiD - Aquisição Directa - A aquisição de dados é feita a partir de um ecógrafo ou transdutor de ultra-som. Este módulo varia consoante o equipamento utilizado. O objectivo é recolher um fluxo de dados reais, normalmente sujeitos a algum processamento analógico que se pretende mínimo.

DSim - Dados Simulados - A simulação dos dados é feita por um sistema externo, sendo prevista a necessidade de uma conversão simples do formato dos dados. Um sinal simulado, correspondente a um intervalo de tempo será segmentado, originando um conjunto de ficheiros, cujo

nome será composto por uma raiz comum e um número sequencial, cada um contendo apenas um ciclo cardíaco. É também necessário manter informação que permita identificar a simulação em causa tal como vaso ou válvula, nível de ruído ou frequência de amostragem.

DConv - Dados Convertidos - A conversão de dados obtidos por outros sistemas. Este módulo pode contemplar a conversão de registos não magnéticos. Caso seja possível a obtenção de dados "puros", estes são armazenados de acordo com o standard. Caso a informação se traduza num espectro de sinal cardíaco será armazenado como tal para posterior análise e comparação. Também neste caso, manter registo das características dos dados.

De forma a garantir a independência da fonte de dados a processar, a cada uma das formas de aquisição de dados surge associado um módulo de armazenamento. Os dados são convertidos para representação binária.

Estimação espectral – Cada um dos ficheiros é transformado, pela aplicação de um método de estimação espectral, no espectrograma. Naturalmente, em casos de aplicação em paralelo de dois métodos de estimação, serão produzidos dois espectrogramas. Para que a estimação seja obtida em tempo-real o processamento associado a cada estimador não poderá ser superior ao tempo de amostragem.

AA - Algoritmo-Arquitectura - Serão considerados módulos alternativos que reúnem o par algoritmo / arquitectura. A arquitectura Intel e o algoritmo sequencial constituem o ponto de partida. Sucessivas simplificações algorítmicas e diferentes arquitecturas serão testadas com o objectivo de determinar o par que produz os melhores resultados.

AE - Armazenamento de espectros - podendo variar de acordo com as necessidades do sistema de visualização, este módulo prevê a conversão do espectro de um formato numérico para gráfico ou a criação do espectro médio sobre o qual serão extraídos os indicadores.

Visualização – A visualização dispõe os espectros atrás referidos (AE) e extrai indicadores por pós-processamento da imagem.

EU - Especificação do Utilizador - Conformidade e satisfação de requisitos

específicos dos utilizadores, como o código de cores utilizado e que indicadores devem ser incluídos em cada caso.

AS - Apresentação simples - A visualização simples do espectrograma. Este será talvez o módulo mais usado num sistema em tempo-real.

A&C - Análise e comparação - Análise e comparação de espectros. A utilização específica deste módulo destina-se à verificação do sistema face a outros em utilização corrente e face a resultados obtidos de sinais simulados.

E&I - Espectros e indicadores clínicos - O módulo para exploração médica do sistema. Para além de medições específicas deve permitir a análise em diferido do sinal cardíaco, a comparação com sinais históricos do paciente ou outros julgados padrão. Deve ainda permitir a configuração dinâmica de forma a incluir os indicadores mais pertinentes.

Avaliação – Este subsistema fornecerá indicadores de qualidade, conforme o objectivo de utilização do sistema global. Apresenta-se na Figura 3.6 associado apenas ao sistema em tempo-real embora se relacione também com os demais subsistemas.

AI - Avaliação da implementação - considerando o par algoritmo / arquitectura e a aplicação de técnicas de processamento paralelo recorrendo para tanto às métricas de desempenho computacional como tempo de execução, aceleração, eficiência ou razão execução-comunicação.

AEE - Avaliação da estimação espectral - considerando o resultado da estimação quer comparando espectros quer avaliando a recuperação das características originais de sinais simulados, recorre a indicadores estatísticos como erro quadrático médio, variância ou viés.

3.6 Arquitecturas de Implementação

O ecógrafo existente no Serviço de Cardiologia do Hospital Distrital de Faro, onde se pretende que o sistema seja, em primeira instância, testado e validado, está ligado a um computador Apple Macintosh a correr o sistema operativo Mac OS 9. Outros computadores existentes correm sistemas operativos da Microsoft.

Seria conveniente que a utilização do sistema de processamento de sinais não exigisse formação na exploração de um novo sistema operativo e pudesse ser instalado num dos equipamentos que integram o existente parque informático. Seria também vantajoso evitar custos adicionais, devendo ser limitados a processadores que se revelem necessários e, eventualmente, equipamentos para a transmissão de dados entre equipamentos. Custos com *software*, seja este sistema operativo, ferramenta de desenvolvimento ou outras devem ser minimizados.

3.7 Considerações sobre as Linguagens de Implementação

Para o desenvolvimento do sistema, e não estando definido qual o equipamento que suportará a fase de exploração, a escolha das linguagens e aplicações para a criação de ambientes gráficos para interface com o utilizador foi norteadada pelos critérios de portabilidade, uso livre e acesso ao código fonte, robustez e permanência.

Os compiladores existentes para o *transputer* e o processador digital de sinal C40 favorecem a escolha da linguagem C, a linguagem de mais alto nível suportada por compiladores nas diferentes plataformas de implementação.

Para a articulação da chamada aos programas elementares, e seguindo a tendência referida em (Ousterhout, 1998), procurou-se uma linguagem de escrita¹, interpretada, e leve, sendo escolhida a linguagem *Python*. *Python* é uma linguagem de alto nível, é orientada ao objecto, e muito utilizada para a prototipagem e desenvolvimento rápido de aplicações, podendo ser estendida pela criação de módulos compilados em outras linguagens que podem ser chamados como módulos a partir de *Python*. Desta forma, é normal e frequente verificar-se que uma determinada biblioteca ou aplicação está envolvida em *Python*, isto é, pode ser invocada a partir de um programa em *Python*. É de referir a este propósito a existência de *swig*, um compilador de interfaces entre *C/C++* e linguagens de escrita como *perl*, *Python* e *Tcl/Tk*. O funcionamento do *swig* baseia-se em retirar as declarações comumente encontradas nos ficheiros de cabeçalho de linguagens como *C* ou *C++* e gerar o código envolvente necessário às linguagens de escrita para a chamada e utilização dessas funções.

¹tradução livre de *scripting language*

Para a criação de interfaces considerou-se Tk e wxWindows, embora muitas outras alternativas existam (Middleton & Kulakov, 1999). O sistema de interface com o utilizador tradicionalmente usado com *Python* está disponível com a inclusão do módulo *Tkinter*. Este módulo envolve chamadas a *Tcl/Tk* fornecendo um conjunto de ferramentas para gestão de janelas de forma robusta e independente da plataforma. De referir ainda o módulo *Pmw* – Python Mega Widgets – que facilita o rápido desenvolvimento de aplicações e protótipos. *wxWindows* é uma biblioteca de classes em *C++* para o desenvolvimento de GUIs, cuja portabilidade se deve à criação de um API², independente da plataforma. Isto é conseguido pelo recurso a bibliotecas gráficas nativas de forma que as aplicações aparentam ter sido desenvolvidas de raiz nessa plataforma. Esta ferramenta está envolvida em *Python* no módulo *wxPython*.

A escolha de uma das duas opções é, na maioria das vezes, uma questão de gosto.

Conforme referido em (Buzo, Zoppi & Fabris, 2002), as principais vantagens do *Tk* é ser muito estável, sendo utilizada há já bastante tempo, existindo nas versões para sistemas operativos da Microsoft, para X Window, MacOS e BeOS; ser de fácil aprendizagem, registando-se alguns casos de utilização como linguagem de escrita; ser software livre para todos os usos e, inclusive, encontrar-se incluída na distribuição oficial de *Python*. As principais desvantagens do *Tk* prendem-se com o facto de não ter um aspecto muito moderno e um desempenho comparativamente inferior.

No que diz respeito a *wxPython*, as principais vantagens são um aspecto moderno, incluindo *widgets* para muitas actividades comuns. Como desvantagens cite-se o facto de ter pouca documentação específica e a versão para Mac OS estar, à data do início do desenvolvimento dos trabalhos, num estado mais incipiente do que as versões para as demais plataformas. Optou-se assim pelo uso de *wxPython*.

Foram comparados os tempos de execução entre módulos criados apenas com código *C* puro e os envolvidos em Python e não se registaram diferenças significativas. Considera-se assim aceitável a opção de desenvolver em *C* os módulos cujo tempo de execução seja considerável e envolvê-los em Python.

²API, interface para programação de aplicações, sigla da descrição em inglês “Application Programming Interface”

A vasta gama de *software* livre, que inclui compiladores, ferramentas de desenvolvimento e bibliotecas de diferentes propósitos motivou a adopção do sistema operativo Linux, na sua distribuição Debian / GNU como sistema de desenvolvimento. A escolha de bibliotecas deve privilegiar a existência de *portés* para os sistemas operativos da Microsoft e para Mac OS.

Desta forma, e centrado no subsistema de Estimação Espectral, o subsistema mais relevante do ponto de vista deste trabalho, é possível constituir um sistema de processamento de sinais de fluxo sanguíneo que sirva os diversos propósitos anteriormente enunciados.

3.8 Utilização da Língua Portuguesa

A escolha da língua a utilizar no sistema de interface gráfica foi avaliada várias vezes no decurso do processo de prototipagem evolutiva adoptado.

A interface surgia inicialmente em português. Verificou-se, no entanto, que a utilização da localização portuguesa dificultava a integração com outras aplicações. De maior impacto no sistema é a diferença entre os formatos numéricos, nomeadamente a utilização da vírgula ou do ponto na separação das casas inteiras das decimais. Assim, optou-se por não estender a localização aos valores numéricos.

Na utilização de objectos do wxPython verificou-se que alguns dos campos surgiam apenas em inglês. O pacote Debian que contém a biblioteca do wxGTK, à data do desenvolvimento do sistema, encontrava-se numa fase instável (*unstable*), obrigando a frequentes actualizações. Sendo esperado que a questão da localização seja correctamente suportada em futuras versões da biblioteca, foi decidido que a tradução ou inclusão de suporte à língua portuguesa não seria efectuada nesta fase de desenvolvimento.

3.9 Conclusão

Um dos processos clínicos de diagnóstico cardiovascular inclui o recurso a equipamentos de ultra-som Doppler. A maioria destes equipamentos inclui o modo de funcionamento Doppler pulsado utilizado para a medição de parâmetros de fluxo sanguíneo associados à velocidade e turbulência do sangue.

A aplicação de métodos alternativos de processamento desses dados, obtidos por Doppler pulsado, integrados num sistema de processamento em tempo-real,

poderá produzir informação complementar para suportar diagnósticos clínicos. Variantes desse sistema podem ainda apoiar ensaios clínicos de outros métodos ou constituir uma base para investigação e desenvolvimento de sistemas de processamento de outros sinais.

Considerando as diferentes funcionalidades que o sistema pode servir, foi descrito, neste capítulo, o sistema computacional a desenvolver em cada um dos seus módulos integrantes: entrada de dados, processamento em tempo-real do sinal de fluxo sanguíneo e apresentação da informação produzida, sendo justificadas as opções e estratégias de desenvolvimento mediante “opiniões” expressas por outros investigadores em referências, e, testes executados.

Propositadamente, para apresentação da concepção e planeamento do sistema, não foi adoptada nenhuma metodologia ou formalismo de especificação. Uma tal abordagem prejudicaria a transmissão de uma perspectiva essencial que deve nortear o desenvolvimento: cada módulo de um subsistema pode/deve ser reutilizável e substituível.

Acredita-se que esta concepção do sistema é um contributo intangível, mas real. Os módulos devem ser encarados como *integrados de software*: cada um deles é uma *peça* que cumpre determinada função, nenhum é insubstituível, podendo ser, ou não, usado consoante a finalidade a que se destina.

Com raízes no processamento de sinais de ultra-som de fluxo sanguíneo, um sistema desenvolvido segundo os princípios enunciados pode, com algumas adaptações, ser aplicado a outros fins.

O núcleo do sistema de processamento de sinais de fluxo sanguíneo reside no subsistema de estimação espectral em que o objectivo de execução em tempo-real se procurará pela adequação entre os algoritmos dos métodos de estimação espectral e a arquitectura onde esses algoritmos serão implementados. A descrição detalhada dessa parte do estudo será apresentada nos capítulos 5 e 6.

O próximo capítulo incide sobre os subsistemas de aquisição e visualização de dados, mais complexos em virtude da variedade de objectivos que se pretende servir.

Capítulo 4

Subsistemas de Aquisição e Visualização de Dados

4.1 Introdução

A versatilidade pretendida, no sistema de processamento de sinais de fluxo sanguíneo e nos objectivos que visa cumprir conduz à consideração de diferentes origens dos dados a processar. Esta versatilidade vai introduzir complexidade na definição do sistema devida à generalidade e incapacidade de definição do formato dos dados.

Esta versatilidade também tem expressão no subsistema de visualização. Consoante a utilização dada ao sistema, pode ser relevante a apresentação de espectros, de indicadores ou de ambos. As tarefas do subsistema de visualização, não são fixas, dependendo não só do método de estimação utilizado como também do tipo de vaso ou válvula em estudo.

Como consequência da metodologia de prototipagem evolutiva, na sequência de algumas fases intermédias do subsistema de aquisição de dados, foi definido um formato a utilizar na aplicação. Aos dados provenientes das diferentes origens, no decurso da conversão, é adicionado um cabeçalho com informação necessária ao seu correcto enquadramento na aplicação e posterior visualização. Tentando manter a generalidade do sistema, a visualização de espectros é parametrizável.

O subsistema de processamento ou de estimação espectral é o cerne do sistema analisador de fluxo sanguíneo. No capítulo 8 tecem-se alguns comentários sobre a articulação e opções de desenvolvimento dos subsistemas de aquisição de dados e de visualização com o subsistema de processamento em função do alargamento do contexto da aplicação.

4.2 Fases Intermédias do Sistema de Aquisição de Dados

As práticas tradicionais de manipulação de sinais Doppler consistem na gravação e reprodução desses sinais usando equipamentos como gravadores de cassetes áudio, papel fotográfico ou filme, impressoras de vídeo (Evans, McDicken, Skidmore & Woodcock, 1989). Eram então necessárias algumas precauções para minimizar o erro produzido nos procedimentos de gravação e reprodução. Os sinais analógicos podiam ser convertidos para digital para posterior manipulação computacional. A conversão era, e é ainda, uma possível fonte de erro por ser necessário definir os níveis de resolução pretendidos. A visualização de dados digitalizados requer o uso de interpolação ou de imagens médias¹ caso seja necessária uma imagem contínua.

Mais recentemente é preferida a utilização de *digital audio tapes* (DAT's) ou a utilização de placas de aquisição de dados munidas de conversores analógico-digital (Evans & McDicken, 2000). Alguns dos factores a considerar na ligação entre os equipamentos ecográficos e os sistemas computacionais são número de canais, tempo de gravação, nível de quantização, resposta em frequência, gama dinâmica e razão sinal-ruído. Para além do desempenho do equipamento é também relevante a conveniência de utilização associada à identificação e acesso aos conteúdos.

4.2.1 Dados Simulados

A utilização de dados simulados nas fases de desenvolvimento, teste e entrada em exploração do sistema de processamento de sinais de fluxo sanguíneo é vantajosa: a dificuldade inerente à indefinição do formato dos dados é obviada; é possível testar a qualidade estatística e de implementação de métodos alternativos de estimação espectral e assim fazer uma escolha informada; convertendo os dados reais para o formato dos dados simulados é possível demonstrar, na entrada em exploração do sistema, a correcção dos resultados produzidos.

A simulação é feita a partir de um conjunto de sinais determinísticos de frequência média e largura de banda correspondente ao caso em estudo dando origem a sequências de ficheiros em número definido pelo utilizador. Cada sinal produzido corresponde a um ciclo cardíaco sendo gravado em ficheiros de texto

¹Do inglês, *frame average*

por requisitos de compatibilidade com outros sistemas existentes de estudo de sinais de fluxo sanguíneo.

Numa fase preliminar do módulo de aquisição de dados, optou-se por associar a cada sequência de ficheiros, um ficheiro de texto de descrição cujo conteúdo é actualizado nos diferentes subsistemas até assumir o formato apresentado na Tabela 4.1, contendo então a informação necessária à visualização simples do espectro.

Tabela 4.1: Ficheiro de descrição dos parâmetros dos sinais.

Descrição	Tipo
Frequência de amostragem (Hz)	Numérico (real)
Dimensão do segmento à entrada	Numérico (inteiro)
Passo	Numérico (inteiro)
Janela	Alfabético (Máx. 5 caracteres)
Dimensão do segmento à saída	Numérico (inteiro)
Valor máximo da frequência (Hz)	Numérico (real)
Valor mínimo da frequência (Hz)	Numérico (real)
Valor máximo da potência (Hz)	Numérico (real)
Valor mínimo da potência (Hz)	Numérico (real)
Descrição	Alfanumérico (Máx. 1024 caracteres)

A simulação determina os valores da frequência de amostragem e a dimensão do segmento à entrada. O campo de descrição indica os parâmetros usados na simulação, vaso ou válvula simulado, nível de ruído, ausência ou presença de estenose. Os demais campos, a serem actualizados posteriormente pelos módulos de estimação espectral, são, nesta fase, escritos a branco ou zero.

Todos os sinais são convertidos para formato binário em representação real ou complexa da forma cartesiana, pela evocação de um programa de conversão de dados desenvolvido para o efeito.

```
file_man -h
```

```
Usage: file_man <direction> input_file_name input_file_extension
       files_from files_to output_file_name output_file_extension
```

```
<direction>:  asc2bin = 0
              bin2asc = 1
              asc2cplx = 2
              cplx2asc = 3
```

4.3 Diversas Origens dos Dados para Processamento Diferido

Por dados clínicos entenda-se informação respeitante ao ultra-som retornado de um objecto em análise. Idealmente, para a análise de fluxo sanguíneo em Doppler pulsado, significaria os valores “puros”, correspondendo ao deslocamento em frequência (*Doppler shift*), com o mínimo de filtragem ou outros processamentos. Estes dados correspondem aos que se obteriam interceptando a ligação da sonda de ultra-som ao ecógrafo.

No entanto, nem sempre é esse o caso. Quando os sinais foram já alterados de forma significativa são aqui designados por sinais processados.

Através da colaboração com diversas entidades foi possível obter dados de fluxo sanguíneo provenientes de sistemas de ultra-som. Os diferentes formatos abaixo identificados encontram-se descritos em detalhe no apêndice A.

Estes dados provêm das origens especificadas nas subsecções seguintes.

4.3.1 Hospital Distrital de Faro

O sistema ecográfico adquirido em 1999 pelo HDF é um *GE Vingmed System V* da *General Electrics* (GE, 2000a). Este sistema implementa o standard *DICOM*² conforme se encontra descrito na página de Internet do fabricante (GE, 2000b).

Outros fabricantes já adoptam também o standard *DICOM* que regula o formato digital de imagens e comunicação em medicina (DICOM, 2000), na sua terceira versão à data da escrita desta dissertação. Existe, no domínio público, um conjunto de ferramentas que visam a manipulação de ficheiros em formato *DICOM*, versão 3 (Clunie, 2002). O standard privilegia as imagens aos dados nas operações de recolha, transferência e armazenamento.

Associado ao ecógrafo existe um sistema designado *EchoPac* que consiste num computador equipado com *software* para armazenamento e tratamento de imagens seleccionadas e recolhidas durante o exame clínico. Esse computador é um Macintosh a correr o sistema operativo MacOS 9 ligado por fibra óptica (*FireWire*) ao ecógrafo. Este sistema prevê a exportação de dados reais não tratados sendo necessário o produto *EchoMat*. A fusão entre as firmas GE e Vingmed e a conseqüente reestruturação de produtos e serviços impossibilitou a aquisição do

²*Digital Imaging and Communication in Medicine*

EchoMat.

Assim, não é possível obter os dados não tratados provenientes da sonda, mas apenas o armazenamento e posterior recolha das imagens produzidas durante um exame clínico. A comparação dessas imagens com as produzidas pelo sistema aqui descrito é de significado questionável. Ainda que fossem obtidas a partir do mesmo paciente, devido ao diferente processamento a que são sujeitos não é possível validar quantitativamente o sistema desenvolvido.

4.3.2 Universidade de Florença - Itália

Num contacto com um grupo do *Dipartimento di Elettronica e Telecomunicazioni* da *Università degli Studi di Firenze*, Itália, foram obtidos alguns ficheiros em formato 'TIM' contendo sinais clínicos de fluxo sanguíneo da veia jugular obtidos com uma sonda *multigate* (Tortoli, Guidi, Guidi & Atzeni, 1996; Guidi, Guidi, Ricci, Atzeni & Tortoli, 1998). Esses sinais, em fase e em quadratura, permitem a extracção do sinal de fluxo directo e inverso.

4.3.3 Universidade de Leicester - Reino Unido

Ao abrigo de colaboração com o *Medical Physics Department* da *Leicester University* foram obtidos alguns ficheiros em diversos formatos ('SON', 'ALG' e 'DOP') contendo sinais de fluxo sanguíneo também de diversos vasos de diversos indivíduos. Foram também disponibilizados dados reais recolhidos da artéria cerebral média por Doppler intracraniano ('EMB' e 'MIX'). De referir que em alguns destes dados se verifica a ocorrência de embolias.

4.3.4 Universidade Nacional Autónoma do México

Ao abrigo de colaboração com o *Departamento de Ingeniería de Sistemas Computacionales y Automación* do *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas* da *Universidad Nacional Autónoma de México* foram obtidos alguns ficheiros em formato 'WAV' contendo sinais de fluxo sanguíneo de diferentes vasos de diversos indivíduos. A sonda (Fuentes, Sotomayor, García, Acevedo & Moreno, 2002; García, Moreno, Solano, Berragán, Sotomayor & Fuentes, 2002) – um transdutor ultra-sónico de Doppler contínuo – utilizada para a recolha destas amostras é produto de trabalho de investigação e desenvolvimento de diversas equipas de investigação do projecto internacional MAGIAS (Métodos

Avanzados de Generación de Imágenes Acústicas, projecto UNESCO: 2201.09, 3304.17, 3307.22, 3311.10). Previa-se que um exemplar destes transdutores seria atribuído ao laboratório onde este trabalho foi desenvolvido, possibilitando assim a consolidação da aplicação em tempo-real.

Numa fase final do desenvolvimento do trabalho aqui descrito obteve-se então um exemplar de um de estes transdutores de ultra-som Doppler contínuo, com o qual é possível proceder à aquisição de sinais Doppler de ultra-som contínuo para um computador equipado com uma carta de som.

Está em curso o desenvolvimento e teste de um transdutor de ultra-som Doppler pulsado, trabalho das mesmas equipas.

4.3.5 Geradores de Sinais

Foram utilizados geradores de sinais de fluxo sanguíneo cujo método de simulação se baseia na proposta por (Wang & Fish, 1996). Esses geradores de sinais permitem a produção de sinais, estacionários ou não, de fluxo sanguíneo com as características genéricas dos obtidos recorrendo a instrumentação Doppler de onda contínua ou pulsada. A simulação é feita a partir de um conjunto de sinais determinísticos de frequência média, largura de banda e variação de potência correspondente ao caso em estudo.

Esses geradores foram desenvolvidos por diferentes autores (Costa, 1997; Cardoso, 1998) em Matlab e integrados no sistema de simulação IPSISBio (Vistas & Reis, 1999), também em Matlab. IPSISBio é parametrizável podendo ser definido o vaso ou válvula em questão, o nível de ruído, a ausência ou presença (e em que grau) de estenose bem como produzir sinais estacionários e não estacionários.

IPSISBio foi posteriormente estendido para incluir um outro gerador de sinais (Sousa, 1999; Leiria, 2000) e sendo então acrescida a funcionalidade de, a partir de imagens digitalizadas de exames ecográficos, extrair por envelopagem do sinal, as curvas teóricas de outros vasos ou válvulas.

Qualquer um dos simuladores origina ficheiros de texto contendo apenas o sinal.

4.4 Formato dos Dados

Pretendendo obviar o impacto das diferentes origens do sinal optou-se por definir os formatos a utilizar neste sistema para o qual serão convertidos os sinais

exteriores.

Os diferentes tipos de dados de Doppler pulsado de sinais de fluxo sanguíneo a considerar incluem sinais simulados e reais, processados ou não.

Todos os ficheiros contêm um cabeçalho de teor informativo com a dimensão máxima de 128 *byte* onde se procurou manter a informação relevante para o processamento dos sinais e para a identificação da origem e processamento a que foram sujeitos. Os dados surgem sempre a partir do *byte* 128.

O tipo de representação numérica adoptada para os sinais de entrada pode ter um impacto significativo no desempenho das aplicações. Os tempos de leitura e escrita, o volume de memória ocupado, a complexidade algorítmica dos métodos e a acuidade da estimação (como consequência da acumulação de erros numéricos) podem variar consoante a representação numérica adoptada, variando de método para método. Foi decidido manter no cabeçalho dos ficheiros de dados a indicação sobre o tipo (real ou complexo) dos dados bem como a aconselhada representação numérica (precisão simples ou dupla) dos dados. A execução de cada um dos métodos poderá assim ser ajustada consoante a natureza dos dados.

4.4.1 Dados Simulados

Foi adoptado a formatação descrita na Tabela 4.2 para o cabeçalho dos ficheiros de dados temporais simulados obtidos por um dos vários simuladores.

4.4.2 Dados Reais

Dados temporais não tratados, correspondentes ao deslocamento em frequência, podem ser armazenados digitalmente para posterior tratamento. No sistema desenvolvido define-se que cada ficheiro contém os dados de um único canal, sendo alguns dados informativos mantidos no cabeçalho do ficheiro conforme descrito na Tabela 4.3.

4.4.3 Dados Processados

Os espectrogramas, resultantes do processamento de sinais simulados ou reais, são armazenados em ficheiros cujo cabeçalho é descrito na Tabela 4.4.

Tabela 4.2: Formato do cabeçalho dos ficheiros de sinais simulados.

Posição (byte)	Descrição	Tipo
0 - 3	Origem 'SIMV'	(char * 4)
4 - 5	Vaso ou válvula	(char * 2)
	'01' Artéria carótida primitiva	
	'02' Válvula aórtica	
	'03' Válvula mitral	
	'04' Artéria cerebral média	
6 - 9	Frequência de amostragem (MHz)	(float * 1)
10 - 13	Frequência Doppler (MHz)	(float * 1)
14 - 17	Ângulo (graus)	(float * 1)
18 - 21	Gama de frequências (kHz)	(float * 1)
22 - 25	Frequência heterodina (kHz)	(float * 1)
26 - 29	Velocidade do ultra-som no meio ($m s^{-1}$)	(int * 1)
30 - 33	Número de ciclos cardíacos	(int * 1)
34 - 37	Número de pontos por ciclo cardíaco	(int * 1)
38 - 49	Reservado	(char * 12)
50 - 61	Sinal teórico	(char * 12)
62	Tipo de banda (f/v)	(char * 1)
63 - 66	Valor da banda	(float * 1)
	'f' → valor em Hz	
	'v' → % sinal teórico	
67	Ruído (s/n)	(char * 1)
68 - 71	's' → SNR (dB)	(float * 1)
72 - 125	Reservado	(char * 54)
126	Tipo de dados	(char * 1)
127	Precisão do tipo de dados	(char * 1)

4.5 Conversores

A informação necessária aos cabeçalhos dos ficheiros convertidos é recolhida na opção "Conversão". Na leitura e escrita de ficheiros em *Python* são privilegiados o modo texto e um modo específico da linguagem - *pickle* - que corresponde à criação de objectos persistentes.

A conversão dos vários formatos é necessária apenas numa fase intermédia resultante da não integração dos geradores de sinal no sistema de processamento de sinais biomédicos e da não definição do sistema de aquisição a utilizar em modo clínico. Por não ser uma área crítica do sistema e na continuidade da solução anteriormente adoptada, a conversão é, em cada particularização, realizada por pequenos programas em C em extensão do *Python*.

Tabela 4.3: Formato do cabeçalho dos ficheiros de sinais reais.

Posição (byte)	Descrição	Tipo
0 - 5	Origem 'CONVxx' xx designa a versão utilizada	(char * 6)
6 - 9	Frequência de amostragem (MHz)	(float * 1)
10 - 13	Frequência Doppler (MHz)	(float * 1)
14 - 17	Ângulo (graus)	(float * 1)
18 - 21	Gama de frequências (kHz)	(float * 1)
22 - 25	Frequência heterodina (kHz)	(float * 1)
26 - 29	Velocidade do ultra-som no meio ($m s^{-1}$)	(int * 1)
30 - 33	Comprimento dos dados	(int * 1)
34 - 49	Reservado	(char * 16)
50 - 61	Sinal original	(char * 12)
62 - 125	Reservado	(char * 64)
126	Tipo de dados	(char * 1)
127	Precisão do tipo de dados	(char * 1)

As Figuras 4.1, 4.2 e 4.3 ilustram os vários conversores desenvolvidos no decurso do projecto A semelhança entre os vários conversores deve-se ao facto de o formato adoptado pela aplicação requerer um bloco de informação comum que não é possível extrair dos ficheiros em formato estrangeiro.

O conversor de dados simulados (Fig. 4.1) produz, a partir dos ficheiros de texto ASCII indicados, ficheiros binários contendo, no cabeçalho a informação fornecida pelo utilizador.

Tabela 4.4: Formato do cabeçalho dos ficheiros de espectrogramas.

Posição (byte)	Descrição	Tipo
0 - 3	Método	(char * 4)
4 - 5	Versão do método	(char * 2)
6 - 9	Frequência de amostragem (MHz)	(float * 1)
10 - 13	Frequência Doppler (MHz)	(float * 1)
14 - 17	Ângulo (graus)	(float * 1)
18 - 21	Gama de frequências (kHz)	(float * 1)
22 - 25	Frequência heterodina (kHz)	(float * 1)
26 - 29	Velocidade do ultra-som no meio ($m s^{-1}$)	(int * 1)
30 - 49	Reservado	(char * 20)
50 - 61	Sinal estimado	(char * 12)
62 - 81	Parâmetros gerais ex: ' han 2048 1024 gsan '	(char * 20)
82 - 113	Parâmetros específicos ex: ' m = 5 sigma = 0.5 krn = std '	(char * 32)
114 - 117	Dimensão do segmento	(float * 1)
118 - 121	Tempo de cada segmento	(float * 1)
122 - 125	Percentagem de sobreposição	(float * 1)
126	Tipo de dados	(char * 1)
127	Precisão do tipo de dados	(char * 1)

Converter dados simulados

Escolher a sequência de ficheiros de sinal

De A

Ficheiro de destino Listar

Sinal teórico

Vaso/Válvula

Freq. Amostragem (MHz) Freq. Doppler (MHz)

Ângulo (grau) Gama de freq. (KHz)

Freq. heterodina (KHz) Velocidade US (m/s)

No. ciclos cardíacos No. pontos por ciclo

Banda Valor

Ruído (dB)

Converter Sair

Figura 4.1: Conversor de dados simulados

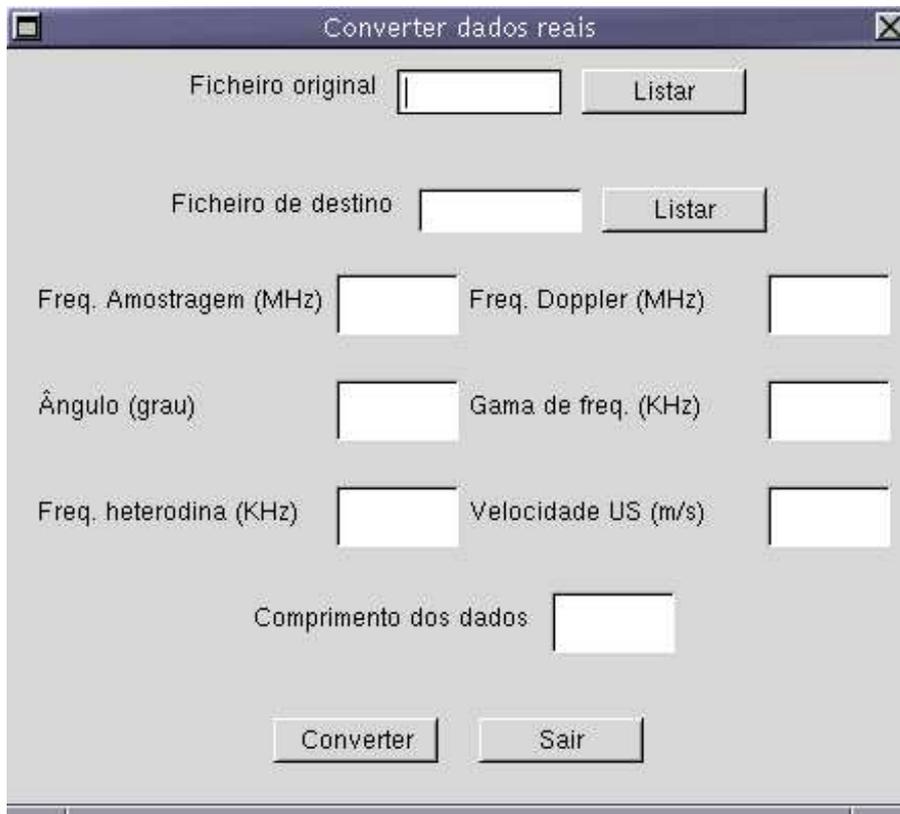


Figura 4.2: Conversor de dados reais

O conversor de dados reais (Fig. 4.2) converte os ficheiros de tipo 'WAV' e 'TIM'.

As amostras contidas nos ficheiros de tipo 'WAV' utilizados (Microsoft PCM) têm sempre 16 *bit*, tendo o conversor sido vocacionado para este tipo de ficheiros. O tempo de conversão de um ficheiro de 33,61 s é inferior a um segundo não introduzindo latência significativa no sistema de processamento.

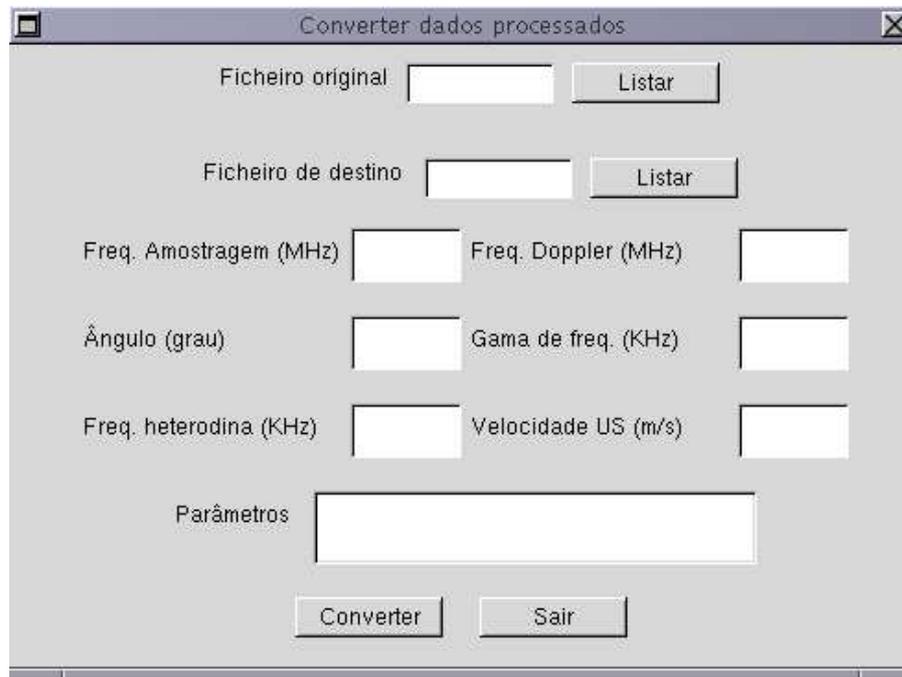


Figura 4.3: Conversor de dados processados

4.6 Aquisição de Dados Reais em Tempo-Real

Considerando a sonda obtida (Fuentes, Sotomayor, García, Acevedo & Moreno, 2002), o formato “WAV” assumiu especial relevância, como a necessidade de adquirir sinais a partir da carta de som. O pacote *wavtools* (Ligertwood, 2002) da Debian que permite ler e escrever ficheiros do tipo “WAV” foi escolhido por ser simples mas eficaz. Este pacote foi alterado de forma a incluir um programa que lê os dados da carta de som e, para além da escrita dos mesmos para um ficheiro, os duplica para um canal (*named pipe - FIFO special file*) onde serão lidos pelos módulos de estimação espectral.

A par deste programa, foi criado um segundo que transfere para o mesmo canal os dados contidos num ficheiro.

Estes procedimentos são transparentes para o utilizador, em função da utilização de uma mesma janela (Fig. 4.4): se a opção “Directo” for seleccionada, a aquisição de dados faz-se a partir da carta de som e os dados são guardados no ficheiro indicado e escritos no canal; caso contrário, os dados são lidos do(s) ficheiro(s) indicado(s) e escritos no canal.

Novas versões dos estimadores passaram a considerar a leitura do canal.



Figura 4.4: Aquisição de dados para estimação espectral

Verificou-se, em termos médios, um acréscimo no tempo de execução dos estimadores na máquina *coral*, um processador a 500 MHz e 64 MB de RAM, devido à utilização dos canais. Pelo contrário, no caso da máquina *ticoteco*, com dois processadores a 350 MHz e 128 MB de RAM, regista-se um decréscimo dos tempos de execução face às versões anteriores que incluíam a leitura de dados de ficheiros.

4.7 Visualização

A representação gráfica do resultado da aplicação dos métodos de estimação espectral pode variar conforme o objectivo pretendido, podendo resultar na apresentação simples de um ou mais espectrogramas, acompanhados ou não de valores de parâmetros clínicos.

4.7.1 Apresentação Simples

Em qualquer dos casos, o cerne deste subsistema é a apresentação simples do espectrograma em tempo-real. Esta forma de visualização, com a sonda obtida, viabiliza a ultra-sonografia do fluxo sanguíneo; utilizada em duplicado (paralelo), permite a comparação de espectros; utilizada em tempo diferido, permite o re-exame de sinais e poderá ser a base de desenvolvimento do módulo para exploração clínica do sistema.

Para cumprir os requisitos de tempo-real, recorreu-se a *g2* (Milanovic & Wagner, 2002), uma biblioteca gráfica *freeware* para sistemas Linux com sistema

gráfico baseado em XFree. De realçar que esta biblioteca, que implementa algumas primitivas gráficas, considera dispositivos virtuais que se podem concretizar como uma janela gráfica ou um ficheiro *Postscript*.

Em termos de utilização, a apresentação do espectrograma é conseguida pela selecção da opção de visualização na janela de estimação espectral, conforme ilustrado na figura 4.5, o que desencadeia a abertura de uma janela, conforme se



Figura 4.5: Definição da opção de visualização da estimação.

apresenta na Figura 4.6.

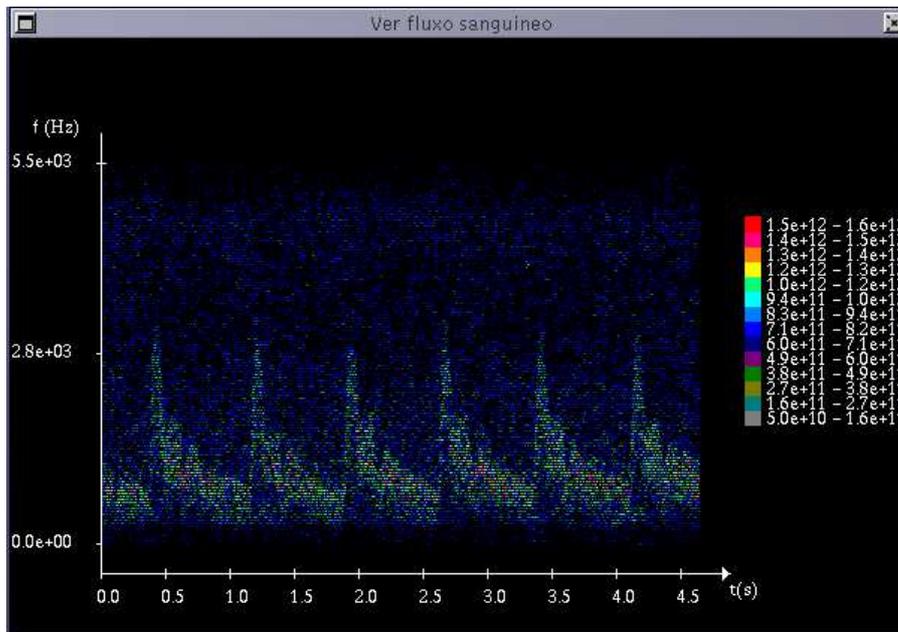


Figura 4.6: Espectrograma simples

4.7.2 Espectro Médio de Sinais Simulados

Uma das formas de diminuir a variância na estimação espectral consiste em calcular o espectro médio de um número de espectros individuais. Esse número pode variar consoante o método de estimação aplicado.

Considerando o espectro de dados simulados, procedeu-se à implementação de um programa para o cálculo do espectro médio, sendo o número de sinais a considerar um parâmetro de entrada. Esse programa lê um ficheiro de descrição (cf. Tabela 4.1) associado a uma sequência de ficheiros, descrição essa criada pelo conversor de dados simulados e actualizada pelo método de estimação espectral.

Seja N o número de espectros individuais que contribuem para cada espectro médio. Assim que é criado o primeiro espectro médio, a cada novo espectro individual é criado um novo espectro médio. Foram tentadas várias abordagens, tendo-se conseguido uma implementação bastante satisfatória que pode ser descrita por

$$\bar{X}_0 = \sum_{k=0}^{N-1} X_k/N \quad (4.1)$$

$$\bar{X}_1 = \frac{1}{N}(X_N - X_0) + \bar{X}_0 \quad (4.2)$$

$$\vdots \quad \vdots \quad (4.3)$$

$$\bar{X}_t = \frac{1}{N}(X_{t+(N-1)} - X_{t-1}) + \bar{X}_{t-1} \quad (4.4)$$

$$(4.5)$$

Os valores máximos e mínimos de frequência e de potência foram incluídos no ficheiro de descrição para tornar a visualização independente do processo de estimação espectral. Esses valores limites de frequência indicam qual a maior ampliação (*zoom*) possível do espectro sem que haja corte de informação útil. Os valores limites de potência permitem distribuir o código de cores por uma gama de amplitudes do sinal.

4.7.3 Espectro Médio de Sinais Clínicos

A criação do espectro médio de sinais clínicos pressupõe a segmentação em ciclos cardíacos individuais. Uma abordagem possível foi descrita na secção 2.3.6. A segmentação de sinais clínicos estava a ser estudada pela Eng^a Ana Leiria pelo que, à data da implementação deste subsistema, foi decidido que a implementação

do método escolhido seria feita após a conclusão desses estudos.

4.8 Conclusão

A aquisição de dados é, pela flexibilidade pretendida, um sistema complexo e de difícil optimização. A utilização de dados simulados permite o estudo de sistemas alternativos de processamento de sinais. Caso a caso será necessário definir a estratégia a adoptar que pode incluir o desenvolvimento de conversores ou dispositivos de controlo de entrada de dados ou apenas a aquisição de uma placa de interligação entre o equipamento ecográfico e o sistema computacional.

Os dados, independentemente da sua origem, serão sujeitos a métodos de processamento de sinal alternativos à FFT, originando espectros de sinal de fluxo sanguíneo. Esses espectros podem ser visualizados individualmente em tempo-real. Caso os espectros tenham sido obtidos pela estimação de sinais simulados, é também possível visualizar o espectro médio.

A flexibilidade de utilização do sistema é conseguida pela inclusão de vários módulos de funcionalidade semelhante, cada um vocacionado para um cenário possível. É de realçar o facto de o sistema, na sua totalidade, não depender de nenhum deles.

O subsistema de visualização, utilizando uma livreria potente e muito simples, representa um equilíbrio entre a carga admissível de um subsistema e a funcionalidade pretendida. Outras bibliotecas, mais pesadas, permitiriam obter resultados esteticamente mais interessantes sacrificando talvez a capacidade da máquina de proceder à estimação de espectros em tempo-real.

O elevado volume de processamento dos métodos considerados sugere a necessidade de utilização de técnicas de processamento paralelo que viabilizem a implementação em tempo-real, tema dos próximos capítulos.

Capítulo 5

Subsistema A de Estimação Espectral

5.1 Introdução

Na etapa inicial deste trabalho, em meados de 1998, considerava-se a adição de processadores dedicados ao processamento digital de sinal para aumentar a capacidade de processamento de um computador.

Os processadores dedicados exclusivamente à execução dos módulos de estimação espectral foram ligados entre si, constituindo o que designaremos genericamente por plataformas aditivas.

O processamento atribuído às plataformas aditivas consistia na estimação de um segmento de dados, sendo usados os valores de desempenho óptimo para os parâmetros dos estimadores.

Neste capítulo é descrita a etapa do estudo da implementação dos métodos de estimação espectral considerados em plataformas aditivas, sendo descritas as implementações mais relevantes, remetendo-se para o capítulo 6 a etapa do estudo efectuado em processadores genéricos.

5.2 Uso de Plataformas Aditivas

Nesta etapa são consideradas as plataformas aditivas integrando os processadores *transputers*, C40's, i860 e Sharc descritos na secção 2.4.1. Foram testadas três plataformas de processamento paralelo que consideram a adição a um computador de processadores específicos para processamento digital de sinal.

5.2.1 Arquitecturas de Implementação

A Figura 5.1 representa o que designaremos por Plataforma Aditiva A, incorporando até três T8's, três C40's e um i860. A máquina hospedeira é uma máquina Sun, correndo o sistema operativo Solaris. O código a executar nos processadores foi codificado em C, sendo usados compiladores distintos de acordo com o fabricante/fornecedor dos processadores: para os T8, usou-se *Inmos ANSI C* (INMOS Limited, n.d.) ; para os C40, *Transtech C* (Transtech Parallel Systems Ltd., 1992); e para o i860, *Portland Group C compiler (pgcc)*. Esta plataforma,

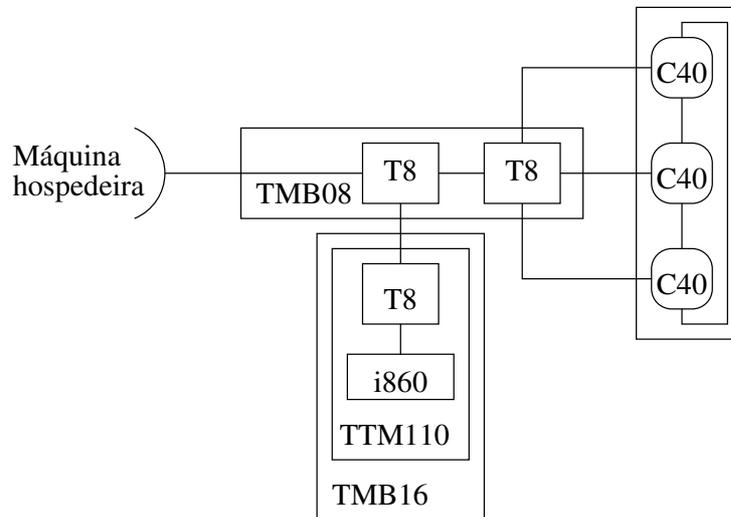


Figura 5.1: Plataforma aditiva A.

propriedade do *Automatic Control & Systems Engineering Department* da *The Sheffield University*, foi utilizada no decurso de trabalhos de investigação efectuados ao abrigo do tratado de Windsor.

Outra plataforma testada, que designaremos por Plataforma Aditiva B, é um computador pessoal incorporando um C40 é representada na Figura 5.2. A escolha desta plataforma deve-se ao facto de não serem usados *transputers* na comunicação com a máquina hospedeira. O computador pessoal era equipado com um processador Pentium a 133 MHz, com 16 *Mbyte* de RAM correndo o sistema operativo Microsoft Windows 95 sendo, na fase final do estudo, um Pentium Pro a 200 MHz, com 128 *MByte* de RAM sob o sistema operativo Microsoft Windows 98.

Um processador Sharc (ADSP 20162) associado a um computador pessoal é

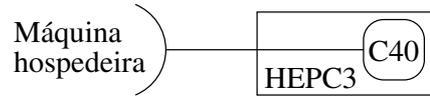


Figura 5.2: Plataforma aditiva B.

representado na Figura 5.3 correspondendo à Plataforma Aditiva C. Este processador, que não existia na Universidade do Algarve à data do estudo, é propriedade do *Departamento de Ingeniería de Sistemas Computacionales y Automatización* do *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas* da *Universidad Nacional Autónoma de México*. Para testar esta plataforma procedeu-se ao envio do código fonte utilizado na implementação da Plataforma Aditiva B. Nesta plataforma foi usado o compilador *3L C* (3L Limited, 1995).

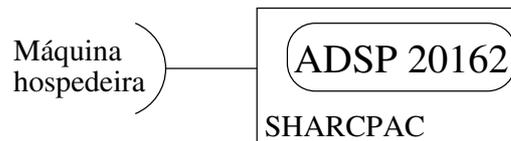


Figura 5.3: Plataforma aditiva C.

5.2.2 Algoritmos

Os algoritmos considerados correspondem aos métodos AR-MC e CWD descritos na secção 2.3, em relação aos quais existia a expectativa de que pudessem substituir a STFT na estimação em tempo-real de espectros de fluxo sanguíneo.

5.2.2.1 AR - Modified Covariance

O algoritmo do estimador AR-MC (conforme a implementação (Madeira, 1997)) pode ser decomposto em, pelo menos, quatro blocos funcionais que deverão ser aplicados sequencialmente: a constituição do sistema de equações lineares de covariância modificada (cf. Eq. 2.10), a resolução desse sistema pela decomposição de Cholesky conduzindo à determinação dos parâmetros do modelo auto-regressivo com base nos quais é calculada a variância do ruído branco (cf. Eq. 2.9) e, finalmente, a densidade espectral de potência (cf. Eq. 2.8). O algoritmo computacional utilizado corresponde à codificação em linguagem C do algoritmo em Matlab fornecido como especificação, implementação que traduz a formalização

apresentada em (Kay, 1988). Foi considerado um modelo de quatro pólos, conforme determinado por (Ruano, 1992) para sinais simulados da artéria carótida.

Os segmentos de dados fornecidos para teste correspondem à aplicação de janelas de 10 ms. a sinais simulados de artéria carótida primitiva com diferentes frequências de amostragem, nomeadamente 6,4, 12,8, 25,6 e 51,2 kHz (Ruano, 1992). Assim, a estimação de um segmento deve ser obtida em menos de 10 ms. quer o comprimento dos segmentos seja 64, 128, 256 ou 512 pontos, para respeitar os limites de tempo-real.

5.2.2.2 TFD de Choi-Williams

O algoritmo da DCWD (cf. Eq. 2.19) consiste no produto da função instantânea da autocorrelação do sinal por uma janela temporal e pelo núcleo exponencial que caracteriza a distribuição, seguido do cálculo da densidade espectral de potência. O algoritmo utilizado considera a forma analítica *standard* do sinal temporal. A decomposição em blocos sequenciais corresponde assim ao cálculo de: sinal *analítico*, autocorrelação e produto pela janela e núcleo, e, densidade espectral de potência. O algoritmo computacional utilizado corresponde à codificação em linguagem C do algoritmo em Matlab fornecido como especificação, conforme descrito em (Cardoso, 1998), tendo sido considerados os valores óptimos dos parâmetros do núcleo determinados por (Cardoso, 1998) para sinais simulados de artéria carótida.

O segmento de dados fornecido para teste corresponde a um segmento de 640 ms. de um sinal simulado de artéria carótida primitiva correspondente a uma frequência de amostragem de 12,8 kHz. Para facilitar a comparação, a aplicação da distribuição considera janelas de 1,25, 2,5, 5, 10, 20 e 40 ms. aplicadas a cada 1,5625 ms. (tempo correspondente a 20 pontos) originando assim segmentos de dados de comprimento 16, 32, 64, 128, 256 e 512 pontos com taxas de sobreposição de 0, 37,5, 69, 84, 92 e 96% respectivamente.

5.2.3 Implementações na Plataforma Aditiva A

Em (Madeira, 1997) foi apresentado um estudo da implementação do AR-MC em arquiteturas paralelas homogêneas sobre a plataforma aditiva A baseadas em *transputers* ou C40's em que cada um dos blocos foi estudado separadamente,

visando a diminuição da complexidade algorítmica e a criação de algoritmos paralelos. Os tempos de comunicação entre a máquina hospedeira e a arquitectura homogénea de *transputers* são de tal ordem que inviabilizam este tipo de solução para futura integração num sistema em tempo-real. Excluindo os tempos de transferência dos dados entre a máquina hospedeira e os elementos de processamento, foi avaliado o desempenho das arquitecturas na estimação de segmentos de dados de diferentes tamanhos. A arquitectura homogénea de C40's revela ser capaz de implementar em tempo-real o algoritmo embora no caso estudado a comunicação com a máquina hospedeira seja também feita através de *transputers*.

Na continuação desse trabalho foram estudadas, e são aqui apresentadas, as implementações do estimador AR-MC na Plataforma Aditiva A usando apenas o processador vectorial i860 ou usando uma arquitectura heterógena composta por C40s e o i860 (Madeira, Tokhi & Ruano, 1997; Madeira, Tokhi & Ruano, 2000). Ainda sobre a Plataforma Aditiva A foi feito um estudo comparativo das implementações da CWD (Madeira, Tokhi & Ruano, 1999).

5.2.3.1 AR-MC no Processador i860

A implementação utilizando o processador i860 da Plataforma Aditiva A como elemento de processamento utiliza, para o encaminhamento de código e dados, os dois T8s que se situam entre a máquina hospedeira e o i860.

O processo de criar uma aplicação para o processador i860 é mais elaborado que o necessário para aplicações usando *transputers* ou C40s. O compilador utilizado foi o *Portland Group C Compiler* (pgcc) sobre o sistema operativo MS-DOS sendo o código objecto transferido para a máquina hospedeira por *ftp* binário onde é configurado para posterior execução. A Figura 5.4 sumaria os passos envolvidos no processo de criação de uma aplicação para processador i860 (Chambers, 1995), sendo os blocos rectangulares representativos de comandos emitidos.

Conforme se pode verificar na Figura 5.4, são utilizados três ficheiros de configuração: *system.cfg* relaciona a máquina hospedeira com os dois *transputers*; *kernel.cfs* define a ligação entre as cartas TMB16 e TTM110; *loader.cfs* para a ligação entre o i860 e o *transputer* associado. Desta forma, são criadas duas aplicações que iniciam os processadores e são executadas simultaneamente.

O processo que implementa o estimador AR-MC foi atribuído ao processador i860 da plataforma aditiva A, recebendo os dados da máquina hospedeira através

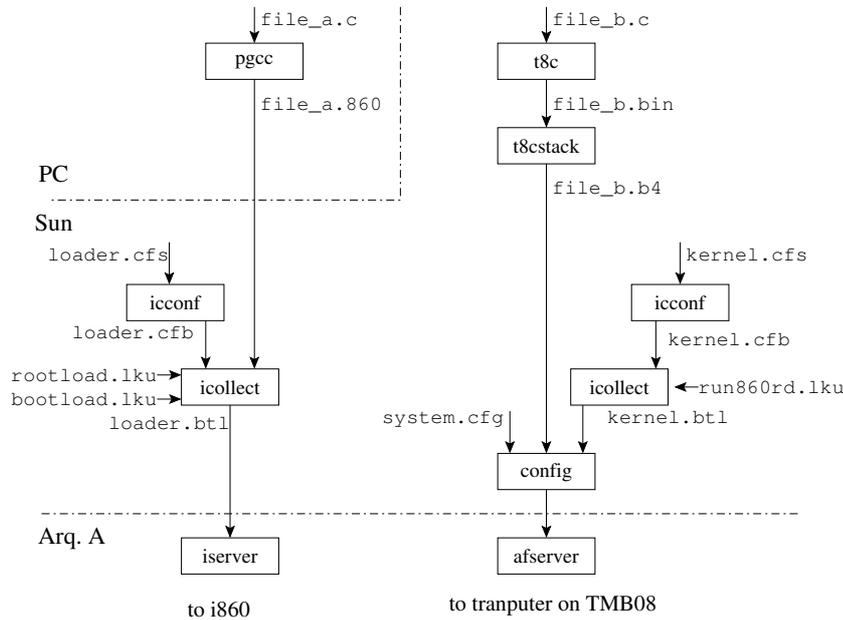


Figura 5.4: Processo de criação de uma aplicação para processador i860.

de dois *transputers* cuja função se restringia a comunicação. A Tabela 5.1 apresenta o tempo de cálculo (em ms.), medido no i860, correspondente à aplicação do estimador AR-MC para segmentos de dados, de 10 ms. de duração, de vários comprimentos.

Comprimento do Segmento	Tempo de execução (ms.)
64	2,766
128	5,332
256	10,779
512	20,673

Tabela 5.1: Tempos de cálculo do estimador AR-MC no processador i860.

Em (Chambers, 1995) é referido que o uso das opções de otimização do *pgcc* se traduz em diferenças significativas dos tempos de execução. Conforme (Transtech Parallel Systems Ltd., 1992) o nível um de otimização é aconselhado para algoritmos irregulares, com poucas instruções de ciclo e pouco uso de aritmética de vírgula flutuante; o nível dois é aconselhado quando o algoritmo implementa funções curtas com muitas operações de vírgula flutuante; o nível três é aconselhado para a maioria dos algoritmos sendo ressalvado o uso do quarto nível para quando o número de iterações dos ciclos pode ser calculado em tempo de execução.

Os tempos de cálculo em ms. obtidos com os níveis de otimização de um a quatro são apresentados na Tabela 5.2, e ilustrados na Figura 5.5 que apresenta os tempos de execução, sem e com uso das opções de otimização do compilador, obtidos na estimação pelo processador i860 de um segmento de dados.

Comprimento do Segmento	Nível 1 (ms.)	Nível 2 (ms.)	Nível 3 (ms.)	Nível 4 (ms.)
64	2,168	1,319	0,929	0,933
128	4,141	2,386	1,548	1,555
256	8,09	4,533	2,801	2,808
512	15,9	8,756	5,214	5,215

Tabela 5.2: Efeito da otimização efectuada pelo compilador no algoritmo do estimador AR-MC executado pelo processador i860.

Ao gráfico foi acrescentada uma linha representativa do limite de tempo-real, sendo patente que, para níveis de otimização superiores a um, a estimação é obtida em tempo-real.

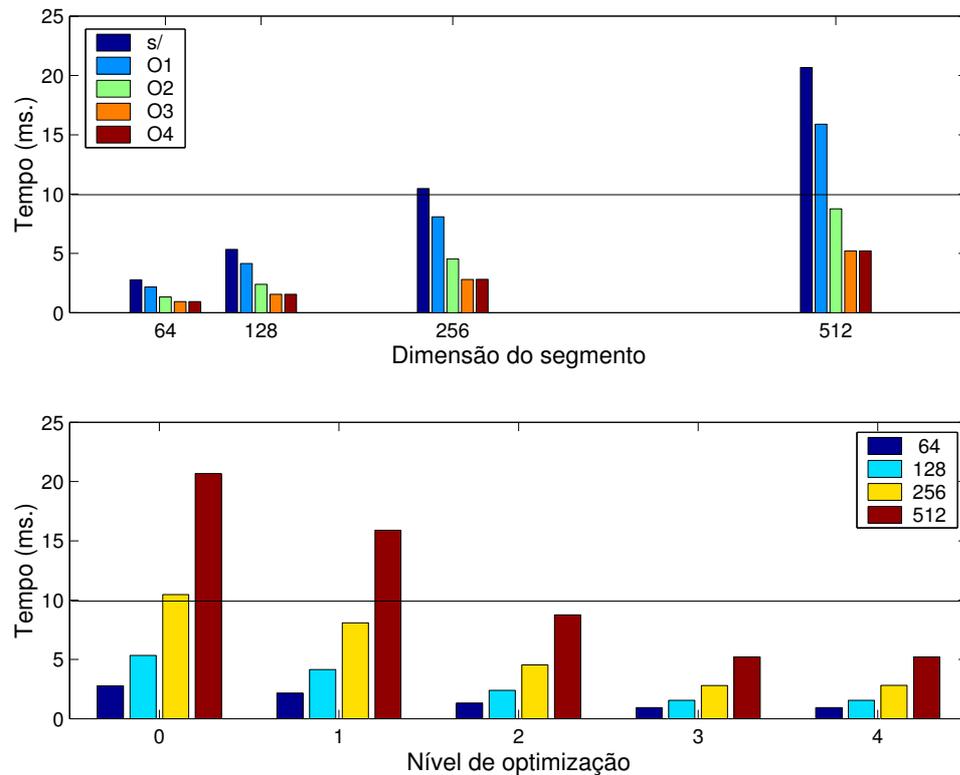


Figura 5.5: Tempos de execução, em ms., do algoritmo do estimador AR-MC no processador i860. Linha horizontal a 10 ms. correspondente ao limiar de tempo-real.

A regularidade do algoritmo é expressa pelo crescimento linear dos tempos de execução com a dimensão do segmento, conforme se pode verificar na Figura 5.6.

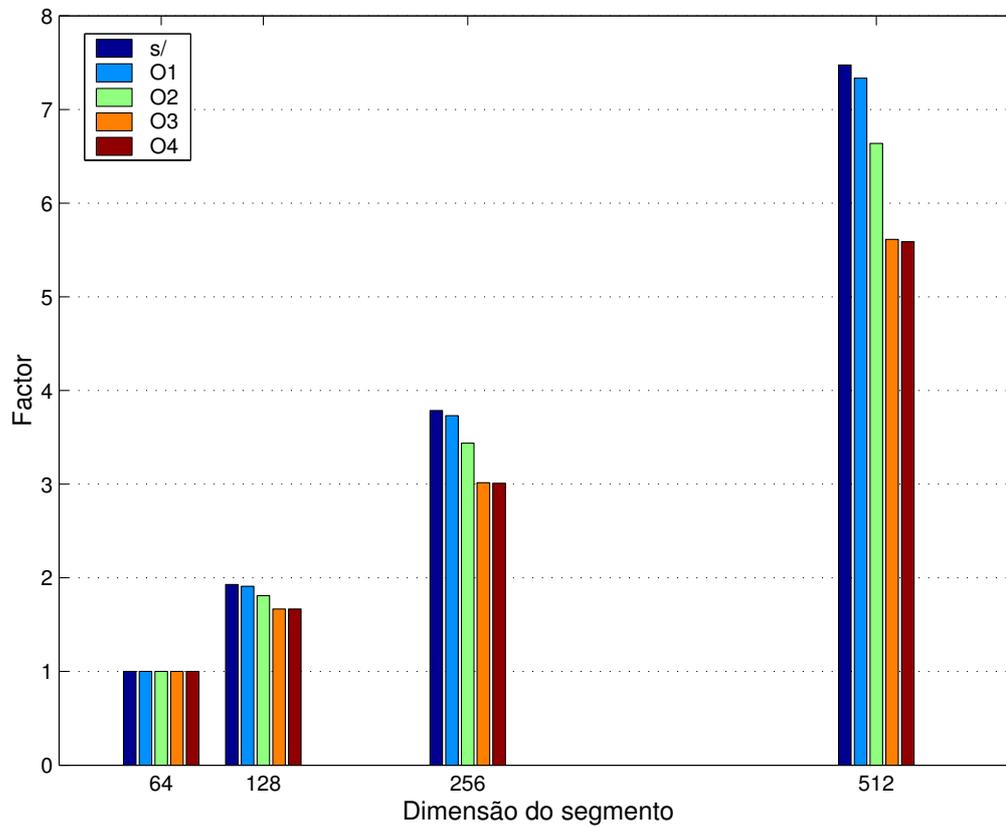


Figura 5.6: Regularidade dos tempos de execução do algoritmo do estimador AR-MC no processador i860.

Essa regularidade é patente na Figura 5.7 onde se apresenta o gradiente de execução do algoritmo no processador i860. O tempo médio de execução apresenta-se estratificado em cada um dos níveis de otimização, sendo os dois últimos bastante idênticos.

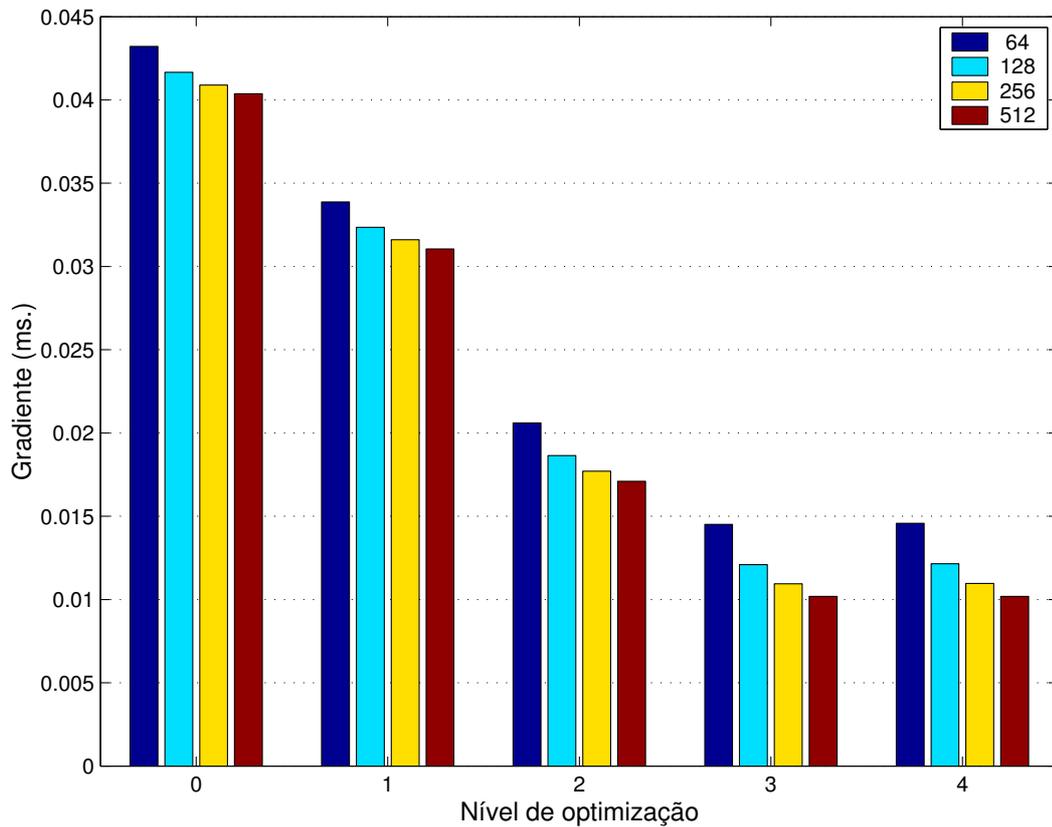


Figura 5.7: Gradiente de execução do algoritmo do estimador AR-MC no processador i860.

A Figura 5.8 apresenta a redução dos tempos de execução conseguida com os diferentes níveis de optimização. É de notar que os níveis de optimização três e quatro produzem resultados equivalentes em termos de tempo de execução.

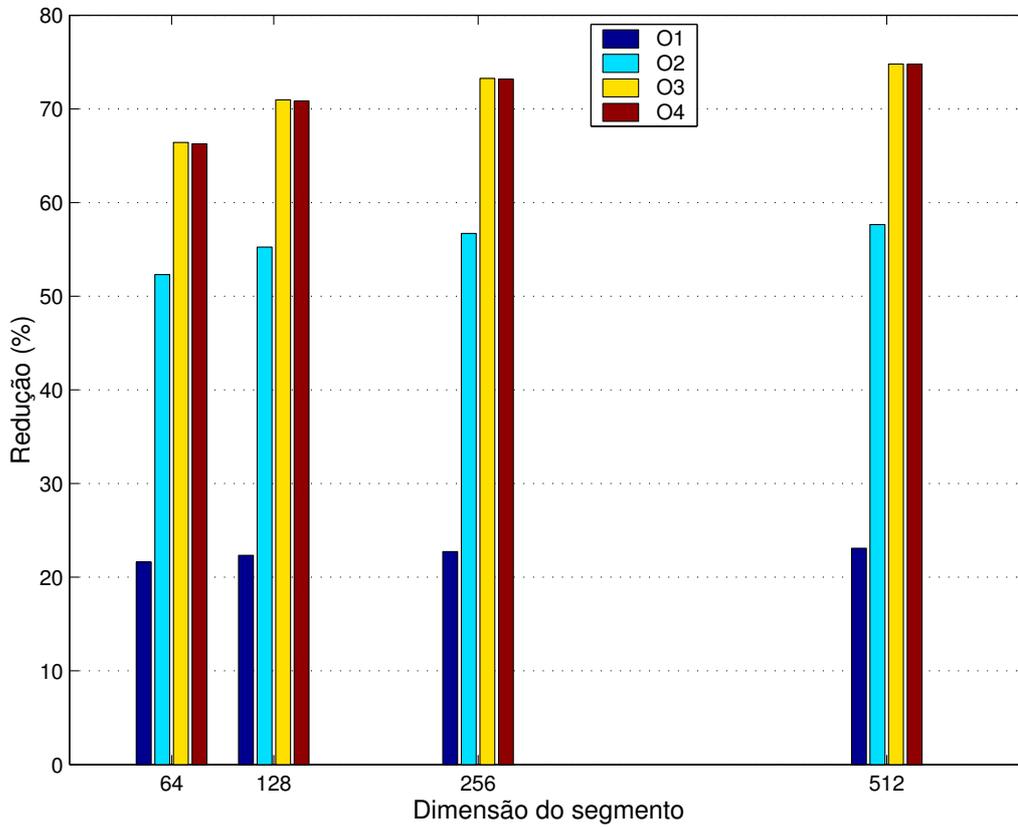


Figura 5.8: Percentagem de redução dos tempos de execução conseguida utilizando as opções de optimização do compilador para o estimador AR-MC no processador i860.

Esta redução dos tempos de execução pode ser encarada como a aceleração (*speedup*) obtida pela otimização, representada na Figura 5.9.

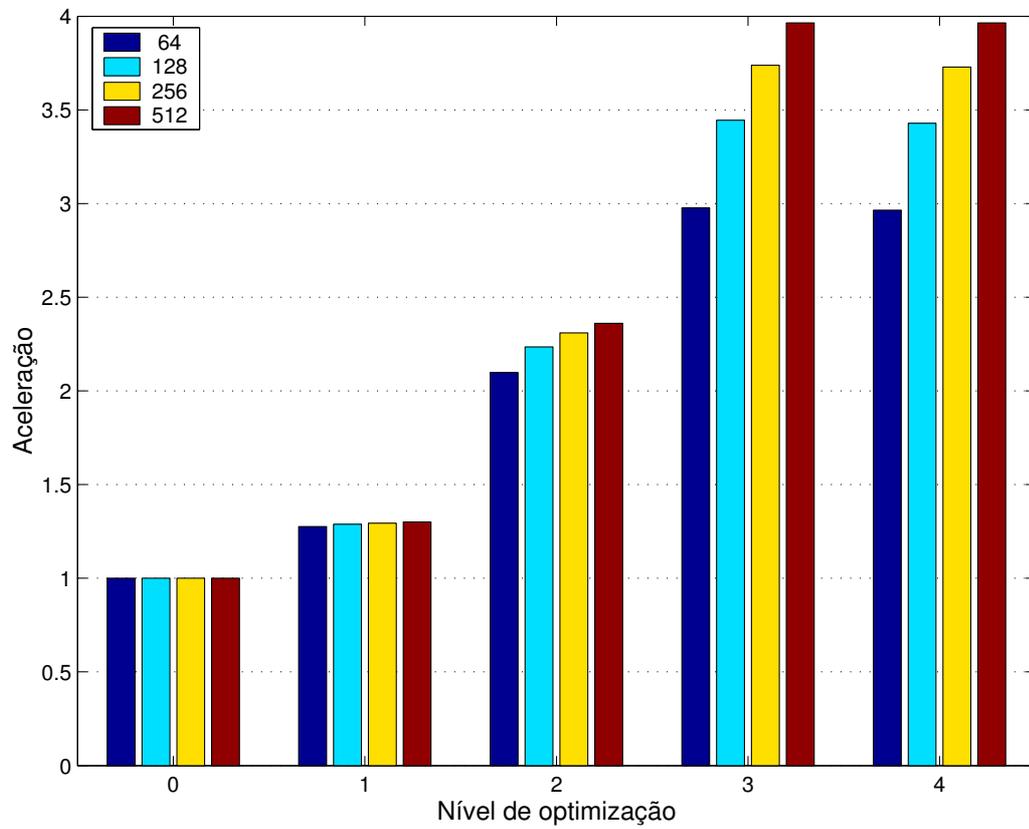


Figura 5.9: Aceleração do algoritmo do estimador AR-MC no processador i860.

5.2.3.2 AR-MC na Arquitectura Heterogénea

Foi avaliada a implementação utilizando os três tipos de processadores da Plataforma Aditiva A.

A aplicação executada pelo i860 é essencialmente idêntica à descrita anteriormente; a segunda aplicação reúne três processos, idênticos entre si, que implementam o estimador AR-MC, processos esses atribuídos aos processadores C40. Esta implementação foi conseguida ligando as três C40s e o i860 com as duas aplicações, que podem ser entendidas como *caixas negras* sendo o encaminhamento de dados atribuído aos *transputers*.

A Tabela 5.3 apresenta os resultados da implementação heterogénea do algoritmo AR-MC. Estes valores correspondem ao cálculo em paralelo de quatro segmentos de dados distribuídos pelas C40s e o i860.

Comprimento do Segmento	i860	C40	T8
64	0,929	3,461	3,748
128	1,548	6,827	7,331
256	2,801	13,54	14,474
512	5,214	26,662	28,461

Tabela 5.3: Tempos (em ms.) da implementação heterogénea do estimador AR-MC na estimação de quatro segmentos de 10 ms..

Os três primeiros segmentos de dados recebidos pelo T8 ligado à máquina hospedeira (processador raiz) são transmitidos para o segundo T8 que, agindo como intermediário, os distribuirá pelas C40s. O quarto segmento de dados é enviado pelo processador raiz ao i860 através do T8 na carta TTM110. É de notar que o uso do segundo *transputer* da Figura 5.1 como intermediário entre o processador raiz e as C40s permite obter uma redução do tempo de execução global.

Os tempos apresentados na Tabela 5.3 são medidos no processador raiz para o processamento efectuado pelo i860 e no T8 intermédio para o processamento efectuado pelas C40s. Os tempos apresentados na coluna T8 foram medidos no processador raiz no intervalo desde o início do envio de dados até à recepção do último resultado, correspondendo assim à comunicação inter-processador e ao cálculo paralelo dos quatro segmentos.

Analisando os tempos de execução do estimador AR-MC na plataforma heterogénea verifica-se que, nesta situação, o tempo médio por segmento é inferior

ao tempo de cada segmento, conforme ilustrado na Figura 5.10.

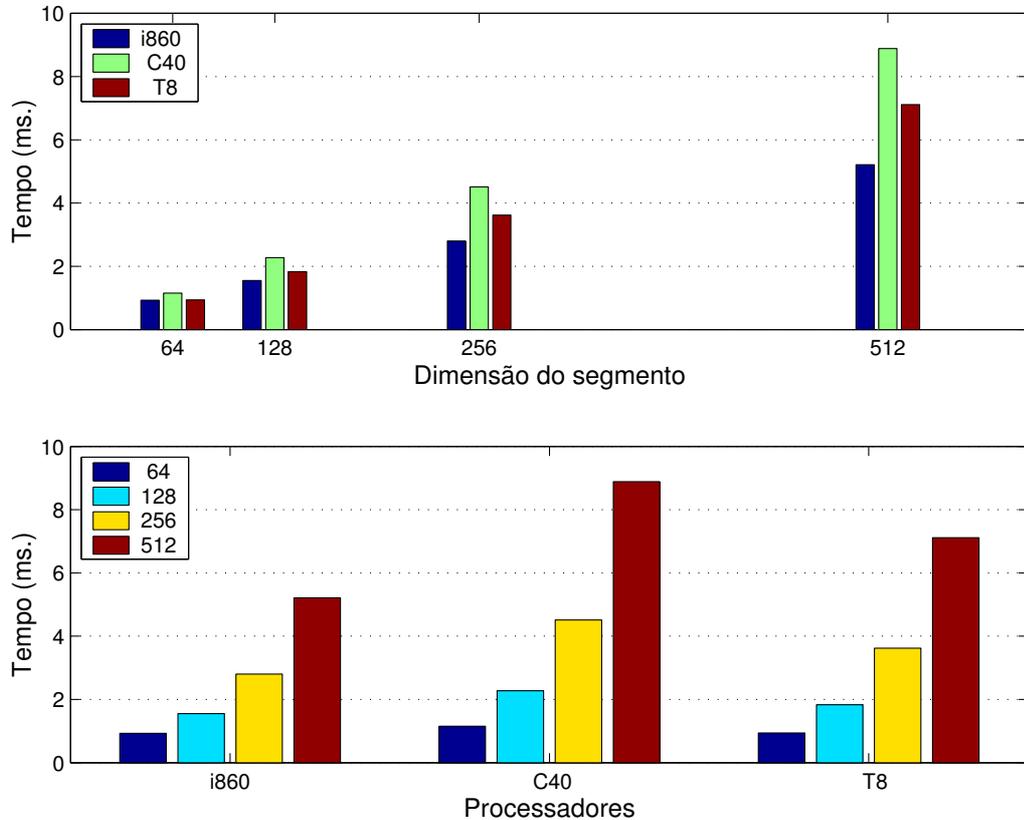


Figura 5.10: Tempos médios por segmento de dados do estimador AR-MC na plataforma heterogênea.

5.2.3.3 Estudo Prévio do CWD

Para estabelecer uma referência para a comparação de resultados de investigações subsequentes, o algoritmo do estimador CWD foi previamente implementado em Matlab (The Math Works Inc., 1991). A carga computacional e complexidade do algoritmo foi avaliada em termos de tempo e número de operações de vírgula flutuante (funções de Matlab "flops" e "cputime", respectivamente).

O núcleo da distribuição é função de um parâmetro σ , específico para a aplicação em causa, e do comprimento da janela temporal aplicada. Tal como para o algoritmo da distribuição na sua totalidade, foi também estudada a carga e complexidade do núcleo. A dupla simetria exibida pelo núcleo permite que se possa reduzir a ocupação de memória, com a contrapartida de cálculo extra para a determinação dos índices dos elementos correspondentes. Acresce ainda que o

cálculo do núcleo pode ser efectuado numa fase inicial de pré-processamento, uma vez que se manterá constante para a mesma dimensão de segmento e parâmetro σ . Assim sendo, o núcleo pode ser armazenado numa matriz de duas dimensões transferida para memória durante o processo de inicialização e, a partir de então, usada como constante.

Esses valores são apresentados na Tabela 5.4 correspondendo à estimação de dados de um ciclo cardíaco de 640 ms. de duração.

Tamanho do segmento dados	Tempo algoritmo (sec)	FLOP algoritmo (*10E6)	Tempo núcleo (sec)	FLOP núcleo (*10E3)
16	8,4	5,4	0,1	1,8
32	14,2	11,2	0,1	3,8
64	26,1	23,1	0,1	7,8
128	49,6	47,1	0,1	15,8
256	95,2	95,7	0,3	31,9
512	189,8	194,1	0,6	64,1

Tabela 5.4: Avaliação em Matlab do algoritmo e núcleo da distribuição CW.

Como pode ser observado na Tabela o tempo de execução ultrapassa largamente o limite de tempo-real. É de notar que, de forma geral, quando o tamanho do segmento dobra, dobram também a complexidade algorítmica e a carga do algoritmo. Este facto revela a regularidade do algoritmo.

Note-se ainda o tempo necessário ao cálculo do núcleo, no pior caso para um segmento de 40 milissegundos, é de aproximadamente seis décimas de segundo. Este cálculo é efectuado apenas uma vez e não para os 380 segmentos considerados no caso presente.

5.2.3.4 CWD no Processador T8

A Plataforma Aditiva A permite a utilização de um a três *transputers*.

Esta implementação considera o cálculo completo da CWD num único T8. A Tabela 5.5 apresenta o tempo total (em ms.) e a dimensão do segmento de dados em memória da implementação do algoritmo CWD num único T8. Esses tempos correspondem à leitura, execução do algoritmo e escrita dos resultados. O tempo de execução total inclui o cálculo de 380 segmentos das diferentes dimensões, correspondentes a um ciclo cardíaco. Segmentos com dimensão superior a 128 elementos não puderam ser considerados apesar de a configuração da aplicação

considerar valores progressivamente maiores para o bloco de dados em memória.

Comprimento do segmento	Tempo total (ms.)	Bloco de dados (Kbytes)
16	37657	100
32	84532	160
64	188170	256
128	412389	448

Tabela 5.5: Tempo total do estimador CWD num único T8.

É de notar que o tempo gasto na estimação é sempre muito superior à duração do sinal que, relembramos é de 640 ms.. Para o segmento de 16 pontos, cujo tempo total é superior a 37 segundos, cerca de 30 segundos são gastos na transferência de dados entre a plataforma aditiva A e a máquina hospedeira. O cálculo da autocorrelação do sinal e o produto pelo núcleo e janela são responsáveis por cerca de 53% do tempo de cálculo, sendo 34% do tempo ocupado no cálculo da forma analítica do sinal e os restantes 13% ocupados no cálculo da densidade espectral de potência.

A regularidade do algoritmo do estimador CWD observável na Tabela 5.4 não se verifica na implementação no processador T8. Como se pode verificar pela Tabela 5.5, quando o tamanho do segmento é duplicado verifica-se um aumento significativo do tempo de execução para além dos valores que se poderiam esperar. Este facto deve-se à gestão de espaço em memória.

Outra implementação utilizando dois *transputers* foi utilizada. Dois segmentos de dados (um em cada processador) foram estimados em paralelo. Apesar de o tempo de execução por segmento de dados diminuir para aproximadamente metade, conforme esperado, os requisitos de tempo-real não foram cumpridos. Esta abordagem de paralelismo de dados em que cada segmento de dados é atribuído a um processador distinto já anteriormente utilizada, tinha também conduzido a um elevado valor de eficiência.

5.2.3.5 CWD no Processador C40

A implementação do estimador CWD no processador C40 considera a utilização de dois T8 na comunicação com a máquina hospedeira, não sendo atribuído aos T8 qualquer processamento. Uma tarefa do sistema designada *filter*, que garante os serviços básicos de comunicação com a máquina hospedeira, foi atribuída ao

primeiro T8; uma tarefa designada *tio*, responsável pela leitura de dados e escrita de resultados, foi atribuída ao segundo *transputer*.

Apenas um processador C40 foi utilizado nesta implementação. Também neste caso foi necessário ajustar a dimensão do segmento de memória reservado aos dados, sendo neste caso possível abranger a gama de janelas pretendida. Nesta implementação foi considerada uma fase de inicialização que contempla o carregamento em memória da janela e do núcleo. A tarefa *dcwd* implementa o estimador.

Era requerido, à data, que cada segmento de dados fosse estimado em tempo inferior ao de amostragem. Na Tabela 5.6 são apresentados os requisitos de tempo-real, o tempo de execução médio por segmento e o espaço da memória de dados atribuído a cada tarefa. Como pode ser observado na Tabela 5.6 os requisitos de tempo-real são cumpridos com apenas um processador pelo que não foi necessário estudar outras arquitecturas.

Requisitos TR (ms.)	Tempo Execução (ms.)	Memória (<i>filter, tio, dcwd</i>)
1,25	0,064	70k, 200k, 200k
2,5	0,139	100k, 256k, 256k
5	0,294	128k, 448, 448k
10	0,618	227k, 448k, 448k
20	1,28	471k, 512k, 448k
40	2,685	471k, 512k, 448k

Tabela 5.6: Implementação do estimador CWD num único C40.

A Tabela 5.7 apresenta em milisegundos a dimensão temporal da janela que segmenta o sinal, o tempo de execução de cada segmento e o valor calculado correspondente à execução para todos os segmentos do sinal, obtido pela multiplicação do tempo por segmento pelo número de segmentos.

Exceptuando o caso da janela de 40 ms., em que a sobreposição é de 96%, esta implementação atinge tempo-real. Se, neste caso, a sobreposição fosse de 92%, o que corresponde ao cálculo de um espectro a cada 3,125 ms., o tempo de cálculo para o sinal de 640 ms. seria de 515,52 ms. correspondendo a uma implementação em tempo-real.

Janela temporal	Tempo por segmento	Tempo para todo o sinal
1,25	0,064	26,163
2,5	0,139	56,712
5	0,294	119,480
10	0,618	249,180
20	1,280	507,900
40	2,685	1031,000

Tabela 5.7: Tempos de execução (em ms.) da implementação do algoritmo da CWD num único processador C40. O tempo para todo o sinal é uma estimativa.

5.2.3.6 CWD no Processador i860

Durante os estudos da implementação no processador i860 verificou-se que o processador parava em operações de multiplicação em que um dos factores era muito pequeno. A execução completa do programa era conseguida apenas se se considerasse um valor mínimo para os factores, a partir do qual os valores deveriam ser explicitamente forçados a zero. Considerando que tal implementação comprometeria a precisão da estimação, o teste do algoritmo no processador i860 foi abolido de futuros trabalhos. Seria necessário adoptar outro algoritmo numérico para a CWD o que não foi conseguido durante o tempo em que a plataforma esteve disponível.

5.2.3.7 Previsibilidade Temporal na Plataforma Aditiva A

Os tempos de cálculo ou execução apresentados anteriormente são tempos médios. Tendo sido observado que os valores dos tempos decorridos em duas execuções consecutivas variavam, por vezes de forma significativa, procedeu-se ao estudo da previsibilidade temporal de cada um dos processadores usados.

Sabendo que a execução de um comando na máquina hospedeira Sun, correndo um sistema operativo Solaris, poderia ser afectado por outras tarefas em execução na máquina e, na impossibilidade de utilizar a máquina em modo mono-utilizador, foi preparada uma aplicação que provocasse 100 vezes o cálculo de um mesmo segmento. Foram recolhidos os tempos de cálculo, medidos no processador em estudo, para cada uma das 100 execuções.

Os tempos de cálculo no processador T8 não apresentavam variações dignas de menção. As Tabelas 5.8 e 5.9 apresentam os tempos mínimo, médio e máximo do cálculo de um segmento nos processadores C40 e i860.

Comprimento do Segmento	Tempos (ms.)		
	mínimo	médio	máximo
64	477	647,5	1107
128	674	747,2	1390
256	961	1146,7	1927
512	1673	1840,3	2621

Tabela 5.8: Variação de tempos de cálculo exibida pelo processador C40.

Comprimento do Segmento	Tempos (ms.)		
	mínimo	médio	máximo
64	880	894,2	932
128	1506	1512,1	1522
256	2729	2748,2	2769
512	5187	5197	5219

Tabela 5.9: Variação de tempos de cálculo exibida pelo processador i860.

Como pode ser observado, a variação dos tempos medidos no processador C40 é de tal maneira elevada que pode comprometer os objectivos temporais.

5.2.3.8 Transferência de Dados na Plataforma Aditiva A

Os tempos de comunicação são relevantes na escolha da arquitectura paralela mais adequada para um algoritmo em particular. Na comunicação entre processadores pode-se distinguir duas etapas: a transferência de dados associada à topologia física da arquitectura e a, por vezes necessária, conversão de dados de/para o formato nativo do processador.

Um T8, mantido como processador raiz, garantiu a comunicação entre a máquina hospedeira e os demais processadores, C40 e i860. A conversão é necessária para dados de tipos *float* ou *double* quando processadores C40 estão envolvidos porque esses processadores representam internamente os dados num formato proprietário do fabricante *Texas Instruments*. A conversão é realizada por funções de biblioteca e depende do compilador usado. Os T8 e C40 são processadores de 32 *bit* e o i860 é um processador de 64 *bit*. A transferência T8 – T8 é série, a T8 – C40 utiliza um adaptador série-paralelo e T8 – i860 na placa TTM100 faz-se por memória partilhada.

O tempo de transferência de dados, independente do compilador utilizado, assume maior relevância. Assim, neste contexto, apresentam-se não tempos de comunicação mas sim de transferência de dados. Os tempos agora considerados

correspondem ao envio e retorno de segmentos de dados de tipo *float* de dimensão variável (64, 128, 256 ou 512 pontos).

Considere-se os tempos correspondentes à transferência de dados entre processadores, apresentada na Tabela 5.10.

Comprimento do Segmento	T8 – T8	T8 – C40	T8 – i860
64	0,997	0,592	0,56
128	1,906	1,155	1,011
256	3,752	2,234	1,872
512	7,409	4,514	3,49

Tabela 5.10: Tempos de transferência (em ms.) de dados na arquitectura heterogénea.

Os tempos de transferência de dados T8 – C40 seriam bastante elevados se se considerasse também a conversão de dados. A comunicação por memória partilhada entre T8–i860 é a mais rápida entre os processadores envolvidos.

O uso das opções de optimização do compilador *pgcc* tem também efeito nos tempos de comunicação registados, representados na Figura 5.11.

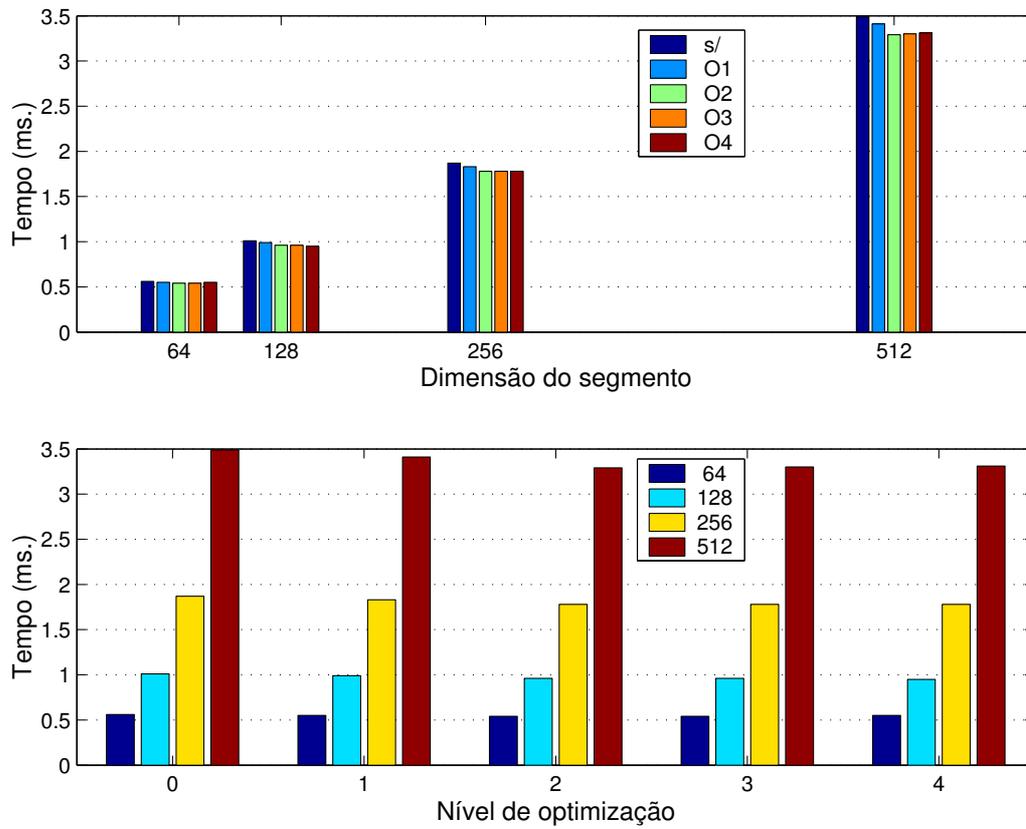


Figura 5.11: Efeito das opções de otimização do compilador *pgcc* nos tempos de comunicação entre o T8-i860.

Os tempos de transferência de dados entre a máquina hospedeira e o processador raiz foram medidos para o estimador AR-MC no âmbito do estudo referido em (Madeira, 1997), sendo aqui apresentados no enquadramento do estudo da Plataforma Aditiva A.

O tempo da transferência de dados utilizados pelo estimador AR-MC nos diferentes casos, segmentos de 10 ms., comprimento variável e sem sobreposição são representados na Figura 5.12.

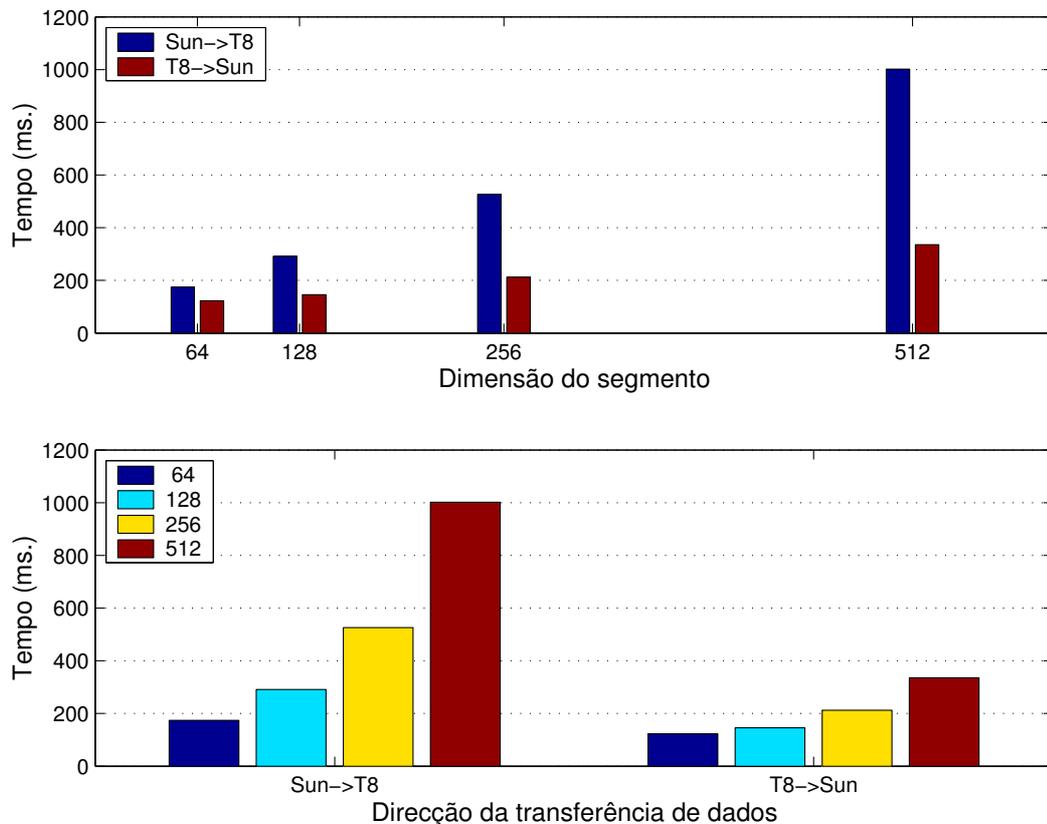


Figura 5.12: Tempos de comunicação dos segmentos de dados utilizados pelo estimador AR-MC entre a máquina hospedeira Sun e o processador raiz.

De notar que os valores do tempo de transferência de dados entre a máquina hospedeira e a Sun ultrapassam largamente os limites de tempo-real para este caso. A diferença exibida entre os tempos de comunicação de/para a Plataforma Aditiva A deve-se ao facto de a partir de um segmento de dimensão N se estimar um espectro de dimensão $N/2$. Abstraindo essa diferença pela análise do gradiente de transferência de dados (Fig. 5.13) reforça-se a conclusão de que a transferência de dados limita a utilização desta plataforma para a implementação

em tempo-real do estimador AR-MC. Note-se na Figura 5.13 que o tempo médio para a transferência de um elemento de dados ronda os dois milisegundos.

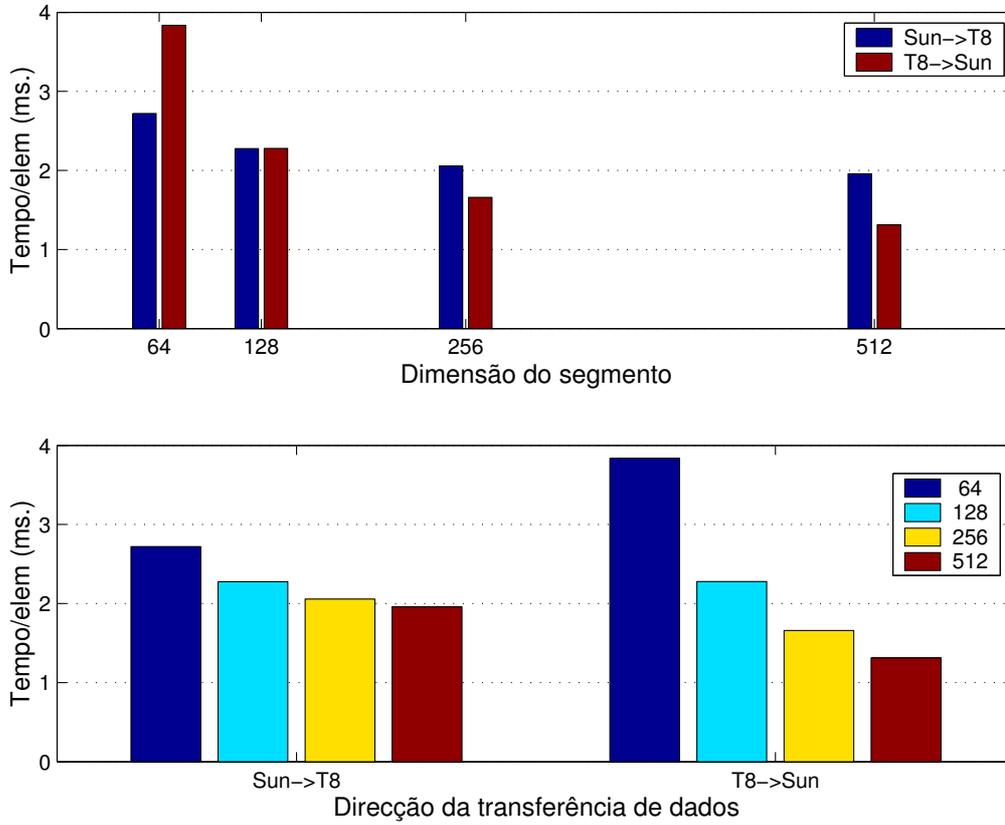


Figura 5.13: Gradiente de comunicação de dados utilizados pelo estimador AR-MC entre a máquina hospedeira Sun e o processador raiz.

Apesar das conclusões do estudo atrás referido, foi também avaliada a transferência de dados com a Plataforma Aditiva A para o caso do estimador CWD em que são utilizados dois *transputers* no encaminhamento dos dados. Os dados utilizados para a avaliação da implementação do estimador CWD representam um ciclo cardíaco de 640 ms., um sinal simulado de 8 *Mbytes*. Este sinal é segmentado por janelas temporais de comprimentos variáveis, podendo ser sobrepostas, originando 380 segmentos a estimar. A Figura 5.14 apresenta o tempo da transferência de dados utilizados pelo estimador CWD entre a máquina hospedeira e a Plataforma Aditiva A neste cenário.

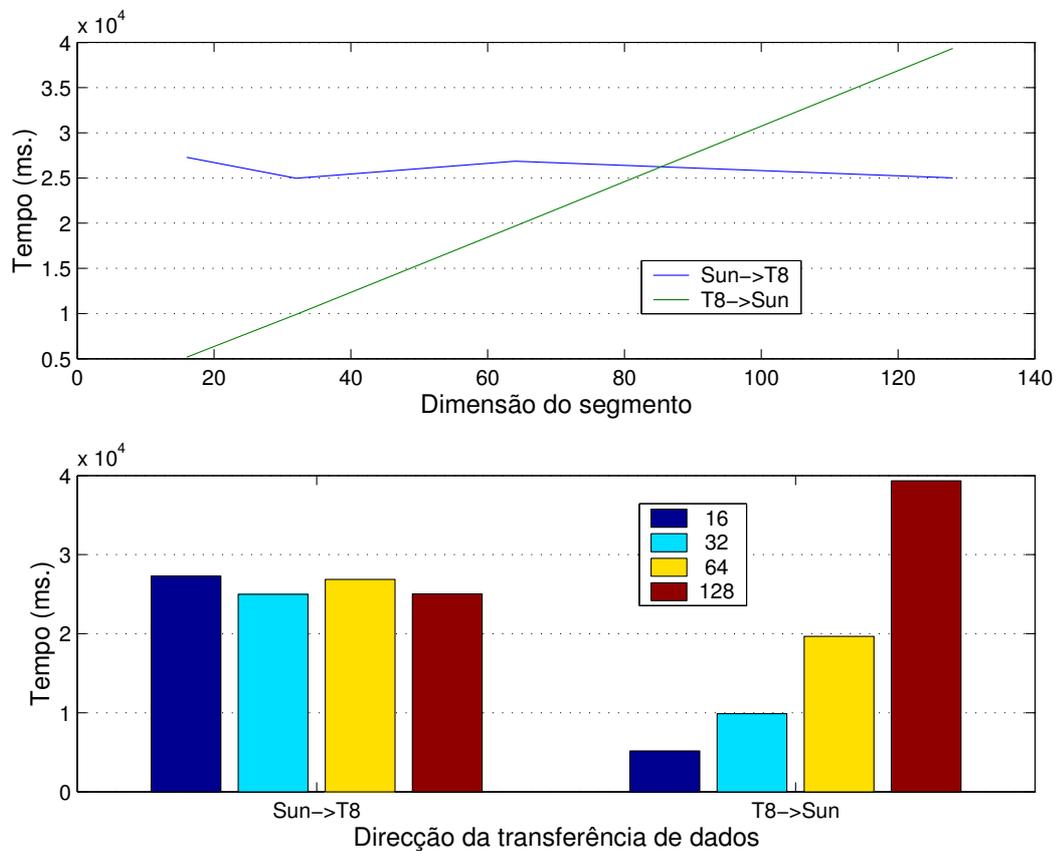


Figura 5.14: Tempos de comunicação dos segmentos de dados utilizados pelo estimador CWD entre a máquina hospedeira Sun e o processador raiz.

O tempo de leitura de dados mantém-se aproximadamente constante, o mesmo não sucedendo na escrita de resultados. De facto, cada elemento de dados é lido apenas uma vez, traduzindo-se a sobreposição de segmentos por deslocação, em memória dos elementos do segmento lidos anteriormente. Como cada segmento lido corresponde a um espectro a ser escrito, verifica-se a regularidade dos tempos

de escrita de resultados face à dimensão do segmento.

Analisando o tempo por segmento de dados (Fig. 5.15) pode-se verificar que, enquanto o tempo de transferência entre a máquina hospedeira e o processador raiz se mantém aproximadamente constante, o mesmo não sucede com a escrita de resultados. O maior segmento testado, correspondente a uma amostra de 10 ms. é lido em pouco mais de 60 ms., valor incompatível com as pretensões de tempo-real do sistema.

O tempo de leitura é constante devido ao facto de, apesar da sobreposição de segmentos, cada elemento de dados só ser lido uma vez. O aumento da dimensão do segmento (e, conseqüentemente, do espectro estimado) provoca aumento do tempo de escrita de resultados.

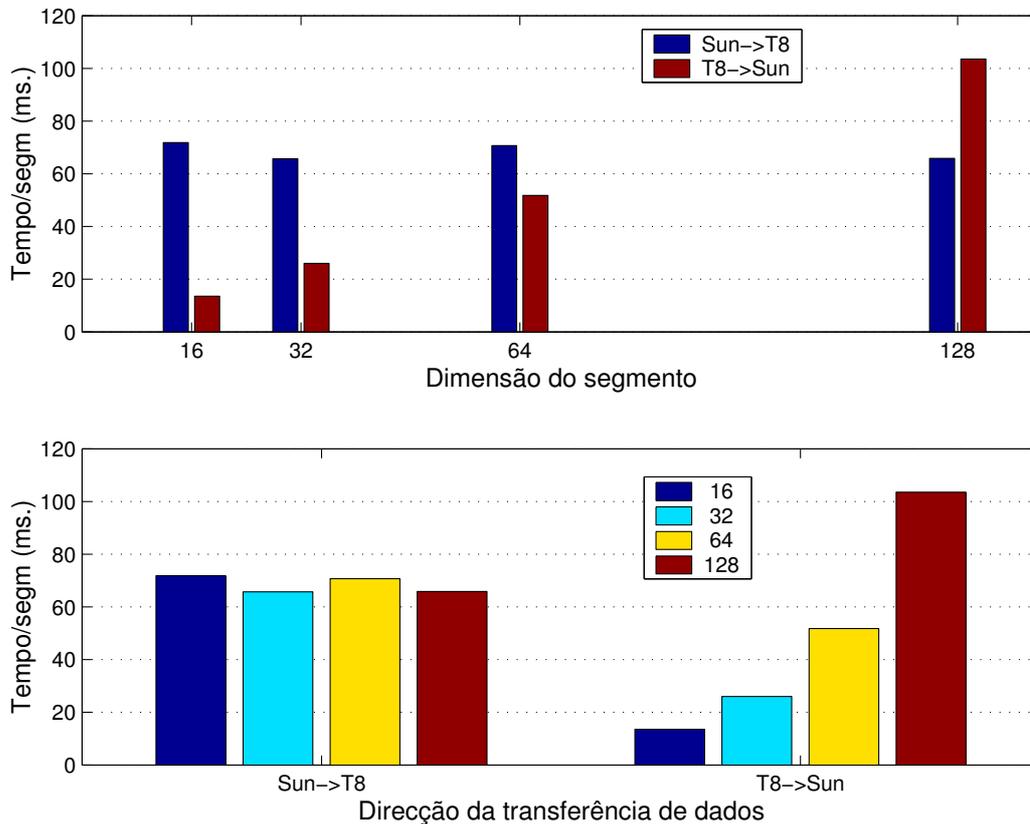


Figura 5.15: Tempos de comunicação, por segmento, dos dados utilizados pelo estimador CWD entre a máquina hospedeira Sun e o processador raiz.

Maior detalhe é obtido na análise do gradiente de comunicação expresso na Figura 5.16, onde se pode notar que a leitura beneficia do aumento do comprimento do segmento, diluindo-se o tempo de estabelecimento da comunicação ¹.

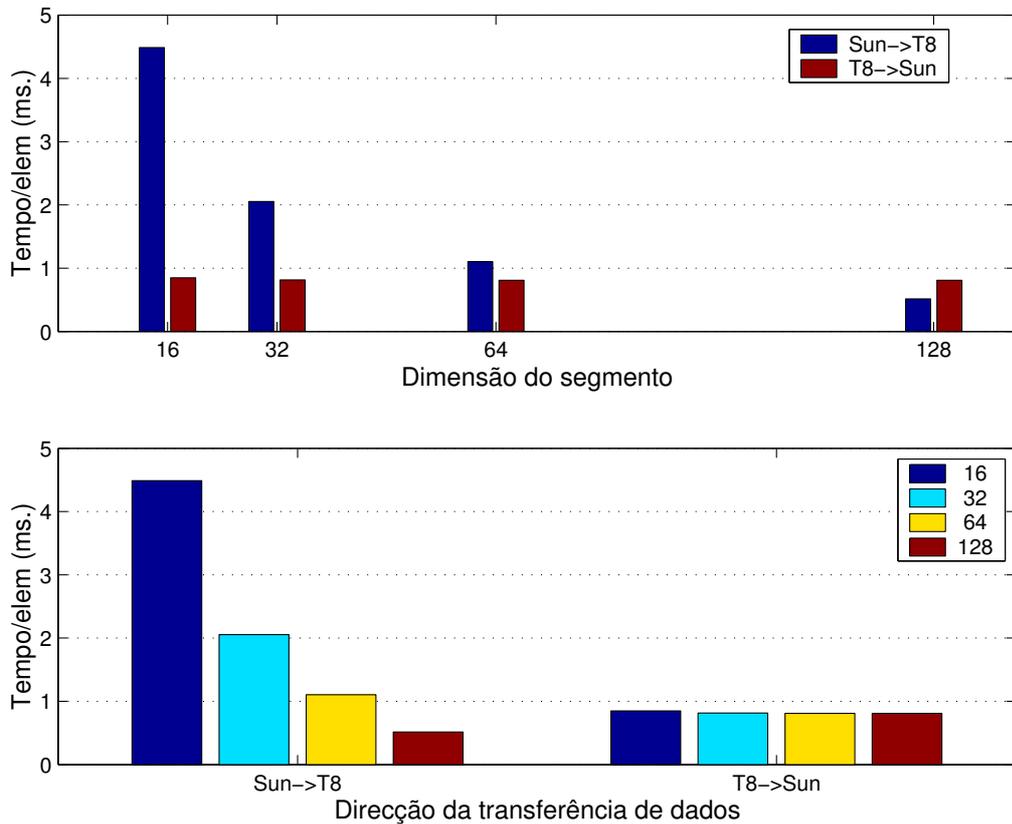


Figura 5.16: Tempos de comunicação, por elemento, dos dados utilizados pelo estimador CWD entre a máquina hospedeira Sun e o processador raiz.

O menor tempo verificado na transferência de dados para o estimador CWD deve-se ao facto de não ser atribuída qualquer outra actividade ao *transputer*.

¹Em inglês, *setup time*

5.2.4 Implementações na Plataforma Aditiva B

O desempenho da plataforma aditiva B, em que o processador C40 está ligado directamente ao *bus* da máquina hospedeira, foi avaliado em implementações do estimador AR-MC (Madeira, Beltran, González, Noceti, Tokhi & Ruano, 1998) e em relação à transferência de dados.

5.2.4.1 AR-MC

Numa primeira fase, o algoritmo do estimador AR-MC anteriormente implementado na Plataforma Aditiva A foi avaliado na Plataforma B.

Considerou-se a divisão do algoritmo nos quatro blocos funcionais sendo observados os tempos de cada um dos blocos – cálculo da matriz real de covariância modificada (RMC), resolução do sistema de equações lineares (Cholesky), cálculo da variância do ruído branco (WNV) e da densidade espectral de potência (PSD) – e do algoritmo integral, i.é, não segmentado.

Esses valores são apresentados na Tabela 5.11 para um modelo de quarta ordem. Note-se que a resolução do relógio é um milissegundo.

Comprimento do Segmento	Tempos (ms.)				
	RMC	Cholesky	WNV	PSD	integral
64	1	0	3	5	9
128	2	0	4	6	12
256	4	0	5	6	15
512	9	1	6	8	23

Tabela 5.11: Tempos de execução do estimador AR-MC (quarta ordem) no processador C40.

Estes valores foram comparados com os obtidos na Plataforma Aditiva A (Madeira, Beltran, González, Noceti, Tokhi & Ruano, 1998) confirmando-se que a utilização de *transputers* no encaminhamento de dados para a plataforma C40 degradava o desempenho global da arquitectura.

Como resultado de estudos para melhorar a adequação do algoritmo ao processador utilizado verificou-se uma melhoria significativa pela diminuição de chamadas a funções e pelo cálculo prévio de factores (*twiddle factors*) utilizados no cálculo da densidade espectral de potência. Os valores obtidos são apresentados na Figura 5.17. Foram consideradas ordens do modelo de 4, 6, 8 e 10, possíveis numa aplicação geral do estimador.

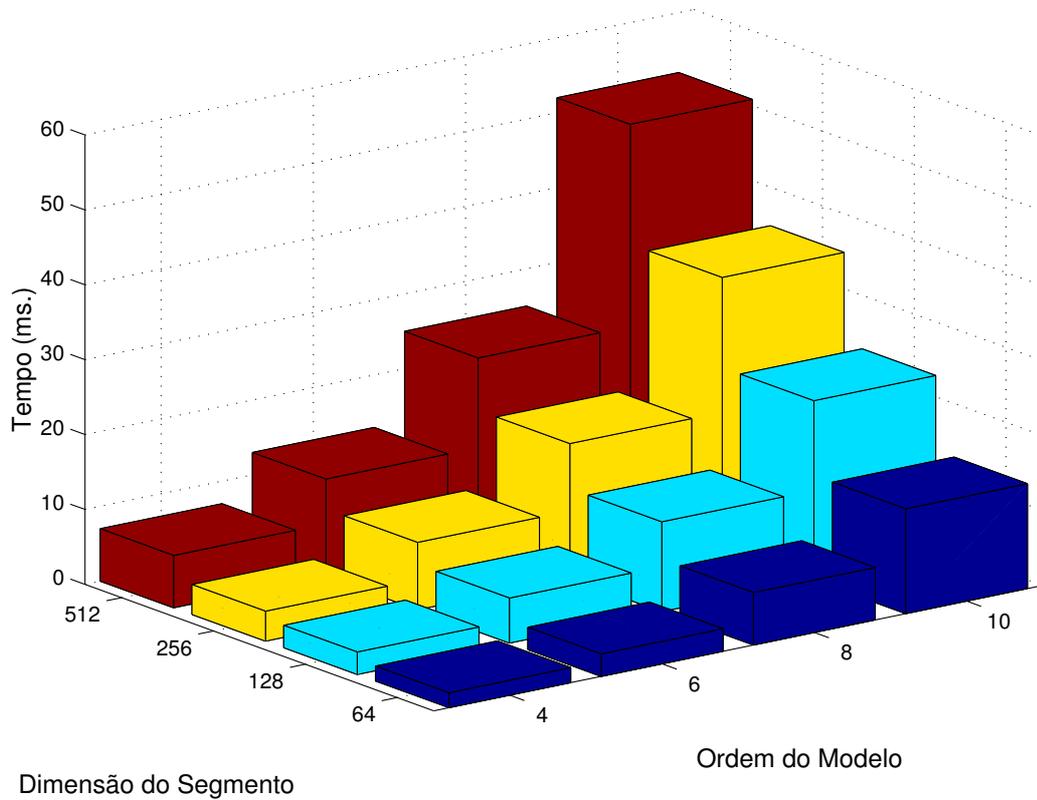


Figura 5.17: Implementação melhorada do estimador AR-MC na Plataforma Aditiva B.

Neste caso, para um modelo de quarta ordem, em que a janela é sempre de 10 ms. e não se prevê sobreposição dos segmentos, a implementação respeita os requisitos de tempo-real então definidos.

É aparente uma melhor resposta do algoritmo ao aumento da dimensão do segmento que ao aumento da ordem do modelo, o que motivou maior detalhe na análise do algoritmo.

Analisando o comportamento do algoritmo em termos de regularidade (Fig. 5.18), confirma-se que a resposta provocada pelo aumento de dimensão do segmento é mais previsível que a provocada pelo aumento de ordem do modelo.

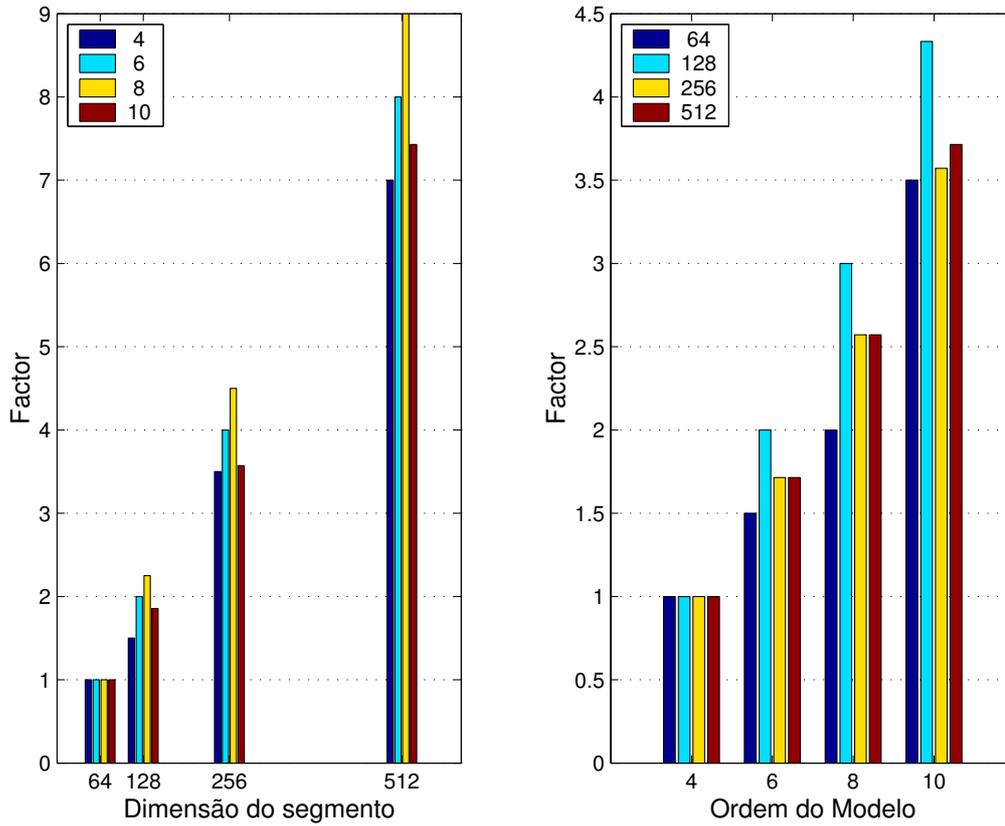


Figura 5.18: Regularidade da implementação do estimador AR-MC na Plataforma Aditiva B.

Foram ainda observados os tempos de execução de cada um dos quatro blocos funcionais considerados, tempos esses apresentados na Figura 5.19.

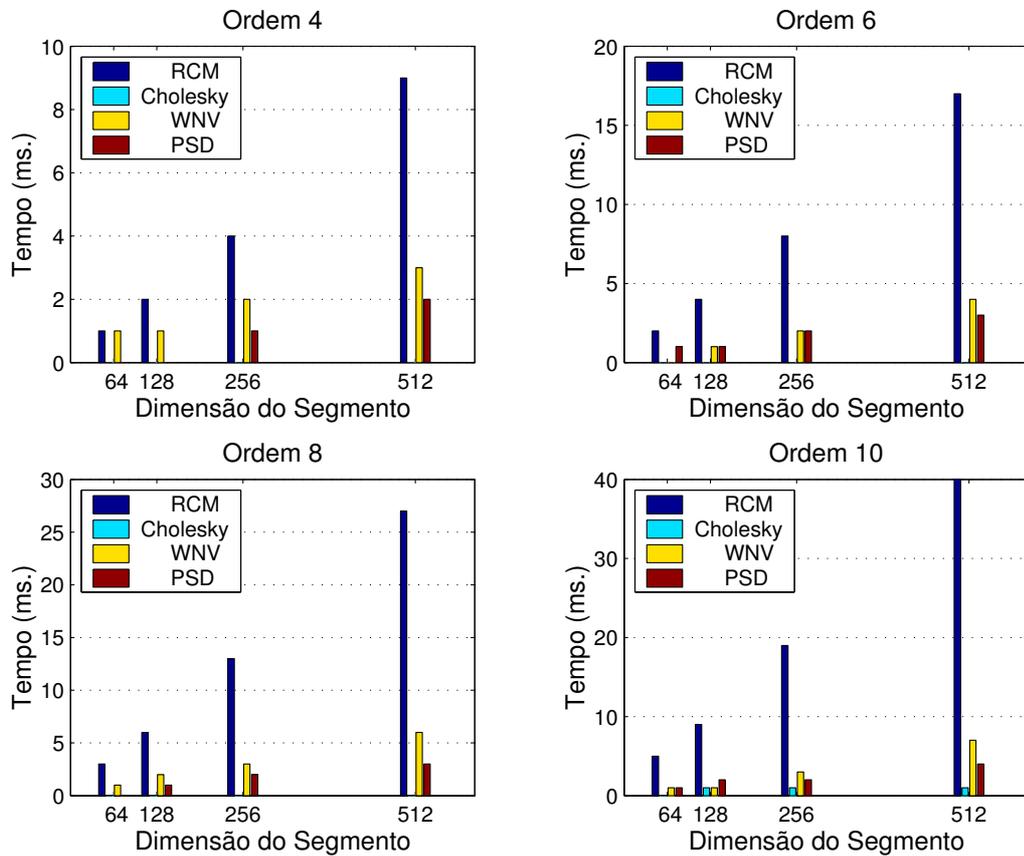


Figura 5.19: Detalhe da implementação melhorada do estimador AR-MC na Plataforma Aditiva B.

O peso do módulo de cálculo da matriz real de covariância modificada (RMC) na execução do algoritmo é flagrante na Figura 5.19. O módulo de cálculo da variância do ruído branco (WNV) é o segundo módulo mais pesado sendo a carga da resolução do sistema de equações lineares pelo método de Cholesky muito reduzida.

A relação entre a resolução do relógio e o tempo dos blocos funcionais impede maior detalhe na análise dos mesmos, excepto para o bloco RMC. A regularidade deste bloco, aquele que mais contribui para a carga do algoritmo, é representada na Figura 5.20 sendo notável a semelhança com a Figura 5.18.

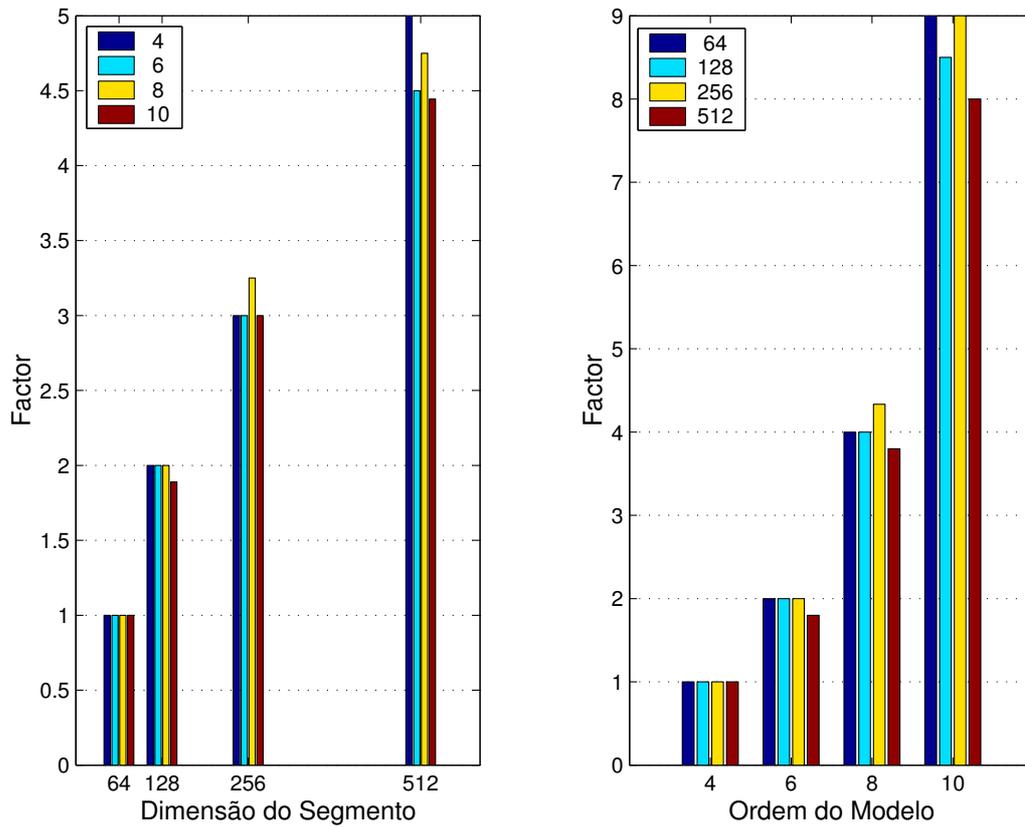


Figura 5.20: Regularidade da implementação do bloco RMC do estimador AR-MC na Plataforma Aditiva B.

5.2.4.2 Tempos de Transferência de Dados

Considerando que a transferência de dados entre a máquina hospedeira e a plataforma aditiva pode estrangular o desempenho global da arquitectura numa aplicação tempo-real, esse aspecto foi estudado. Para o efeito, foram medidos os tempos de leitura e escrita de dados, não incluindo operações de abertura e fecho de ficheiros. Foram usados ficheiros com 32, 64, 128, 256 e 512 pontos.

A Tabela 5.12 apresenta os tempos de transferência de dados entre a máquina hospedeira, um Pentium Pro a 200 MHz, e a DSP C40.

Comprimento do Segmento	Tempo de leitura (ms.)	Tempo de escrita (ms.)
32	13	4
64	27	7
128	53	15
256	135	30
512	209	65

Tabela 5.12: Tempos de transferência de dados medidos na plataforma aditiva B.

Considerando que, para o estimador AR-MC, em que 10 ms. usando frequências de amostragem de 6,4, 12,8, 25,6 e 51,2 kHz, correspondem segmentos de entrada de 64, 128, 256 e 512 pontos e espectros de 32, 64, 128 e 256 pontos respectivamente; e para o estimador CWD, sendo considerada uma frequência de amostragem fixa de 12,8 kHz e a dimensão dos segmentos de entrada igual à dos espectros produzidos, 64, 128, 256 e 512 pontos correspondem a 5, 10, 20 e 40 ms., respectivamente; foi calculado o tempo correspondente a operações de leitura e escrita, apresentado na Tabela 5.13.

Comprimento do Segmento	AR-MC (ms.)	CWD (ms.)
64	31	34
128	60	68
256	150	165
512	239	274

Tabela 5.13: Tempos calculados de transferência de dados na plataforma aditiva B para os estimadores AR-MC e CWD.

Em qualquer dos cenários traçados, os tempos de transferência de dados impedem a efectiva utilização desta plataforma para o processamento em tempo-real dos estimadores de fluxo sanguíneo.

5.2.5 Implementação na Plataforma Aditiva C

O algoritmo do estimador AR-MC implementado na plataforma aditiva A foi implementado num processador Sharc. Este estudo foi possível através da colaboração da equipa do IIMAS (UNAM, México) uma vez que o processador não existia, à data, na Universidade do Algarve. Tal como foi referido sobre a implementação do estimador AR-MC na plataforma Aditiva B, procurou-se uma melhor adequação do algoritmo à arquitectura Sharc. O algoritmo foi melhorado pela eliminação de funções, pelo cálculo prévio dos factores a utilizar no cálculo da densidade espectral de potência e ainda, pela atribuição das variáveis a diferentes blocos de memória de forma a usufruir do duplo *bus* interno. Assim, foi também conseguida a implementação do estimador AR-MC respeitando os requisitos de tempo-real - para cada segmento sem sobreposição - para os casos testados (Madeira, Bellis, Beltran, González, Noceti, Marnane, Tokhi & Ruano, 1999).

Comprimento do Segmento	Ordem			
	4	6	8	10
64	0,5316	0,8929	1,3324	1,8438
128	1,0532	1,7874	2,6956	3,7717
256	2,0964	3,5761	5,4220	7,6277
512	4,1828	7,1537	10,8748	15,3398

Tabela 5.14: Tempos de execução (em ms.) do estimador AR-MC na plataforma aditiva C.

O comportamento do algoritmo foi analisado em termos de regularidade (Fig. 5.21), confirmando-se que, também neste caso, a resposta provocada pelo aumento de dimensão do segmento é mais previsível que a provocada pelo aumento de ordem do modelo.

Comparando a regularidade da implementação do estimador AR-MC nas plataformas aditivas B e C, pode notar-se que, em média, o processador C40 reage melhor ao aumento de dimensão do segmento que o processador Sharc, reagindo melhor este último ao aumento da ordem do modelo.

Em termos gerais, a execução no processador Sharc requer um tempo inferior a um terço do registado com o processador C40.

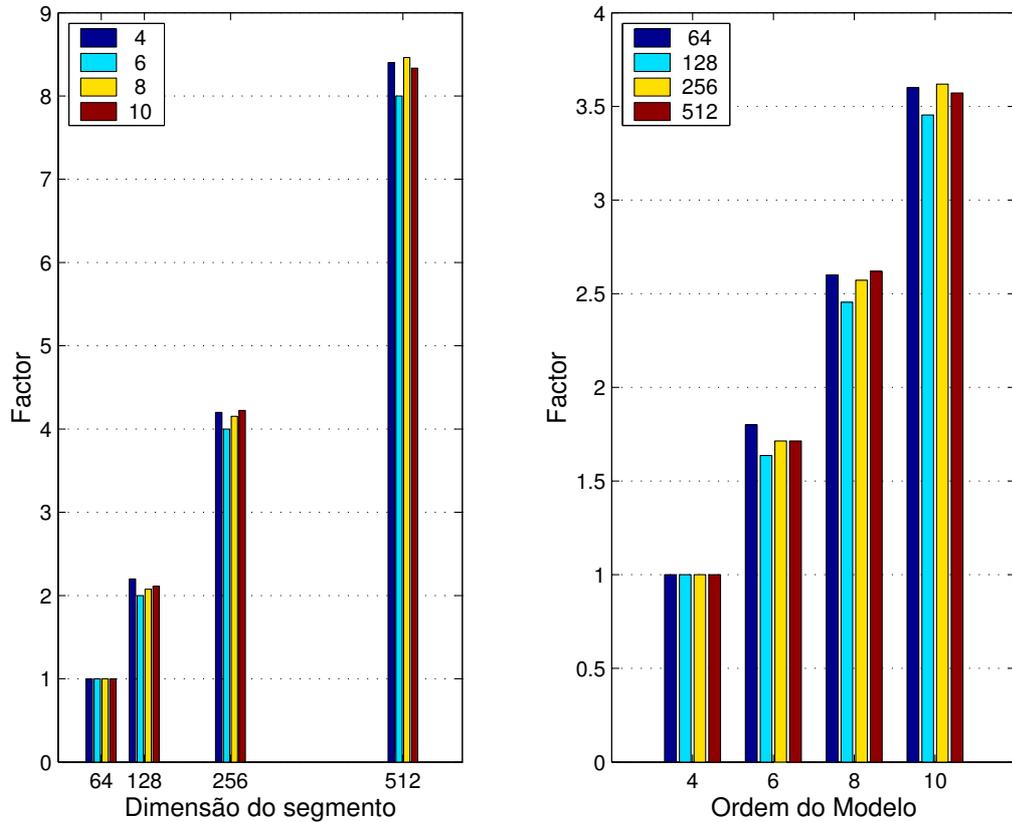


Figura 5.21: Regularidade da implementação do estimador AR-MC na Plataforma Aditiva C.

5.2.6 Outra Implementação

Em colaboração com uma equipa de investigação do *Department of Electrical Engineering and Microelectronics* da *University College Cork, Cork, Ireland*, foi também avaliada a possibilidade de implementar o estimador AR-MC utilizando uma solução mista integrando um processador C40 e *hardware* configurável, *Field Programmable Gate Arrays* (FPGA) (Madeira, Bellis, Marnane & Ruano, 1998).

Uma abordagem modular à implementação do estimador AR-MC usando arquiteturas sistólicas tinha também sido seguida por essa equipa, sendo especialmente relevantes os estudos incidindo sobre os cálculos da matriz real de Covariância Modificada (Bellis, Marnane, Wilde & Fish, 1994) e dos parâmetros do modelo utilizando o algoritmo de Cholesky (Bellis, Marnane & Fish, 1997). Tendo sido demonstrada (Bellis, Fish & Marnane, 1997) a possibilidade de implementar o estimador AR-MC em *hardware* reconfigurável, foi estudada a possibilidade de criação de um sistema misto em que cada um dos módulos fosse implementado

utilizando a tecnologia mais vantajosa.

O cálculo do módulo RMC e dos elementos de covariância necessários ao cálculo da variância do ruído branco (cf. Eq. 2.9) são atribuídos à FPGA sendo o restante processamento atribuído ao processador C40.

A síntese das abordagens estudadas para o estimador AR-MC (Madeira, Bellis, Beltran, González, Nocetti, Marnane, Tokhi & Ruano, 1999) permite identificar duas soluções que possibilitam a estimação de um segmento em tempo inferior ao de amostragem: usando processadores Sharc ou um misto de *hardware* reconfigurável e *software*.

5.3 Conclusão

Os tempos de transferência de dados e resultados na plataforma A mostram que esta não deve ser utilizada num sistema de processamento em que se pretende tempo-real. Dos processadores que a integram: os T8 não devem ser considerados devido aos elevados tempos de execução; as C40 apresentam, para uma mesma tarefa, tempos de execução variáveis; o processador i860 não permite uma implementação fiável do DCWD.

Considerando que a variação dos tempos de execução da C40 na plataforma aditiva A é da ordem dos 10-17% poderia ainda ser utilizada se, integrada num *host* em que os tempos de comunicação fossem inferiores, o tempo de execução no pior caso fosse inferior ao limite imposto pelo requisito de tempo-real. No entanto essa situação não se verifica para a implementação do estimador AR-MC na plataforma aditiva B. Acresce que os tempos de transferência de dados entre a máquina hospedeira e o processador digital de sinal utilizado se situa para além dos limites de tempo-real.

Pelo atrás exposto, atendendo à evolução tecnológica desde o início deste trabalho (em 1998), à obsolescência programada dos sistemas, ao custo, em *hardware* e *software*, de uma solução que contemple a utilização de plataformas aditivas comerciais e visando manter a actualidade e interesse do sistema desenvolvido houve que repensar a estratégia de escolha de arquitecturas. O cenário e justificação da mudança de estratégia serão abordados no capítulo 7. As decisões tomadas têm um grande impacto no subsistema de processamento espectral. A redefinição deste subsistema e o conseqüente desenvolvimento são abordados no próximo capítulo.

Capítulo 6

Subsistema B de Estimação Espectral

6.1 Introdução

Em determinada altura, era flagrante que o tempo de execução obtido com as plataformas aditivas disponíveis era superior ao obtido com processadores correntes de utilização genérica. Foi então considerada a utilização de plataformas não aditivas pelo desempenho superior dos processadores genéricos devido à sua actualidade tecnológica aliada ao baixo custo da solução.

Também a formalização e as condições de aplicação dos algoritmos mudaram ao longo do tempo como resultado da incorporação da investigação em processamento de sinal e a aplicação dos protótipos desenvolvidos. O processamento atribuído às plataformas aditivas consistia na estimação de um segmento de dados. Para as plataformas genéricas considera-se a estimação consecutiva de um sinal discreto. A aplicação genérica dos métodos obriga também a considerar uma carga variável para os algoritmos, uma vez que a parametrização dos métodos depende do sinal em estudo.

Tendo sido avaliada a utilização de plataformas aditivas como forma de aumentar a capacidade de processamento de um computador e privilegiando o baixo custo e actualidade do sistema desenvolvido, foi estudada a implementação do subsistema de estimação espectral utilizando processadores genéricos e *software* livre.

A adopção de processadores genéricos e do sistema operativo Linux não encerra o estudo da plataforma mais adequada ao processamento em tempo-real de

sinais ultra-sónicos. Para além dos diferentes tipos de processadores disponíveis, é ainda necessário definir o modelo de programação mais conveniente. Nesse sentido, poder-se-ia optar entre programação sequencial, programação concorrente, utilização de memória partilhada e utilização de multiprocessadores simétricos, recorrendo sempre ao compilador *gcc* no sistema operativo Debian/GNU Linux.

No Cap. 8 apresenta-se o enquadramento temporal e a fundamentação de algumas decisões estratégicas tomadas em prol da flexibilidade e versatilidade do sistema que determinaram o desenvolvimento abaixo descrito.

6.2 Formas Alternativas dos Estimadores

A aplicação dos métodos de estimação a outros sinais para além dos de fluxo sanguíneo na artéria carótida comum obriga à implementação genérica dos métodos para uma execução parametrizável. Esta exigência é reforçada pelo objectivo de suportar a investigação em processamento de sinal, nomeadamente a configuração dos métodos de estimação. Como exemplo refira-se o estudo efectuado por (Leiria, 2000) para a determinação das condições de aplicação dos estimadores a sinais de fluxo sanguíneo da válvula aórtica. Considerando um sinal a estimar com características distintas, é necessário determinar quais os parâmetros óptimos. Este processo baseia-se em comparar a qualidade da estimação variando as resoluções no tempo e na frequência, a taxa de sobreposição entre segmentos, o tipo de janela, e os parâmetros dos métodos. Em casos em que no sinal possa existir informação direccional do fluxo sanguíneo, poderá ser interessante estudar se é vantajoso separar as duas componentes do fluxo.

A especificação dos métodos, anteriormente baseada nos valores óptimos determinados para o caso da artéria carótida comum, foi alterada a partir deste ponto do estudo. Os métodos de estimação devem poder ser aplicados a sinais de tipo real ou complexo, o que implica que algumas das simplificações utilizadas em implementações anteriores deixem de ser válidas. As janelas temporais a considerar incluem, de momento, as de tipo rectangular ou Boxcar, Hanning, Hamming ou Bartlett. O tamanho das janelas, que anteriormente se considerava como igual a uma potência de dois e variando tipicamente entre 64 e 512, deve ser um qualquer número inteiro. A sobreposição das janelas é livre, sendo indicado o número de pontos entre duas janelas consecutivas. O número de pólos do estimador AR-MC, anteriormente considerado igual a quatro, deve ser um

qualquer número inteiro. A implementação da CWD que anteriormente considerava apenas o núcleo standard deve aplicar um dos quatro tipos de núcleo; os parâmetros do núcleo σ e M anteriormente fixos, são livres e de tipos real e inteiro, respectivamente. De notar que M só surge nos núcleos de tipo *standard* e *alias free*.

Em resumo, devem ser considerados diferentes tipos e tamanhos de janelas, com valores de sobreposição variáveis, e os parâmetros dos métodos devem ser livres. O estudo do modelo de programação a utilizar baseia-se neste novo conjunto de especificações.

Nos casos que a seguir se estudam foi considerado um sinal clínico de artéria carótida com a duração de 33,61 segundos, obtidos com uma frequência de amostragem de 11025 Hz, cada elemento com 16 bit, gravados num único canal de um ficheiro de tipo *wav*. A compilação dos algoritmos utilizando *gcc* foi feita com o nível de optimização 3.

Consoante a taxa de sobreposição utilizada, os 370550 elementos traduzir-se-ão num número variável de segmentos e, conseqüentemente, num também variável número de pontos processados. As Tabelas 6.1 e 6.2 apresentam a carga, respectivamente em número de segmentos e número de pontos, a que os algoritmos são sujeitos considerando as dimensões de segmentos de 64, 128, 256 e 512 pontos e taxas de sobreposição de 0, 25, 50 e 75%.

Dimensão segmento	0%	25%	50%	75%
64	5788	7717	11576	23152
128	2893	3858	5787	11574
256	1446	1928	2892	5785
512	722	963	1445	2890

Tabela 6.1: Carga (em número de segmentos) variando a taxa de sobreposição de segmentos.

6.2.1 Impacto da Representação Numérica

O tipo de representação numérica adoptada para os sinais de entrada tem um impacto significativo no desempenho das aplicações. Para além do volume de memória ocupado, a complexidade algorítmica dos métodos também varia.

Na Tabela 6.3 é apresentado o desempenho de uma das implementações da AR-MC de quarta ordem, para os casos de segmentos reais ou complexos. Foi

Dimensão segmento	0%	25%	50%	75%
64	370432	493888	740864	1481728
128	370304	493824	740736	1481472
256	370176	493568	740352	1480960
512	369664	493056	739840	1479680

Tabela 6.2: Carga (em número de pontos) variando a taxa de sobreposição de segmentos.

variado o tamanho das janelas rectangulares e não foi considerada sobreposição dos segmentos. Os valores apresentados correspondem ao tempo da estimação dos segmentos utilizando um algoritmo que traduz a formulação matemática do método codificado em C. Estes tempos foram medidos na plataforma *arquimedes*,

Dados reais	Dados complexos	Dimensão do segmento
4,220	10,647	64
4,158	10,470	128
4,140	10,675	256
4,152	10,396	512

Tabela 6.3: Impacto no tempo de execução (em segundos) da representação de dados adoptada no desempenho da AR-MC de quarta ordem.

um Pentium a 133 MHz com 32 Mbyte de RAM, e correspondem à estimação do sinal de teste. A utilização da representação complexa dos dados traduz-se, neste caso, num aumento dos tempos de execução de aproximadamente 150%. Como ponto de partida desta etapa do estudo da implementação do estimador CWD considerou-se o algoritmo descrito em (Cardoso, Fish & Ruano, 1996). Procedeu-se à codificação do algoritmo em C, fez-se variar o tamanho de janelas rectangulares e não foi considerada sobreposição dos segmentos ou utilização do sinal *analítico*. O desempenho do algoritmo, codificado em C, para $M = 30$ e $\sigma = 5$, é apresentado na Tabela 6.4 para segmentos reais ou complexos.

Estes tempos foram medidos na plataforma *arquimedes* e correspondem à estimação do mesmo sinal de teste. A utilização da representação complexa dos dados não tem tanto impacto significativo como para o método AR-MC uma vez que a versão base do método CWD considera aritmética complexa.

Deverá assim considerar-se para o método AR-MC o pior caso em que os sinais são complexos –, sendo cada ponto representado na sua forma cartesiana

Dados reais	Dados complexos	Dimensão do segmento
13,013	13,007	64
16,699	16,683	128
18,810	18,861	256
20,373	20,622	512

Tabela 6.4: Impacto no tempo de execução (em segundos) da representação de dados adoptada no desempenho da CWD ($M = 30$, $\sigma = 5$).

através de uma estrutura contendo as partes real e imaginária; para o método CWD existe margem para simplificação caso os dados sejam reais.

6.3 Implementação Sequencial

Procedeu-se ao estudo das novas implementações dos métodos STFT, AR-MC e CWD, criando, para cada método, um único programa em que a parametrização é conseguida através de alguns dos argumentos de chamada.

Esta opção tem como vantagem ser mais flexível adequando-se ao apoio da investigação em processamento de sinal servindo maior número de casos. A desvantagem esperada é o pior desempenho devido à complexidade de incluir e gerir múltiplos casos e à utilização dinâmica da área de dados.

6.3.1 STFT

Existem numerosos estudos sobre a implementação sequencial ou paralela da transformada discreta de Fourier em diferentes arquitecturas. A STFT considerada um método padrão em muitas áreas de aplicação de processamento de sinal é-o também no processamento de sinais de fluxo sanguíneo. Para além de servir como termo de comparação, a transformada discreta de Fourier é usada pelos demais métodos de estimação em estudo. Acresce ainda que o impacto de algumas decisões do desenho dos métodos ser mais visível quando observadas neste método.

Nas anteriores implementações tinha-se optado por um algoritmo da transformada discreta de Fourier proposto por (Kay, 1988), usado nas diferentes implementações do método CWD. No caso presente, recorreu-se à biblioteca *Fastest Fourier Transform in the West* (FFTW) (Frigo & Johnson, 1998), cuja utilização

é transparente ainda que o número de pontos não seja igual a uma potência de dois.

Na biblioteca da FFTW, a adequação do algoritmo da transformada discreta de Fourier à arquitectura do sistema em causa é conseguido de duas formas: pela compilação optimizada da biblioteca na arquitectura e pela utilização de um plano que contém a indicação de qual é o melhor algoritmo para determinada dimensão. A geração desse plano, a pedra de toque do desempenho conseguido, é conseguida pela estimação ou medição dos tempos de execução de algoritmos alternativos. Os diferentes planos, cada um referente a uma dimensão da transformada, são designados por sabedoria¹. A biblioteca contém um mecanismo que permite gerar, armazenar e recuperar a sabedoria.

Tendo-se verificado que a utilização da FFTW sem a medição do plano, apenas com a sua estimação, era sub-ótima e que a geração de um plano em todos os casos implicava um acréscimo temporal que penalizava fortemente o desempenho de uma implementação, foi incorporado nas implementações um mecanismo de utilização da sabedoria. Essa sabedoria é criada por um programa cuja execução pode ser efectuada quando conveniente.

Alguns dos argumentos de chamada do programa sequencial da STFT determinam o tipo de janela a aplicar, a dimensão da janela, a taxa de sobreposição de janelas e o tipo de sinal *analítico* a considerar.

6.3.1.1 Comparação entre os Algoritmos Sequenciais da STFT

Como as implementações sequenciais da STFT produzem estimações bastante rápidas, utilizou-se o computador mais antigo, *arquimedes*, um Pentium a 133 MHz com 32 Mbyte de RAM, de forma a obter maior distinção nos valores temporais medidos.

Na Tabela 6.5 são relatados os tempos de execução (em segundos) do algoritmo sequencial da STFT, sobre o sinal de 33,61 segundos, medidos na plataforma *arquimedes*, usando o algoritmo proposto por (Kay, 1988) e usando a biblioteca FFTW, sem ou com o mecanismo de sabedoria, para diferentes janelas, não sobrepostas, de dimensão igual a uma potência de dois.

É interessante reparar que, com o uso da FFTW aliado à sabedoria, o tempo de execução do sinal pouco varia, sendo o aumento do tempo de execução associado à maior dimensão do segmento compensado pelo menor número de segmentos a

¹No original inglês, *wisdom*

Dimensão da janela	Tempo (Kay)	Tempo (FFTW)	Tempo (FFTW + Wisdom)
64	5,933	4,058	1,684
128	6,507	4,885	1,605
256	7,329	5,726	1,699
512	8,225	6,517	1,787

Tabela 6.5: Tempos de execução (em segundos) dos algoritmos sequenciais da STFT.

calcular.

Na Figura 6.1 são apresentados os tempos de execução do sinal de teste obtidos com os três algoritmos para os segmentos de tamanho 64, 128, 256 e 512, respectivamente, para os quatro valores de sobreposição de janelas.

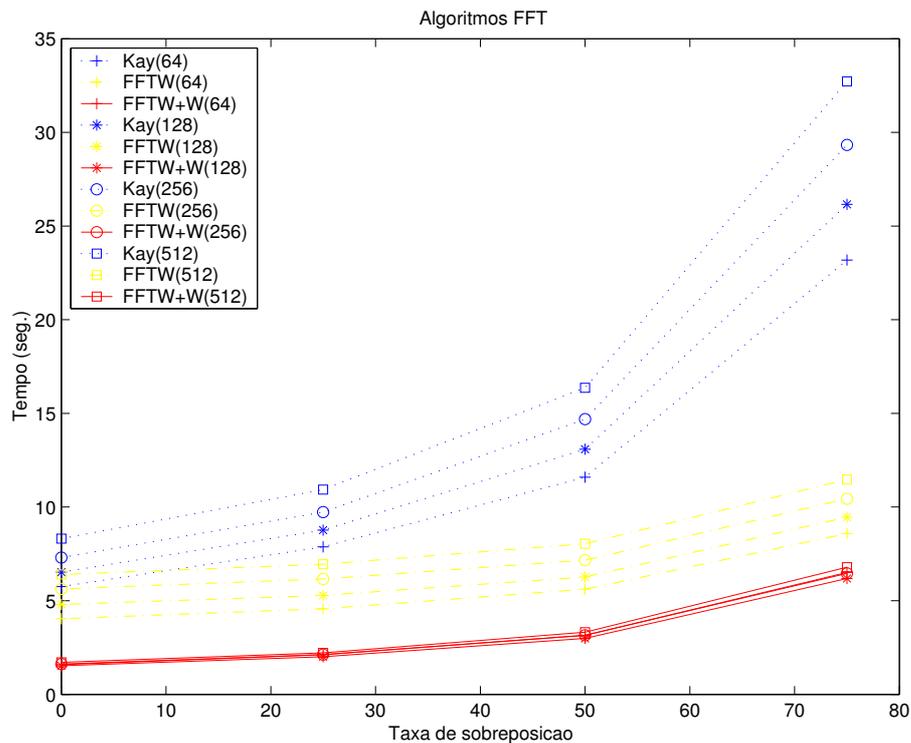


Figura 6.1: Tempos de execução dos algoritmos sequenciais da FFT na máquina *arquimedes*.

Os tempos de execução dos algoritmos reflectem o trabalho adicional resultante da sobreposição dos segmentos. Verifica-se que a implementação segundo a proposta (Kay, 1988) é a mais afectada sendo a implementação recorrendo à FFTW aliada à sabedoria a menos afectada.

6.3.1.2 Algoritmo proposto por Kay

Considerando o algoritmo proposto por (Kay, 1988), os tempos de execução são apresentados na Figura 6.2, no gráfico superior em ordem ao tamanho do segmento e no inferior em ordem à taxa de sobreposição considerada. É de realçar que, neste caso com a máquina *arquimedes*, é sempre respeitado o limite de tempo-real de 33,61 segundos apresentando o pior caso – para segmentos de 512 pontos com 75% de sobreposição – o tempo de execução de 32,719 segundos.

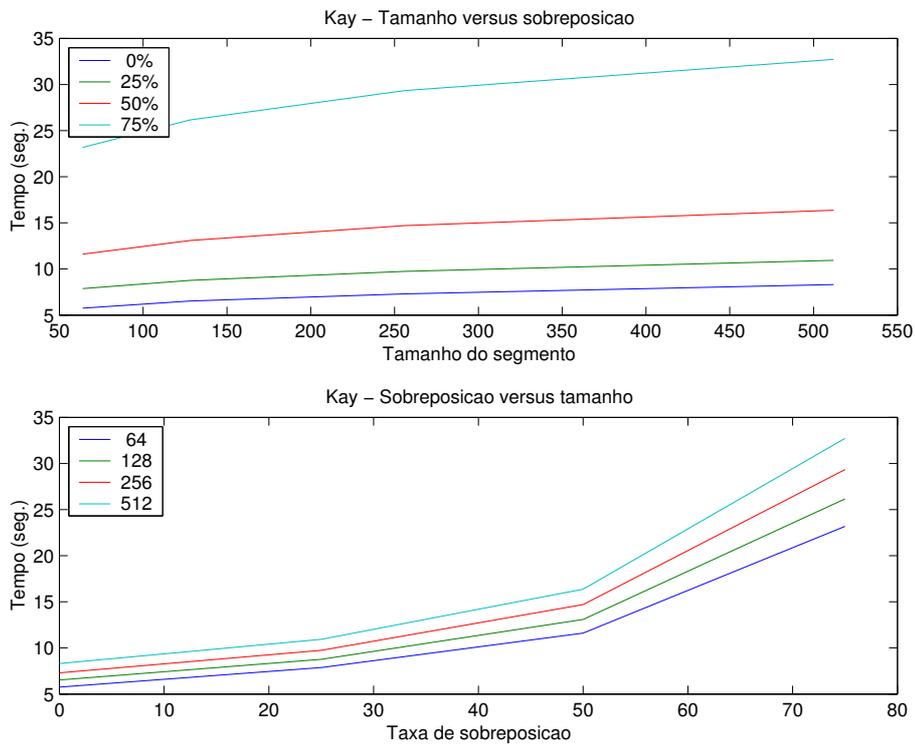


Figura 6.2: Tempos de execução do algoritmo de Kay na máquina *arquimedes*.

A Figura 6.3 apresenta o tempo de execução médio por elemento de dados – também designado por gradiente e correspondendo ao inverso da taxa de processamento de dados (*data throughput*) – do algoritmo de Kay na máquina *arquimedes*, obtido dividindo os tempos de execução pela carga expressa em número de pontos (cf. Tabela 6.2).

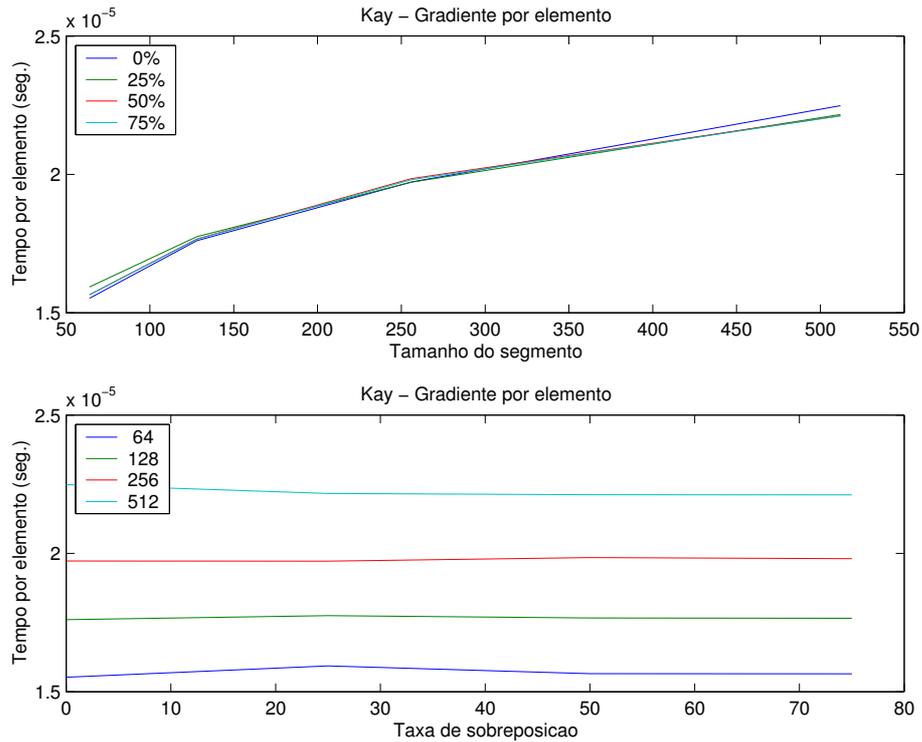


Figura 6.3: Tempos de execução médios por elemento de dados do algoritmo de Kay na máquina *arquimedes*.

Da Figura pode-se concluir a sobreposição dos segmentos não tem impacto no desempenho da implementação enquanto que a maior dimensão do segmento corresponde um tempo médio mais elevado.

A mesma conclusão é evidenciada pela Figura 6.4 onde se apresenta o tempo de execução médio por segmento de dados do algoritmo de Kay na máquina *arquimedes*, obtido dividindo os tempos de execução pela carga expressa em número de segmentos (cf. Tabela 6.1). É clara a linearidade do tempo de execução em função da dimensão do segmento.

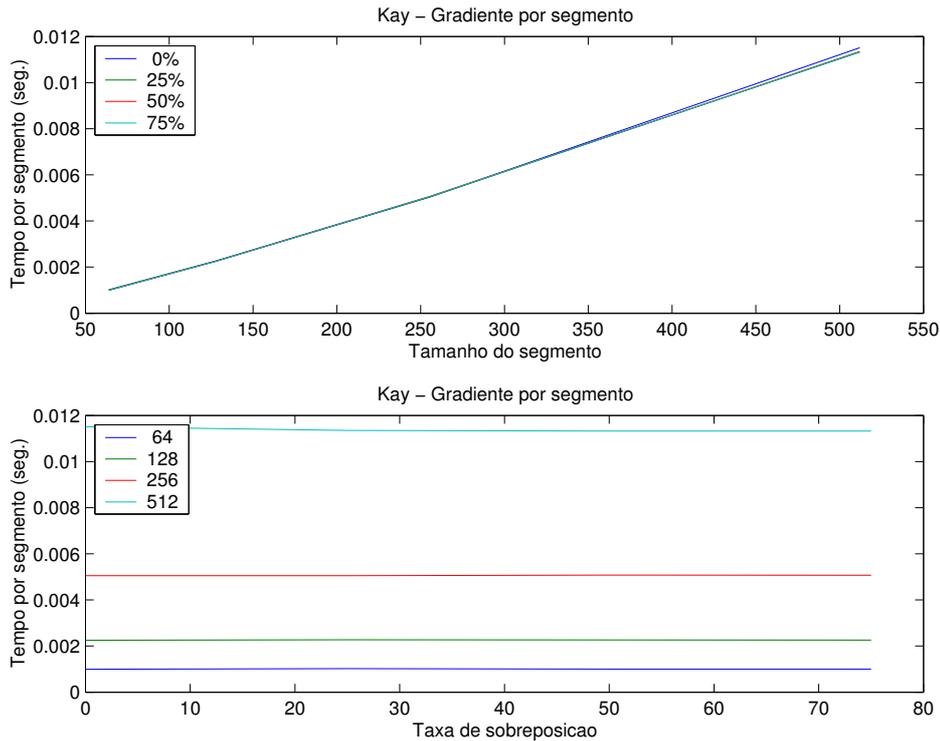


Figura 6.4: Tempos de execução médios por segmento de dados do algoritmo de Kay na máquina *arquimedes*.

6.3.1.3 Algoritmo usando a *FTW* e Sabedoria

Analisando de forma igualmente detalhada o desempenho da implementação, que designaremos por *ftww*, do estimador STFT recorrendo à biblioteca *FTW* e usando o mecanismo de utilização da sabedoria, os tempos de execução são apresentados na Figura 6.5, no gráfico superior em ordem ao tamanho do segmento e no inferior em ordem à taxa de sobreposição considerada.

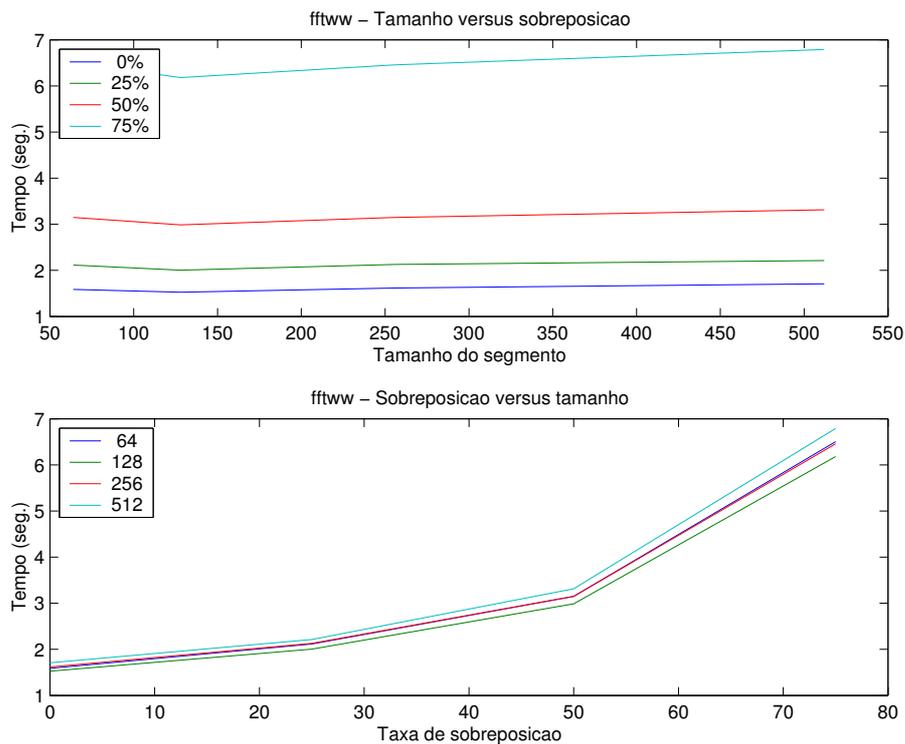


Figura 6.5: Tempos de execução do algoritmo *ftww* na máquina *arquimedes*.

As Figuras 6.6 e Figura 6.7 apresentam, respectivamente, os tempos de execução médios por elemento e segmento de dados do algoritmo *ftww* na máquina *arquimedes*. Os valores representados foram obtidos dividindo os tempos de execução pela carga.

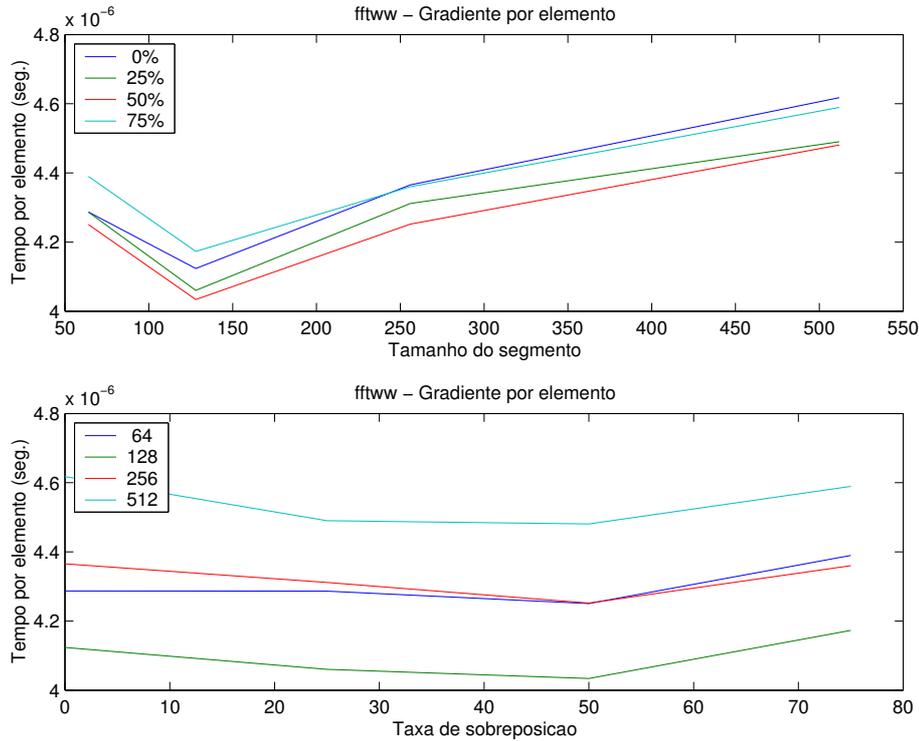


Figura 6.6: Tempos de execução médios por elemento de dados do algoritmo fftww na máquina *arquimedes*.

A Figura 6.6 parece indicar que existiria vantagem em utilizar segmentos de 128 pontos com 50% de sobreposição. Note-se o facto de os valores do gráfico da Figura 6.6 não apresentarem variações superiores a um milissegundo (abaixo da resolução do relógio), pelo que não é possível tirar conclusões. Se assim não fosse, tal facto poderia ser explicado pela adequação do algoritmo da DFT utilizado pela biblioteca no caso particular da máquina *arquimedes*.

Diminuindo o nível de detalhe, analise-se o desempenho médio do algoritmo por segmento de dados.

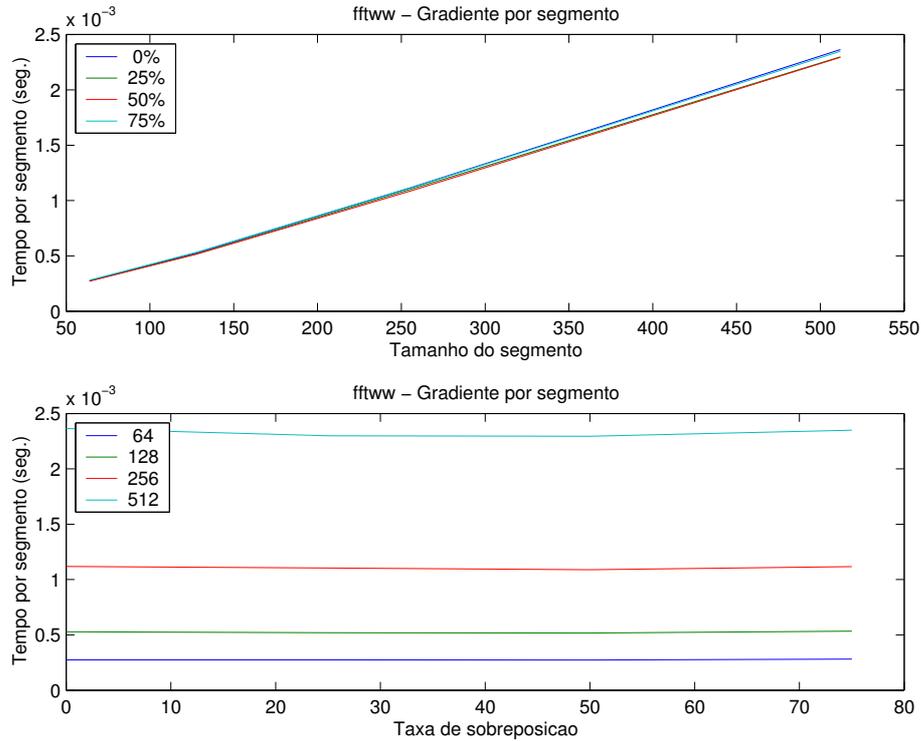


Figura 6.7: Tempos de execução médios por segmento de dados do algoritmo fftw na máquina *arquimedes*.

Da Figura 6.7 pode-se concluir a sobreposição dos segmentos não tem impacto no desempenho da implementação enquanto que a maior dimensão do segmento corresponde um tempo médio mais elevado.

6.3.1.4 Comentários sobre os Algoritmos Sequenciais da STFT

A regularidade esperada da execução dos algoritmos da STFT pôde ser verificada nos diferentes casos. A dimensão do segmento de dados é a variável que mais influi no desempenho dos algoritmos. O impacto da taxa de sobreposição do algoritmo traduz-se na carga computacional do método, que pode ser abstraído pela comparação dos gradientes de execução.

A implementação do estimador STFT usando a biblioteca *FFTW* e utilizando os mecanismos de geração, armazenamento e recuperação da sabedoria apresentou o melhor resultado entre os algoritmos até agora apresentados.

Foram ainda ensaiadas algumas variações das implementações aqui apresentadas, no contexto do estudo do sinal *analítico* e dos demais estimadores.

6.3.2 Sinal *Analítico*

A representação complexa de um sinal real (que se designa por sinal *analítico*) é utilizada na implementação dos estimadores AR-MC, CWD, BD, e na separação das componentes direccionais do fluxo sanguíneo quando tal não é realizado de forma analógica. Tão extensa utilização, bem como a carga computacional que acarreta, motivou um estudo mais aprofundado da técnica e do algoritmo a implementar. Seguindo as diferentes abordagens, no domínio da frequência e do tempo, para a obtenção do sinal *analítico* (revistas na secção 2.3.5), foram estudados os algoritmos de implementação nomeadamente, via DFT, via DHaT ou por convolução com um filtro que aproxima a transformada de Hilbert. Foram implementados vários algoritmos para o cálculo do sinal *analítico* com a mesma frequência do sinal original.

O sinal *analítico* deve ser calculado para cada ciclo cardíaco. Por restrições de tempo-real, foi prática comum aproximar o sinal *analítico*, aplicando a definição a segmentos do sinal (Boashash & Black, 1987). Assim, utilizando 100 sinais correspondentes a 100 ciclos cardíacos de 640 milisegundos cada, considerou-se o cálculo do sinal *analítico* de ciclos cardíacos completos ou de segmentos de 10 ms. com 84,375% de sobreposição (ou seja, a cada 20 pontos é obtido um segmento de 128 pontos). Neste caso, o limite de tempo-real para o cálculo do sinal *analítico* e posterior estimação espectral é de 64 segundos.

No estudo destes algoritmos utilizou-se o computador *coral*, um Pentium III a 500 MHz com 64 *Mbyte* de RAM.

6.3.2.1 Abordagens via DFT

Esta abordagem possibilita o cálculo do sinal *analítico*, recorrendo à transformada de Fourier (Marple Jr., 1999), obtendo um sinal com a mesma frequência do sinal original – standard – ou de diferentes frequências – interpolado ou decimado.

FFTW aliada à sabedoria

Considerando que os melhores resultados na implementação sequencial da STFT foram obtidos com a utilização da FFTW aliada à sabedoria, adoptou-se a mesma abordagem para o cálculo do sinal *analítico*.

Procedeu-se ao cálculo prévio da *sabedoria* utilizada pela FFTW, para transformadas directas e inversas de dimensão compreendida entre a quinta e a décima

quarta potência de dois. O tempo necessário a esta operação é inferior a 17 segundos. O cálculo e armazenamento de *sabedoria* deverá ser efectuado na fase de instalação do sistema ou quando ocorram alterações de hardware ou software. Foram criadas duas aplicações para o cálculo do sinal *analítico*, para o ciclo cardíaco e para segmentos sobrepostos do ciclo cardíaco. O tempo total registado para os 100 sinais, utilizando o comando *time*, é 3,437 s para ciclos completos e 16,640 s para os segmentos sobrepostos. Em qualquer um dos casos, estes valores são inferiores aos 64 segundos, havendo ainda margem para o processo de estimação espectral.

Embora fora do âmbito da comparação que aqui se apresenta, note-se que, no estudo desta abordagem, foram também implementados os algoritmos para o cálculo do sinal *analítico* decimado e interpolado, com um factor $M = 2$.

Cálculo prévio da matriz de transformação

Tinha também sido ensaiada uma forma alternativa de implementação do sinal *analítico*, pelo cálculo prévio da matriz de transformação *CSA*, armazenada num espaço de memória partilhada, a ser acedido pelos estimadores quando necessário.

A definição desta matriz *CSA* segue o procedimento definido por (Marple Jr., 1999), sendo

$$CSA = \frac{1}{N} \times (FT \cdot SA \cdot RT) \quad (6.1)$$

onde N é a dimensão do sinal temporal $x(t)$, FT e RT são matrizes com os coeficientes complexos da transformada de Fourier directa e inversa, respectivamente, definidas por

$$FT_{kl} = e^{-j2\pi kl/N} = \cos\left(\frac{2\pi kl}{N}\right) - j\sin\left(\frac{2\pi kl}{N}\right) \quad (6.2)$$

e

$$RT_{kl} = e^{j2\pi kl/N} = \cos\left(\frac{2\pi kl}{N}\right) + j\sin\left(\frac{2\pi kl}{N}\right) \quad (6.3)$$

onde $k, l \in [0, N - 1]$ e SA_{ii} uma matriz diagonal com os coeficientes de correção da energia espectral tal que,

$$SA_{ii} = \begin{cases} 1 & \text{para } i = 0 \\ 2 & \text{para } 1 \leq i \leq \frac{N}{2} - 1 \\ 1 & \text{para } i = N/2 \\ 0 & \text{para } \frac{N}{2} + 1 \leq i \leq N - 1 \end{cases} \quad (6.4)$$

O tempo de execução, em segundos, da aplicação que calcula e armazena em memória partilhada, a matriz CSA é apresentado na tabela 6.6 para algumas dimensões de segmentos.

Dimensão do segmento	Tempo (s)
64	0,093
128	0,696
256	6,084
512	62,687

Tabela 6.6: Tempos de execução (em segundos) do cálculo e armazenamento em memória partilhada da matriz de transformação do sinal *analítico*.

Sendo sabido que a parte real do sinal *analítico* é o sinal original, só é necessário proceder ao cálculo da parte imaginária do sinal *analítico*. Para o caso de segmentos sobrepostos, de 128 pontos, o tempo acumulado, em segundos, da chamada da função e cálculo do sinal *analítico* recorrendo à matriz de transformação em memória partilhada é de 28 s. Não é possível utilizar esta abordagem para o cálculo do sinal *analítico* para ciclos cardíacos completos devido ao facto de a dimensão do maior segmento de memória partilhada – valor dependente da implementação do sistema operativo – ser actualmente 4 Mbyte.

6.3.2.2 Abordagens via DHaT

Outra forma de obter o sinal *analítico* consiste na aproximação à transformada de Hilbert pela transformada de Hartley (Bracewell, 2000). Tal como para a abordagem anterior foram criadas duas aplicações, uma para o cálculo do sinal *analítico* de ciclos cardíacos, outra para o cálculo do sinal *analítico* de segmentos sobrepostos do ciclo cardíaco. A transformada discreta de Hartley considerou a implementação de Robert Scott (Scott, 2000). O tempo total registado para os 100 sinais, utilizando o comando *time*, é 3,488 s para ciclos completos e 19,284 s para os segmentos sobrepostos.

6.3.2.3 Abordagem por Filtragem no Tempo

Foi ainda ensaiada a convolução com um filtro conforme sugerido em (Boashash & Black, 1987; Picone, Prezas, Hartwell & LoCicero, 1988). Esta opção, ainda que bastante rápida, pode implicar erros significativos no processo de estimação que se segue.

Para maior comodidade de utilização, o cálculo dos N coeficientes do filtro foi implementado separadamente, sendo escrito para um ficheiro. Os tempos da aplicação de cálculo dos filtros são desprezáveis, considerando que serão dispendidos um única vez para cada processo de estimação e inferiores a 10 ms. para dimensões inferiores a 1024 pontos.

Um outro programa é responsável pelo cálculo da convolução que produz a componente imaginária do sinal *analítico* que, com um atraso equivalente a metade da dimensão do filtro, é associado ao sinal temporal original e armazenado.

Foram registados os tempos de execução dos programas que procedem ao cálculo do sinal *analítico* para o ciclo cardíaco completo e para segmentos sobrepostos. A tabela 6.7 apresenta, para os 100 sinais, sendo variada a dimensão do filtro, os tempos de execução, em segundos, para ciclos completos e para segmentos sobrepostos.

Dimensão do filtro	Ciclo cardíaco	Segmentos sobrepostos
79	3,044	4,016
81	3,089	4,100
128	3,738	5,811
256	5,685	11,480
512	9,736	30,005

Tabela 6.7: Tempos de execução (em segundos) para o cálculo do sinal *analítico* por convolução com um filtro.

6.3.2.4 Síntese Comparativa das Abordagens ao Cálculo do Sinal *Analítico*

Para facilitar uma visão global das diferentes abordagens, os valores registados e já relatados foram transportados para a tabela 6.8.

Abordagem	Ciclo cardíaco	Segmentos sobrepostos	Custos adicionais
FFTW	3,437	16,640	17 s
CSA	–	28	0,696
DHaT	3,488	19,284	–
no tempo	3,738	5,811	–

Tabela 6.8: Síntese dos tempos de execução (em segundos) das abordagens ao cálculo do sinal *analítico*.

Analisando os tempos transcritos na tabela, é possível verificar que a utilização

de uma matriz de transformação em memória partilhada é muito dispendiosa em termos computacionais, não sendo possível para o caso de ciclos cardíacos completos. Nas demais abordagens, existe vantagem em calcular o sinal *analítico* de ciclos cardíacos completos. A FFTW e a transformada discreta de Hartley produzem resultados semelhantes e a melhor abordagem seria a convolução com um filtro.

Para a adoção do cálculo do sinal *analítico* no tempo seria necessário proceder a estudos sobre o tipo e dimensão do filtro, que se considerou fora do âmbito deste trabalho.

Assim, a escolha da abordagem repousa entre a FFTW e a DHaT. A DHaT tem como vantagem permitir aritmética real, requer menos recursos mas a sua utilização é restrita ao cálculo do sinal *analítico*. A FFTW é mais flexível, podendo ainda ser utilizada nos métodos de estimação, pelo que é a abordagem adoptada.

6.3.2.5 Utilização do Estimador STFT

A STFT foi também integrada no conjunto dos estimadores disponíveis no sistema de processamento de sinais. Em tempo de execução, os parâmetros de chamada do programa, utilizando a FFTW aliada à sabedoria, permitem definir tipo e dimensão das janelas, sobreposição entre janelas, uso ou não de sinal *analítico* (e em caso afirmativo, de que tipo). As diferentes opções são geridas pela janela de parametrização do método, conforme se apresenta na figura 6.8.



Figura 6.8: Definição das opções de execução do estimador STFT.

6.3.3 AR - Modified Covariance

O estudo do algoritmo computacional do estimador AR-MC foi retomado visando a especificação alargada.

Do trabalho relatado no Capítulo 5 assume especial relevância o facto de os cálculos da densidade espectral de potência e dos elementos da matriz de covariância modificada serem os blocos mais pesados em termos computacionais correspondendo, respectivamente, a 51% e 30% do tempo de execução. Assim, foram estudadas as simplificações do cálculo da matriz de covariância modificada considerando dados de tipo complexo e o criou-se um algoritmo de cálculo da densidade espectral de potência usando a biblioteca *FFTW*.

Para além dos, já referidos, quatro blocos funcionais de aplicação sequencial, considera-se que a segmentação do sinal temporal pela aplicação de janelas, bem como a possível utilização do sinal *analítico* correspondente a cada segmento podem constituir outros dois blocos funcionais. Caso se utilizem janelas Box-car (Brigham, 1988), a sobreposição das janelas utilizadas na segmentação do sinal poderia dar azo a um outro conjunto de simplificações, que não poderão ser usadas se for aplicado o sinal *analítico*.

6.3.3.1 Simplificação do Cálculo da Matriz de Covariância Modificada

A natureza repetitiva dos cálculos necessários à constituição da matriz de covariância modificada (Eq. 2.10) permite algumas simplificações numéricas. Cada elemento da matriz (cf. Eq. 2.11) é obtido pela adição das parcelas $c1_{xx}$ (Eq. 2.12) e $c2_{xx}$ (Eq. 2.13).

Em (Ruano, 1992) foi apresentada a simplificação do cálculo da matriz de covariância modificada considerando que o sinal discreto $x[n]$ é um sinal real. Em tal caso, e para a ordem do modelo $p = P$, os P^2 coeficientes da matriz de covariância e P coeficientes do vector do lado direito da Eq. 2.10 são obtidos a partir de $P + 1$ elementos fulcrais. O facto da matriz de covariância modificada ser Hermitiana e peri-simétrica permite que seja necessário determinar apenas $(P/2 + 1)^2$ elementos pela equação 2.11.

Em tal caso, e para uma ordem do modelo $p = P$, os $(P/2 + 1)^2$ coeficientes $c_{xx}[i, j]$ necessários à constituição do sistema linear da Eq. 2.10 são obtidos a partir de $P + 1$ elementos fulcrais. Os elementos fulcrais contemplam os coeficientes necessários também para o cálculo da variância do ruído branco (Eq. 2.9).

Considere-se, para simplicidade da exposição, que o sinal $x[n]$ é um sinal complexo e $N = 256$ o comprimento do segmento de dados a considerar num modelo de quarta ordem. Os coeficientes c_{xx} das Equações 2.9 e 2.10 podem ser dispostos da forma

$$\begin{bmatrix} c_{xx}[0, 0] & c_{xx}[0, 1] & c_{xx}[0, 2] & \dots & c_{xx}[0, p] \\ c_{xx}[1, 0] & c_{xx}[1, 1] & c_{xx}[1, 2] & \dots & c_{xx}[1, p] \\ c_{xx}[2, 0] & c_{xx}[2, 1] & c_{xx}[2, 2] & \dots & c_{xx}[2, p] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{xx}[p, 0] & c_{xx}[p, 1] & c_{xx}[p, 2] & \dots & c_{xx}[p, p] \end{bmatrix} \quad (6.5)$$

em que cada coeficiente $c_{xx}[i, j]$ é obtido por

$$c_{xx}[i, j] = \frac{1}{2(N-p)} \cdot (c1_{xx}[i, j] + c2_{xx}[i, j]) \quad (6.6)$$

onde

$$c1_{xx}[i, j] = \sum_{n=p}^{N-1} x^*[n-i].x[n-j] \quad (6.7)$$

e

$$c2_{xx}[i, j] = \sum_{n=0}^{N-1-p} x[n+i].x^*[n+j]. \quad (6.8)$$

dando origem a duas matrizes $C1$ e $C2$, de forma semelhante à apresentada na Eq. 6.5.

A matriz de autocorrelação de dimensão $(p+1) \times (p+1)$ é Hermitiana, definida positiva (Kay, 1988) e peri-simétrica (Marple Jr., 1987) o que permite algumas simplificações. Caracterizando para cada coeficiente, a distância da autocorrelação parcial ($i-j$), pode verificar-se a distribuição

$$\begin{bmatrix} 0 & -1 & -2 & -3 & -4 \\ 1 & 0 & -1 & -2 & -3 \\ 2 & 1 & 0 & -1 & -2 \\ 3 & 2 & 1 & 0 & -1 \\ 4 & 3 & 2 & 1 & 0 \end{bmatrix} \quad (6.9)$$

As simplificações são procuradas para os coeficientes com a mesma distância, sendo de esperar que parte da autocorrelação parcial seja comum.

Matriz C1

Distância 0

Tomem-se os coeficientes da matriz $C1$ em que $i = j$, ou seja distância = 0, desenvolvidos de acordo com a Eq. 6.7

$$\begin{aligned}
 c1_{xx}[0, 0] &= \sum_{n=4}^{255} x^*[n].x[n] = \\
 &= x^*[4].x[4] + x^*[5].x[5] + \dots + x^*[254].x[254] + x^*[255].x[255] \\
 c1_{xx}[1, 1] &= \sum_{n=4}^{255} x^*[n-1].x[n-1] = \\
 &= x^*[3].x[3] + x^*[4].x[4] + \dots + x^*[253].x[253] + x^*[254].x[254] \\
 c1_{xx}[2, 2] &= \sum_{n=4}^{255} x^*[n-2].x[n-2] = \\
 &= x^*[2].x[2] + x^*[3].x[3] + \dots + x^*[252].x[252] + x^*[253].x[253] \\
 c1_{xx}[3, 3] &= \sum_{n=4}^{255} x^*[n-3].x[n-3] = \\
 &= x^*[1].x[1] + x^*[2].x[2] + \dots + x^*[251].x[251] + x^*[252].x[252] \\
 c1_{xx}[4, 4] &= \sum_{n=4}^{255} x^*[n-4].x[n-4] = \\
 &= x^*[0].x[0] + x^*[1].x[1] + \dots + x^*[250].x[250] + x^*[251].x[251]
 \end{aligned}$$

Conforme se pode observar na Figura 6.9 onde se representam os coeficientes de autocorrelação parcial de distância zero, denotando cada elemento de dados $x(i)$ pelo seu índice i ,

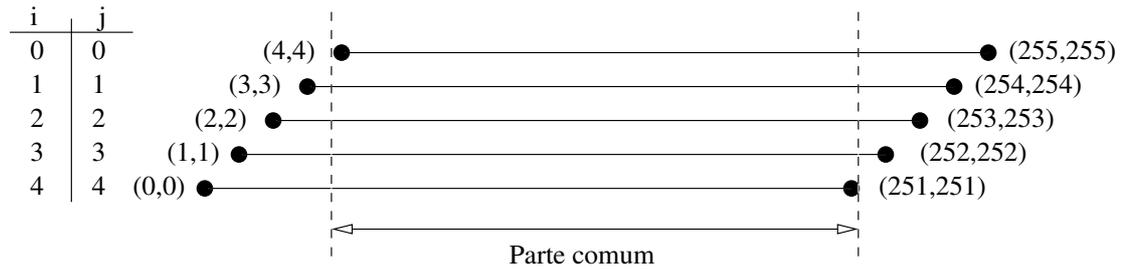


Figura 6.9: Coeficientes de distância zero.

algumas parcelas são comuns. Designando a parte comum por $c1_{xx}^{d=0}$, obtida por

$$c1_{xx}^{d=0} = \sum_{n=p}^{N-1-p} x^*[n].x[n] = \quad (6.10)$$

$$= x^*[4].x[4] + \dots + x^*[250].x[250] + x^*[251].x[251]$$

os cinco coeficientes de distância zero podem ser determinados por

$$\begin{aligned} c1_{xx}[0, 0] &= c1_{xx}^{d=0} + \\ &\quad + x^*[252].x[252] + x^*[253].x[253] + x^*[254].x[254] + x^*[255].x[255] \\ c1_{xx}[1, 1] &= x^*[3].x[3] + c1_{xx}^{d=0} + \\ &\quad + x^*[252].x[252] + x^*[253].x[253] + x^*[254].x[254] \\ c1_{xx}[2, 2] &= x^*[2].x[2] + x^*[3].x[3] + c1_{xx}^{d=0} + x^*[252].x[252] + x^*[253].x[253] \\ c1_{xx}[3, 3] &= x^*[1].x[1] + x^*[2].x[2] + x^*[3].x[3] + c1_{xx}^{d=0} + x^*[252].x[252] \\ c1_{xx}[4, 4] &= x^*[0].x[0] + x^*[1].x[1] + x^*[2].x[2] + x^*[3].x[3] + c1_{xx}^{d=0} \end{aligned} \quad (6.11)$$

Note-se ainda que, para valores reais de $x[n]$, $x^*[k].x[k] = x^2[k]$ e, para valores complexos de $x[n]$ $x^*[k].x[k] = |x[k]|^2 = real(x[n])^2 + imag(x[n])^2$.

Conclui-se que os cinco coeficientes da diagonal principal da matriz $C1$ podem assim ser calculados a partir de $c1_{xx}^{d=0}$, acertando com os outros elementos (de índices $k = 0, 1, 2, 3, 252, 253, 254$ e 255).

Distância 1

De forma idêntica, o desenvolvimento da Eq. 6.7 para os coeficientes da matriz $C1$ em que $i = j + 1$, ou seja distância um,

$$\begin{aligned} c1_{xx}[1, 0] &= \sum_{n=4}^{255} x^*[n-1].x[n] = \\ &= x^*[3].x[4] + x^*[4].x[5] + \dots + x^*[253].x[254] + x^*[254].x[255] \\ c1_{xx}[2, 1] &= \sum_{n=4}^{255} x^*[n-2].x[n-1] = \\ &= x^*[2].x[3] + x^*[3].x[4] + \dots + x^*[252].x[253] + x^*[253].x[254] \\ c1_{xx}[3, 2] &= \sum_{n=4}^{255} x^*[n-3].x[n-2] = \\ &= x^*[1].x[2] + x^*[2].x[3] + \dots + x^*[251].x[252] + x^*[252].x[253] \\ c1_{xx}[4, 3] &= \sum_{n=4}^{255} x^*[n-4].x[n-3] = \\ &= x^*[0].x[1] + x^*[1].x[2] + \dots + x^*[250].x[251] + x^*[251].x[252] \end{aligned}$$

²*real()* e *imag()* extraem, respectivamente, a parte real e imaginária de um valor complexo

é representado na Figura 6.10.

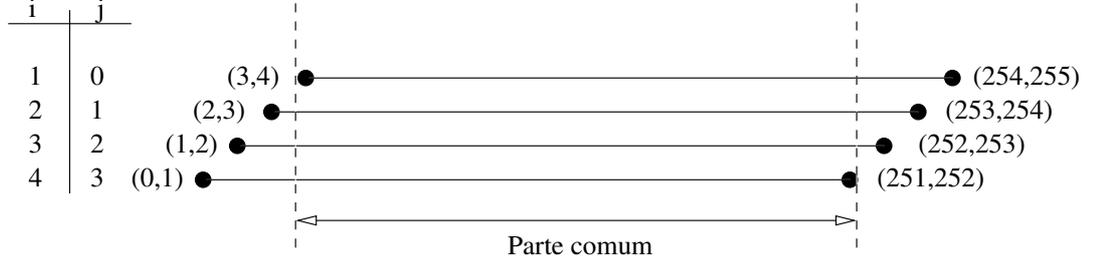


Figura 6.10: Coeficientes de distância um.

Designando a parte comum por $c1_{xx}^{d=1}$, obtida por

$$\begin{aligned} c1_{xx}^{d=1} &= \sum_{n=p}^{N-p} x^*[n-1].x[n] = & (6.12) \\ &= x^*[3].x[4] + x^*[4].x[5] + \dots + x^*[250].x[251] + x^*[251].x[252] \end{aligned}$$

os coeficientes são obtidos por

$$\begin{aligned} c1_{xx}[1, 0] &= c1_{xx}^{d=1} + x^*[252].x[253] + x^*[253].x[254] + x^*[254].x[255] \\ c1_{xx}[2, 1] &= x^*[2].x[3] + c1_{xx}^{d=1} + x^*[252].x[253] + x^*[253].x[254] \\ c1_{xx}[3, 2] &= x^*[1].x[2] + x^*[2].x[3] + c1_{xx}^{d=1} + x^*[252].x[253] \\ c1_{xx}[4, 3] &= x^*[0].x[1] + x^*[1].x[2] + x^*[2].x[3] + c1_{xx}^{d=1} \end{aligned} \quad (6.13)$$

Conclui-se que os quatros coeficientes da diagonal $d = 1$ da matriz $C1$ podem assim ser calculados a partir de $c1_{xx}^{d=1}$, acertando com os outros seis elementos (de índices $k = 0, 1, 2, 252, 253$ e 254).

Distância 2

De forma idêntica, para os elementos da matriz $C1$ em que $i = j + 2$, ou seja distância 2, desenvolvidos de acordo com a Equação 6.7 e representados na Figura 6.11,

a parte comum $c1_{xx}^{d=2}$ pode ser obtida por

$$\begin{aligned} c1_{xx}^{d=2} &= \sum_{n=p}^{N-p+1} x^*[n-2].x[n] = & (6.14) \\ &= x^*[2].x[4] + x^*[3].x[5] + \dots + x^*[250].x[252] + x^*[251].x[253] \end{aligned}$$

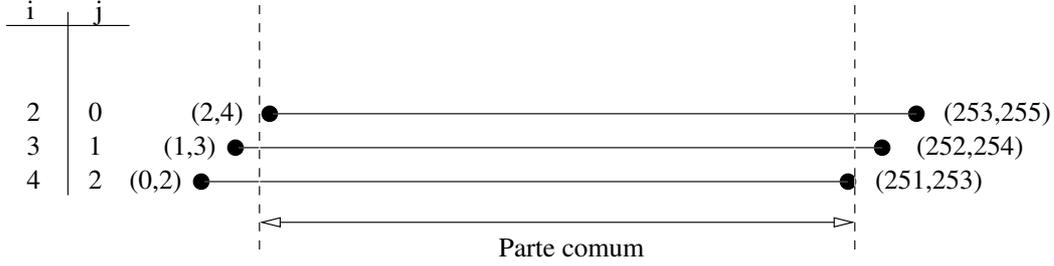


Figura 6.11: Coeficientes de distância dois.

dando origem a

$$\begin{aligned}
 c1_{xx}[2, 0] &= c1_{xx}^{d=2} + x^*[252].x[254] + x^*[253].x[255] \\
 c1_{xx}[3, 1] &= x^*[1].x[3] + c1_{xx}^{d=2} + x^*[252].x[254] \\
 c1_{xx}[4, 2] &= x^*[0].x[2] + x^*[1].x[3] + c1_{xx}^{d=2}
 \end{aligned} \tag{6.15}$$

Os três coeficientes da diagonal $d = 2$ da matriz $C1$ podem assim ser calculados a partir de $c1_{xx}^{d=2}$, acertando com os outros quatro elementos (de índices $k = 0, 1, 252$ e 253).

Distância 3

Para os dois coeficientes da matriz $C1$ em que $i = j + 3$, ou seja distância 3, representados na Figura 6.12

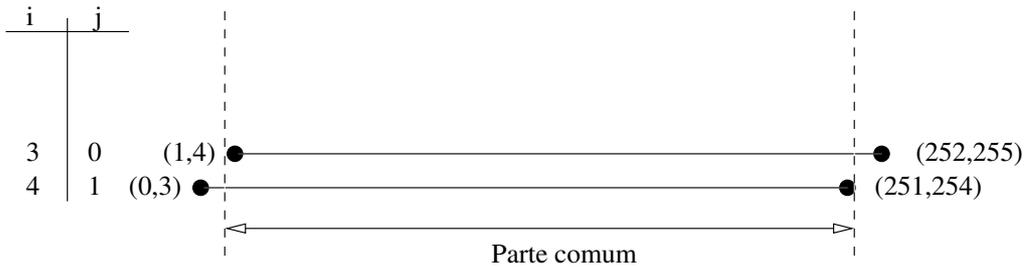


Figura 6.12: Coeficientes de distância três.

a parte comum $c1_{xx}^{d=3}$ pode ser obtida por

$$\begin{aligned}
 c1_{xx}^{d=3} &= \sum_{n=p}^{N-p+2} x^*[n-3].x[n] = \\
 &= x^*[1].x[4] + x^*[2].x[5] + \dots + x^*[250].x[253] + x^*[251].x[254]
 \end{aligned} \tag{6.16}$$

dando origem a

$$\begin{aligned} c1_{xx}[3, 0] &= c1_{xx}^{d=3} + x^*[252].x[255] \\ c1_{xx}[4, 1] &= x^*[0].x[3] + c1_{xx}^{d=3} \end{aligned} \quad (6.17)$$

Os dois coeficientes da diagonal $d = 3$ da matriz $C1$ podem assim ser calculados a partir de $c1_{xx}^{d=3}$, acertando com os elementos de índice $k = 0$ e 252 .

Distância 4

Neste caso trata-se de um único coeficiente que terá que ser calculado na íntegra pela autocorrelação parcial dos elementos do segmento de sinal com distância igual a 4, representados na Figura 6.13.



Figura 6.13: Coeficiente de distância quatro.

Esse coeficiente $c1_{xx}^{d=4}$ pode ser obtido por

$$\begin{aligned} c1_{xx}^{d=4} &= \sum_{n=p}^{N-p+3} x^*[n-4].x[n] = \\ &= x^*[0].x[4] + x^*[1].x[5] + \dots + x^*[250].x[254] + x^*[251].x[255] \end{aligned} \quad (6.18)$$

Distâncias negativas

De forma idêntica às descritas anteriormente poder-se-ia procurar a parte comum entre os coeficientes com a mesma distância. Sabendo-se no entanto que a matriz C é Hermitiana, os coeficientes da matriz triangular superior são os conjugados dos da matriz triangular inferior, podem assim ser obtidos a partir destes últimos.

Matriz $C2$

Distância 0

Tomem-se os coeficientes da matriz $C2$ em que $i = j$, ou seja distância = 0, desenvolvidos de acordo com a Eq. 6.8

$$\begin{aligned}
 c2_{xx}[0, 0] &= \sum_{n=0}^{251} x[n].x^*[n] = \\
 &= x[0].x^*[0] + x[1].x^*[1] + \dots + x[250].x^*[250] + x[251].x^*[251] \\
 c2_{xx}[1, 1] &= \sum_{n=0}^{251} x[n + 1].x^*[n + 1] = \\
 &= x[1].x^*[1] + x[2].x^*[2] + \dots + x[251].x^*[251] + x[252].x^*[252] \\
 c2_{xx}[2, 2] &= \sum_{n=0}^{251} x[n + 2].x^*[n + 2] = \\
 &= x[2].x^*[2] + x[3].x^*[3] + \dots + x[252].x^*[252] + x[253].x^*[253] \\
 c2_{xx}[3, 3] &= \sum_{n=0}^{251} x[n + 3].x^*[n + 3] = \\
 &= x[3].x^*[3] + x[4].x^*[4] + \dots + x[253].x^*[253] + x[254].x^*[254] \\
 c2_{xx}[4, 4] &= \sum_{n=0}^{251} x[n + 4].x^*[n + 4] = \\
 &= x[4].x^*[4] + x[5].x^*[5] + \dots + x[254].x^*[254] + x[255].x^*[255]
 \end{aligned}$$

Na Figura 6.14 representam-se os coeficientes de autocorrelação parcial da matriz $C2$ de distância zero, denotando cada elemento de dados $x(i)$ pelo seu índice i . Note-se que para evidenciar a semelhança a ordem dos índices (i, j) foi invertida.

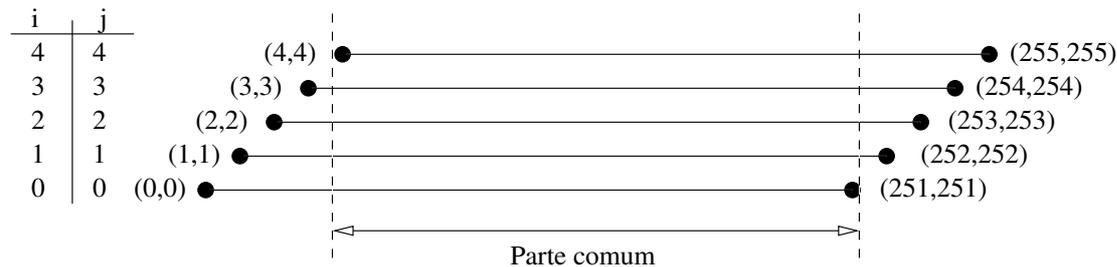


Figura 6.14: Coeficientes de distância zero.

De maior relevância para a simplificação algorítmica visada, para além do esperado facto de algumas parcelas serem comuns, nota-se que

$$c2_{xx}[0, 0] = c1_{xx}[4, 4]$$

$$\begin{aligned}c2_{xx}[1, 1] &= c1_{xx}[3, 3] \\c2_{xx}[2, 2] &= c1_{xx}[2, 2] \\c2_{xx}[3, 3] &= c1_{xx}[1, 1]\end{aligned}$$

Assim, a parte comum $c2_{xx}^{d=0}$, igual a $c1_{xx}^{d=0}$,

$$c2_{xx}^{d=0} = c1_{xx}^{d=0} = \sum_{n=p}^{N-p-1} x[n].x^*[n] \quad (6.19)$$

Distância 1

O desenvolvimento da Eq. 6.8 para os coeficientes da matriz $C2$ em que $i = j+1$, ou seja distância um, evidencia a correspondência entre os coeficientes da matriz $C2$ e $C1$.

$$\begin{aligned}c2_{xx}[1, 0] &= \sum_{n=0}^{251} x[n+1].x^*[n] = \\&= x[1].x^*[0] + x[2].x^*[1] + \dots + x[251].x^*[250] + x[252].x^*[251] = \\&= x^*[0].x[1] + x^*[1].x[2] + \dots + x^*[250].x[251] + x^*[251].x[252] = \\&= c1_{xx}[4, 3] \\c2_{xx}[2, 1] &= \sum_{n=0}^{251} x[n+2].x^*[n+1] = \\&= x[2].x^*[1] + x[3].x^*[2] + \dots + x[252].x^*[251] + x[253].x^*[252] \\&= x^*[1].x[2] + x^*[2].x[3] + \dots + x^*[251].x[252] + x^*[252].x[253] \\&= c1_{xx}[3, 2] \\c2_{xx}[3, 2] &= \sum_{n=0}^{251} x[n+3].x^*[n+2] = \\&= x[3].x^*[2] + x[4].x^*[3] + \dots + x[253].x^*[252] + x[254].x^*[253] \\&= x^*[2].x[3] + x^*[3].x[4] + \dots + x^*[252].x[253] + x^*[253].x[254] \\&= c1_{xx}[2, 1] \\c2_{xx}[4, 3] &= \sum_{n=0}^{251} x[n+4].x^*[n+3] = \\&= x[4].x^*[3] + x[5].x^*[4] + \dots + x[254].x^*[253] + x[255].x^*[254] \\&= x^*[3].x[4] + x^*[4].x[5] + \dots + x^*[253].x[254] + x^*[254].x[255] \\&= c1_{xx}[1, 0]\end{aligned}$$

é representado na Figura 6.15. Note-se que para evidenciar a semelhança neste caso, representam os índices (j, i) invertidos sendo o primeiro elemento de dados do par ordenado o elemento conjugado.

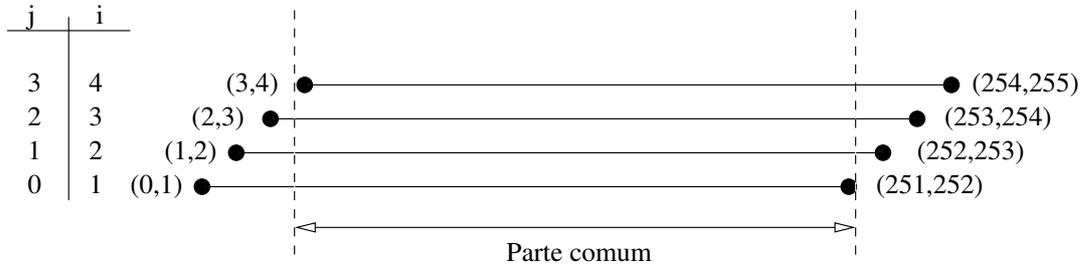


Figura 6.15: Coeficientes de distância um.

Outras distâncias

Verifica-se, para toda a matriz $C2$, que os seus elementos são iguais a outros da matriz $C1$. A matriz $C2$ pode ser vista como

$$\begin{bmatrix}
 c1_{xx}[4, 4] & c1_{xx}[3, 4] & c1_{xx}[2, 4] & c1_{xx}[1, 4] & c1_{xx}[0, 4] \\
 c1_{xx}[4, 3] & c1_{xx}[3, 3] & c1_{xx}[2, 3] & c1_{xx}[1, 3] & c1_{xx}[0, 3] \\
 c1_{xx}[4, 2] & c1_{xx}[3, 2] & c1_{xx}[2, 2] & c1_{xx}[1, 2] & c1_{xx}[0, 2] \\
 c1_{xx}[4, 1] & c1_{xx}[3, 1] & c1_{xx}[2, 1] & c1_{xx}[1, 1] & c1_{xx}[0, 1] \\
 c1_{xx}[4, 0] & c1_{xx}[3, 0] & c1_{xx}[2, 0] & c1_{xx}[1, 0] & c1_{xx}[0, 0]
 \end{bmatrix} \tag{6.20}$$

Algoritmo

Generalizando, considere-se o sinal $x[n]$ sinal complexo sendo N o comprimento do segmento de dados a considerar num modelo de ordem p . Os elementos fulcrais $c1_{xx}^{d=D}$, $D = 0, 1, \dots, p$ podem ser obtidos por

$$c1_{xx}^{d=D} = \sum_{n=p}^{N-p+d-1} x^*[n-d].x[n] \tag{6.21}$$

Os coeficientes da matriz $C1$ que se encontram na diagonal principal e matriz triangular inferior podem ser obtidos a partir dos elementos fulcrais de igual distância por

$$c1_{xx}^{d=D}[M+d, M] = c1_{xx}^{d=D} +$$

$$\sum_{k=p-M}^{p-1} x^*[k-d].x[k] + \sum_{k=N-p+d}^{N-M-1} x^*[k-d].x[k] \quad (6.22)$$

para $M = 0, \dots, p-d$.

Os coeficientes de $C2$ podem ser obtidos pré-multiplicando e pós-multiplicando a matriz $C1$ pela matriz de reflexão ³ de ordem p . Outra forma de obter estes elementos é por manipulação de índices, sendo $c2(p-M+1, p-(M+d)+1) = c1(M+d+1, M+1)$ para $d = 0, \dots, p$ e $M = 0, \dots, p-d$.

O código em *octave* que a seguir se apresenta traduz o algoritmo aqui descrito.

³A matriz de reflexão de ordem p é uma matriz quadrada em que os elementos sobre a diagonal secundária são iguais a um e os demais elementos são nulos (Marple Jr., 1987).

```

% Cálculo da matriz de covariância estendida - simplificado para complexos
%
function [Cg]=modcov_simp_gm(x,p)
% Input: x - segmento de dados; p - ordem do modelo;
% Output: Cg - matriz estendida de covariância modificada

N = length(x);
c1d = zeros(p+1, 1); % Elementos fulcrais
c1 = zeros(p+1, p+1); % matriz parcial
c2 = zeros(p+1, p+1); % matriz parcial
Cg = zeros(p+1, p+1); % matriz de covariância modificada

for d = 0:p,
    for n = p:(N-p+d-1),
        c1d(d+1) = c1d(d+1) + conj(x(n-d+1)) * x(n+1); % Elementos fulcrais
    end
    for M = 0:p-d,
        c1(M+d+1, M+1) = c1d(d+1);
        for k = p-M:p-1,
            c1(M+d+1, M+1) = c1(M+d+1, M+1) + conj(x(k-d+1)) * x(k+1);
        end
        for k = N-p+d:N-M-1,
            c1(M+d+1, M+1) = c1(M+d+1, M+1) + conj(x(k-d+1)) * x(k+1);
        end
        c2(p-M+1, p-(M+d)+1) = c1(M+d+1, M+1);
    end
end
end

den=2*(N-p); Cg= (c1 + c2) ./ den;
for i=0:p, % elementos transpostos
    for j = i+1: p,
        Cg (i+1, j+1) = conj(Cg(j+1,i+1));
    end;
end;
end;
end;

```

Comparação da complexidade dos algoritmos de cálculo da matriz de Covariância Modificada

Recorrendo à ferramenta *Matlab* (The Math Works Inc., 1991), foi anotado o número de operações de vírgula flutuante (*flops*) relativos ao cálculo da matriz estendida C segundo A) a equação 6.6, B) o algoritmo simplificado referido em (Ruano, 1992) e C) o algoritmo referido na secção 6.3.3.1. Os valores recolhidos os três casos para números reais considerando segmentos de dimensão 64, 128, 256 e 512 pontos e modelos de ordem 2, 4, 6, 8 e 10 são apresentados na Figura 6.16. Os valores com os índices entre parênteses, referentes ao algoritmo A, foram divididos por 10 para melhorar representação gráfica.

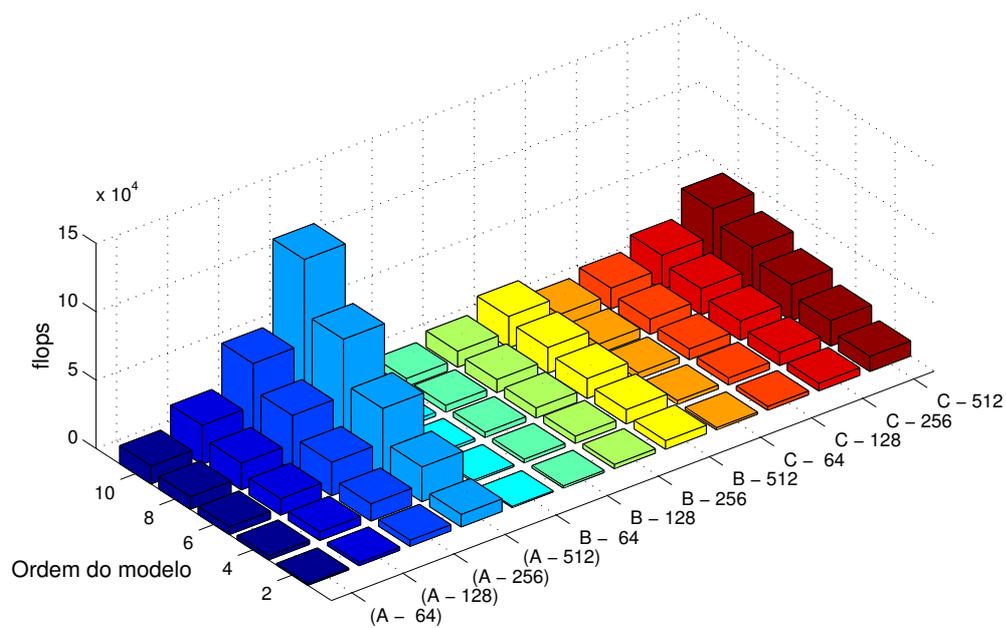


Figura 6.16: Comparação entre os algoritmos de cálculo da matriz de Covariância Modificada aplicados a dados reais.

A razão entre o número de operações de vírgula flutuante dos algoritmos simplificados C e B é representada na Figura 6.17. Conforme esperado o algoritmo C exige maior número de operações devido ao facto de contemplar operações sobre números complexos uma vez que o produto de dois elementos de dados corresponde a quatro multiplicações e duas somas. Verifica-se também que o cálculo de apenas um dos somatórios parcelas (c_1 ou c_2) se traduz em maior

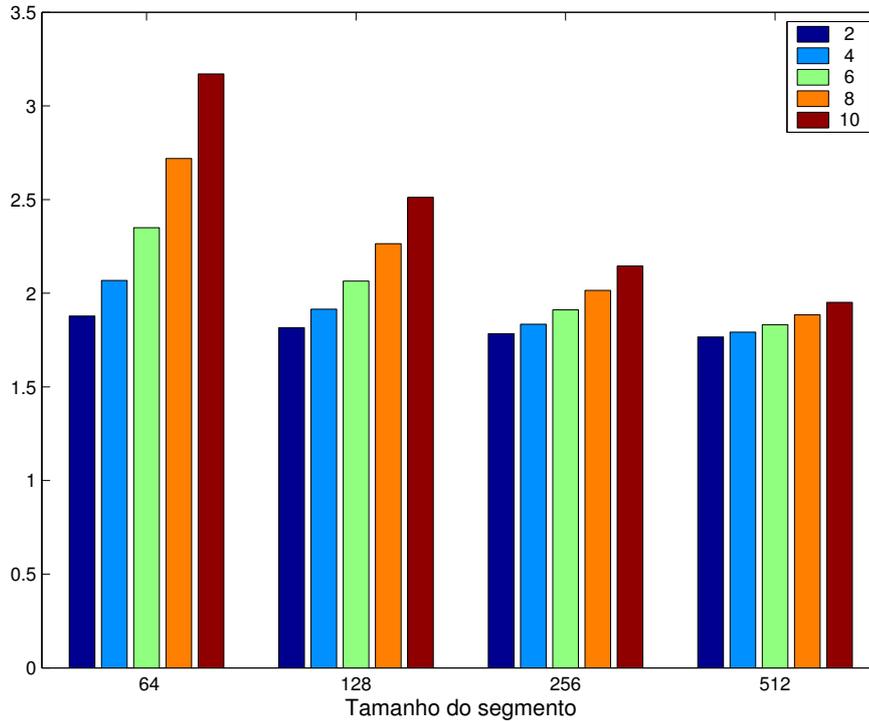


Figura 6.17: Razão entre os algoritmos simplificados de cálculo da matriz de Covariância Modificada aplicados a dados reais.

simplificação quanto maior é o tamanho do segmento.

A Figura 6.18 apresenta o número de operações de vírgula flutuante recolhidas pela ferramenta *Matlab* relativos ao cálculo da matriz estendida C utilizando os algoritmos A) e C), considerando segmentos de dimensão 64, 128, 256 e 512 pontos e modelos de ordem 2, 4, 6, 8 e 10. Os valores com os índices entre parênteses, referentes ao algoritmo A, foram divididos por 10 para melhorar a representação gráfica.

Comentários sobre as simplificações efectuadas

Conclui-se então que, sendo $x[n]$ é um sinal complexo, são necessários $(p + 1)$ elementos fulcrais para deduzir os $((p + 2) * (p + 1))/2$ coeficientes complexos da matriz de covariância e vector do lado direito.

Comparando com a solução anteriormente apontada em (Ruano, 1992) para um sinal real em que se calculavam os dois somatórios correspondentes às matrizes $C1$ e $C2$ para um quarto da matriz C estendida, neste caso os elementos fulcrais $c1_{xx}^{d=D}$ ($D = 0, 1, \dots, p$) consideram apenas um dos somatórios correspondentes

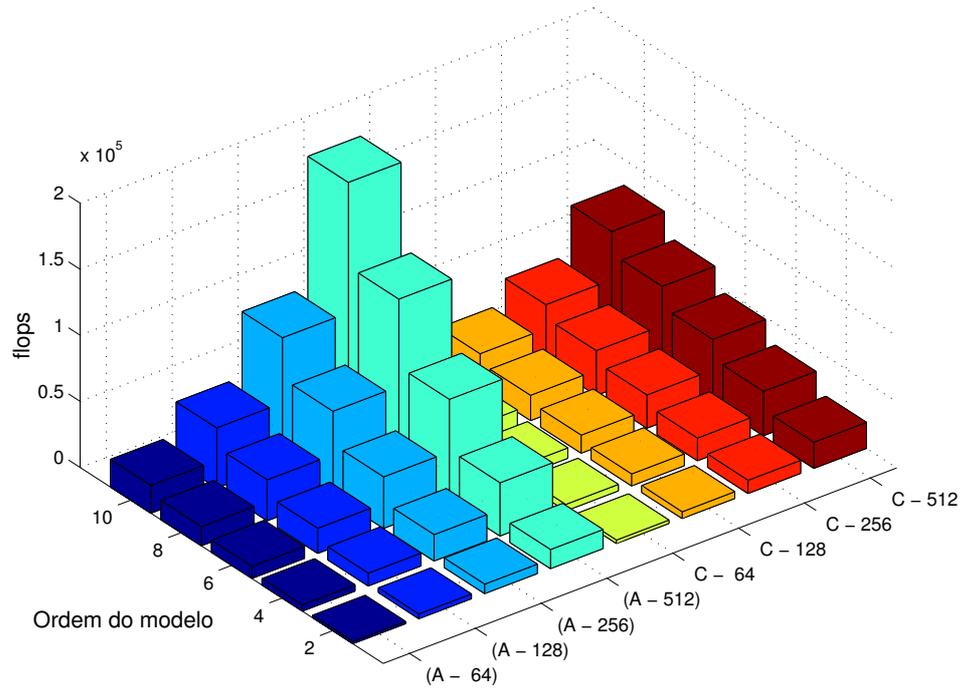


Figura 6.18: Operações de vírgula flutuante dos algoritmos de cálculo da matriz de Covariância Modificada aplicados a dados complexos.

à matriz $C1$, no entanto, a calcular para metade da matriz C . A solução apontada em (Ruano, 1992) deve ser usada para sinais reais e a solução aqui apontada deve ser usada para sinais complexos.

6.3.3.2 Cálculo da Densidade Espectral de Potência

Observando a fórmula da densidade espectral de potência, $P_{MC}(f_n)$ (Eq. 2.8) que aqui se transcreve

$$\hat{P}_{MC}(f_n) = \frac{\hat{\sigma}^2}{|A(f_n)|^2} = \frac{\hat{\sigma}^2}{\left| 1 + \sum_{k=1}^p a[k]z^{-k} \Big|_{z=e^{j2\pi f_n}} \right|^2} \quad (6.23)$$

em que $\hat{\sigma}^2$ representa a estimativa da variância do ruído branco do sinal e $A(f_n)$ constitui uma função polinomial de ordem p das estimativas $a[k]$ dos parâmetros auto regressivos do modelo, nota-se que o denominador pode ser obtido a partir de uma transformada discreta de Fourier das estimativas dos parâmetros $a[k]$. Esta abordagem, proposta em (Kay, 1988), tem como vantagem a possibilidade de

utilização de um algoritmo de DFT considerado adequado. Para o caso presente será utilizada a biblioteca *FFTW* aliada à gestão de sabedoria.

Para tanto há que considerar que a sequência $a_1[k]$ tal que,

$$a_1[k] = \begin{cases} 1 & \text{para } k = 0 \\ a[k] & \text{para } 1 \leq k \leq p \\ 0 & \text{para } p + 1 \leq m \leq N - 1 \end{cases} \quad (6.24)$$

O denominador da fórmula da densidade espectral de potência pode assim ser obtido pelo quadrado do módulo da DFT de $a_1[k]$.

6.3.3.3 Comparação entre os Algoritmos Sequenciais do AR-MC

Foram criados programas estáticos dos algoritmos sequenciais do AR-MC, de quarta ordem, usando o algoritmo proposto por (Kay, 1988) para dados reais e complexos e o algoritmo com as simplificações agora propostas para dados complexos. Foram consideradas várias dimensões de janelas, não sobrepostas, de dimensão igual a uma potência de dois. O cálculo da densidade espectral de energia no algoritmo simplificado faz apelo à *FFTW* aliada à sabedoria.

Os tempos de execução, em segundos, desses algoritmos, relatados na Tabela 6.9, foram medidos na plataforma *coral*, usando o comando *time* e correspondem à estimação de 100 sinais simulados de artéria carótida primitiva com a duração de 640 ms. cada. Para os testes usando dados complexos foram utilizados os sinais *analíticos* dos sinais simulados.

Dimensão da janela	Tempo		
	reais (Kay)	complexos (Kay)	complexos (simp)
64	2,406	5,100	1,149
128	2,881	5,018	0,809
256	2,774	4,908	0,732
512	2,838	4,975	0,695

Tabela 6.9: Tempos de execução dos algoritmos sequenciais do estimador AR-MC.

É de notar a pequena variação, para cada algoritmo, produzida pela variação da dimensão do segmento. Acontece que, por os segmentos não serem sobrepostos e a ordem do modelo se manter constante, a carga de cômputo também se mantém constante. A melhoria obtida nos tempos de execução com o algoritmo

simplificado face aos obtidos com a implementação do algoritmo proposto por Kay é representada na Figura 6.19.

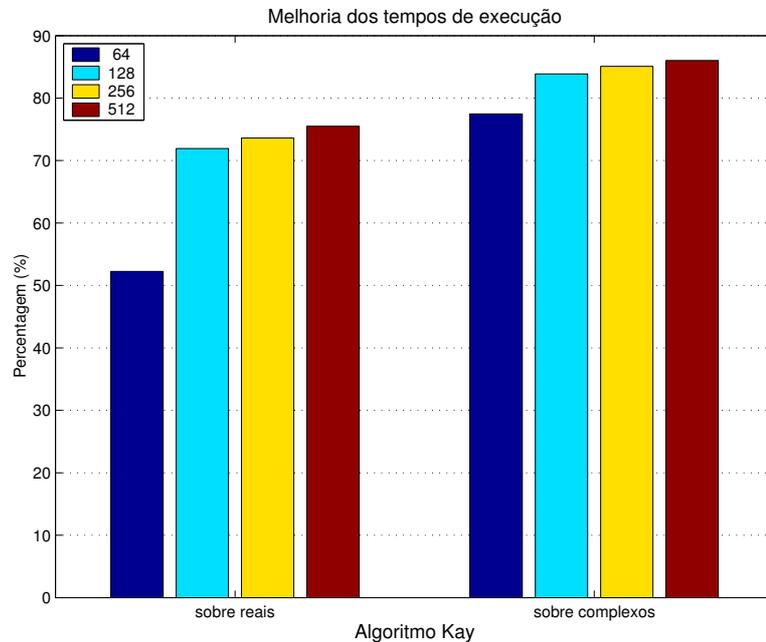


Figura 6.19: Melhoria nos tempos de execução obtida com o algoritmo simplificado face aos obtidos com o algoritmo proposto por Kay aplicado a dados reais e complexos.

Apesar da melhoria registada ser significativa, esta percentagem não se propaga, na íntegra, para a implementação global do estimador. Para além de ser a sobreposição dos segmentos e/ou o aumento da ordem do modelo que se traduz em maior carga para o sistema, é a gestão dos diferentes casos, nomeadamente a utilização do espaço de dados alocado dinamicamente, que maior impacto tem na qualidade da implementação.

6.3.3.4 Utilização do Estimador AR-MC

Em tempo de execução, os parâmetros de chamada do programa, utilizando o algoritmo simplificado, permitem definir tipo e dimensão das janelas, sobreposição entre janelas, uso ou não de sinal *analítico* (e em caso afirmativo, de que tipo), e a ordem do modelo. As diferentes opções são geridas pela janela de parametrização do método, conforme se apresenta na figura 6.20.



Figura 6.20: Definição das opções de execução do estimador AR-MC.

6.3.4 Algoritmos da DTF de Choi-Williams

Os estudos das implementações da distribuição tempo-frequência de Choi-Williams em plataformas aditivas consideravam algoritmos mais simples desenhados para os valores óptimos para a estimação de parâmetros de fluxo sanguíneo na artéria carótida primitiva. Os algoritmos da distribuição tempo-frequência de Choi-Williams agora estudados consideram a possibilidade de utilização de um dos três tipos de sinais *analítico* e um dos três núcleos modificados, conforme revisto na secção 2.3.3.

O algoritmo utilizado na implementação nas plataformas aditivas foi sendo progressivamente alterado, até permitir a parametrização de tipo e dimensão das janelas, sobreposição entre janelas, uso ou não de sinal *analítico* (e em caso afirmativo, de que tipo), tipo do núcleo e valores dos parâmetros σ e m .

Uma primeira fase deste estudo visou a determinação de possíveis simplificações no cálculo do núcleo, sendo em seguida articulada a multiplicação da autocorrelação do sinal com o núcleo em causa, operação que aqui se designa por produtos cruzados. Posteriormente foi feita a integração do cálculo do sinal *analítico*, sendo utilizada a implementação que recorre à FFTW aliada à sabedoria (veja-se a secção 6.3.2.1), sendo considerado um factor $M = 2$ quer na decimação quer na interpolação.

6.3.4.1 Cálculo dos Núcleos da Distribuição

Para os três novos núcleos procuraram-se as possíveis simplificações. A inclusão da descrição do núcleo standard (Cardoso, 1998; Choi & Williams, 1989; Cardoso, Fish & Ruano, 1996) visa permitir uma visão global.

Núcleo standard

O núcleo exponencial, originalmente proposto por Choi e Williams (Choi & Williams, 1989), pode ser definido no domínio discreto por (Cardoso, Fish & Ruano, 1996)

$$STD_KCW(\tau, \mu) = \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} \quad (6.25)$$

para $-L \leq \tau \leq L$ e $-m \leq \mu \leq m$ sendo $L = (N - 1)/2$, N a dimensão do segmento, m um parâmetro determinante da gama de variação temporal da

função de autocorrelação e σ um factor de escala que determina a suavidade da distribuição.

As simetrias exibidas pelo núcleo permitem calcular aproximadamente um quarto dos seus elementos (Cardoso, Fish & Ruano, 1996), uma vez que

$$\begin{aligned} STD_KCW(\tau, \mu) &= STD_KCW(-\tau, \mu) = \\ &= STD_KCW(\tau, -\mu) = \\ &= STD_KCW(-\tau, -\mu) \end{aligned} \quad (6.26)$$

Nas figuras 6.21 e 6.22 é possível observar as referidas simetrias. Em termos de implementação, a abordagem descrita em (Cardoso, Ruano & Fish, 1996a) considera o cálculo e armazenamento de cerca de um quarto da matriz do núcleo.

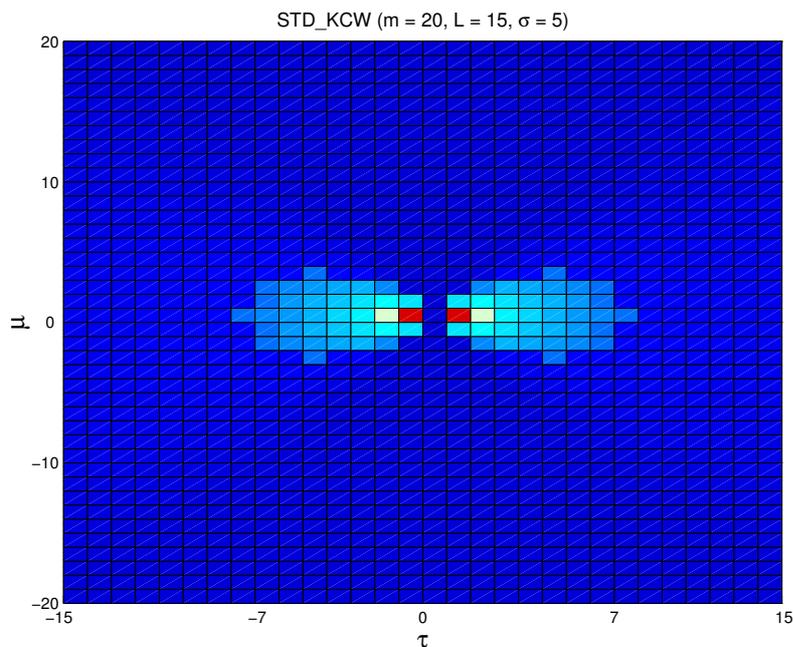


Figura 6.21: Vista de topo de um núcleo standard da CWD.

Considerando que, no caso presente, a dimensão do núcleo é definida em função dos parâmetros de chamada do programa, obrigando à alocação dinâmica do espaço para a matriz, sentiu-se maior necessidade em privilegiar o acesso sequencial ao espaço alocado do que a economia desse mesmo espaço. Desta forma, o cálculo dos índices poderá ser simplificado beneficiando o cálculo dos produtos cruzados, descrito na próxima secção. Cada ponto calculado é guardado nos quatro quadrantes.

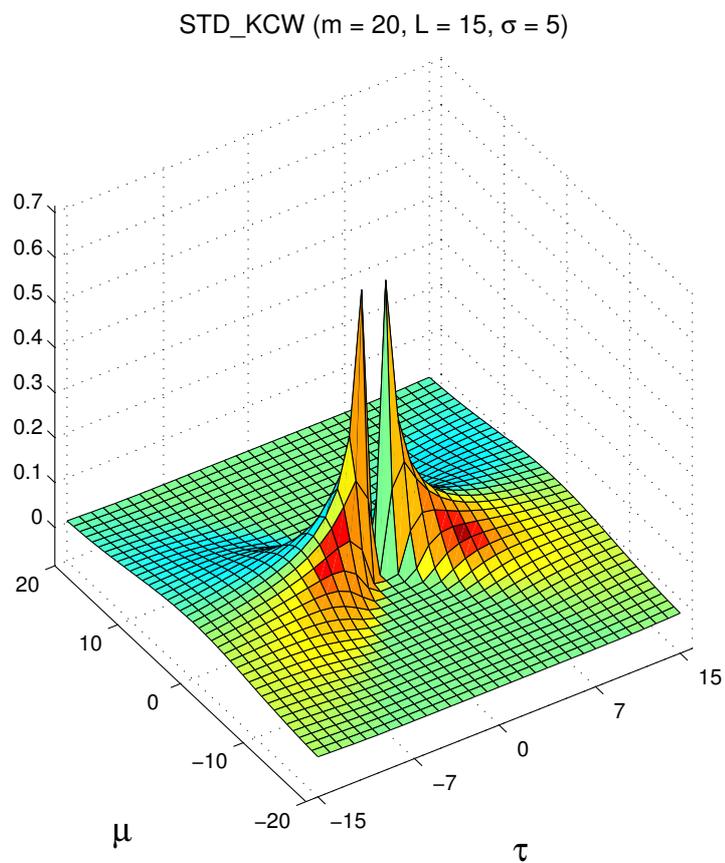


Figura 6.22: Perspectiva de um núcleo standard da CWD.

Verificando-se que τ e μ surgem na equação 6.27 elevados ao quadrado, basta calcular pouco mais de um oitavo da matriz, sendo

$$RI_KCW(\tau, -\mu) = RI_KCW(-\tau, \mu) \quad (6.28)$$

e os demais elementos nulos.

As figuras 6.23 e 6.24 apresentam duas representações de um mesmo núcleo RI, sendo possível notar a rotação de 180° em torno do centro $\tau = 0, \mu = 0$.

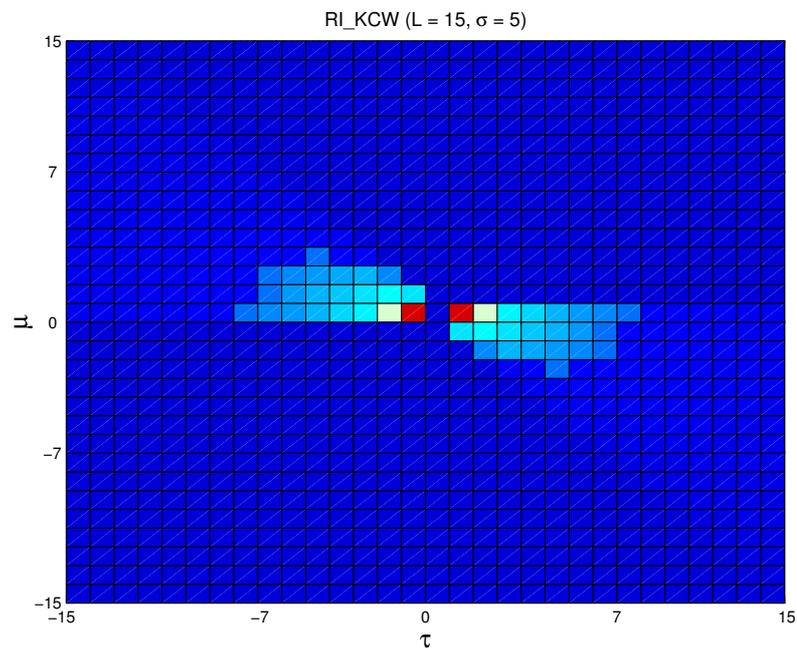


Figura 6.23: Vista de topo de um núcleo RI da CWD.

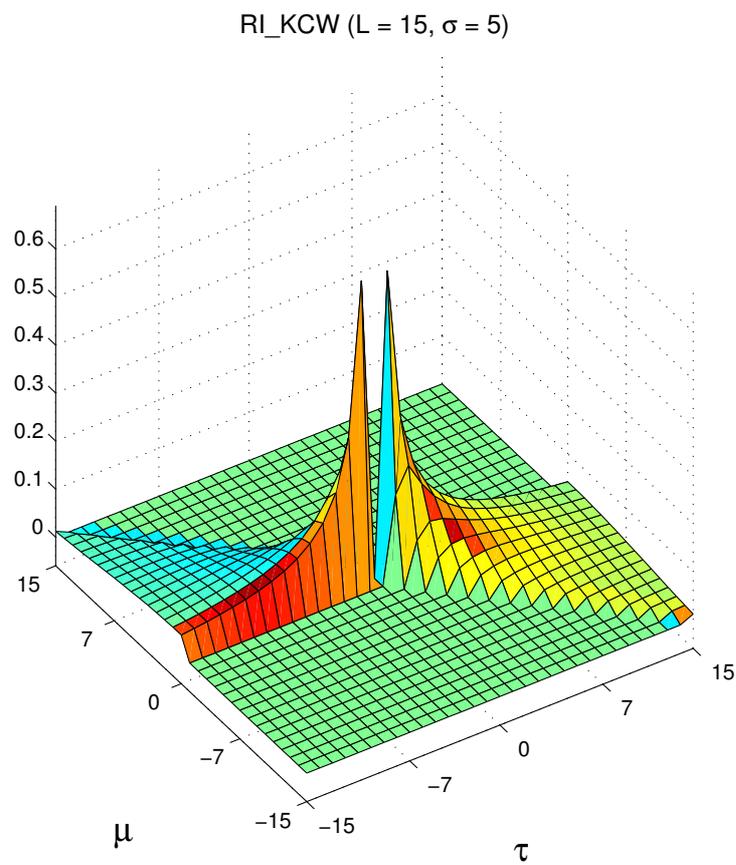


Figura 6.24: Perspectiva de um núcleo RI da CWD.

Núcleo *Alias Free*

O núcleo *alias free* (AF) para a CWD pode ser definido, no domínio discreto por (Jeong & Williams, 1992a; Leiria, Moura & Ruano, 2001) por

$$AF_KCW(\tau, \mu) = \frac{W_M(\mu)}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{(2\mu+\tau)^2}{16\tau^2/\sigma}} \quad (6.29)$$

para $-L \leq \tau \leq L$ e $-m \leq \mu \leq m$ sendo $L = (N - 1)/2$, N a dimensão do segmento, m um parâmetro determinante da gama de variação temporal da função de autocorrelação, σ um factor de escala que determina a suavidade da distribuição e $W_M(\mu)$ uma janela simétrica, normalmente rectangular, com valores nulos fora do intervalo $-m/2 \leq \mu \leq m/2$.

Observando as figuras 6.25 e 6.26 é possível notar a rotação de 180° em torno do centro $\tau = 0, \mu = 0$. Neste caso, é necessário calcular a metade da matriz

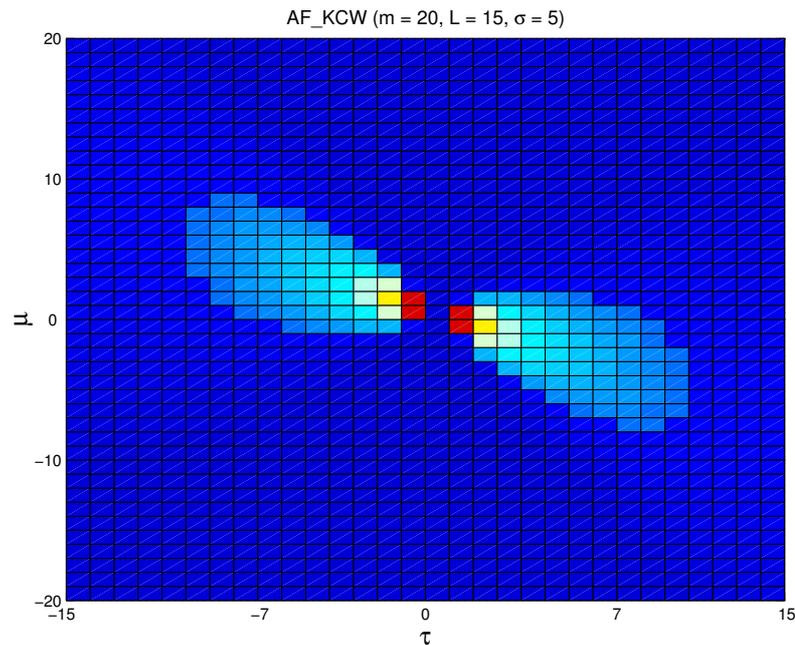


Figura 6.25: Vista de topo de um núcleo AF da CWD.

para $1 \leq \mu \leq L$, sendo iguais o primeiro e terceiro quadrantes e o segundo e quarto quadrantes.

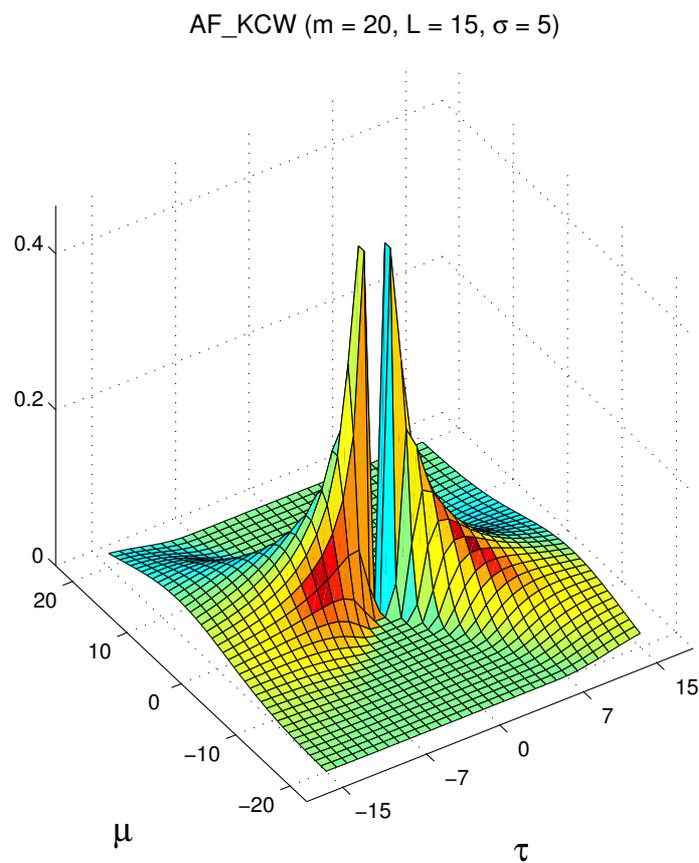


Figura 6.26: Perspectiva de um núcleo AF da CWD.

Núcleo *Reduced Interference Alias Free*

O núcleo *reduced interference alias free* (RIAF) para a CWD pode ser definido, no domínio discreto por (Leiria, Moura & Ruano, 2001) por

$$RIAF_KCW(\tau, \mu) = \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{(2\mu+\tau)^2}{16\tau^2/\sigma}} \quad (6.30)$$

para $-L \leq \tau \leq L$ e $(-\tau - |\tau|)/2 \leq \mu \leq (-\tau + |\tau|)/2$ sendo $L = (N - 1)/2$, N a dimensão do segmento e σ um factor de escala que determina a suavidade da distribuição.

Neste caso, o núcleo é igual ao AF, diferindo no domínio de aplicação, este último o mesmo que o do núcleo RI. Tal como com o núcleo RI, o parâmetro m não existe, sendo a dimensão da matriz do núcleo inteiramente dependente da dimensão do segmento.

Observando as figuras 6.27 e 6.28 é possível notar a semelhança com o núcleo RI, sendo necessário calcular aproximadamente um oitavo da matriz, sendo o segundo e o quarto quadrantes iguais.

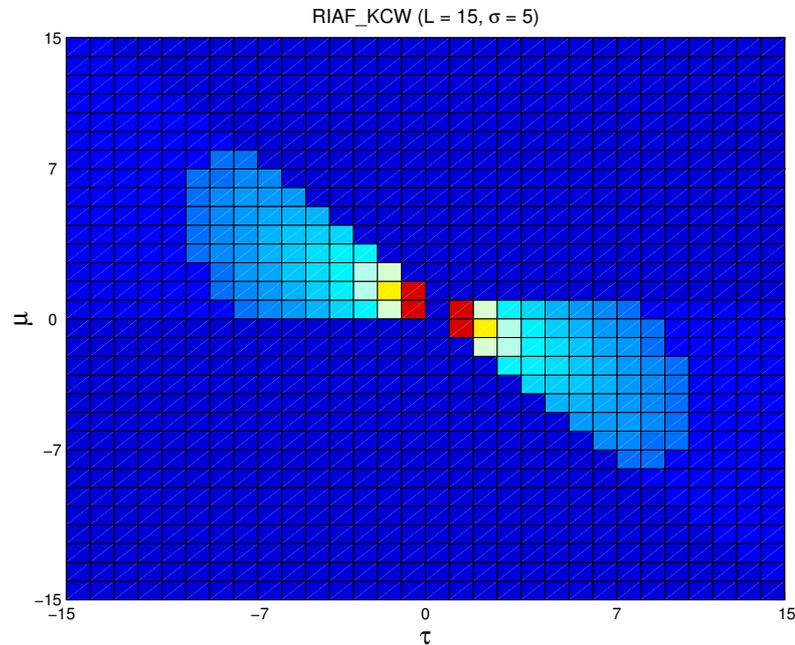


Figura 6.27: Vista de topo de um núcleo RIAF da CWD.

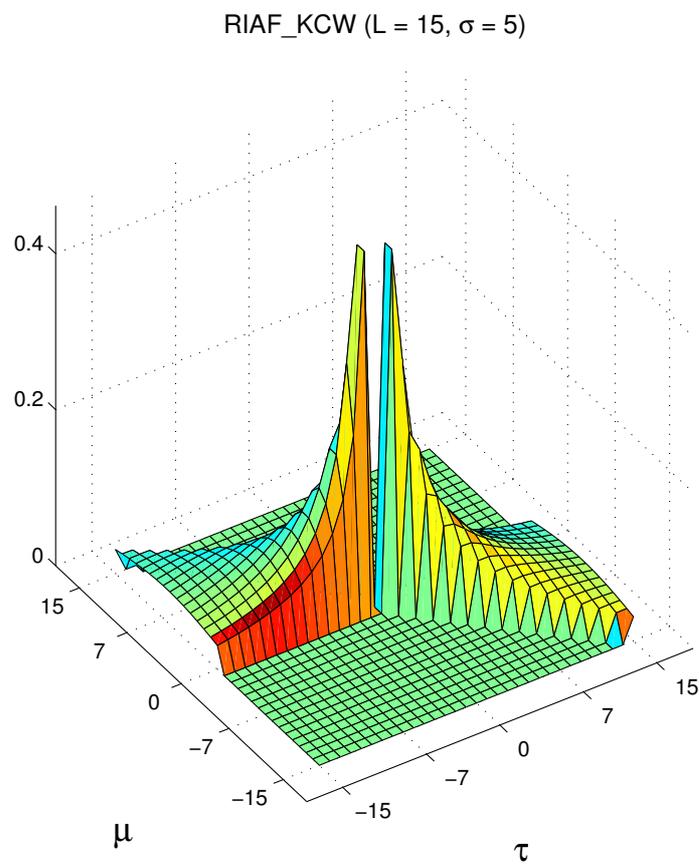


Figura 6.28: Perspectiva de um núcleo RIAF da CWD.

Tempo de cálculo dos núcleos

O cálculo do núcleo de uma distribuição ocorre apenas na fase de inicialização da aplicação que implementa o estimador. A tabela 6.10 apresenta, em milisegundos, os tempos de cálculo dos três tipos de núcleos. A escolha da dimensão dos núcleos considerou os casos mais frequentemente utilizados no processamento de sinais de fluxo sanguíneo de artéria carótida.

Dimensão da janela	STD (m = 30)	RI	AF (m = 30)	RIAF
63	0	0	0	0
127	0	0	10	0
255	0	10	10	10
511	10	20	20	30

Tabela 6.10: Tempos de cálculo, em milisegundos, de alguns núcleos da CWD.

Comparando os núcleos STD e AF, que têm o mesmo domínio de aplicação, este último requer um tempo de cálculo superior devido ao maior número de operações. Apesar de só ser necessário calcular cerca de um oitavo da matriz dos núcleos RI e RIAF, como a dimensão destes núcleos depende da dimensão da janela de dados aplicada, verifica-se um aumento dos tempos de cálculo em relação aos núcleos STD e AF, respectivamente.

6.3.4.2 Cálculo dos Produtos Cruzados da Distribuição

Por comodidade de leitura, transcreve-se a forma discreta da CWD, designada DCWD (Eq. 2.19),

$$DCWD(n, k) = 2 \sum_{\tau=-L}^L W_N(\tau) e^{-j2\pi k\tau/N} \left[\sum_{\mu=-m}^{+m} \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(n + \mu + \tau)x^*(n + \mu - \tau) \right] \quad (6.31)$$

onde $W_N(\tau)$ é uma janela simétrica com valores nulos fora do intervalo $-L \leq \tau \leq L$ e o valor de m indica o intervalo dentro do qual a função de autocorrelação será estimada.

As implementações da CWD consideraram o algoritmo descrito em (Cardoso, Fish & Ruano, 1996), em que a DCWD pode ser definida, em cada instante de

tempo, por

$$DCWD(0, k) = 2 \sum_{\tau=0}^{N-1} P'(\tau) e^{-j2\pi k\tau/N} \quad (6.32)$$

em que

$$P'(\tau) = \begin{cases} P(\tau) & 0 \leq \tau \leq \frac{N}{2} - 1 \\ 0 & \tau = \frac{N}{2} \\ P(\tau - N) & \frac{N}{2} + 1 \leq \tau \leq N - 1 \end{cases} \quad (6.33)$$

sendo

$$P(\tau) = W_N(\tau) \sum_{\mu=-m}^{+m} \frac{1}{\sqrt{4\pi\tau^2/\sigma}} e^{-\frac{\mu^2}{4\tau^2/\sigma}} x(\mu + \tau) x^*(\mu - \tau) \quad (6.34)$$

Seguindo essa abordagem, e designando por produtos cruzados ($P(\tau)$, com $-L \leq \tau \leq L$), a ponderação da função de autocorrelação do sinal pelo núcleo da distribuição, procedeu-se ao estudo da implementação da distribuição CWD incluindo núcleos modificados.

No caso do núcleo standard, os produtos cruzados são descritos pela Eq. 6.35, sendo calculados para $-L \leq \tau \leq L$.

$$P_STD(\tau) = W_N(\tau) \sum_{\mu=-m}^{+m} STD_KCW(\mu, \tau) x(\mu + \tau) x^*(\mu - \tau) \quad (6.35)$$

A Eq. 6.36 descreve o cálculo, utilizando um núcleo RI, dos produtos cruzados calculados para $-L \leq \tau \leq L$.

$$P_RI(\tau) = W_N(\tau) \sum_{\mu=(-\tau-|\tau|)/2}^{(-\tau+|\tau|)/2} RI_KCW(\mu, \tau) x(\mu + \tau) x^*(\mu) \quad (6.36)$$

Os produtos cruzados, quando é utilizado o núcleo AF são descritos pela Eq. 6.37, sendo calculados para $-L \leq \tau \leq L$.

$$P_AF(\tau) = W_N(\tau) \sum_{\mu=-m}^{+m} AF_KCW(\mu, \tau) x(\mu + \tau) x^*(\mu) \quad (6.37)$$

Finalmente, no caso de se utilizar o núcleo RIAF, o cálculo dos produtos cruzados é descrito pela Eq. 6.38.

$$P_RIAF(\tau) = W_N(\tau) \sum_{\mu=(-\tau-|\tau|)/2}^{(-\tau+|\tau|)/2} RIAF_KCW(\mu, \tau) x(\mu + \tau) x^*(\mu) \quad (6.38)$$

Tempo de cálculo dos produtos cruzados

A Fig. 6.29 ilustra a função de autocorrelação incluída no cálculo dos produtos cruzados, considerando $m = 5$ e $N = 11$ ($-5 \leq L \leq 5$), consoante o tipo de núcleo. Nos eixos são representados os índices dos dois termos, sendo o índice do termo conjugado representado no eixo vertical. O domínio de cálculo, para cada instante τ , é marcado por uma linha contínua, sendo assinalados com pontos os elementos não nulos que contribuem para o cálculo.

Comparando as funções de autocorrelação, nota-se que P_{RI} e P_{RIAF} requerem menor número de operações e P_{AF} o maior. No entanto, o número de operações necessário ao cálculo dos índices usados em P_{STD} é superior ao de P_{AF} .

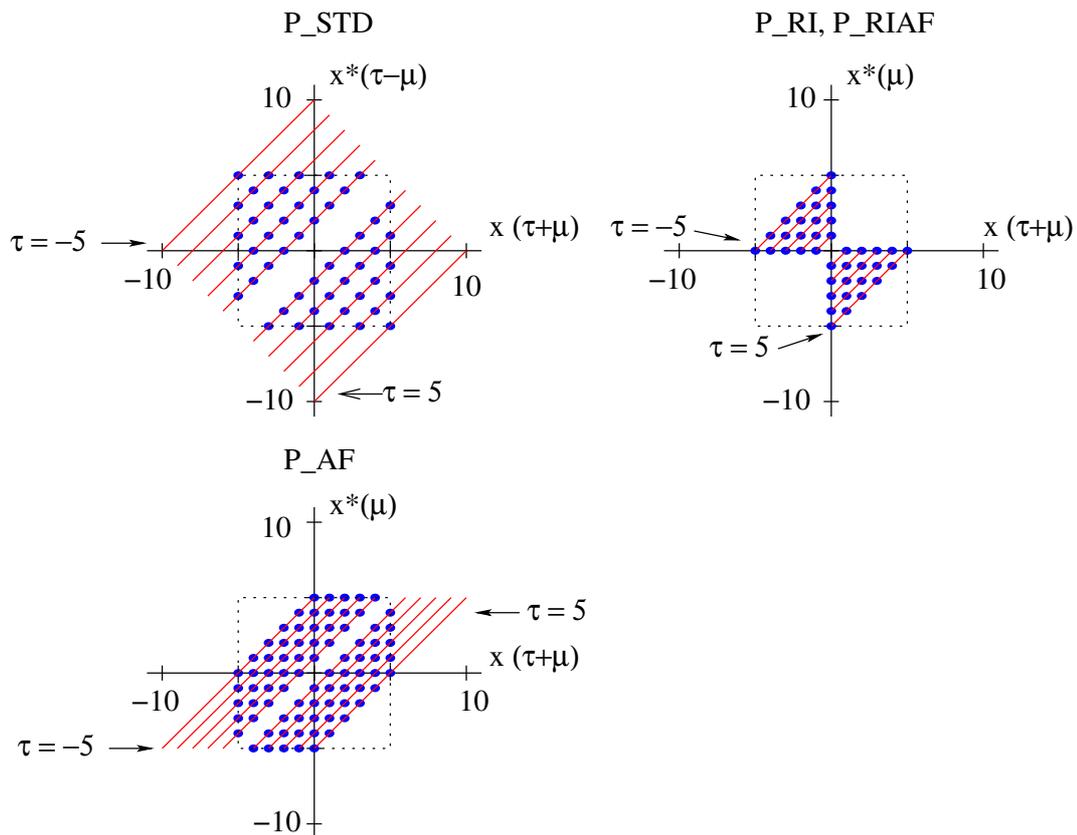


Figura 6.29: Função de autocorrelação no cálculo dos produtos cruzados da distribuição CWD

Estas observações são apoiadas pela análise dos valores da tabela 6.11, que apresenta, em segundos, os tempos acumulados correspondentes ao cálculo dos

produtos cruzados, para um sinal de fluxo sanguíneo de 33,61 s, considerando segmentos não sobrepostos com a dimensão expressa na tabela. Nos núcleos RI e RIAF considerou-se $m = 30$. Foram incluídos os valores respeitantes à utilização do núcleo standard de forma a permitir uma visão global.

Dimensão da janela	STD ($m = 30$)	RI	AF ($m = 30$)	RIAF
63	2,25	1,16	2,81	0,94
127	3,35	2,19	3,54	2,18
255	3,78	4,29	3,60	4,40
511	4,09	10,6	3,93	10,48

Tabela 6.11: Tempos de cálculo, em milisegundos, dos produtos cruzados da implementação da CWD.

O tipo de núcleo e a dimensão da matriz do núcleo (ou o número de parcelas da função de autocorrelação) são os factores determinantes da complexidade do cálculo dos produtos cruzados e, conseqüentemente, do tempo necessário ao cálculo.

Para as dimensões 63 e 127, o cômputo dos produtos cruzados para os núcleos RI e RIAF é inferior aos demais. Este facto deve-se ao número de elementos não nulos, implicado pelo domínio de aplicação do núcleo e da função de autocorrelação. Exemplificando, quando a dimensão da janela é igual a 63 ($L = (N - 1)/2 = 31$), os quatro núcleos têm dimensões aproximadas: $\#_{STD_KCW} = \#_{AF_KCW} = 63 \times 61 (= 2 * m + 1)$ e $\#_{RI_KCW} = \#_{RIAF_KCW} = 63 \times 63$. Nesse caso, o número de elementos não nulos nos núcleos RI e RIAF ($2 * (\sum_{i=2}^{31} i = 238.5)$), bastante inferior ao dos demais núcleos ($\approx (62 * 61) = 1922$), conduz a um tempo dispendido no cálculo bastante inferior.

Para as demais dimensões referidas na tabela 6.11, o contributo da dimensão da matriz no tempo de cálculo faz-se sentir mais para os núcleos RI e RIAF. Comparando os núcleos STD e AF, nota-se que, com o aumento da dimensão da matriz do núcleo, este último é relativamente mais leve, o que é explicado pelo cálculo dos índices das parcelas da autocorrelação.

6.3.4.3 Integração do Cálculo do Sinal *Analítico*

Excepto no caso de utilização de núcleos AF ou RIAF, para evitar *aliasing* é conveniente utilizar o sinal *analítico* em alternativa à amostragem do sinal com uma frequência dupla da frequência de Nyquist ($f_a = 2 * f_{Nyquist} = 4 * f_{max}$).

Assim, nesta aplicação foi integrado o cálculo do sinal *analítico* de cada segmento a estimar, utilizando a implementação sobre a FFTW aliada à sabedoria (ver secção 6.3.2.1). Na actual versão do subsistema, para além de ser possível escolher um dos três tipos de sinal *analítico*, é também possível indicar a não utilização de sinal *analítico*.

Para a outra alternativa que consiste no cálculo do sinal *analítico* de um ciclo cardíaco completo, a implementação do estimador CWD deve contemplar a entrada de dados complexos. Essa implementação foi também efectuada, embora não tenha sido integrada sob o interface gráfico.

O impacto registado com a integração dos diferentes tipos de sinal *analítico* na aplicação não se limita ao acréscimo temporal implicado por esse cálculo. A alteração de dimensão implicada pelo uso do sinal *analítico* decimado ou interpolado afecta a dimensão das janelas, do núcleo e, naturalmente, o domínio dos produtos cruzados.

A complexidade da aplicação deste estimador, que contempla vários tipos de núcleos e vários tipos de sinal *analítico* é ilustrada no diagrama da figura 6.30.

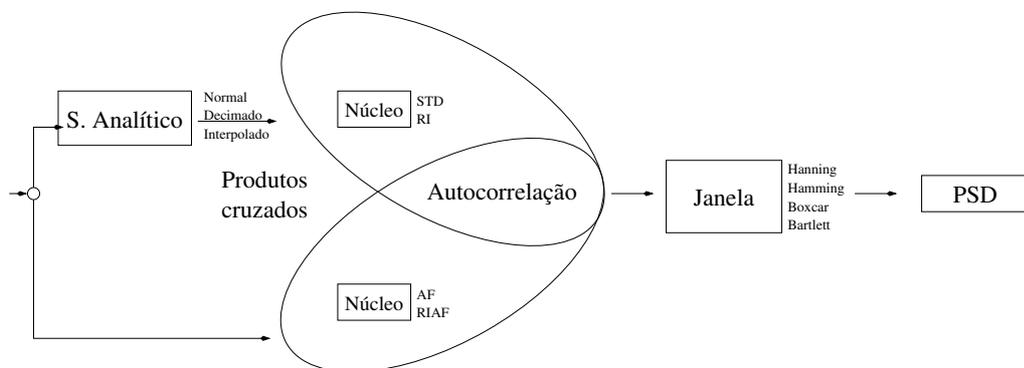


Figura 6.30: Diagrama da implementação do estimador CWD

A aplicação garante que a dimensão da janela é um número ímpar, que com os núcleos AF ou RIAF não é usado sinal *analítico* e que com os núcleos RI ou RIAF o valor de m não é especificado pelo utilizador.

6.3.4.4 Tempos da Implementação da CWD

Tome-se a última das várias versões construídas no processo de desenvolvimento desta implementação. Não sendo razoável apresentar os tempos de execução das várias alternativas, optou-se por apresentar apenas o correspondente à estimação

de segmentos de dimensões 63, 127, 255 ou 511 pontos, usando janelas de Hanning e para os vários tipos de núcleo. O parâmetro σ é igual a cinco e m , quando aplicável, igual a 30. Os tempos de execução foram medidos na plataforma *coral*, usando o comando *time* e correspondem à estimação de um sinal de fluxo sanguíneo da artéria carótida primitiva com a duração de 33,61 s.

Na tabela 6.12 apresentam-se os tempos, em segundos, considerando segmentos não sobrepostos.

Dimensão da janela	STD	RI	AF	RIAF
63	3,705	2,245	3,618	2,036
127	4,550	3,204	4,390	3,020
255	4,789	5,139	4,611	5,084
511	5,167	11,597	4,771	11,348

Tabela 6.12: Tempos de execução, em segundos, do estimador CWD de núcleos modificados.

Considerando segmentos não sobrepostos, a estimação respeita os limites de tempo-real. Comparando com os valores da tabela 6.11, é possível verificar que o cálculo dos produtos cruzados determina o desempenho do estimador.

Os tempos de execução do estimador, considerando taxas de sobreposição de 0, 25, 50 e 75%, são representados na figura 6.31. Aos gráficos foi acrescentada uma linha em 33,61s, representativa do limite de tempo-real (TR).

Um valor não nulo da taxa de sobreposição determina um aumento da carga computacional e conseqüente aumento do tempo de execução. Note-se que, na maioria dos casos, o tempo de execução se mantém abaixo do limite de tempo-real, excepto para os casos de segmentos de 511 pontos, com 75% de sobreposição, usando núcleos RI ou RIAF.

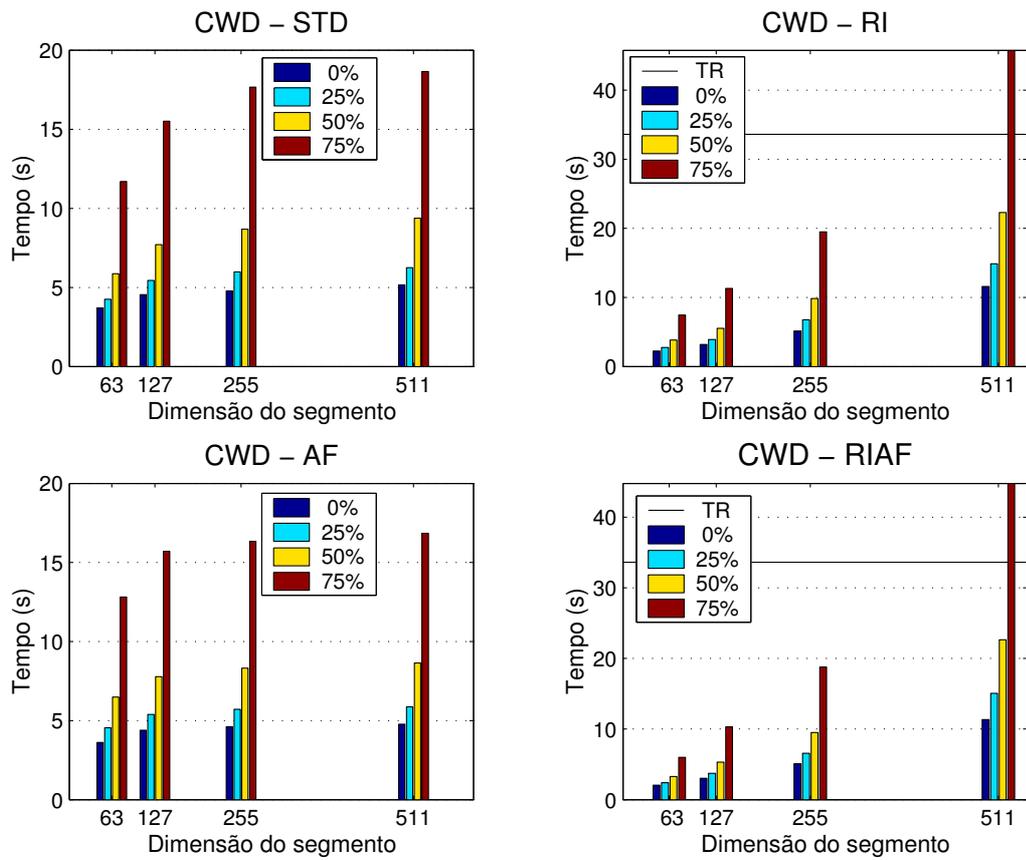


Figura 6.31: Tempos de execução da implementação do estimador CWD para segmentos de 64, 128, 256 e 512 pontos com 0, 25, 50 e 75% de sobreposição.

6.3.4.5 Comentários sobre a Implementação da CWD

A adoção de qualquer uma das diferentes configurações do estimador implica variações na carga do algoritmo como ficou aqui demonstrado. A otimização da implementação, reduzindo a complexidade inerente à gestão dos vários casos, só é possível após uma definição mais concreta da utilização final pretendida.

Para utilização na investigação sobre a adequabilidade para diversos sinais de fluxo sanguíneo, é conveniente ter uma única aplicação que concentre todas as possibilidades em estudo. Como exemplo, refira-se o estudo feito sobre a utilização dos algoritmos de núcleo modificado na determinação de dois parâmetros espectrais de fluxo sanguíneo na válvula aórtica(Leiria, Moura & Ruano, 2001), em que a utilização de uma única aplicação permitiu o estudo exaustivo da configuração do método.

6.3.4.6 Utilização do Estimador CWD

Como são as características do sinal de fluxo sanguíneo a estimar que determinam os valores óptimos dos parâmetros dos estimadores, esses valores são definidos no início da execução. Os parâmetros considerados incluem tipo e dimensão das janelas, sobreposição entre janelas, uso ou não de sinal *analítico* (e em caso afirmativo, de que tipo), tipo do núcleo e valores dos parâmetros σ e m . Tal como para os demais estimadores, as diferentes opções são geridas pela janela de parametrização do método, conforme se apresenta na figura 6.32.



Figura 6.32: Definição das opções de execução do estimador CWD.

6.3.5 Algoritmo da DTF de Bessel

O estudo da implementação da distribuição tempo-frequência de Bessel (DBD) foi incluído para as plataformas genéricas devido à qualidade da estimação referida em diversas publicações (Cardoso, 1998; Guo, Durand & Lee, 1994a; Guo, Durand & Lee, 1994b; Cardoso, Ruano & Fish, 1996a).

Sendo também uma distribuição da classe de Cohen, a sua formalização é semelhante à Choi-Williams. Verifica-se que é possível seguir a mesma abordagem adoptada para a CWD, analisando o núcleo, determinando a função dos produtos cruzados e integrando o cálculo do sinal *analítico*.

No estudo da implementação deste estimador foram consideradas diferentes abordagens e analisados os tempos de execução em diversas máquinas. Parte do trabalho aqui descrito foi já divulgado internacionalmente (Moura, Leiria & Ruano, 2002).

Transcreve-se, para maior comodidade de leitura, a forma discreta da distribuição tempo-frequência de Bessel (cf. Eq. 2.24) (Guo, Durand & Lee, 1994b), definida por

$$DBD_f(n, k) = 2 \sum_{\tau=-\infty}^{+\infty} W_N(\tau) e^{-j2\pi k\tau/N} \left[\sum_{\mu=-2\alpha|\tau|}^{+2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(n + \mu + \tau) x^*(n + \mu - \tau) \right] \quad (6.39)$$

com

$$\left[\sum_{\mu=-2\alpha|\tau|}^{+2\alpha|\tau|} \frac{1}{\pi\alpha|\tau|} \sqrt{1 - \left(\frac{\mu}{2\alpha\tau}\right)^2} x(n + \mu + \tau) x^*(n + \mu - \tau) \right]_{\tau=0} = x(n)x^*(n) \quad (6.40)$$

onde n e k correspondem às variáveis discretas tempo e frequência, respectivamente, e $W_N(\tau)$ é uma janela simétrica com valores nulos fora do intervalo $[-N/2, N/2]$.

A aplicação de uma distribuição tempo-frequência a um sinal pressupõe um conhecimento genérico sobre o comportamento desse sinal e a afinação dos parâmetros do método ao sinal em causa. Centrando a aplicação do método em sinais de fluxo sanguíneo, usar-se-à uma gama de valores alargada determinada em estudos de diversos autores (Guo, Durand & Lee, 1994a; Guo, Durand & Lee, 1994b; Cardoso, Ruano & Fish, 1996a; Leiria, 2000). Na adequação da

distribuição ao sinal, é dada especial relevância à escolha do parâmetro α do núcleo, um factor de escala, sendo referido $\alpha = 2$ para sinais de fluxo sanguíneo em artéria carótida (Cardoso, 1998) e $\alpha = 16$ para sinais da artéria femoral (Guo, Durand & Lee, 1994a).

O teste da implementação considerou o mesmo sinal clínico de artéria carótida primitiva, com duração de 33,61 s.

6.3.5.1 Arquitecturas Testadas

Para avaliar a exequibilidade da implementação em tempo-real desta distribuição em computadores pessoais comuns, a implementação foi testada em três máquinas equipadas com processadores Intel, todas usando o sistema operativo Debian/Linux, versão 2.2.

Os principais elementos de distinção dos computadores utilizados são apresentados na tabela 6.13. De forma a permitir conclusões gerais, apenas um processador do computador *ticoteco* foi utilizado.

Nome	Modelo Proc.	Frequência Relógio (MHz)	RAM (MByte)	No. Proc.	Versão <i>kernel</i>
arquimedes	MMX	166	32	1	2.2.17
coral	III	500	64	1	2.2.18pre21
ticoteco	II	350	128	2	2.2.15

Tabela 6.13: Distinção entre os computadores utilizados.

6.3.5.2 Estudo do Algoritmo

O algoritmo computacional da distribuição tempo-frequência de Bessel é, essencialmente, sequencial: a forma analítica de um segmento de dados é calculada; autocorrelações, com diversas distâncias, dessa forma analítica são ponderadas pelo núcleo da distribuição e transformadas para o domínio da frequência pelo cálculo da transformada de Fourier, obtendo-se a representação da densidade espectral de potência do instante correspondente ao segmento em estudo.

Procedeu-se a um estudo preliminar do comportamento do algoritmo, semelhante ao utilizado para a CWD.

O núcleo da distribuição é função do parâmetro α e da dimensão dos segmentos. Evitando cálculos repetitivos à custa de espaço de memória, este núcleo é

pré-calculado numa fase de inicialização da aplicação e mantido em memória.

Considerou-se que a dimensão das janelas seria uma potência de dois, e foi utilizada a implementação da transformada rápida de Fourier segundo Kay (Kay, 1988), para o cálculo do sinal *analítico* e, na última fase do algoritmo, para o cálculo da densidade espectral de energia.

Um primeiro estudo considerou uma única aplicação em C, fortemente dependente de alocação dinâmica de memória que considera o sinal de entrada de tipo complexo. Os parâmetros dessa aplicação permitem especificar o tipo de janela – Boxcar, Bartlett, Hanning or Hamming; a dimensão dessa janela, em pontos – 64, 128, 256 ou 512; o passo, isto é, o número de pontos entre o início de dois segmentos consecutivos, traduzindo a sobreposição de janelas; a utilização, ou não, de sinal *analítico* e o valor do parâmetro α (0,5, 1, 2, 5, 10, 15 ou 20).

O tipo de janela utilizada não influencia o tempo de execução da aplicação, como se pode verificar na figura 6.33, onde se representam os tempos de execução da aplicação nas três máquinas considerando segmentos não sobrepostos de 256 pontos.

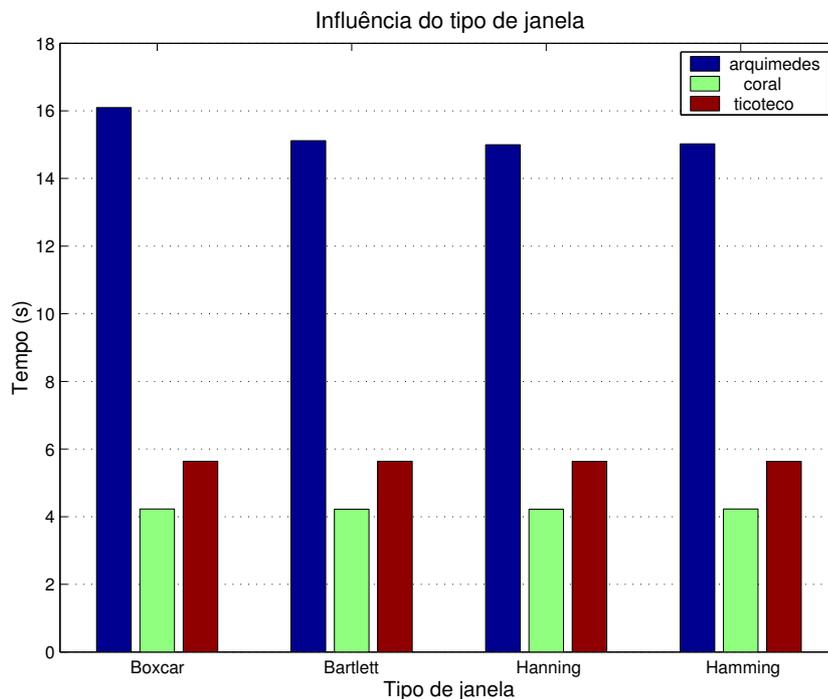


Figura 6.33: Tempos de execução da distribuição Bessel, nos computadores *arquimedes*, *coral* e *ticoteco*, para janelas não sobrepostas de 256 pontos, variando o tipo de janela utilizada.

O tamanho da janela condiciona o tempo de execução mas, para janelas não sobrepostas com as dimensões testadas, o tempo de execução é sempre inferior ao limite de tempo-real (figura 6.34) para janelas de 64, 128, 256 e 512.

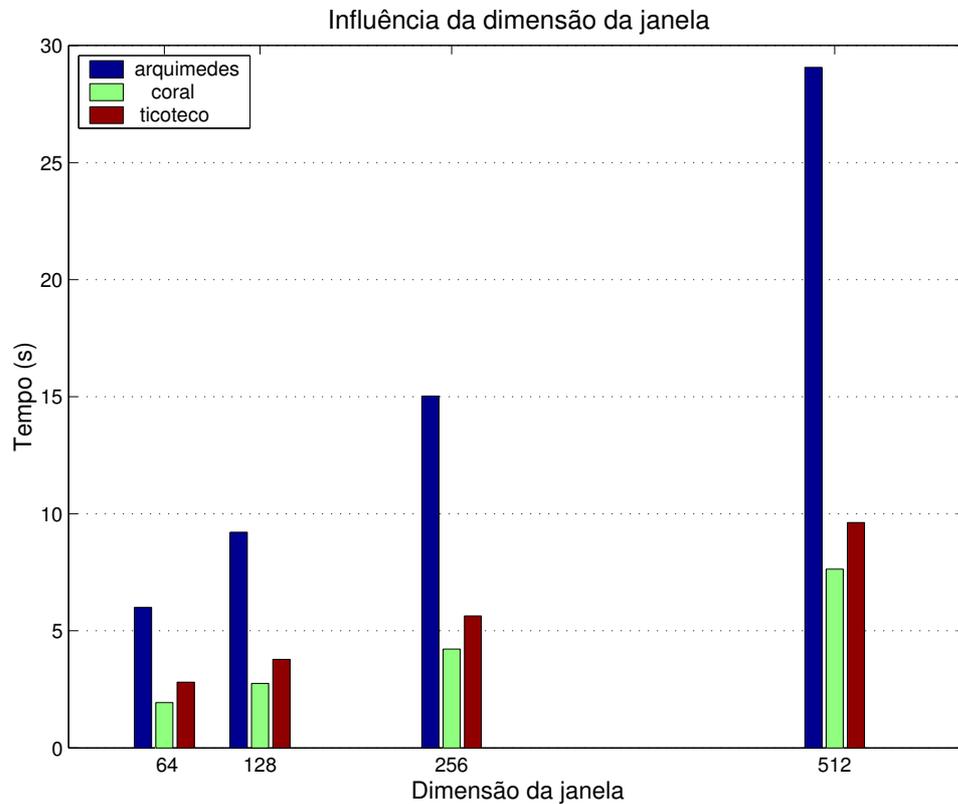


Figura 6.34: Tempos de execução da distribuição Bessel, nos computadores *arquimedes*, *coral* e *ticoteco*, para janelas não sobrepostas de 64, 128, 256 e 512 pontos.

Foi observado o comportamento da aplicação para algumas taxas de sobreposição das janelas para a maior dimensão de segmentos considerada (512 pontos). Os tempos medidos são representados na figura 6.35, tendo sido adicionada uma linha a 33,61 s, representando o limite de tempo-real. Verifica-se a existência de

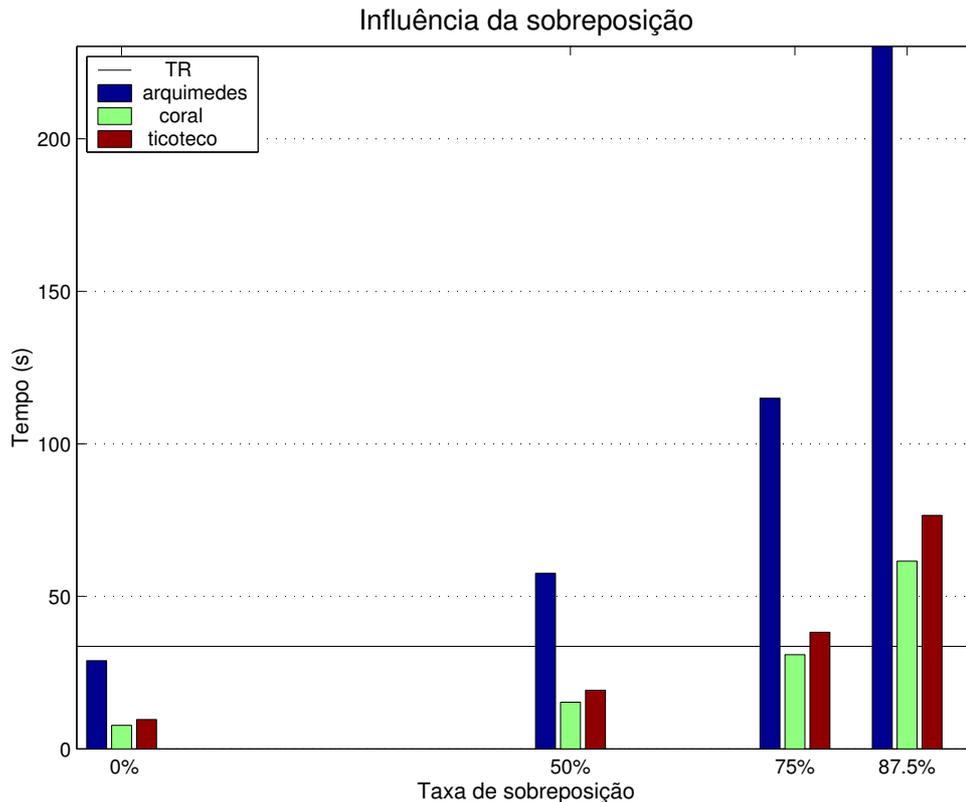


Figura 6.35: Tempos de execução da distribuição Bessel, nos computadores *arquimedes*, *coral* e *ticoteco*, para segmentos de 512 pontos com 0, 50, 75 e 87,5% de sobreposição. Linha horizontal representa o limite de tempo-real (33,61 s).

proporcionalidade entre os tempos de execução registados e o aumento da carga, nem sempre sendo cumprido o limite de tempo-real, consoante a velocidade da máquina empregue. Considerando segmentos sobrepostos, os tempos registados no computador *arquimedes* ultrapassam o limite de tempo-real; os registados no computador *ticoteco* mantêm-se abaixo do limite até 50% de sobreposição e no *coral* até 75% de sobreposição.

Considerando o cálculo do sinal *analítico* de segmentos de 64, 128, 256 e 512 pontos não sobrepostos (figura 6.36), verifica-se que o tempo de estimação se mantém abaixo do limite de tempo-real com a única exceção do computador *arquimedes* na estimação de segmentos de 512 pontos. Também neste caso foi adicionada uma linha marcando o limite de tempo-real.

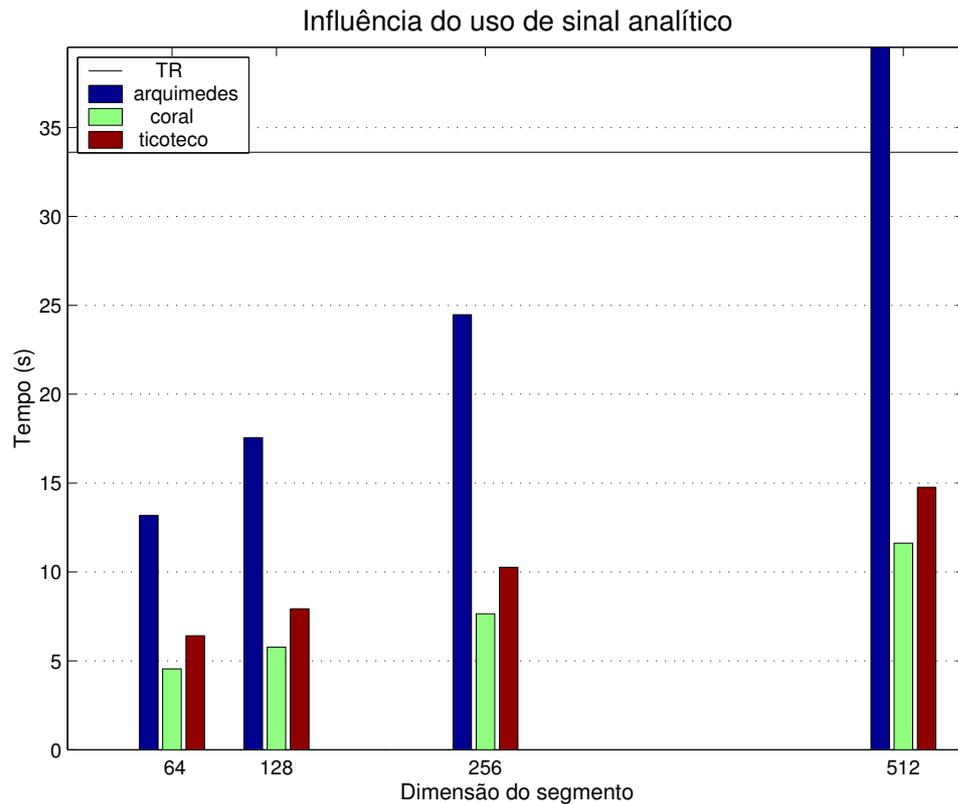


Figura 6.36: Tempos de execução da distribuição Bessel, nos computadores *arquimedes*, *coral* e *ticoteco*, considerando o cálculo do sinal *analítico* de segmentos não sobrepostos de 64, 128, 256 e 512 pontos. Linha horizontal representa o limite de tempo-real (33,61 s).

O parâmetro α é o que mais influencia o comportamento da aplicação. A figura 6.37 apresenta os tempos de execução, em segundos, da implementação do algoritmo da distribuição Bessel no computador *arquimedes*. Foram considerados segmentos de dimensão 64, 128, 256 e 512 pontos e os valores do parâmetro α do núcleo de 0,5, 1, 2, 5, 10, 15 e 20. Ao gráfico foi adicionada uma linha represen-

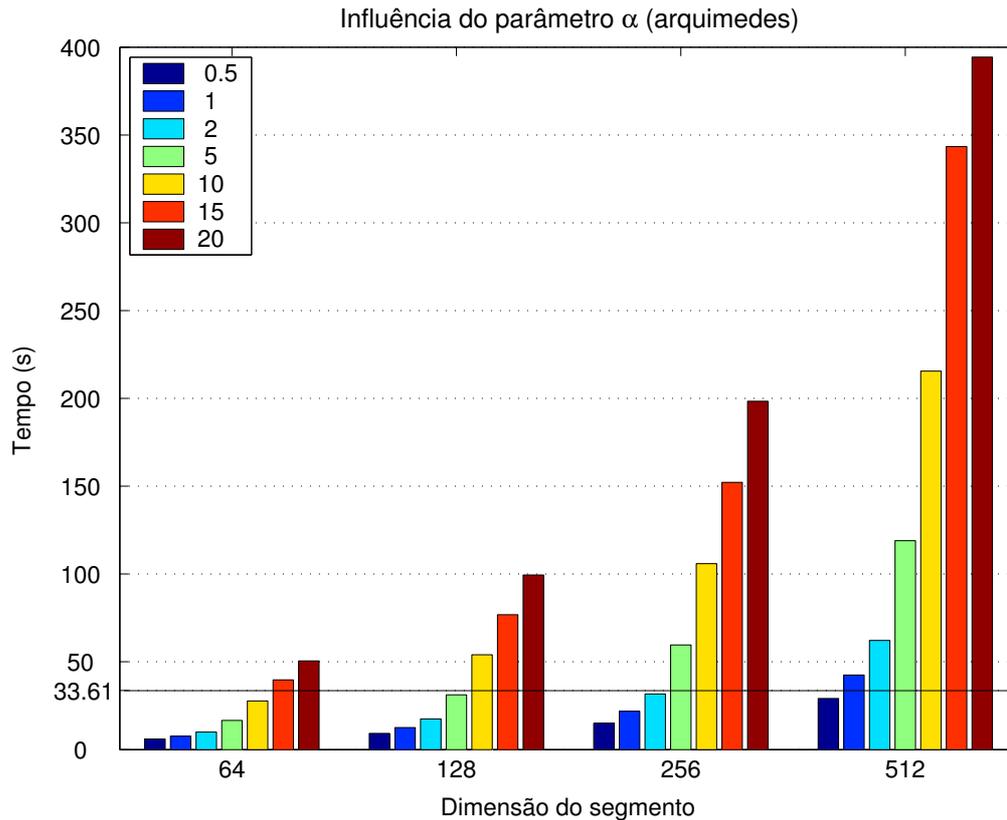


Figura 6.37: Tempos de execução da distribuição Bessel, no computador *arquimedes*, variando a dimensão do segmento e do parâmetro α do núcleo. Linha horizontal representa o limite de tempo-real (33,61 s).

tando o limite de tempo-real, neste caso, 33.61s. Como pode ser observado na Fig., esse limite é conseguido em menos de metade dos casos.

A figura 6.38 ilustra os valores registados nos computadores *coral* e *ticoteco*, com linhas e barras, respectivamente. Ao gráfico foi também adicionada uma linha representando o limite de tempo-real.

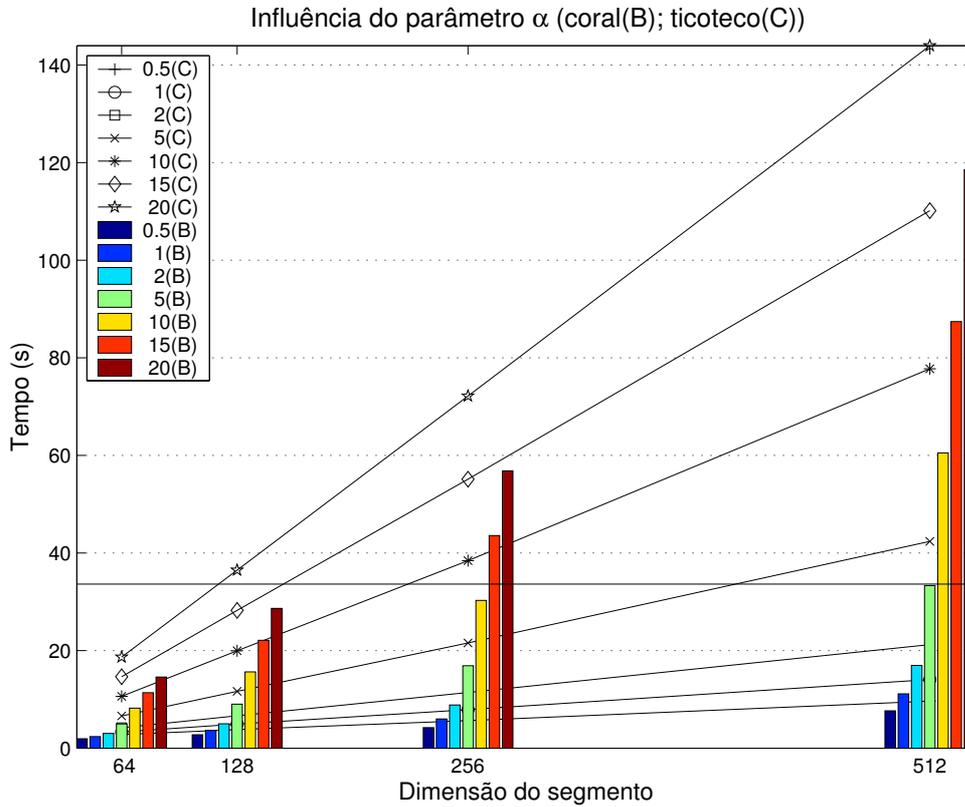


Figura 6.38: Tempos de execução da distribuição Bessel, nos computadores *coral* e *ticoteco*, variando a dimensão do segmento e do parâmetro α do núcleo. Linha horizontal representa o limite de tempo-real (33,61 s).

Para a menor dimensão dos segmentos, ambos os computadores são capazes de executar a estimação em tempo-real para os valores de α testados. Para sinais de fluxo sanguíneo da artéria carótida, em que $\alpha = 2$, é possível obter a estimação em tempo-real para todas as dimensões do segmento testadas.

É de notar que, com o aumento da dimensão dos segmentos, o computador *coral* apresenta melhores valores, apesar de o computador *ticoteco* dispor de mais memória. Neste par algoritmo/arquitetura, conclui-se que a abordagem de minimizar a carga computacional à custa de espaço de memória é adequada.

Na sequência deste estudo, foram estudados outros algoritmos que, pelo uso de memória estática ou partilhada, permitissem obter menores tempos de estimação. Nesse contexto foi ensaiada a possibilidade de armazenar a matriz de transformação do sinal *analítico* (*CSA*) num espaço de memória partilhada (ver secção 6.3.2.1). Essa alternativa não é a mais vantajosa para a implementação do sinal *analítico*, particularmente para o caso do ciclo cardíaco completo, pelo que foi preterida.

6.3.5.3 Comentários sobre a Implementação da BD

Vários estudos de outros autores demonstraram a superioridade da distribuição tempo-frequência Bessel na estimação de sinais não estacionários em diversos campos de aplicação. No caso particular da estimação de parâmetros de fluxo sanguíneo, a sua utilização tinha sido preterida devido ao custo computacional e a dificuldade na sua implementação em tempo-real.

No trabalho aqui descrito, foi dado ênfase ao caso particular de aplicação da distribuição à estimação de parâmetros do fluxo sanguíneo. Foram testadas várias máquinas e algoritmos, usando um vasto conjunto de possíveis parâmetros da aplicação, demonstrando-se assim, de forma concreta, que é possível utilizar a BD para a estimação em tempo-real.

6.3.5.4 Utilização do Estimador BD

Também neste caso, os parâmetros de chamada do programa permitem definir tipo e dimensão das janelas, sobreposição entre janelas, uso ou não de sinal *analítico* (e em caso afirmativo, de que tipo), e o valor do parâmetro α do núcleo. As diferentes opções são geridas pela janela de parametrização do método, conforme se apresenta na figura 6.39.



Figura 6.39: Definição das opções de execução do estimador BD.

6.4 Conclusão

Desde o início deste estudo, como consequência da evolução tecnológica e alteração da relação custo/capacidade de processamento, foi necessário rever a escolha de processadores a utilizar, para manter a actualidade e interesse do sistema desenvolvido. Como alternativa à utilização de plataformas aditivas foi estudado o desempenho de computadores pessoais, que actualmente dispõem de capacidade de processamento bastante elevada, e *software* livre, factores que contribuem para o menor custo do sistema global.

Os métodos de estimação espectral foram revistos, sendo generalizada a sua aplicação. Neste capítulo foram descritos os estudos e resultados da implementação dos métodos de estimação espectral considerados.

As abordagens à STFT e ao cálculo do sinal *analítico* foram estudadas no contexto da integração no sistema de processamento. Visando o apoio à validação clínica, e permitindo flexibilizar a utilização de sondas, quer a quadratura seja feita em hardware ou não, estes métodos, embora periféricos, são relevantes pelo impacto que têm nos demais estimadores.

Foi criado um algoritmo simplificado para o cálculo da matriz de Covariância Modificada, considerando a utilização de dados complexos, e procedeu-se à implementação do mesmo.

Para o caso da distribuição tempo-frequência de núcleos modificados, foi criada uma nova implementação, sendo descritos os elementos individuais e esboçada a interligação. No cálculo dos produtos cruzados usando os núcleos modificados, o alinhamento dos segmentos usados no cálculo da função de autocorrelação generalizada é fundamental para o desempenho da aplicação.

Foi criada a implementação da Bessel, em várias máquinas, permitindo concluir que o factor crítico de sucesso de esta implementação é velocidade de CPU e não memória.

Os estimadores concorrem com a aquisição/serviço de dados pelo uso de CPU, sendo usado um mecanismo de comunicação entre processos do linux.

É de salientar que, para o apoio à investigação em processamento de sinais, o tempo-real não reflecte a meta visada. Sendo frequentemente necessário proceder a volumes de processamento muito elevados devido ao teste de múltiplas alternativas, mesmo uma pequena melhoria em termos de tempo execução, é altamente desejável. Se a obtenção de resultados pode ser antecipada em alguns minutos ou horas, a frequência com que tais resultados são necessários traduz-se

na necessidade constante de mais velocidade e mais processamento.

A avaliação dos métodos face ao limite de tempo-real deve ser encarada de forma qualitativa, sendo satisfatório se esse limite for observado.

Outro aspecto a salientar do trabalho apresentado neste capítulo consiste na variedade de módulos, ou componentes do subsistema, existentes para alguns dos estimadores. Algumas alternativas são mais fáceis de manter que outras, mais flexíveis ou mais rápidas, mas de manutenção mais difícil. O carácter facultativo da utilização de uma das alternativas de um método, sem comprometer a independência do sistema, é fundamental para a universalidade e flexibilidade de aplicação pretendida.

No próximo capítulo, sob o tema de avaliação de desempenho, serão descritos alguns cenários em que, apoiando a investigação em processamento de sinal, o sistema contribuiu e beneficiou dessa mesma investigação.

Capítulo 7

Subsistema de Avaliação de Qualidade

7.1 Introdução

O subsistema de avaliação da qualidade visa fornecer indicadores de qualidade conforme o objectivo de utilização do sistema global.

A qualidade estatística surge, no contexto do subsistema de avaliação, na avaliação da estimação espectral, quando o sistema é usado na investigação de técnicas de processamento de sinal. Importa, nesse caso, quantificar a acuidade da estimação efectuada, seja de um ou, conjuntamente, de vários parâmetros espectrais. Os diferentes métodos de estimação podem ser avaliados comparando estatisticamente os valores estimados com valores teóricos utilizados na produção de sinais simulados. Consoante o objecto em análise e o estado representado, os indicadores clínicos relevantes (conforme descrito na secção 2.2.4) podem variar. Centrando a atenção no fluxo sanguíneo da artéria carótida, assumem especial relevância os indicadores velocidade média, turbulência e velocidade máxima, associados respectivamente aos parâmetros espectrais frequência média, largura de banda e frequência máxima.

A qualidade computacional surge, no contexto do subsistema de Avaliação, na avaliação da implementação, quando o sistema é usado na caracterização de fluxo sanguíneo podendo ser encarado como um protótipo de um sistema tempo-real de apoio ao diagnóstico clínico. Face à diversidade de processadores, de velocidades e arquitecturas distintas, o sucesso de uma implementação depende não só do algoritmo utilizado mas também da sua adequação à arquitectura. Inúmeras

métricas podem ser aplicadas na avaliação de desempenho de uma implementação. Considerando que o objectivo último é obter uma implementação em tempo-real, o tempo de execução é considerado o indicador decisivo na escolha de um par algoritmo-arquitectura. A temporalidade do tempo de execução numa família de processadores conduziu ao estudo de métricas alternativas que permitissem uma avaliação se possível intemporal de uma implementação.

7.2 Métodos de Avaliação de Desempenho

Na sua formulação original (Ruano & Fish, 1993), o método Custo/Benefício permite comparar e determinar qual o método que apresenta melhores estimativas (em termos de viés e desvio padrão), face à complexidade algorítmica do método utilizado na estimação. O critério permite quantificar a qualidade de uma implementação, podendo assim ser identificado não só o melhor método mas também, quando aplicável, os valores óptimos dos parâmetros dos métodos. Na quantificação do benefício da estimação podem ser considerados vários parâmetros espectrais sendo possível incluir factores de ponderação que reflectam a importância dos parâmetros em causa.

Para além da sua aplicação na escolha de um algoritmo de estimação espectral, este critério tem sido também aplicado, com pequenas variações, à avaliação de desempenho de estimadores em tempo-real. Registe-se a variação na aplicação do critério, referida em (Cardoso, 1998), em que se procura adequar o critério Custo/Benefício à avaliação de uma implementação substituindo, na função *custo*, a complexidade algorítmica por número de multiplicações. Ao longo do trabalho aqui descrito foram ensaiadas algumas variações, procurando adequar o critério Custo/Benefício à avaliação da qualidade computacional.

A primeira variação foi a utilização do tempo de cálculo em representação do custo. Como se observou na utilização das plataformas aditivas, essa abordagem não é conclusiva. O tempo de cálculo num dado sistema pode ser inferior ao tempo de amostragem dos dados e, ainda assim, não ser possível implementar um sistema tempo-real, devido à contenção imposta pela leitura e escrita de dados. Em alternativa, passou a ser usado o tempo de execução, incluindo transferência de dados.

Sucedem também, na escolha de arquitecturas, serem comparados processadores de diferentes velocidades. A relação entre tempo de execução e a frequência

de relógio de um processador invalida quaisquer conclusões sobre a arquitetura. Assim, ensaiou-se a utilização de um indicador de desempenho alternativo que permita não só avaliar a adequação à arquitetura como também estimar o tempo de execução numa arquitetura da mesma família. O indicador escolhido foi o número de ciclos, obtido pela normalização do tempo de execução com a frequência de relógio do processador. A adopção do tempo de execução com a opção de inclusão dos tempos de leitura e escrita de dados bem como um factor normalizador da velocidade do processador, utilizado em (Leiria, Madeira & Ruano, 1999), permite uma análise mais detalhada da implementação em estudo sendo nesse caso, a função *custo* definida por

$$\begin{aligned} \text{custo}(W_c, t_{in}, t_{exe}, t_{out}, nf) &= \\ &= \frac{w_{t_{in}} t_{in} + w_{t_{exe}} t_{exe} + w_{t_{out}} t_{out}}{nf} \end{aligned} \quad (7.1)$$

onde t_{in} é o tempo de leitura de dados, t_{exe} é o tempo de execução, t_{out} é o tempo de escrita de dados e nf é um factor de normalização que traduz a conversão das medições temporais em número de ciclos do processador. Tal como na função *benefício*, W_c são factores de ponderação dos parâmetros de custo devendo totalizar 100%. Considere-se ainda a escalabilidade do critério. Sendo a carga, traduzida pelo tempo total normalizado pela velocidade do processador, a capacidade de trabalho do processador da família no algoritmo em causa, é possível prever a velocidade necessária para uma execução em tempo-real para algoritmos regulares.

Outra alternativa, utilizada em (Leiria, Madeira & Ruano, 2000), consiste na normalização da função Custo/Benefício usando como referência os valores de Custo e Benefício da implementação da STFT no sistema em estudo

$$C/B = \frac{\frac{\text{custo}_i}{\text{custo}_{ref}}}{\frac{\text{Benefício}_i}{\text{Benefício}_{ref}}} \quad (7.2)$$

permitindo a comparação dos pares método/sistema usado. Nesta variante o custo é representado pelo tempo total da implementação, conseguindo-se normalizar as grandezas em causa. Esta abordagem mais simples permite rapidamente comparar formas alternativas de implementação de um algoritmo, enfatizando diferenças numéricas das implementações dos métodos.

A aplicação do critério em duas fases, avaliando separadamente custo e benefício, tem também sido realizada por permitir hierarquizar os métodos de acordo com o benefício que conferem e, consoante o equipamento disponível à data de uma implementação, escolher a implementação que melhor desempenho oferece.

7.2.1 Avaliação Custo/Benefício

Foram escolhidos três estudos que exemplificam a importância da inclusão do subsistema de Avaliação de Qualidade e ilustram a importância das variações introduzidas ao critério Custo/Benefício. Esses estudos, desenvolvidos no laboratório sob a coordenação da Prof. Doutora Maria da Graça Ruano, contaram também com a participação da Eng^a Ana Isabel Leiria, que, à data dos dois primeiros, estudava a estimação de parâmetros do fluxo sanguíneo em válvula aórtica.

O primeiro desses estudos, que designaremos por Caso 1, visava a escolha do par algoritmo/arquitetura mais adequado à estimação espectral de fluxo sanguíneo em válvula aórtica(Leiria, Madeira & Ruano, 1999).

O segundo estudo, Caso 2, visava comparar a utilização de distribuições tempo-frequência de Choi-Williams de núcleos modificados(Leiria, Moura & Ruano, 2001), também na estimação de parâmetros do fluxo da válvula aórtica.

O Caso 3 incide sobre a utilização do sinal *analítico* associada aos estimadores. O cálculo da forma analítica de segmentos do sinal é uma herança dos compromissos feitos, em favor do tempo-real e à custa da qualidade de estimação.

As conclusões deste caso de estudo motivaram o estudo das implementações do cálculo do sinal *analítico* e do método AR-MC sobre dados complexos (secção 6.3.3).

7.2.1.1 Caso 1

A utilização de um Pentium MMX a 350 MHz com os sistemas operativos Microsoft Windows NT e Debian/GNU Linux foi comparada com a utilização de um computador pessoal equipado com um Pentium e uma C40, usando os métodos STFT, AR-MC e CWD, na estimação de sinais simulados da válvula aórtica com três valores da relação sinal-ruído (10, 20, ∞ dB). Em termos globais, o estudo visava determinar, em termos de qualidade estatística e computacional, o desempenho dos estimadores na presença de ruído e a plataforma mais adequada,

respectivamente. Do ponto de vista do sistema de processamento de sinal, objecto desta tese, pretendia-se comparar o desempenho dos estimadores AR-MC e CWD no Pentium sob Linux e na DSP C40.

As condições de aplicação dos métodos basearam-se num estudo da estimação de sinais de válvula aórtica (Leiria, 2000). Nesse estudo tinham sido determinados os tipos e tamanhos das janelas temporais a aplicar, o número de pólos do estimador AR-MC e os parâmetros do núcleo da CWD. Registe-se que apenas foi necessário, neste caso, alterar um dos parâmetros do núcleo da CWD, face aos valores usados para a artéria carótida. Com o simulador IPISBio (Vistas & Reis, 1999) foram criados sinais de ciclos cardíacos de 640 ms. de duração, cada um dos quais foi, pela aplicação de janelas não sobrepostas de 5ms., dividido em 127 segmentos.

No sistema operativo Microsoft Windows NT foram utilizadas implementações dos métodos em Matlab (a que se referirá por Matlab), enquanto que no sistema operativo Debian/GNU Linux as implementações foram compiladas com GNU C Compiler sendo executadas no processador Pentium (a que se referirá por Pentium) e as implementações na DSP C40 (que se referirá por DSP) utilizou-se o compilador 3L Parallel C para a produção das aplicações. As implementações no Pentium em Linux e na DSP C40 foram optimizadas para os parâmetros dos métodos em estudo, seguindo as definições em vigor para as plataformas aditivas.

Foram utilizadas as implementações dos métodos de estimação em Matlab existentes no laboratório, de vários autores, com o objectivo de permitir a aferição de resultados. A versão modificada do critério Custo/Benefício, concebida pelas três autoras da comunicação, consistiu na redefinição da função *custo*. A implementação em Matlab foi da autoria da Eng^a Ana Isabel Leiria. Como medida de custo, foram considerados os tempos de cálculo e os tempos de leitura de dados e escrita de resultados; para abstrair a diferença de velocidade entre a C40 e o Pentium, foi adoptada uma métrica que designaremos por ciclos e corresponde à multiplicação dos tempos medidos pela frequência de relógio do processador em causa (cf. Eq. 7.1). Assim, o processador Pentium é penalizado sete vezes mais por ser sete vezes mais rápido que a DSP C40.

O custo das implementações é ilustrado na figura 7.1. A análise da figura revela que a plataforma mais vantajosa é conseguida utilizando no processador Pentium a aplicação compilada com *gcc* sobre o sistema operativo Debian/GNU Linux, sendo também a solução de menor custo económico.

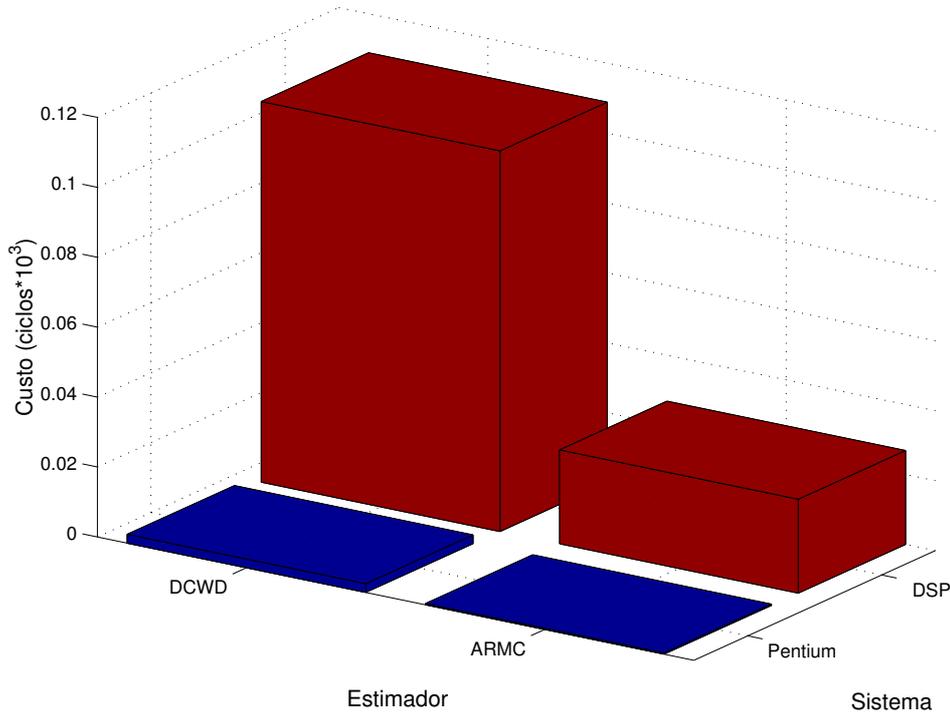


Figura 7.1: Custo das implementações dos métodos AR-MC e CWD em distintas plataformas.

Foi também interessante notar diferenças significativas no benefício das implementações de um mesmo método. Sem entrar em detalhes sobre a estimação de parâmetros de sinais de fluxo sanguíneo na válvula aórtica, fora do âmbito deste trabalho, refiram-se apenas os resultados do Benefício na estimação do parâmetro largura de banda, ilustrados na tabela 7.1. Sendo natural algumas

	STFT	AR-MC	CWD
Pentium	1,91	2,41	2,59
DSP	1,91	2,41	2,4
Matlab	1,91	2,41	2,6

Tabela 7.1: Benefício na estimação da largura de banda de sinais de válvula aórtica com SNR de 10 dB.

diferenças numéricas nos resultados das diferentes implementações, normalmente estas não afectam a qualidade da estimação. Neste caso, notam-se diferenças no benefício da estimação do método CWD. A qualidade de estimação na DSP C40 é afectada pelo facto de as divisões serem efectuadas por inversão do divisor seguida de multiplicação; a estimação no Pentium em Linux apresenta menor benefício por serem usado dados de precisão simples e não dupla. Foi avaliada a qualidade

da estimação caso fosse adoptada a dupla precisão, desvanecendo-se o impacto no benefício, nesse caso a maior custo.

Considerando apenas a qualidade estatística, escolher-se-ia a implementação da CWD sobre o Pentium em Linux; atendendo a ambos os critérios, qualidade da implementação (aqui conotada com o número de ciclos) e qualidade estatística, escolher-se-ia a implementação do estimador AR-MC.

Relembrando que a motivação principal para a participação neste estudo foi a comparação das plataformas, os resultados da Fig. 7.1 indicam claramente que é preferível a utilização de máquinas de propósito geral. Isto não traduz um juízo de valor sobre os processadores envolvidos, traduz sim o facto de a transferência de dados entre a DSP e a máquina hospedeira passar a ser considerada na avaliação da arquitectura, transferência essa que inviabiliza a utilização da DSP C40 num sistema tempo-real com esta aplicação.

7.2.1.2 Caso 2

A implementação, no computador *ticoteco*, da distribuição CW de núcleos modificados foi utilizada na avaliação da estimação de parâmetros de fluxo sanguíneo que simulam os da válvula aórtica com três valores da relação sinal-ruído (10, 20, ∞ dB). A avaliação da qualidade de estimação foi efectuada utilizando a função normalizada de Custo/Benefício (Eq. 7.2), concebida pelas três autoras da comunicação, e, também neste caso, implementada em Matlab pela Eng^a Ana Isabel Leiria.

O critério Custo/Benefício foi aplicado em duas fases: análise do benefício para identificar quais os parâmetros óptimos para a distribuição em cada uma das suas formas, com núcleos *standard*(STD), *Reduced Interference*(RI), *Alias Free*(AF) e *Reduced Interference Alias Free*(RIAF); análise de Custo/Benefício para a melhor configuração.

A figura 7.2 apresenta os benefícios normalizados da CWD com os núcleos STD, RI, AF e RIAF considerando os valores óptimos em cada caso. O melhor benefício é obtido com a utilização do núcleo AF na estimação da frequência média, e com o núcleo STD na estimação da largura de banda e na estimação de ambos os parâmetros. Note-se que o benefício da estimação da frequência média é inferior a um, ou seja, inferior ao obtido com a STFT. De facto, esta distribuição distingue-se pela qualidade de estimação da banda, o que verifica para todas as formas com excepção do RIAF.

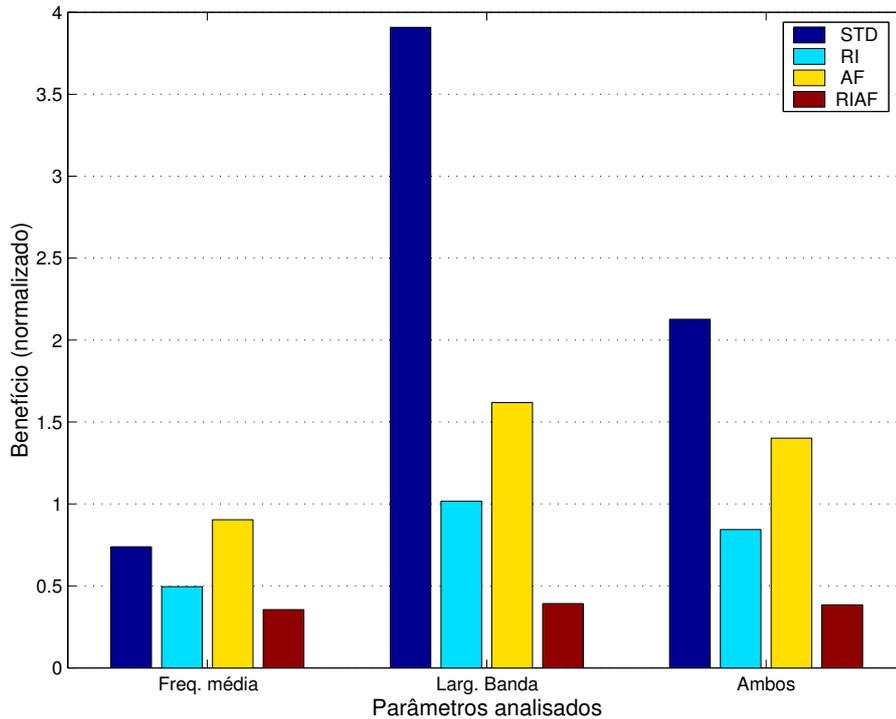


Figura 7.2: Benefício normalizado da CWD com núcleos STD, RI, AF e RIAF na estimação de sinais de válvula aórtica.

Sendo concluído que, para os sinais em causa, a distribuição com o núcleo STD seria a mais adequada por produzir o melhor benefício na estimação conjunta da frequência média e largura de banda, procedeu-se a uma análise detalhada da relação Custo/Benefício para a CW-STD, apresentada na figura 7.3. Consideraram-se valores de m entre um e 100 porque a qualidade da estimação com a CW-STD varia com o parâmetro m , quer em termos de custo como de benefício, como é patente na fig. 7.3.

Com esta abordagem Custo/Benefício é possível escolher a implementação em tempo-real que, para um dado equipamento, maior benefício confere à estimação, neste caso com $m = 20$, ou, fazendo a escolha para o valor óptimo de Custo/Benefício, escolher $m = 5$.

Relembrando que a participação neste estudo visava determinar se a qualidade estatística da estimação justificava o estudo mais aprofundado das implementações de núcleo modificado, o que, neste caso, não se verificava, tal estudo foi adiado.

Neste caso, não sendo estudado o tamanho óptimo das janelas, o tipo de janela ou a taxa de sobreposição, procedeu-se à avaliação do benefício de 50 estimativas

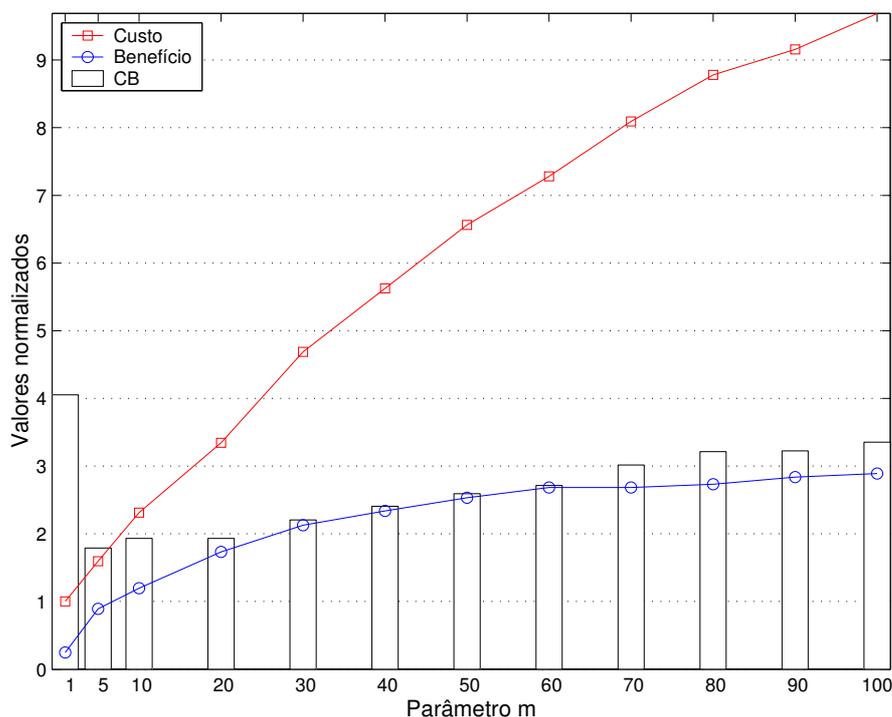


Figura 7.3: Custo/benefício, custo e benefício normalizados da CWD com núcleo STD na estimação de sinais de válvula aórtica.

do espectro de cada um dos três conjuntos de 100 sinais, tendo sido testados os valores de $\sigma = 0.1, 0.3, 0.6, 1, 2, 6, 10, 15, 20$ e, fixando o σ , para STD e AF, os valores de m já referidos. É de notar que cada um destes estudos é muito moroso e requer muitos recursos humanos, sendo notada a necessidade de criar um procedimento que permita a sua automatização.

7.2.1.3 Caso 3

O Caso 3 incide sobre a utilização do sinal *analítico* associada aos estimadores. A abordagem tradicional (Boashash & Black, 1987) para a implementação em tempo-real das distribuições tempo-frequência, considera o cálculo do sinal *analítico* de cada segmento. Em termos teóricos, o sinal *analítico* deve ser calculado para o sinal completo (Cohen, 1995). A questão que se colocava era, tendo o corte que ser feito, que diferença faria considerar o ciclo cardíaco ou segmentos desse ciclo. Procurou-se a resposta analisando a qualidade da estimação sem a utilização de sinal *analítico* – cenário A – e aplicando-o a segmentos – cenário B

– ou a ciclos cardíacos completos – cenário C. Foram utilizados 100 sinais correspondentes a 100 ciclos cardíacos de 640 milisegundos cada, sendo o limite de tempo-real de 64 segundos.

Os estimadores e respectivos parâmetros para sinais de fluxo sanguíneo em artéria carótida foram escolhidos em função de diversos estudos:

para o estimador STFT considerou-se uma resolução temporal de 1.5625 ms. correspondendo a 20 pontos e uma resolução na frequência de 100 Hz correspondendo a segmentos de 128 pontos (Cardoso, Ruano & Fish, 1996a); para o estimador AR-MC adoptou-se a mesma resolução temporal e na frequência, usando um modelo de quarta ordem quando aplicado a sinais reais (Ruano, 1992; Wang & Fish, 1996) e de segunda ordem quando aplicado ao sinal *analítico* (Picone, Prezas, Hartwell & LoCicero, 1988); para o estimador CWD, a resolução temporal escolhida foi a mesma, sendo a resolução na frequência de 25 Hz correspondendo a 512 pontos, e os parâmetros do núcleo STD $\sigma = 5$ e $m = 30$ (Cardoso, Ruano & Fish, 1996a); na aplicação do CWD a sinais reais, considerou-se um sinal interpolado simulando a amostragem ao dobro da frequência de Nyquist (Boashash & Black, 1987).

A qualidade estatística da estimação foi avaliada nos três cenários em termos de raiz do erro médio quadrático ($rmse^1$), função inversa do benefício. As tabelas 7.2, 7.3 and 7.4 apresentam os valores obtidos com os estimadores STFT, AR-MC e CWD, respectivamente.

	Cenário A	Cenário B	Cenário C
Parametro(s)			
f_m	44,062	44,208	43,0527
bw	261,731	259,01	222,3358
$f_m + bw$	305,793	303,2200	265,3885

Tabela 7.2: Desempenho do estimador STFT na estimação dos parâmetros f_m e bw , em termos de $rmse$ (Hz).

No caso do estimador STFT, verifica-se uma melhoria superior a 10% na estimação conjunta dos dois parâmetros.

É de notar que no caso do estimador AR-MC que o cenário B permite uma pequena diminuição do erro na estimação da frequência média penalizando fortemente a estimação da largura de banda e a estimação conjunta dos dois parâmetros. No cenário C verifica-se que a qualidade da estimação não se deteriora.

¹em inglês, *root mean square error*

	Cenário A	Cenário B	Cenário C
Parametro(s)			
f_m	169,9896	157,9251	168,9491
bw	27,8032	198,9254	26,6473
$f_m + bw$	197,7928	356,8505	195,5964

Tabela 7.3: Desempenho do estimador AR-MC na estimação dos parâmetros f_m e bw , em termos de rmse (Hz).

	Cenário A	Cenário B	Cenário C
Parametro(s)			
f_c	104,6553	134,6464	134,6313
bw	205,9762	14,0217	13,3549
$f_c + bw$	310,6314	148,6681	147,9862

Tabela 7.4: Desempenho do estimador CWD na estimação dos parâmetros f_m e bw , em termos de rmse (Hz).

O cenário A apresenta resultados muito aquém do esperado, o que é atribuído à interpolação utilizada. O resultado da estimação é semelhante nos cenários B e C.

Como se pode verificar pela análise das tabelas, o cenário mais vantajoso é o C, apresentando menores valores de erro na estimação conjunta dos parâmetros frequência média e largura de banda. Ainda que não se traduza numa melhoria significativa da qualidade estatística da estimação, não a prejudicam.

Atendendo à importância das implementações dos métodos AR-MC e CWD neste estudo, e a que as implementações do estimador CWD consideravam já dados complexos, concluiu-se ser apropriado procurar diminuir o custo da implementação do estimador AR-MC aplicado a dados complexos e estudar formas alternativas de proceder ao cálculo do sinal *analítico* para ciclos cardíacos completos.

Para o estudo do custo computacional das implementações do estimador AR-MC nos três cenários, foram utilizadas aplicações separadas. O cenário A considerou a estimação do sinal real; o cenário B considerou o cálculo do sinal *analítico* de cada segmento e a conseqüente estimação; o cenário C considerou outras duas aplicações, para a criação do sinal *analítico* de cada ciclo cardíaco e a estimação de sinais de tipo complexo.

Foram avaliadas as implementações criadas na redefinição dos métodos (cf. descrito no início do capítulo 6). Os tempos médios, obtidos na máquina *coral*

com o comando *time* em segundos, correspondentes à estimação dos 100 sinais foram 10,07 s, 21,28 s e 19,60 s, para os cenários A, B e C, respectivamente.

Embora a estimação seja obtida em tempo-real neste caso particular, os resultados dos cenários B e C foram reavaliados após o estudo das implementações do sinal *analítico* e do algoritmo de cálculo da matriz complexa de covariância modificada (descritos nas secções 6.3.2 e 6.3.3, respectivamente). O custo dos diferentes cenários é apresentado na tabela 7.5.

	Cenário A	Cenário B	Cenário C
MC	10,07		
SA/segm+MC		7,10	
SA+MC			7,54

Tabela 7.5: Custo (em segundos) das implementações do estimador AR-MC.

Comparando com a implementação original, sobre dados reais, verifica-se uma redução dos custos dos cenários B e C de 29,49% e 25,12%, respectivamente. Considerando apenas o custo computacional dos diferentes casos, o cenário B, a que corresponde o menor tempo total, deveria ser adoptado.

Considerando que, se se avaliar apenas o benefício, deveria seguir-se a metodologia representada pelo cenário C, e se se avaliar apenas o custo, deveria seguir-se o cenário B, foram analisados os valores do critério Custo/Benefício nos três cenários, considerando a estimação separada e conjunta dos parâmetros espectrais frequência média e largura de banda. Esses valores são representados na figura 7.4, tendo sido acrescentadas setas identificando o melhor cenário em cada caso.

Se se pretender estimar apenas a frequência média, o cenário B representa a abordagem mais conveniente; para a estimação da largura de banda ou ambos os parâmetros espectrais, deve ser utilizada a abordagem seguida no cenário C.

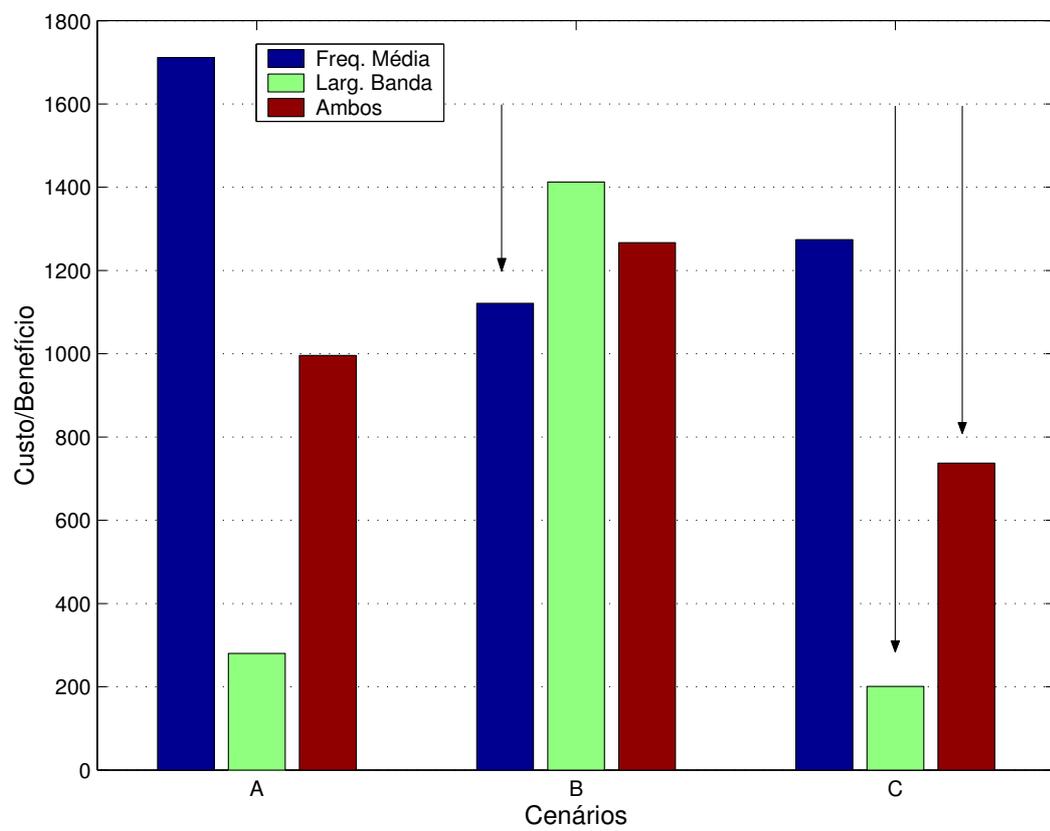


Figura 7.4: Custo/Benefício das implementações do estimador AR-MC.

7.3 Conclusão

O subsistema de avaliação de qualidade, baseado no critério Custo/Benefício, considera a qualidade da estimação espectral e a qualidade da implementação de um estimador, separada ou conjuntamente.

Foram introduzidas algumas alterações, em relação à sua formulação original, à função Custo e ao critério na sua globalidade.

O custo de uma implementação, anteriormente definido em função da complexidade algorítmica ou em função do número de multiplicações de um dado algoritmo, passou a ser quantificado através do tempo de execução total, ou parcial, de uma implementação de um estimador. Desta forma, para além de poder ser aplicado o critério à comparação de pares algoritmo/arquitetura, o custo pode ser avaliado em função dos requisitos de resposta temporal do sistema.

Foi também introduzida a possibilidade de ponderar o custo de uma implementação com a velocidade do processador, e, pela abstracção dessa característica, comparar o comportamento global de uma implementação numa arquitectura.

Outra alteração introduzida consiste na possibilidade de normalizar os indicadores de custo e benefício, tornando-os mais “tangíveis”. Essa alteração assume particular relevância na aplicação do critério em duas fases, analisando separadamente o benefício e o custo.

A versatilidade e utilidade do critério Custo/Benefício foi ilustrada em três casos. O Caso 1 comparou implementações de um mesmo algoritmo em diferentes arquitecturas, nomeadamente plataformas aditivas e processadores de propósito geral; o Caso 2, variantes de um método de estimação numa mesma arquitectura; o Caso 3, em duas fases, formas alternativas de aplicação de estimadores e, para um destes, mais detalhadamente, implementações alternativas.

A avaliação de qualidade do sistema, baseada no critério Custo/Benefício, dos métodos de estimação e implementações, assume um carácter preponderante para, de forma pragmática, esclarecer a linha de desenvolvimento e fundamentar a divergência de algumas abordagens tradicionais.

As conclusões retiradas do Caso 1 e o custo de actualização das plataformas aditivas motivaram o abandono dessas plataformas a favor da utilização de processadores de uso geral. Os resultados do Caso 2, pelo contrário, não motivaram um estudo mais aprofundado das distribuições CWD de núcleo modificado. Esse estudo está adiado até que surja uma situação em que o uso de um dos núcleos

modificados seja vantajoso em termos da qualidade da estimação espectral e, pelo conjunto de parâmetros a utilizar, não se cumpram os limites de tempo-real. Os resultados do Caso 3 motivaram um estudo aprofundado do algoritmo a usar para o estimador AR-MC e a re-equacionar a utilização do sinal *analítico*, anteriormente aplicado a segmentos do ciclo cardíaco e apenas nos estimadores tempo-frequência.

Pela relevância do critério Custo/Benefício no apoio à investigação em processamento de sinal e ao desenvolvimento computacional do sistema, seria conveniente que produção das medidas de custo e benefício fosse feita de forma automática, sendo mantidos registros que facilitassem a sua análise posterior. A sua integração no protótipo do sistema está prevista, aguardando-se a conclusão de estudos sobre a caracterização de sinais clínicos de fluxo sanguíneo (tema de doutoramento de uma colega de equipa), pois a determinação do benefício de uma estimação só é possível se forem conhecidas as características (parâmetros espectrais) do sinal estimado.

Capítulo 8

Comentários Finais

8.1 Introdução

Nos capítulos anteriores foram apresentados os trabalhos desenvolvidos para concretizar cada um dos subsistemas, minimizando o impacto das escolhas em termos da funcionalidade e generalidade do sistema global.

Relembrando que o sistema desenvolvido assume duas funções principais, a caracterização de fluxo sanguíneo e o apoio à investigação em processamento de sinal de fluxo sanguíneo, é agora possível tecer algumas considerações gerais sobre as estratégias adoptadas no desenvolvimento e integração do sistema, no intuito de clarificar os motivos conducentes à adopção dessas mesmas estratégias.

Neste contexto são referidos alguns aspectos relevantes dos estimadores espectrais considerados e os factores determinantes da presente configuração do sistema.

A título conclusivo, são apresentadas as considerações finais sumariando as conclusões deste trabalho e algumas linhas de acção para a melhoria do sistema de análise de fluxo sanguíneo.

8.2 Estimadores Espectrais

O sinal de fluxo sanguíneo obtido por instrumentação ultrasónica, como outros sinais fisiológicos, é não estacionário e variante no tempo. As características deste sinal são distintas consoante o vaso ou válvula a analisar e, fundamentalmente, com o quadro clínico da pessoa observada (Wells, 1998; Evans & McDickens, 2000). Ainda que considerando um mesmo indivíduo, a variância no tempo

é intrínseca, podendo aumentar com a condição fisiológica da pessoa observada no momento particular do exame (como tensão arterial), ou com factores externos (como temperatura ambiente). Independentemente do motivo, esta variância afecta as características do sinal espectral a analisar, podendo, em caso de omissão de consideração, conduzir a conclusões erróneas sobre o estado do fluxo sanguíneo no vaso em estudo.

A aplicação de um estimador espectral num caso particular depende das características do sinal temporal de entrada no sistema e dos indicadores que se pretendem avaliar.

8.2.1 Indicadores Clínicos e Parâmetros Espectrais

O sangue circula impulsionado pelo coração. A partir da válvula aórtica, a artéria aorta, com maior fluxo e de maior dimensão, vai-se ramificando, levando o sangue às diferentes partes do corpo. O sangue é expelido do coração com alta pressão e conseqüentemente de forma turbulenta. A pressão e turbulência vão então diminuindo com o aumento da distância à válvula aórtica. Além disso, consoante a zona irrigada, pode haver fluxo durante todo o ciclo cardíaco ou não; é também possível que a direcção do fluxo se inverta durante o ciclo cardíaco.

A título de exemplo refira-se que, em pessoas sãs, a turbulência sempre presente na válvula aórtica é menor em artéria carótida primitiva e ainda menor em artéria cerebral média; na artéria carótida primitiva existe sempre fluxo sanguíneo bem como nas artérias carótida interna e cerebral média; no entanto, durante parte do ciclo cardíaco, na artéria femoral o fluxo muda de direcção e na artéria carótida externa pode não haver fluxo.

Sendo a velocidade do sangue proporcional à frequência Doppler, a velocidade máxima é obtida a partir do parâmetro espectral frequência máxima; a velocidade média do sangue no volume em análise é obtida pelo parâmetro frequência média; a gama de velocidades (e correspondentes frequências Doppler), traduzida pelo parâmetro largura de banda, pode ser interpretada como uma medida da turbulência. A reflexão do ultra-som, determinada pela natureza, dimensão e quantidade das partículas em movimento no volume amostral considerado, é traduzida pela energia do sinal e corresponde à potência espectral.

O objectivo de caracterização global de fluxo sanguíneo implica que as diferentes artérias sejam consideradas, contemplando anomalias do fluxo sanguíneo ou patologias associadas à detecção de estenoses e embolias.

De forma geral, a curva de frequência máxima do espectrograma permite uma observação rápida para indicar suspeita de fluxo anómalo, indiciando a necessidade de cálculo de alguns indicadores clínicos; nomeadamente, a análise da frequência média e largura de banda permitem a identificação de estenoses e a sua classificação de acordo com os valores assumidos pelos indicadores; a análise da potência é efectuada na determinação de embolias e, em conjunto com a frequência média, na classificação de embolias (Wells, 1998; Smith, Evans, Lingke, Bell & Naylor, 1996).

Tendo em vista a determinação do espectrograma, a conjugação das características do vaso com as da efermidade podem levar à escolha de diferentes métodos e/ou de diferentes parâmetros dos métodos.

Assim, se se compararem vasos como a artéria aorta (junto à saída do coração), a artéria femoral, a artéria carótida e a artéria cerebral média pode-se verificar que a artéria aorta e a femoral apresentam variações bruscas dos parâmetros em frequência e mesmo em potência implicando que um método apropriado para sinais não estacionários seja o mais indicado (Leiria, Madeira & Ruano, 1999; Guo, Durand & Lee, 1994a) enquanto que na carótida primitiva ou na artéria cerebral média esse factor pode não ser tão evidente (Cardoso, Ruano & Fish, 1996a; Fan & Evans, 1994).

O facto de o vaso apresentar indícios de estenose pode levar a que as distribuições tempo-frequência voltem a ser as mais indicadas. Neste sentido, e de acordo com as características dos vasos, também os parâmetros dos vários métodos podem ser diferentes. Variações mais bruscas dos parâmetros espectrais de interesse clínico podem significar a necessidade de maior resolução temporal e conseqüente modificação dos parâmetros de funcionamento optimizado dos algoritmos de cálculo do espectro.

Se um sinal apresenta ruído então pode ser preferível a utilização de métodos paramétricos, cuja estimação se baseia em observações fora do intervalo de observação, ou de métodos tempo-frequência, que permitem suavizar o espectro do sinal.

8.2.2 Comparação do Desempenho dos Métodos de Estimção Espectral

Como a escolha do método de estimção espectral é função das características do método, apresenta-se uma discussão das vantagens e desvantagens relativas dos métodos implementados, baseada nas suas propriedades de estimção dos parâmetros clinicamente relevantes. Cada um dos métodos foi descrito separadamente na secção 2.3.

A aplicação da DFT a segmentos curtos de um sinal produz o espriamento do espectro e conseqüente perda de potência na distribuição da potência em estudo e, pelo compromisso entre a resolução no tempo e na frequência, dá origem a um espectrograma de baixa resolução na frequência. O compromisso entre a resolução no tempo e na frequência pode ser minimizado pela sobreposição das janelas que segmentam o sinal de ciclo cardíaco. Outra abordagem possível é a aplicação de métodos da classe de Cohen, que não a STFT. No entanto, estes métodos têm alguns inconvenientes: do ponto de vista da estimção espectral, criam termos cruzados, isto é, o espectro apresenta energia a frequências que não existiam no sinal temporal podendo comprometer a qualidade dos parâmetros espectrais a estimar mediante a inclusão de artefactos; do ponto de vista da implementação computacional, são significativamente mais exigentes que a DFT, requerendo especiais cuidados para se manter compromissos de tempo-real.

A STFT, que também pode ser interpretada como uma distribuição tempo-frequência da classe de Cohen, permite em alguns casos obviar o problema dos termos cruzados. Dependendo das características do sinal, nomeadamente da localização dos autotermos, a utilização do espectrograma pode conduzir à eliminação dos termos-cruzados, ou, a escolha correcta da função janela pode reduzir significativamente a ocorrência dos termos cruzados (Jeong & Williams, 1990; Jeong & Williams, 1992c).

Tal como a STFT, o método AR-MC (paramétrico) também se destina a sinais estacionários. No entanto, aplicando-se o AR-MC a janelas sobrepostas de segmentações do ciclo cardíaco, de forma semelhante à usada na STFT e com as mesmas desvantagens, é possível a sua aplicação a sinais de características não estacionárias, ou seja, recorrendo a janelas de duração superior a 20 milisegundos. Em geral, a AR-MC permite obter bons resultados quanto à resolução na frequência. A principal dificuldade consiste em preservar a informação, quer da potência instantânea (variação da potência ao longo do tempo), quer da potência

média, quando se estudam sinais com características não estacionárias (Pola, Macerata, Emdin & Marchesi, 1996).

As distribuições tempo-frequência da classe de Cohen têm como grande vantagem o facto de serem apropriadas para sinais não estacionários; a desvantagem que normalmente se lhes aponta é a carga computacional que envolvem (Boashash & Black, 1987; Cardoso, 1998). No entanto, distribuições como a STFT (ou a Wigner-Ville, não implementada) apresentam problemas quando são usadas em sinais multicomponentes: a ocorrência de termos cruzados. Os termos cruzados e os auto-termos do sinal podem ocorrer em variadas combinações. Nomeadamente, os termos cruzados podem ocorrer na posição dos autoterms e "mascará-los".

A quantidade e forma desta interferência está directamente relacionada com a função de núcleo usada na distribuição tempo-frequência.

O núcleo exponencial introduzido em (Choi & Williams, 1989) permite controlar o efeito dos termos cruzados preservando a forma dos autoterms consoante as características do sinal em análise. No entanto, a CWD não satisfaz completamente as propriedades de suporte no tempo e na frequência. Estas propriedades garantem que se não existir sinal temporal num dado intervalo de tempo a distribuição é nula nesse intervalo e se uma gama de frequências não existir a distribuição é nula para essa gama.

Em alguns casos, a distribuição Bessel (Guo, Durand & Lee, 1994b) satisfaz essas propriedades de suporte e apresenta algumas vantagens em relação à CWD no que diz respeito à estimação da frequência média. Algumas limitações das distribuições podem ser contornadas utilizando núcleos modificados, para redução de interferência ou eliminação de *aliasing*, mas a sua utilização deve ser avaliada em cada caso particular.

Embora vocacionadas para sinais não estacionários, as distribuições tempo-frequência são mais sensíveis aos seus parâmetros típicos que a STFT ou a AR-MC na estimação de sinais Doppler de fluxo sanguíneo.

8.2.3 Informação Clínica

Dependendo da artéria e do tipo de patologia em causa os indicadores a usar podem variar e, conseqüentemente, o método de estimação a escolher é dependente da sua capacidade em produzir estimadores dos parâmetros espectrais com maior acuidade.

Para rápidas variações na frequência, sendo o fluxo sempre positivo, a CWD

pode ser uma boa opção. Nesse caso, os parâmetros σ e m têm que ser testados para os sinais do vaso em estudo, e, posteriormente efectuados os cálculos dos indicadores de interesse clínico. Caso as variações na frequência sejam lentas e o fluxo possa mudar de direcção durante o ciclo cardíaco, a Bessel é provavelmente a melhor escolha, obrigando à prévia identificação do par Bessel-vaso sanguíneo.

Perante um sinal mais estável, como o obtido por Doppler transcraniano, a escolha pode oscilar entre a AR-MC ou STFT: caso o fluxo seja normal, o indicador frequência máxima é suficiente para a indicação da falta de patologia, pelo que o comprimento da janela a aplicar a cada ciclo cardíaco em análise, a taxa de sobreposição dos segmentos do sinal estacionários, e, no caso AR-MC, a ordem do modelo paramétrico, são requeridos para o cálculo da curva de frequência máxima de cada ciclo cardíaco; se houver necessidade de estimar a potência do sinal (suspeita de patologia), a STFT é recomendada; se se pretender a frequência média e a largura de banda, a AR-MC poderá apresentar melhores valores (Leiria, 2004).

No entanto, se o vaso apresentar indícios de estenose as distribuições tempo-frequência serão provavelmente as mais indicadas. A classificação do tipo de embolias depende da estimação da potência, o que conduz à escolha da STFT, com o respectivo cálculo prévio dos seus parâmetros.

Se um sinal apresenta ruído então a utilização de métodos paramétricos pode ser preferível, uma vez que permite o aumento da relação sinal-ruído através da contribuição de dados do sinal fora da janela de observação. Também os métodos tempo-frequência que permitam suavizar o espectro do sinal (como a CWD), podem ter vantagens.

8.2.4 Sumário

As principais conclusões que podem ser retiradas destes estudos são que o melhor método de estimação depende das características do vaso a observar, do tipo de diagnóstico que se pretende, e da relação sinal-ruído observada. Como a qualidade da estimação de parâmetros do espectro, utilizados para o cálculo de indicadores clínicos, depende das características do sinal, há que escolher os estimadores espectrais a usar em cada caso.

Do que atrás foi exposto chega-se à conclusão de que não é possível determinar o melhor método para todos os sinais. É necessário verificar primeiro quais as

características do sinal e o que se quer observar. Mais, é possível que as características particulares de determinado indivíduo (por exemplo irregularidades no ritmo cardíaco, etc.) possam justificar a utilização de um método diferente, ou pelo menos de diferentes parâmetros do mesmo método, para a análise de uma mesma enfermidade nessa artéria, para a qual se apontou como preferível um dado método de estimação.

8.3 Estratégias no Desenvolvimento do Sistema

8.3.1 Desenho do Sistema

A abordagem seguida na concepção e desenho do sistema visou um modelo funcional (Calvez, 1993) independente da tecnologia a adoptar, que não contemplasse considerações específicas da implementação como dimensão e taxa de transferência de dados, tipo e velocidade do processador, capacidade de transferência do bus¹, sistema operativo, *drivers*. Procurou-se com tal abordagem que a solução adoptada para o desenho do sistema pudesse ser implementada em diferentes arquitecturas de hardware e diferentes sistemas operativos.

O desenho do sistema apresentado no capítulo 3 visa potenciar a qualidade do sistema. Do ponto de vista externo da qualidade, assumem especial importância os factores compatibilidade – facilidade de combinação deste sistema com outros programas – e extensibilidade – facilidade de adaptação a alterações de especificação. Outros factores de qualidade a destacar incluem a reutilização e a portabilidade.

A modularidade impressa no desenho do sistema, pela divisão em módulos coerentes e funcionalmente independentes (como se se tratasse de *integrados de software*), permite o controlo da complexidade. A decomposição de cada subsistema em várias componentes, permite considerar uma multiplicidade de objectivos. Variantes do sistema são compostas a partir de módulos pré-definidos, em que cada módulo executa uma função.

Os módulos, sequencialmente coesos, são ligados pelo fluxo de dados. Esta abordagem traduz o modelo de *pipeline*, conforme requerido pelo projecto MAGIAS.

¹do inglês, bus throughput

8.3.2 Implementações

A fase de implementação do sistema foi executada seguindo a ideologia subjacente ao desenho do sistema. Desta forma, cada módulo foi implementado separadamente, e, por vezes, dentro de um mesmo módulo foram testadas alternativas a componentes do módulo, as quais, apresentando características indesejáveis, ou que não provassem acrescentar mais-valia ao sistema, não foram consideradas nos testes de desempenho dos módulos relatados nesta tese. De forma sumária, expõem-se agora algumas das alternativas consideradas e as razões da sua não adopção à data.

8.3.2.1 Multiprocessamento Simétrico

O suporte a multiprocessamento simétrico nos núcleos Linux versão 2.0, em 1999, era ainda experimental. Embora vários processos pudessem ser executados simultaneamente em diferentes processadores, uma fila de espera regulava o acesso das diferentes tarefas ao modo de sistema e um trinco no núcleo garantia a integridade das estruturas de dados internas.

Por consequência, embora os processos executados em modo utilizador pouco afectados fossem, aqueles que requeriam passagem a modo de sistema – como processos com muitas entradas e saídas – pouco beneficiavam da possibilidade de multiprocessamento.

8.3.2.2 Memória Partilhada

É sabido que um processo que executa uma chamada de funções de sistema é sujeito a uma mudança de contexto – de modo utilizador a modo de sistema – que se pode traduzir num aumento do tempo de execução em comparação com uma chamada a uma função de biblioteca que execute a mesma chamada (Burns & Wellings, 1997). Por este motivo, o acesso a memória partilhada requerendo funções do sistema em aplicações em modo utilizador penaliza o desempenho da aplicação em causa.

Como alternativa poder-se-ia considerar que a aplicação, no seu todo ou em parte, deveria funcionar em modo de sistema. Esta alternativa não foi considerada politicamente correcta por o sistema se encontrar numa fase de desenvolvimento, naturalmente pouco estável.

A utilização de memória partilhada foi ensaiada para o armazenamento da

matriz de transformação do sinal analítico numa das implementações da distribuição Bessel. Verificou-se que a utilização de memória partilhada penalizava o desempenho das implementações dos estimadores espectrais, pelo que essa alternativa não foi adoptada.

8.3.2.3 Plataformas Aditivas ou de Propósito Geral

No início do trabalho pressupunha-se a utilização de plataformas paralelas como forma eficiente de implementação em tempo-real do sistema analisador de fluxo sanguíneo. A rápida ascensão das capacidades de cálculo dos computadores pessoais, associada ao conhecimento da existência de computadores pessoais nas salas de exames cardiovasculares do Hospital Distrital de Faro, induziu o uso de plataformas de propósito geral. Não sendo óbvia na altura o tipo de plataforma a escolher, preferiu-se acreditar na continuação da melhoria das condições de cálculo dos computadores pessoais, e, optou-se pelo desenvolvimento em plataformas de propósito geral.

Além disso, a utilização eficiente de plataformas aditivas exige a adequação do algoritmo à arquitectura, envolvendo um processo de afinação que pode ser moroso e raramente reutilizável, facto que obrigaria ao recurso a diversas plataformas aditivas para se poder generalizar o sistema para outras artérias e não apenas para a artéria carótida primitiva. A inclusão de várias plataformas aditivas, além de exigir o desenvolvimento de software e hardware de suporte, produziria um sistema complexo e financeiramente mais caro (contrariando um dos objectivos globais do trabalho).

No entanto, a alternativa de recurso a plataformas aditivas não deve ser liminarmente eliminada. Pontualmente, quando se pretende que parte da carga de processamento no processador central seja aliviada, o uso de plataformas aditivas é uma boa opção (como exemplo, veja-se (Nikolov, González & Jensen, 2003)). Ainda neste contexto, é de considerar a utilização do sistema AMPLIA com módulos equipados, por exemplo, com C40s a executar a implementação da ARMC ou da CWD (ver também secção 8.3.3, *AMPLIA*).

Quando não está definida a plataforma a utilizar, o desenvolvimento em plataformas de propósito geral é facilitado não só pelo baixo custo do equipamento como pela variedade de ambientes de desenvolvimento, compiladores e livrarias.

Graças à opção de utilização de processadores de propósito geral, o protótipo do sistema analisador de fluxo sanguíneo baseia-se no protótipo laboratorial do

transdutor ultra-sónico de Doppler contínuo e num computador portátil comum.

8.3.2.4 Múltiplas Aplicações ou uma Aplicação Múltipla

A implementação de múltiplas aplicações consoante a dimensão dos dados a tratar é parte da estratégia de desenvolvimento de aplicações rápidas, particularmente no caso de utilização de plataformas aditivas. Como vantagem pode referir-se a alocação estática das áreas de memória necessárias e a sua atribuição a diferentes blocos de memória de forma a potenciar o uso de múltiplos *buses* e *pipelining* de instruções.

A principal desvantagem deste tipo de abordagem consiste na necessidade de manter várias instâncias da mesma aplicação. Naturalmente, o uso de macros de substituição pode minimizar este inconveniente em ambiente laboratorial se o tempo de compilação necessário ao teste de várias configurações da aplicação for pequeno. No entanto essa opção não é aceitável para um ambiente de exploração, em que o utilizador final, após a escolha de novos parâmetros, tenha que aguardar o fim do processo de compilação. Outro inconveniente traduz-se na necessidade de incluir compilador e livrarias de desenvolvimento com cada instalação do sistema.

Outra das alternativas considerada transfere a gestão das múltiplas aplicações para a interface com o utilizador. Apesar da variedade de casos cobertos por uma implementação de estimador, a modularização adoptada na codificação permite, com pouco esforço, que sejam criadas diversas aplicações, uma para cada conjunto de especificações dos métodos. Até determinada fase do desenvolvimento, essa alternativa foi sendo considerada, para a extensão do *python* com C, criando módulos específicos para cada caso e assim obviar os inconvenientes referidos. Verificando-se que a gestão dos diferentes casos (com elevado custo de manutenção) penalizava o desempenho da interface, essa opção foi descontinuada.

Face ao exposto, e à variedade dos casos a considerar, optou-se pela decisão de manter uma aplicação única para cada método, cujos parâmetros de utilização seriam indicados como argumentos do programa em tempo de execução.

Naturalmente, esta opção tem custos de desempenho. A alocação de memória tem que ser feita por excesso, contemplando o pior caso, ou de forma dinâmica. Tendo sido ensaiadas as duas alternativas, determinou-se que a alocação dinâmica de memória seria preferível.

A parametrização das aplicações por parte do utilizador é feita de forma

gráfica. É definida uma configuração por defeito, que traduz o melhor desempenho estatístico em casos normais (sem patologia). As escolhas feitas pelo utilizador são guardadas em ficheiros de configuração pessoal num directório sob o seu directório raiz, estando prevista nas janelas da interface gráfica a opção de reposição das configurações por defeito. A gestão da correspondência entre as opções oferecidas em modo gráfico e as utilizadas pelas aplicações é mantida por um módulo de ligação apresentado no Apêndice B, simplificando a mudança entre versões.

8.3.2.5 Impacto do Tipo de Dados no Subsistema de Processamento

O tempo de comunicação entre as plataformas aditivas e a máquina hospedeira é responsável pela degradação do desempenho global da solução. Esse problema só pode ser tratado no caso específico da plataforma de *hardware* adoptada.

A verificação da correcção numérica dos métodos implementados foi feita por comparação com os resultados obtidos com os programas em Matlab considerados como especificação. Esse processo foi, por vezes, moroso, obrigando a um processo detalhado de comparação de resultados. Verificavam-se diferenças numéricas, por vezes bastante significativas, resultantes da diferença de precisão numérica, da acumulação de arredondamentos ou mesmo das funções implementadas. Não estando reportada na literatura a comparação da qualidade numérica das implementações em plataformas aditivas ou genéricas, foi analisado o benefício da estimação (secção 7.2.1.1), sendo identificados os motivos e possíveis soluções.

Na primeira versão das implementações dos métodos, os ficheiros de dados gerados pelo simulador tinham o formato de texto. Não sendo possível impor a mudança dos dados, foram criados programas de conversão dos dados para binário em vírgula flutuante (cf. secção 4.2.1). A necessidade de contemplar as diversas origens de dados (cf. 4.3) foi sendo colmatada com a criação de módulos de conversão para cada caso específico, mantendo num cabeçalho a informação contida nos ficheiros originais. Em termos do desempenho global dos estimadores, esta opção não acarreta custos pois na maioria dos casos a conversão só tem que ser feita uma vez.

Assume especial relevância em termos da implementação a aquisição de dados do transdutor ultra-sónico.

A aquisição de dados em fase e em quadratura de fase, que traduzem as

componentes de fluxo directo e reverso, requer que os métodos contemplem sinais complexos. Esse facto tem impacto na formulação matemática dos algoritmos, no algoritmo computacional que implementa esses algoritmos e, conseqüentemente na complexidade algorítmica. No caso dos estimadores CWD e Bessel, caso os dados sejam do domínio complexo, a carga de cálculo do sinal analítico é retirada; no caso do AR-MC, as simplificações do algoritmo computacional já não são válidas, nomeadamente no cálculo da matriz de covariância modificada e no cálculo da densidade espectral de potência.

No que diz respeito à implementação das distribuições tempo-frequência, as alternativas são claras: separar o cálculo do sinal analítico do da distribuição propriamente dita ou condicionar a execução desse cálculo. A separação das aplicações tem a vantagem de se adequar ao processamento em *pipeline*; o tempo de execução das duas alternativas foi comparado e, tendo-se concluído que a soma do tempo de execução das duas aplicações é inferior ao da aplicação que implementa o conjunto sinal analítico e distribuição tempo-frequência, optou-se pelo cálculo do sinal analítico em separado das distribuições tempo-frequência.

Os módulos da AR-MC afectados pela alteração do tipo de dados são optimizados, sendo conseguido um novo algoritmo simplificado do cálculo da matriz de covariância e, numa nova abordagem, o cálculo da densidade espectral de potência a partir das estimativas dos parâmetros do modelo é feito recorrendo à *FFTW*.

8.3.3 Obstáculos e Consequentes Decisões

8.3.3.1 Obtenção de Dados RF

Foi identificado um produto comercial que permitia a recolha de dados RF de um VingMed System FiVe existente no HDF. Com a aquisição da VingMed pela General Electric em 1998 e conseqüente reestruturação de produtos e serviços, não foi possível proceder à aquisição deste produto.

A impossibilidade de definir o subsistema de aquisição de dados teve um impacto significativo no desenho do sistema. A estratégia adoptada foi a possibilidade de considerar dados simulados e vários tipos de dados reais para tratamento diferido (cf. secção 4.3).

Quando, em finais de 2002, se obteve a sonda (Fuentes, Sotomayor, García, Acevedo & Moreno, 2002; García, Moreno, Solano, Berragán, Sotomayor & Fuentes, 2002) – um transdutor ultra-sónico de Doppler contínuo – foi possível adaptar

e ensaiar o protótipo do sistema de análise clínica de sinais de fluxo sanguíneo.

8.3.3.2 AMPLIA

No âmbito do projecto MAGIAS, o grupo de investigação do *Instituto de Automatica Industrial do Consejo Superior de Investigaciones Científicas*, Espanha, propôs a utilização do sistema AMPLIA como arquitectura de suporte. O sistema AMPLIA é uma arquitectura dotada de um *bus* dedicado e de uma unidade de controlo e interface entre um computador hospedeiro e os módulos de aquisição de dados ou processamento. Estes módulos poderiam ser processadores digitais de sinal ou *hardware* configurável.

Actualmente na sua versão v2.3.2, este sistema teve uma evolução bastante rápida: em Março de 2000, v.1.0; Janeiro de 2001, v.1.1 para módulos de oito bits; Novembro de 2001, v.2.0 com melhor desempenho e controlo simplificado; Fevereiro de 2002, v.2.1 com alterações nas ligações físicas e valores da fonte de alimentação; Outubro de 2002, v.2.2 com alterações na sequência de execução de comandos; Abril de 2003, v.2.3 em teste de aplicações; em Outubro de 2003 é divulgada a versão final da documentação.

Estando a ser desenvolvido em paralelo com o presente sistema, considerando o ritmo de alterações e os inconvenientes do uso de plataformas aditivas já referido na secção anterior desta tese, optou-se por não considerar essa arquitectura específica mas apenas alguns dos elementos de processamento que a mesma poderia integrar.

8.3.4 Outras Opções

8.3.4.1 Transferência de Dados entre Subsistemas

No caso do processamento diferido de dados, os programas de conversão permitem a conversão de formatos “estrangeiros” para os usados pelo sistema de processamento de dados.

No caso dos ficheiros de tipo “WAV” foi também criado um programa de conversão. No caso da aquisição de dados a partir do transdutor, não seria aceitável esperar que a colheita de dados fosse feita, para então proceder à sua estimação espectral e conseqüente visualização de resultados. Alguns dos mecanismos de comunicação entre processos foram considerados. Nesta fase inicial, foi adoptado um mecanismo semelhante ao *tee*, interceptando a escrita de dados a partir da

carta de som e, a par da escrita num ficheiro, escrevendo-o num *FIFO* (*named pipe*). Esta opção é mais simples e rápida que a utilização de *sockets* e, estando definida no standard POSIX oferece garantias de portabilidade. Uma vez que as implementações dos métodos no subsistema de processamento foram reformuladas, para garantir a continuidade de tratamento em diferido de dados, foi criada uma módulo que lê os dados de ficheiros e os escreve no *FIFO*.

Em termos da investigação em processamento de sinal, a possibilidade, até então inexistente, de tratar repetidamente dados clínicos com diferentes configurações dos métodos, e, se necessário, em simultâneo é extremamente vantajosa. Do ponto de vista do desempenho do sistema, a utilização do *FIFO* pode conduzir a uma degradação dos tempos globais da aplicação embora não comprometa os requisitos de tempo-real.

A ligação entre o subsistema de processamento e o subsistema de visualização de dados não é tão robusta e é um dos casos em que a modularização impõe o seu custo. Devido às diferentes possibilidades de configuração dos métodos de estimação espectral não é possível prever a configuração da janela de visualização: a resolução temporal e na frequência (eixos horizontal e vertical, respectivamente) dependem do estimador aplicado, do tamanho da janela aplicada à segmentação do ciclo cardíaco, da sobreposição dessas janelas e da utilização e tipo de sinal analítico. A implementação relatada nesta tese apoia-se num ficheiro que configura a janela de visualização, permitindo a visualização de espectros instantâneos. Este ficheiro, ou a parametrização que ele expressa, surge como resultado do cálculo do espectro médio. É de notar que, sendo frequentemente referida a utilização de espectros médios de forma a reduzir a variância intrínseca resultante da utilização destes sinais, a literatura não é unânime sobre o número de espectros que contribuem para a média. É também sobre este espectro médio que são calculados alguns dos indicadores clínicos, sendo a sua utilização desaconselhada em outros indicadores (como por exemplo, no caso de detecção de embolias).

A implementação do cálculo do espectro médio foi feita para sinais simulados, todos com a mesma duração do ciclo cardíaco. Considerando que na maioria dos casos a duração da sístole é constante e que as variações da duração do ciclo cardíaco ocorrem na diástole (cl clinicamente menos relevante), é necessário identificar o início do ciclo cardíaco; a metodologia a seguir foi recentemente divulgada (Leiria, 2004), devendo agora proceder-se à definição do algoritmo computacional e conseqüente implementação.

A utilização do subsistema de avaliação para a investigação em processamento de sinal requer a adequação caso a caso. Centrado no critério Custo/Benefício, este subsistema regista alterações muito frequentes, quer na criação de novos simuladores (síntese de sinais) quer na definição dos parâmetros estatísticos e espectrais em análise. As alternativas de utilização contemplam a chamada, em segundo plano, do simulador e/ou da função de benefício (como exemplo, veja-se (Moura, Leiria & Ruano, 2003)), ou a chamada dos módulos que realizam o subsistema de processamento.

8.3.4.2 Sinal Analítico

Pelos estudos feitos até à data, a utilização do sinal analítico em conjunto com os diferentes estimadores revela-se benéfica em termos da qualidade da estimação espectral. Caso se venha a constatar que a carga computacional excede a capacidade do sistema a utilizar, é possível transferir esse cálculo da parte computacional para a analógica. Não se prevê que tal opção acarrete diminuição da taxa de produção de resultados, pois as frequências de amostragem dos sinais Doppler de fluxo sanguíneo são normalmente da ordem das dezenas de kHz, muito abaixo dos valores actualmente conseguidos no âmbito das comunicações digitais (Sieben, Conradi & Dodss, 1999).

8.3.5 Sumário

A generalidade e versatilidade do sistema analisador de fluxo sanguíneo deve-se à frequente reavaliação de objectivos do presente trabalho para abranger outros métodos, vasos e patologias.

A evolução do desenvolvimento do sistema e a sua integração privilegiou a compatibilidade e extensibilidade, tentando combater a obsolescência tecnológica e promover a reutilização. Justifica-se assim a opção pelas plataformas de propósito geral e a criação de uma aplicação para cada método cuja sintonia é determinada em tempo de execução.

As diferentes implementações do subsistema de processamento, que traduzem diferentes algoritmos computacionais, foram sendo avaliadas em termos de tempo de execução, incorporando formas alternativas de cálculo de forma a maximizar os casos considerados sem comprometer o desempenho.

A ligação do subsistema de processamento aos demais subsistemas não é muito

robusta devido às frequentes actualizações do primeiro.

8.4 Considerações finais

A finalidade deste trabalho foi a criação de um protótipo de um sistema tempo-real de análise de sinais de ultra-som de fluxo sanguíneo.

Este fim tem motivações de carácter social, económico e científico. As doenças cardiovasculares constituem a maior causa de mortalidade e uma das maiores causas de invalidez, em Portugal, na Europa e nos Estados Unidos da América. O rastreio de anomalias cardiovasculares é limitado pelo custo dos equipamentos clínicos de ecografia, sendo diminuto o número de exames ecográficos de diagnóstico precoce. Em termos científicos, foi sentida a necessidade de comprovar, inequivocamente e de forma pragmática, a contribuição de vários métodos e técnicas desenvolvidos, não só pela equipa a que a autora pertence, como também por outros grupos colaborantes.

Estes grupos de investigação, pertencentes a uma rede ibero-americana, unidos pelo comum interesse na tecnologia de ultra-som, reuniram-se em torno de um projecto comum, MAGIAS, apoiado pela UNESCO no reconhecimento da relevância da tecnologia de ultra-som e da necessidade de independência tecnológica; no seu conjunto, esses grupos de investigação reúnem as competências necessárias para a criação de equipamentos com aplicação industrial ou médica.

A especialização cria um afunilar de competências, verificando-se, muitas vezes, que é na fronteira das diferentes áreas que surgem os desafios à maturidade de uma tecnologia.

É neste contexto que se situa o trabalho desenvolvido, contexto que justifica uma contribuição intangível deste trabalho: a concepção do sistema global. Os subsistemas são definidos em termos funcionais, em que cada realização deve ser encarada como um componente reutilizável e substituível. É esta última característica que confere versatilidade ao sistema e garante a universalidade de aplicação. Cada módulo, como se de uma peça mecânica se tratasse, deve poder ser substituído por outro, tecnologicamente mais evoluído, com um impacto mínimo na totalidade do sistema. Esta independência do sistema face aos seus componentes permitirá também potenciar a sinergias da colaboração entre os diversos grupos.

Neste caso específico, o sistema é realizado para a análise de sinais de ultrassom de fluxo sanguíneo em vários vasos, podendo cumprir objectivos clínicos ou de investigação.

Após a concepção do sistema, a atenção foi dirigida para o cerne do protótipo, visando a implementação em tempo-real de métodos de estimação espectral.

Havia o conhecimento prévio dos trabalhos de identificação e implementação paralela em redes de *transputers* dos estimadores AR-MC (Ruano, 1992) de M. Graça Ruano, e TFD-CW (Cardoso, 1998) de J. C. Cardoso e o estudo da implementação paralela em arquitecturas homogéneas, incorporando *transputers*, C40s e um i860, tinha já sido efectuado pela autora (Madeira, 1997).

Assim, foi estudada a utilização de arquitecturas heterogéneas em plataformas aditivas, o que ainda não tinha sido feito com estes elementos de processamento no contexto da presente aplicação. A participação em dois estudos, também de carácter inovador, com dois grupos colaborantes permitiu avaliar a utilização do processador SHARC e a utilização parcial de hardware reconfigurável na implementação do estimador AR-MC.

A avaliação qualitativa das plataformas heterogéneas utilizadas não foi satisfatória, apesar de o tempo de execução das implementações efectuadas ser quantitativamente inferior ao limite de tempo-real. O tempo medido da execução das implementações nas C40 da plataforma aditiva A variava significativamente, facto que punha em causa a predictibilidade do desempenho do sistema. O processador i860, embora apresentasse bons resultados na implementação do método AR-MC, não suportava o algoritmo computacional usado para o método CWD. Há um aspecto muito importante a considerar na utilização de plataformas aditivas em implementações tempo-real: a transferência de dados entre a máquina hospedeira e a plataforma de processamento. Este aspecto inviabiliza a utilização da plataforma aditiva B, não tendo sido avaliado nas plataformas com o SHARC ou hardware reconfigurável.

Cumprindo o objectivo de apoiar a investigação em processamento de sinal, as implementações já disponíveis na arquitectura B são utilizadas num estudo de estimação de parâmetros espectrais em válvula aórtica. É de referir que, sendo necessário o teste de um conjunto de alternativas, é desejável que os resultados sejam obtidos em intervalos de tempo muito curtos. Neste contexto de teste de métodos, a obtenção de resultados em tempo-real não é suficiente. Usando os diferentes recursos computacionais disponíveis, há oportunidade de comparar os

pares algoritmo/arquitetura. Ainda que compensando a diferença de velocidade dos processadores utilizados, verifica-se a vantagem de, em vez da plataforma aditiva, usar processadores de uso genérico no ambiente de desenvolvimento habitual, ou seja, sistema operativo Debian/GNU Linux, utilizando o compilador da linguagem C, *gcc*.

Esta conclusão vem a revelar-se fulcral, optando-se então, por desenvolver sobre/para o sistema operativo Linux, usufruindo da existência de código livre, aberto e portátil. Naturalmente, esta opção foi também reforçada pelo custo económico da utilização de plataformas aditivas, não só em termos de equipamento para desenvolvimento dos trabalhos como também de aquele necessário a uma solução final.

A participação nesse estudo pôs também à prova a versatilidade pretendida, por ser necessário ajustar os parâmetros dos estimadores.

Considerando a rápida evolução tecnológica que se regista actualmente, é necessário, no desenvolvimento de sistemas computacionais, ter em conta a obsolescência programada do equipamento, mais do que a optimização das implementações particulares, pelo que se procurou o paradigma de programação mais adequado.

Procedeu-se então a uma nova especificação dos estimadores, considerando todos os casos que se previam possíveis. Foi executada a consequente implementação, sendo os estimadores parametrizados pelos argumentos dos programas. A solução adoptada introduz uma mais-valia significativa na funcionalidade do sistema, sendo a sua utilização mais amigável. A utilização de argumentos nos programas que implementam os estimadores não é, tanto quanto se pode avaliar, uma solução trivial, sendo habitual a criação de procedimentos que criem/alterem o código fonte e desencadeiem a sua compilação ou a criação de bibliotecas que reúnam as diferentes alternativas previstas.

Esta versatilidade e facilidade de utilização são conseguidas aceitando os custos inerentes à gestão dos vários casos e à alocação dinâmica de memória. Como esses custos não são visíveis na totalidade (caso se analisasse o tempo de cálculo de segmentos de dados), para permitir a avaliação inequívoca do cumprimento do requisito tempo-real e do desempenho de cada implementação na sua totalidade, passou a ser usado um sinal de 33,61 s e a medição de tempos passou a ser absoluta, usando o comando *time* do sistema operativo.

Nesta fase dos trabalhos o estimador STFT foi também incluído, por constituir uma referência no processamento de sinais, e ser necessária para alguns estudos nessa área, e por assentar na transformada discreta de Fourier usada nos demais estimadores. Foram implementadas e comparadas três alternativas do estimador STFT. Qualquer uma delas apresentou resultados em tempo-real, podendo ser utilizada no subsistema de estimação espectral. É de realçar, neste caso, a vantagem da utilização de software livre, fiável e em evolução sem custos de desenvolvimento. A existência de alternativas garante a independência do sistema, podendo ser adoptada qualquer das alternativas sem que o sistema, na sua totalidade, seja comprometido.

Tanto quanto é possível avaliar, outra contribuição do trabalho desenvolvido consiste no estudo da implementação do sinal *analítico* e na sua integração nos métodos de estimação espectral considerados. Verificou-se que a abordagem habitual, de aplicar o método a segmentos de sinal em vez de à totalidade dos sinais, já não se justifica. Essa abordagem, um legado de estudos que ainda hoje são uma referência na área, não tinha sido re-equacionada após as recentes evoluções tecnológicas. Também neste caso, foram implementadas e comparadas três abordagens, qualquer delas executada em tempo-real. Para o subsistema de estimação espectral foi adoptada a abordagem de integração dos algoritmos que mais beneficiasse a versatilidade do sistema.

Como o sinal *analítico* é um sinal complexo, algumas das simplificações no cálculo do estimador AR-MC para o caso de sinais reais, não podiam ser aplicadas. A generalização da utilização do sinal *analítico* implicou assim a revisão do estudo da implementação do método AR-MC resultando num novo algoritmo simplificado para o cálculo da matriz de covariância modificada (componente do algoritmo que maior volume computacional requer).

Como resultado da especificação alargada dos estimadores, as principais inovações introduzidas na implementação da CWD consistem na possibilidade de utilização de um de vários núcleos e na possibilidade de incluir, ou não, o cálculo de sinal *analítico* de cada segmento. Embora, do ponto de vista da qualidade da estimação, a vantagem em utilizar um dos novos núcleos não seja absoluta, a possibilidade de proceder a esse estudo é disponibilizada. No que diz respeito ao sinal *analítico*, sendo a sua utilização opcional, é possível manter a abordagem tradicional ou, como apontado, transferir esse cálculo para uma fase anterior à estimação espectral.

Sendo desejável que os potenciais “bons” estimadores estejam disponíveis para estudo em termos da qualidade estatística de estimação e para eventual utilização clínica e considerando também uma especificação alargada da distribuição tempo-frequência de Bessel, procedeu-se à sua implementação tendo sido comparado o tempo de execução total medido em três máquinas. Na maioria dos casos, o tempo de estimação é inferior ao limite de tempo-real, não se mantendo o constrangimento, referido por diversos autores, quanto à sua utilização.

Até uma fase adiantada do desenvolvimento do sistema, não sendo possível proceder à aquisição de dados clínicos, consideraram-se várias alternativas para o subsistema de aquisição de dados. Essas alternativas contemplam a utilização de dados criados por programas de simulação, conversores para adaptação do formato de dados de proveniência diversa. Tendo um dos grupos colaborantes do projecto MAGIAS fornecido uma sonda em que os dados são obtidos através de uma placa de som no computador pessoal, o módulo para a sua utilização foi então construído pela modificação de software livre. Graças à modularidade e independência, de raiz, dos subsistemas, esse módulo foi integrado no sistema existente sem comprometer o desenvolvimento feito até à data.

Atendendo a que o subsistema de visualização pode acarretar uma carga computacional considerável, foram estudadas várias alternativas tendo-se optado por uma biblioteca livre, com funcionalidades mínimas, mas que permitisse a visualização rápida de espectros. Ainda, associado à área de interface com o utilizador, foi desenvolvido um sistema gráfico que permite a realização do sistema reunindo os módulos de cada subsistema que se escolha utilizar. É de referir que neste sistema gráfico a versatilidade é parcialmente mantida por ficheiros de configuração onde são especificados os módulos a incluir em cada caso. Para substituir uma versão da implementação de um estimador por outra, basta editar uma linha desse ficheiro de configuração.

8.5 Trabalho Futuro

Com este protótipo consegue-se uma base de trabalho permitindo analisar espectros de fluxo sanguíneo em alguns vasos. Tendo em conta os objectivos do sistema analisador de fluxo sanguíneo, as principais prioridades são:

- em termos de sistema de caracterização de fluxo sanguíneo, este sistema beneficiaria do cálculo do espectro médio de um número de ciclos. Computacionalmente, o cálculo desse espectro não é crítico sendo necessário uma técnica robusta, versátil e que não requeira a intervenção do utilizador;
- em termos do sistema de apoio à investigação em processamento de sinal, o sistema beneficiaria da inclusão de um procedimento automático de teste de parâmetros de estimadores ou da determinação automática do ponto de funcionamento óptimo;
- do ponto de vista do sistema desenvolvido, a articulação dos módulos de cada subsistema deveria ser feita a mais alto nível.

De forma mais pragmática, considera-se que o sistema deve ser disponibilizado, de forma a cumprir a sua vocação agregadora. Para tanto, é necessário definir, de forma mais concreta, a fronteira entre subsistemas, documentar, de forma sintética, cada módulo e criar condições para o desenvolvimento concorrente.

Bibliografia

3L Limited (1995). *C4x Parallel C V2.0.2*, 3L Ltd., Scotland.

Alex Parallel Computers, Inc. (1996). Sharc1000 user's manual.

Barry, D. T. (1992). Fast calculation of the choi-williams time-frequency distribution, *IEEE Transactions on Signal Processing* **40**(2): 450–455.

Bellis, S. J., Fish, P. J. & Marnane, W. P. (1997). Optimal systolic arrays for real-time implementation of the Modified Covariance spectral estimator, *Parallel Algorithms and Applications* **11**(1-2): 71–96.

Bellis, S. J., Marnane, W. P. & Fish, P. J. (1997). Alternative systolic array for non-square-root Cholesky decomposition, *IEE Proceedings: Computers and Digital Techniques* **144**(2): 57–64.

Bellis, S. J., Marnane, W. P., Wilde, D. & Fish, P. J. (1994). Systolic arrays for Modified Covariance spectral estimation used with ultrasonic Doppler blood flow detectors, *Signal Processing VII - Theories and Applications, Proceedings of EUSIPCO-94* **3**: 1361–1364.

Boashash, B. & Black, P. J. (1987). An efficient real-time implementation of the wigner-ville distribution, *IEEE Trans. on Acoustics, Speech, Signal Processing* **35**: 1611–1618.

Bracewell, R. N. (2000). *The Fourier Transform and its applications*, third edn, McGraw-Hill Book Co - Singapore, 685 Canton Street, Norwood, MA 02062, USA.

Brigham, E. O. (1988). *The fast Fourier transform and its applications*, Prentice Hall, Inc., Englewood Cliffs, New Jersey.

- Burns, A. & Wellings, A. (1997). *Real-Time Systems and Programming Languages*, second edn, Addison-Wesley, Addison Wesley Longman Limited, England.
- Buzo, M., Zoppi, G. & Fabris, R. (2002). Introduzione a python e gui - parte prima. <http://www.python.it/gui/gui-1.html>.
- Calvez, J. P. (1993). *Embedded Real-Time Systems, A Specification and Design Methodology*, Wiley Series in Software Engineering Practice, John Wiley & Sons, England, UK.
- Cardoso, J. (1998). *Investigation and implementation of real-time spectral estimation techniques for use with pulsed Doppler blood flow detectors*, PhD thesis, University College of North Wales, Bangor, U. K.
- Cardoso, J. C. S., Ruano, M. G. & Fish, P. J. (1996a). Nonstationarity broadening reduction in pulsed doppler spectrum measurements using time-frequency estimators, *IEEE Trans. on Biomedical Eng.* **43**(12): 1176–1186.
- Cardoso, J., Fish, P. & Ruano, M. (1996). Choi-williams time-frequency estimator, a parallel implementation, *VECPAR'96, 2nd International Meeting on Vector and Parallel Processing* .
- Cardoso, J., Ruano, M. G. & Fish, P. (1996b). Distribuições tempo-frequência aplicadas à estimação da largura de banda e frequência central instantânea do sinal doppler, *4º Congresso Português de Engenharia Biomédica, BIO-ENG'96 II.1*.
- Challis, R. E. & Kitney, R. I. (1990). Biomedical signal processing (in four parts) part i: Time-domain methods, *Medical and Biological Engineering and Computing* pp. 509–524.
- Chambers, C. (1995). A study of performance issues in heterogeneous and homogeneous architectures, The University of Sheffield, U.K. MSc. Thesis.
- Choi, H. I. & Williams, W. J. (1989). Improved time-frequency representation of multicomponent signals using exponential kernels, *IEEE Trans. Acoust., Speech, Signal Processing* **37**: 862–871.

- Cloutier, G., Chen, D. & Durand, L.-G. (2001). Performance of time-frequency representation techniques to measure blood flow turbulence with pulsed-wave doppler ultrasound, *Ultrasound in Medicine & Biology* **27**(4): 535–550.
- Clunie, D. A. (2002). David clunie's medical image format site. <http://dclunie.com/>.
- Cohen, L. (1989). Time-frequency distributions - a review, *Proc. IEEE* **77**(7): 941–981.
- Cohen, L. (1995). *Time-Frequency Analysis*, Prentice Hall Signal Processing Series, Prentice-Hall PTR, New Jersey, USA.
- Costa, A. P. (1997). Desenvolvimento de um simulador de sinais de fluxo sanguíneo na válvula aórtica, Universidade do Algarve, Faro, Portugal.
- DICOM (2000). *Digital Imaging and Communication in Medicine*. <http://global.ihs.com>.
- Evans, D. H. (1988). A pulse-foot-seeking algorithm for doppler ultrasound waveforms, *Clin. Phys. Physiol. Meas.* **9**(3): 267–271.
- Evans, D. H. & McDicken, W. N. (2000). *Doppler ultrasound physics, instrumentation, and clinical application*, second edn, John Wiley & Sons, UK.
- Evans, D. H., McDicken, W. N., Skidmore, R. & Woodcock, J. P. (1989). *Doppler ultrasound physics, instrumentation, and clinical application*, John Wiley & Sons, New York.
- Fan, L. & Evans, D. H. (1994). Extracting instantaneous mean frequency information from doppler signals using the wigner distribution function, *Ultrasound in Medicine & Biology* **20**: 429–443.
- Frigo, M. & Johnson, S. G. (1998). <http://theory.lcs.mit.edu/~fftw/> ou <http://www.fftw.org>.
- Fuentes, M., Sotomayor, A., García, F., Acevedo, P. & Moreno, E. (2002). Punta detectora de flujo sanguíneo para un sistema ultrasónico doppler bidireccional, *Proc. 4th International Conference on Control, Virtual Instrumentation and Digital Systems*. CD-ROM.

- Galletly, J. (1990). *Occam 2*, Pitman Publishing, 128 Long Acre, London WC2E 9AN, UK.
- García, F., Moreno, E., Solano, J., Berragán, M. I., Sotomayor, A. & Fuentes, M. (2002). Sistema doppler de flujo sanguíneo con procesamiento digital para la separación de flujos, *Memorias del XVII Congreso de Instrumentación, SOMI-02*. 17FG289, CD-ROM.
- García-Nocetti, D. F., Flores, J. M. & González, J. S. (1995). Improving signal processing performance using a transputer-dsp parallel architecture, *Preprints of 2th IFAC Workshop on Algorithms and Architectures for Real-Time Control* pp. 569–573.
- García-Nocetti, D. F., González, J. S. & Flores, J. M. (1994). Heterogeneous architecture for parallel real-time spectral estimation in doppler blood flow instrumentation, *International Conference on Control* **1,2**: 37–41.
- GE (2000a). <http://www.ge.com/medical/ultrasound/msusys5.htm> ou http://www.gehealthcare.com/rad/us/products/sys_5/msusys5.html.
- GE (2000b). <ftp://ftp.med.ge.com/pub/DICOM/>.
- González, J. S. & García-Nocetti, D. F. (1994). Implementation of a parametric spectral estimator using genetic algorithms, *International Conference on Control* **1,2**: 754–759.
- González, J. S., García-Nocetti, D. F. & Vásquez, K. R. (1995). Parallel genetic algorithms in spectral estimation of doppler signals, *Preprints of 2th IFAC Workshop on Algorithms and Architectures for Real-Time Control* pp. 35–40.
- Guidi, F., Guidi, G., Ricci, S., Atzeni, C. & Tortoli, P. (1998). High-speed parallel processing of biomedical ultrasound signals, *The 2nd European DSP Education and Research Conference* pp. 327–330.
- Guo, Z., Durand, L.-G. & Lee, H. C. (1994a). Comparison of time-frequency distribution techniques for analysis of simulated doppler ultrasound signals of the femoral artery, *IEEE Trans. on Biomedical Eng.* **41**(4): 332–342.
- Guo, Z., Durand, L.-G. & Lee, H. C. (1994b). The time-frequency distributions of nonstationary signals based on a bessel kernel, *IEEE Trans. on Signal Processing* **42**(7): 1700–1707.

- Hahn, S. L. (1996). *Hilbert Transforms in signal processing*, Artech House, Inc., 685 Canton Street, Norwood, MA 02062, USA.
- INMOS Limited (n.d.). *Transputer Architecture Overview*, INMOS Limited. manual of Transputer Education Kit.
- Intel Corporation (2001). <http://www.intel.com/PentiumII/home.htm>.
- Jeong, J. & Williams, W. J. (1990). On the cross-terms in spectrograms, *Proc. of IEEE Int. Symp. on Circuits and Systems* **2**: 1565–1568.
- Jeong, J. & Williams, W. J. (1992a). Alias-free discrete-time time-frequency distributions, *IEEE Trans. on Signal Processing* **40**(11): 2757–2765.
- Jeong, J. & Williams, W. J. (1992b). Kernel design for reduced interference distributions, *IEEE Trans. on Signal Processing* **40**(2): 402–412.
- Jeong, J. & Williams, W. J. (1992c). Mechanism of the cross-terms in spectrograms, *IEEE Trans. on Signal Processing* **40**(10): 2608–2613.
- Kay, S. M. (1988). *Modern Spectral Estimation - Theory & Application*, Prentice Hall.
- Kitney, R. I. & Giddens, D. P. (1983). Analysis of blood velocity waveforms by phase shift averaging and autoregressive spectral estimation, *Journal of Biomechanical Engineering* **105**: 398–404.
- Kohn, L. & Margulis, N. (1989). Introducing the intel i860 64-bit microprocessor, *IEEE Micro* **9**(4): 15–30.
- Leiria, A. (2000). Estimação espectral do sinal doppler ultrassom em válvula aórtica, Universidade do Algarve, Faro, Portugal. Provas de Capacidade Científica.
- Leiria, A. (2004). *Spectral Analysis of Embolic Signals*, PhD thesis, Universidade do Algarve, Portugal. submetida em Fevereiro de 2004.
- Leiria, A., Madeira, M. M. & Ruano, M. G. (1999). Aortic valve analyser: a costbenefit study, *Proceedings of ICSPAT99*. 325.pdf, CDROM.

- Leiria, A., Madeira, M. M. & Ruano, M. G. (2000). Kernel modified choi-williams distributions for blood flow evaluation. Apresentado na Workshop in Signal Processing for Medical Ultrasound, UTAD, Vila Real, Portugal.
- Leiria, A., Moura, M. M. M. & Ruano, M. G. (2001). Comparison of discrete choi-williams based distributions on aortic valve blood flow centre frequency and bandwidth estimation, *Sixth Portuguese Conference on Biomedical Engineering*. CDROM, ID48.pdf.
- Ligertwood, C. (2002). <http://packages.debian.org/stable/sound/wavtools.html>.
- Madeira, M. M. (1997). Estudo da implementação do estimador ar - modified covariance em arquiteturas de processamento paralelo homogéneas, Universidade do Algarve, Faro, Portugal. Provas de Capacidade Científica.
- Madeira, M. M., Bellis, S. J., Beltran, L. A. A., González, J. S., Noceti, D. F. G., Marnane, W. P., Tokhi, M. O. & Ruano, M. G. (1999). High performance computing for real-time spectral estimation, *Control Engineering Practice* pp. 679–686.
- Madeira, M. M., Bellis, S., Marnane, W. & Ruano, M. G. (1998). Configurable processing for real time spectral estimation, *Preprints of 5th IFAC Workshop on Algorithms and Architectures for Real-Time Control* pp. 209–214.
- Madeira, M. M., Beltran, L. A. A., González, J. S., Noceti, D. F. G., Tokhi, M. O. & Ruano, M. G. (1998). High performance real time implementation of a spectral estimator, *Preprints of 5th IFAC Workshop on Algorithms and Architectures for Real-Time Control* pp. 215–219.
- Madeira, M. M., Tokhi, M. O. & Ruano, M. (2000). Real-time implementation of a doppler signal spectral estimator using sequential and parallel processing techniques, *Microprocessors & Microsystems* **24**(3): 153–167.
- Madeira, M. M., Tokhi, M. O. & Ruano, M. G. (1997). Comparative study of different doppler spectral estimator implementations, *Preprints of 4th IFAC Workshop on Algorithms and Architectures for Real-Time Control* pp. 293–298.

- Madeira, M. M., Tokhi, M. O. & Ruano, M. G. (1999). A time-frequency spectral implementation for a real-time biomedical application, *IFAC World Congress 1999* pp. 429–433.
- Marple Jr., S. L. (1987). *Digital Spectral Analysis with Applications*, Prentice Hall.
- Marple Jr., S. L. (1999). Computing the discrete-time "analytic" signal via fft, *IEEE Trans. on Signal Processing* **47**(9): 2600–2603.
- Mendanha, V. (1996). *Diálogos Filosóficos e Alquímicos: Conversas com Portugueses Notáveis*, Editora Pergaminho, Rua Rodrigues Sampaio, 79, R/C Esq°, 1150 Lisboa, Portugal.
- Middleton, T. & Kulakov, A. (1999). Vaults of parnassus. <http://py.vaults.ca/parnassus/>.
- Milanovic, L. & Wagner, H. (2002). <http://packages.debian.org/stable/libs/libg20.html>.
- Moraes, R., Aydin, N. & Evans, D. H. (1995). The performance of three maximum frequency envelope detection algorithms for doppler signals, *J. Vascular Investigation* **1**: 126–134.
- Moura, L. & Monteiro, P. (2002). Design method for fir-based hilbert transform filters suitable for broadband am-ssb, *Electronics Letters* **38**(12): 605–606.
- Moura, M. M. M., Leiria, A. & Ruano, M. G. (2002). Fast calculation of the bessel time-frequency distribution, *Proceedings of b'02 – 15th Triennial World Congress of IFAC*. CDROM, 1252.pdf.
- Moura, M. M. M., Leiria, A. & Ruano, M. G. (2003). Adaptive system for blood flow estimation, *Proc. of WISP'2003 - IEEE Symposium in Intelligent Signal Processing* pp. 121–125.
- Nikolov, S. I., González, J. P. G. & Jensen, J. A. (2003). Real time 3d visualization of ultrasonic data using a standard pc, *Ultrasonics* **41**: 412–426.
- Ousterhout, J. K. (1998). Scripting: Higher level programming for the 21st century, *IEEE Computer Magazine* pp. 23–30.

- Picone, J., Prezas, D., Hartwell, W. & LoCicero, J. (1988). Spectrum estimation using an analytic signal representation, *Signal Processing* **15**(2): 169–182.
- Pola, S., Macerata, A., Emdin, M. & Marchesi, C. (1996). Estimation of the power spectral density in nonstationary cardiovascular time series: assessing the role of the time-frequency representations (tfr), *IEEE Trans. on Biomedical Engineering* **43**(1): 46.
- Proakis, J. G. & Manolakis, D. G. (1988). *Introduction to Digital Signal Processing*, second edn, MacMillan Publishing Company, New York.
- Ruano, A. E. B. (2000). Sistemas de tempo real, Sebenta de apoio à Disciplina de Opção, Universidade do Algarve, Faro, Portugal.
- Ruano, M. G. (1992). *Investigation of real time spectral analysis techniques for the use with pulsed ultrasound Doppler blood flow detectors*, PhD thesis, University College of North Wales, Bangor, U. K.
- Ruano, M. G. & Fish, P. (1992). Cost/benefit selection of spectral estimators for use in a doppler blood flow instrument, *Proceedings of the IEEE 1992 International Conference on Acoustics, Speech, and Signal Processing* **V**: 513–516.
- Ruano, M. G. & Fish, P. (1993). Cost/benefit criterion for selection of pulsed doppler ultrasound spectral mean frequency and bandwidth estimators, *IEEE Trans. on Biomedical Eng.* **40**(12): 1338–1341.
- Ruano, M. G., García-Nocetti, D. F., Fish, P. & Fleming, P. J. (1992). A spectral estimator using parallel processing for use in a doppler blood flow instrument, in W. Joosen & E. Milgrom (eds), *Parallel Computing: from Theory to Sound Practice*, IOS Press, pp. 397–400.
- Ruano, M. G., García-Nocetti, D. F., Fish, P. & Fleming, P. J. (1993). Alternative parallel implementations of an ar modified covariance spectral estimator for diagnostic ultrasonic blood flow studies, *Parallel Computing* pp. 463–476.
- Ruano, M. G. & Madeira, M. M. (1999). Introdução ao processamento paralelo, Sebenta de apoio à Disciplina de Opção, Universidade do Algarve.
- Ruano, M. G., Moura, M. M. M., Leiria, A., Garcia, F. & Moreno, E. (2003). A framework for blood flow analysis and research, *EMBS'03 - Conference of Enginnering in Medice and Biology Society* . CDROM, P3165.pdf.

- Sahni, S. & Thanvantri, V. (1996). Performance metrics: Keeping the focus on runtime, *IEEE Parallel and Distributed Technology* pp. 43–56.
- Scott, R. (2000). Doing hartley smartly, *Embedded Systems Programming* **13**(10). <http://www.embedded.com/2000/0009/0009feat3.html>.
- Sieben, M., Conradi, J. & Dodss, D. E. (1999). Optical single side band transmission at 10 gb/s using only electrical dispersion compensation, *Journal of Light. Tech.* **17**(10): 1742–1749.
- Smith, J. L., Evans, D. H., Lingke, F., Bell, P. R. F. & Naylor, A. R. (1996). Differentiation between emboli and artifacts using dual-gated transcranial doppler ultrasound, *Ultrasound in Medicine & Biology* **22**(8): 1031–1036.
- Somerville, I. (1992). *Software Engineering*, fourth edn, Addison-Wesley Publishers Ltd., USA.
- Sousa, L. (1999). Simulação de fluxo sanguíneo em válvula mitral, Universidade do Algarve, Faro, Portugal. Disciplina de Projecto do Curso de Licenciatura em Informática de Gestão.
- Texas Instruments (1991). *TMS320C40 User's Guide*, Texas Instruments, USA.
- The Math Works Inc. (1991). *Matlab users' guide*, The Math Works Inc., South Natick, MA01760,USA.
- Tokhi, M. O. & Hossain, M. A. (1995). Cisc, risc and dsp processors in real-time signal processing and control, *Microprocessors & Microsystems* **19**(5): 291–300.
- Tokhi, M. O., Ramos-Hernandez, D. N., Chambers, C. & Hossain, M. (1997). Performance evaluation of dsp, risc and transputer based systems in real-time implementation of signal processing and control algorithms, *Preprints of 4th IFAC Workshop on Algorithms and Architectures for Real-Time Control* pp. 275–280.
- Tortoli, P., Guidi, F., Guidi, G. & Atzeni, C. (1996). Spectral velocity profiles for detailed untrasound flow analysis, *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* **43**(4): 654–659.

- Transtech Parallel Systems Ltd. (1992). *The i860 Toolset, Transtech i860 Parallel Programming System*, Transtech Parallel Systems Limited.
- Vaitkus, P. J. & Cobbold, R. C. (1988). A comparative study and assessment of doppler ultrasound spectral estimation techniques, part i: Estimation methods, *Ultrasound in Medicine & Biology* **14**(8): 661–672.
- Vistas, M. & Reis, H. (1999). Desenvolvimento de um ambiente de programação suportando unidades heterógeneas de processamento, Relatório de Projecto do Curso de Licenciatura em Informática de Gestão, Universidade do Algarve, Faro, Portugal.
- Wang, Y. & Fish, P. J. (1996). Arterial doppler signal simulation by time domain processing, *European Journal of Ultrasound* **3**: 71–81.
- Wells, P. N. T. (1998). Ultrasound in vascular pathologies, *European Radiology* **8**: 849–857.

Apêndice A

Formatos de Ficheiros Estrangeiros

O sistema de processamento de sinais biomédicos suporta a conversão de ficheiros em formatos estrangeiros, isto é, específicos de outros sistemas.

Os formatos referidos no capítulo 4 são aqui descritos detalhadamente.

Note-se a origem e autoria em cada caso.

A.1 Sinais Simulados

Os sinais simulados produzem ficheiros ASCII, sem qualquer cabeçalho, em que os elementos de dados em notação científica são separados por espaço. Cada ficheiro contém a simulação do sinal Doppler de um ciclo cardíaco.

A.2 TIM

Origem: Enviado por Francesco Guidi.

Autoria : Piero Tortoli, Francesco Guidi, Gabriele Guidi and Carlo Atzeni.

Ver: "Spectral Velocity Profiles for Detailed Ultrasound Flow Analysis" in *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 43, no. 4, July 1996, pg. 654-59.

Os ficheiros *TIM* (TIME domain acquisition) contêm sinais obtidos com uma sonda *multigate*.

O eco recebido no transdutor é demodulado por uma portadora em fase e em quadratura, dando origem a um sinal complexo que é amostrado a uma frequência

suficientemente alta produzindo 64 amostras complexas (I e Q).

Os dados provenientes de um primeiro impulso são armazenados pela ordem $r_0, r_1, \dots, r_{63}, i_0, i_1, \dots, i_{63}$. Qualquer um desses elementos é um inteiro com sinal, representado por 16 *bit* (1 bit de sinal e 15 bit de dados). Assim, os dados associados a um impulso têm 256 *byte* de comprimento. Os dados de impulsos subsequentes são acrescentados ao mesmo ficheiro cujo comprimento máximo é 4 *Mbyte*, equivalente a 8192 impulsos.

A frequência de amostragem (F_b) utilizada situa-se na gama de $2 \dots 8$ *MHz* e a frequência de repetição de impulsos (PRF)¹ situa-se na gama de $2 \dots 8$ *kHz*. No caso particular dos ficheiros que foram fornecidos considere-se $F_b = 3.8$ *MHz* e $PRF = 8.5$ *kHz*.

A.3 WAV

Conforme descrito na documentação do pacote *wavtools*(Ligertwood, 2002) da distribuição Debian Linux, os ficheiros WAV são compostos por cabeçalho e dados.

OS ficheiros WAV são uma das variantes dos ficheiros PCM e a estrutura básica do cabeçalho é:

```
struct{
    char RiffTag[4]; /* 'RIFF' */
    int32_t FileLength;
    char FormatTag[8]; /* 'WAVEfmt ' */
    int32_t FormatLength; /* 16 */
    int16_t DataFormat;
    int16_t NumChannels;
    int32_t SampleRate;
    int32_t BytesPerSecond;
    int16_t BlockAlignment;
    int16_t SampleDepth;
    char DataTag[4]; /* 'data' */
    int32_t DataLength;
} WavHeader;
```

em que

¹Em inglês *Pulse Repetition Frequency*

FileLength O comprimento do ficheiro deste campo em diante, igual a 32 (demais campos do cabeçalho) + $DataLength(byte)$.

DataFormat O formato dos dados amostrados. Ao formato áudio PCM corresponde o valor 1 para este parâmetro.

NumChannels 1 ou 2 correspondendo a mono ou estéreo, respectivamente.

SampleRate Número de amostras por segundo.

BytesPerSecond Número de *byte* por segundo. Este valor é útil na estimação do tamanho do *buffer*. $SampleRate \times (SampleDepth/8) \times NumChannels$

BlockAlignment Número de *byte* que devem ser lidos (ou múltiplos deste) para que os dados sejam correctamente geridos pelo dispositivo. $(SampleDepth/8) \times NumChannels$

SampleDepth Número de *bit* por amostra, geralmente 8 ou 16.

DataLength Comprimento dos dados amostrados em *byte*.

$$RecordTime(segundos) \times (SampleDepth/8) \times SampleRate \times NumChannels$$

Origem dos ficheiros: recolhidos por Fabían García-Nocetti, UNAM, IIMAS com um protótipo de um ecógrafo portátil.

A.4 SON

```
struct{
    float temp_res,      /* temporal resolution (mseg.) */
          Fd,           /* Doppler frequency (Hz) */
          angle,       /* Angle between artery and beam */
          f_range,     /* Frequency range (Hz) */
          f_het;       /* Heterodyne frequency (kHz) */
    int32_t dlength;    /* Data column length */
    char Txt1[11];      /* read up to spike rejection processing*/
    char Txt2[91];      /* complete the reading of the header*/
} SonHeader;
```

Ao cabeçalho seguem-se os dados de tipo "short" que constituem uma matriz de 128 linhas de dimensão *dlength*.

```
snggram=fread(fid,[128, dlength],'short');
```

A.5 ALG

Este formato é mais complicado, é talvez para retirar de futuras versões se não se obtiver a descrição completa. Os dados surgem entre um cabeçalho e um fecho.

Contém sete canais de dados: 1 - envelope da frequência máxima; 2 - NA; 3 - NA; 4 - pressão sanguínea; 5 - ECG; 6 - NA; 7 - NA.

```

struct{
    float temp_res,    /* temporal resolution (mseg.) */
        Ft,          /* Transmitted Doppler frequency (MHz) */
        angle,       /* Angle between artery and beam */
        f_range,     /* Frequency range (kHz) */
        f_het;       /* Heterodyne frequency (kHz) */
    int32_t dlength;  /* Data length */
    char Txt1[11];    /* read up to spike rejection processing */
    char Txt2[91];    /* complete the reading of the header */
} AlgHeader;

    short data = fread(fid,[7, dlength],'short'); % Data

struct{
    short Fs,         /* Analogue sampling frequency (Hz) */
        channels,     /* Number of channels */
        full_range[3]; /* Full range of values */

    dummy=fread(fid,1,'float');
    dummy=fread(fid,18,'char');

    freetext=fread(fid,200,'char');
    freetext=setstr(freetext);
    marks=fread(fid,inf,'long');
} AlgTrailer;

```

A.6 DOP

```

struct{
    float temp_res,      /* temporal resolution (mseg.) */
        Ft,             /* Transmitted Doppler frequency (MHz) */
        angle,         /* Angle between artery and beam */
        f_range,       /* Frequency range (kHz) */
        f_het;         /* Heterodyne frequency (kHz) */
    int32_t dlength;    /* Data length */
    char Txt1[103];     /* complete the reading of the header */
} AlgHeader;

    data=fread(fid,[3, dlength], 'short'); % Data

```

A.7 EMB

A estrutura dos ficheiros "EMB" que contêm sinais embólicos é composta por três partes:

I. Cabeçalho:

```

bytes 1-2: Data length (int * 1) /* (em grupos de 50;
                                   de momento sempre 600) */
bytes 3-6: Sampling frequency (float * 1)
bytes 7-8: Numbers of markers (int * 1)
bytes 9-128: Reserved

```

II. Corpo:

```

bytes 1-100: Data for channel A (int*50)
bytes 101-200: Data for channel B (int*50)

bytes 201-300: Data for channel A (int*50)
bytes 301-400: Data for channel B (int*50)

```

etc.....

III. Fecho:

bytes 1-200: General information (char * 200)
 (patient specific)
bytes 201-400: Individual information (char * 200)
 (Embolus specific)
bytes 401-600: If (number of markers>0) Marker positions
 (int * 100)

A.8 MIX

Os ficheiros "MIX" contêm também sinais embólicos. A sua estrutura é composta por três blocos de informação e um de dados.

Bloco I:

bytes 1-4: temporal resolution (float * 1)
bytes 5-8: frequency of the used Doppler system (MHz)
 (float * 1)
bytes 9-12: angle used(degree) (float * 1)
bytes 13-16: frequency range(12.5kHz) (float * 1)
bytes 17-20: heterodyne frequency(kHz) (float * 1)
bytes 21-24: data length in blocks (50 samples/blocks)
 (long * 1)
byte 25: is there a signal in the reverse channel (y/n)
 (char * 1)
byte 26: is the limited range processing used (y/n) (char * 1)
byte 27: is the counter-aliasing processing used (y/n)
 (char * 1)
bytes 28-29: the high pass filter cut-off frequency bin
 (int * 1)
bytes 30-31: the PFR (in bin numbers) (int * 1)
bytes 32-33: the counter-aliasing range (in bin numbers)
 (int * 1)
bytes 34-35: the threshold used in spike rejection (limiar para
 rejeição de picos espúrios)
 (in bin numbers) (int * 1)
byte 36: is the spike rejection processing used(y/n)(char * 1)
bytes 37-128: reserved

Bloco II:

bytes 129-328: free text(char * 200)

Bloco III:

bytes 329-330: number of signals inserted (int * 1)
bytes 331-332: number of channels used (int * 1)
bytes 333-372: 40 bytes for 10 long integers for inserting
point in channel ONE (long * 10)
bytes 373-412: 40 bytes for 10 long integers for inserting
point in channel TWO (long * 10)
bytes 413-432: 20 bytes for 10 integers for starting points
of signals chosen (int * 10)
bytes 433-452: 20 bytes for 10 integers for end points of
signals chosen (int * 10)
bytes 453-492: 40 bytes for 10 floating-point numbers for
the SNRs used in dB channel ONE (float * 20)
bytes 493-532: 40 bytes for 10 floating-point numbers for
the SNRs used in dB channel TWO (float * 20)
bytes 533-572: 40 bytes for 10 floating-point numbers for
the signals coefficients (to control SNR)
used in channel ONE (float * 20)
bytes 573-612: 40 bytes for 10 floating-point numbers for
the signals coefficients (to control SNR)
used in channel TWO (float * 20)
bytes 613-652: 40 bytes for 10 floating-point numbers for
the signal delays used in channel TWO (in ms)
(float * 20)
bytes 653-702: 50 bytes for the name of noise file No.1
(float * 25)
bytes 703-752: 50 bytes for the name of noise file No.2
(float * 25)
bytes 753-1552: 800 bytes for the file names of the signals
inserted (each name can be up to 80 characters,
there can be up to 10 names) (float * 400)

Bloco IV:

bytes 1553-1652: Data for channel ONE (int*50)

bytes 1653-1752: Data for channel TWO (int*50)

bytes 1753-1852: Data for channel ONE (int*50)

bytes 1853-1952: Data for channel TWO (int*50)

etc.....

Apêndice B

Parametrização dos Estimadores Espectrais

A gestão da correspondência entre as opções oferecidas em modo gráfico e as utilizadas pelas aplicações que implementam os vários estimadores espectrais é mantida por um módulo de ligação em *python*, aqui apresentado.

```
#!/usr/bin/env python
"""Este módulo apoia a gestão dos parâmetros \
    dos métodos de estimação espectral """

import sys, os, string

met_descricao = ['STFT', 'AR-MC', 'CWD', 'BSD']      # Para o utilizador
met_descritor_i = ['stft', 'armc', 'cwd', 'bsd']    # interno ao código
met_descritor_c = ['gsfw', 'gmcv', 'gcww', 'gbsw'] # para sequências
met_descritor_d = ['gsfwd', 'gmcd', '', 'gcwd']    # para o FIFO

jan_descricao = ['Barlett', 'Boxcar', 'Hanning', 'Hamming']
jan_descritor = ['bar', 'box', 'han', 'ham']

stdsize = [' 64', '128', '256', '512', '1024']

siana_descricao = ['Não usar', 'Standard', 'Decimado', 'Interpolado']
siana_descritor = ['', 'gsan', 'gsad', 'gsai']

modsize = ['4', '6', '8', '10']
```

```

krn_descricao = ['Original', 'Interferência Reduzida (RID)',
                 'Sem Alias (AF)', 'RI-AF']
krn_descritores = ['', 'ri', 'af', 'riaf']

class STFT:
    """Short Time Fourier Transform class"""
    def __init__(self, iWin = 1, iSize = 256, iStep = 128, iSiana = 0):
        self.iWin = iWin
self.iSize = iSize
self.iStep = iStep
        self.iSiana = iSiana

    def __str__(self):
conteudo = ""
conteudo = conteudo + jan_descritores[self.iWin] + ' '
conteudo = conteudo + 'self.iSize' + ' '
conteudo = conteudo + 'self.iStep' + ' '
conteudo = conteudo + siana_descritores[self.iSiana] + ' '
return conteudo

    def guardar(self, nome, fich):
        conteudo = nome + ' = parametros.STFT('
        conteudo = conteudo + 'self.iWin' + ', '
        conteudo = conteudo + 'self.iSize' + ', '
        conteudo = conteudo + 'self.iStep' + ', '
        conteudo = conteudo + 'self.iSiana' + ') '

f = open(fich, 'w')
f.write('if \'parametros\' not in dir():\n')
f.write('    import parametros\n')
f.write('\n')
f.write(conteudo)
f.write('\n')
f.close()

class ARMC:
    """Auto Regressive Modified Covariance class"""
    def __init__(self, iWin = 1, iSize = 256, iStep = 128,

```

```

    iSiana = 0, iOrdem = 4):
        self.iWin = iWin
self.iSize = iSize
self.iStep = iStep
        self.iSiana = iSiana
        self.iOrdem = iOrdem

    def __str__(self):
conteudo = ""
conteudo = conteudo + jan_descritor[self.iWin] + ' '
conteudo = conteudo + 'self.iSize' + ' '
conteudo = conteudo + 'self.iStep' + ' '
conteudo = conteudo + siana_descritor[self.iSiana] + ' '
conteudo = conteudo + 'self.iOrdem' + ' '
return conteudo

    def guardar(self, nome, fich):
        conteudo = nome + ' = parametros.ARM('
        conteudo = conteudo + 'self.iWin' + ', '
        conteudo = conteudo + 'self.iSize' + ', '
        conteudo = conteudo + 'self.iStep' + ', '
        conteudo = conteudo + 'self.iSiana' + ', '
        conteudo = conteudo + 'self.iOrdem' + ')
f = open(fich, 'w')
    f.write('if \'parametros\' not in dir():\n')
    f.write('    import parametros\n')
    f.write('\n')
    f.write(conteudo)
    f.write('\n')
    f.close()

class CWD:
    """Choi-Williams Distribution class"""
    def __init__(self, iWin = 1, iSize = 256, iStep = 128,
iSiana = 0, iKernel = 0, iEme = 30, dSigma = 5.0):
        self.iWin = iWin
self.iSize = iSize
self.iStep = iStep

```

```

        self.iSiana = iSiana
    self.iKernel = iKernel
    self.iEme = iEme
    self.dSigma = dSigma

    def __str__(self):
    conteudo = ""
    conteudo = conteudo + jan_descritor[self.iWin] + ' '
    conteudo = conteudo + 'self.iSize' + ' '
    conteudo = conteudo + 'self.iStep' + ' '
    conteudo = conteudo + siana_descritor[self.iSiana] + ' '
    conteudo = conteudo + krn_descritor[self.iKernel] + ' '
    conteudo = conteudo + 'self.iEme' + ' '
    conteudo = conteudo + 'self.dSigma' + ' '
    return conteudo

    def guardar(self, nome, fich):
        conteudo = nome + ' = parametros.CWD('
        conteudo = conteudo + 'self.iWin' + ', '
        conteudo = conteudo + 'self.iSize' + ', '
        conteudo = conteudo + 'self.iStep' + ', '
        conteudo = conteudo + 'self.iSiana' + ', '
        conteudo = conteudo + 'self.iKernel' + ', '
        conteudo = conteudo + 'self.iEme' + ', '
        conteudo = conteudo + 'self.dSigma' + ')
    f = open(fich, 'w')
        f.write('if \'parametros\' not in dir():\n')
        f.write('    import parametros\n')
        f.write('\n')
        f.write(conteudo)
        f.write('\n')
        f.close()

class BSD:
    """Bessel Distribution class"""
    def __init__(self, iWin = 1, iSize = 256, iStep = 128,
    iSiana = 0, dAlfa = 2.0):
        self.iWin = iWin

```

```
self.iSize = iSize
self.iStep = iStep
    self.iSiana = iSiana
    self.dAlfa = dAlfa

    def __str__(self):
conteudo = ""
conteudo = conteudo + jan_descritor[self.iWin] + ' '
conteudo = conteudo + 'self.iSize' + ' '
conteudo = conteudo + 'self.iStep' + ' '
conteudo = conteudo + siana_descritor[self.iSiana] + ' '
conteudo = conteudo + 'self.dAlfa' + ' '
return conteudo

    def guardar(self, nome, fich):
    conteudo = nome + ' = parametros.BSD('
    conteudo = conteudo + 'self.iWin' + ', '
    conteudo = conteudo + 'self.iSize' + ', '
    conteudo = conteudo + 'self.iStep' + ', '
    conteudo = conteudo + 'self.iSiana' + ', '
    conteudo = conteudo + 'self.dAlfa' + ')
f = open(fich, 'w')
    f.write('if \'parametros\' not in dir():\n')
    f.write('    import parametros\n')
    f.write('\n')
    f.write(conteudo)
    f.write('\n')
    f.close()

class WAVE:
    """Wave definitions class"""
    def __init__( self, NumChannels = 1, SampleRate = 22050,
SampleDepth = 16):
        self.NumChannels = NumChannels
self.SampleRate = SampleRate
self.SampleDepth = SampleDepth

    def __str__(self):
```

```
conteudo = ""
conteudo = conteudo + 'self.NumChannels' + ' '
conteudo = conteudo + 'self.SampleRate' + ' '
conteudo = conteudo + 'self.SampleDepth' + ' '
return conteudo

def guardar(self, nome, fich):
    conteudo = nome + ' = parametros.WAVE('
    conteudo = conteudo + 'self.NumChannels' + ', '
    conteudo = conteudo + 'self.SampleRate' + ', '
    conteudo = conteudo + 'self.SampleDepth' + ') '
f = open(fich, 'w')
f.write('if \'parametros\' not in dir():\n')
f.write('    import parametros\n')
f.write('\n')
f.write(conteudo)
f.write('\n')
f.close()
```