

JOÃO FERNANDO REFOSCO BAGGIO

**PROCESSAMENTO DE SINAIS DOPPLER PARA
OBTENÇÃO DE SONOGRAMA EM TEMPO REAL**

FLORIANÓPOLIS

2003

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**PROCESSAMENTO DE SINAIS DOPPLER PARA
OBTENÇÃO DE SONOGRAMA EM TEMPO REAL**

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica.


JOÃO FERNANDO REFOSCO BAGGIO

Florianópolis, maio de 2003.


PROCESSAMENTO DE SINAIS DOPPLER PARA OBTENÇÃO DE SONOGRAMA EM TEMPO REAL

JOÃO FERNANDO REFOSCO BAGGIO

‘Esta Tese foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em *Engenharia Biomédica*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’



Prof. Raimes Moraes, PhD
Orientador



Prof. Edson Roberto De Pieri, DSc.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:



Prof. Raimes Moraes, PhD.
Presidente



Prof. Joni da Silva Fraga, DSc.



Prof. Fernando Mendes de Azevedo, DSc.



Prof. Jefferson Luiz Brum Marques, PhD.

A Giovana pela compreensão e apoio.

Aos colegas e todos que comigo conviveram, pelo auxílio e amizade.

Ao Prof. Raimes pela oportunidade concedida, confiança depositada e empenho em orientar-me.

A Deus e aos meus pais Omiro e Salete, sem os quais não teria chegado até aqui.

Obrigado.

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

PROCESSAMENTO DE SINAIS DOPPLER PARA OBTENÇÃO DE SONOGRAMA EM TEMPO REAL

João Fernando Refosco Baggio

Maio / 2003

Orientador: Raimes Moraes, EE, Msc, PhD.

Área de Concentração: Engenharia Biomédica.

Palavras-chave: instrumentação biomédica, fluxo sanguíneo, reconstrução vascular, aterosclerose, ultra-som, registro de sinais biológicos.

Número de Páginas: 91.

RESUMO:

A aterosclerose reduz a *luz de vasos* sanguíneos, causando isquemia. Para manter a irrigação distal à lesão, realizam-se cirurgias de reconstrução vascular que estabelecem um caminho alternativo para o fluxo do sangue.

A monitoração pós-operatória de cirurgias de reconstrução vascular periférica é de grande importância para a detecção de possíveis erros técnicos. Tais erros podem ter como consequência a amputação do membro afetado ou mesmo, a morte do paciente. O diagnóstico prematuro destas falhas possibilita uma segunda intervenção para correção do problema.

Pesquisas vêm sendo realizadas para o emprego de parâmetros quantitativos para avaliar o sucesso de cirurgia de reconstrução vascular. Estes parâmetros são obtidos a partir de formas de onda do fluxo sanguíneo amostradas com ultra-som Doppler.

Visando construir um sistema versátil que possa contribuir para com estas investigações, este trabalho aborda o desenvolvimento de software para aquisição, registro e processamento de sinais Doppler. Estes sinais são gerados por uma placa de ultra-som

Doppler de ondas contínuas. O programa computa o sonograma e as curvas de velocidade máxima e média velocidade, como também os exibe em tempo real na tela do PC.

Buscando reduzir os custos e aumentar a flexibilidade de tais sistemas, o programa desenvolvido substitui Placa de Processamento Digital de Sinais inserido em slot do PC que vem sendo utilizada para realizar tais tarefas. A amostragem dos sinais é realizada com a Placa de Som existente em PCs comerciais. Bibliotecas otimizadas e técnicas de programação permitem realizar este processamento em tempo real.

Como o sistema MS-Windows™ não foi desenvolvido para aplicações em tempo real, este trabalho discute as técnicas utilizadas para garantir os resultados obtidos.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

DOPPLER SIGNALS PROCESSING FOR SONOGRAM COMPUTING IN REAL TIME

João Fernando Refosco Baggio

May / 2003

Advisor: Raimes Moraes, PhD.

Area of Concentration: Biomedical Engineering.

Keywords: biomedical instrumentation, blood flow, vascular reconstruction, atherosclerosis, ultrasound, registration of biological signals.

Number of Pages: 91.

ABSTRACT:

Arteriosclerosis reduces the lumen of blood vessels leading to ischaemia. Vascular surgeries implant grafts to bypass the lesions, allowing the perfusion of the distal tissue.

The postoperative follow-up of the vascular reconstruction is highly desirable to detect technical errors. These errors can cause member amputation or even, the patient death. These outcomes can be avoided by the premature diagnosis of graft failures that allows a second intervention to repair it.

Researchers have investigated quantitative parameters obtained from blood flow waveforms to assess vascular reconstruction surgeries. These waveforms are sampled by ultrasound Doppler.

Aiming to build a versatile system to research these parameters, this work presents software developed to acquire and process Doppler signals. A Continuous Wave Ultrasound Doppler board generates the signals to be processed. The software computes the sonogram, maximum and mean velocity curves that are shown on the PC screen in real time.

Reducing costs and increasing the flexibility, the developed software replaces DSP (Digital Signals Processor) boards to be inserted into PC slots that are generally used to carry out these tasks.

The signals are sampled via the Sound card that are available in commercial PCs. Optimized libraries and programming techniques allow the heavy processing to be accomplished in real time.

Since MS-Windows™ Operating System was not developed to run software in real time, this work discusses the techniques that guarantee such achievement.

SUMÁRIO

LISTA DE FIGURAS	x
ABREVIATURAS	xiv
1. Introdução.....	1
1.1. O Sistema Cardiovascular e a Circulação Sanguínea	1
1.2. Métodos de Diagnóstico	6
1.2.1. <i>Ultra-som Doppler de Ondas Contínuas</i>	7
1.3. Objetivo	7
1.4. Justificativa	8
1.5. Organização do Trabalho	8
2. Fundamentação Teórica	10
2.1. Ultra-som.....	10
2.1.1. <i>Efeito Doppler</i>	11
2.1.2. <i>Sistema de Ultra-som Doppler de Ondas Contínuas (SUDOC)</i>	14
2.2. Processamento Digital de Sinais Doppler	15
2.2.1. <i>Sinal em Quadratura de Fase</i>	15
2.2.2. <i>Sonograma</i>	19
2.2.3. <i>Transformada de Fourier</i>	21
2.2.4. <i>Janelamento</i>	25
2.2.5. <i>Envoltórias</i>	26
3. Metodologia.....	29
3.1. Hardware Utilizado	30
3.1.1. <i>Placa Doppler</i>	30
3.1.2. <i>Aquisição dos Sinais Doppler</i>	31
3.2. Software Implementado	36
3.2.1. <i>Descrição do Software</i>	37
3.2.2. <i>Diagrama de Blocos do Software Implementado</i>	41
3.2.3. <i>Aquisição dos Sinais</i>	43
3.2.4. <i>Coordenação do Processamento em Tempo Real</i>	49
3.2.5. <i>Implementação dos Cálculos</i>	54

3.2.6. <i>Exibição de Sonograma e Envoltórias</i>	60
3.2.7. <i>Registro dos Sinais em Arquivo</i>	62
4. Resultados	67
4.1. Teste de Processamento dos Sinais Doppler em Quadratura de Fase.....	67
4.1.1. <i>Discriminação da Direção do Fluxo</i>	68
4.2. Análise Temporal do Software	69
4.2.1. <i>Tempo de Acesso ao Vídeo do PC</i>	70
4.2.2. <i>Tempo de Cálculo da FFT</i>	70
4.2.3. <i>Comportamento das Tarefas no Tempo</i>	70
4.2.4. <i>Gravando Dados no Disco Rígido</i>	77
4.2.5. <i>Interação com Placa Doppler</i>	78
5. Discussão, Conclusão e Trabalhos Futuros	80
5.1. Conclusão	82
5.2. Trabalhos Futuros	83
6. GLOSSÁRIO	84
7. REFERÊNCIAS BIBLIOGRÁFICAS	85

LISTA DE FIGURAS

Figura 1.1 – Esquema do circuito cardiovascular.	2
Figura 1.2 – Distribuição de óbitos por principais grupos de causas nas capitais Brasileiras (Ministério da Saúde, 2002).	2
Figura 1.3 – Causas mais frequentes de morte no Brasil (Ministério da Saúde, 2002).....	3
Figura 1.4 – Camadas dos vasos sanguíneos.....	3
Figura 1.5 – Representação das cirurgias de reconstrução vascular.....	4
Figura 1.6 – Falhas mais comuns em cirurgias de reconstrução vascular por <i>bypass</i>	5
Figura 2.1 – Efeito Doppler.....	12
Figura 2.2 – Reflexão da onda sonora.	13
Figura 2.3 – Transdutor de ultra-som.	14
Figura 2.4 – Obtenção de sonograma.	15
Figura 2.5 – Placa Doppler.....	16
Figura 2.6 – Demodulador em quadratura de fase.....	17
Figura 2.7 – Sonograma bem delineado.	20
Figura 2.8 – Sonograma com espectro disperso.....	21
Figura 2.9 – Discriminação de fluxo direto através da CFT.	23
Figura 2.10 – Discriminação de fluxo reverso através da CFT.....	23
Figura 2.11 – DFT complexa.	24
Figura 2.12 – Janela retangular delimitando sinal.	25
Figura 2.13 – Envoltórias do sonograma.	26
Figura 2.14 – Método geométrico.	28
Figura 3.1 – Etapas constituintes do SUDOC.....	30
Figura 3.2 – Estágios dos CIs utilizados em placas de som.....	33
Figura 3.3 – Instrumentação virtual.....	34

Figura 3.4 – Esquemático do circuito de defasagem.....	34
Figura 3.5 – Tela do Osciloscópio virtual.....	35
Figura 3.6 – Tela principal do software.....	38
Figura 3.7 – Tela de configuração.....	38
Figura 3.8 – Configuração dos arquivos de registro.....	39
Figura 3.9 – Tela com opções avançadas.....	41
Figura 3.10 – Fluxograma.....	42
Figura 3.11 – Fluxograma para uso de <i>DirectSound</i>	44
Figura 3.12 – Sugestão de acréscimo de PAS para digitalizar sinais.....	49
Figura 3.13 – Memória circular.....	51
Figura 3.14 – Principais etapas de processamento dos sinais Doppler.....	54
Figura 3.15 – Sonograma utilizando amostragem de 44100Hz.....	56
Figura 3.16 – Sonograma utilizando amostragem de 22050Hz.....	56
Figura 3.17 – Sonograma sem filtragem.....	57
Figura 3.18 – Sonograma com filtragem.....	57
Figura 3.19 – Base para algoritmo do método geométrico.....	58
Figura 3.20 – Valores para coloração do sonograma.....	62
Figura 3.21 – Tela para exibição de arquivos <i>WAV</i>	65
Figura 4.1 – Sinais obtidos com osciloscópio virtual.....	68
Figura 4.2 – Espectro de um sinal senoidal complexo.....	69
Figura 4.3 – Gráfico de um sinal senoidal complexo.....	69
Figura 4.4 – Ciclo realizado para cada coluna de sonograma gerada.....	71
Figura 4.5 – Sequência de processamento.....	72
Figura 4.6 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 1º caso.....	73
Figura 4.7 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 2º caso.....	74

Figura 4.8 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 3º caso.....	75
Figura 4.9 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 4º caso.....	76
Figura 4.10 – Sonograma sem envoltórias.....	78
Figura 4.11 – Sonograma com envoltórias.....	79
Figura 4.12 – Envoltórias de frequência máxima e média.....	79

LISTA DE TABELAS

Tabela 2.1 – Velocidade do som conforme o meio de propagação.....	10
Tabela 3.1 – Configuração mínima para PAS (Placa de Aquisição de Sinais).....	31
Tabela 3.2 – <i>Audio Minimum Performance Requirements</i>	32
Tabela 3.3 – Estruturas dos dados de áudio.	46
Tabela 3.4 – Formatos de arquivos de áudio.....	63
Tabela 4.1 – Configuração dos PCs utilizados para teste do software.....	67
Tabela 4.2 – Tempo de acesso ao vídeo do PC utilizando diferentes métodos.....	70
Tabela 4.3 – Valores obtidos a partir de testes com a CPU1.	74
Tabela 4.4 – Valores obtidos a partir de testes com a CPU2.	74
Tabela 4.5 – Valores obtidos a partir de testes com a CPU3.	75
Tabela 4.6 – Valores obtidos a partir de testes com a CPU4.	76
Tabela 4.7 – Valores obtidos a partir de análise estatística dos tempos de computação.....	76

ABREVIATURAS

μ s	microsegundos.
AGP	Amplificador de Ganho Programável.
API	<i>Application Program Interface</i> – Interface para programação de aplicativos.
CFT	<i>Complex Fourier Transformer</i> – Transformada Complexa de Fourier.
DFT	<i>Discreet Fourier Transformer</i> – Transformada Discreta de Fourier.
ECG	Eletrocardiograma.
EEG	Eletroencefalograma.
FFT	<i>Fast Fourier Transformer</i> – Transformada Rápida de Fourier.
FS	Fluxo Sanguíneo.
FT	<i>Fourier Transformer</i> – Transformada de Fourier.
GUID	<i>Globally Unique Identifiers</i> – Identificador Único Global, ou identificador globalmente exclusivo.
Hz	Unidade de medida de frequência de um fenômeno periódico, ciclo por segundo.
ISA	<i>Industry Standard Architecture</i> . Barramento interno.
ms	milisegundos.
MSDN	<i>Microsoft Development Network</i> , organização para programadores de software Microsoft compatível.
NT	<i>New Technology</i> (Windows NT).
PA	Pressão Arterial.
PAS	Placa de Aquisição de Sinais.
PC	<i>Personal Computer</i> – Plataforma IBM-PC compatível.
PCI	<i>Peripheral Component Interconnect</i> . Barramento interno.
PCM	<i>Pulse Code Modulation</i> - Modulação por código de pulso.
PPDS	Placa de Processamento Digital de Sinais.

SO	Sistema Operacional.
SUDOC	Sistema de Ultra-som Doppler de Ondas Contínuas.
USB	<i>Universal Serial Bus</i> – Protocolo de comunicação serial, para conexão de dispositivos, periféricos ao PC.
WAV	Formato de som <i>Wave Microsoft</i> . Tipo de arquivo usado para armazenar áudio digitalizado.

CAPÍTULO 1

1. Introdução

A aterosclerose pode destruir parcial ou totalmente a *luz* do vaso sanguíneo, comprometendo o FS (Fluxo Sanguíneo) distal à lesão. Nestes casos, podem ser necessárias cirurgias de reconstrução vascular nas quais a monitoração do FS é de grande importância para detecção de falhas técnicas e na pesquisa de parâmetros para sua avaliação.

Este capítulo apresenta, através de revisão bibliográfica, as características do problema abordado. Descreve ainda, técnicas atualmente utilizadas para avaliação da cirurgia e apresenta proposta de investigação através dos recursos tecnologicamente disponíveis. Em seguida, são justificados os objetivos para viabilizar tal investigação.

O último item deste capítulo descreve, de forma sucinta, a organização do trabalho nos capítulos subsequentes.

1.1. O Sistema Cardiovascular e a Circulação Sanguínea

O sistema cardiovascular é formado por um órgão central, o coração, e por uma vasta rede de vasos distribuída por todo o corpo. Sua principal função é suprir, por intermédio do sangue, nutrientes a todas as células. O sistema cardiovascular desempenha papel vital na troca de gases, regulação da temperatura corporal, secreção e distribuição de hormônios.

O coração é responsável pelo bombeamento do sangue, sendo seu percurso estabelecido pela rede de vasos sanguíneos que alcança demais tecidos do corpo (Figura 1.1).

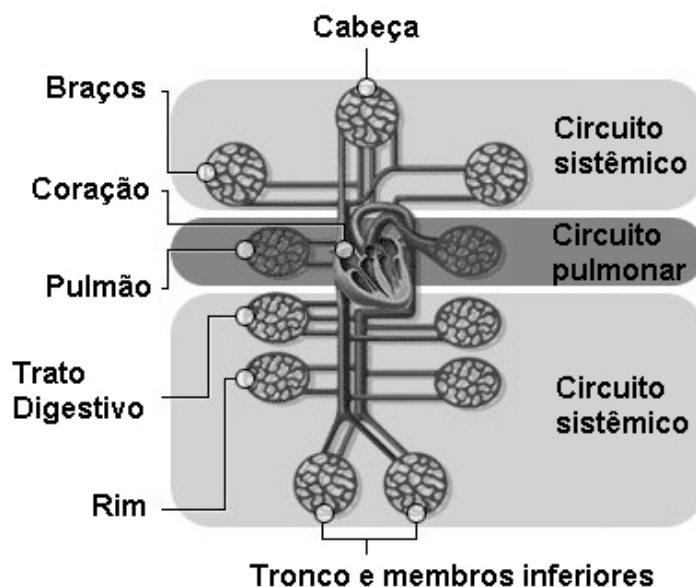


Figura 1.1 – Esquema do circuito cardiovascular.

Representação dos diversos leitos do sistema circulatório que irrigam todo o corpo, composto pelo coração e vasos.

O sistema vascular está sujeito à diversas enfermidades que debilitam a parede dos vasos, que provocam sua dilatação anormal e causam lesões endoteliais.

Nas capitais Brasileiras, as doenças do aparelho circulatório apresentaram um crescimento expressivo nas últimas décadas (Figura 1.2). As doenças do sistema cardiovascular compõem o grupo de causas mais frequente de morte no Brasil (Figura 1.3) (Ministério da Saúde, 2002).

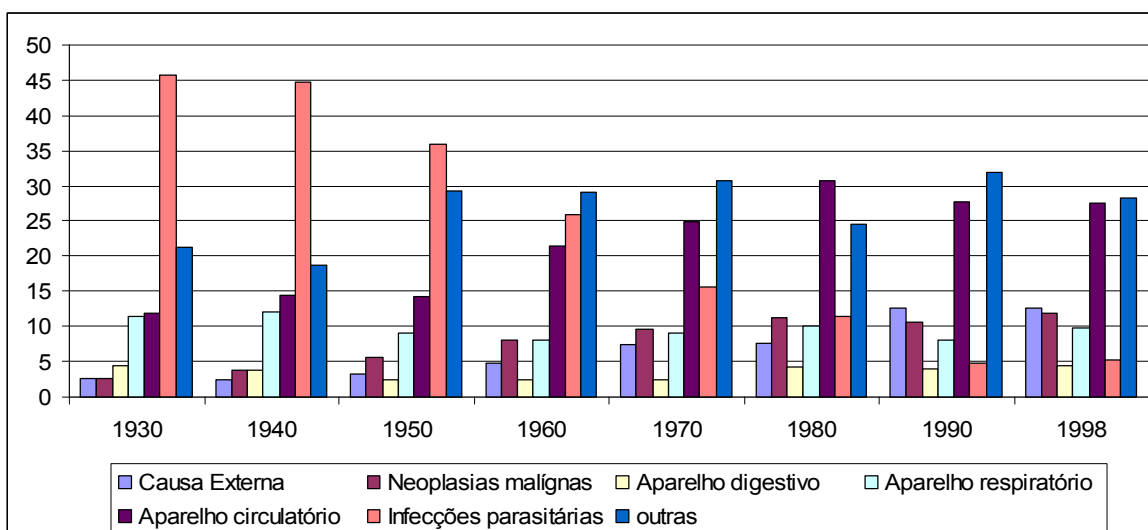


Figura 1.2 – Distribuição de óbitos por principais grupos de causas nas capitais Brasileiras (Ministério da Saúde, 2002).

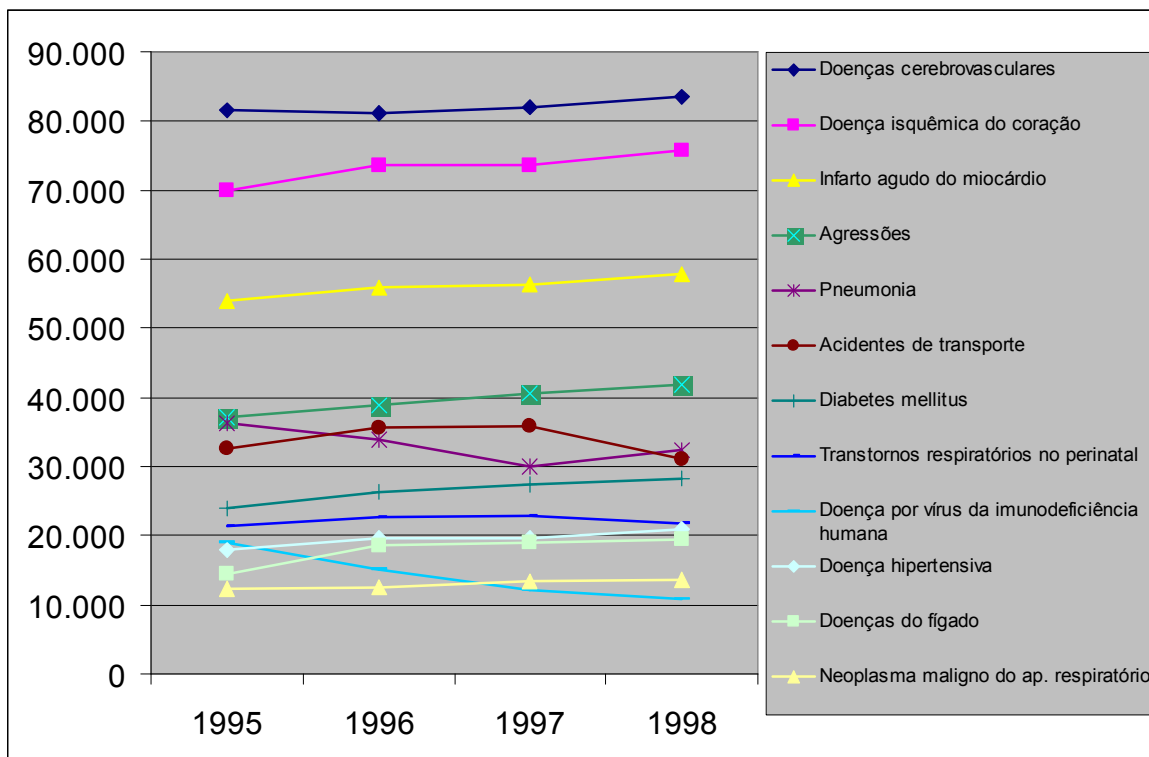


Figura 1.3 – Causas mais frequentes de morte no Brasil (Ministério da Saúde, 2002).

Considerada a origem de moléstias como o acidente vascular cerebral, ou derrame e o infarto agudo do miocárdio, a aterosclerose constitui-se na principal causa de mortes no mundo ocidental (Kadar, 2001).

A aterosclerose é uma doença inflamatória das artérias de médio e grande calibre (Libby, 2002), caracterizada pelo desenvolvimento de lesões gordurosas, denominadas placas ateromatosas no subendotélio (na superfície interna da parede arterial) (Figura 1.4) (Vestraete, 1989; Carvalho, 2001; Foppa, 2001).

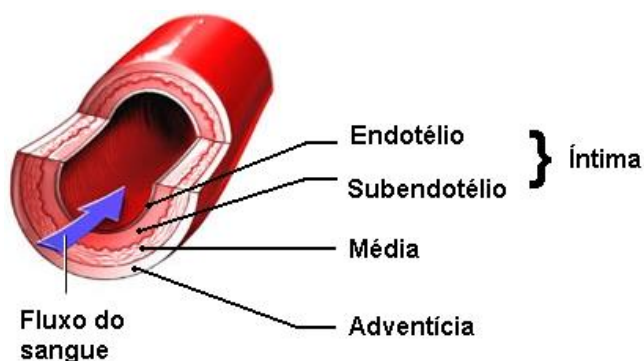


Figura 1.4 – Camadas dos vasos sanguíneos. Representação das túnicas que compõe os vasos de médio e grande calibre.

Em estágio mais avançado, a aterosclerose pode reduzir a *luz* da artéria afetada, provocando estenose e predispondo o vaso a uma trombose. Tais eventos têm como consequência o infarto agudo do miocárdio, acidente vascular cerebral ou doença vascular periférica, de acordo com a artéria afetada.

Nos casos onde a obstrução da *luz* do vaso compromete o fluxo sanguíneo distal à lesão, o médico especialista pode optar por uma intervenção cirúrgica para correção do problema.

São adotados procedimentos clínicos quando a isquemia encontra-se em estágio inicial, caracterizado por ausência de sintomas ou claudicação intermitente. Quando o quadro evolui, apresentando dor em repouso ou lesões tróficas relativas à interrupção da nutrição, a conduta provável é a cirúrgica (Cuneo, 2000; Raymundo, 2001).

Para solucionar o problema cirurgicamente, uma alternativa é a construção de um caminho alternativo para o fluxo do sangue, através de um desvio conhecido como ponte ou *bypass* (Figura 1.5.a). A cirurgia de *bypass* é feita com o enxerto venoso ou através de uma prótese vascular apropriada. Outra solução é desobstruir a artéria, recuperando o canal original. A angioplastia é um procedimento para reabrir ou ampliar a *luz* de um vaso sanguíneo ocluído ou estreitado (Figura 1.5.b).

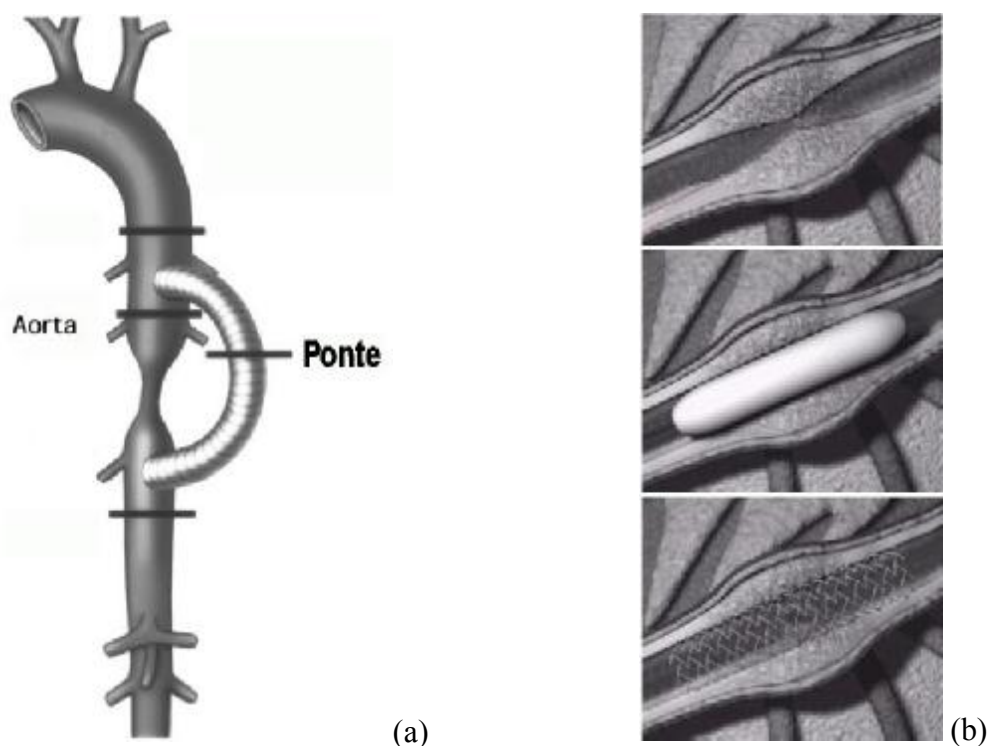


Figura 1.5 – Representação das cirurgias de reconstrução vascular.
Em (a) desvio com ponte, (b) reconstrução por angioplastia, onde um balão é inflado comprimindo o ateroma.

A cirurgia de *bypass* é indicada quando grande extensão da artéria encontra-se comprometida, propiciando um desvio do fluxo para o tecido distal. A angioplastia permite a recuperação de uma região pontual (Figura 1.5.b).

As cirurgias de reconstrução vascular estão sujeitas a falhas (Berkowitz, 1985; Erickson *et al.*, 1996), sendo mais frequentes nos primeiros meses após a intervenção cirúrgica (Steven *et al.*, 2001). Destas falhas, as que surgem logo no primeiro mês pós-operatório são, geralmente, devido a erros técnicos ocorridos durante a cirurgia (Stept *et al.*, 1987; Whittemore *et al.*, 1981). Além destes fatores, a instituição na qual é realizado o implante (Watson *et al.*, 2000), a técnica e o material utilizado no implante (Dougherty *et al.*, 1989) também afetam o sucesso da cirurgia. Todos estes fatores confirmam a necessidade da monitoração pós-operatória. A Figura 1.6 ilustra tipos de falhas em cirurgias de reconstrução vascular.

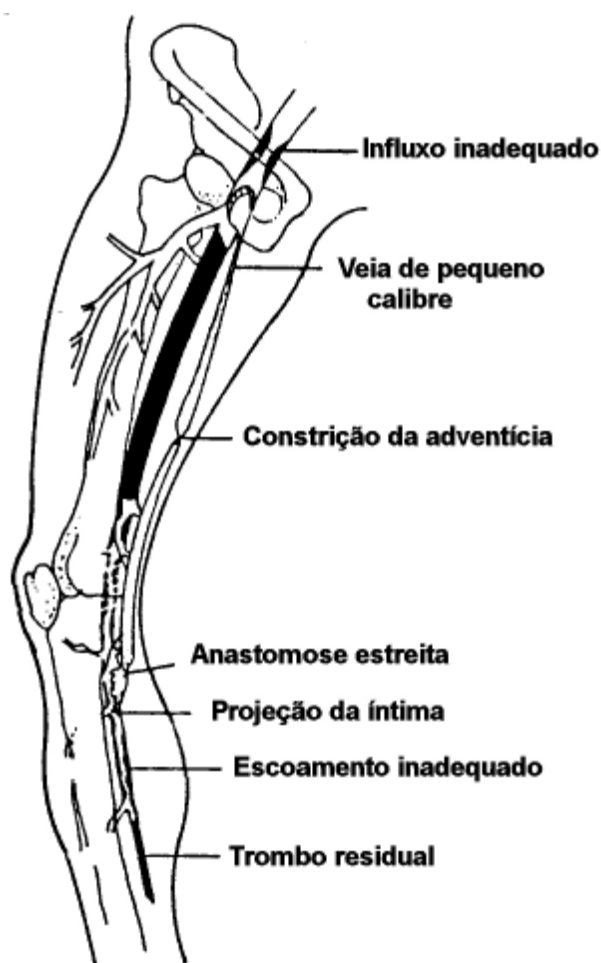


Figura 1.6 – Falhas mais comuns em cirurgias de reconstrução vascular por *bypass*.
Adaptado de Moraes, 1990.

1.2. Métodos de Diagnóstico

Existem diversos métodos de diagnóstico para verificar a integridade das artérias no que se refere ao fluxo do sangue, fornecendo um meio de apoio à propedêutica vascular. Estes métodos possibilitam avaliar as estenoses, fornecendo um indicativo de sua severidade, ou avaliar as cirurgias de reconstrução vascular. Cada método tem suas vantagens e desvantagens, o ideal seria utilizar mais de uma técnica para auxiliar no diagnóstico e na semiologia das doenças vasculares.

Entre as técnicas para avaliação vascular, as mais utilizadas são angiografia, ressonância magnética, medida de pressão e ultra-sonografia.

A angiografia, considerada o padrão ouro, consiste em um método de visualização anatômica dos vasos. Injeta-se um contraste (composto iodado) no vaso sanguíneo que é então radiografado, preferencialmente em mais de um plano. Contudo, esta técnica não fornece informações hemodinâmicas do fluxo sanguíneo. O contraste de iodo é expelido normalmente pelo organismo, mas pode provocar reações alérgicas, devendo ter dosagem controlada para que não cause problemas renais. A angiografia é um exame caro, demorado e invasivo, apresentando desconforto e risco de vida para o paciente.

Uma alternativa à angiografia convencional é a angiografia por ressonância magnética. Esta, permite a avaliação do sistema cardiovascular de maneira minimamente invasiva. A técnica é conhecida há mais de uma década, não necessita de meio de contraste tampouco utiliza radiação ionizante. Contudo, requer um tempo de exposição prolongado e fica susceptível a artefatos. Nos últimos, anos foram desenvolvidas novas técnicas, como a utilização de contraste paramagnético que é menos agressivo que o composto iodado utilizado na angiografia convencional, permitindo exames com tempo de exposição reduzido (inferior a um minuto) e propiciando maior imunidade a artefatos. Contudo, é um exame de alto custo e não pode ser utilizado em pacientes com próteses metálicas, cita-se: marca-passo.

A medida de pressão é uma técnica simples e de baixo custo. Permite a obtenção de índices que revelam reduções no fluxo sanguíneo devido a obstruções. O índice ABI (*Ankle Brachial Index*) é dado pela razão entre a pressão sanguínea sistólica do tornozelo e a do braço. Um valor muito baixo indica fluxo insuficiente. Este método não é rico em informações, apenas apresenta um indicativo para realização de exames mais minuciosos.

Outros exames como a arterioscopia, fonoangiografia, oscilometria ou termometria, também são utilizadas (Plasencia, 1994). Técnicas envolvendo ultra-som,

permitem a obtenção de imagens tridimensionais de órgãos (B-mode) e de informações dinâmicas do FS (Doppler de ondas contínuas ou pulsadas) (Brunner, 2002). O método de interesse para este trabalho é o Ultra-som Doppler de Ondas Contínuas.

1.2.1. Ultra-som Doppler de Ondas Contínuas

O sistema de ultra-som Doppler de ondas contínuas é o mais empregado para o diagnóstico do sistema vascular periférico. Embora seja uma técnica conhecida há mais de três décadas, continua sendo amplamente utilizada em pesquisas relacionadas ao fluxo sanguíneo (Güler, 2002).

O único desconforto para o paciente é a pequena pressão do transdutor sobre a pele. Esta técnica vem sendo utilizada por longa data, não tendo sido reportado quaisquer efeitos colaterais ou nocivos devido ao seu uso na prática clínica, desde que respeitados os limites de intensidade (Kremkau, 1996; Marussi, 2001).

Esta técnica fornece informações relativas ao fluxo sanguíneo a partir das quais é possível extrair diversos parâmetros. Possibilita investigações não só do estado dos vasos que conduzem o sangue, mas também fornece informações indiretas de outras patologias que alteram as características do fluxo (Maldonado, 1999). Podendo ser utilizado para análise não só do FS periférico, mas também de vasos sanguíneos mais profundos (Takase, 1997; Bazzocchi, 1998; Bosan, 1998; Neto, 1998; Panerai 1996 e 1998; Verma, 2000; Xing, 2000; Schuchter, 2001; Diehl, 2002 Seitz, 2001).

O SUDOC (Sistema de Ultra-som Doppler de Ondas Contínuas) apresenta custo relativamente baixo; facilitando portanto sua disseminação. Este trabalho apresenta técnicas de processamento dos sinais Doppler para reduzir ainda mais o seu custo.

1.3. Objetivo

O objetivo deste trabalho reside na investigação e implementação de técnicas para registro e processamento de sinais Doppler em quadratura de fase, visando apresentação de sonograma e suas envoltórias em tempo real na tela do PC utilizando Sistema Operacional MS-Windows™. Visando assim, construir um sistema de baixo custo e interface amigável em ambiente computacional mais largamente empregado pela comunidade médica (Instituto norte-americano de pesquisas RedSheriff, <<http://www.redsheriff.com>>).

1.4. Justificativa

A monitoração do fluxo sanguíneo após operações de reconstrução vascular periférica é importante para diagnóstico prematuro de falhas que podem levar a amputação de membro ou até a morte do paciente.

O sistema Doppler de Ondas Contínuas destaca-se pelo baixo custo, simples operabilidade, possibilidade de exames prolongados e de serem repetidos em curtos intervalos de tempo, além de fornecer informações hemodinâmicas do fluxo sanguíneo de forma não invasiva.

Atualmente pesquisas envolvendo análise das formas de onda do FS obtida a partir de SUDOCs (Sistemas de Ultra-som Doppler de Ondas Contínuas) estão sendo desenvolvidas. A pesquisa de parâmetros quantitativos para detecção de falhas em implantes nas cirurgias de reconstrução vascular vem sendo investigada e utiliza o espectro de velocidades do FS obtido a partir de um SUDOC (Aaslid, 1982; Nizametin, 1994; Paum, 2000; Steiman, 2000; Xing, 2000; Noguchi, 2001; Weitzel, 2001; Güler, 2002).

A utilização de sistemas comerciais apresenta custo proibitivo, além de não possibilitar alterações de software de modo a agregar novos módulos para pesquisa.

Até o presente momento, era comum utilização de Placa de Processamento Digital de Sinais (PPDS) para aquisição e processamento dos sinais Doppler, o que onera o sistema (Ramos, 2002). Utilizando recursos nativos do PC é possível dispensar o emprego da PPDS, conseguindo assim, reduzir custos e facilitar a reprodutibilidade do sistema.

O atual avanço no desempenho de processamento dos microcomputadores e de gerenciamento multimídia, permite a utilização de recursos existentes no próprio PC para realizar a aquisição, registro e processamento dos sinais Doppler, para então serem apresentados na tela do PC em tempo real e com garantias de integridade do sinal gerado.

A redução de custos assim alcançada permite que tal metodologia possa ser disseminada em países em desenvolvimento, possibilitando um melhor atendimento para a população.

1.5. Organização do Trabalho

No primeiro capítulo é realizada uma introdução ao problema de interesse na parte fisiológica envolvida no problema a ser enfrentado. É então apresentado o objetivo e a justificativa do trabalho.

No segundo capítulo, a fundamentação teórica necessária para compreensão das técnicas de processamento utilizadas é abordada. O capítulo é dividido em itens que descrevem o fenômeno físico envolvido (efeito Doppler), matemática envolvida (demodulação em quadratura de fase, transformada de Fourier, cálculo de parâmetros) assim como, as características do sinal e do hardware utilizados para obtenção dos sinais Doppler.

A metodologia utilizada é descrita no terceiro capítulo, descrevendo o sistema desenvolvido, suas características e definições. Discute-se o hardware utilizado, ferramentas de software, recursos e técnicas de programação.

Os resultados são apresentados no quarto capítulo. No capítulo de conclusão, apresenta-se a finalização do trabalho, propondo e instigando o leitor a trabalhos futuros.

CAPÍTULO 2

2. Fundamentação Teórica

Este capítulo aborda os fundamentos essenciais para a implementação do sistema. Trata, primeiramente, das definições e efeitos relacionados ao ultra-som e do sistema Doppler de ondas contínuas. O hardware utilizado e a base das técnicas de processamento de sinais envolvidos também são discutidos neste capítulo.

2.1. Ultra-som

O som é definido como vibração mecânica das partículas de um meio elástico, sendo que, quanto maior a densidade do meio, maior será a velocidade de propagação da onda. Ultra-som são vibrações mecânicas superiores a 20Khz.

A Tabela 2.1 mostra a velocidade do som em alguns meios.

Tabela 2.1 – Velocidade do som conforme o meio de propagação.

Meio	Velocidade do som em m/s (valor aproximado)
Ar	343
Água	1493
ouro	3240
Cérebro	153
Músculo	158
Gordura	145
Osso	404

* Considerando condições normais de temperatura e pressão.

A atenuação do sinal de ultra-som pelo tecido biológico é dependente da frequência, conforme descrito pela Equação 2.1 (Nakajima, 1999). Assim, a escolha da frequência do sinal de ultra-som varia de acordo com sua utilização.

$$\alpha = a \cdot f^b \quad 2.1$$

Onde:

α = coeficiente de atenuação [dB / mm];

f = frequência do sinal de ultra-som;

b : para tecidos biológicos e para frequências abaixo de 10MHz o expoente b tem valor 1.

Para o estudo de estruturas mais profundas, como o polígono de Willis, a veia femoral, ilíacas e cava inferior, é utilizado um sinal com frequência mais baixa, próxima aos 2MHz. Já para vasos próximos à superfície, como é o caso das artérias e veias dos membros, utilizam-se frequências mais altas e portanto com maior poder de resolução. É comumente empregada frequência em torno de 8MHz .

Para emissão e recepção dos sinais de ultra-som é utilizado o fenômeno piezelétrico. A piezeletricidade é uma propriedade intrínseca dos cristais anisotrópicos, como por exemplo, o cristal de quartzo. Um campo elétrico, quando aplicado em um cristal de quartzo, provoca uma deformação mecânica. A pressão mecânica por sua vez, produz polarizações elétricas em determinadas direções do mesmo cristal. Tal efeito torna possível gerar sinais de ultra-som com frequências da ordem de milhões de oscilações por segundo a partir destes cristais (Webster, 1992).

2.1.1. Efeito Doppler

Quando um sinal de ultra-som incide sobre o fluxo sanguíneo, parte deste sinal é refletido pelos eritrócitos em movimento, retornando à fonte emissora com frequência diferente daquela emitida. Estas alterações na frequência do sinal, estão relacionadas às componentes de velocidade do sangue.

O efeito Doppler foi descrito em 1842 pelo austríaco Johann Christian Doppler no artigo “*On the Colored of Double Stars and Some other Haenvely Bodies*” (Plasencia, 1994). Este artigo descreve que a frequência de formas de onda percebidos por um observador muda conforme a variação da distância entre a fonte emissora de radiação e o observador. Tal efeito, observado originalmente em corpos que emitem radiação eletromagnética, também é válido para ondas sonoras e de ultra-som.

Em outras palavras, se a distância entre a fonte e o observador aumenta em função do tempo, a frequência do sinal recebido será percebida como inferior a frequência do sinal transmitido (Figura 2.1.a). Quando a distância não for alterada (Figura 2.1.b), a frequência do sinal recebido será igual à frequência do sinal emitido. Caso haja aproximação (Figura 2.1.c), a frequência percebida será maior. Este efeito é funcional dentro de certos limites da velocidade de deslocamento dos corpos e da natureza da onda em questão. Em se tratando de ultra-som, o limite é imposto pela velocidade de deslocamento das ondas de som (ver Tabela 2.1 e Equação 2.2).

No sistema cardiovascular, as velocidades envolvidas estão dentro destes limites.

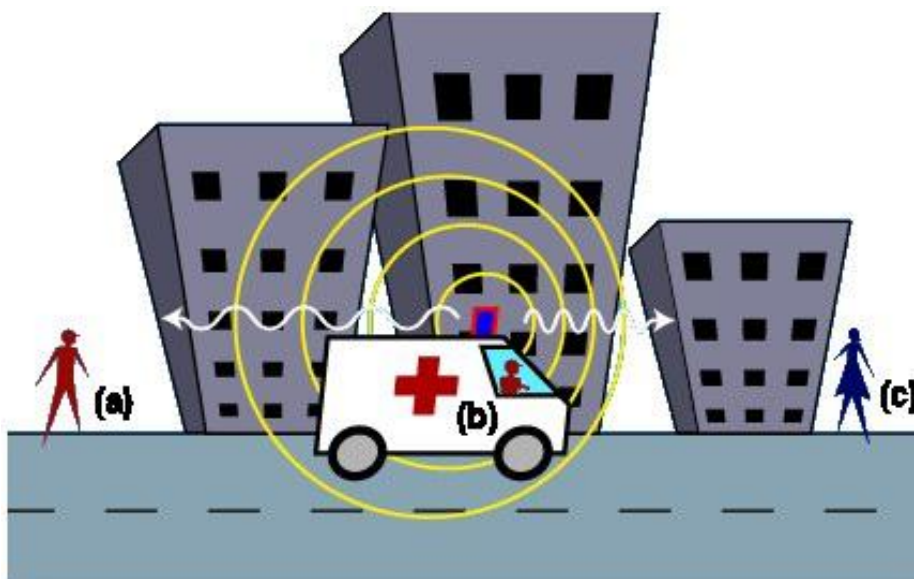


Figura 2.1 – Efeito Doppler.

Alteração no comprimento de onda do sinal recebido de uma sirene em movimento. (a) Comprimento de onda aumenta quando a sirene afasta-se do observador. (b) Não há alteração no comprimento de onda quando a sirene está estacionária em relação ao observador e (c) diminui quando se aproxima do observador. Em (a) e (c), a frequência do sinal recebido é diferente da frequência do sinal emitido.

O efeito Doppler também se manifesta quando a fonte emissora e o observador permanecem estacionários, porém um corpo em movimento reflete o sinal emitido para o observador (Figura 2.2). Este é o caso da ultra-sonografia Doppler, onde o cristal emissor e o receptor são estacionários e com distância considerada fixa em relação ao meio, o tecido. Sendo através da reflexão do sinal, causada pelos eritrócitos em suspensão no sangue em movimento, que se obtém as informações de intensidade e sentido do FS (Moraes, 1995).

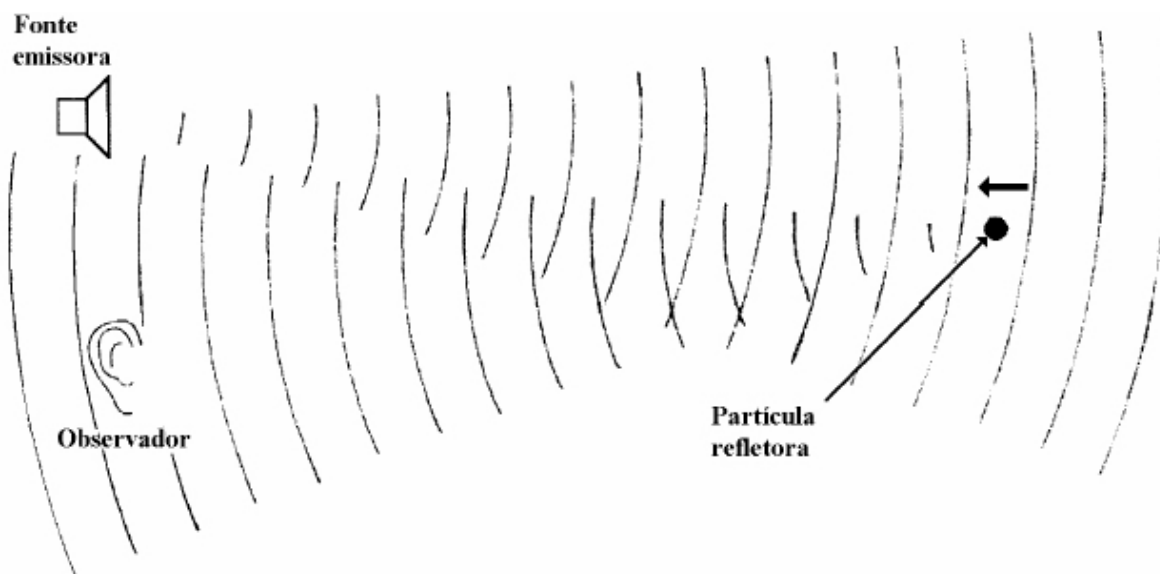


Figura 2.2 – Reflexão da onda sonora.

Efeito Doppler causado por partícula em movimento. O sinal refletido pela partícula em aproximação tem frequência superior a do sinal emitido.

A frequência Doppler está, portanto, relacionada com o vetor de velocidade da partícula refletora. Tal frequência é expressa pela Equação 2.2 para um sistema transdutor com a conformação dada na Figura 2.3.

$$f_d = \frac{2 \cdot V \cdot f_0 \cdot \cos(\theta) \cdot \cos\left(\frac{\phi}{2}\right)}{c} \quad 2.2$$

Onde:

f_d = Frequência Doppler;

V = Velocidade da partícula refletora;

f_0 = Frequência transmitida;

θ = Ângulo de incidência entre o feixe de ultra-som e o vetor velocidade da partícula;

ϕ = Ângulo entre o feixe do transmissor e do receptor.

c = velocidade da onda de ultra-som no meio biológico (aproximadamente 1540 m/s);

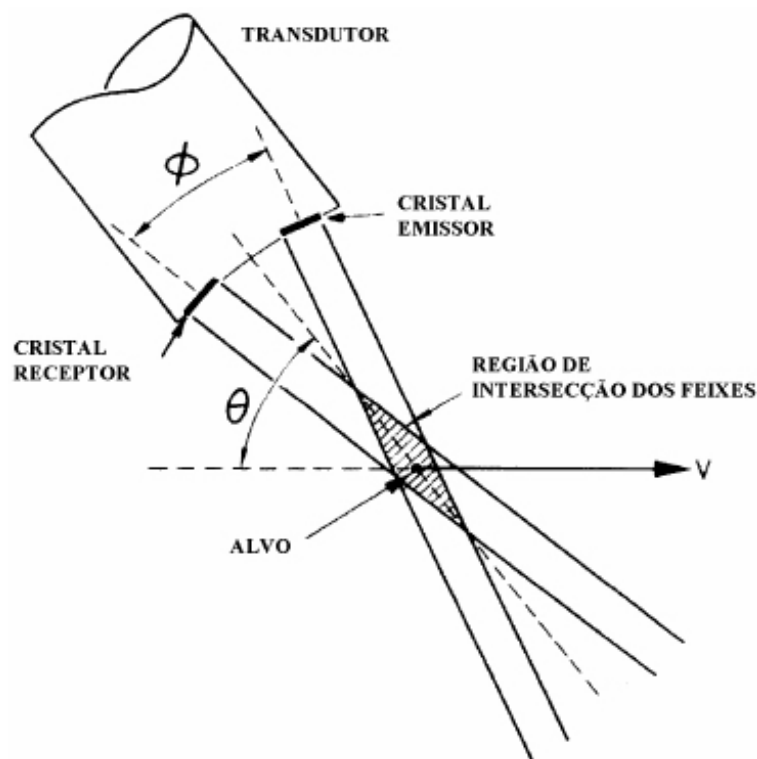


Figura 2.3 – Transdutor de ultra-som.

Esquema do transdutor de ultra-som Doppler de ondas contínuas com ângulos de insonação.

2.1.2. Sistema de Ultra-som Doppler de Ondas Contínuas (SUDOC)

Em 1956, Satomura demonstrou pela primeira vez um sistema ultra-sônico de ondas contínuas capaz de detectar estruturas móveis utilizando o efeito Doppler. A primeira utilização voltada para medicina deu-se através do trabalho de Franklin *et al.* (1961). A técnica foi utilizada para detectar o fluxo de sangue. Contudo, o primeiro instrumento prático para monitorar o fluxo sanguíneo foi o de Baker *et al.* (1964) (Baker, 1970; Rickey, 1995)

A faixa de frequência geralmente utilizada nos SUDOCs corresponde a valores entre 1MHz e 10MHz, conforme a profundidade que se deseja alcançar. De acordo com a Equação 2.1, quanto mais alta a frequência do sinal de ultra-som, maior será a atenuação e portanto, menor o poder de penetração.

Os SUDOCs podem ser direcionais ou não direcionais, sendo que o sistema direcional permite a avaliação do sentido de direção do fluxo sanguíneo. A diferença reside na característica do sistema direcional em identificar os desvios de frequência acima ou abaixo da frequência de transmissão. Um desvio acima da frequência de referência indica aproximação. Desvio abaixo desta frequência significa afastamento do corpo refletor.

Neste trabalho, utilizou-se o sistema direcional para mensurar o fluxo sanguíneo.

2.2. Processamento Digital de Sinais Doppler

As curvas de velocidade do FS são a base para o cálculo de parâmetros quantitativos utilizados no auxílio do diagnóstico na prática clínica, sendo obtidas a partir do sonograma.

O sinal Doppler analógico é digitalizado para então se utilizar técnicas de processamento digital de sinais. Para tal, é necessário descrever a conformação do sinal em quadratura de fase, assim como a fundamentação matemática para o desenvolvimento dos algoritmos que permitem obtenção de sonograma e envoltórias necessárias para cálculo dos parâmetros de diagnóstico.

A Figura 2.4 ilustra os principais estágios para conseguir exibir o sonograma e suas envoltórias na tela do PC. Para tal, é necessário passar por uma etapa de processamento digital dos sinais, responsável pela obtenção do sonograma a partir dos sinais Doppler em quadratura de fase.

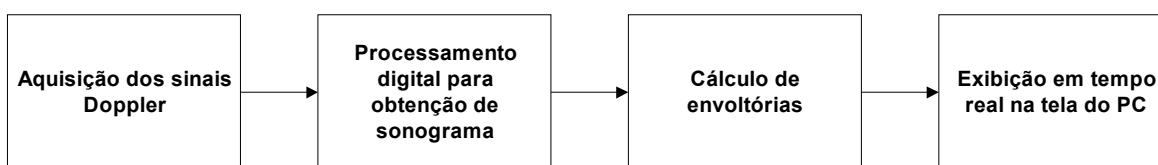


Figura 2.4 – Obtenção de sonograma.

Principais etapas para exibição de sonograma e envoltórias a partir do sinal Doppler.

2.2.1. Sinal em Quadratura de Fase

O método mais comumente empregado para identificar a direção do fluxo sanguíneo é o de quadratura de fase.

Para processar sinais em quadratura de fase é necessário descrevê-los matematicamente. Começando por sua geração na Placa Doppler até chegar à conformação final que é enviada ao PC para obtenção de sonograma.

Seja V_{out} o sinal gerado (Equação 2.3). O sinal resultante da interação com os eritrócitos retorna ao transdutor contendo informações de intensidade e direção do fluxo sanguíneo (Equação 2.4).

$$V_{out}(t) = A \cdot \cos(w_0 \cdot t + \phi_0) \quad 2.3$$

$$V_{in}(t) = \cos(w_0 \cdot t + \phi_0) + A_{ap} \cdot \cos((w_0 + w_{ap}) \cdot t + \phi_{ap}) + A_{af} \cdot \cos((w_0 - w_{af}) \cdot t + \phi_{af}) \quad 2.4$$

Onde:

$V_{in}(t)$ = sinal aplicado na entrada nos demoduladores;

A = Amplitude do sinal, conforme índice;

w_0 = Frequência emitida pelo transdutor de ultra-som;

w = frequência, conforme índice;

Índice “ap” = relativo ao objeto aproximando-se do transdutor;

Índice “af” = relativo ao objeto afastando-se do transdutor;

Assim, o sinal emitido retorna contendo a frequência fundamental da emissão somada com as frequências de deslocamento Doppler, sendo suprido aos circuitos de demodulação (Figura 2.5 e Figura 2.6).

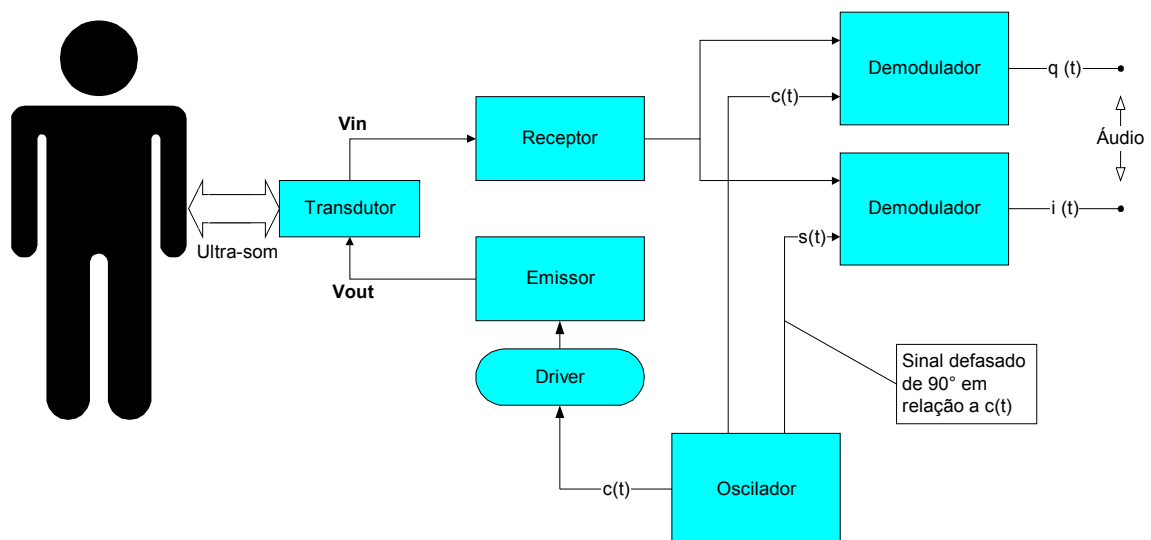


Figura 2.5 – Placa Doppler.

Diagrama de blocos básico de uma Placa Doppler. O sinal gerado pelo oscilador é amplificado e emitido pelo transdutor, retornando a frequência Doppler que é então demodulada em quadratura de fase.

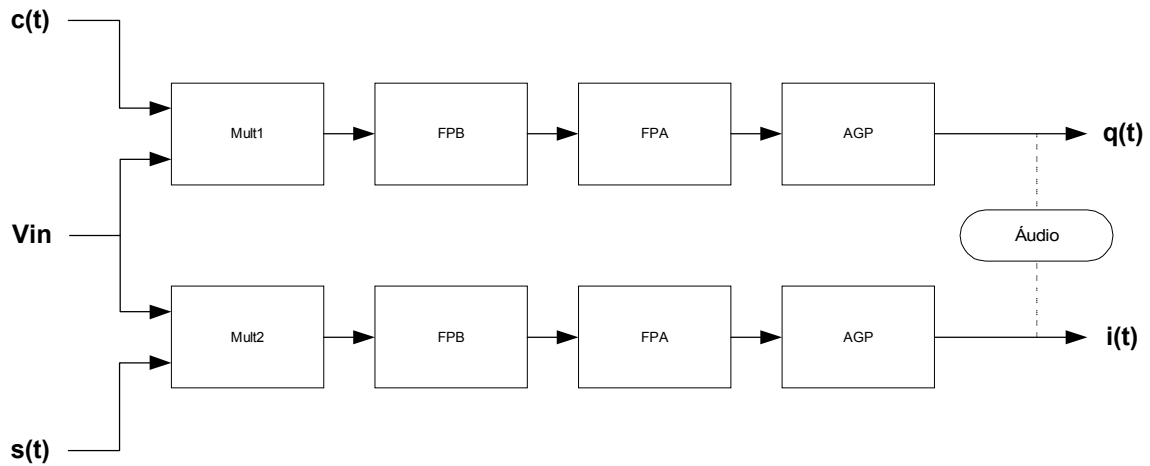


Figura 2.6 – Demodulador em quadratura de fase.

Etapa de demodulação em quadratura de fase. Apresentando inicialmente bloco de multiplicação, filtros para atenuar portadora e espectros indesejáveis. O Amplificador de Ganho Programável permite obtenção de nível de tensão adequado para o próximo estágio de aquisição dos sinais.

O bloco *mult1* (Figura 2.6) multiplica o sinal de entrada $V_{in}(t)$ pelo sinal $c(t)$ que está em fase com o sinal emitido. Conforme Equação a seguir.

$$i(t) = \text{sen}(w_0 \cdot t) \cdot [A_0 \cdot \cos(w_0 \cdot t + \phi_0) + A_{ap} \cdot \cos((w_0 + w_{ap}) \cdot t + \phi_{ap}) + A_{af} \cdot \cos((w_0 - w_{af}) \cdot t + \phi_{af})] \quad 2.5$$

Efetuando a multiplicação:

$$i(t) = A_0 \cdot [\text{sen}(w_0 \cdot t) \cdot \cos(w_0 \cdot t + \phi_0)] + A_{ap} \cdot [\text{sen}(w_0 \cdot t) \cdot \cos((w_0 + w_{ap}) \cdot t + \phi_{ap})] + A_{af} \cdot [\text{sen}(w_0 \cdot t) \cdot \cos((w_0 - w_{af}) \cdot t + \phi_{af})] \quad 2.6$$

Aplicando a fórmula trigonométrica abaixo:

$$\text{sen}(A) \cdot \cos(B) = \frac{1}{2} \cdot [\text{sen}(A - B) + \text{sen}(A + B)] \quad 2.7$$

Obtém-se:

$$\begin{aligned}
 i(t) = & \frac{A_0}{2} \cdot [\text{sen}(-\phi_0) + \text{sen}(2 \cdot \omega_0 \cdot t + \phi_0)] + & 2.8 \\
 & \frac{A_{ap}}{2} \cdot [\text{sen}(\omega_{ap} \cdot t - \phi_{ap}) + \text{sen}((2 \cdot \omega_0 - \omega_{ap}) \cdot t + \phi_{ap})] + \\
 & \frac{A_{af}}{2} \cdot [\text{sen}(-\omega_{af} \cdot t - \phi_{af}) + \text{sen}((2 \cdot \omega_0 + \omega_{af}) \cdot t + \phi_{af})]
 \end{aligned}$$

O sinal acima passa por filtros para atenuar as componentes contínuas (dc) e frequências fora do espectro de interesse (Figura 2.6), resultando no sinal a seguir.

$$i(t) = \frac{A_{ap}}{2} \cdot \text{sen}(\omega_{ap} \cdot t - \phi_{ap}) + \frac{A_{af}}{2} \cdot \text{sen}(-\omega_{af} \cdot t - \phi_{af}) \quad 2.9$$

Considerando a seguinte relação trigonométrica:

$$\text{sen}(-A) = -\text{sen}(A) \quad 2.10$$

Obtém-se o sinal $i(t)$ presente na saída do sistema de demodulação (Figura 2.6).

$$i(t) = \frac{A_{ap}}{2} \cdot \text{sen}(\omega_{ap} \cdot t - \phi_{ap}) - \frac{A_{af}}{2} \cdot \text{sen}(\omega_{af} \cdot t + \phi_{af}) \quad 2.11$$

O processo é semelhante para o sinal $q(t)$, porém o sinal $V_{in}(t)$ é multiplicado com um sinal de frequência igual à do sinal emitido deslocado em fase de 90° , ou seja, em quadratura de fase. Equação 2.12.

$$\begin{aligned}
 q(t) = & A_0 \cdot [\cos(\omega_0 \cdot t) \cdot \cos(\omega_0 \cdot t + \phi_0)] + & 2.12 \\
 & A_{ap} \cdot [\cos(\omega_0 \cdot t) \cdot \cos((\omega_0 + \omega_{ap}) \cdot t + \phi_{ap})] + \\
 & A_{af} \cdot [\cos(\omega_0 \cdot t) \cdot \cos((\omega_0 - \omega_{af}) \cdot t + \phi_{af})]
 \end{aligned}$$

Aplicando a relação trigonométrica:

$$\cos(A) \cdot \cos(B) = \frac{1}{2} \cdot [\cos(A - B) + \cos(A + B)] \quad 2.13$$

Tem-se:

$$q(t) = \frac{A_0}{2} \cdot [\cos(\phi_0) + \cos(2 \cdot \omega_0 \cdot t + \phi_0)] + \frac{A_{ap}}{2} \cdot [\cos(-\omega_{ap} \cdot t - \phi_{ap}) + \cos((2 \cdot \omega_0 + \omega_{ap}) \cdot t + \phi_{ap})] + \frac{A_{af}}{2} \cdot [\cos(\omega_{af} \cdot t - \phi_{af}) + \cos((2 \cdot \omega_0 - \omega_{af}) \cdot t + \phi_{af})] \quad 2.14$$

Aplicando a filtragem, conforme Figura 2.6, tem-se:

$$q(t) = \frac{A_{ap}}{2} \cdot \cos(-\omega_{ap} \cdot t - \phi_{ap}) + \frac{A_{af}}{2} \cdot \cos(\omega_{af} \cdot t - \phi_{af}) \quad 2.15$$

Usando a relação que segue:

$$\cos(-A) = \cos(A) \quad 2.16$$

Obtém-se o sinal $q(t)$:

$$q(t) = \frac{A_{ap}}{2} \cdot \cos(\omega_{ap} \cdot t + \phi_{ap}) + \frac{A_{af}}{2} \cdot \cos(\omega_{af} \cdot t - \phi_{af}) \quad 2.17$$

As equações 2.11 e 2.17 mostram o sinal na saída do demodulador, já com a portadora filtrada.

Quando há somente fluxo direto o sinal $q(t)$ tem como componente principal um sinal co-seno com amplitude A_{ap} e defasado de 90° . Quando ocorre apenas fluxo reverso a componente $q(t)$ apresenta um co-seno de amplitude A_{af} e adiantado de 90° em relação ao sinal $i(t)$. Desta forma, mantém-se as informações relativas à amplitude e direção do fluxo.

Nos próximos itens, são discutidas as técnicas de processamento de sinais que permitem recuperar as informações de direção e intensidade do fluxo dos partir da amostragem dos sinais $i(t)$ e $q(t)$.

2.2.2. Sonograma

Sonograma é a representação gráfica da distribuição de velocidade dos eritrócitos no fluxo sanguíneo. Permite a visualização da intensidade e sentido das componentes de velocidade do sangue ao longo do tempo.

O eixo das ordenadas é composto pelas componentes de frequência do sinal de deslocamento Doppler e fornece uma estimativa da velocidade do sangue, sendo a intensidade do fluxo representado por código de cor. O tempo é indicado no eixo das abscissas.

A linha que divide o sonograma horizontalmente delimita o sentido do fluxo. Sinal abaixo desta linha indica fluxo se aproximando (direto) acima fluxo se afastando (reverso), sempre em relação ao transdutor de ultra-som.

No exemplo de sonograma da Figura 2.7, a intensidade está discriminada pela intensidade de tons de cinza. O tom mais escuro representa maior intensidade de fluxo do sangue.

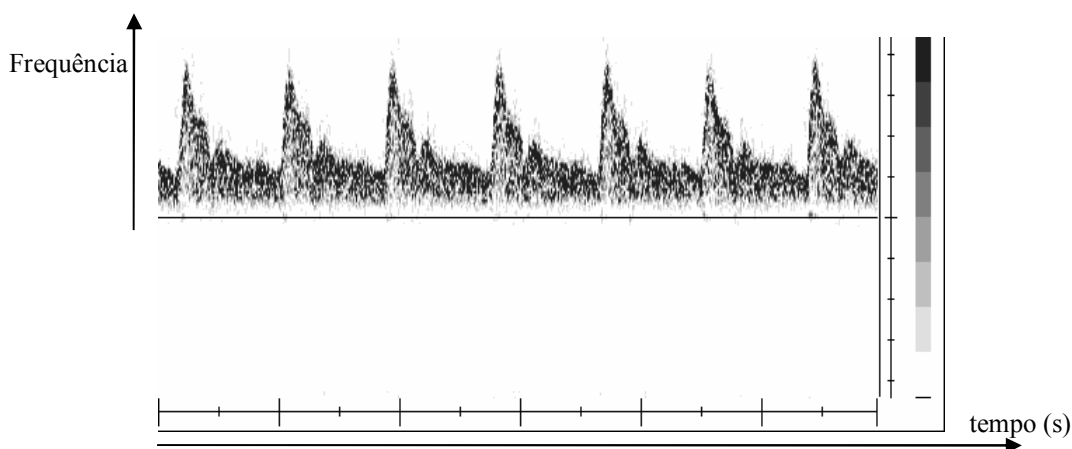


Figura 2.7 – Sonograma bem delineado.

Sonograma normal perdurando por sete ciclos cardíacos, apresenta-se bem delineado. Onde, o eixo das abscissas indica frequência e das ordenadas o tempo.

Para um fluxo normal, o sonograma é bem delineado (Figura 2.7). Quando o fluxo entra em turbulência devido a estenoses, ou outros fatores, o espectro de frequências do sonograma apresenta caráter mais aleatório (Figura 2.8) (Hill, 1986). Esta característica permite identificar anomalias no fluxo sanguíneo.

O sinal Doppler é a base para a obtenção do sonograma. Após demodular os sinais expressos pelas equações 2.11 e 2.17, estes são digitalizados e processados pelo PC. Devem ser observados certos procedimentos para tal operação (Moraes *et al*, 1996; Aydin, 1994; Wright *et al*, 1999). Nas próximas seções deste mesmo capítulo são apresentados os fundamentos do processamento de sinais para obtenção do sonograma.

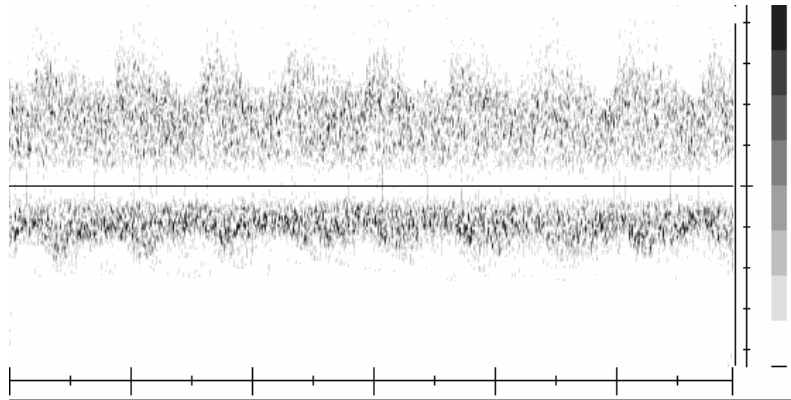


Figura 2.8 – Sonograma com espectro disperso.

Sonograma de artéria com estenose, mal delimitado (seis ciclos cardíacos).

2.2.3. Transformada de Fourier

O matemático e físico francês Jean Baptiste Joseph Fourier (1768-1830) apresentou em 1807 um artigo ao “*Institut de France*” onde descreveu um método para representar a distribuição de temperatura através de senóides. O artigo continha uma afirmação controversa para a época, na qual qualquer sinal contínuo e periódico poderia ser representado pela soma apropriada de ondas senoidais. Entre os revisores estavam J.L. Lagrange (1736-1813) e P.S. Laplace (1749-1827). Enquanto Laplace e os demais votaram pela publicação do artigo, Lagrange protestou firmemente. O “*Institut de France*” cedeu ao prestígio de Lagrange e rejeitou o trabalho de Fourier, somente quinze anos depois, em 1822, o trabalho de Fourier foi publicado (Smith, 1999).

Através da Transformada de Fourier (FT) é possível decompor um sinal contínuo no tempo $x(t)$ em seus componentes de frequência (harmônicos), representando o sinal por uma série de exponenciais complexas, ou de senos e co-senos. Assim, cada componente denota a importância, ou peso, daquela frequência na aproximação da composição do sinal original. Desta forma, a Transformada de Fourier possibilita descrever no domínio da frequência $X(f)$, um sinal expresso no domínio do tempo $x(t)$.

A equação 2.18 define a Transformada de Fourier.

$$X(f) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j \cdot 2 \cdot \pi \cdot f \cdot t} dt \quad 2.18$$

Para obtenção do sonograma, é preciso apresentar o sinal Doppler no domínio da frequência, diferenciando o fluxo direto do fluxo reverso. Existem diversos métodos para realizar este procedimento, como técnicas envolvendo método auto-regressivo ou *wavelets* (Keeton, 1997 e 1998; Güler, 2001; Kuang, 2002).

O FS é dinâmico, sendo a velocidade dos eritrócitos constantemente alterada e portanto é caracterizado como não periódico. Uma condição comum em sinais de origem biológica (Pereira, 2002).

A utilização da Transformada de Fourier em sinais não estacionários é considerada crítica. Entretanto, seu uso para obtenção de sonograma é viabilizado devido à característica estacionária que o FS assume em intervalos de tempo inferiores a 10ms (Nizametin, 1994; Guo *et al*, 1993).

Neste trabalho, optou-se pela utilização da Transformada de Fourier por ter o processamento suficientemente rápido e atender as necessidades requeridas pela aplicação, separando o fluxo direto do reverso.

Para construção do sonograma, utilizou-se a CFT (Transformada Complexa de Fourier), conforme apresentado na equação 2.19.

$$X(f) = R(f) + jI(f) = |X(f)| \cdot e^{j\theta(f)} \quad 2.19$$

Onde:

$R(f)$ = parte real da FT;

$I(f)$ = parte imaginária da FT.

Aplicando a CFT nos sinais $i(t)$ e $q(t)$, expressos nas equações 2.11 e 2.17, é possível obter o espectro de frequências e também, as informações de direção do fluxo sanguíneo.

A Figura 2.9 e a Figura 2.10 mostram a recuperação da informação de sentido do fluxo através da CFT, considerando os sinais $i(t)$ e $q(t)$ como a parte real e imaginária, respectivamente.

Numa situação hipotética, quando há aproximação de somente um eritrócito em relação ao transdutor, a componente A_{af} das equações 2.11 e 2.17 é zero. Assim, $i(t)$ é composto apenas por seno, a ser aplicado na parte real da CFT, enquanto $q(t)$ é composto por co-seno que constitui a parte imaginária da CFT, ambos sinais têm a mesma amplitude. O resultado é um sinal na parte positiva do espectro de frequências (Figura 2.9), indicando

o fluxo direto.

Contudo, se houver apenas um eritrócito afastando-se (Figura 2.10), a parte real da CFT recebe um seno negativo e a parte imaginária, um co-seno. Resultando em um único sinal no lado das frequências negativas, indicando fluxo reverso.

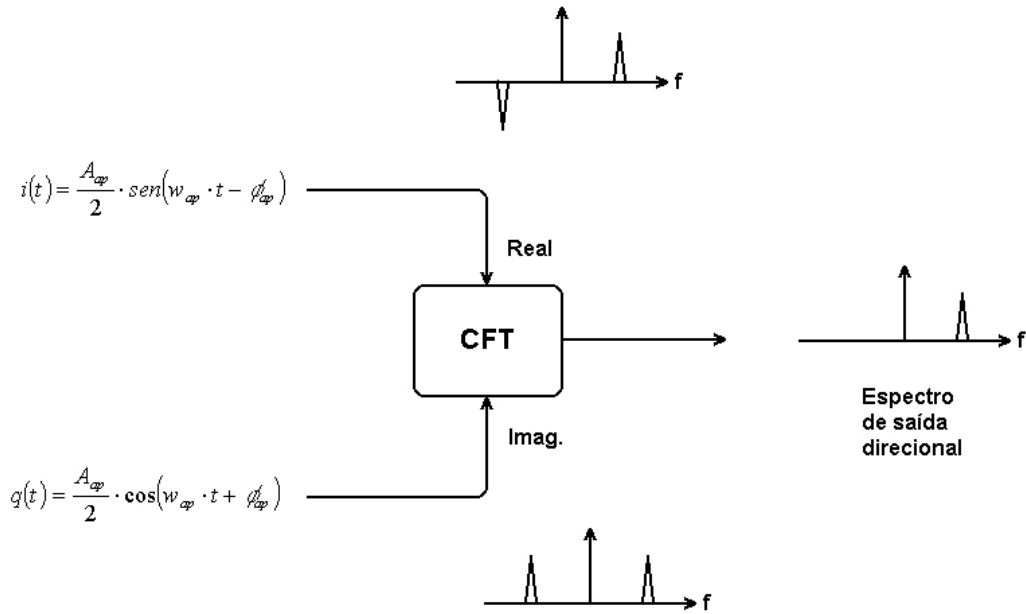


Figura 2.9 – Discriminação de fluxo direto através da CFT.

Espectro de saída direcional obtido devido à aproximação hipotética de uma partícula refletora.

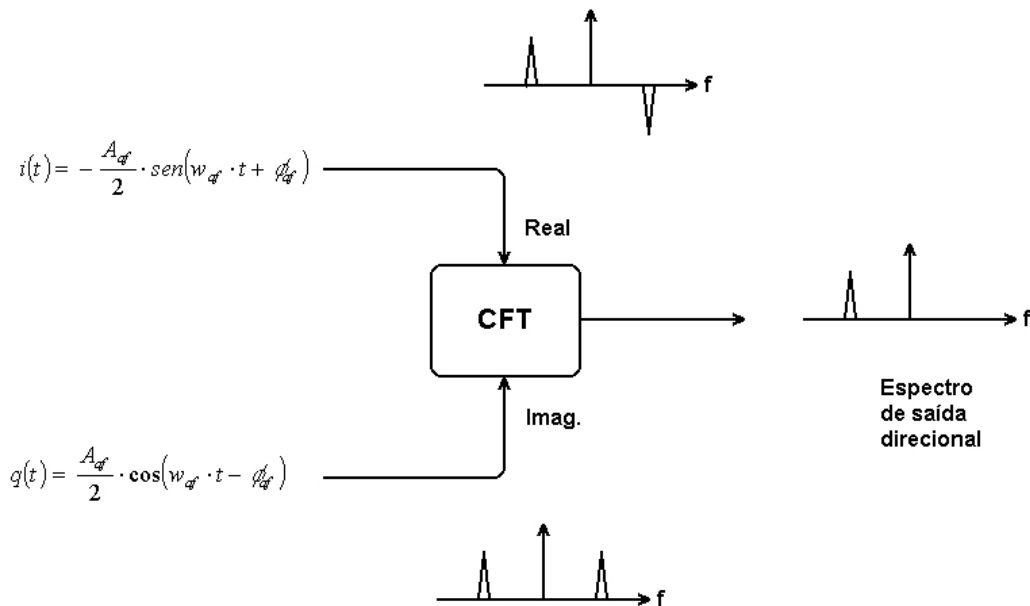


Figura 2.10 – Discriminação de fluxo reverso através da CFT.

Espectro de saída direcional obtido devido ao afastamento de uma partícula refletora.

Desta forma, o resultado da CFT é o espectro de saída direcional que irá compor uma coluna de sonograma, onde as frequências positivas indicam fluxo direto e as negativas, o fluxo reverso.

Uma vez que os sinais Doppler tenham sido digitalizados e o processamento digital dos sinais realizado, torna-se necessário utilizar um caso especial da FT para a implementação do software. A DFT (Transformada Discreta de Fourier) trabalha com sinais digitalizados, ou “discretizados”. Considerando um sinal $x(n)$, composto por N amostras, a DFT deste também terá N elementos. A Transformada Discreta de Fourier está expressa na Equação 2.20. A Figura 2.11 traz a representação nas N amostras complexas do sinal no domínio do tempo que são passadas para o domínio da frequência, também com N amostras complexas.

$$X(w_k) = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x(n) \cdot e^{-j \cdot w_k \cdot \frac{n}{N}} \quad 2.20$$

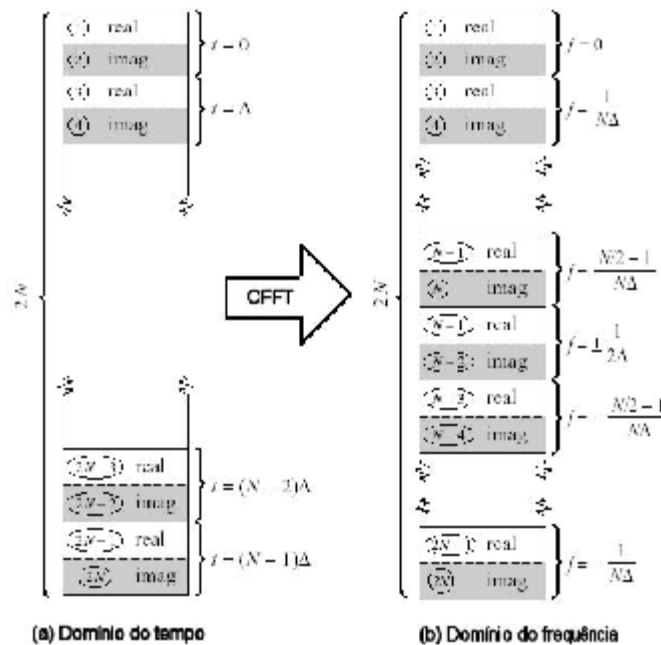


Figura 2.11 – DFT complexa.

Sinal real com N amostras, no domínio do tempo, é representado por parte real e imaginária no domínio da frequência.

Para melhorar o desempenho computacional é utilizada a Transformada Rápida de Fourier (FFT), que consiste em um algoritmo que calcula a DFT com um número reduzido de somas e multiplicações. O número de operações é reduzido devido à reutilização dos

resultados de cálculos anteriores. O método explora a periodicidade e a simetria das funções trigonométricas, necessitando aproximadamente $N \cdot \log_2 N$ operações (onde N é o número de amostras).

Diversos algoritmos otimizados estão disponíveis para implementação da CFFT (Transformada Rápida de Fourier Complexa) já incorporando os janelamentos mais utilizados em sinais bioelétricos (Pereira, 2002). No Capítulo referente à metodologia são comentadas as bibliotecas necessárias ao projeto.

2.2.4. Janelamento

Os sinais Doppler amostrados não podem ser analisados diretamente, é preciso definir um intervalo finito para análise com a CFFT. Para tal utiliza-se o método de janelamento.

O janelamento possibilita trabalhar apenas com uma parte do sinal, na Figura 2.12 é aplicada uma janela retangular para delimitação do sinal a ser analisado, zerando os sinais fora do período de interesse.

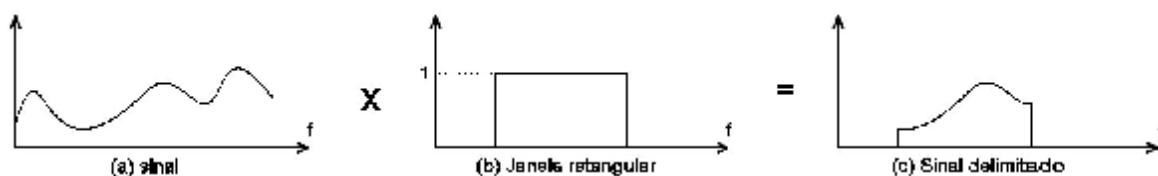


Figura 2.12 – Janela retangular delimitando sinal.

Contudo, o método de janelamento não tem como única função pegar apenas uma janela, ou trecho do sinal. Conforme já mencionado, o sinal Doppler é essencialmente não periódico, uma janela adequada para o tipo de sinal analisado permite melhorar a detecção das frequências que compõe o sinal original.

Duas características muito importantes a serem definidas são o formato e o tamanho da janela a ser utilizada (Pereira, 2002).

A janela deve ter entre 5ms e 10ms para o processamento dos sinais Doppler e construção do sonograma.

Trabalhos anteriores indicam a janela tipo Hanning (Equação 2.21) como sendo a mais adequada para a aplicação corrente (Nizametin, 1994). Na implementação do software optou-se por permitir a escolha de janelas tradicionais de forma dinâmica.

$$w(n) = \begin{cases} 0,5 - 0,5 \cdot \cos\left(\frac{\pi \cdot n}{M}\right) & \text{para } 0 \leq n \leq M \\ 0 & \text{caso contrário} \end{cases} \quad 2.21$$

2.2.5. Envoltórias

A base para extração dos parâmetros quantitativos são as envoltórias de máxima frequência e a média ponderada pela intensidade de cada componente de frequência (Figura 2.13). Também é obtida a envoltória da frequência de maior intensidade, ou seja, a frequência de maior peso em cada análise espectral (Moraes, 1995; Tortoli, 1995; Seitz, 2001).

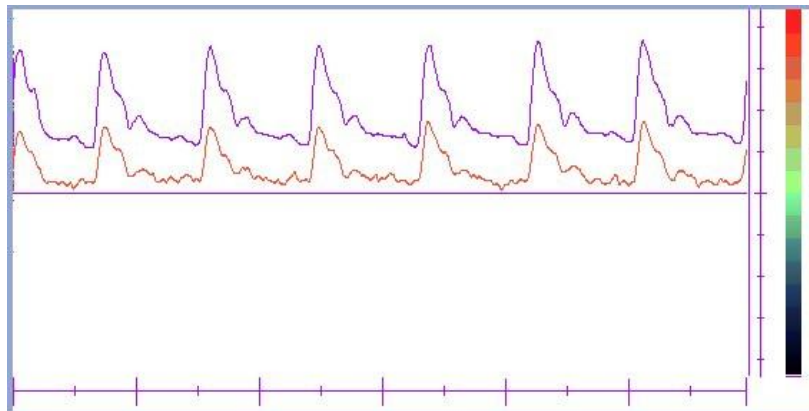


Figura 2.13 – Envoltórias do sonograma.

Envoltórias de frequência média e máxima do sonograma.

A envoltória de frequência média é calculada a cada análise espectral e está expressa na Equação 2.22. Esta envoltória tem uma correlação muito próxima à velocidade média do sangue, porém é bastante influenciada pelo padrão de insonação do transdutor.

$$f_m = \frac{\sum_{i=1}^N P_i \cdot i}{\sum_{i=1}^N P_i} \quad 2.22$$

A natureza não determinística do sinal e a presença de ruído tornam inviável utilizar a máxima frequência presente em cada análise espectral para construir a envoltória de máxima frequência. Foram desenvolvidos diversos métodos para determinação da frequência máxima (Mo *et al*, 1998; Marasek *et al*, 1993; Aydin, 1994) e como esta envoltória é pouco influenciada pelo padrão de insonação¹ do transdutor, passou a ser utilizada clinicamente na extração de índices para avaliação quantitativa.

O método percentual para extração da frequência máxima é computacionalmente o mais simples. Neste método, para cada análise espectral é feito um somatório de todas as intensidades das componentes de frequência. O valor total deste somatório é representado por Φ_T .

Parte do valor de Φ_T deve-se ao ruído. É considerada como frequência máxima a componente cuja intensidade somada às intensidades das componentes inferiores resulte no que falta para completar Φ_T , conforme equação 2.23.

$$\phi(f_{\max}) = \frac{(100 - \alpha)}{100} \cdot \phi_T \quad 2.23$$

Onde:

Φ_T = Somatório de todas as intensidades da análise espectral do sinal Doppler;

f_{\max} = frequência máxima, desconsiderando o ruído;

$\Phi(f_{\max})$ = somatório de todas as intensidades até f_{\max} ;

α = Porcentagem do somatório de todas as intensidades do espectro Doppler devido ao ruído. Depende da relação sinal-ruído, sendo um valor entre 2% e 5% considerado uma boa estimativa (No *et al*, 1988; Aydin, 1994).

Outra maneira para determinar a frequência máxima é através do método geométrico. A Figura 2.14 ilustra o procedimento, num primeiro momento é calculada a integral das N amostras do espectro do sinal Doppler, representado pela curva “ cI ”. Em seguida é traçada uma linha reta “ rI ” entre o ponto de origem “ O ” e o final do somatório, indicado pelo ponto “ T ”.

¹ Ângulo de Insonação refere-se ao ângulo do feixe das ondas de ultra-som e o vetor direção do fluxo sanguíneo.

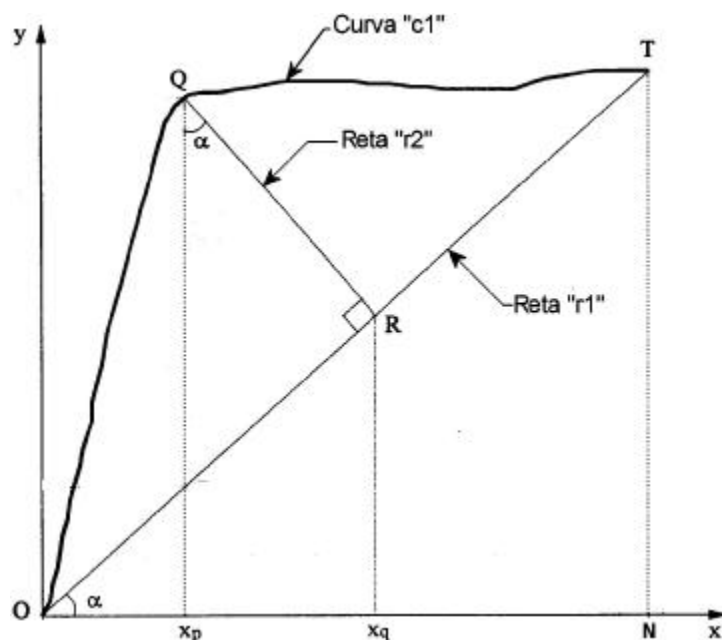


Figura 2.14 – Método geométrico.

Método geométrico para obtenção da envoltória de frequência máxima.

Considerando a reta r_2 perpendicular à r_1 , sendo o ponto R a interseção da reta r_2 com a reta r_1 e o ponto Q a interseção da reta r_2 com a curva c_1 . Encontrando a maior distância entre o ponto Q e o ponto R que satisfaça as condições acima, encontra-se o ponto de frequência máxima, indicado por X_p .

O método geométrico possui maior complexidade de cálculo, tornando-o mais demorado computacionalmente se comparado com o método percentual. Contudo o tempo para cálculo do espectro de frequências é muito superior, fazendo com que o tempo utilizado para determinar a frequência máxima seja considerado desprezível.

As envoltórias do sonograma são necessárias para calcular índices que possibilitam uma análise quantitativa do FS. Dentre os parâmetros de maior destaque é possível citar: índice de pulsatilidade; atraso eletromecânico; tempo de meia subida da fase do fluxo sistólico; quociente de amortecimento proximal; tempo de subida da fase de fluxo sistólico e largura da fase de fluxo sistólico (Moraes, 1990).

CAPÍTULO 3

3. Metodologia

Conforme mencionado, no capítulo introdutório, esta dissertação trata da implementação de software para processamento de sinais Doppler em tempo real, utilizando para este fim recursos nativos de um PC padrão. Neste capítulo, estão descritos o hardware utilizado e o software implementado para alcançar o objetivo proposto.

Ao longo deste capítulo, trechos de código são apresentados almejando servir como referência para trabalhos futuros.

Optou-se portanto, por uma redação que apresente o código, deixando para o leitor mais interessado o seu estudo com a utilização dos itens de ajuda do C++ Builder™, do MSDN (*Microsoft Development Network*) e de livros que abordam o assunto, cita-se: Leão, (1998); Lamothe, (1999); Mateus, (2000); Adilson, (2000); Hollingworth, (2000) e Schildt, (2001).

Objetiva-se assim, proporcionar uma visão global das tarefas de programação envolvidas neste trabalho, bem como suprir informações e referências para aqueles que desejam se aprofundar no tema.

O sistema é composto por uma Placa Doppler, (responsável pela geração, transmissão, recepção e demodulação dos sinais Doppler de Ondas Contínuas) (Molina, 2002) e por software rodando em um microcomputador PC com S.O. (Sistema Operacional) MS-Windows™. O sinal Doppler deve ser digitalizado para processamento e obtenção do sonograma e suas envoltórias.

A Figura 3.1 exhibe as etapas, de hardware e de software, que constituem o SUDOC. O software, rodando em PC, tem por função controlar a amostragem dos sinais oriundos da Placa Doppler via placa de som, realizando o processamento necessário para arquivar os sinais Doppler e exibir sonograma e envoltórias na tela do PC em tempo real. A etapa de hardware, utilizada para gerar os sinais Doppler a partir do FS, é composta pela Placa Doppler. As etapas mencionadas são discutidas nos itens que seguem.

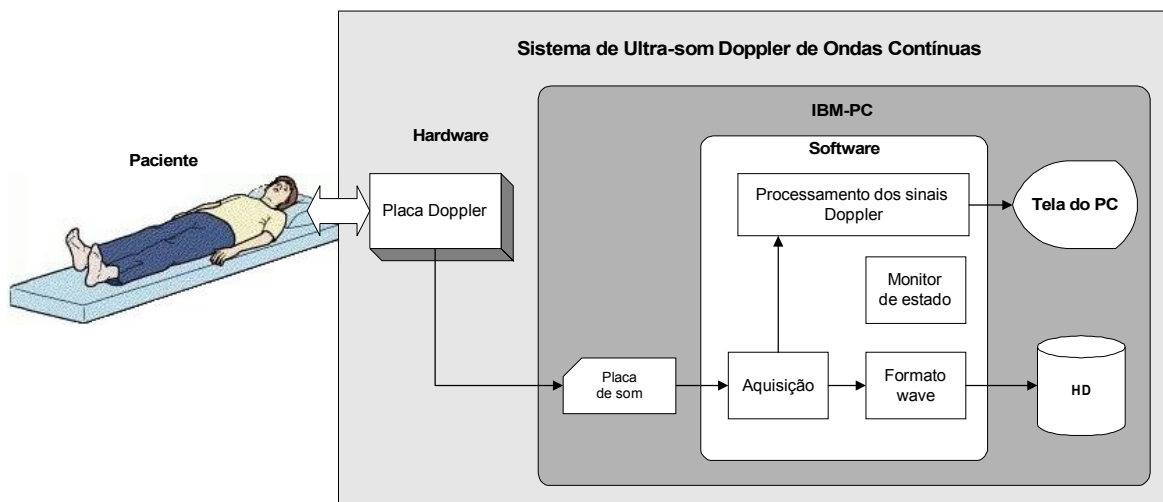


Figura 3.1 – Etapas constituintes do SUDOC.

3.1. Hardware Utilizado

Embora o software implementado permita o processamento de sinais Doppler em quadratura de fase oriundos de qualquer hardware (desde que respeitadas faixas de tensão e frequência de trabalho), a Placa Doppler desenvolvida por Molina *et al* (2002) foi utilizada para constituir e validar o sistema.

Os sinais Doppler são analógicos e devem ser digitalizados. Para tal, optou-se pela investigação da Placa de Som do PC como meio de aquisição. O Hardware utilizado e as características dos sinais Doppler são descritos a seguir.

3.1.1. Placa Doppler

Através da Placa Doppler, os sinais empregados para cálculo do sonograma (equações 2.11 e 2.17), são produzidos conforme descrito no item 2.2.1. A Placa Doppler gera sinais analógicos com componentes de frequência de até 10kHz. Esta frequência pode ser inferior, sendo dependente das componentes de velocidade do sangue e frequência do sinal de ultra-som incidente no vaso sanguíneo.

A amplitude do sinal oriundo da Placa Doppler deve ser condizente com as características da placa de aquisição. No sistema utilizado, os níveis de tensão do sinal chegam até 2 Volts de pico, na Placa Doppler utilizada há controle de amplitude do sinal entregue ao PC.

3.1.2. Aquisição dos Sinais Doppler

A aquisição do sinal, situado na faixa de áudio, deve atender a certas questões. Como o sinal está em quadratura de fase, é necessária aquisição pelos dois canais de forma casada, isto é, mantendo a relação entre amplitude e fase dos dois canais.

Assim, a aquisição deve ser simultânea e os sinais em quadratura de fase devem ser igualmente amplificados. Ao utilizar filtros, ambos devem apresentar a mesma atenuação para uma mesma frequência. As características do sinal em quadratura estão descritas no item 2.2.1. Os requisitos mínimos necessários ao sistema de aquisição estão expostos na Tabela 3.1.

Tabela 3.1 – Configuração mínima para PAS (Placa de Aquisição de Sinais).

Característica	Valores
Resolução	≥ 12 bits
Taxa de amostragem	≥ 22050 APS ¹
Número de canais	2
Erro de fase entre canais	$\leq 1^\circ$ (idealmente deve ter aquisição simultânea)
Tensão de entrada (fundo de escala)	1Vrms $\pm 25\%$
Impedância de entrada	$\geq 1k\Omega$

¹ APS – Amostras por Segundo

Como os sinais Doppler encontram-se dentro da faixa de frequência de áudio, foi investigada a possibilidade de utilizar Placa de Som, presente nos PCs, para aquisição dos sinais (Watkinson, 1995; Saldivar, 1997; Güler, 2002).

O ponto de partida foi a obtenção das características de hardware da placa de som do PC, buscando verificar se esta apresentava as características necessárias à aquisição de sinais Doppler em quadratura de fase (Ridge, 1995; Intel, 2003).

O padrão PC é guiado pelo *PC System Design Guide*, patrocinado pela Microsoft e Intel, empresas responsáveis por condições de mudança do mercado de PCs. Este guia proporciona padrões para a indústria de periféricos e de PCs. Portanto, contém informações sobre projetos atuais e futuros, provendo diretrizes para desenvolvimento de hardware que resulte em um melhor desempenho junto aos sistemas operacionais da Microsoft (Intel, 2003).

Placas de som padrão, estão em conformidade com o *PC 2001 System Design Guide* (Intel, 2003). Alguns dos requisitos mínimos do sistema de áudio para PCs estão descritos na Tabela 3.2.

Tabela 3.2 – *Audio Minimum Performance Requirements*.
Listados somente os itens relevantes para a aquisição dos sinais Doppler (Adaptado de *PC 2001 System Design Guide*).

Descrição	Requerimento	Valores
Tensão de entrada	FSIV ¹ (A-D-PC) <i>line input</i>	≥1.0 Vrms
(fundo de escala)	FSIV (A-D-PC) <i>microphone input</i>	≥100 mVrms
<i>Frequency response</i>		
Gravação digital (A-D-PC) para entrada <i>line input</i>	44.1 kHz <i>destination</i>	20 Hz to 17.6 kHz
	48.0 kHz <i>destination</i>	20 Hz to 19.2 kHz
	<i>Passband ripple</i>	<+/-0.5 dB
	<i>Dynamic range</i>	≥70 dB FS A2, 3
	THD+N ² (-3 dB FS)	≤-60 dB FS3 (<i>input-referenced</i>)
<i>Frequency response (-3 dB)</i>		
Gravação digital (A-D-PC) para entrada de microfone	22.05 kHz <i>destination</i>	100 Hz to 8.8 kHz
	<i>Passband ripple</i>	<+/-0.5 dB
	<i>Dynamic range</i>	≥70 dB FS A2, 3
	THD+N ² (-3 dB FS)	≤-60 dB FS (<i>input-referenced</i>)
Acurácia da frequência de amostragem	<i>Playback</i>	0.1%
	<i>Record</i>	0.1%

¹ FSIV – Full-scale input voltage – Tensão de entrada, fundo de escala.

² THD+N – Total harmonic distortion with noise – Distorção harmônica total mais ruído.

É comum a utilização de placas de som internas (barramento ISA ou PCI) e externas (FireWire ou USB). Além destas, existem as placas de som integradas à placa-mãe (*on-board*). Ambas seguem as definições do *PC Design Guide*.

Fabricantes de placas-mãe para PCs (Abit, Asus, Gigabyte, Intel, Iwill, MSI, Soltek e outras), habitualmente utilizam o Circuito Integrado (CI) AC'97 (Intel Inc.) ou CMI8738 (C-Media Electronics Inc.). Os mesmos CIs são utilizados por fabricantes de placa de som (Creative Labs.) (C-Media, 2003).

A Figura 3.2 traz em diagramas de blocos as principais funções de interesse dos CIs de áudio mencionados. Nesta ilustração, há uma entrada com dois canais (*Line in*) e outra de um canal (*MIC*). Há ainda, duas saídas de dois canais (*Line out e AUX*).

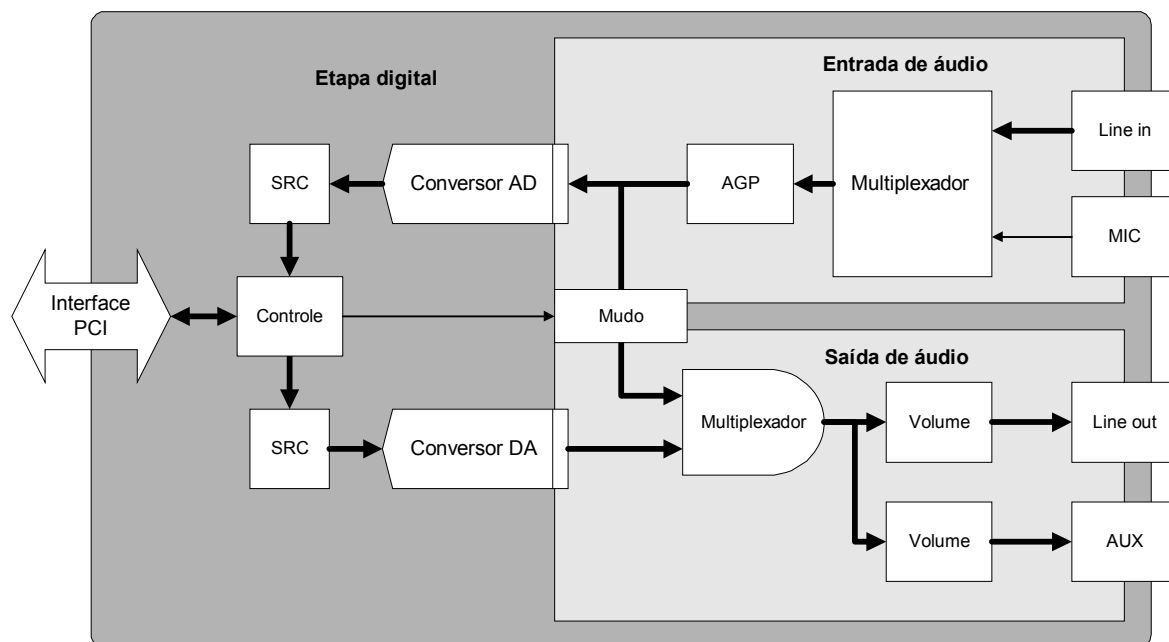


Figura 3.2 – Estágios dos CIs utilizados em placas de som.

Simplificação do diagrama de blocos dos componentes codificadores de áudio AC'97 e CMI8738.
Baseando em Intel, (2003) e Cmedia, (2003).

O sinal adquirido é aplicado a um multiplexador analógico, filtros e estágio de ganho (AGP). Posteriormente a este estágio analógico de tratamento do sinal, o mesmo é convertido para digital (Conversor AD) e aplicado ao conversor de taxa de amostragem (SRC) para só então ser enviado ao barramento PCI.

O sinal aplicado à entrada *Line in* pode ser realimentado para a saída (*Line out*) através do estágio *Mudo*, possibilitando ao usuário do sistema ouvir o sinal Doppler proveniente do FS (frequência de deslocamento Doppler). Valores de taxa de conversão, resolução, amostragem, controle do AGP e outros, são configuráveis por software conforme modelo da placa de som.

Ao analisar os manuais das placas de som e de seus principais componentes, é possível obter suas características fundamentais. Estas características estão em conformidade com as exigências descritas para a aquisição dos sinais em quadratura de fase. Tais manuais indicam que é possível realizar a aquisição dos sinais sem perder informações cruciais para construção de sonograma, respeitando fase, amplitude, frequência e resolução.

O esquema apresentado na Figura 3.3 foi implementado para comprovar a eficiência da placa de som na aquisição de sinais em quadratura de fase. Com esta montagem, é possível enviar sinais em quadratura de fase para o PC. Assim, é possível

verificar se os sinais processados mantêm a diferença de fase e amplitude. O osciloscópio digital permite monitorar o estado real do sinal entregue à placa do som.

Um software foi implementado especialmente para realizar testes de fase, frequência e amplitude para os sinais amostrados pela placa de som.

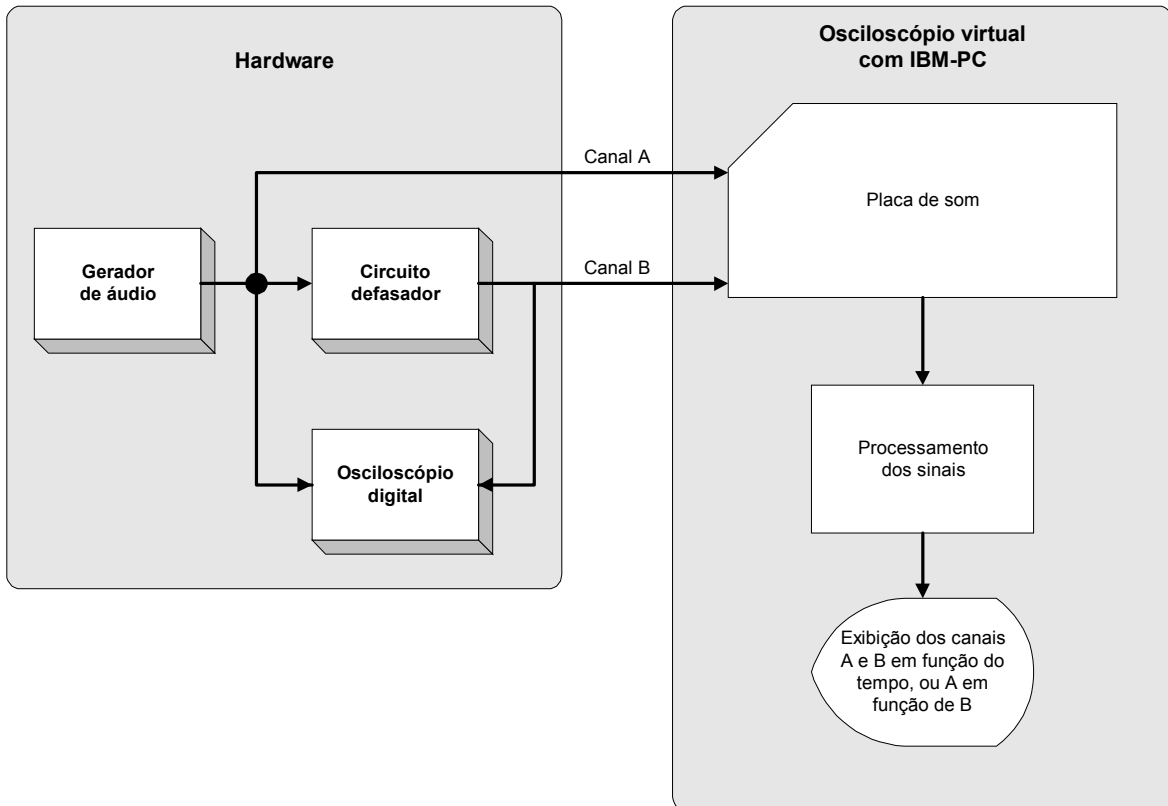
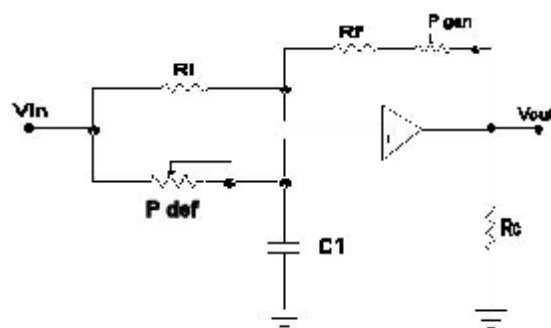


Figura 3.3 – Instrumentação virtual.

Montagem para verificação de fase, frequência e amplitude dos sinais amostrados pela placa de som.

O software escrito em linguagem C++, basicamente implementa um osciloscópio realizando a aquisição de sinais pela placa de som (Figura 3.3). O esquema do circuito de defasagem é apresentado na Figura 3.4.



7

Figura 3.4 – Esquemático do circuito de defasagem.

Implementado com controles de sincronismo entre canais, ganho de amplitude e controle de frequência de amostragem, o software possibilita desenhar na tela do PC os canais direito e esquerdo da placa de som em função do tempo. Possibilita também, desenhar um canal em função do outro, obtendo-se assim, figuras de Lissajous para testes de fase dos sinais. Permite ainda, arquivar os sinais adquiridos. A tela principal do software é mostrada na Figura 3.5.

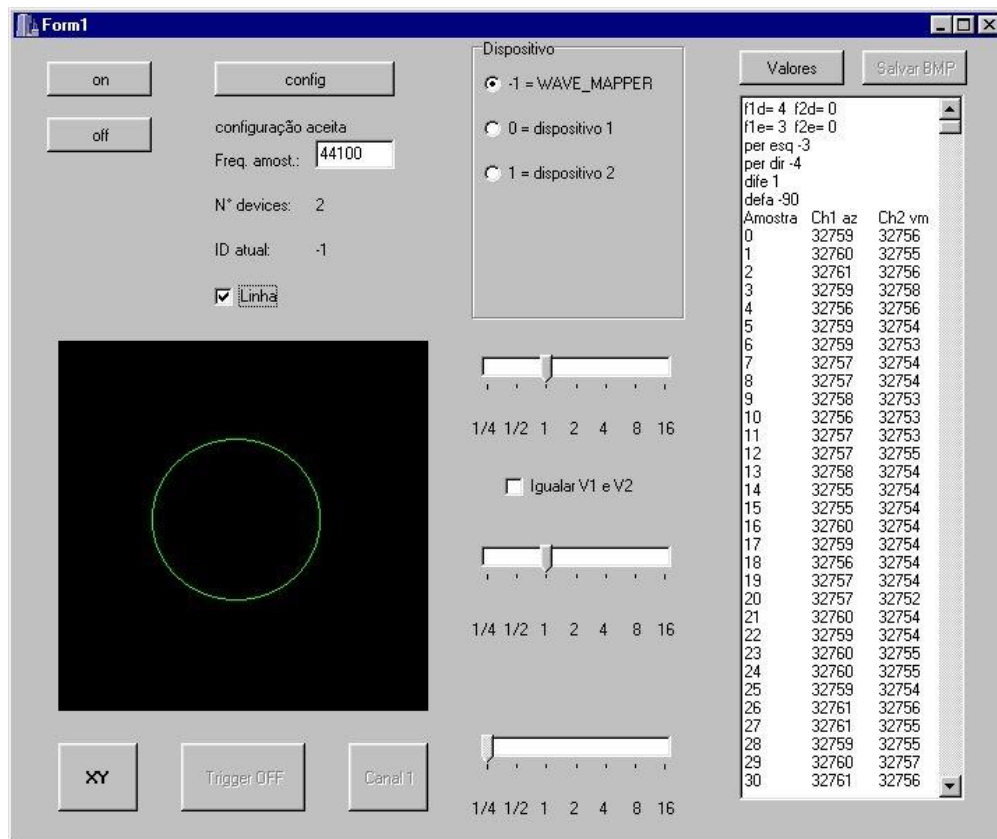


Figura 3.5 – Tela do Osciloscópio virtual.

Exibindo figura de Lissajous de sinais em quadratura de fase.

Resultados deste teste ratificam as conclusões obtidas a partir da análise dos manuais. A mesma diferença de fase indicada pelo osciloscópio é apresentada na tela do PC pelo uso do osciloscópio virtual.

O acesso à configuração da Placa de Som e aos dados amostrados podem ser realizados através de funções da API (*Application Program Interface*) do Windows ou utilizando biblioteca *DirectX* (seção 3.2.3).

Além da placa de som, os sinais podem ser adquiridos utilizando PAS (Placa de Aquisição de Sinais) a ser conectada ao PC. Desde que esta possua as configurações apresentadas na Tabela 3.1. A vantagem de utilizar PAS reside na possibilidade de adquirir outros sinais além do Doppler, como ECG ou PA (Pressão Arterial) (Evans, 1989, Ramos, 2002).

3.2. Software Implementado

Esta seção aborda o software através de três etapas. Iniciando-se com uma breve discussão sobre processamento em tempo real. Em seguida, discute de forma sucinta suas funções e características através de exposição e descrição da interface do aplicativo. Por fim, são apresentados em diagrama de blocos os principais estágios para processamento dos sinais Doppler. A partir deste, expõe-se a metodologia utilizada para implementar o software e detalha-se suas funções.

Aplicações em tempo real, caracterizam-se pelo cumprimento de tarefas em prazo pré-estabelecido. Não importa o quão ínfimo ou grande é o prazo, este deve ser respeitado. Este tempo limite é chamado *deadline*.

Sistemas de tempo real são divididos em tempo real brando (*Soft Real Time*) e tempo real crítico (*Hard Real Time*). Maiores detalhes sobre características e técnicas para garantir cumprimento de *deadline*, assim como aspectos da teoria de escalonamento são encontrados em Farines (2000) e Oliveira (2002).

Existem SOs (Sistemas Operacionais) adequados para aplicações em tempo real. Um exemplo é o Linux, que pode ser utilizado para trabalhar em tempo real brando com bons resultados (Tatibana, 2002). Contudo, há necessidade de implementar o software para o SO utilizado pelo público alvo do projeto. De forma a facilitar a assimilação da interface e permitir o uso nos PCs dos médicos especialistas sem a necessidade da instalação de um segundo SO, optou-se por utilizar o SO MS-Windows™.

Embora o SO Windows não tenha sido desenvolvido para aplicações em tempo real, é possível adotar estratégias para obter melhor desempenho e monitorar o cumprimento de *deadlines*. Através de técnicas de programação, utilização de bibliotecas otimizadas, monitoração do estado dos recursos do PC, *multithreading*² e prioridades,

² *multithreading*: relativo a capacidade de um programa executar múltiplas tarefas ao mesmo tempo. Enquanto multitarefa, é a capacidade que o SO tem de executar mais de um programa ao mesmo tempo.

passa a ser viável o uso deste SO multitarefa (Ramamritham, 1998; Johnson, 2001; MSDN, 2003).

O software implementado para SO Windows trabalha em tempo real brando e roda nas versões 98 e NT (Windows 2000 e Windows XP).

3.2.1. Descrição do Software

O software controla a placa de som para amostrar sinais Doppler relativos ao fluxo sanguíneo (FS) do vaso sanguíneo examinado. Processa também estes sinais tornando possível a obtenção de sonograma e suas envoltórias. Além da exibição de sonograma na tela do PC em tempo real, é possível arquivar os sinais Doppler, assim como as envoltórias (seção 2.2.5) e comentários sobre os dados coletados. Tal abordagem viabiliza futura implementação de banco de dados.

Os sinais Doppler arquivados estão em formato *Wave-PCM* (*Pulse Code Modulation* – Modulação por Código de Pulso, extensão *WAV*), podendo ser manipulados por qualquer software para edição de áudio, embora estes não calculem o sonograma. O software permite ainda, a leitura dos arquivos *WAV* para exibição de sonograma e análise *off-line* dos dados já coletados. Por possuir código aberto, permite implementar novos recursos e realizar adaptações para outros empregos clínicos. Constitui desta forma, um sistema ideal para diagnóstico e pesquisas envolvendo o sistema cardiovascular.

A interface do software está apresentada na Figura 3.6. Sua tela inicial apresenta comandos para escolha da origem dos dados, ou seja, se os sinais Doppler devem ser obtidos a partir da Placa de Som, de arquivo *WAV* ou de outra PAS. Os comandos “*iniciar*” e “*pausa*”, permitem controlar o início e fim da aquisição dos sinais Doppler e a consequente apresentação de sonograma na tela.

A Figura 3.7.a mostra as possibilidades de configuração das vias de aquisição dos sinais. Caso seja utilizada a Placa de Som para aquisição, o comando “*configurações*” da Figura 3.7.a abre o formulário apresentado na Figura 3.7.b. Este contém opções para especificar qual dispositivo usar quando houver mais de uma Placa de Som. A taxa de aquisição, a resolução e tipo de janela para a FFT são também configurados neste formulário.

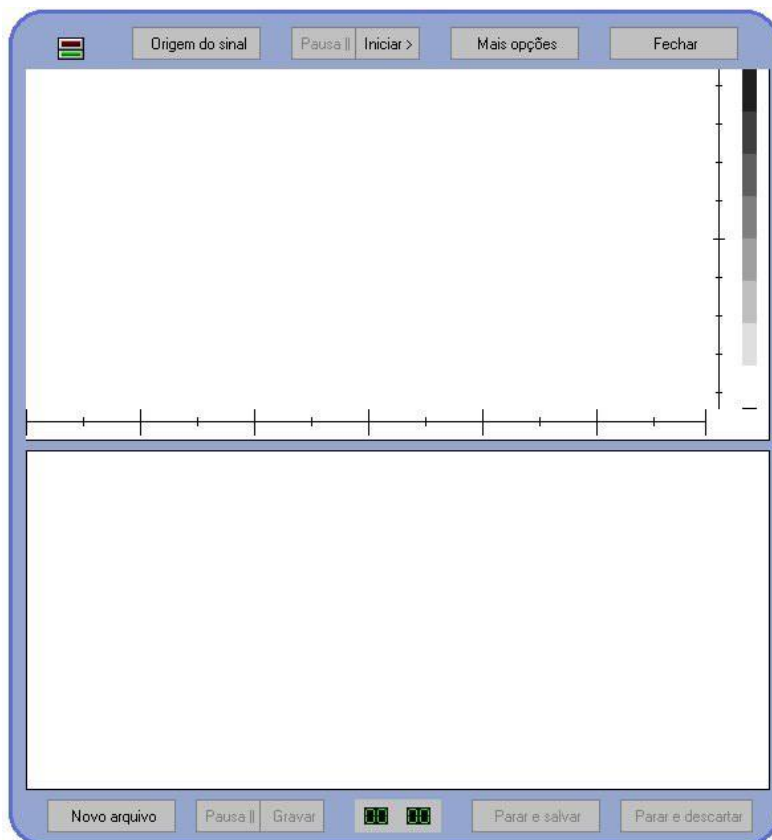


Figura 3.6 – Tela principal do software.

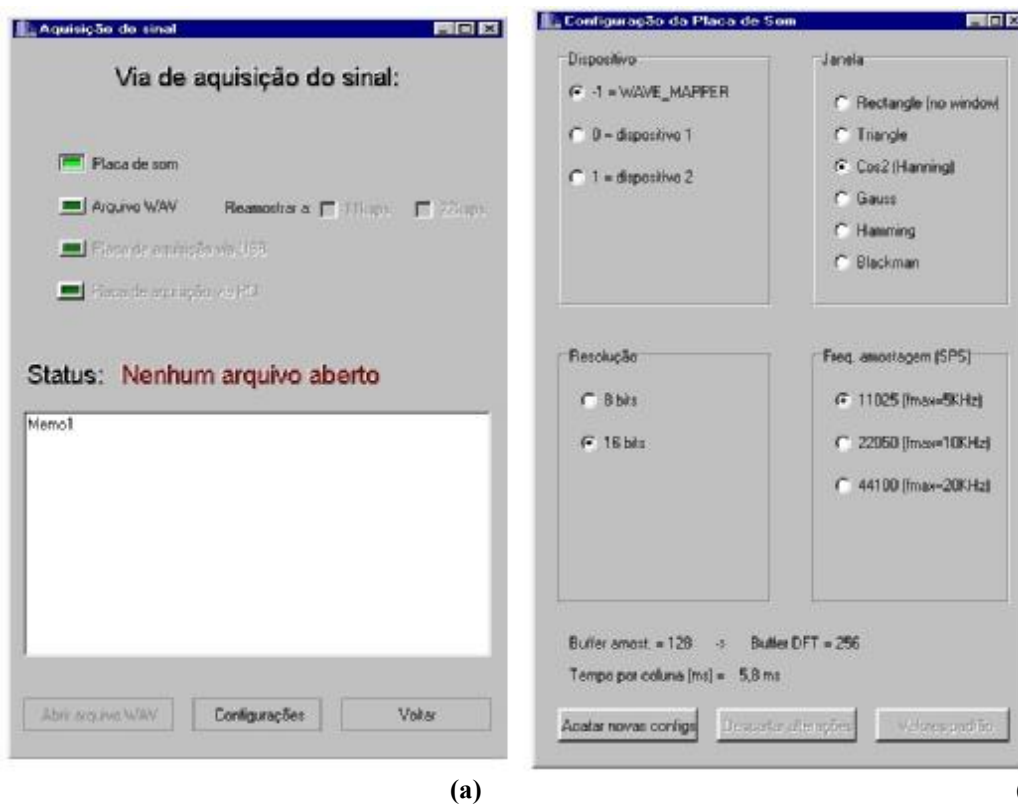


Figura 3.7 – Tela de configuração.

Configuração da via de aquisição dos sinais Doppler (esquerda). Configuração da placa de som (direita).

O comando “*novo arquivo*” (Figura 3.6), abre a tela apresentada na Figura 3.8 para criação dos arquivos de registro de dados.

Após a criação dos arquivos de dados, os comandos “*Pausa*”, “*Gravar*”, “*Parar e salvar*” e “*Parar e descartar*” (Figura 3.6) são habilitados, permitindo ao usuário o controle de gravação do sinal Doppler. É possível interromper momentaneamente a amostragem, assim como parar de gravar e descartar os dados ou parar e salvar em arquivo.

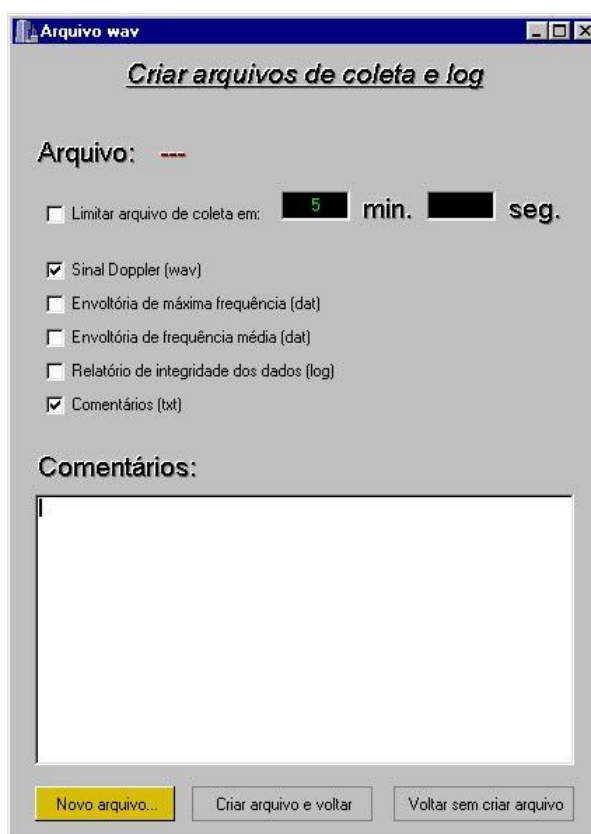


Figura 3.8 – Configuração dos arquivos de registro.

Tela para registro de comentário a respeito dos sinais amostrados e configuração dos arquivos a serem salvos.

Outras opções são disponibilizadas através do comando “*Mais opções*” (Figura 3.6), que apresenta um campo (aba) à direita da tela principal (Figura 3.9). Neste, é possível configurar parâmetros avançados (numerados na legenda para identificação):

- 1 – Re-configura os campos com valores padrão, definidos em arquivo “*ini*”;
- 2 – Controla a intensidade do sonograma exibido na tela através de ganho

aplicado ao sinal obtido pela FFT;

3 – Habilita exibição de sonograma; caso seja utilizado zoom, (3.a) efetua interpolação linear; (3.b) coluna exibida é a média da coluna de FFT anterior com a atual; (3.c) passa um filtro de média móvel nos dados de sonograma (seção 2.2);

4 – Exibe envoltórias (seção 2.2.5), desenhando um ponto de forma independente da coluna anterior, ou interligando-o com ponto da envoltória calculada para coluna anterior, é possível calcular media com a anterior;

5 – Controle da tela do sonograma. Configura exibição do eixo horizontal que separa FS direto e reverso sendo possível inverter os canais. Para uma melhor visualização é possível trabalhar com 16 níveis de cor ou suavizar a apresentação utilizando 256 níveis com ou sem controle de luminosidade (*alpha*) (seção 3.2.6). A tela pode ter fundo preto e sonograma colorido (melhor para visualização na tela do PC) ou fundo branco e sonograma em escala de cinza (ideal para impressão). Selecionando a opção “*Plotar uma tela*”, o software encarrega-se de pausar a aquisição do sonograma ao atingir o fim de tela, compondo uma tela contínua de sonograma.

6 – Conforme exposto na seção 2.2.5, existem diversos algoritmos para cálculo da envoltória de máxima frequência. Este módulo permite escolha do algoritmo a ser utilizado.

7 – Caso o sinal Doppler possua apenas componentes de baixa frequência, esta opção permite *zoom* de 2 ou 4 vezes.

8 – Abre uma nova aba para depuração do software. Utilizado apenas durante a implementação do programa.

9 – Este comando permite salvar a imagem da tela de sonograma em formato *Bitmap*. Obtém-se melhores resultados se utilizado após selecionar opção “*Plotar uma tela*” do quadro 5.

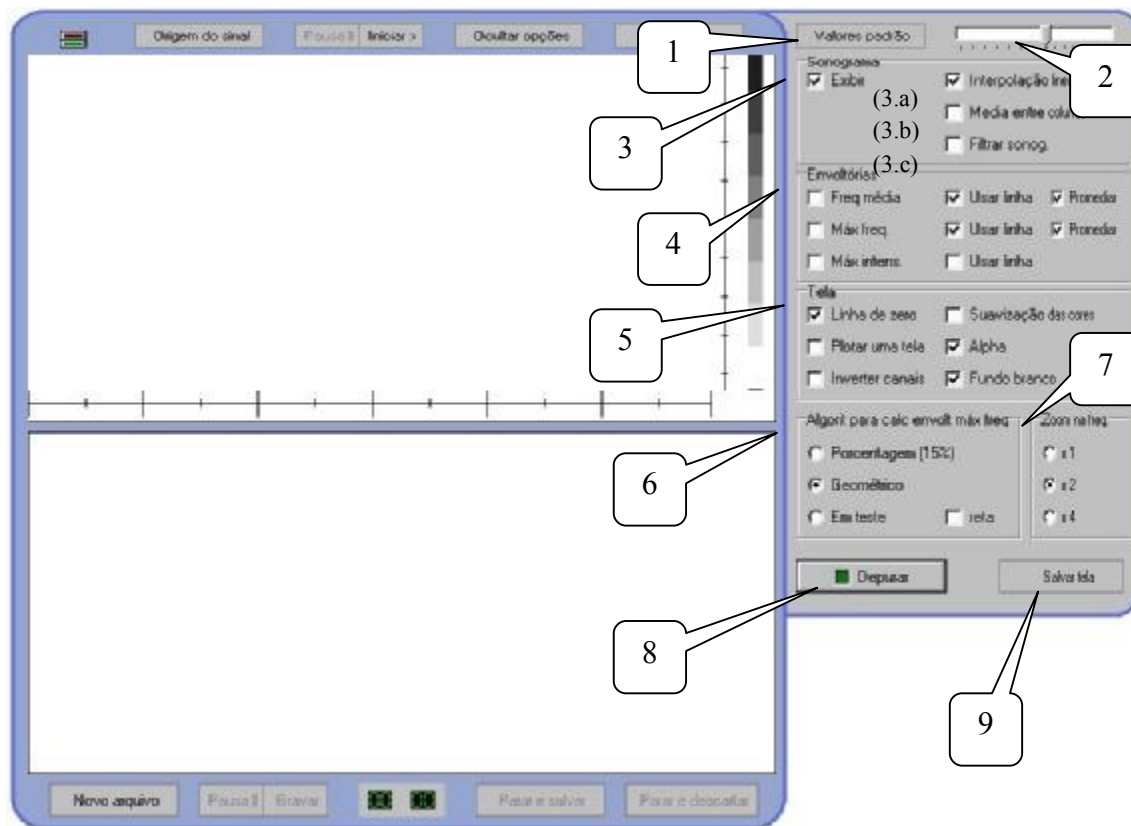


Figura 3.9 – Tela com opções avançadas.
Opções de controle em aba auxiliar da tela principal.

3.2.2. Diagrama de Blocos do Software Implementado

Esta seção descreve o fluxo de tarefas sem detalhar a programação. O software está dividido em três *threads* principais. O diagrama de blocos da Figura 3.10 apresenta cores diferenciadas para cada *thread* utilizada no tratamento dos sinais Doppler, desde a aquisição, processamento, armazenagem até sua exibição.

Ao iniciar a execução do programa, são realizados testes para garantir única instância do aplicativo; mensurar frequência da CPU; verificar demais processos sendo executados pelo SO multitarefa e identificar seus níveis de prioridade. A idéia é garantir que o processo responsável pelo tratamento dos sinais Doppler tenha recursos de máquina suficientes para seu correto funcionamento. Em seguida, a prioridade do processo corrente é incrementada, as *threads* também têm suas prioridades alteradas, conforme necessidade imposta pela tarefa executada.

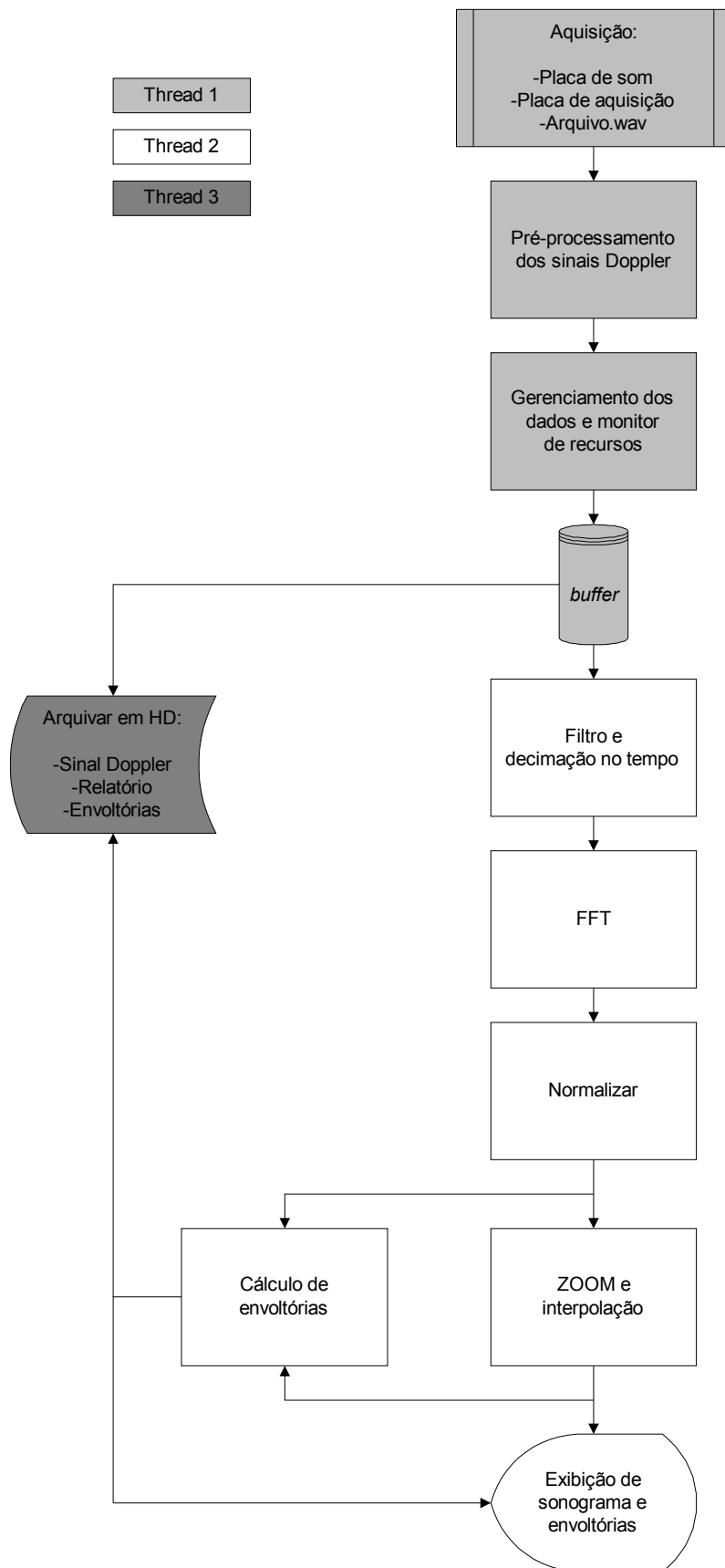


Figura 3.10 – Fluxograma.

Destacando as três principais threads do software.

A *thread* principal do processo e de maior prioridade, é responsável pela amostragem dos sinais Doppler, armazenando-os na memória de onde serão posteriormente lidos pelas *threads* encarregadas de arquivar, calcular e exibir os sinais processados. Esta mesma *thread* efetua verificações para confirmar o desempenho do sistema e se os *deadlines* os estão sendo cumpridos.

Tratamentos e exibição dos sinais são incumbências da segunda *thread*. Esta lê os dados armazenados na memória, efetua decimação no tempo, calcula a FFT destes dados, calcula envoltórias e realiza tratamento do sinal no domínio da frequência para então apresentá-los na tela do PC.

Os dados presentes na memória são também lidos pela terceira *thread*. Esta é responsável por arquivar relatórios e sinais Doppler no disco rígido. Possui a menor prioridade de todas as três *threads* citadas anteriormente.

Os próximos itens deste capítulo descrevem os aspectos de maior relevância para a implementação do software. Detalhando as etapas expostas no diagrama de blocos (Figura 3.10).

3.2.3. Aquisição dos Sinais

SOs que realizam uma única tarefa por vez, não têm restrições de acesso ao hardware. Nos SOs multitarefa, este acesso é uma tarefa complexa, pois dois processos podem requerer o mesmo hardware ao mesmo tempo. Para gerenciar os recursos do PC, o MS-Windows™ trabalha com um sistema de semáforos que controla a utilização dos recursos da máquina. Um processo pode utilizar o hardware quando o sistema torná-lo disponível. Tal operação pode ser realizada através do emprego das APIs (*Application Program Interface*). Contudo, quando se trata de programas com recursos multimídia, (como por exemplo jogos de computador), este acesso não é suficientemente rápido, tampouco eficiente.

O *DirectX* foi criado para solucionar este problema, uma vez que não era possível a execução de jogos no Windows com o mesmo desempenho do MS-DOS. Segundo a Microsoft, *DirectX* é um grupo de tecnologias, processos e métodos, projetado para permitir que um computador com SO Windows constitua uma plataforma ideal para rodar e exibir aplicativos ricos em elementos multimídia, como gráficos, vídeos, animações 3-D

e som (MSDN, 2003). Atualmente, existem movimentos como o *Project Jedi* que visam obter no SO Linux os mesmos resultados propiciados pelo *DirectX* ao Windows (<<http://delphi-jedi.org/>>).

Portanto, o *DirectX* pode ser definido como um conjunto de APIs de baixo nível para criação de jogos e aplicações multimídia de alta performance. Tem como módulos principais o *DirectDraw* (biblioteca utilizada para acesso ao vídeo), o *DirectInput* (acessa dispositivos HID (*Human Interface Device*) como teclado ou joystick), o *DirectPlay* (voltado para vídeos) e o *DirectSound* (destinado para gerenciamento dos recursos de som, componente de maior interesse neste trabalho).

Com a padronização fornecida pelo uso do *DirectX*, além de ser possível desenvolver aplicativos para aquisição de som com alta performance, consegue-se ainda tornar o software compatível com novos dispositivos de hardware. O *DirectX* realiza a interação entre o aplicativo e o hardware, tornando o software capaz de controlar novos dispositivos sem a necessidade de atualizações de programa feito com recursos *DirectX*, basta atualizar o *DirectX*.

O software implementado nesta dissertação requer utilização de recursos multimídia, por este motivo é utilizado o *DirectSound* para configuração da placa de som, gerenciamento de memória e outros recursos que possibilitam a aquisição dos sinais Doppler.

O diagrama de blocos a seguir mostra os passos utilizados para realizar a aquisição dos sinais via placa de som com a utilização do *DirectSound*. As etapas são descritas na sequência, utilizando estrutura das funções, métodos e exemplos de código fonte.

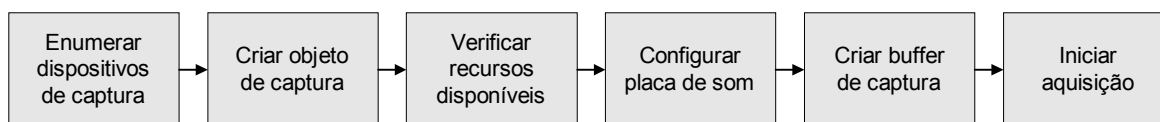


Figura 3.11 – Fluxograma para uso de *DirectSound*.

Inicialmente, detecta-se os dispositivos de som conectados ao PC. Através da função `DirectSoundCaptureEnumerate`, obtém-se a lista dos dispositivos de som

existentes. Permite também, obter informações sobre quais dispositivos de hardware estão disponíveis. A função fornecer ainda um GUID³ (*Globally Unique Identifiers*) para cada dispositivo e checa as capacidades de cada um deles, conforme o código:

```
void __fastcall TForm1::Dx_SoundCapture1DSCaptureEnumerate(TObject
*Sender, GUID *Guid, AnsiString DriverDescription, AnsiString
DriverModule, void *Context, bool &Finished)
{
    if (NumCaptureDevices<MAX_DEVICES_SUPPORTED)
    {
        Guids[NumCaptureDevices] = Guid;
        DriverDescriptions[NumCaptureDevices] = DriverDescription;
        DriverModules[NumCaptureDevices] = DriverModule;
        NumCaptureDevices++;
        ListBox1->Items->Add( DriverDescription+"["+DriverModule+"]");
    }
    else
        Finished = true;
}
```

Nesta listagem, a descrição e módulo do *driver* de som para aquisição são adicionados ao componente `ListBox1`, a partir do qual se escolhe o dispositivo a ser usado para a aquisição. O primeiro item da lista é o dispositivo padrão utilizado pelo Windows, cuja configuração é realizada acessando o ícone *Multimídia* do *Painel de Controle*. No PC utilizado neste trabalho, há duas placas de som. Assim, o comando acima retorna os valores mostrados abaixo. Listando os dispositivos de captura de som instalados no PC.

```
Driver de captura de som primário []
Gravação de SB AudioPCI [emulado][WaveIn0]
VIA Sound System Record [DC00][emulado][WaveIn1]
```

Escolhido o dispositivo para aquisição (entre os listados acima), o próximo passo é criar o objeto `DirectSoundCapture` através da função `DirectSoundCaptureCreate`. Esta, é responsável por criar e iniciar o objeto que suporta a interface `IDirectSoundCapture`, usada para criar *buffers* de captura de som, no qual são armazenados os sinais amostrados. O quadro a seguir exibe o corpo da função utilizada

³ GUID (*Globally Unique Identifiers*): O conceito de identificador único global, ou identificador globalmente exclusivo, provê identificação única para interfaces. Representado por 128bits.

para este fim, onde `lpcGUID` deve ser um dos valores GUID retornados pela função `DirectSoundCaptureEnumerate`.

```
HRESULT WINAPI DirectSoundCaptureCreate(
    LPCGUID lpcGUID,
    LPDIRECTSOUNDCAPTURE *lpDSC,
    LPUNKNOWN pUnkOuter );
```

Para encontrar as características do dispositivo de captura, utiliza-se chamada ao método `IDirectSoundCapture::GetCaps`. O parâmetro deste método é um ponteiro para a estrutura `DSCCAPS`, indicando o número de canais e formato do sinal suportado pelo dispositivo.

```
HRESULT GetCaps(
    LPDSCCAPS lpDSCCaps );

typedef struct {
    DWORD dwSize;
    DWORD dwFlags;
    DWORD dwFormats;
    DWORD dwChannels;
} DSCCAPS, *LPDSCCAPS;

typedef const DSCCAPS *LPCDSCCAPS;
```

A variável `dwFormats` contém a estrutura usada nas funções de áudio para formas de onda. A Tabela 3.3 indica os possíveis valores de retorno e seus significados.

Tabela 3.3 – Estruturas dos dados de áudio.

Valor	Significado (taxa de amostragem, canais e resolução)
WAVE_FORMAT_1M08	11.025 kHz, mono, 8-bit
WAVE_FORMAT_1M16	11.025 kHz, mono, 16-bit
WAVE_FORMAT_1S08	11.025 kHz, stereo, 8-bit
WAVE_FORMAT_1S16	11.025 kHz, stereo, 16-bit
WAVE_FORMAT_2M08	22.05 kHz, mono, 8-bit
WAVE_FORMAT_2M16	22.05 kHz, mono, 16-bit
WAVE_FORMAT_2S08	22.05 kHz, stereo, 8-bit
WAVE_FORMAT_2S16	22.05 kHz, stereo, 16-bit
WAVE_FORMAT_4M08	44.1 kHz, mono, 8-bit
WAVE_FORMAT_4M16	44.1 kHz, mono, 16-bit
WAVE_FORMAT_4S08	44.1 kHz, stereo, 8-bit
WAVE_FORMAT_4S16	44.1 kHz, stereo, 16-bit

Através do método `IDirectSoundCapture::CreateCaptureBuffer` do objeto `DirectSoundCapture`, é criado *buffer* de captura de dados para armazenar os sinais amostrados. São passados parâmetros para definir o formato do *buffer*, devendo ser o mesmo do *buffer* primário de captura. O código abaixo, exemplo do arquivo de ajuda do *DirectX*, ilustra a criação de *buffer* para captura de áudio, através da estrutura `WAVEFORMATEX` para trabalhar com o formato PCM sem compactação, com dois canais, 44100 amostras por segundo, 176400 bytes por segundo e resolução de 16 bits. É possível utilizar outros formatos áudio que são especificados em *mmreg.h*.

```
DSCBUFFERDESC dscbd;
LPDIRECTSOUNDCAPTUREBUFFER pDSCB;
WAVEFORMATEX wfx = {WAVE_FORMAT_PCM, 2, 44100, 176400, 4, 16, 0};
dscbd.dwSize = sizeof(DSCBUFFERDESC);
dscbd.dwFlags = 0;
dscbd.dwBufferBytes = wfx.nAvgBytesPerSec;
dscbd.dwReserved = 0;
dscbd.lpwfxFormat = &wfx;
pDSCB = NULL;
HRESULT hr = pDSC->CreateCaptureBuffer(&dscbd, &pDSCB, NULL);
```

Realizadas as configurações, inicia-se a captura dos sinais de áudio através do método `IDirectSoundCaptureBuffer::Start`. É usual configurar o *buffer* de captura para um tamanho ditado pelo tempo que se deseja adquirir os sinais. Assim, quando o *buffer* é preenchido, a captura é finalizada automaticamente e os dados são armazenados em arquivo através do código `Dx_SoundCaptureBuffer1->SaveToFile(filename, &WaveFormat, sizeof(WAVEFORMATEX));`.

Contudo, neste trabalho é necessário amostrar constantemente os dados para cálculos e exibição dos resultados (sonograma e envoltórias) na tela do PC em tempo real de forma ininterrupta. Além disto, o usuário pode requerer arquivar um trecho do sinal Doppler cujo tempo inicial de captura e período total não é informado antecipadamente. Conforme citado na seção 3.2.1, o usuário pode requerer a gravação de sinais, como também interrompê-la a qualquer momento (Figura 3.6). Para satisfazer esta condição, o *buffer* é configurado para funcionar em modo circular.

Devido a este modo de funcionamento, é necessário obter a posição em que os dados estão sendo armazenados na parte da memória reservada como *buffer* de dados

(*lpdwCapturePosition*). De posse deste e do endereço que indica a posição da qual é seguro ler os dados (*lpdwReadPosition*), a próxima ação é reservar uma região do *buffer* para efetuar a leitura dos dados. Após a reserva, esta região do *buffer* torna seu conteúdo indisponível para alterações. Para isto são utilizados os métodos de captura da posição atual no *buffer* de captura (`IDirectSoundCaptureBuffer::GetCurrentPosition`) e o método de reserva (`IDirectSoundCaptureBuffer::Lock`).

```
HRESULT GetCurrentPosition(  
    LPDWORD lpdwCapturePosition,  
    LPDWORD lpdwReadPosition );  
  
HRESULT Lock(  
    DWORD dwReadCursor,  
    DWORD dwReadBytes,  
    LPVOID *lplpvAudioPtr1,  
    LPDWORD lpdwAudioBytes1,  
    LPVOID *lplpvAudioPtr2,  
    LPDWORD lpdwAudioBytes2,  
    DWORD dwFlags );
```

Dois ponteiros são requeridos para ler a memória ao utilizar *buffer* circular. O método *Lock* recebe valores de *offset* e tamanho em bytes a serem reservados para leitura. Retorna assim, a posição da memória do primeiro conjunto de dados que podem ser lidos. Se o primeiro conjunto a ser lido engloba o fim do *buffer*, o restante é direcionado para o início do *buffer*. Este segundo conjunto é então acessado através do ponteiro *lplpvAudioPtr2*, sendo seu tamanho armazenado em *lpdwAudioBytes2*.

A função *CopyMemory* copia os dados do *buffer* de captura para outra região da memória que será acessado pelas *threads* descritas na seção 3.2.2. Por fim, a região reservada para leitura é disponibilizada para alterações através do método `IDirectSoundCaptureBuffer::Lock` até a próxima leitura. É crucial seguir esta sequência de comandos: reservar copiar e disponibilizar, para só depois processar os dados. O *buffer* não pode ficar muito tempo em reserva para evitar perda de dados ou gerar erro no processo. Este trecho de código mostra como copiar os dados presentes no *buffer* de captura para *buffer* auxiliar:


```

if (Dx_SoundCaptureBuffer1->Lock( 0, DSCBufferDesc1->BufferBytes,
                                     &srce, &srce_bytes, &srce_dummy,
                                     &srce_bytes_dummy,
                                     DSCBLOCK_ENTIREBUFFER ))
{
    if (Dx_SoundBuffer1->Lock( 0, DSBufferDesc1->BufferBytes,
                               &dest, &dest_bytes, &dest_dummy,
                               &dest_bytes_dummy,
                               DSBLOCK_ENTIREBUFFER ))
    {
        CopyMemory( dest, srce, dest_bytes );
        Dx_SoundBuffer1->Unlock( dest, dest_bytes, dest_dummy,
                                 dest_bytes_dummy );
    }
    Dx_SoundCaptureBuffer1->Unlock( srce, srce_bytes,
                                    srce_dummy, srce_bytes_dummy );
}

```

Embora tenha sido utilizada a placa de som do PC para amostrar os sinais Doppler, é possível adotar a mesma estratégia descrita anteriormente para digitalizar sinais através de outra placa de aquisição de dados (Figura 3.12). O sistema utilizado para compartilhar os sinais amostrados com as demais *threads* do processo sendo executadas no PC é descrito na próxima seção.

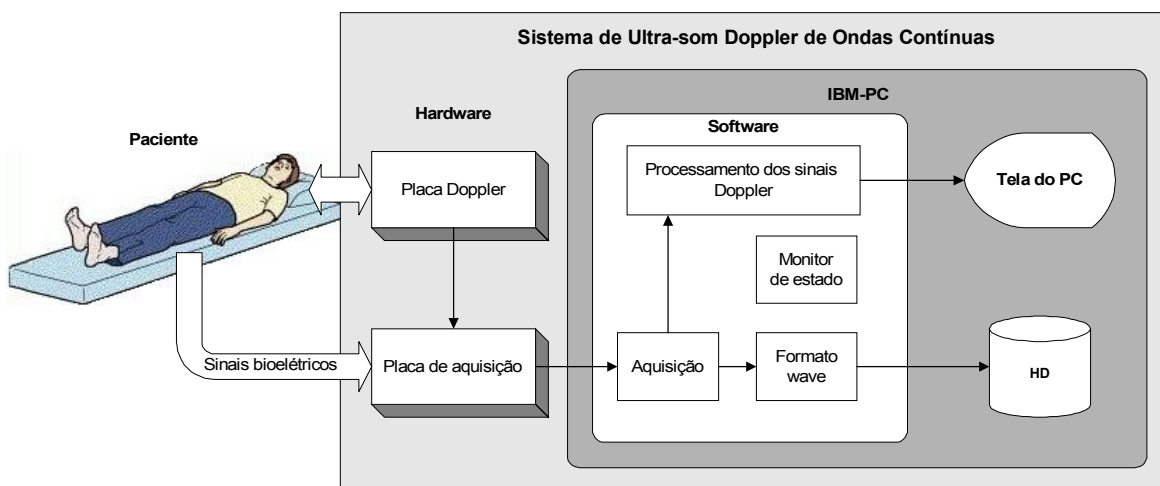


Figura 3.12 – Sugestão de acréscimo de PAS para digitalizar sinais.

3.2.4. Coordenação do Processamento em Tempo Real

São utilizadas 256 amostras do sinal Doppler para compor uma coluna de sonograma, determinando um intervalo de amostragem de aproximadamente 5,8ms para

uma taxa de amostragem de 44kHz. Por este motivo é utilizado *buffer* de 256 posições, que quando preenchido é disponibilizados para as *threads* de cálculo e de armazenamento (Figura 3.13). Utilizar *threads* separadas para cada tarefa torna o software mais robusto e permite executar tarefas simultâneas dentro de um mesmo processo, sem que uma função precise aguardar o término da anterior.

Assim, conforme discutido na seção 3.2.2, três *threads* passam a acessar a mesma posição de memória: A *thread1* escreve na memória os sinais Doppler pré-processados, a *thread2* lê estes dados para processá-los e exibir na tela do PC, enquanto a *thread3* acessa a memória e envia os sinais Doppler para disco rígido, armazenando-os em arquivo de áudio com extensão *WAV*, formato PCM sem compactação (Figura 3.10 e Figura 3.13).

Para evitar conflitos, inconsistência nos dados e mal funcionamento do sistema, é habitual o uso de semáforos e variável do tipo *mutex*⁴. Contudo, tais práticas tornam o sistema mais lento. Existem outras soluções, como o uso de *timers* de sincronismo ou comandos de sincronismo entre *threads* (*sync()*) (Hollingworth, 2000). Este trabalho visa buscar o máximo de desempenho e confiabilidade. Para atingir tal meta, optou-se pelo uso de função de API para entrada em seção crítica (Johnson, 2001), estabelecendo desta forma um sistema próprio de *flags* que atuam como semáforos para acesso aos dados e recursos compartilhados.

O SO Windows NT possui APIs avançadas para criação, entrada, saída e teste de sessão crítica. Melhorando assim, o desempenho e flexibilizando a implementação do código. Contudo, como o objetivo é desenvolver um software para ser executado tanto em Windows NT como 98, são utilizadas apenas as APIs que funcionam em ambas versões do Windows.

A Figura 3.13 ilustra o mecanismo de compartilhamento dos dados. São utilizados *n buffers* de 256 posições, onde *n* é dinâmico. O valor habitualmente utilizado é $n=512$. Este valor representa uma tela inteira de sonograma. Utiliza-se o conceito de memória circular, ao chegar no *buffer n*, o próximo a ser acessado será o *buffer 0* (zero).

⁴ *MUTEX* – *Mutual Exclusion*, método de sincronização de acessos múltiplos para fontes de informação combinadas, através do mecanismo “*lock / unlock*” que tranca e abre alternadamente o acesso ao recurso compartilhado (<<http://www.britannica.com>>).

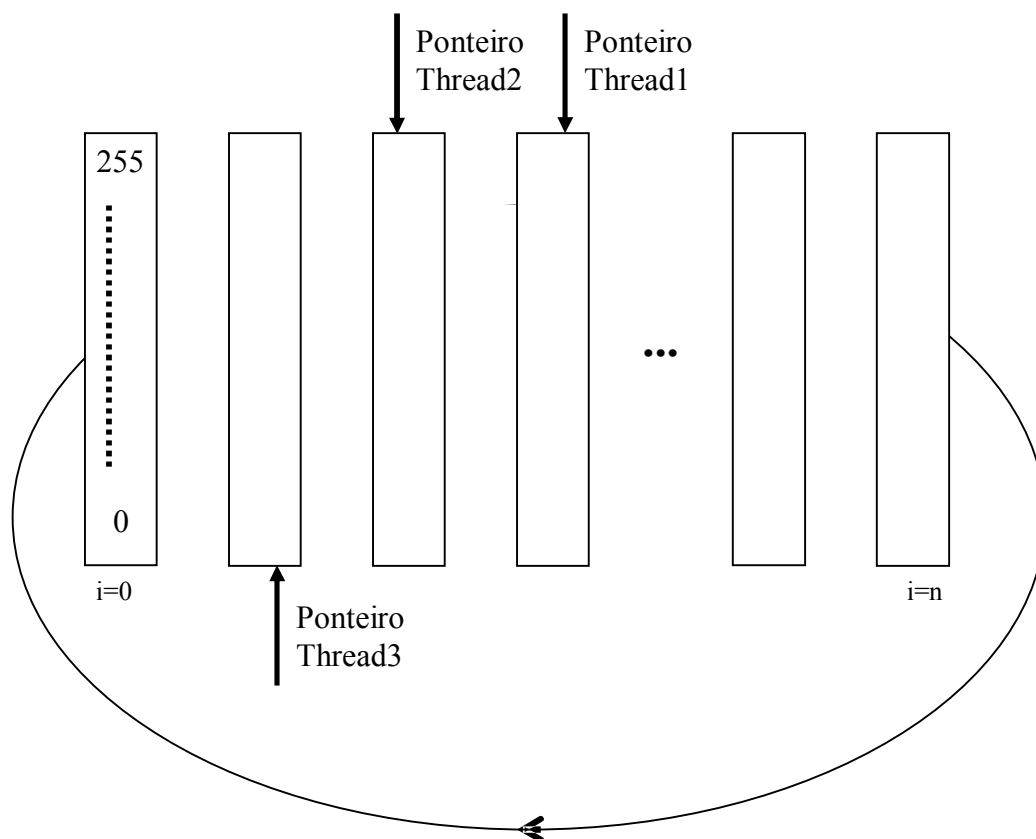


Figura 3.13 – Memória circular.

Compartilhamento dos dados entre as *threads* do processo utilizando memória circular, composta por n setores (*buffers*) de 256 posições.

Cada *thread* tem um ponteiro de acesso global que fornece o endereço da memória que está sendo acessada. No exemplo da Figura 3.13, a *thread1* é responsável por escrever no buffer, enquanto as *thread 2* e *3* são responsáveis pelo processamento e estocagem dos dados, respectivamente.

Toda vez que a *thread1* acessar um buffer para escrita, ela entra numa seção crítica. Assim, a *thread1* tem acesso exclusivo a esta posição de memória. As *threads 2* e *3* não podem acessar esta região de memória, ficarão aguardando até que a *thread1* encerre sua tarefa e libere o buffer. Quando a *thread1* sai da sessão crítica e libera o buffer, os ponteiros das *threads 2* e *3* são incrementados para acesso aos dados disponibilizados pela *thread1*. Quando as *threads 2* e *3* iniciam o processo de leitura, entram em suas respectivas sessões críticas.

O código a seguir exemplifica a entrada de uma *thread* em sessão crítica para acesso a ponteiros.

```

void __fastcall Tthread3_HD::Execute()
{
    EnterCriticalSection(&Form1->GCSbi1);
    Form1->Gbi2=Form1->Gbi1;
    LeaveCriticalSection(&Form1->GCSbi1);
    FreeOnTerminate=true;
    while (1)
    {
        EnterCriticalSection(&Form1->GCSgw);
        if (Form1->GgravarWav==false) break;
        LeaveCriticalSection(&Form1->GCSgw);
        EnterCriticalSection(&Form1->GCSbi1);
        auxbi1=Form1->Gbi1;
        LeaveCriticalSection(&Form1->GCSbi1);
        if (Form1->Gbi2!=auxbi1)
        {
            Executaumafuncaoqualquer();
            EnterCriticalSection(&Form1->GCSbi2);
            Form1->Gbi2++;
            if (Form1->Gbi2>99) Form1->Gbi2=0;
            LeaveCriticalSection(&Form1->GCSbi2);
        }
    }
}

```

Caso a *thread 2* ou *3* sofra atrasos, o ponteiro da *thread1* pode vir a alcançar o endereço no qual se encontra o ponteiro da *thread* atrasada, uma vez que é utilizada memória circular. Para evitar tal evento, antes da *Thread1* incrementar seu ponteiro, entra uma sessão crítica e verifica se há alguma outra *thread* utilizando a próxima posição de memória. Caso houver, está caracterizado atraso excessivo de processamento e memória cheia (*overflow*). Neste caso, o procedimento a ser adotado é informar falta de recursos ao usuário e o software toma medidas para recuperar sua eficiência.

Ao invés de descartar os dados, o ponteiro da *thread* atrasada é incrementado. Desta forma, se a outra *thread* estiver funcionando corretamente, sua função não será afetada. Um exemplo: se a *thread2*, que é responsável pelo cálculo e exibição de sonograma apresentar atraso, a ponto da *thread1* alcançar a *thread2*, o ponteiro da *thread2* é incrementado e alguns dados não serão exibidos na tela. Contudo, a *thread3* não será afetada. Caso houvesse descarte de dados (esvaziar a memória para recuperar recursos e eficiência), mesmo se a *thread3* não tivesse atrasada, seria afetada.

Neste caso, um arquivo texto com extensão *log* é gerado, contendo informações sobre a ocorrência do evento.

Existe uma variável do tipo `CRITICAL_SECTION` para controlar o acesso a cada ponteiro (*GCSbi1*, *GCSbi2* e *GCSgw*, ver código acima). A *thread* entra em sessão crítica para verificar o endereçamento dos ponteiros e só então, incrementa ou não, o seu próprio

ponteiro. Desta forma, é possível verificar o nível de preenchimento do *buffer*, bastando para isto verificar o espaço de memória entre a *thread* de leitura e a *thread* de escrita, a partir do valor de seus ponteiros.

Integridade dos Dados Gerados

Além de verificar o nível de preenchimento do *buffer*, conforme descrito acima, são adotados outros testes para determinar se o processo está trabalhando em tempo real e se não está ocorrendo perda de dados.

Sabendo que, uma tela completa de sonograma é constituída por 512 colunas. Cada coluna é construída com 256 amostras. A taxa de aquisição é conhecida, logo é possível calcular em quanto tempo uma tela deve ser preenchida. Verificando o tempo de início e término de exibição de uma tela de sonograma, é possível verificar se naquela tela de houve perda significativa de dados. Caso haja falta de recurso computacional, uma ou mais colunas não serão exibidas no prazo previsto (atraso no processamento e exibição), desta forma, será despendido mais tempo para preencher uma tela de sonograma (composta por 512 colunas). Fazendo a diferença, entre o tempo previsto para exibição de uma tela completa e o tempo utilizado para tal, obtém-se um indicativo do desempenho de exibição do sonograma.

$$tempo = \frac{colunas \cdot pontos}{taxa} \quad 3.1$$

Onde:

tempo = tempo gasto para exibir uma tela completa de sonograma (segundos);

pontos = número de pontos que compõe uma coluna de sonograma;

colunas = número de colunas que compõe uma tela completa de sonograma;

taxa = taxa de aquisição de dados (Hz).

Considerando que, devido a eventuais atrasos, uma tela necessita de alguns milissegundos a mais para ser preenchida, se na próxima tela houver folga de recursos este tempo será recuperado. Assim o índice *tempo por tela* não é exato, mas pode ser estimado pela Equação 3.1. Caso o tempo exceda o valor previsto (considerando um erro de até 10% como aceitável), indica que o sistema não está trabalhando em tempo real e os *deadlines* não estão sendo cumpridos.

Outro procedimento utilizado é monitorar o uso da CPU, através de funções da API do Windows. Desta forma, quando houver falta de recursos, o usuário é avisado antes de ocorrer estouro de *buffer*. Os índices de desempenho aos quais o usuário pode ter acesso são: o nível de preenchimento do *buffer*, o tempo por tela de sonograma e uso da CPU.

Com os procedimentos descritos neste item, os dados são compartilhados pelas *threads* que constituem o processo. O sistema é monitorado para garantir a integridade dos dados gerados. Assim, os dados são acessados pelas *threads* responsáveis pelo cálculo, exibição e gravação dos dados possibilitando a execução das tarefas em tempo real brando.

3.2.5. Implementação dos Cálculos

Conforme descrito na sessão 2.2, para construir o sonograma é necessário processar matematicamente os sinais. A sequência de implementação dos algoritmos que efetuam tal tarefa está no diagrama de blocos apresentado na Figura 3.14.

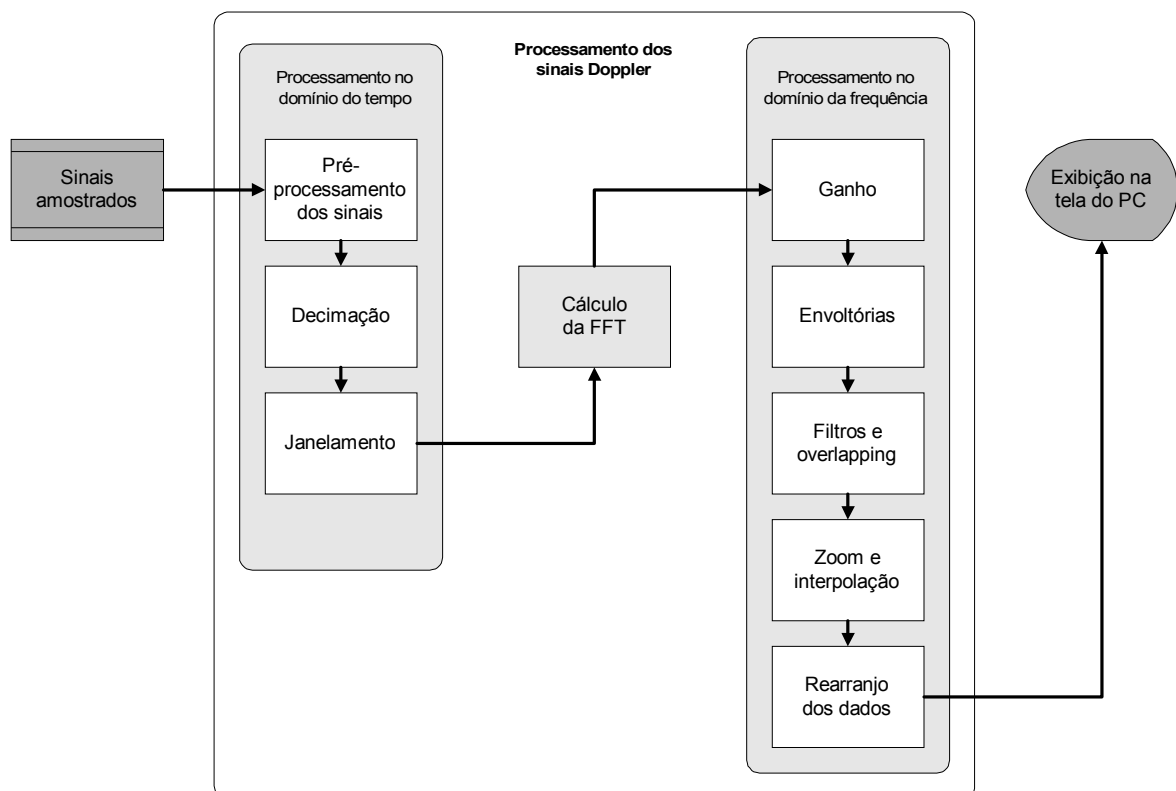


Figura 3.14 – Principais etapas de processamento dos sinais Doppler. Processamento se dá no domínio do tempo (esquerda) e da frequência (direita).

Pré-processamento

O pré-processamento dos sinais Doppler consiste em reorganizar os dados amostrados do *buffer* da placa de som, de forma a serem compatíveis com o próximo estágio de cálculo. Basicamente, os dados são convertidos do formato definido pela estrutura *WAVETEX* para variáveis *WORD* (16 bits).

Com os sinais no formato adequado, é utilizada técnica de *overlapping*, ou sobreposição, para contornar problemas de descontinuidade causados pela segmentação dos dados ocorridos durante o processo de digitalização (Aydin, 1994).

Decimação

Conforme a frequência da portadora de ultra-som utilizada e as componentes de velocidade do sangue, o espectro de frequências do sinal Doppler é alterado. Caso a maior frequência de interesse no sinal Doppler seja inferior a um quarto da taxa de amostragem, a resolução será inadequada para discriminar frequências, prejudicando o cálculo de envoltória. Assim, quando os sinais Doppler contêm componentes de baixa frequência é possível utilizar decimação no tempo para melhorar a apresentação e visualização do sonograma.

A decimação no tempo consiste em reduzir a frequência de amostragem do sinal, podendo levar a ocorrência de *aliasing*. Contudo, este procedimento é utilizado somente quando a maior componente de frequência analisada é menor que a metade da frequência de amostragem do sinal contínuo, respeitando o critério de Nyquist (Oppenheim, 1983).

Tal procedimento implica numa redução da banda de frequências analisadas, levando, contudo a um aumento da resolução em frequência na área de interesse do sinal analisado. Tendo como resultado o efeito de *zoom* (aproximação) nas componentes de baixa frequência do sonograma.

A Figura 3.15 e Figura 3.16 mostram um sinal com o mesmo espectro de frequências, onde a maior componente de frequência é 10KHz. Na Figura 3.15 foi utilizada amostragem de aquisição de 44100Hz e na Figura 3.16 a taxa de amostragem foi de 22050Hz. A diferença de visualização é nítida.

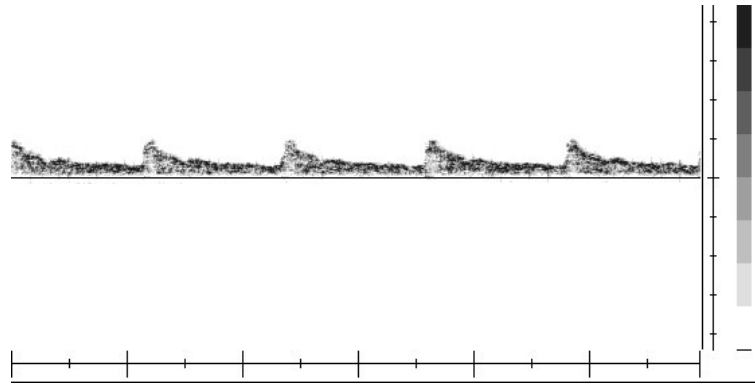


Figura 3.15 – Sonograma utilizando amostragem de 44100Hz.

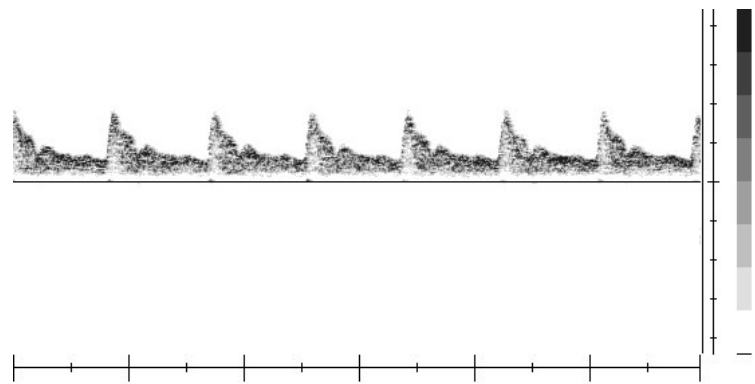


Figura 3.16 – Sonograma utilizando amostragem de 22050Hz.

Cálculo da FFT

Após a decimação no tempo dos sinais Doppler (caso esta seja necessária), o próximo estágio é passá-lo do domínio do tempo para o da frequência separando o fluxo direto do fluxo reverso. Utilizando a FFT complexa tal tarefa é realizada com sucesso. Algoritmos para FFT são encontrados em Tompkins (1995) e Press (2002). Contudo, existem bibliotecas otimizadas utilizando linguagem *assembly* do processador. Como exemplo citam-se as bibliotecas oferecidas pela Intel (<<http://www.intel.com>> e <<http://www.fftw.org>>). Muitos destes algoritmos incluem o código de janelamento. Por terem código aberto (*Open Source*) torna-se possível efetuar alterações no código.

O sinal digitalizado pode conter ruídos e descontinuidades interferindo na qualidade do sonograma gerado. Para resolver estes problemas são utilizadas técnicas de filtragem do espectro e promediação com amostras passadas. A diferença entre os resultados originais e filtrados podem ser apreciados nas Figura 3.17 e Figura 3.18.

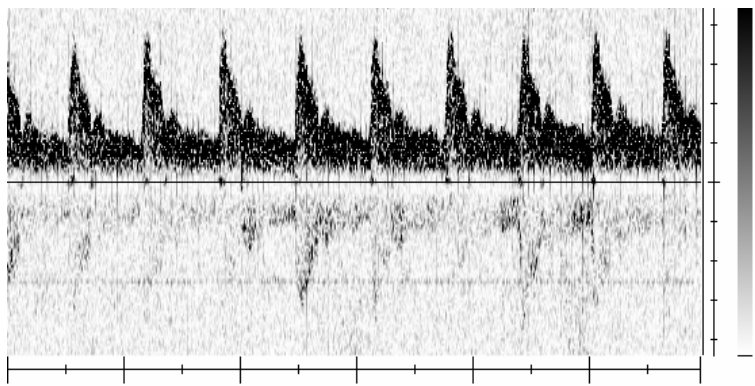


Figura 3.17 – Sonograma sem filtragem.

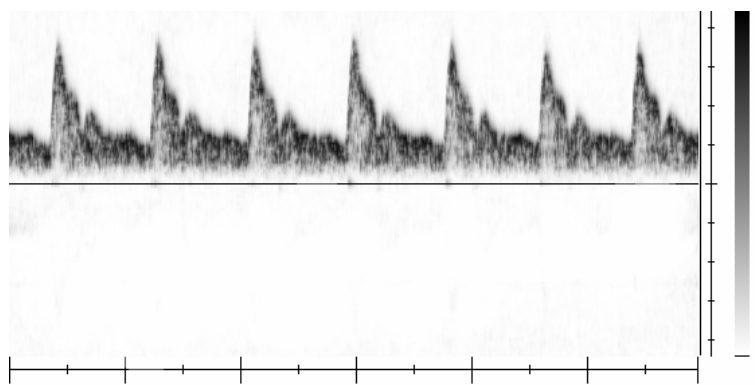


Figura 3.18 – Sonograma com filtragem.

Com o sinal no domínio da frequência, é possível fazer correções de ganho para determinadas faixas de frequência, atenuando componentes de frequência sem interesse ou que representem sinais indesejados.

Cálculo de envoltórias

As envoltórias são calculadas a partir do espectro de frequências. O algoritmo segue os passos descritos na seção 2.2.5, havendo a opção de escolher qual algoritmo a ser utilizado na sua determinação.

O sinal no domínio da frequência contendo o espectro das frequências positivas e negativas é armazenado na memória. A partir destes dados, a componente de maior frequência, tanto para o espectro positivo quanto para o negativo, é obtida. É também calculada a frequência média conforme equação 2.22. De posse destes valores, calcula-se a frequência máxima. Se a frequência média for positiva, a envoltória de máxima frequência deve ser calculada para as frequências positivas, caso a média seja negativa, a envoltória de máxima frequência será calculada no campo das frequências negativas. É utilizada

histerese para obter o ponto de mudança.

Para cálculo da frequência máxima, é empregada a equação 2.23, ou regras trigonométricas para achar o ponto de máxima frequência conforme descrito pelo método geométrico (Figura 2.14).

Para o método geométrico, é calculada a integral do espectro de frequência e arquivado em um *buffer* auxiliar (*Integrado[n]*). A Figura 3.19 contém os valores da integral obtida, nesta mesma figura estão indicados os pontos P_1 e P_2 , a partir dos quais encontra-se a reta r_1 .

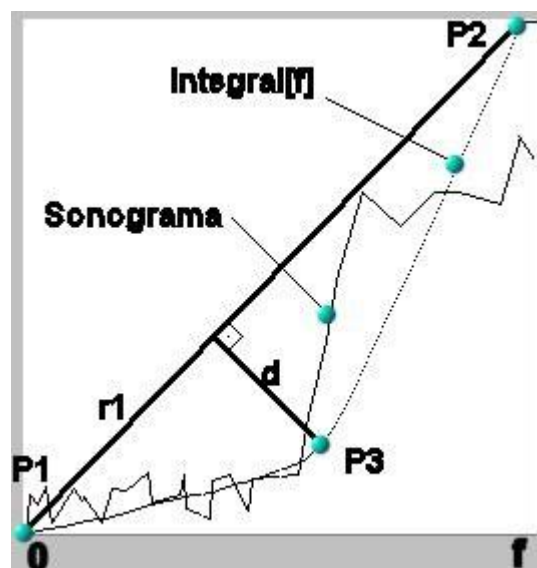


Figura 3.19 – Base para algoritmo do método geométrico.

A equação da reta r_1 é encontrada através da regra de Sarrus para determinante. Considerando dois pontos conhecidos (P_1 e P_2) da reta r_1 (Figura 3.19), é possível achar a equação da reta:

$$\begin{vmatrix} P_{1x} & P_{1y} & 1 \\ P_{2x} & P_{2y} & 1 \\ P_x & P_y & 1 \end{vmatrix} = 0$$

Pela regra de Sarrus:

$$\begin{vmatrix} P_{1x} & P_{1y} & 1 \\ P_{2x} & P_{2y} & 1 \\ P_x & P_y & 1 \end{vmatrix} \begin{vmatrix} P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ P_x & P_y \end{vmatrix}$$

$$P_{1x} \cdot P_{2y} + P_{1y} \cdot P_x + P_{2x} \cdot P_y - P_x \cdot P_{2y} - P_y \cdot P_{1x} - P_{2x} \cdot P_{1y} = 0$$

Considerando P_1 na origem:

$$P_{1x} = P_{1y} = 0$$

Considerando P_2 no final do *buffer* que contém os valores de integração (*Integrado[ff]*):

$$P_{2x}=f \text{ e } P_{2y}=\text{Integral}[f]$$

Onde f representa a posição da componente de frequência de maior amplitude, e *Integral[ff]* sua amplitude. Obtém-se a equação da reta $r1$ da Figura 3.19:

$$-P_x \cdot \text{Integral}[f] + f \cdot P_y = 0$$

Sendo a equação da reta conhecida como:

$$Ax + By + C = 0$$

Os valores de A e B são:

$$A = -\text{Integral}[f] \text{ e } B = f$$

Com os coeficientes A e B da reta $r1$, é possível achar a distância d , entre a reta $r1$ e a curva representada pela integral (*Integrado[ff]*) (Figura 3.19). A seguir é apresentada a equação para achar a menor distância entre uma reta e um ponto.

$$d = \frac{|A \cdot x + B \cdot y + C|}{\sqrt{A^2 + B^2}} \quad 3.2$$

Onde:

d = distância entre reta $r1$ e ponto pertencente à curva de integração;

A , B e C = coeficientes da reta $r1$;

x e y = coordenada do ponto pertencente à curva de integração.

O ponto no qual a distância d apresenta seu maior valor indica a máxima frequência. Abaixo está listado trecho do código implementado para achar a máxima frequência no espectro negativo.

```
raiz=sqrt( (integrado[f]*integrado[f])+(f*f) );
ar=integrado[f];
br=f;
dmax=0; FreqMax=0;
for (i=0;i<=f;i++)
{
    d=abs(-ar*i + br*integrado[i])/raiz;
    if (d>=dmax)
    {
        dmax=d;
        FreqMax=i-1;
    }
}
```

Ampliação da Imagem (controle de zoom)

Foi também implementado um algoritmo para realizar *zoom* na frequência (eixo das ordenadas) com valores de multiplicação de duas ou quatro vezes. Os pontos entre um valor e o próximo podem ser repetidos ou interpolados linearmente. Estas opções são configuráveis pelo usuário do software em tempo de execução.

Após o tratamento matemático descrito, os dados são rearranjados para obter compatibilidade com a resolução da tela e então, enviados à tela do PC. Compõem assim uma coluna de sonograma com as envoltórias de frequência média e máxima. Para obter uma tela de sonograma, todo este processo deve se repetir a cada 5,8ms, este tempo varia conforme taxa de aquisição e quantidade de amostras por coluna for utilizada.

3.2.6. Exibição de Sonograma e Envoltórias

Conforme a taxa de aquisição utilizada e o tamanho da FFT implementada, uma coluna de sonograma é exibida a cada 2,9ms ou 5,8ms (milissegundos). Utilizando comandos nativos do compilador Builder C++ da Borland™ ou APIs do Windows, o acesso ao vídeo é demasiadamente lento e produz oscilações da imagem (*flickers*). Inviabilizando a exibição do sonograma em tempo real.

Para acessar rapidamente o vídeo, pesquisaram-se técnicas e ferramentas, as quais: *TCanvas* (nativo do Builder C++), *OpenGL*, *DirectDraw* e *G32*.

A propriedade *Canvas* é ideal para desenvolvimentos convencionais, extremamente simples de usar e com muitos recursos para desenhos bidimensionais. Contudo, é extremamente lenta, principalmente quando é necessário acessar *pixel* individualmente.

OpenGL é uma biblioteca introduzida em 1992 pela Silicon Graphics no intuito de

conceber uma API gráfica independente do dispositivo de exibição. Estabelece uma ponte entre o processo de modelagem geométrica de objetos e as rotinas de exibição e processamento de imagens implementadas em dispositivos de hardware. Tem muitas vantagens, tais como: código aberto, boa documentação, é multiplataforma e velocidade compatível com as necessidades deste projeto. Contudo, é indicada para gráficos 3D e mostrou-se instável para aplicação no trabalho corrente (<http://www.opengl.org>).

DirectDraw (parte integrante do *DirectX*), é rápido e suporta aceleração de hardware. Contudo necessita de muita memória, tem lenta “inicialização” e depende da estabilidade do *driver* da placa de vídeo (Microsoft, 2003).

A ferramenta G32 foi escolhida neste trabalho para prover acesso rápido ao vídeo do PC. É de uso livre e utiliza códigos de máquina (*Assembly*) para acesso ao vídeo. Fornece todas as funções necessárias para exibição de sonograma em tempo real na tela do PC. Maiores informações sobre o funcionamento do G32 são encontradas em <http://www.g32.org>.

O componente G32 utilizado para exibir o sonograma é o `TPaintBox32`. Este permite o acesso para alterar cada “pixel” de forma individual, atribuindo valores RGBA (*Red, Green, Blue, Alpha* – vermelho, verde, azul e luminosidade). Após alterar o *buffer* do componente, a estrutura *Trect* é utilizada para exibir na tela a região do buffer que foi alterada.

O usuário pode escolher entre exibir sonograma em tons de cinza com fundo branco ou sonograma colorido com fundo preto. O código a seguir desenha uma coluna de sonograma no *buffer* e realiza sua exibição com o método `PaintBox321->Flush(Trect);`.

```
for (f=0; f<=FFT1->SpectrumSize; f++)
{
    y= (* (pCFFT+f));
    if (y>0xff) y=0xff; if (y<1) y=0;
    if (CBAAlpha->Checked==true) {a=2*y; if (a>0xff) a=0xff;}
    else {a=4*y; if (a>0xff) a=0xff;}
    r=y;
    g=(abs(0xff-(abs((y-0x7f)*2))));
    b=(0xff^y);
    if (y<1) {g=0; b=coreixo^0xff;}
    if (CheckBoxSCores->Checked==false) { r=r|0x1f; g=g|0x1f;
                                          b=b|0x1f; }
    if (CheckBoxFdoBco->Checked==true) {r=b; g=b; a=0xff;}
    PaintBox321->Buffer->SetPixelT(x,0xff-f,Color32(r,g,b,a));
}
```

As cores estabelecidas pelos valores RGBA são função da amplitude do sinal, onde cada *pixel* representa uma componente de frequência.

A Figura 3.20 mostra as cores assumidas no sonograma em função da intensidade da componente de frequência, resultando numa saída que varia de transparente (quando a intensidade é zero), passando por azul, verde, amarelo até chegar ao vermelho, quando a intensidade é máxima.

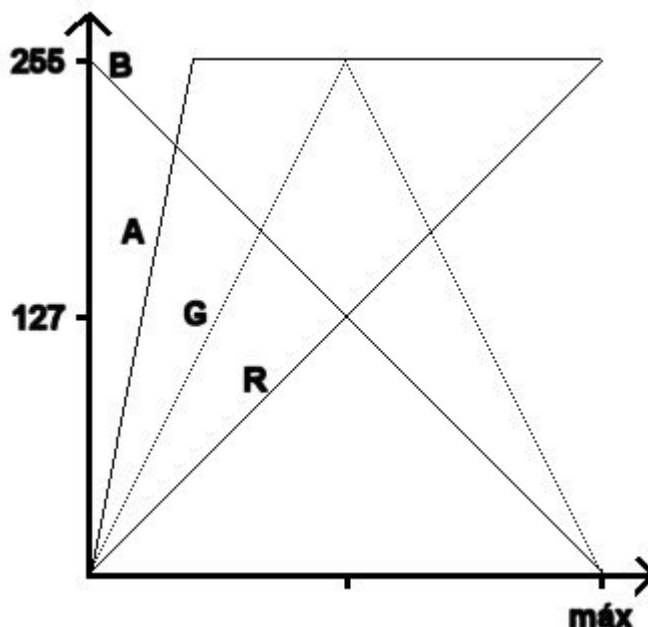


Figura 3.20 – Valores para coloração do sonograma.

O eixo das ordenadas indica o valor do *pixel*, conforme intensidade da componente de frequência que este irá representar. A intensidade está no eixo das abscissas.

Sobre o sonograma são desenhadas as envoltórias de frequência média e máxima, ponto a ponto ou interligadas como linha. Acessando o *buffer* do componente e fazendo todas as alterações neste para acessar a tela somente no momento da alteração do seu conteúdo (*refresh*) a velocidade de processamento é sensivelmente maior. Os resultados de testes de velocidade são apresentados no Capítulo 4.

3.2.7. Registro dos Sinais em Arquivo

Para diagnóstico, a exibição de sonograma em tempo real pode ser suficiente. Contudo, é importante arquivar os dados coletados para viabilizar pesquisas futuras com os dados armazenados. Com os dados dos pacientes, é possível montar um histórico e realizar análises da evolução de um determinado quadro.

Existem formatos de arquivo padronizados, para armazenar sinais de áudio

digitalizado, com ou sem compactação de dados. A Tabela 3.4 mostra alguns formatos de arquivos de áudio. Em se tratando de sinais Doppler, o algoritmo utilizado para compactação não deve admitir perdas (Bush, 1993). Um formato sem perdas, amplamente difundido, de fácil interpretação e que se mostrou eficiente na armazenagem dos dados foi o padrão PCM utilizado em arquivos *WAV*, ainda que, este padrão gere um volume relativamente grande de dados. Para uma taxa de 44100 amostras por segundo e resolução de 16 bits, utilizando os dois canais, este formato gera 176400 bytes por segundo, ou seja, aproximadamente 10Mbytes por minuto (Taylor, 1997; Poepping, 2003).

Tabela 3.4 – Formatos de arquivos de áudio.

Formato	Extensão
ACM waveform	.wav
Amiga IFF-8sVX	.iff ou .svx
Apple AIFF	.aif ou .snd
ASCII text data	.txt
Creative Sound Blaster	.voc
Dialogic ADPCM	.vox
DiamondWare Digitized	.dwd
DVI/IMA ADPCM	.wav
Microsoft ADPCM	.vox
MPEG 3 (FhG)	.mp3
mu-law wave	.wav
Next/Sun	.au ou .snd
OGG-Vorbis	ogg
PCM Raw Data	.pcm ou .ra
Real Media G2	.rm
Windows PCM	.wav

Além dos sinais Doppler, o usuário pode habilitar o registro das envoltórias. Estas são armazenadas em arquivo DAT. Além dos dados citados, o software gera um arquivo LOG contendo informações sobre a integridade do sinal gerado, permitindo verificar se os dados registrados são contínuos ou se foram obtidos em diferentes momentos. Isto pode ocorrer devido à perda de dados como consequência da falta de recursos para processamento ou por comando de pausa realizado pelo usuário (Figura 3.6 da seção 3.2.1).

O programa supre ao usuário um prontuário no campo apresentado na Figura 3.8. Dados para compor um histórico devem ser supridos para análise futura juntamente com os sinais Doppler. Estas informações são registradas em arquivo texto (extensão TXT) com o mesmo nome do arquivo *WAV* gerado.

Envoltórias de máxima e média frequência possuem resolução de oito bits e são gravadas em arquivos separados. No momento que o usuário inicia a coleta os dados são enviados *byte a byte* para o arquivo.

A estrutura do arquivo *WAV* é padrão. Tanto o *DirectX* quanto o arquivo de inclusão de APIs multimídia do Windows (`mmsystem.h`) possuem rotinas para gerar arquivos de áudio no formato *WAV*.

O cabeçalho de um arquivo *WAV* sem compressão inicia com os caracteres “RIFF” seguido pelo tamanho do resto do arquivo. Na sequência, encontram-se os caracteres “WAVE”, que indicam o tipo de arquivo. A seguir, é utilizado um identificador de blocos, indicação se há compactação, número de canais utilizados, frequência de amostragem, média de *bytes* por segundo, bits por amostra e tamanho dos dados. Em seguida, os dados são enviados para o arquivo: Um canal por vez, em *bytes*. Quando a aquisição tem resolução superior a oito bits, inicialmente são registrados os oito bits menos significativos e em seguida os oito mais significativos.

O código a seguir é um exemplo do procedimento adotado para gerar um arquivo *WAV* utilizando funções do `mmsystem.h`. Os dados contidos no buffer `pBufferDoppler` são arquivados no arquivo `doppler.wav`.

```
MMCKINFO ChunkInfo, FormatChunkInfo, DataChunkInfo;
WAVEFORMATEX WaveFormat;
WaveFormat.wFormatTag      = WAVE_FORMAT_PCM;
WaveFormat.nChannels       = 2;
WaveFormat.nSamplesPerSec  = 44100;
WaveFormat.nAvgBytesPerSec = 2*44100;
WaveFormat.nBlockAlign     = WORD((2 * 8) / 8);
WaveFormat.wBitsPerSample  = 8;
WaveFormat.cbSize          = 0;

handle = mmioOpen("doppler.wav", 0, MMIO_CREATE | MMIO_WRITE);
mmioCreateChunk(handle, &ChunkInfo, MMIO_CREATERIFF);
mmioCreateChunk(handle, &FormatChunkInfo, 0);
mmioWrite(handle, (char*)&WaveFormat, sizeof(WAVEFORMATEX) - 2);
mmioAscend(handle, &FormatChunkInfo, 0);
mmioCreateChunk(handle, &DataChunkInfo, 0);
mmioWrite(handle, pBufferDoppler, DataChunkInfo.cksize);
mmioClose(handle, 0);
```

O arquivo de inclusão `mmsystem.h` também possui funções para leitura de arquivos *WAV*. Conforme exposto no item 3.2, o usuário pode abrir arquivo *WAV* para obter o sonograma. Os dados lidos do arquivo contendo os sinais Doppler passam pelos processos indicados na Figura 3.14, a diferença é que ao contrário do processamento *on-*

line, no processamento *off-line* é calculada a quantidade de dados suficiente para desenhar uma tela de sonograma de uma só vez e não uma coluna a cada 256 amostras. O comando “*iniciar*” (Figura 3.6) passa a ter a função de exibir a próxima tela, ou seja, avança o ponteiro de leitura do arquivo *WAV*. Enquanto o comando “*pausa*” funciona como regresso, voltando à tela anterior de sonograma. A Figura 3.21 exhibe os comandos já alterados.

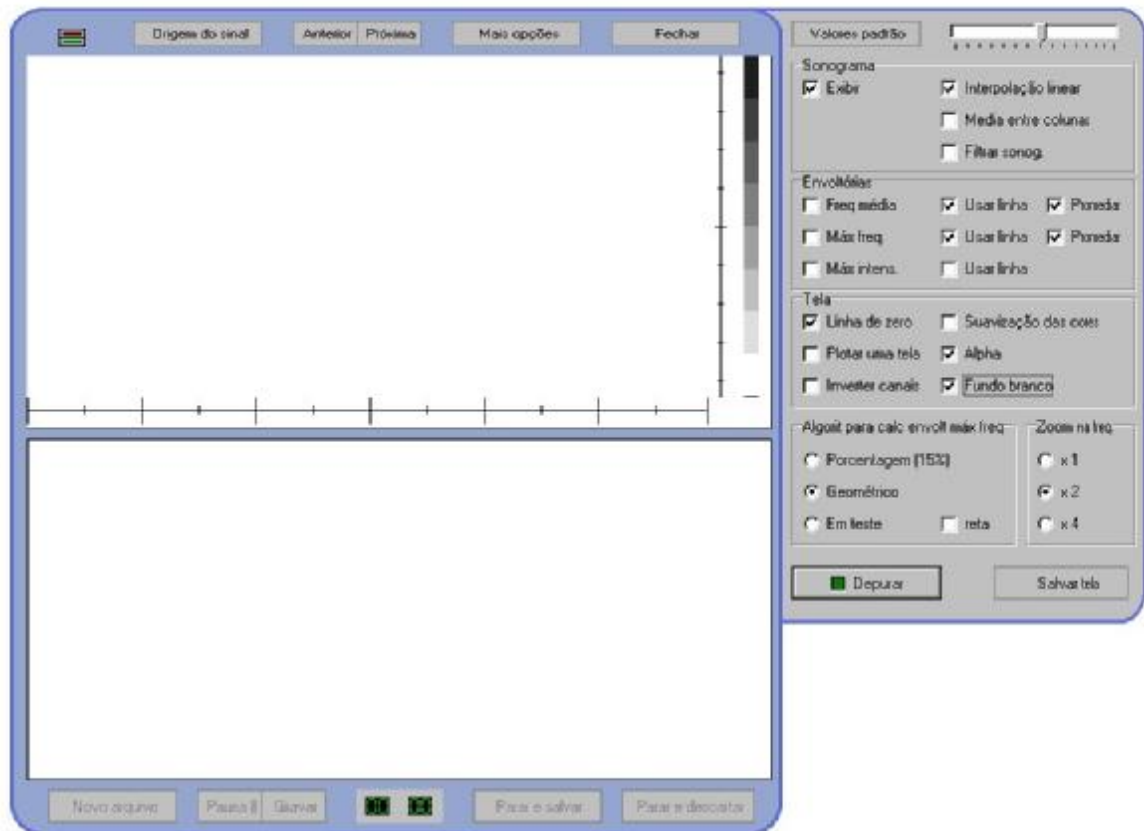


Figura 3.21 – Tela para exibição de arquivos *WAV*.

Tela principal do software quando este está voltado para exibir dados oriundos de arquivo *WAV*.

O arquivo com extensão *LOG* contém informações sobre eventos relacionados ao comportamento temporal da tarefa de registro dos sinais Doppler, avaliando a integridade dos sinais. A seguinte estrutura de mensagens é utilizada para relatar os eventos.

```

Nome do arquivo e tempo de criação.
Início de arquivo: Mlog
Término de arquivo: Mlog -> Arquivo WAV: Mwav
Pausa na aquisição: Mlog -> Arquivo WAV: Mwav
Reinício de aquisição: Mlog
Estouro de buffer, dados descartados em: Mlog -> Arq. WAV: Mwav
Falha nos dados em: Mlog
Dados gerados são confiáveis, arquivo íntegro.
Dados gerados não são confiáveis, arquivo corrompido.

```

Onde a mensagem *Mlog* indica o momento em que os eventos ocorreram, o tempo é capturado pela função `GetLocalTime()`. Enquanto a mensagem *Mwav* indica o tempo de gravação no arquivo *WAV*. Caso o volume de dados gerados seja condizente com o tempo de amostragem, o arquivo recebe aval de integridade dos dados gerados, caso contrário é salientado que os dados contem falhas.

CAPÍTULO 4

4. Resultados

Uma série de testes foi implementada para validar o sistema e obter suas características e comportamentos em diferentes cenários. Os testes estão divididos em duas partes. A primeira, verifica o processamento dos sinais Doppler em quadratura de fase; a segunda, testa o comportamento temporal de execução das tarefas. Os procedimentos dos testes e seus resultados são descritos neste capítulo.

A configuração de hardware e SO utilizados nos PCs de teste estão listadas na Tabela 4.1. Os testes foram realizados em oito PCs, tendo sido escolhidas quatro configurações para representar toda gama de testes. Foram escolhidas aquelas que apresentaram o melhor desempenho, o pior desempenho e duas intermediárias. Para ambos os PCs, a configuração de vídeo com resolução de 1024 x 768 e 16 bits para cores foi utilizada.

Tabela 4.1 – Configuração dos PCs utilizados para teste do software.

	CPU1	CPU2	CPU3	CPU4
S. O.	XP	XP	98	98
HRPC	279ns	279ns	838ns	838ns
Processador	AMD Athlon XP 1800	AMD Athlon	Pentium III	AMD K6-2
Fam./Mod./sID	6/6/2	6/4/2	6/8/10	5/8/12
Int. Clock	1500MHz	1300 MHz	1000MHz	500MHz
System BUS	266MHz	266MHz	133MHz	100MHz
L2cache	256KB	256KB	256KB	-
VGA device	GeForce2 MX/MX400	VIA Apollo KT266	GeForce2 MX/MX 400	SiS 620/530 onboard
RAM	512MB	256MB	256MB	120MB

HRPC = *high-resolution performance counter*

S. O. = Sistema Operacional

4.1. Teste de Processamento dos Sinais Doppler em Quadratura de Fase

Esta seção descreve o procedimento e os resultados obtidos ao testar o software no processamento de sinais Doppler. Não envolve exigências quanto ao cumprimento de prazos de entrega de dados.

4.1.1. Discriminação da Direção do Fluxo

A seção 3.1.2 menciona a utilização do PC como osciloscópio de áudio, permitindo a verificação da fase dos sinais adquiridos pela Placa de Som. Não houve atraso na amostragem dos dados entre os canais. Foi utilizada a montagem da Figura 3.3 para adquirir 2048 pares de amostras dos sinais em quadratura de fase. Verificou-se que as informações de fase entre os sinais se mantiveram. A Figura 4.1 exibe um trecho destes sinais em função do tempo e de um canal em função do outro para obtenção de figuras de Lissajous.

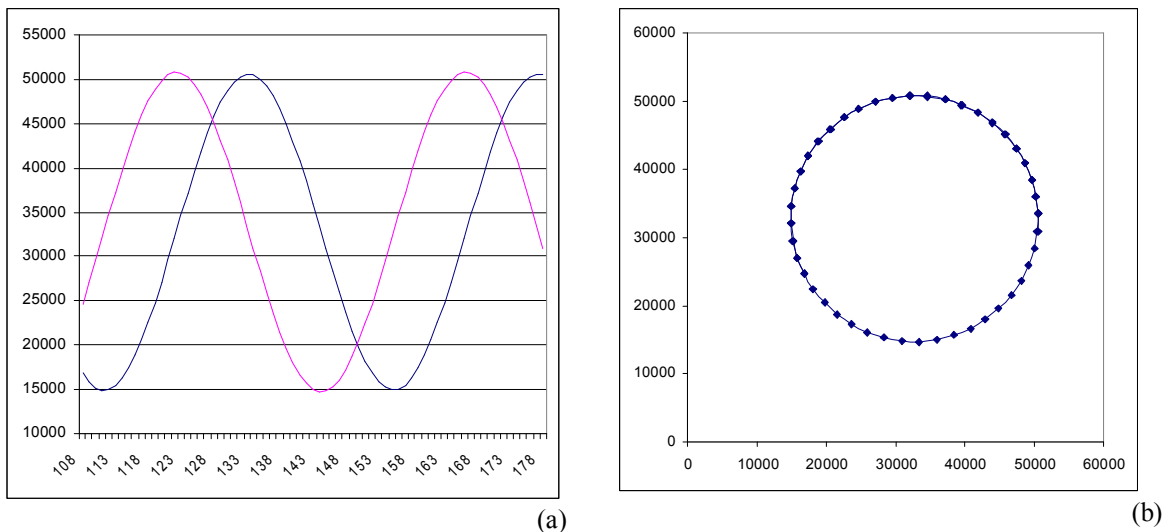


Figura 4.1 – Sinais obtidos com osciloscópio virtual.

Em (a) os sinais amostrados a partir da montagem apresentada na Figura 3.3 em função do tempo, em (b) representação por figuras de Lissajous.

A mesma montagem utilizada nestes testes, é empregada para verificar se o software permite discriminar fluxo direto de fluxo reverso, sem necessidade de correções de *crosstalk*, fase ou amplitude (Moraes, 1996; Wang, 2000).

Conforme discutido na seção 2.2.3, quando um sinal seno (com uma única componente de frequência) é aplicado em um dos canais, sendo um sinal co-seno aplicado ao outro, o espectro resultante apresenta apenas uma componente de frequência no campo das frequências positivas (Figura 4.2.a). Trocando-se os sinais aplicados aos canais, a componente de frequência será exibida apenas no espectro de frequências negativas (Figura 4.2.b).

Observa-se, na Figura 4.2, que o *crosstalk* é mínimo. Sendo que, devido ao algoritmo de exibição de sonograma, com controle de *alpha* (seção 3.2.6), o erro causado pelo *crosstalk* torna-se imperceptível (Figura 4.3).

A Figura 4.3 exibe o sonograma resultante do teste em questão. Com este teste, é possível comprovar que o software consegue recuperar informações relativas ao sentido do fluxo sanguíneo presentes nos sinais em quadratura de fase.

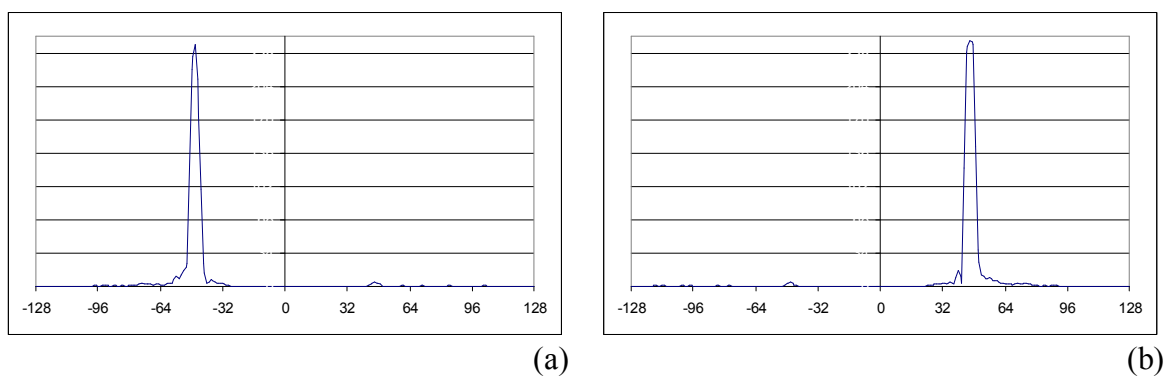


Figura 4.2 – Espectro de um sinal senoidal complexo.

Em (a) é exibida componente no campo de frequências negativas, com uma pequena componente observada no espectro positivo devido ao *crosstalk*. Em (b) o sinal apresenta a componente principal no espectro positivo.

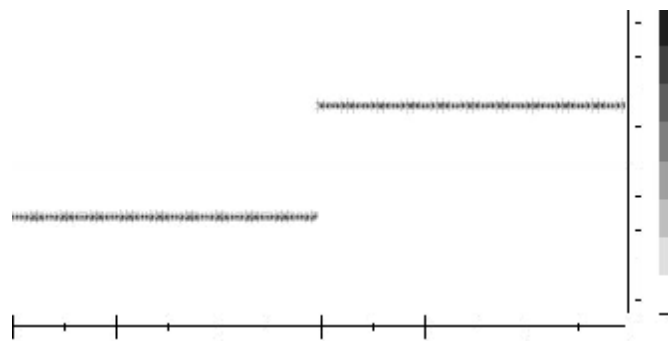


Figura 4.3 – Gráfico de um sinal senoidal complexo

(frequência no eixo das ordenadas e tempo nas abscissas).

4.2. Análise Temporal do Software

Esta seção descreve o comportamento do software implementado em função do tempo; ou seja, apresenta o tempo utilizado para realização das tarefas e a periodicidade com que os dados são processados. Caso não haja tempo suficiente para o processamento e exibição dos sinais, podem ocorrer oscilações na tela (*flicker*) ou, em situações mais severas, perda de dados.

4.2.1. Tempo de Acesso ao Vídeo do PC

A Tabela 4.2 mostra o tempo necessário para acesso da memória de vídeo do PC, utilizando o método Canvas (nativo do Builder C++) e o G32. A diferença na velocidade de execução desta tarefa é um fator crucial para a escolha do componente G32. A ausência de bibliotecas otimizadas para acesso ao vídeo inviabilizaria a exibição de sonograma em tempo real.

Tabela 4.2 – Tempo de acesso ao vídeo do PC utilizando diferentes métodos.

	Tempo necessário para exibir 20 telas de sonograma (equivale a 2.621.440 pontos na tela)	Tempo médio para exibir uma coluna (equivale a 256 pontos)
Canvas	30720 milisegundos	3 milisegundos
G32	1000 milisegundos	0,1 milisegundos

Estes testes foram obtidos com o emprego da CPU3 (Tabela 4.1). Foi implementado um software para acessar uma janela de 512x256 *pixels*, o que representa uma tela de sonograma. Esta janela foi acessada vinte vezes *pixel a pixel*, perfazendo um total de 2621440 pontos acessados. Para o acesso foram utilizados dois métodos, através de *TCanvas* e de acesso ao *buffer* (mapa *bitmap*) do componente *TPaintBox32* do G32.

4.2.2. Tempo de Cálculo da FFT

O tempo para calcular uma FFT complexa de 256 pontos com as configurações citadas na Tabela 4.1, sem outros processos em andamento, encontra-se em aproximadamente 0,3ms a 3ms. Para o teste, dez mil cálculos de FFT complexa de 256 pontos foram efetuados, demorando aproximadamente dez segundos. Se outros processos exigirem o processador, o tempo tende a aumentar.

Por estes motivos foram adotadas técnicas de programação (Capítulo 3) para garantir o desempenho desejado do sistema.

4.2.3. Comportamento das Tarefas no Tempo

Após a comprovação da viabilidade de efetuar as tarefas necessárias dentro do tempo disponível, verificou-se o tempo de execução de todas as tarefas cuja sequência de

processamento é apresentada na Figura 4.4. Lembrando que, a cada 2,9ms uma coluna de sonograma deve ser calculada (utilizando taxa de 44100Hz).

Para determinar o tempo utilizado, empregou-se função da API que permite estabelecer tempo inicial e final da execução de uma tarefa. O tempo decorrido (tempo de processamento (tp) ou tempo de computação) é obtido calculando a diferença destes valores. A resolução deste contador depende do hardware utilizado. Nos casos citados, atinge-se resoluções de 279ns ou 838ns, conforme o PC utilizado (Tabela 4.1).

No teste descrito a seguir, os sinais são amostrados a cada 2,9ms e enviados à memória do PC (taxa de 170kB/s). Um evento é gerado para requerer o processamento destes dados. Depois de encerradas estas tarefas, a *thread* de processamento (seção 3.2.2; 3.2.5 e 3.2.6) aguarda a próxima requisição. Passados 2,9ms, o processo recomeça e desta forma uma coluna de sonograma é exibida a cada período de processamento.

A sequência de tarefas está indicada no diagrama de blocos da Figura 4.4. Pormenores do processamento são descritos na seção 3.2.5.

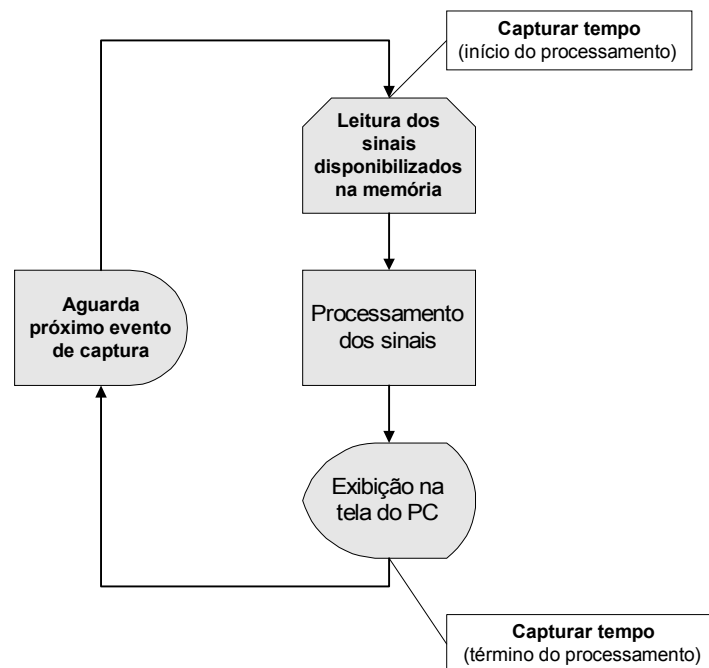


Figura 4.4 – Ciclo realizado para cada coluna de sonograma gerada.

A Figura 4.5.a mostra os intervalos em que os dados são disponibilizados para processamento, dentro dos quais as tarefas devem ser executadas. Sendo que o *deadline* de uma tarefa é o início da próxima tarefa. Na Figura 4.5.b as tarefas estão sendo executadas dentro dos prazos previstos, onde tp indica o tempo de processamento, devendo

ser inferior a 2,9ms para não ultrapassar seu *deadline*.

Em caso de morosidade no processamento, haverá atraso no início da tarefa seguinte, ou seja, quando os dados estiverem disponibilizados para início desta tarefa. São armazenados para posterior processamento. Implicando em acúmulo de dados na memória, assim que houver disponibilidade de recursos, as tarefas serão disparadas em menor intervalo de tempo, até esvaziar a memória e recuperar o sincronismo.

Se não houver folga de recursos, os dados armazenados preencherão o tamanho limite da memória e serão descartados pela *thread* de processamento. Na situação ilustrada na Figura 4.5.c, quando os dados são liberados para a *tarefa 2*, a *tarefa 1* ainda está em execução, e os dados da *tarefa 2* serão armazenados. No instante $t=23,2$ os dados pertencentes às tarefas 6 e 7 ainda estão aguardando e a *tarefa 8* também será adiada. Se houver folga de recursos, os dados estocados são enviados para processamento em seqüência, até re-estabelecer o fluxo adequado de processamento.

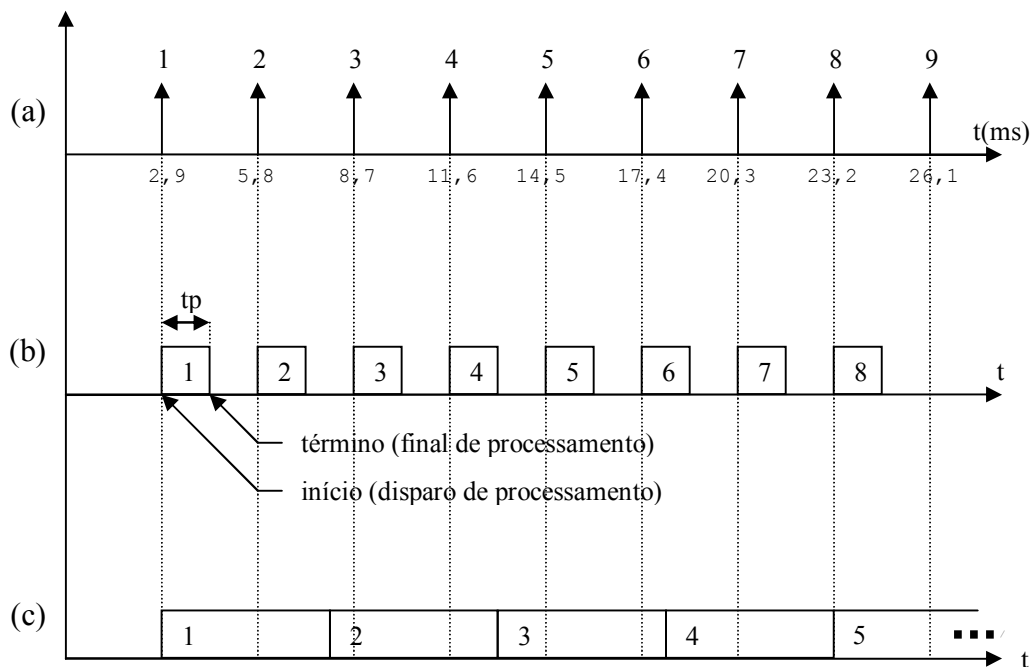


Figura 4.5 – Sequência de processamento.

Tempo de processamento das tarefas de cálculo e exibição de sonograma.

Obtendo-se os valores de início e término das tarefas, é possível verificar o tempo utilizado no processamento do pacote de sinais (tp). Se for superior a 2,9ms, não há tempo suficiente para processamento dos sinais. Neste caso, os dados serão armazenados na

memória para posterior processamento. Contudo, caso o quadro se mantenha, será necessário descartar dados; ou seja, esvaziar a memória para recuperar a eficiência do sistema.

Foi avaliado o comportamento do software em diferentes configurações de PCs e de SOs (Tabela 4.1). Para cada caso, é apresentado um gráfico com linha de tendência exibindo o tempo dedicado ao processamento. Para facilitar a visualização da distribuição dos valores é utilizado histograma, presente ao lado do gráfico mencionado. No final de cada estudo de caso, tabelas com valores de tempo mínimo, máximo, médio, desvio padrão e variância são apresentadas. Conforme dito, tolera-se um acúmulo de dados para posterior processamento, desde que este não se prolongue. Uma variância alta indica inconstância na taxa de processamento e exibição.

Os valores obtidos (expressos nos gráficos) servem para caracterizar o software e validar o sistema, disponibilizando indicações do hardware mínimo necessário para seu correto funcionamento.

No primeiro teste, foi utilizado um PC de melhor eficiência computacional, superior ao PC no qual o software foi desenvolvido. Em seguida, PCs de menores recursos computacionais foram utilizados, até se atingir uma configuração onde o software já não mais respondeu de forma adequada. Indicando assim, a configuração mínima para executar o software com segurança.

Caso 1:

Para o primeiro teste foi utilizada a CPU1 (Tabela 4.1), os resultados estão expressos em gráficos na Figura 4.6. O gráfico exibe um intervalo onde foram processadas e exibidas 512 colunas de sonograma. É possível observar a concentração dos valores, com baixa variância.

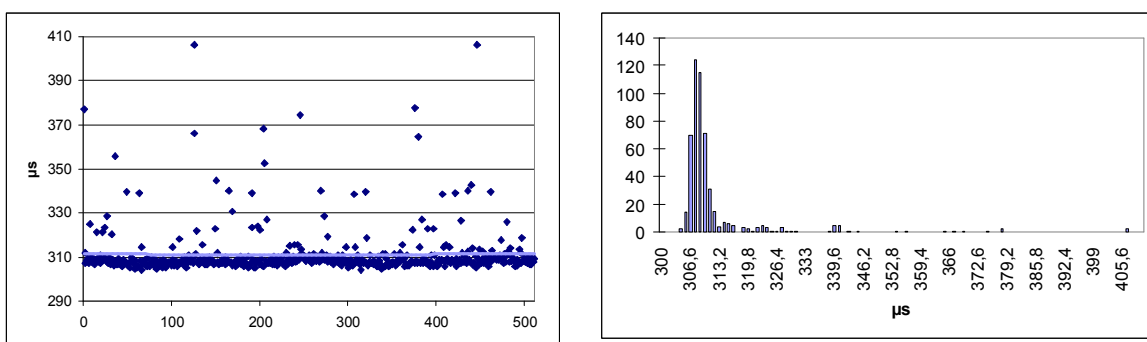


Figura 4.6 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 1º caso.

A Tabela 4.3 traz os valores estatísticos que descrevem o comportamento do software com a CPU1. O tempo máximo de 406 μ s indica que não houve acúmulo de dados na memória. A baixa variância indica que o sistema mostrou-se estável quanto ao tempo de processamento.

Tabela 4.3 – Valores obtidos a partir de testes com a CPU1.

	Mínimo	Máximo	Média	Desvio padrão	Variância
Tempo de processamento	303,9492	406,4762	311,0949	11,23604	126,002

* Todos os valores de tempo estão em microsegundos.

Neste caso, o software conseguiu processar e exibir os sinais Doppler de forma condizente com as necessidades impostas.

Caso2:

Os resultados obtidos utilizando a CPU2 estão expressos na Figura 4.7 e Tabela 4.4.

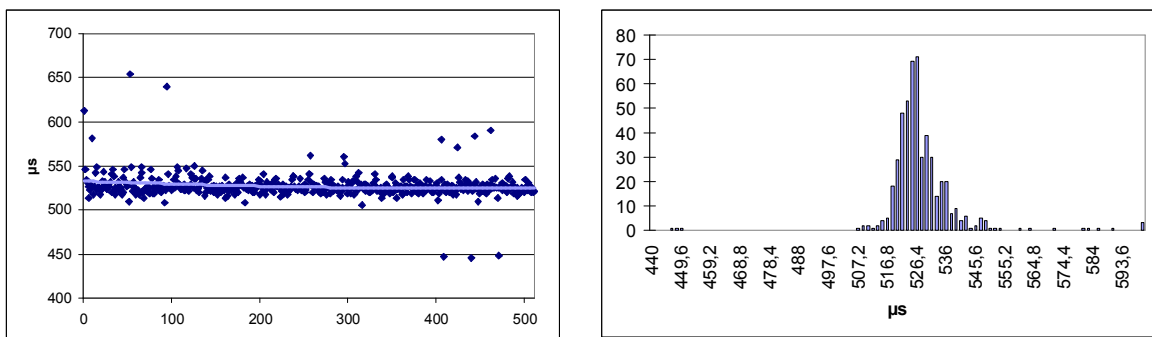


Figura 4.7 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 2º caso.

No segundo caso, houve um acréscimo significativo no tempo médio de processamento. Ainda assim, a variância e o desvio padrão são baixos, indicando boas condições de funcionamento. O software se comportou de forma estável e com folga de recursos.

Tabela 4.4 – Valores obtidos a partir de testes com a CPU2.

	Mínimo	Máximo	Média	Desvio padrão	Variância
Tempo de processamento	445,0292	653,7153	527,2282	13,71857	187,8316

* Todos os valores de tempo estão em microsegundos.

Caso 3:

O PC utilizado para desenvolvimento do software tem a configuração da CPU3. Os resultados podem ser visualizados na Figura 4.8.

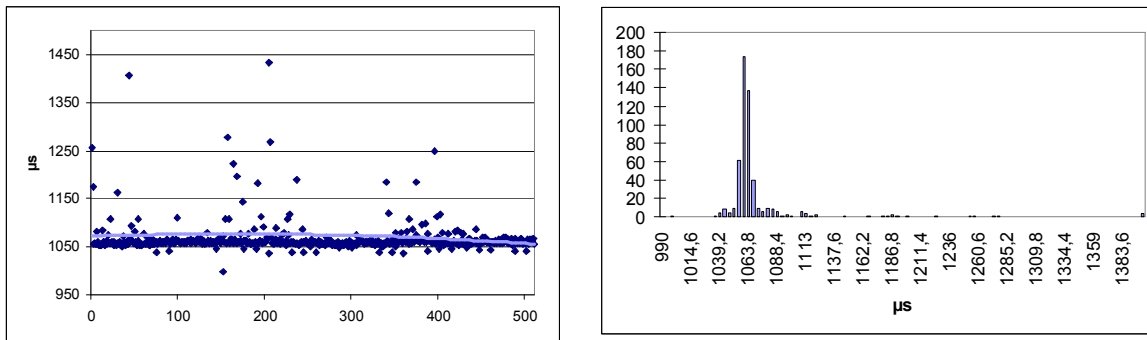


Figura 4.8 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 3º caso.

Para o terceiro caso, o software trabalhou no limite dos recursos do PC. Sem outros processos em andamento, o tempo despendido para processamento ficou, na média, em 1ms, restando aproximadamente 1,9ms de folga entre as tarefas. O valor máximo de 3ms (acima dos 2,9ms ideais) indica que, em dados momentos, houve acúmulo de dados em memória para posterior processamento. Porém, como a média é inferior a 2,9ms, o sistema consegue se recuperar. A variância elevada indica inconstância na taxa de processamento e exibição dos dados. Os valores apresentados na Tabela 4.5 indicam que, embora estando no limite, o software conseguiu atingir o desempenho mínimo necessário para seu correto funcionamento.

Tabela 4.5 – Valores obtidos a partir de testes com a CPU3.

	Mínimo	Máximo	Média	Desvio padrão	Variância
Tempo de processamento	997,3349	3158,786	1071,619	102,3056	10445,99

* Todos os valores de tempo estão em microsegundos.

Caso 4:

A última máquina utilizada para testar o software (CPU4) não conseguiu atingir o tempo mínimo reservado para processamento dos sinais. A seguir estão expostos os índices obtidos do tempo de processamento.

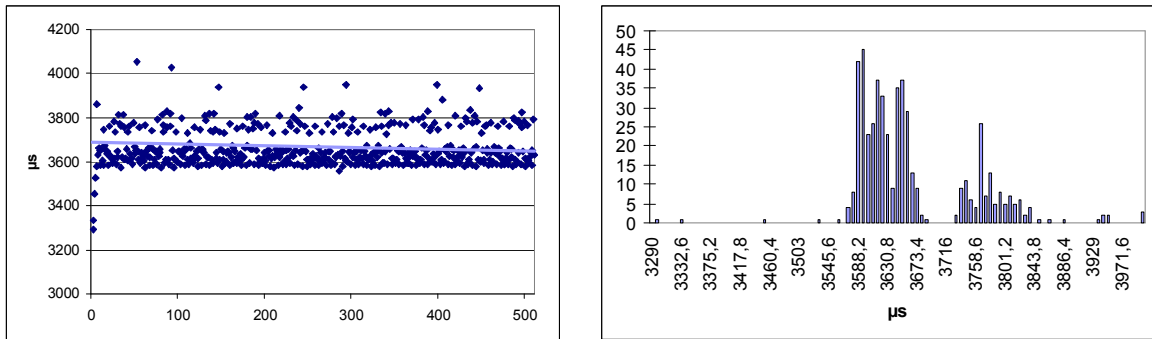


Figura 4.9 – Tempo para processamento de 512 colunas de sonograma e respectivo histograma, 4º caso.

A Tabela 4.6 exhibe valores demasiadamente elevados para viabilizar o processamento dos sinais em tempo real. O tempo médio de processamento ficou em 3,7ms, enquanto o período máximo permitido é de 2,9ms. São necessários 4,5 segundos para exibição de 512 colunas de sonograma, enquanto o ideal é um segundo e meio. Neste período, foram descartadas 1063 colunas. Portanto, a configuração da CPU3 mostrou-se inadequada para o software em questão.

Tabela 4.6 – Valores obtidos a partir de testes com a CPU4.

	Mínimo	Máximo	Média	Desvio padrão	Variância
Tempo de processamento	3291,205	7877,269	3668,47	204,562	41763,89

* Todos os valores de tempo estão em microsegundos.

A Tabela 4.7 reúne os valores obtidos nos testes descritos, caracterizando o software ao mostrar seu desempenho em diversos ambientes de execução.

Tabela 4.7 – Valores obtidos a partir de análise estatística dos tempos de computação.

		CPU1	CPU2	CPU3	CPU4
Tempo de computação (tarefa de processamento e exibição de uma coluna de sonograma)	Mínimo	303,9492	445,0292	997,3349	3291,205
	Máximo	406,4762	653,7153	3158,786	7877,269
	Média	311,0949	527,2282	1071,619	3668,47
	Desvio padrão	11,23604	13,71857	102,3056	204,562
	Variância	126,002	187,8316	10445,99	41763,89

* Todos os valores de tempo estão em microsegundos.

O intervalo entre processos é 2,9ms, assim, tempo de processamento superior a este valor provoca acúmulo de dados na memória para posterior processamento, resultando em atrasos de exibição dos sinais. Em casos extremos falta espaço na memória e há perda de dados. Na Tabela 4.7, a CPU4 apresentou um maior tempo de processamento (além do limite), indicando que esta configuração não consegue processar os sinais em tempo hábil.

A CPU3 possui configuração que a deixa no limite. Sem outros processos em andamento, esta configuração consegue apresentar características adequadas. Porém, quando outros processos requerem uso do processador, os índices caem e a eficiência do sistema é prejudicada, caso o quadro se prolongue ou se agrave. Assim, em casos ideais, sem outros processos em andamento, uma máquina com configuração da CPU3 consegue responder bem. Contudo, para haver folga de recursos e melhorar as garantias de funcionamento, é aconselhável utilizar a configuração da máquina CPU2 ou superior.

Outros testes foram realizados, como a verificação da constância na chamada das tarefas (periodicidade), para verificar se as colunas são exibidas com um ritmo constante. Contudo, em todos os casos testados estes índices não mostraram pontos passíveis de prejudicar o desempenho do software, os testes listados neste capítulo são de maior expressão e neles já estão embutidos os efeitos dos testes aqui não detalhados.

Com relação à perda de dados, há uma dependência direta com as características do hardware utilizado. Uma vez que os dados são capturados pelo *DirectSound*, o processamento se dá de forma segura. Caso ocorram falhas com as quais o software esteja preparado para lidar, o usuário é avisado e detalhes do evento são registrados em arquivo.

4.2.4. Gravando Dados no Disco Rígido

Os sinais Doppler amostrados são arquivados conforme explicado no Capítulo 3. O Windows armazena os dados na memória e, assim que houver folga de recursos, os dados são enviados ao disco rígido. O tempo de execução da tarefa está portanto, relacionado ao gerenciamento realizado pelo Windows, à taxa de dados que o disco rígido trabalha para gravação e demais recursos da máquina. Nos PCs utilizados para os teste, este valor ficou em 7MB por segundo para acesso randômico e 10MB por segundo para acesso sequencial, no pior caso. A taxa de dados gerados a partir dos sinais Doppler é de 170kB por segundo, condizente com os recursos disponíveis.

Para o teste, um pacote de dados foi enviado ao disco rígido utilizando funções de

entrada e saída de arquivos multimídia (*mmsystem.h*). Os tempos de início e término foram registrados para se obter a taxa de transferência. Esta taxa diminui quando os dados são enviados em pacotes de menor tamanho ou se o acesso ao arquivo for randômico.

Uma *thread* é utilizada para arquivar os dados. Assim, mesmo que o Windows demore a enviar os dados para o disco, o restante do processo continua a trabalhar sem atrasos. Como medida de otimização de acesso ao disco rígido, um arquivo de 10Mbytes é criado; este terá seu tamanho incrementado quando estiver 80% ocupado (tamanho dinâmico). Os dados são enviados em pacotes de 512Kbytes, o que representa meia tela de sonograma.

4.2.5. Interação com Placa Doppler

Depois de confirmada a eficiência do software em processar e arquivar os sinais Doppler, o próximo passo foi testá-lo *on-line* com hardware de geração de sinais Doppler em quadratura de fase, como proposto inicialmente. O hardware utilizado é uma Placa Doppler desenvolvida por Molina *et al* (2002).

Os resultados obtidos são exibidos na Figura 4.10, Figura 4.11 e Figura 4.12. Estas mostram a tela de sonograma obtida pelo processamento dos sinais Doppler, exibindo sonograma e envoltórias. A resolução da impressão prejudica a visualização do sonograma, mas ainda assim, é possível observar características essenciais no sonograma que indicam o estado do fluxo sanguíneo analisado.

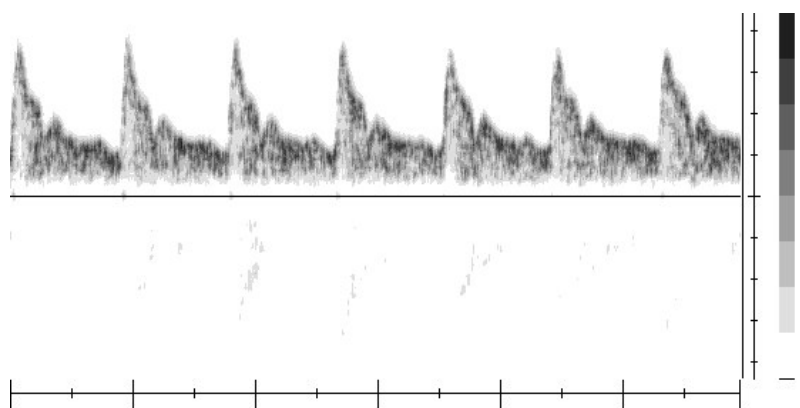


Figura 4.10 – Sonograma sem envoltórias.
(frequência no eixo das ordenadas e tempo nas abscissas)

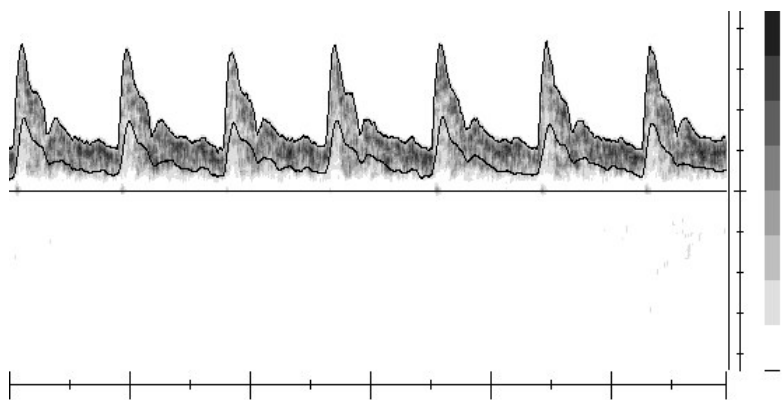


Figura 4.11 – Sonograma com envoltórias.

Envoltórias de frequência média e máxima. Obtido via Placa Doppler.
Sonograma com envoltórias (frequência no eixo das ordenadas e tempo nas abscissas).

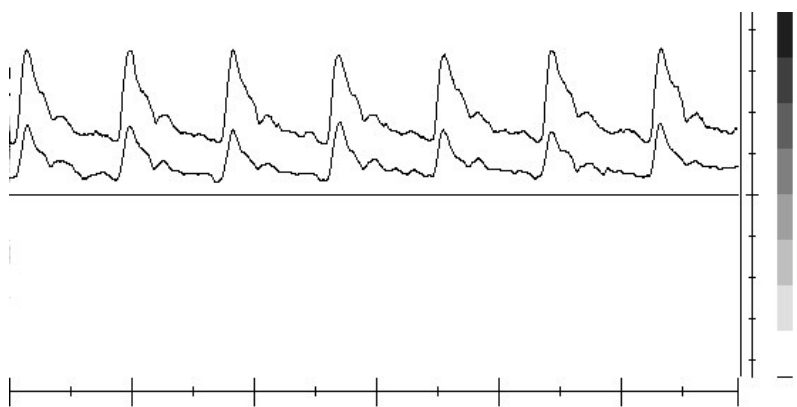


Figura 4.12 – Envoltórias de frequência máxima e média.

Sem a exibição do sonograma (frequência no eixo das ordenadas e tempo nas abscissas).

CAPÍTULO 5

5. Discussão, Conclusão e Trabalhos Futuros

O Windows é o SO (Sistema Operacional) mais popular entre os profissionais da área médica. Contudo, o ambiente multitarefa do Windows impõe dificuldades para realização de atividades em tempo real. Torna-se necessária a utilização de bibliotecas otimizadas e técnicas de programação para melhorar o desempenho, levando sua eficiência a índices aceitáveis para execução do aplicativo.

Para concluir o desenvolvimento com um sistema adequado às necessidades, foi preciso passar por diferentes fases de pesquisa e implementação de idéias, relatadas nesta dissertação.

Para implementação final do software, fez-se necessária pesquisa para avaliar as características e condições de origem dos dados, descritas no capítulo introdutório. Em seguida, utilizando técnicas de processamento digital de sinais (seção 2.2), os fenômenos físicos envolvidos (seção 2.1) foram tratados através de técnicas (Capítulo 3) para obtenção dos resultados almejados.

Conforme observado nos resultados obtidos (Capítulo 4), o Windows 98 não se constitui no SO mais adequado para trabalhar em tempo real. Seu escalonador não executa as tarefas com a frequência adequada, mesmo quando o processo tem prioridade aumentada. A versão NT do Windows possui APIs mais indicadas para gerenciamento temporal das tarefas e o seu kernel possui um algoritmo de escalonamento otimizado, em relação à versão 98 do Windows.

Contudo, respeitando uma configuração mínima requerida para executar o programa implementado, o Windows propiciou as condições necessárias para que o programa fornecesse resultados satisfatórios. Atendendo às necessidades de processamento dos sinais Doppler em tempo real brando.

A configuração mínima para executar o software implementado com segurança é um microcomputador com processador Pentium III de 1GHz ou similar, com no mínimo 128MB de memória RAM, utilizando SO Windows 98. Contudo, é indicado utilizar processador de 1.3GHz e SO Windows XP. A resolução mínima para o vídeo é 800 x 600

com 16 bits para cores, sendo aconselhável utilizar resolução de 1024 x 768 com taxa de atualização do adaptador de vídeo em 70Hz, ou superior. Resultados com diversas configurações foram obtidos e exibidos no Capítulo 4, justificando esta especificação de hardware.

A técnica de gerenciamento dos dados adquiridos e distribuídos para as demais etapas de processamento (seção 3.2.4) permite que, mesmo quando não for possível processar e exibir os sinais em tempo real, os sinais Doppler ainda podem ser enviados para HD sem perda de informações, até um certo limite de carga dos recursos do PC.

A qualidade do sonograma obtido pode ser visto na Figura 4.10. A eficiência é comprovada pelos resultados obtidos nos testes realizados. A integridade dos sinais é monitorada pelas técnicas descritas no Capítulo 3 que também relata a metodologia utilizada para chegar a tais resultados.

O SUDOC mostrou-se simples de operar em qualquer PC. O software implementado em blocos, com utilização de *threads*, permite ser modificado com facilidade, possibilitando ao pesquisador efetuar futuras alterações, conforme a necessidade.

Além dos sinais relativos ao fluxo sanguíneo, pode ser necessário aquisição e sincronismo do sinal Doppler com outros sinais, como eletroencefalograma (EEG), eletrocardiograma (ECG), nível de dióxido de carbono (CO₂), Pressão Arterial (PA), *etc.* O software tem flexibilidade para aquisição destes sinais, desde que utilizada PAS adequada.

Além do uso em pesquisa, o SUDOC também é útil no diagnóstico, permitindo a avaliação qualitativa de cirurgias de reconstrução vascular periférica. Pode servir como procedimento de triagem para exame de maior custo. Havendo indícios de complicações, o especialista pode optar por realizar outros exames como a angiografia. Caso contrário, tal exame é descartado.

Em resumo, apesar das dificuldades encontradas ao longo do desenvolvimento da pesquisa, os resultados ficaram acima do esperado. Desenvolveu-se um programa que poderá ser utilizado em trabalho futuro com sinais Doppler originados a partir do fluxo sanguíneo. As técnicas utilizadas e a metodologia empregada são relatadas para alterações necessárias a outras aplicações que requeiram tal tecnologia.

Nos últimos anos, a instrumentação virtual, um método de baixo custo e alta eficiência, tem auxiliado em diversas áreas do conhecimento. Na área da Engenharia Biomédica, possibilita aquisição e processamento dos sinais de origem fisiológica e

apresentação de dados qualitativos e quantitativos através de recursos multimídia. Recursos estes atualmente disponíveis em qualquer plataforma PC IBM compatível (microcomputador PC). Os dados obtidos são apresentados de forma simples e eficiente.

Como resultado da abordagem utilizada, a implementação do SUDOC mostrou-se de baixo custo, fator importante principalmente nos países em desenvolvimento. Conseguiu-se ainda, um sistema portátil e com tecnologia aberta, viabilizando futuras expansões e eventual reprodução do sistema. Com mais de um SUDOC em funcionamento é possível cobrir vários pacientes simultaneamente, resolvendo um dos problemas enfrentados em pesquisas que envolvem coleta de sinais *in vivo*, a falta de equipamento e a dificuldade de transporte dos mesmos. A simplicidade de operação possibilita sua utilização por pessoa com poucos conhecimentos técnicos. Permite assim, uma utilização mais ampla e portanto, facilita a monitoração de pacientes em diversas situações, como cirurgias de reconstrução vascular ou hemodiálise.

A implementação de um software que trabalhe com sinais Doppler em quadratura de fase, permite maior flexibilidade para utilização com qualquer SUDOC, dentro das limitações de taxa e resolução dos dados adquiridos. Num primeiro momento, o software foi utilizado com a Placa Doppler desenvolvida por Molina (2002).

Assim, esta dissertação abriu novas possibilidades e perspectivas para estudos do FS ao implementar um sistema aberto, robusto, de baixo custo e fácil operabilidade.

5.1. Conclusão

Da metodologia utilizada e dos resultados obtidos conclui-se que:

A eficiência das técnicas utilizadas foi comprovada através dos testes realizados. Constituindo assim, um sistema que responde com eficácia as tarefas a que se propôs resolver.

Conforme o Capítulo 4, o software implementado efetuou com sucesso as tarefas de processamento de sinais Doppler, exibindo o sonograma na tela do PC em tempo real tanto com Windows XP como com o Windows 98.

Ao utilizar os recursos existentes no próprio PC, reduz-se o custo geral do SUDOC e não havendo necessidade de componentes importados, sua reprodutibilidade é facilitada. Esta redução de custos, comparado a sistemas tradicionais, permite alocação do SUDOC nos hospitais a baixo custo, difundindo sua utilização e portanto auxiliando na

melhoria das condições da saúde.

A interação com o hardware responsável por gerar os sinais Doppler foi realizada com sucesso. Compõem assim, um sistema capaz de adquirir sinais Doppler originados a partir do FS, arquivando-os e processando-os para obtenção de sonograma e envoltórias em tempo real. Constitui-se portanto em um sistema ideal para pesquisa e análise do FS.

5.2. Trabalhos Futuros

O projeto cumpriu o que propôs. Contudo, no decorrer da pesquisa foram observados pontos a serem aperfeiçoamentos e tópicos a serem agregados ao sistema. Buscando desta forma alcançar novas funções e objetivos. São os pontos:

1 – É de grande valor arquivar informações extras com dados sobre o paciente e as condições de coleta. Utilizar banco de dados é a solução indicada;

2 – Os dados gerados ocupam um grande espaço em disco, compactar os sinais Doppler possibilita arquivar grande volume de dados. Rico em informações, os sinais Doppler podem ser utilizados não só para análise do FS, mas também para implementação e testes de algoritmos de tratamento de sinais.

3 – Pode vir a ser necessária a aquisição de outros sinais além do sinal Doppler, como ECG, EEG e outros. Utilizar PAS com canais extra, viabiliza aquisição e processamento destes sinais.

4 – Baseado no projeto desenvolvido, é possível investigar a possibilidade de construção de um sistema dedicado. Neste caso, é aconselhável utilizar sistema aberto, sendo que o SO Linux passa a ser o mais indicado.

Como último tópico sugerido, ficam em aberto as diversas possibilidades de pesquisas a serem realizadas utilizando o SUDOC implementado. Ficando a critério do pesquisador vislumbrar novas aplicações para o SUDOC.

6. GLOSSÁRIO

Buffer: Área da memória do PC usada para armazenamento temporário de dados.

DirectX: Biblioteca para acesso multimídia da Microsoft.

Estenose: Estreitamento de qualquer canal ou orifício.

Etiologia: Estudo sobre a origem das coisas. Parte da medicina que trata da causa de cada doença.

Hematócrito: Volume percentual de hemácias presente em amostra de sangue total.

Isquemia: Insuficiência localizada de irrigação sangüínea, devida a constrição ou a obstrução arterial, e que pode ocorrer em maior ou menor grau.

Kernel: Parte do Sistema Operacional que trata do hardware básico.

Luz do vaso: cavidade existente no interior de um vaso

Multithreading: capacidade de um programa executar múltiplas tarefas ao mesmo tempo. Enquanto multitarefa, é a capacidade que o SO tem de executar mais de um programa ao mesmo tempo.

Mutex – Mutual Exclusion: método de sincronização de acessos múltiplos para fontes de informação combinadas, através do mecanismo “*lock / unlock*” que tranca e abre alternadamente o acesso ao recurso compartilhado (<<http://www.britannica.com>>).

Patologia: Ramo da medicina que se ocupa da natureza e das modificações estruturais e/ou funcionais produzidas por doença no organismo.

Pixel: Unidade de informação que descreve um ponto numa imagem gráfica computadorizada. Menor ponto de luz cuja cor e luminosidade podem ser controladas.

Propedêutica: Conjunto de indagações orais e de técnicas de exame físico que serve como base a partir da qual o médico se orienta para, por investigações mais extensas, se necessário, chegar a diagnóstico.

Semiologia: Estudo e descrição dos sinais e sintomas de uma doença.

Sonograma: Gráfico no qual são apresentadas as formas de onda do fluxo sanguíneo. Contem a intensidade de cada componente de velocidade do sangue ao longo do tempo.

thread: Processo de execução de parte de um aplicativo.

Trombose: Coagulação do sangue processada, durante a vida, dentro do aparelho circulatório e em extensão e localização variáveis.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- AASLID, R. MARKWALDER TM, NORNES H. Non-Invasive Transcranial Doppler Ultrasonic Recording of Flow Velocity in Basal Cerebral Arteries. *Journal Neurosurg.* 57, 769-774, 1982.
- DIAS, A. S.; 2000. *Desenvolvendo em Borland C++ Builder™ 5.0*. Rio de Janeiro - RJ. Ed. Ciência Moderna.
- BAKER, D.W.; Pulsed Ultrasonic Doppler Blood-Flow Sensing. *IEEE Transaction on Sonics and Ultrasonics*, v.17, n.3, p170-185. 1970.
- BAZZOCCHI, M.; QUAIA, E. ZUIANI, C. *et al.* Trascranial Doppler: State of the Art. *European Journal of Radiology.* 27, s141-s148. 1998.
- BERKOWITZ, H.D. *Postoperative Screening in Peripheral Arterial Disease in Noninvasive Diagnostic Techniques in Vascular Disease.* Berstein EF, CV Mosby co, ST. Luis, p633-638, 1985.
- BOSAN, M. C.; *Estudo Da Relação Entre Os Sinais De Pressão Arterial e Velocidade do Fluxo Sanguíneo Cerebral em Recém Nascidos.* Rio de Janeiro, 1998. Tese de Doutorado – COPPE-UFRJ.
- BRUNNER, E.; *Ultrasound System Considerations and Their Impact on Front-End Components.* Analog Devices, Inc. 2002.
- BUSH, G.; EVANS, D.H.; Digital Audio Tape as a Method of Storing Doppler Ultrasound Signals. *Physiol. Meas.* 14, p381-386. 1993.
- CARVALHO, M.H.C.; NIGRO, D.; LEMOS, V.S. *et al.* Hipertensão arterial: o endotélio e suas múltiplas funções, *Revista Brasileira de Hipertensão*, 8, 1, 76-88, 2001.
- Carvalho, M.H.C.; Nigro, D.; Lemos, V.S. *et al.* Hipertensão arterial: o endotélio e suas múltiplas funções, *Revista Brasileira de Hipertensão*, 8, 1, 76-88, 2001.
- MATEUS, C.A.; 2000. *C++ Builder™ 5 Guia Prático*. São Paulo - SP. Ed. Érica.
- C-MEDIA; *CMI-8738 Series Audio Chip Specification v1.5.* Disponível em: <<http://www.cmedia.com.tw>>. Acesso em: abril de 2003.
- CUNEO, J.; *Cirurgia Vascular na Isquemia Arterial Crônica.* Disponível em: <<http://bibliomed.com.br>> Acesso em: abril de 2003.
- DIEHL, R. R.; Cerebral Autoregulation Studies in Clinical Practice. *European Journal of Ultrasound.* 16 : 31-36. 2002.
- DOUGHERTY, M.J.; CALLIGARO, K.D. DELAURENTIS, D.A. *The Natural History of*

- “Failing” Arterial Bypass Grafts in a Duplex Surveillance Protocol. *Ann Vasc, Surg*, v.2, n.3, p.255-259, 1998.
- ERICKSON, C.A.; TOWNE, J.B.; SEABROOK, G.R.; *et al.* Outgoing Vascular Laboratory Surveillance is Essential to Maximize long-term in Situ Saphenous Vein Bypass Patency. *Journal Vasc. Surg.*, n.23, p18-27, 1996.
- EVANS, D.H.; SCHLINDWEIN, F.S.; LEVENE, M.; *An Automatic System for Capturing and Processing Ultrasonic Doppler Signals and Blood Pressure Signals*. *Clin. Phys. Physiol.*, 1989, v.10, n.3, p241-251.
- FARINES, J.M.; FRAGA, J. S.; OLIVEIRA, R.S.; *Sistemas de Tempo Real. Escola de Computação*. 2000, IME – USP, São Paulo – SP.
- FOPPA, M.; FUCHS, F.D.; DUNCAN, B.B. *Álcool e Doença Aterosclerótica*, *Arq. Bras. Cardiol.*, Porto Alegre, 76, 2, 165-170, 2001.
- GÜLER, I.; HARDALAÇ, F.; BARIŞÇ; Application of FFT analyzed Cardiac Doppler Signals to Fuzzy Algorithm. *Computers in Biology and Medicine* 32. p.435-444. 2002.
- GÜLER, I.; HARDALAÇ, F.; KAYMAZ, M.; Comparason of FFT and Adaptative ARMA Methods in Transcranial Doppler Signals Recorded from the Cerebral Vessels. *Computers in Biology and Medicine* 32. p445-453. 2002.
- GÜLER, I.; HARDALAÇ, F.; MÜLDÜR, S.; Determination of aorta failure with the application of FFT, AR and wavelet methods to Doppler technique. *Computers in Biology and Medicine* 31. p229-238. 2001.
- GÜLER, I.; HARDALAÇ, F.; ÜBEYLI, E.D.; Determination of Behcet Disease with the Application of FFT and AR Mothods. *Computers in Biology and Medicine* 32. p419-434. 2002.
- SCHILDT, H.; GUNTLE, G.; 2001. *Borland C++ Builder™, a referência completa*. Rio de Janeiro. Ed. Campus.
- HILL, C.R.; *Physical Principles of Medical Ultrasonics*. Ellis Horwwd Limited, England. 1986.
- HOLLINGWORTH, J.; BUTTERFIELD, D.; SWART, B. *et al.* 2000. *C++ Builder™ 5 Developer’s Guide*. USA. Ed. Sams.
- INTEL; *Audio Codec ’97 Component Specification rev. 2.2*. Sept. 2000. Disponível em: <<http://www.intel.com>>. Acesso em: abril de 2003.
- INTEL; MICROSOFT. *PC 2001 System Design Guide*. 2000. Disponível em: <<http://www.pcdesguide.org>>. Acesso em abril de 2003.
- INTEL; MICROSOFT; COMPAQ. *PC 98 System Design Guide*. 1997. Disponível em: <<http://www.pcdesguide.org>>. Acesso em abril de 2003.
- IPSIROGLU, O. S.; KUHLE, S.; KÖHLER, J. *et al.* Effects of Intubation on CBFV in

- Infants with Compromised Cerebrovascular Autoregulation, *European Journal of Ultrasound*, Volume 5, Supplement 1, S58-S59. 1997.
- JOHNSON, A.P.; MACAULEY, M.W.S.; High Precision Timing Within Microsoft Windows: Thread, Scheduling and System Interrupts. *Microprocessors and Microsystems*. 25, p297-307. 2001.
- KADAR, A.; GLASZ, T. Development of atherosclerosis and plaque biology. *Cardiovascular Surgery*, Great Britain, 9, 2, 109-121, 2001.
- KEETON, P.I.J.; SCHLINDWEIN, F.S.; EVANS, D.H.; A Study of the Spectral Broadening of Simulated Doppler Signals Using FFT and AR Modeling. *Ultrasound in Med. & Biol.* v.23, n.7, p1033-1045. 1997.
- KEETON, P.I.J.; SCHLINDWEIN, F.S.; Spectral Broadening of Clinical Doppler Signals Using FFT and Autoregressive Modeling. *European Journal of Ultrasound* 7, p209-218. 1998.
- LAMOTHE, A.; 1999. *Tricks of the Windows Game Programming Gurus: Fundamentals of 2D and 3D Game Programming*. USA. Ed. Sams.
- LEÃO, M.; 1998. *Introdução ao Borland C++ Builder*. Rio de Janeiro – RJ. Ed. Axcel Books do Brasil.
- LIBBY, P.; Inflammation in Atherosclerosis. *Nature*. vol.420. 2002.
- MALDONADO, F.J.G.; PERALTA, A.G.; CAMPUBRI, A.O. *et al.* Estudo da Validade do Doppler Contínuo. *Rev. Neurol.* v.29, n.10, p980. 1999.
- MARASEK, K.; NOWICKI, A.; Comparasion of the Performance of Three Maximum Doppler Frequency Estimators Coupled With Different Spectral Estimation Methods. *Ultrasound in Med. & Biol.*, v.20, n.7, p.629-638, 1994.
- MARCELO LEÃO. 1998. *Introdução ao Borland C++ Builder™*. Rio de Janeiro - RJ. Ed. Axcel Books.
- MICROSOFT; *MSDN – Microsoft Development Network*. 2003. Disponível em: <<http://www.msdn.microsoft.com>>. Acesso em abril de 2003.
- MINISTÉRIO DA SAÚDE. *A Saúde no Brasil: estatísticas essenciais 1990 – 2000 / Ministério da Saúde, Secretaria Executiva*. Brasília: Ministério da Saúde, 2002.
- MO, L.Y.L; YUN, L.C.M.; COBBOLD, R.S.C.; Comparasion of Four Digital Maximum Frequency Estimators for Doppler Ultrasound. *Ultrasound in Med. & Biol.*, v.14, n.5, p355-363, 1988.
- MOLINA, P. S. C.; MORAES, R. *Sistema de Ultra-som Doppler de Ondas Contínuas para Investigação do Sistema Vascular Periférico*. In: CONGRESSO BRASILEIRO DE ENGENHARIA BIOMÉDICA. (18 : set. 2002 : São José dos Campos – SP) *Anais de resumos*. São José dos Campos : Univap, 2002. 225.

- MORAES, R. *Sistema Microcomputadorizado Dedicado para a Obtenção de Parâmetros de Análise da Forma de Onda da Velocidade do Fluxo Sanguíneo*. Campinas, São Paulo. 1990. Dissertação de Mestrado em Engenharia Elétrica. Departamento de Engenharia Biomédica. Universidade Estadual de Campinas.
- MORAES, R.; AYDIN, N.; EVANS, D.H.; The Performance of Three Maximum Frequency Envelope Detection Algorithms for Doppler Signals. *Journal of Vascular Investigations* 1:3 p126-134. 1995.
- MORAES, R.; EVANS, D.H.; Compensation for phase and amplitude imbalance in quadrature doppler signals. *Ultrasound in Medicine & Biology*, Volume 22, Issue 1, 1996, Pages 129-13.
- MORAES, R. *Doppler Catheter measurements of blood flow velocity in coronary arteries*. 1995, U.K. Division of Medical Physics, Faculty of Medicine. University of Leicester.
- NAKAJIMA, K.; KUDO, N; YAMAMOTO, K. *et al. A study on Frequency Dependence of Ultrasound Attenuation of Biological Tissue in the Frequency Range of 2-40MHz*. IEEE Ultrasonics Symposium, p1381-1384. 1999.
- NETO, A. A.; *Velocidade de Fluxo Sanguíneo Cerebral, por Ultra-sonografia Doppler no Primeiro Dia de Vida de Recém Nascidos Policitemicos*. São Paulo, 1998. Tese de Doutorado – UNICAMP.
- NIZAMETTIN, A.; *Computerised Graft Monitoring*. U.K., 1994. Tese de Doutorado – Division of Medical Physics, Faculty of Medicine, University of Leicester.
- NIZAMETTIN, A.; EVANS, D. H.; Implementation of Directional Doppler Techniques Using a Digital Signal Processor. *Elec-trocardiography, Myocardial Contraction and Blood Flow Supplement. Med & Biol Eng & Compt* 1994; 32: s157-s164.
- NIZAMETTIN, A.; EVANS, D.H.; *A Fast Frequency Domain Algorithm For Decoding Quadrature Doppler Signals*. 18th Annual Conference of IEEE Engineering in Medicine and Biology Society, Amsterdam 1996, pp 993-994.
- NIZAMETTIN, A.; LINGKE, F.; EVANS, D.H.; *Quadrature-to-directional format conversion of Doppler signals using digital methods*. *Physiol. Meas.* No. 15, 1994; 181-199, IOP Publishing Ltd.
- NOGUCHI Y, KASHIWAGI E, WATANABE K, MATSUMOTO F, SUGIMOTO S.; 2001. *Time-frequency analysis of the blood flow Doppler ultrasound signal*. Japão. Dept. Appl. Physical. Yokosuka.
- OLIVEIRA, R.S.; CARISSIMI, A.S.; TOSCANI, S.S.; *Organização de Sistemas Operacionais Convencionais e de Tempo Real*. XXII Congresso da Sociedade Brasileira de Computação. Florianópolis – SC. 2002.
- OLUFSEN, M.; NADIM, A.; LIPSITZ, L. Bioengineering Conference, 2000. *Proceedings of the IEEE 26th Annual Northeast*, 2000 : 41-42.
- OPPENHEIM, A. V.; WILLSKY, A. S., 1983. *Signal and Systems*. London: Prentice-Hall

International.

- PANERAI, R. B.; Assessment of Cerebral Pressure Autoregulation in Humans – a Review of Measurements Methods. *Physiol . Meas.* v19 : n3, 305-338. 1998.
- PANERAI, R. B.; KELSALL, A.W.R.; EVANS, D. H.; *Analysis of Cerebral Blood Flow Autoregulation in Neonates.* IEEE Transactions on Engineering, v43, n8. 1996.
- PAUM, M.; BEACH, K.; AHMAD, S. *et al.* New Ultrasound Approaches to Dialysis Access Monitoring. *American Journal of Kidney Diseases*, v.35, n.3, p.477-481. 2000.
- PEREIRA, M.C.V.; AZEVEDO, F.M. *Análise para um Janelamento Adequado para Sinais Bioelétricos.* In: XVIII CONGRESSO BRASILEIRO DE ENGENHARIA BIOMÉDICA (18. : set. 2002 : São José dos Campos, São Paulo). Anais. São Paulo, 2002. p. 323-328.
- RIDGE, P.M.; GOLDEN, D.M.; LUK, I. *et al.* 1995. *Sound Blaster: O Livro Oficial.* São Paulo – SP. Ed. Makron.
- PLASENCIA, J.L.M.; *Projeto de um Sistema Ultra-sônico Doppler Pulsátil Multicanal para Avaliação da Vascularização Periférica.* Campinas, São Paulo, 1994. Dissertação de Mestrado – Departamento de Engenharia Biomédica, UNICAMP.
- POEPPING, T.L.; GILL, J.; FENSTER, A. *et al.* MP3 compression of Doppler ultrasound signals. *Ultrasound in Medicine & Biology*, Volume 29, Issue 1, January 2003, Pages 65-76.
- PRESS, W.H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY B. P. *Numerical Recipes in C : The Art of Scientific Computing.* Second Edition. USA : Cambridge University Press, reprinted in 2002.
- RAMAMRITHAM, K; SHENT, C.; GONZALEZ, O. *et al.*; *Using Windows NT for Real-Time Applications: Experimental Observations and Recommendations.* IEEE Real-Time Technology and Applications Symposium. 1998.
- RAMOS E.G.; NADAL, J.; VILLAR, M.N. *et al.* *Comparação Entre Técnicas de Avaliação da Auto-Regulação do Fluxo Sanguíneo Cerebral de Neonatos.* in: CONGRESSO BRASILEIRO DE ENGENHARIA BIOMÉDICA. (18 : set. 2002 : São José dos Campos – SP) *Anais.* São José dos Campos : Univap, 2002.
- RICKEY, D.W.; *Evaluation of Doppler Ultrasound Velocity and Flow Measurements.* London, U.K. Department of Medical Biophysics. University Western Ontario, 1995.
- ROSS, R. Atherosclerosis – An Inflammatory Disease. *The New England Journal of Medicine*, Boston – USA, 1999.
- SALDIVAR, I.D.L; 1997. *Processador Multimídia de sons cardíacos.* Florianópolis. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica – Universidade Federal de Santa Catarina.
- SCHUCHTER, K.; METZENBAUER, M.; HAFNER, E.; *et al.* 2001. *Uterine artery*

- Doppler and placental volume in the first trimester in the prediction of pregnancy complications.* Vienna, Austria. Dept Obstet & Gynecol, Langobardenstr.
- SEITZ, J.; STROTZER, M.; WILD, T.; *et al.* 2001. *Quantification of blood flow in the carotid arteries - Comparison of Doppler ultrasound and three different phase-contrast magnetic resonance imaging sequences.* Alemanha. University Hospital Regensburg.
- SMITH, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing.* Segunda edição. San Diego – USA : California Technical Publishing, 1999.
- STEINMAN, A.H.; TAVAKKOLI, J.; MEYERS, J.G. *et al.* *A New Approach for Determining Maximum Frequency in Clinical Doppler Ultrasound Spectral Estimates.* 22nd Annual EMBS International Conference, Chicago. p23-28, 2000.
- STEPT, L.L.; FLINN, W.R.; McCARTHY, W.J.; *et al.* *Technical Defects as a Cause of Early Graft Failure After Femorodistal Bypass.* Arch. Surg., n.122, p.599-604, 1987.
- STEVEN, M.R.; DENNIS, F.B.; BRAD, L.J. *at al.* *Bypass Graft Surveillance.* Disponível em: <<http://www.med.usf.edu/surgery/vascular/topicintro.htm>> Acesso em: março de 2003.
- TAKASE, K.; KASHIHARA, M.; HASHIMOTO, T.; *Transcranial Doppler Ultrasonography in Patients With Moyamoya Disease.* *Clinical Neurology and Neurosurgery*, Volume 99, Supplement 2, S100-S104. 1997.
- TATIBANA, C.Y.; *Estudo do Linux como Plataforma para Aplicações de Tempo Real Brando.* Florianópolis – SC, 2002. Dissertação de Mestrado. PPGEEL, UFSC.
- TAYLOR, J.; PULL, C.J.; GILL, B.R. *et al.* *Data Compression of Fetal Doppler Ultrasound Audio Signals Using Zero-Crossings Analysis.* *Med. Eng. Phys.* v.19, p.572-580. 1997.
- TOMPKINS, W.J.; *Biomedical Digital Signal Processing.* Prentice-Hall, 1995.
- TORTOLI, P.; GUIDE, G.; NEWHOUSE, V.L.; *Improved Blood Velocity Estimation Using the Maximum Doppler Frequency.* *Ultrasound in Med. & Biol.* v.21, n.4, p527-532, 1995.
- VERMA, P. K.; PANERAI, R.B.; EVANS, D. H. *Grading of Cerebral Autoregulation in Preterm and Term Neonates.* *Pediatric Neurology.* 23: 3. 2000.
- VERSTRAETE, M.; Vermynen, J. *Trombose.* São Paulo – Brasil: Editora Sarvier, 1989.
- WANG, Y.; GAO, S.; *The Software Method for Cross-talk Correction in Ultrasound Doppler Blood Flow Measurement.* *Journal of Biomedical Engineering*, v17, i3, p301-304, 2000.
- WATKINSON, J.; *An Introduction to Digital Audio.* Clays Ltd., 1995.
- WATSON, H.R.; BUTH, J.; SCHROEDER, T. V.; *et al.* *Incidence of Stenoses in Femorodistal Bypass Vein Graft in a Multicentre Study.* *Eur. J. Vasc. Endovasc. Surg.*

v.20, n.1, p67-71, jul. 2000.

WEBSTER, J.G.; CLARK, J.W. *Medical Instrumentation: Application and Design*. 2^oed. Boston : H. Mifflin, 1992.

WEITZEL, W.F.; RUBIN, J.M.; LEAVEY, S.F. *et al.* Analysis of Variable Flow Doppler Hemodialysis Access Flow Measurements and Comparison With Ultrasound Dilution. *American Journal of Kidney Diseases*, v.38, n.5, p.935-940. 2001.

WHITTEMORE, A.D.; CLOWES, A.W.; COUCH, N.P.; *et al.* *Secondary Femoropopliteal Reconstruction*. *Ann. Surg.*, n.1, p.35-42, 1981.

XING, Z. W.; LU, Z. C.; CHEN, G. X. *et al.* *Cerebral Blood Flow Velocity by Transcranial Doppler During a Vertical-Rotating Table Simulation of the Push-Pull Effect*. *Aviat. Space Environ. Med.* 71 : 485-8. 2000.