



Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

**Desenvolvimento de um Sistema de Aquisição e  
Processamento de Dados de Baixo Custo para Análise  
de Sinais Elétricos em Sistemas Trifásicos**

Fagner de Araujo Pereira

Orientador: Prof. Francisco das Chagas Fernandes Guerra, Dr.

novembro de 2009



Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

**Desenvolvimento de um Sistema de Aquisição e  
Processamento de Dados de Baixo Custo para Análise  
de Sinais Elétricos em Sistemas Trifásicos**

Relatório apresentado à Coordenação de Graduação em  
Engenharia Elétrica da UFCG como parte dos requisitos  
para obtenção do título de Engenheiro Eletricista.

---

Fagner de Araujo Pereira  
Aluno

---

Prof. Francisco das Chagas Fernandes Guerra, Dr.  
Orientador

novembro de 2009

# Agradecimentos

Agradeço ao Senhor Deus por me dar forças, determinação e abençoar minha vida com graças incontáveis. Obrigado Senhor Jesus Cristo.

A minha família: minha mãe, Maria do Socorro Pereira, que sempre me incentivou e que me forneceu meios de me dedicar aos estudos. Aos meus avós, Inácio José de Sousa e Severina Faustulino de Sousa, que acolheram a mim, minha mãe e meus irmãos em sua casa durante toda a vida, dando-nos carinho e amor. Aos meus irmãos. Todos vocês são responsáveis por essa vitória.

A minha namorada, Erica Manuela do Nascimento, que me auxiliou em muitas etapas da minha vida acadêmica e que compreendeu minhas ausências.

Aos meus familiares, amigos e colegas de trabalho que sempre acreditaram na minha capacidade, e que me apoiaram em meus estudos.

Ao professor Chagas, que acreditou no meu trabalho e me orientou neste projeto. Pela sua paciência e pela sua acessibilidade.

# Sumário

Sumário.....	4
Lista de Figuras.....	5
Lista de Tabelas.....	7
1. Introdução.....	8
1.1 Considerações Gerais.....	8
1.2 Objetivos.....	9
1.3 Metodologia.....	9
2. Aquisição e Processamento de Dados.....	10
2.1 Medição e Instrumentação.....	10
2.2 Sensores e/ou Transdutores.....	11
2.3 Sinais e Condicionamento de Sinais.....	12
2.4 Conversão Analógico-Digital.....	15
2.5 Processamento de Sinais.....	20
3. Descrição do Medidor Virtual.....	21
3.1 Especificação do Sistema Proposto.....	21
3.2 Caracterização do Sistema Proposto.....	26
3.3 Justificativa do Sistema Proposto.....	32
4. Desenvolvimento do <i>Software</i> .....	33
4.1 A Linguagem <i>Matlab</i> .....	33
4.2 Osciloscópio no <i>MatlabScope</i> .....	36
4.3 Analisador de Espectro no <i>MatlabScope</i> .....	44
4.4 Gravador de Transientes no <i>MatlabScope</i> .....	49
4.5 Plano de Fase no <i>MatlabScope</i> .....	52
5. Teste do Sistema Implementado.....	54
6. Comparação com Outros Sistemas de Medição.....	61
7. Conclusão.....	64
8. Bibliografia.....	65

# Lista de Figuras

Fig. 2.1 Elementos essenciais de um sistema de aquisição de dados.....	11
Fig. 2.2 Sensores para aplicação no sistema proposto.....	11
Fig. 2.3 Tipos de sinal.....	13
Fig. 2.4 Espectro de frequência de um sinal analógico.....	14
Fig. 2.5 Espectro de frequência de um sinal analógico amostrado com uma frequência $f_s$ .....	14
Fig. 2.6 Espectro de frequência de um sinal analógico amostrado com uma frequência $f_s < 2B$ .....	14
Fig. 2.7 Aliasing de um sinal no domínio do tempo.....	15
Fig. 2.8 Amostragem de sinais.....	16
Fig. 2.9 Efeito da amostragem de um sinal analógico de banda limitada em $W$ a uma taxa de $T_s = 1/f_s$ , resultando na repetição da banda de frequência a cada $2\pi$ radianos.....	17
Fig. 2.10 Níveis de quantização.....	17
Fig. 2.11 Entrada/saída e erro de quantização de um sinal analógico Quantizado por um conversor A/D de 3 bits.....	18
Fig. 2.12 Formas de onda presentes no processo de conversão.....	19
Fig. 3.1 Representação do sistema proposto.....	21
Fig. 3.2 Placa de Som Delta 1010LT, usada para aquisição de dados do sistema proposto.....	22
Fig. 3.3 Conexão dos cabos analógico e digital à placa.....	23
Fig. 3.4 Interface principal do <i>MatlabScope</i> .....	24
Fig. 3.5 Interface do <i>Spectrum Analyzer</i> .....	25
Fig. 3.6 Interface do <i>Transient Recorder</i> .....	25
Fig. 3.7 Sistema implementado.....	26
Fig. 3.8 Interface do driver da placa Delta1010LT.....	26
Fig. 3.9 Aba <i>Hardware Settings</i> da interface do drive da placa Delta 1010LT.....	27
Fig. 3.10 Curva característica da placa Delta 1010LT para frequências de 1 Hz a 60 Hz.....	29
Fig. 3.11 Exemplo de circuito condicionador para sinais de tensão.....	31
Fig. 3.12 Fluxograma de aquisição dos dados.....	31
Fig. 4.1 Ambiente de trabalho do <i>Matlab 6.5</i> .....	33
Fig. 4.2 Tela do <i>Guide</i> .....	34
Fig. 4.3 Uma simples interface gráfica criada no <i>Guide</i> .....	35
Fig. 4.4 Interface Principal do <i>MatlabScope</i> .....	36
Fig. 4.5 Detalhe das medidas verticais exibidas para um sinal de tensão.....	37

Fig. 4.6 Detalhe das medidas horizontais exibidas para um sinal de corrente.....	38
Fig. 4.7 Indicadores de operação matemática.....	38
Fig. 4.8 Indicadores dos sliders <i>Vertical Scale</i> e <i>Time/div</i> .....	40
Fig. 4.9 Tela de expansão.....	41
Fig. 4.10 Submenus da barra de menus do <i>MatlabScope</i> .....	42
Fig. 4.11 Módulo de salvamento de dados.....	42
Fig. 4.12 Arquivo de imagem obtido do submenu <i>Save Screen as</i> .....	43
Fig. 4.13 Caixa de diálogo de ajuste dos níveis dos sensores usados no sistema.....	44
Fig. 4.14 Janela com informações sobre o desenvolvimento do <i>MatlabScope</i> .....	44
Fig. 4.15 Interface do analisador de espectro.....	45
Fig. 4.16 Espectro de uma senóide pura de frequência $f_1$ .....	46
Fig. 4.17 Módulo de salvamento de dados do <i>Spectrum Analyzer</i> .....	48
Fig. 4.18 Arquivo de imagem obtido do submenu <i>Save Screen as...</i> do <i>Spectrum Analyzer</i> .....	48
Fig. 4.19 Interface do <i>Transient Recorder</i> .....	49
Fig. 4.20 Arquivo de imagem obtido do submenu <i>Save Screen as...</i> do <i>Transient Recorder</i> .....	51
Fig. 4.21 Interface do <i>Phase Plane</i> .....	52
Fig. 4.22 Arquivo de imagem obtido do submenu <i>Save Screen as...</i> do <i>Phase Plane</i> .....	53
Fig. 5.1 Montagem realizada para teste do sistema de aquisição.....	54
Fig. 5.2 Medidas de tensão no secundário e corrente no primário de TR1 na situação em que a alimentação equivale a 90% do valor de tensão nominal.....	55
Fig. 5.2 Medidas de tensão no secundário e corrente no primário de TR1 na situação de tensão nominal.....	56
Fig. 5.2 Medidas de tensão no secundário e corrente no primário de TR1 na situação em que a alimentação equivale a 110% do valor de tensão nominal.....	57
Fig. 5.5 Espectro de frequências dos sinais de tensão e corrente na fase A para a condição de tensão nominal.....	58
Fig. 5.6 Curva de histerese do transformador TR1 para a condição de tensão nominal.....	59
Fig. 5.7 Sinais de tensão e corrente capturados pelo <i>Transient Recorder</i> .....	60
Fig. 6.1 Instrumentos de medição usados para comparação de medidas do <i>MatlabScope</i> .....	61
Fig. 6.2 Bancada de montagem e testes.....	63
Fig. 6.3 Bancada de montagem e testes, com a presença do <i>varivolt</i> e do transformador utilizados.....	63

# Lista de Tabelas

Tabela 2.1 Relação de entradas da placa com entradas do sistema proposto.....	23
Tabela 2.2 Fatores de linearização da curva de resposta da Delta 1010LT para frequências de 1 a 60Hz.....	30
Tabela 4.1 Tipos de janela FFT disponíveis no <i>Spectrum Analyze</i> .....	47
Tabela 6.1 Comparação das medidas do <i>MatlabScope</i> com outros instrumentos de medição.....	62

# 1. Introdução

## 1.1 Considerações Gerais

Nos últimos anos, uma das áreas da Engenharia que tem apresentado desenvolvimentos mais significativos é a Engenharia Virtual, designação relacionada ao acrograma *CAE* (da expressão em inglês *Computer-Aided Engineering*). Esta área apresenta duas vertentes principais: a simulação por computador e a instrumentação virtual.

A simulação por computador consiste no uso deste equipamento para simular o comportamento de processos, sistemas, dispositivos, meios e materiais. A instrumentação virtual é o uso de *softwares* personalizáveis e de módulos de *hardware* de medição e geração de sinais para criar sistemas de medição e controle definidos pelas necessidades do usuário. Um instrumento virtual, por sua vez, é uma combinação de elementos de *hardware* e/ou *software* que são tipicamente usados com um computador, geralmente do tipo *PC* (*Personal Computer*), e que tem a funcionalidade de um instrumento convencional, como um multímetro, osciloscópio, gerador de sinais, etc.

Instrumentos virtuais representam uma saída da tradicional instrumentação baseada exclusivamente em *hardware* para a instrumentação baseada em *software*, que usa o poder de processamento, bem como as facilidades de conexão em rede dos modernos sistemas baseados em tecnologia digital atualmente disponíveis, com a vantagem de apresentarem preços cada vez mais acessíveis.

Pode-se usar a instrumentação virtual numa grande variedade de aplicações. Em atividades relacionadas a teste, automação, projeto e engenharia de controle, ela é normalmente usada para adquirir informações de uma variedade de fontes e para gerar sinais para o controle de outros dispositivos. Um sistema de instrumentação virtual apresenta uma grande versatilidade, pois apenas com recursos de *software*, um mesmo conjunto de *hardware* pode ser usado em diversas aplicações de medição e/ou controle.

A instrumentação virtual cresceu significativamente desde seu início no final da década de 70. Adicionalmente, pacotes de *softwares* como o *LabVIEW*, da *National Instruments* e outras linguagens de programação gráfica contribuíram para esse crescimento, possibilitando o desenvolvimento sistemas de alta complexidade num relativo curto período de tempo.

A aquisição de dados é a base da instrumentação virtual. Aquisição de dados é o processo de coletar e medir sinais elétricos de sensores, transdutores, pontas de prova ou equipamentos, introduzindo-os a um computador para processamento. A aquisição de dados tipicamente no registro de sinais e formas de onda e o seu posterior processamento para obter informações sobre o comportamento de uma determinada grandeza física presente no mundo real. Os dados adquiridos são armazenados, analisados e mostrados em um computador. Para isso, utilizam-se *softwares* que podem ser desenvolvidos utilizando várias linguagens de programação de propósito geral como *BASIC*, *C/C++*, *Fortran*, *Java*, *Pascal*. Pode-se também utilizar linguagens direcionadas à aquisição de dados, como a já citada *LabVIEW*, ou ainda *EPICS*, usada para construir sistemas de aquisição de dados de larga escala, ou também *Matlab* que além de ser uma linguagem de programação de alto nível também apresenta ferramentas gráficas e bibliotecas de *software* para análise e aquisição de dados.

O *hardware* do sistema de aquisição de dados usualmente constitui a interface entre o processo e o computador. Ele pode estar na forma de módulos que podem ser conectados às portas do computador (paralela, serial, *USB*, etc.) ou na forma de placas conectadas aos *slots* (*PCI*, *ISA*, *PCI-E*) na placa mãe. Usualmente o espaço atrás de uma placa *PCI* é muito pequeno para todas as conexões necessárias, assim um módulo extra de conexões é quase sempre utilizado.

## 1.2 Objetivos

O objetivo principal deste trabalho é o desenvolvimento de um sistema de aquisição de dados de baixo custo, aplicado a uma plataforma de um computador pessoal, que será usado na aquisição e análise de dados de tensão e corrente elétricas de sistemas trifásicos de alta ou baixa tensão.

O desenvolvimento está baseado na especificação do conjunto do *hardware* bem como o desenvolvimento de uma aplicação de *software*, utilizando a linguagem *MATLAB*, largamente empregada no ambiente acadêmico como uma poderosa ferramenta de análise matemática. Tal fato faz com que seu uso neste trabalho seja justificado.

O sistema de aquisição desenvolvido deve ser capaz de medir e analisar, simultaneamente, sinais provenientes de seis canais de entrada. Cada canal deve estar associado à medição de uma tensão ou corrente elétrica, relativa a cada fase do sistema elétrico sob teste.

A sua aplicação base, como citada anteriormente, será feita na análise dos sinais de tensão e corrente elétricas de um sistema trifásico operando em 50 ou 60 Hz. Serão analisadas informações importantes das grandezas do sistema, tais como amplitudes e valores eficazes das componentes harmônicas fundamentais de cada sinal, análise espectral, formas de onda dos sinais adquiridos, detecção de transitórios, entre outras que serão discutidas com mais detalhes no decorrer deste trabalho.

O sistema de aquisição desenvolvido deve ser capaz de exibir todos os resultados desejados na tela do computador, assim como gerar arquivos documentando as medições e análises feitas.

Apesar de um sistema de aquisição de dados ser também capaz de gerar sinais para controlar outros dispositivos, essa funcionalidade não será implementada no sistema desenvolvido.

## 1.3 Metodologia

As etapas desenvolvidas nesse trabalho se referem a:

- Resumo teórico sobre aquisição e processamento de dados;
- Descrição do sistema proposto;
- Desenvolvimento de uma aplicação de *software* para realização das medições e análises dos sinais;
- Testes do sistema implementado;
- Comparação dos resultados obtidos pelo sistema proposto com os resultados obtidos por outros sistemas de medição.

## 2. Aquisição e Processamento de Dados

### 2.1 Medição e Instrumentação

Medição é a atividade de comparar uma grandeza relacionada a um fenômeno da natureza com um padrão pré-definido. Através da medição o homem pode expressar numericamente qualidades de um objeto ou fenômeno. Sem a medição, o homem fica refém de conceitos como "forte/fraco", "largo/fino", "grande/pequeno", etc.; porém, com a medição, o homem pode raciocinar com mais precisão acerca das referidas qualidades.

O ato de medir envolve essencialmente a existência de unidades de medida, que são os comparativos usados na medição. Envolve também a existência de instrumentos de medição que, graduados de acordo com a unidade de medida em questão, fornecem com variados graus de precisão a medida desejada. Instrumentos são dispositivos construídos para manter relações predeterminadas entre os parâmetros que estão sendo medidos e as variáveis físicas sob investigação. O parâmetro físico sob investigação é conhecido como *mensurando*. Sensores e transdutores são os mais básicos e primários elementos que respondem a uma variação física para produzir uma saída. A relação entre o sinal do sensor e a variação física pode ser expressa na forma de função de transferência. A função de transferência pode ser linear ou não-linear. A relação linear, por exemplo, é regida pela equação:

$$y=a+bx, \tag{2.1}$$

onde  $y$  é o sinal elétrico do sensor,  $x$  é o estímulo físico,  $a$  é o sinal de saída para uma entrada nula em  $x$ , e  $b$  é conhecido como sensibilidade.

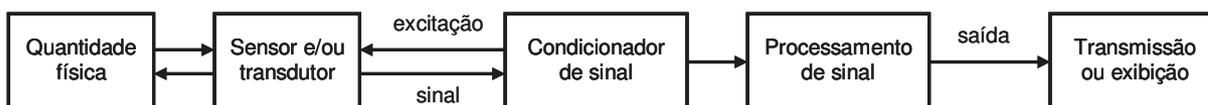
A saída  $y$  pode ser em amplitude, frequência, fase, ou outra propriedade, dependendo das características de um sensor em particular. As funções de transferência não-lineares podem ser logarítmicas, exponenciais ou outras formas, como uma função potência de ordem maior ou igual a dois. Em muitas aplicações, um sensor não-linear pode funcionar linearmente dentro de uma limitada faixa de operação.

A energia de saída do sensor é normalmente fornecida para um transdutor, que converte energia de uma forma em outra. Portanto, um transdutor é um dispositivo capaz de transferir energia entre dois sistemas físicos.

Para uma aplicação específica, uma diversidade de sensores e transdutores é normalmente disponível para satisfazer as necessidades de medição em um sistema físico particular. Deve-se sempre selecionar sensores adequados, assim como empregar métodos de processamento de sinais apropriados para obter a informação requerida com o máximo grau de verossimilhança.

Uma vez obtida a informação, ela pode ser processada, interpretada ou reorganizada. Testes e experimentos geralmente são concebidos, planejados, realizados e repetidos de tal modo a se obter o maior grau de confiança possível nos resultados proporcionados pelos dados coletados.

Um sistema de aquisição de dados típico é constituído pelos elementos indicados na Fig. 2.1.



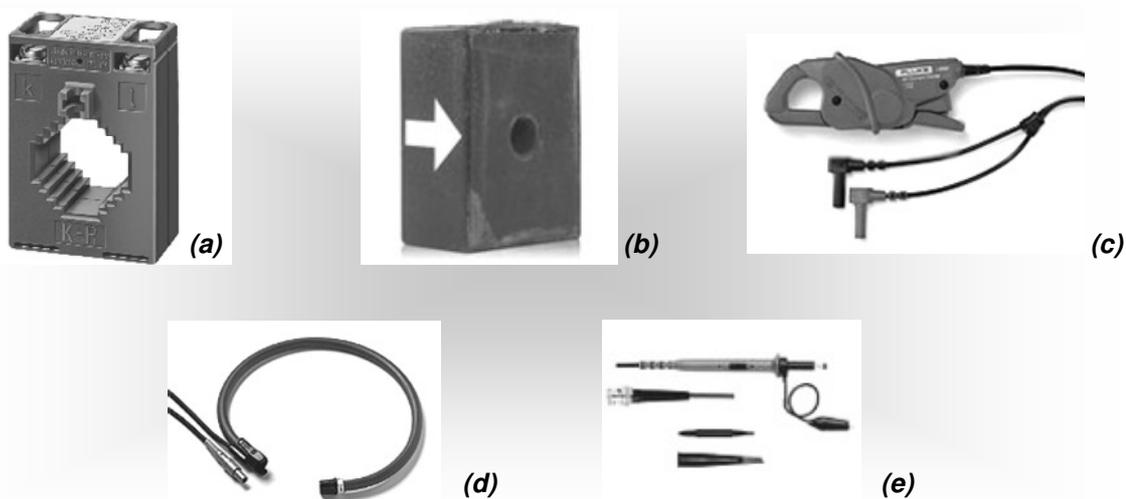
**Fig. 2.1**  
**Elementos essenciais de um sistema de aquisição de dados.**

Um sistema de aquisição de dados pode ter todos ou alguns desses blocos funcionais. Em geral, se o comportamento do sistema físico sob investigação é conhecido, seu desempenho pode ser monitorado e avaliado por meio de métodos adequados de sensoriamento, condicionamento de sinal e processamento, culminando com a exibição e/ou transmissão dos resultados.

## 2.2 Sensores e/ou Transdutores

Como mostra o diagrama da Fig. 2.1, a quantidade física de interesse é sentida, ou presenciada, pelos sensores ou transdutores. Um transdutor, como foi dito anteriormente, é um dispositivo que converte uma propriedade física ou fenômeno em um sinal elétrico mensurável correspondente, como tensão, corrente, variação de resistência ou capacitância, etc. Ele também tem a finalidade de converter um sinal elétrico em um fenômeno físico. Isso explica o fato das setas estarem em sentidos opostos na Fig. 2.1. A capacidade que um sistema de aquisição de dados apresenta em medir com boa precisão diferentes fenômenos depende em grande parte do tipo de transdutor escolhido para compor o *hardware* do sistema.

Em sistemas de aquisição de dados, transdutores e sensores são sinônimos. Há transdutores específicos para uma variedade de aplicações, tais como medição de temperatura, pressão, vibração, etc. A Fig. 2.2 mostra alguns tipos de sensores.



**Fig. 2.2**  
**Sensores para aplicação no sistema proposto: (a) transformador de corrente, (b) sensor de corrente de efeito hall, (c) pinça amperimétrica, (d) pinça amperimétrica flexível, (e) pontas de prova de tensão.**

No sistema desenvolvido neste trabalho, o objetivo é medir tensões e correntes elétricas em sistemas trifásicos. Geralmente, tais grandezas assumem valores elevados quando comparadas às tensões e correntes que podem ser impostas aos instrumentos utilizados no processo de aquisição. Tal fato leva à seleção de dispositivos que sejam capazes de atenuar a ordem de grandeza desses sinais para valores aceitáveis pelo *hardware* do sistema, além de proporcionarem isolamento elétrico, estabelecendo segurança para o pessoal de operação.

Os sensores comercialmente disponíveis para uso no sistema proposto são, basicamente, os transformadores de potencial, os transformadores de corrente, sensores de corrente baseados em efeito Hall, pinças amperimétricas, pinças voltimétricas, entre outros.

## 2.3 Sinais e Condicionamento de Sinais

Sinal é definido como “alguma quantidade física que varia com o tempo, espaço ou com alguma outra variável independente ou variáveis.” Matematicamente, um sinal pode ser definido como uma função de uma ou mais variáveis independentes.

Sinais em sistemas de aquisição de dados podem ser digitais (também chamados de sinais lógicos) ou analógicos, dependendo do transdutor usado.

O condicionamento de sinais pode ser necessário se o sinal do transdutor não é apropriado para o *hardware* do sistema. Por exemplo, um sinal de tensão com ordem de grandeza de centenas de volts não é adequado a ser inserido diretamente num circuito de monitoramento que opera com tensões da ordem de unidades de volts, como é o caso das placas de aquisição de dados usadas em computadores.

O condicionamento de sinais consiste basicamente na amplificação, atenuação, filtragem, isolamento, linearização ou até mesmo demodulação dos sinais provenientes dos sensores ou transdutores, a fim de adaptá-los aos requisitos necessários para processamento.

Sinais analógicos são sempre convertidos para digitais antes de serem enviados ao computador para um posterior processamento.

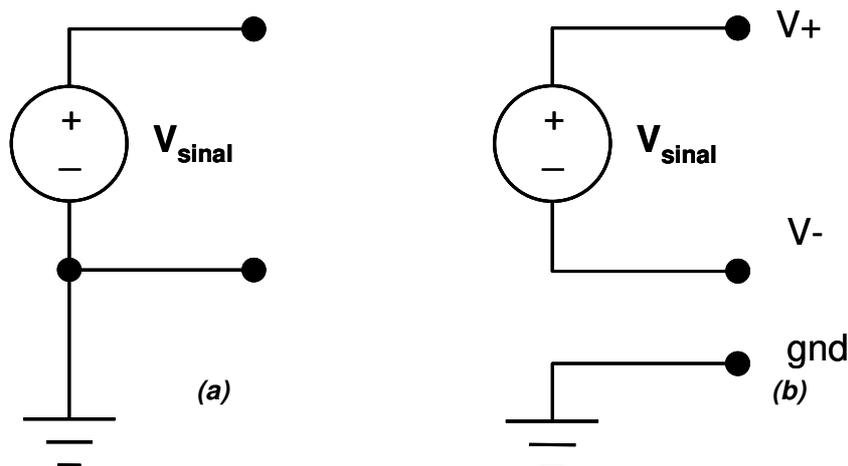
Os sinais podem ser classificados segundo vários critérios: amplitude, relação do potencial dos terminais de sua fonte com o terra, largura de banda, propriedades de impedância de saída da fonte que os fornecem, etc.

Geralmente, sinais da ordem de 100 mV são considerados sinais de baixo nível, e precisam de amplificação. Todavia, grandes sinais podem também necessitar de amplificação em função da gama de entrada do receptor.

O sinal que vem de uma fonte onde um de seus terminais de saída está a um potencial constante, normalmente igual ao potencial de terra, é chamado de *single-ended*, ou, com uma tradução livre, sinal de terminação simples, ou ainda, sinal desbalanceado (Fig. 2.3 (a)). Um sinal que tem dois terminais de saída, nos quais os seus potenciais em relação a um terceiro terminal de referência muda com a mesma intensidade, mas com direções opostas, é conhecido como sinal diferencial ou sinal balanceado (Fig. 2.3 (b)). Geralmente os sinais balanceados são mais imunes a ruídos induzidos ou ruídos que estejam presentes nos dois terminais do sinal, pois operam em circuitos com entrada diferencial que apresentam rejeição em modo comum, atenuando ou eliminando tais ruídos. Os sinais fornecidos pelos sensores descritos na Fig. 2.2, por exemplo, são normalmente desbalanceados, sendo sujeitos, portanto, a níveis de ruído que podem ser elevados para uma determinada aplicação.

Um sinal de banda estreita apresenta em seu espectro frequências que estão dentro de uma pequena banda. Já os sinais de banda larga apresentam um espectro de frequência muito maior.

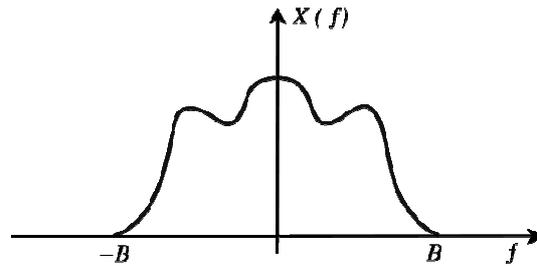
A impedância de saída de um sinal determina os requisitos da impedância de entrada dos circuitos de condicionamento para evitar efeitos de carga, e assim satisfazer ao casamento de impedâncias. Para sinais em baixa frequência, é relativamente fácil conseguir altas impedâncias de entrada, como aquelas de sensores piezoelétricos. Entretanto, em altas frequências, o efeito das capacitâncias de entrada faz com que a impedância seja dependente da frequência, e isto é extremamente problemático para se alcançar o perfeito casamento de impedâncias.



**Fig. 2.3**  
**Tipos de sinal: (a) sinal desbalanceado; (b) sinal balanceado com os terminais referenciados ao terra.**

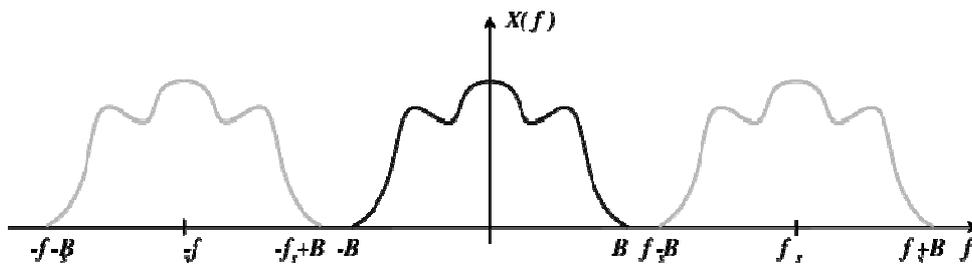
Ao se definir o projeto de um sistema de aquisição de dados, uma das escolhas a fazer é a máxima frequência do sinal capaz de ser lido e analisado pelo sistema. Uma vez que os sinais serão levados ao computador para o processamento, é necessário que os sinais analógicos provenientes dos sensores sejam convertidos adequadamente para o formato digital. O processo de amostragem converte os sinais analógicos de sua forma contínua em sinais discretos. Amostragem pode ser definida como o processo de medição instantânea de valores de um sinal analógico em intervalos regulares. O intervalo entre as amostras é determinado por um pulso de sincronismo e a sua frequência é chamada de taxa de amostragem. De acordo com o teorema de Nyquist, a quantidade de amostras por unidade de tempo de um sinal, chamada taxa ou frequência de amostragem, deve ser maior ou igual ao dobro da maior frequência contida no sinal a ser amostrado, para que esse possa ser reproduzido integralmente sem erro de *aliasing* (este termo será explicado logo em seguida). A metade da frequência de amostragem é chamada frequência de Nyquist e corresponde ao limite máximo de frequência do sinal que pode ser reproduzido. Como é impossível garantir que o sinal não contenha sinais acima deste limite (distorções, interferências, ruídos, etc.), é necessário filtrar o sinal com um filtro passa-baixa com frequência de corte igual ou menor que a frequência de Nyquist. O mesmo é denominado filtro *anti-aliasing*.

Algumas características dos sinais analógicos são mais facilmente entendidas no domínio da frequência. O espectro de um sinal analógico contínuo é a representação gráfica da frequência versus a amplitude do sinal. Na Fig. 2.4 apresenta-se um exemplo do espectro de um sinal limitado em B Hz. O eixo x representa a frequência enquanto que o eixo y corresponde à amplitude.



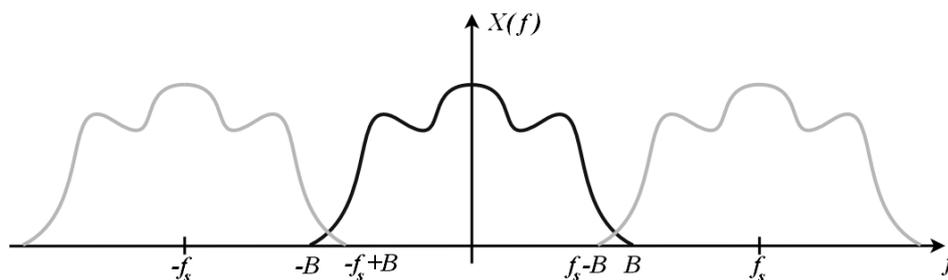
**Fig. 2.4**  
**Espectro de frequência de um sinal analógico.**

Ao se realizar a amostragem de um sinal analógico, estaremos matematicamente realizando no domínio do tempo uma multiplicação do sinal por um trem de impulsos espaçados de  $t_s$  segundos um do outro. O parâmetro  $t_s$  corresponde ao período de amostragem e é equivalente ao inverso da frequência de amostragem,  $f_s$ . Como resultado, o espectro do sinal resultante dessa multiplicação é equivalente ao deslocamento do espectro do sinal analógico original para as frequências  $\pm n.f_s$ , ( $n=1, 2, 3\dots$ ), além de conter o espectro original do próprio sinal. A Fig. 2.5 mostra o efeito da amostragem no espectro do sinal resultante.



**Fig. 2.5**  
**Espectro de frequência de um sinal analógico amostrado com uma frequência  $f_s$ .**

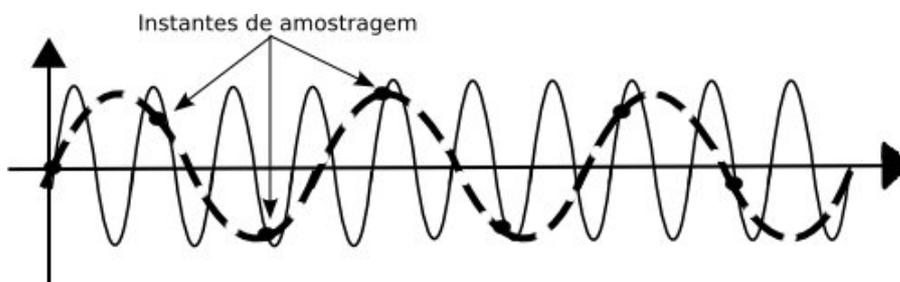
Se ocorrer o caso em que  $f_s - B < B$ , ou seja,  $f_s < 2B$  ocorrerá sobreposição dos espectros deslocados e o espectro resultante fica como mostrado na Fig. 2.6.



**Fig. 2.6**  
**Espectro de frequência de um sinal analógico amostrado com uma frequência  $f_s < 2B$ .**

A superposição das raiais do espectro do sinal é denominada *aliasing*, que se refere ao fenômeno de um componente de alta frequência assumir a identidade de um de baixa frequência. Nesse caso, para que não ocorra esse fenômeno, devemos fazer  $f_s > 2B$ , o que obedece a idéia original de Nyquist.

O *aliasing* distorce o espectro do sinal de tempo contínuo original. Isto ocorre, por exemplo, quando um sinal é amostrado a uma taxa menor que a definida pelo teorema de Nyquist. Na Fig. 2.7 é apresentado um exemplo de *aliasing* do sinal no domínio do tempo. Nota-se que dois sinais possuem o mesmo valor no instante de amostragem, embora suas frequências sejam diferentes. A linha tracejada representa o sinal de *aliasing*, identificado como um sinal de frequência mais baixa.



**Fig. 2.7**  
**Aliasing de um sinal no domínio do tempo.**

No espectro de frequências, as raízes ficam sobrepostas, fazendo com que sinais de alta frequência sejam interpretados como sinais de baixa frequência. Tipicamente, o conteúdo espectral de um sinal portador de informações ocupa uma faixa de frequência com extensão finita.

Para extrair o conteúdo de informação fundamental de um sinal necessita-se de um sistema seletivo de frequência, ou seja, um filtro. As implicações práticas do teorema da amostragem são muito importantes. Como a maioria dos sinais reais não é limitada em banda, podendo facilmente encontrar componentes com frequências maiores que as de interesse, incluindo ruído, é necessário realizar uma pré-filtragem analógica antes da amostragem. O filtro passa-baixas conhecido como filtro anti-aliasing, serve para remover a parte indesejada do espectro do sinal que poderia corromper o sinal convertido. Isso justifica o fato da necessidade de se usar filtros *anti-aliasing* para evitar a representação errada da informação contida na frequência do sinal. Nesse caso, antes de ser amostrado, o sinal de entrada é processado com um filtro eletrônico do tipo passa-baixas para remover todas as frequências acima da frequência de Nyquist (metade da frequência de amostragem). Isto é feito para prevenir *aliasing* durante a amostragem. As características de todos os sinais digitalizados dependerão do tipo do filtro *anti-aliasing* usado na aquisição, que pode ser mais ou menos seletivo, dependendo dos seus parâmetros como frequência de corte e ordem.

## 2.4 Conversão Analógico-Digital

O processo de conversão de um sinal analógico para digital consiste em converter um sinal contínuo no tempo e em amplitude em um sinal digital discreto no tempo e em amplitude. Amostragem e quantização constituem os passos necessários para se alcançar uma conversão analógico-digital. Para minimizar qualquer perda de informação decorrente dessa conversão, é importante entender os princípios da amostragem e da quantização.

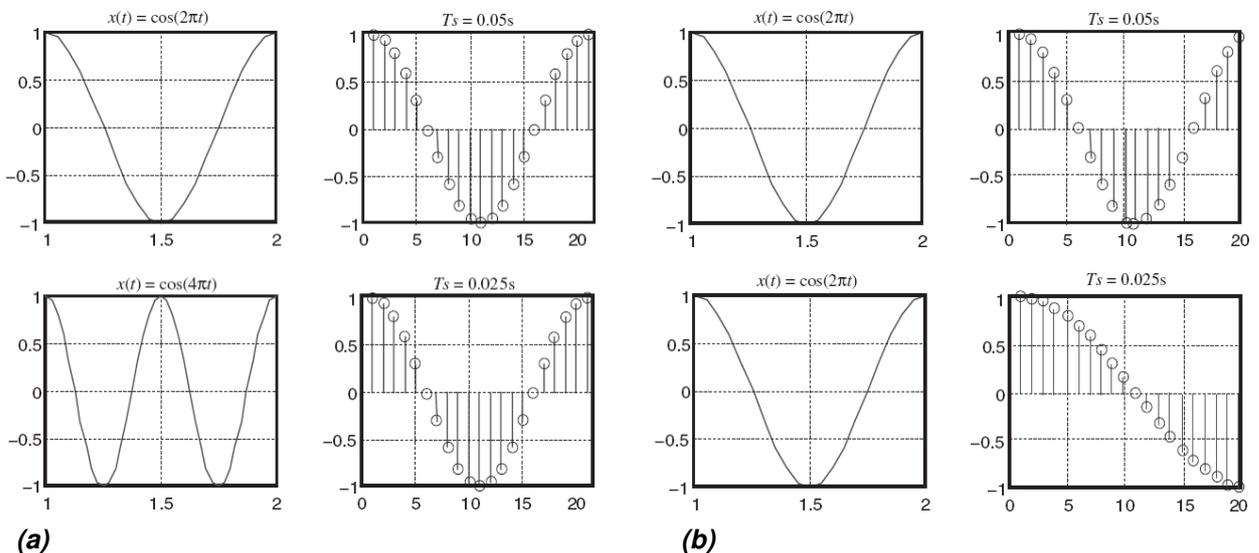
Amostragem, como dito anteriormente, é o processo de gerar amostras discretas no tempo de um sinal analógico. Considera-se um sinal analógico do tipo

$x(t)=A\cos(\omega t+\Phi)$ . Amostrando esse sinal a  $t=n.t_s$ , com o intervalo de amostragem de  $t_s$ , é gerado o sinal discreto no tempo

$$x[n]=A\cos(\omega n t_s + \Phi)=A\cos(\theta n + \Phi), \quad n=0, 1, 2, \dots, \quad (2.2)$$

onde  $\theta=\omega.t_s=2\pi f/f_s$  é a freqüência digital com unidade em radianos (a freqüência do sinal analógico  $\omega$  tem unidade de radianos/segundo).

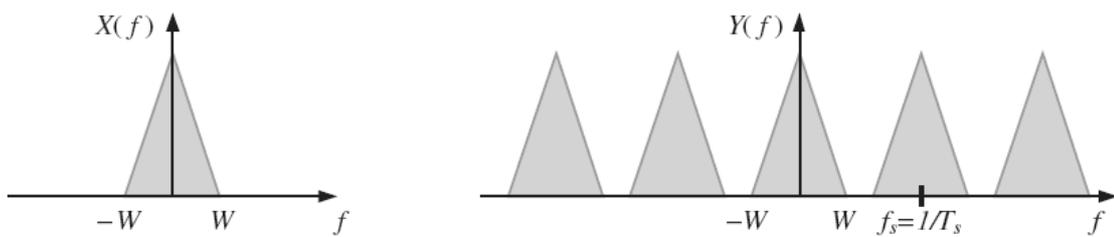
A diferença entre as freqüências digital e analógica é mais evidente observando que o mesmo sinal discreto no tempo é obtido de diferentes sinais contínuos no tempo se o produto  $\omega.t_s$  permanece o mesmo, como mostrado na Fig. 2.8. Do mesmo modo, diferentes sinais discretos no tempo podem ser obtidos do mesmo sinal analógico ou contínuo no tempo para diferentes freqüências de amostragem, como é também mostrado na Fig. 2.8. Em outras palavras, a freqüência  $f$  do sinal analógico e a freqüência de amostragem  $f_s$  definem a freqüência  $\theta$  do correspondente sinal digital.



**Fig. 2.8**  
**Amostragem de sinais:** a) Amostragem de dois diferentes sinais analógicos resultando no mesmo sinal digital; b) amostragem do mesmo sinal analógico resultando em dois diferentes sinais digitais.

Quando um sinal analógico com uma largura de banda de  $W$  (ou com uma máxima freqüência de  $f_{max}$ ) é amostrado a uma taxa de  $t_s=1/f_s$ , sua resposta em freqüência correspondente é repetida a cada  $2\pi$  radianos, ou  $f_s$ , como mostra a Fig. 2.9. Isto justifica o fato de que, para sinais discretos, estamos apenas interessados na faixa de freqüência  $[0, f_s/2]$ .

Para evitar qualquer aliasing ou distorção do conteúdo harmônico do sinal discreto, nós devemos ter  $f_s \geq 2 f_{max}$ . Isto é conhecido como taxa de Nyquist, ou freqüência de Nyquist, isto é, a freqüência de amostragem deve ser pelo menos duas vezes a mais alta freqüência presente no sinal analógico. Amostragem a taxas mais elevadas garante uma descrição mais rica do sinal analógico do que aquela amostrada à taxa de Nyquist. Normalmente, antes de qualquer manipulação digital, um filtro *anti-aliasing* é usado para limitar a banda de freqüência do sinal analógico. Isso é o que o filtro anteriormente descrito faz no sistema.



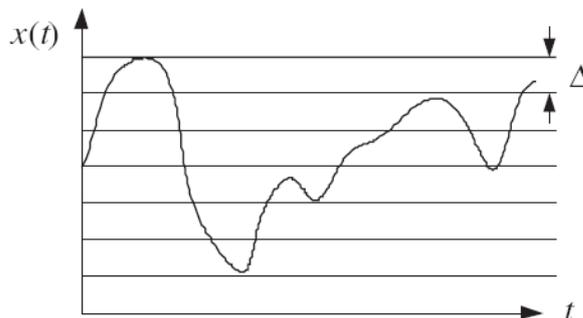
**Fig. 2.9**

**Efeito da amostragem de um sinal analógico de banda limitada em  $W$  a uma taxa de  $T_s=1/f_s$ , resultando na repetição da banda de frequência a cada  $2\pi$  radianos.**

Com o processo de amostragem, obtemos um conjunto de amostras (níveis) correspondentes aos valores instantâneos do sinal analógico nos instantes de amostragem. As amostras podem assumir qualquer valor dentro da faixa dinâmica do sinal original, ou seja, as amostras apresentam ainda valores contínuos de amplitude dentro da faixa dinâmica do sinal.

O processo de converter níveis contínuos de amplitude em níveis discretos é chamado de *quantização*. Um conversor A/D (analógico/digital) tem um número finito de *bits* (também chamado de resolução). Como resultado, valores contínuos de amplitude são representados por níveis discretos de amplitude, configurando assim um processo de quantização. Assim, infinitos valores de amplitudes são representados por uma quantidade finita de amplitudes. Esta aproximação provoca o que se denomina de *erro de quantização* na representação do sinal, e é inerente ao processo de quantização.

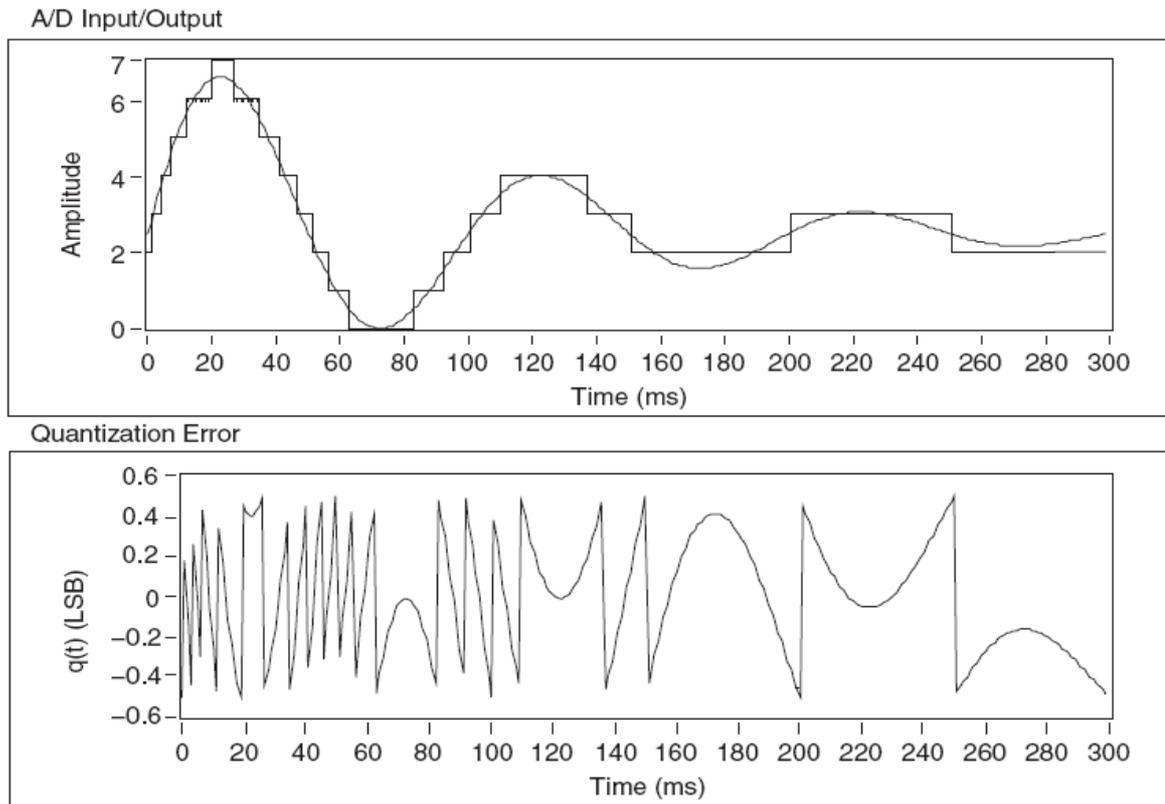
Ao se dividir a faixa dinâmica do sinal em intervalos iguais, especifica-se o *intervalo de quantização*. O intervalo de quantização depende do número de níveis de quantização ou resolução, que é diretamente proporcional ao número de *bits* de um conversor A/D, como mostra a Fig. 2.10. A quantidade de ruído de quantização gerada por um conversor A/D depende do tamanho do intervalo de quantização. Quanto maior o número de *bits*, menor é o intervalo de quantização e menos ruído de quantização será gerado. Na Fig. 2.10, o espaçamento  $\Delta$  entre dois níveis de quantização consecutivos corresponde à mudança de nível do *bit* menos significativo do conversor A/D, chamado de *LSB (least significant bit)*.



**Fig. 2.10**

**Níveis de quantização.**

Usualmente, é assumido que o ruído de quantização é um sinal independente e uniformemente distribuído na faixa de  $-0.5$  LSB a  $0.5$  LSB. A Fig. 2.11 mostra a entrada e a saída e o ruído de quantização de um sinal analógico quantizado por um conversor A/D de 3 *bits*.



**Fig. 2.11**  
**Entrada/saída e erro de quantização de um sinal analógico quantizado por um conversor A/D de 3 bits.**

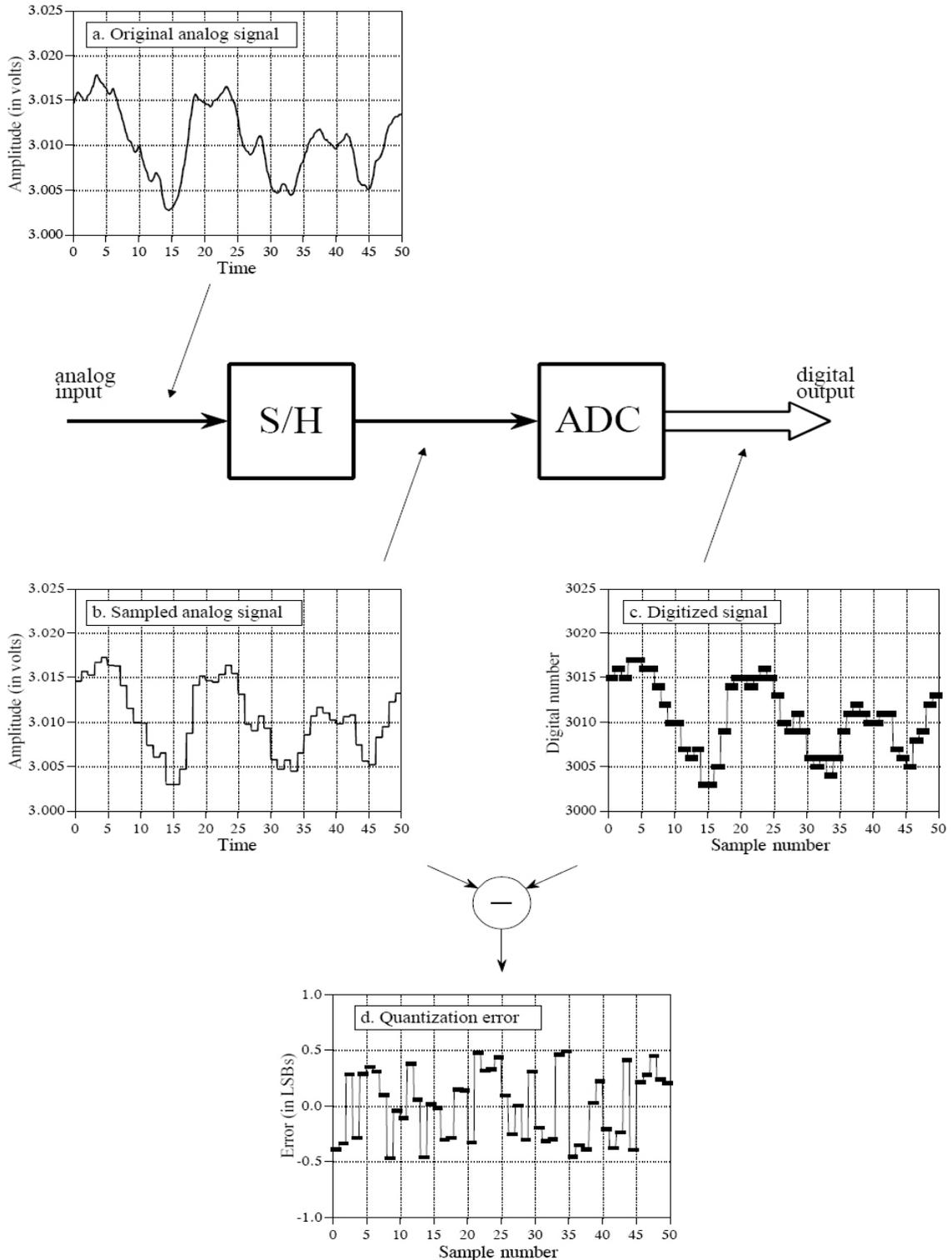
Para proporcionar uma visão geral do processo de conversão analógico-digital, a Fig. 2.12 mostra as formas de onda presentes no processo de conversão. Em (a) tem-se o sinal analógico a ser digitalizado. O diagrama de blocos é quebrado em duas partes. O *sample-and-hold* (*S/H*) é o bloco que amostra o sinal analógico e retém o seu valor enquanto a conversão é feita pelo conversor A/D. O sinal na saída do *S/H* é mostrado em (b). O sinal amostrado pelo *S/H* é quantizado e convertido em um sinal digital formado por  $n$  bits, conforme seja a resolução do conversor A/D, e é mostrado em (c), finalizando o processo de conversão. O sinal mostrado em (d) é o erro de quantização, ou ruído de quantização, resultante da diferença entre o sinal amostrado e o sinal quantizado.

A *resolução* de um conversor indica o número de valores discretos que ele pode produzir abrangendo a faixa dinâmica do sinal a ser digitalizado. Esses valores são usualmente armazenados eletronicamente sob a forma binária, fazendo com que a resolução seja expressa em *bits*. Conseqüentemente, o número de valores discretos disponíveis, ou níveis de quantização, é usualmente uma potência de dois. Por exemplo, um conversor com 3 *bits*, como aquele usado na digitalização do sinal na Fig. 2.11, pode digitalizar um sinal analógico em 8 diferentes níveis.

A resolução também pode ser definida em volts. A resolução em tensão de um conversor analógico digital é igual à sua faixa de valores de entrada dividido pelo número de intervalo de valores discretos, como é mostrado na seguinte equação:

$$Q = \frac{E_{FSR}}{2^M} = \frac{E_{FSR}}{N} \quad (2.3)$$

$Q$  é a resolução em volts por nível, (volts por código binário de saída),  
 $E_{FSR}$  é a faixa de tensão da entrada do conversor =  $V_{max} - V_{min}$ ,  
 $M$  é a resolução em *bits* do conversor,  
 $N$  é o número de intervalos de quantização, dado pelo número de códigos disponíveis na saída do conversor, que é dado por:  $N = 2^M$ .



**Fig. 2.12**  
**Formas de onda presentes no processo de conversão.**

## 2.5 Processamento de Sinais

O processamento de sinais consiste na análise e/ou modificação de sinais de forma a extrair informações dos mesmos e/ou torná-los mais apropriados para alguma aplicação específica. Em linhas gerais, o processamento de sinais pode ser feito de forma analógica ou digital. Para a realização de processamento digital, os sinais devem ser previamente digitalizados, segundo as regras que foram previamente descritas. Após terem sido apropriadamente convertidos para o formato digital, o processamento dos mesmos é realizado mediante algoritmos baseados em modelos matemáticos de naturezas diversas.

Hoje em dia existem diversos dispositivos que podem ser usados no processamento digital de sinais, como os *chips DSP's* (os mais rápidos e versáteis), microcontroladores, *FPGA's*, ou mesmo os computadores pessoais que trazem consigo processadores cada vez mais rápidos.

O Processamento Digital de Sinais ou PDS (também chamado de *DSP*, do inglês *Digital Signal Processing*) possui diversas técnicas computacionais que podem ser utilizadas diretamente em um sistema computacional baseado no *IBM-PC* padrão, sem necessitar do uso de equipamentos de *hardware* específicos como *FPGA's* ou microcontroladores. As transformadas matemáticas como Fourier e *Wavelets* são exemplos deste tipo de processamento. Outras aplicações dizem respeito à filtragem digital, digitalização e reconstrução de sinais analógicos, extração de características, como a conversão de fala em texto, compressão de dados para armazenamento e transmissão, etc. As aplicações de processamento digital de sinais se constituem de elementos de *software* (programas) que são escritos em linguagens convencionais como *C/C++*, *Basic*, *Assembly*, *Java*, ou ainda em linguagens de uso específico como a *LabVIEW* ou *EPICS*.

Os principais campos de aplicação do processamento digital de sinais são: processamento de imagem, processamento de áudio e vídeo, reconhecimento de fala, entre outros. As técnicas de processamento de sinais podem ser de muita utilidade no controle e análise de sistemas físicos de interesse dos mais diversos pesquisadores, não só de engenheiros eletricitas. Atualmente, técnicas de processamento de sinais também têm atraído a atenção de profissionais de outras áreas como economia, biologia e saúde.

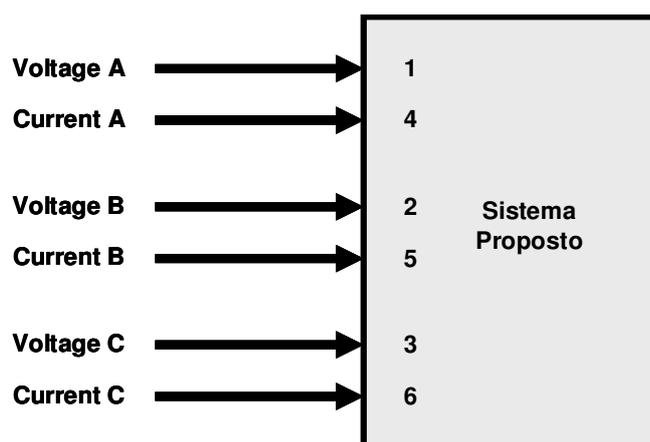
## 3. Descrição do Medidor Virtual

### 3.1 Especificação do Sistema Proposto

O sistema de aquisição de dados proposto neste trabalho deve ser capaz de medir e analisar, simultaneamente, sinais provenientes de seis canais de entrada. Cada canal deve estar associado à medição de uma tensão ou corrente elétrica, relativa a cada fase do sistema elétrico sob análise. Os sinais e os canais de entrada do sistema são nomeados do seguinte modo:

- \*Canal 1: Voltage A (canal de medição da tensão da fase A)*
- \*Canal 2: Voltage B (canal de medição da tensão da fase B)*
- \*Canal 3: Voltage C (canal de medição da tensão da fase C)*
- \*Canal 4: Current A (canal de medição da corrente da fase A)*
- \*Canal 5: Current B (canal de medição da corrente da fase B)*
- \*Canal 6: Current C (canal de medição da corrente da fase C)*

Esquemáticamente, podemos representar o sistema proposto, especificando suas entradas, conforme mostra a Fig. 3.1.



**Fig. 3.1**  
**Representação do sistema proposto.**

As entradas especificadas no esquema acima recebem sinais de tensão e corrente do sistema elétrico trifásico sob análise. Esses sinais devem ter amplitudes adequadas às características do sistema de medição desenvolvido. Também deve ser promovida isolamento elétrico entre os dois sistemas citados, resguardando-se a integridade e a segurança dos instrumentos e do operador.

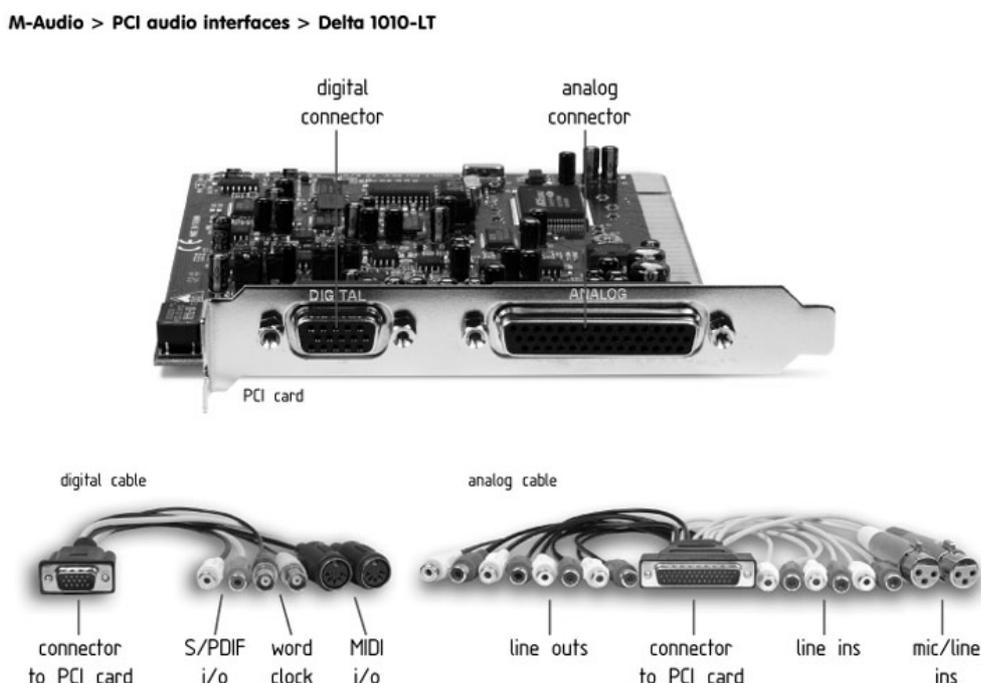
O sistema proposto tem uma faixa de operação capaz de analisar até a 50ª harmônica da frequência fundamental de 60 Hz, ou até a 60ª harmônica da frequência fundamental de 50 Hz, ou seja, possui faixa de operação até 3000 Hz. Essa faixa é suficientemente larga para se analisar as harmônicas de maior interesse presentes nos sistemas elétricos trifásicos.

O sistema proposto é também capaz de analisar sub-harmônicas das suas componentes de frequência fundamentais, assim como é também possível a análise de interharmônicos (componentes de frequência, em tensão ou corrente, que não são múltiplos inteiros da frequência fundamental). A faixa estendida, nesse caso vai de 1Hz até 3000 Hz, como especificado anteriormente, e a resolução de frequência

é de 1Hz. A resolução, nesse caso, é a menor diferença de frequência que pode ser medida entre dois sinais distintos. Resumidamente, tem-se:

- \*Faixa dinâmica: dependente do condicionamento dos sinais
- \*Faixa de frequência: 1 Hz – 3000 Hz
- \*Resolução de frequência: 1 Hz

Normalmente, as placas de uso específico em aquisição de dados têm preços bem elevados, dependendo do tipo de aplicação. O *hardware* usado para a aquisição dos sinais é um dos pontos de destaque do sistema proposto, e é o que caracteriza o seu baixo custo. Optou-se pelo uso de uma placa de som, conectada ao barramento *PCI* do computador. A placa não é de uso geral, e sim, de uso específico para gravação profissional de música, o que confere confiabilidade ao *hardware* escolhido. Desenvolvida pela consagrada marca *M-Audio*, a placa é a *Delta 1010LT*, mostrada na Fig. 3.2.

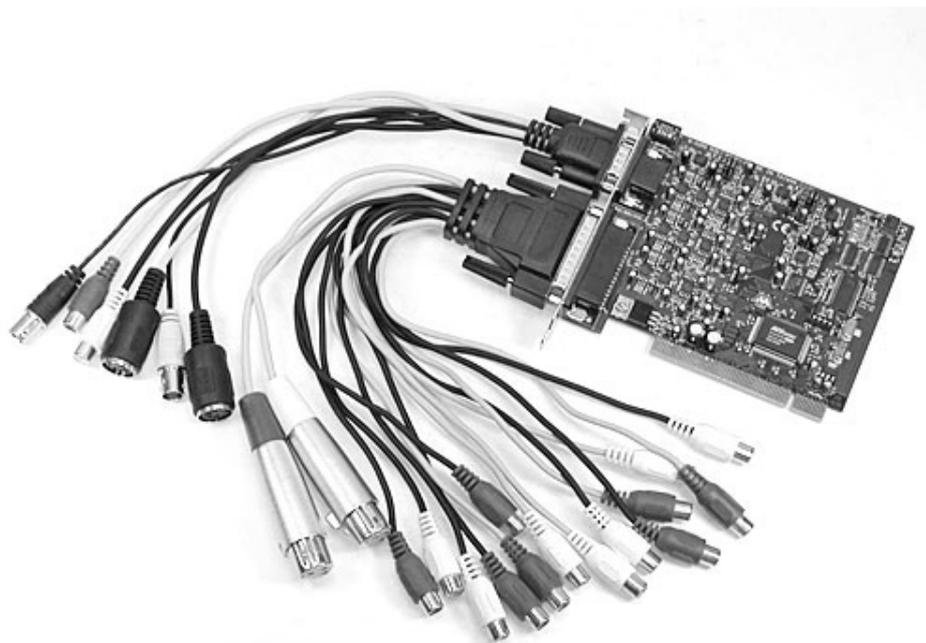


**Fig. 3.2**  
**Placa de Som Delta 1010LT, usada para aquisição de dados do sistema proposto.**

Na Fig. 3.2, pode-se ver que o acesso ao circuito da placa é feito através de dois cabos. O cabo digital possui as conexões de *MIDI* e de *S/PDIF*, além de uma entrada denominada *word clock*. As conexões *MIDI* e *S/PDIF* são interfaces digitais comumente usadas em música. A entrada *word clock* é usada para sincronizar a placa por meio de um *clock* externo, independente do *clock* do processador do computador no qual a placa está instalada. O cabo analógico possui as entradas e as saídas analógicas da placa. São 8 entradas analógicas, sendo duas balanceadas, acessadas por plugues do tipo *XLR*, e 6 desbalanceadas, acessadas por plugues do tipo *RCA*, e 8 saídas analógicas, todas desbalanceadas, acessadas por plugues do tipo *RCA*.

No sistema proposto, apenas as 6 entradas analógicas dotadas de plugues *RCA* são utilizadas como entradas de sinal. Observe que a distinção das entradas e saídas com plugues *RCA* é feita pela cor do cabo: os plugues usados nas entradas têm os cabos de coloração clara. As saídas têm os cabos de coloração escura.

O cabo analógico e o digital são ligados aos respectivos conectores, como é mostrado na Fig. 3.3.



**Fig. 3.3**  
**Conexão dos cabos analógico e digital à placa.**

As entradas da placa são numeradas, nos plugues, de 1 a 8, sendo as duas primeiras relativas às entradas com plugues do tipo *XLR*. As entradas numeradas de 3 a 8 são as que possuem plugues do tipo *RCA*, e são as utilizadas no sistema proposto. O par de entradas 3-4 recebe os sinais da fase A, o par 5-6 recebe os sinais da fase B e o par 7-8 recebe os sinais da fase C. Em cada par, a entrada com menor número recebe sempre o sinal relativo à tensão (*entrada 3, do par 3-4, por exemplo*), e a entrada com maior número recebe o sinal relativo à corrente (*entrada 4, do par 3-4, por exemplo*).

Resumidamente, a associação entre entradas da placa e sinais do sistema proposto é mostrada na Tabela 2.1. Observa-se que nesta tabela há uma separação, em linhas, dos pares de entrada relativos a cada fase do sistema elétrico sob análise.

**Tabela 2.1**  
**Relação de entradas da placa com entradas do sistema proposto.**

Entrada da placa	Entrada do sistema	Sinal do sistema
3	Canal 1	Voltage A
4	Canal 4	Current A
5	Canal 2	Voltage B
6	Canal 5	Current B
7	Canal 3	Voltage C
8	Canal 6	Current C

O circuito de aquisição de dados da placa compreende condicionadores de sinal e conversores analógico-digital para cada canal, além de um *chip DSP* para processamento local dos dados. Os condicionadores de sinal são destinados especificamente a aquisição de sinais de áudio, não servindo diretamente à aplicação no sistema proposto. Assim, deve haver na entrada da placa outro circuito de condicionamento de sinal, com características apropriadas ao sistema.

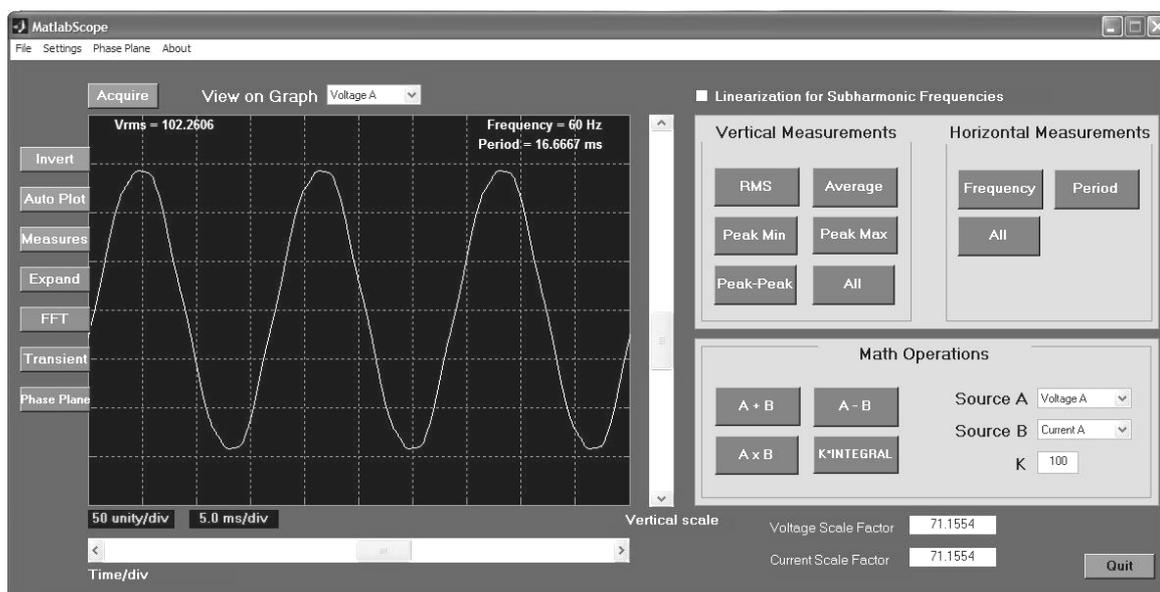
Para que o sistema funcione, é necessário instalar no computador o *driver* da placa, fornecido pelo seu fabricante. Esse *driver* é um *software* destinado a ajustes e análise do funcionamento da placa. Os detalhes específicos sobre o mesmo serão tratados no item relacionado à caracterização do sistema.

Para aquisição e análise dos dados, tem-se a interface principal do sistema, denominada *MatlabScope*, que é um *software*, escrito em linguagem *Matlab* para ambiente *Windows*, responsável pelo controle da aquisição dos dados, bem como do processamento e análise dos mesmos. Também trata da exibição dos resultados de interesse em forma gráfica ou numérica, além de gerar arquivos para documentação destes resultados. É uma interface composta por botões, telas de exibição de gráficos e menus, semelhante a qualquer programa de uso genérico. A interface possui um grande apelo intuitivo, fazendo com que o usuário possa operar o sistema de aquisição sem necessariamente conhecer a linguagem *Matlab*.

O *MatlabScope* é na verdade um conjunto de 3 interfaces gráficas que representam painéis de instrumentos de medição e análise individuais. Esses instrumentos são o *Osciloscópio*, o *Analizador de Espectro* e o *Gravador de Transitórios*.

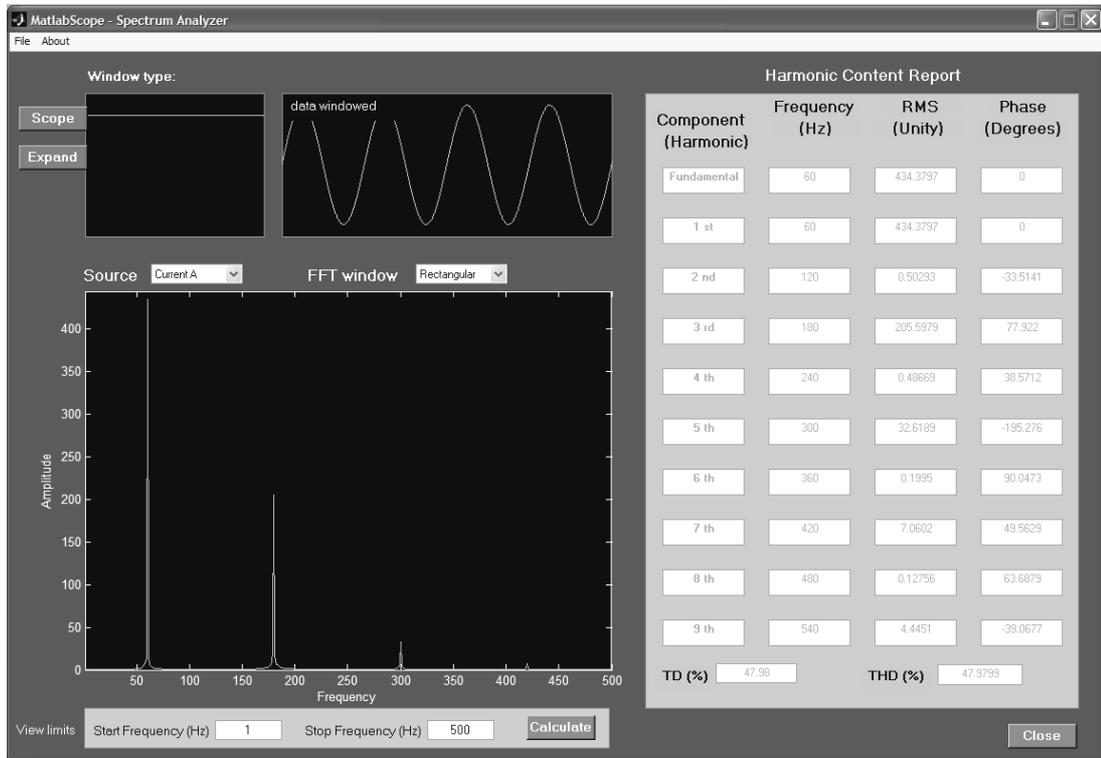
O Osciloscópio, ou simplesmente *MatlabScope*, é a interface principal, mostrada na Fig. 3.4. É responsável pela aquisição e análise básica dos dados. A análise básica dos dados diz respeito ao cálculo de valor *RMS* dos sinais, valor de pico e pico-a-pico, valor médio, frequência e período dos sinais, além de operações matemáticas como soma de sinais, subtração, multiplicação e integração. A interface dispõe ainda das funcionalidades presentes em um osciloscópio clássico, como o ajuste de *volts/div* e *time/div* para uma melhor visualização do sinal na tela, além de possibilitar a inversão da forma de onda exibida, entre outras.

A partir do *MatlabScope*, é possível abrir as outras interfaces para análise mais detalhadas dos dados.



**Fig. 3.4**  
**Interface principal do MatlabScope.**

O Analisador de Espectro (*Spectrum Analyzer*), mostrado na Fig. 3.5, é a interface que efetua a análise espectral dos sinais. Esta análise consiste em determinar o conteúdo de harmônicos do sinal, dentro da faixa de frequência especificada, obtendo as amplitudes dos sinais, suas harmônicas, sub-harmônicas e inter-harmônicas, assim como as defasagens em relação à fundamental.



**Fig. 3.5**  
**Interface do Spectrum Analyzer.**

O gravador de transitórios (*Transient Recorder*) é mostrado na Fig. 3.6.



**Fig. 3.6**  
**Interface do Transient Recorder.**

Esta interface é responsável pela gravação dos sinais de tensão e corrente após a ocorrência de distúrbios no sistema elétrico, causados por operações de chaveamento, aplicação ou retirada súbita de cargas ao sistema, etc. Os parâmetros de detecção do distúrbio são programados pelo usuário e a gravação só ocorre a partir do instante em que o sinal observado satisfizer todos os requisitos programados.

Os detalhes específicos de cada interface serão descritos na seção *Desenvolvimento do Software*.

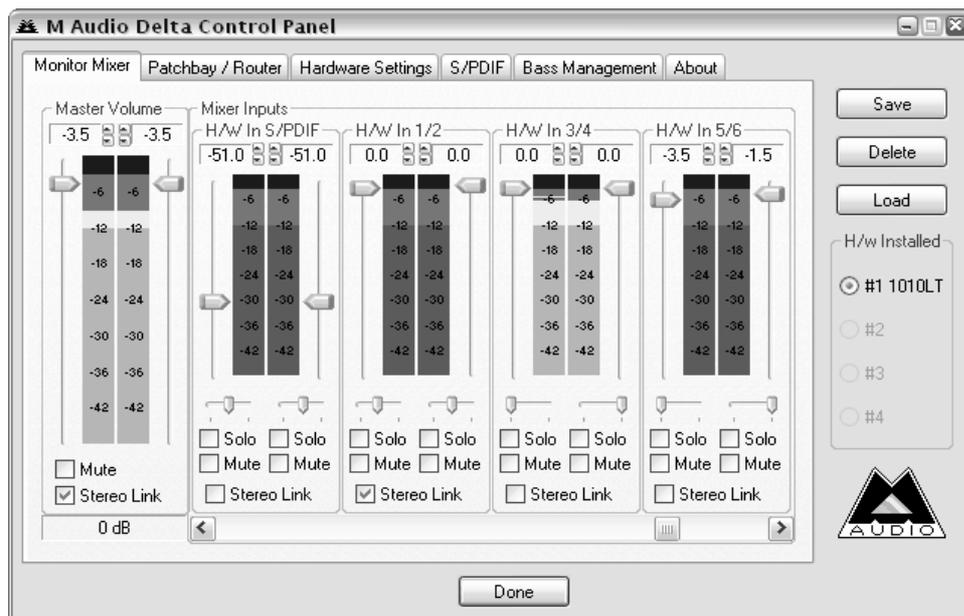
O sistema completo pode ser então, agora, descrito pelo diagrama da Fig. 3.7. Nesta figura, os sinais passam inicialmente por um circuito condicionador que os torna adequados para que a placa de som possa recebê-los com segurança. A placa de som, controlada pelo *MatlabScope*, funcionando em ambiente *Windows*, digitaliza esses sinais e o próprio *MatlabScope* processa esses dados a fim de extrair as informações de interesse do usuário enviando-as para a exibição. No caso, a exibição é feita dentro da própria interface por meio do monitor de vídeo do computador, ou por meio de arquivos gerados para documentação.



**Fig. 3.7**  
**Sistema implementado.**

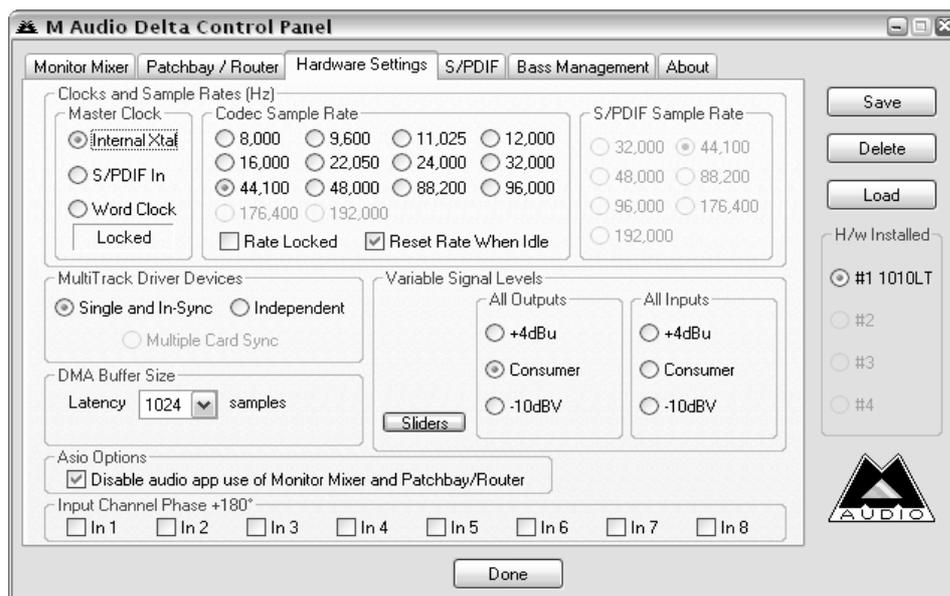
### 3.2 Caracterização do Sistema Proposto

O primeiro trabalho a ser feito é a caracterização completa da placa de som usada no sistema. Para isso, nos baseamos no *driver* da placa instalado no computador. O *driver* é usado no sistema para configurar parâmetros de interesse que controlam a placa. A Fig. 3.8 mostra a interface do *driver* da placa.



**Fig. 3.8**  
**Interface do driver da placa Delta1010LT.**

A interface, vista na Fig. 3.8, apresenta uma estrutura de abas. Cada aba se refere à exibição ou ajustes de parâmetros específicos da placa. Por exemplo, na figura mostrada, podemos observar os detalhes da aba selecionada *Monitor Mixer*. Essa aba exibe graficamente os níveis de sinal, em *dB*, presentes nas entradas e saídas da placa, além de controlar parâmetros específicos para utilização em música, como *Mute*, *Solo*, *Stereo Link*, etc. No sistema proposto, a única aba de interesse é a aba *Hardware Settings*. Essa aba possibilita ao usuário informar os níveis máximos e mínimos de sinal nas saídas e entradas da placa, além de configurar outros parâmetros como frequência de amostragem e fonte de sincronismo da placa. A Fig. 3.9 mostra os detalhes da aba *Hardware Settings*.



**Fig. 3.9**  
**Aba Hardware Settings da interface do drive da placa Delta 1010LT.**

Os campos de interesse nessa aba são os campos *Master Clock*, *Codec Sample Rate* e *Variable Signal Levels*.

O campo *Master Clock*, situado no lado superior direito, se refere ao sincronismo da placa, ou, à fonte de *clock* de trabalho da placa. O parâmetro desse campo é deixado em *Internal Xtal*, como mostrado. Esse parâmetro indica que a placa trabalha de acordo com o *clock* local do processador do computador no qual está ligada a placa.

O campo *Codec Sample Rate*, localizado na parte central superior, se refere à frequência de amostragem usada pela placa. A placa trabalha com frequências discretas de amostragem, que vão desde 8000 Hz até 96000 Hz, no caso mostrado. Como especificado anteriormente, o sistema proposto analisa sinais que tem componentes de frequência de até 3000 Hz, necessitando de uma taxa de aquisição de pelo menos 6000 Hz. Sendo assim, a taxa de aquisição de 8000 Hz já seria suficiente para o propósito do sistema. Entretanto, para obter um maior refinamento e uma melhor digitalização dos sinais, o sistema usa a frequência de amostragem padrão da placa, que é de 44100 Hz. Com essa frequência de amostragem, o sistema é capaz de obter aproximadamente 730 pontos por ciclo da frequência fundamental de 60 Hz. A escolha desta frequência de amostragem também não compromete o uso de memória do sistema, visto que, atualmente, esse é um recurso com ótima disponibilidade. O uso de uma menor taxa de aquisição reduziria o uso de memória, mas, por outro lado, diminuiria o nível de detalhamento dos sinais adquiridos. Assim, no sistema proposto, tem-se:

$$*f_s = 44100 \text{ Hz}$$

O campo *Variable Signal Levels*, localizado na parte central, se refere aos ajustes de níveis de tensão presentes nas entradas e saídas da placa. Para esse fim, temos três opções: *-10dBV*, *Consumer* e *+4dBu*. Esses níveis se referem a padrões usados para gravação de áudio digital. Os níveis de interesse para o sistema proposto são os níveis de entrada (*All Inputs*). Consultando as especificações técnicas no manual da placa, observa-se que, para um padrão de *+4dBu* para as entradas, o nível máximo de sinal que pode ser aplicado à entrada da placa é de *+14,2dBu*, o que equivale a um nível de tensão máximo de *5,6215 V* de pico, ou aproximadamente *4 Vrms* nas entradas da placa (O nível correto é de *3,957 Vrms*). Esse é o parâmetro utilizado no sistema proposto, pois é a situação na qual a placa aceita os maiores valores de tensão possíveis como níveis de entrada. Assim, no sistema proposto, usamos a configuração *+4dBu*, o que garante:

$$*V_{max} \text{ (entrada)} = 4 \text{ Vrms}$$

Os conversores analógico digitais da placa *Delta 1010LT* estão presentes dentro dos *chips AK4524*, que são *Codecs* de *24 bits @ 96 KHz*. Apesar dos conversores poderem operar com resolução de *24 bits*, o sistema proposto usa uma resolução de *16 bits*.

Com essa resolução, tem-se  $2^{16} = 65536$  níveis de quantização e a resolução em volts para o sistema proposto é de aproximadamente  $60 \mu\text{V/nível}$ . Assim, tem-se:

$$*Resolução \text{ em bits} = 16$$

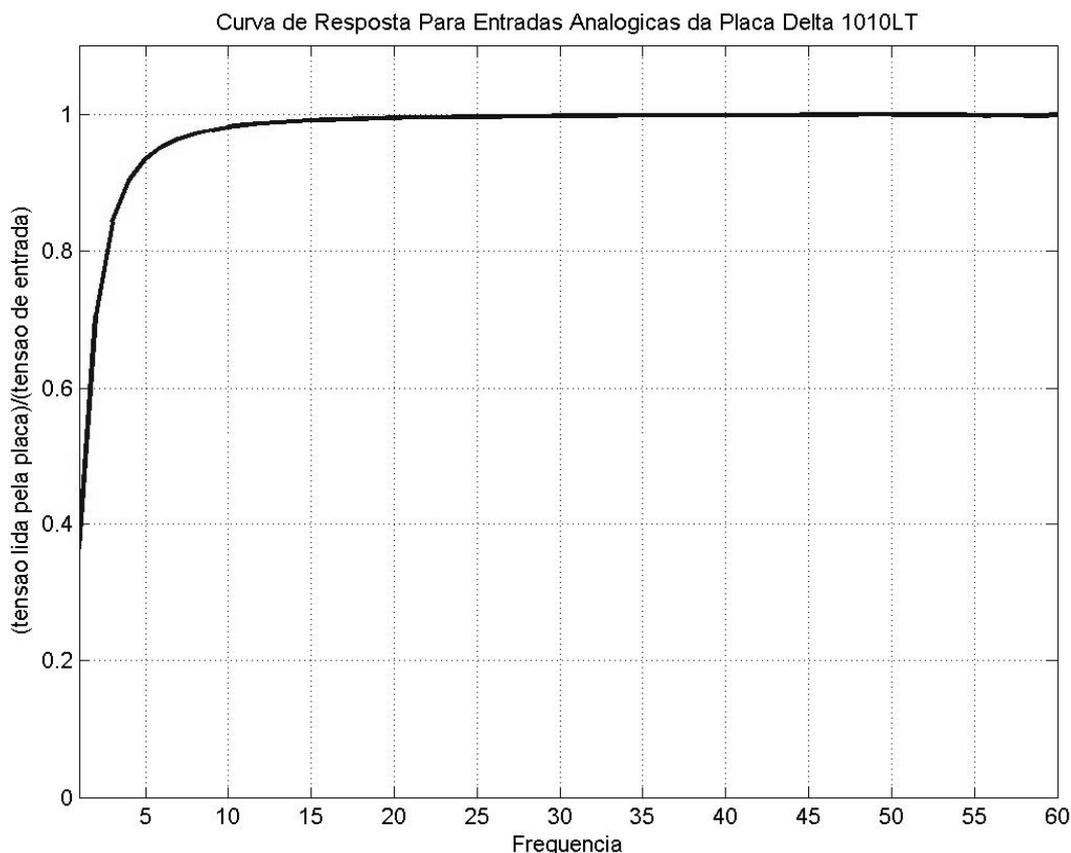
$$*Q = 60 \mu\text{Vrms/nível}$$

Ainda de acordo ainda com as especificações fornecidas no manual da placa, tem-se que a impedância de entrada,  $Z_{in}$ , para as entradas analógicas desbalanceadas é de  $10 \text{ K}\Omega$ . Assim:

$$*Z_{in} = 10 \text{ K}\Omega$$

O circuito da placa é desenvolvido para uso com áudio digital. Os sinais de áudio ocupam uma faixa no espectro que vai de  $20 \text{ Hz}$  a  $20 \text{ KHz}$ . Os sinais abaixo de  $20 \text{ Hz}$  naturalmente sofrem uma atenuação devido aos filtros presentes no circuito de entrada da placa. Isto faz com que as frequências sub-harmônicas e as inter-harmônicas sofram uma atenuação indesejada, fazendo com que as análises sejam incorretas nas frequências abaixo de  $20 \text{ Hz}$ . A Fig. 3.10 apresenta o levantamento da curva característica da placa *Delta 1010LT* para frequências que vão de  $1 \text{ Hz}$  até  $60 \text{ Hz}$ , com passos de  $1 \text{ Hz}$ .

Para obtenção dessa curva, aplicou-se na entrada da placa um sinal senoidal de amplitude definida e frequências na faixa de  $1 \text{ Hz}$  a  $60 \text{ Hz}$ , mediante um gerador de funções modelo *33210A Agilent*. Para cada frequência ajustada, foi medido o nível de sinal lido pela placa, o qual foi lido pela placa a partir de uma rotina computacional escrita em *Matlab*, exibindo o resultado na área de trabalho. Esta rotina usa funções específicas para aquisição de dados, disponíveis no *Toolbox Data Acquisition*. A partir da amplitude definida no gerador de funções e dos valores lidos pela placa, se expressa a relação *tensão lida pela placa / tensão de entrada* mostrada no gráfico. A tensão na frequência de  $60 \text{ Hz}$  é tida como referência de normalização, ou tensão base, pois é a frequência fundamental de operação do sistema de aquisição proposto.



**Fig. 3.10**  
**Curva característica da placa Delta 1010LT para frequências de 1 Hz a 60 Hz.**

Observa-se que, para frequências abaixo dos 20 Hz, a placa lê uma tensão que é menor do que a que realmente é aplicada à sua entrada, uma vez que a relação citada é menor que um. Isso confirma o efeito de atenuação do sinal em baixas frequências produzido pelo circuito de entrada da placa. Como foi dito anteriormente, esse efeito é indesejável no sistema, e, sendo assim, uma *linearização* é necessária para sua correção.

Os fatores de linearização, assim como os valores de tensão lidos pela placa durante sua caracterização são mostrados na Tabela 2.2.

Como pode ser observado, a amplitude do sinal aplicado foi de 3,0 Vrms, valor de tensão lido na frequência de referência de 60 Hz. Com o decréscimo da frequência, as amplitudes lidas pela placa vão gradativamente sofrendo uma atenuação. É importante destacar aqui que a amplitude do sinal na entrada da placa permanece constante, apenas a amplitude lida pela placa é que sofre atenuação. O efeito se torna mais intenso abaixo dos 20 Hz, região na qual, segundo as especificações do fabricante, a placa não opera adequadamente.

O valor da constante de correção corresponde ao valor pelo qual um sinal de frequência especificada deveria ser multiplicado para que sua amplitude fosse lida da mesma forma como se este tivesse frequência de 60Hz. Assim, multiplicando os sinais com as frequências contidas na tabela pelas suas correspondentes constantes de correção, estaremos linearizando a resposta em frequência da placa para as frequências de 1Hz até 60 Hz. Para frequências acima de 60 Hz e menores que 20 KHz, o efeito não ocorre, o que garante que a resposta da placa até os 3000 Hz já é linearizada.

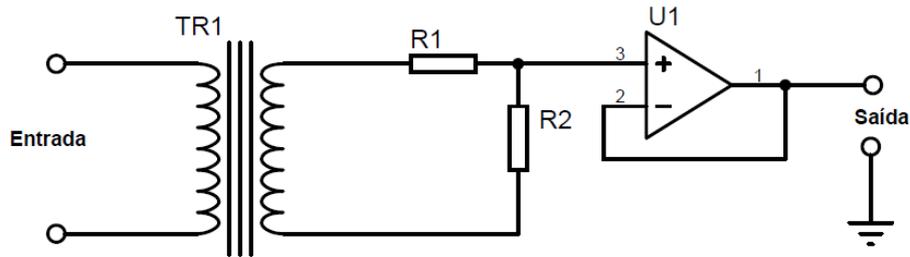
**Tabela 2.2****Fatores de linearização da curva de resposta da Delta 1010LT para frequências de 1 a 60Hz.**

Frequência (Hz)	Tensão (Vrms)	Constante de correção	Frequência (Hz)	Tensão (Vrms)	Constante de correção
1	1,089804	2,752788088	31	2,995018	1,001663390
2	2,119517	1,415416618	32	2,995486	1,001507052
3	2,528147	1,186640008	33	2,995953	1,001350763
4	2,710123	1,106960673	34	2,996189	1,001271996
5	2,804386	1,069752744	35	2,996656	1,001115780
6	2,859120	1,049273900	36	2,996890	1,001037691
7	2,893271	1,036888764	37	2,997124	1,000959613
8	2,916427	1,028655957	38	2,997358	1,000881548
9	2,932800	1,022913207	39	2,997825	1,000725454
10	2,944730	1,018769179	40	2,997825	1,000725454
11	2,953617	1,015703882	41	2,997825	1,000725454
12	2,960869	1,013215976	42	2,998059	1,000647425
13	2,966248	1,011378773	43	2,998293	1,000569408
14	2,970927	1,009785876	44	2,998293	1,000569408
15	2,974435	1,008594816	45	2,998527	1,000491404
16	2,977476	1,007564749	46	2,998760	1,000413412
17	2,980049	1,006694696	47	2,998994	1,000335432
18	2,982153	1,005984480	48	2,999228	1,000257464
19	2,984025	1,005353373	49	2,999228	1,000257464
20	2,985662	1,004802367	50	2,999228	1,000257464
21	2,987317	1,004245523	51	2,999463	1,000178869
22	2,988001	1,004015619	52	2,999697	1,000100925
23	2,988936	1,003701522	53	2,999931	1,000022994
24	2,989873	1,003386978	54	3,000165	0,999945075
25	2,990808	1,003073274	55	2,997591	1,000803495
26	2,991510	1,002838125	56	2,994551	1,001819777
27	2,992211	1,002603086	57	2,994551	1,001819777
28	2,992679	1,002446455	58	2,994551	1,001819777
29	2,993148	1,002289230	59	2,994784	1,001741577
30	2,994317	1,001897989	60	3,000000	1,000000000

Assim, a caracterização da placa fica concluída. Como referência rápida, podemos repetir aqui algumas especificações de interesse:

\*Resolução em bits = 16  
 \*Vmax (entrada) = 4 Vrms  
 \*Q = 60  $\mu$ Vrms/nível  
 \*Z<sub>in</sub> = 10 K $\Omega$   
 \*f<sub>s</sub> = 44100 Hz

Os circuitos de condicionamento de sinais devem respeitar os limites de entrada da placa. Por exemplo, para análise de sinais de tensão com amplitudes da ordem de centenas de volts, os circuitos condicionadores devem prover a isolação necessária assim como efetuar a redução do nível de tensão para que a placa receba, no máximo, 4 Vrms na entrada. Além disso, o casamento de impedâncias do circuito condicionador e da entrada da placa deve também ser observado. A Fig.3.11 mostra um exemplo de circuito condicionador para sinais de tensão que poderia ser aplicado ao sistema proposto.



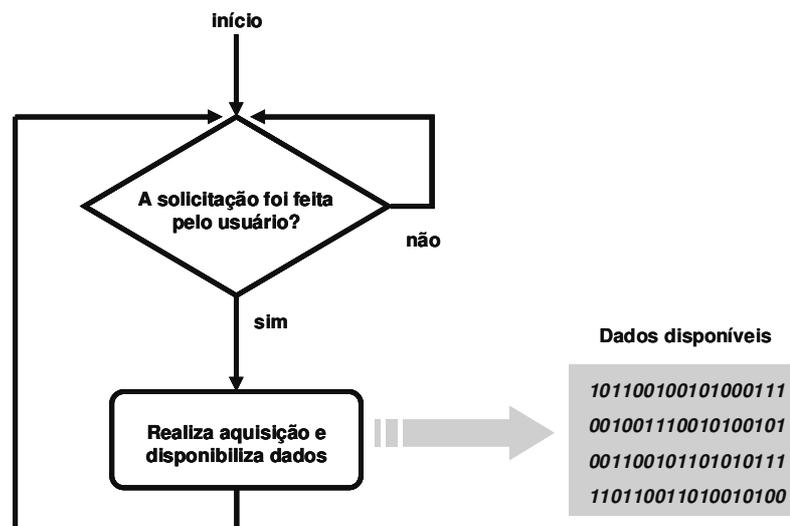
**Fig. 3.11**  
**Exemplo de circuito condicionador para sinais de tensão.**

O sinal de tensão, da ordem de centenas de volts, é aplicado à entrada do circuito. O transformador TR1 provê isolamento entre a rede elétrica e o circuito da placa. Os resistores R1 e R2 formam um divisor de tensão com a finalidade de obter uma redução da tensão do secundário do transformador, obtendo assim uma imagem reduzida e em fase da tensão do primário do transformador. Finalmente, para evitar efeitos de descasamento de impedâncias entre o divisor de tensão e a entrada da placa, usa-se o amplificador operacional U1 na configuração de *buffer*, que apresenta alta impedância de entrada, não interferindo na tensão sobre R2, e baixa impedância de saída. O sinal de saída é então direcionado à entrada da placa.

Os circuitos de condicionamento para sinais de corrente devem apresentar as mesmas características acima, acrescido do fato de que o circuito deve ser ainda capaz de prover uma transformação corrente-tensão, pois a placa só aceita sinais de tensão em suas entradas.

A caracterização do *software* consiste na caracterização do algoritmo empregado para a aquisição e o tratamento dos dados.

O algoritmo de aquisição e análise dos dados no *MatlabScope* se baseia em uma aquisição única, solicitada pelo usuário. Nesta aquisição é obtido um conjunto de dados referente a cada sinal observado (tensões e correntes), os quais serão posteriormente analisados. Só serão realizadas novas aquisições mediante novas solicitações do usuário. No caso de uma nova aquisição ser solicitada, os dados referentes à aquisição anterior serão substituídos pelos novos. A Fig. 3.12 mostra o fluxograma do processo de aquisição dos dados.



**Fig. 3.12**  
**Fluxograma de aquisição dos dados.**

No sistema proposto, não estamos interessados em registrar e analisar transientes espúrios. Em outras palavras, estamos interessados em analisar os sinais de tensão e corrente do sistema elétrico em regimes permanente. Assim, o método de aquisição única é suficiente para atender as especificações do sistema. Nesse caso, deve-se adquirir uma quantidade razoável de dados por solicitação, a fim de se observar uma quantidade satisfatória de ciclos dos sinais adquiridos.

Diferentemente de um instrumento de medição e aquisição contínua, como um osciloscópio digital, a aquisição dos dados é independente da solicitação do usuário. Os dados são armazenados em um *buffer* de tamanho finito e as análises são feitas sobre os dados presentes no *buffer* em um determinado instante de tempo. O *buffer* é constantemente alterado (sobrescrito) pela chegada de novos dados provenientes de novas aquisições.

Os dados, no *MatlabScope*, são armazenados sob a forma de vetores coluna. Cada vetor é formado pelos dados referentes a uma grandeza específica, como por exemplo, a tensão na fase A. O tamanho de cada vetor é igual à quantidade adquirida de dados. Por exemplo, a uma taxa de aquisição de 44100 Hz, durante 1 segundo de aquisição, são obtidos 44100 pontos de cada sinal observado. Assim, o tamanho do vetor referente à fase A seria [44100x1] (44100 linhas por uma coluna). O mesmo aconteceria com os demais sinais. O tempo de 1 segundo é o tempo de aquisição usado no algoritmo do *MatlabScope* para cada solicitação de aquisição.

No vetor de dados de cada sinal, a posição específica do dado é também importante, pois retrata exatamente o instante no qual aquele dado foi adquirido. Por exemplo, o décimo elemento do vetor referente à tensão na fase A equivale à tensão instantânea obtida 10 períodos de amostragem após a aquisição ter se iniciado. Assim, os vetores além de possuírem tamanho específico, possuem também uma ordem na forma como os dados são dispostos dentro de si.

Os elementos dos vetores são também escalonados de acordo com o circuito de condicionamento de sinais. A placa pode receber no máximo sinais de 4 Vrms; porém, os dados devem representar o valor real da grandeza que se está medindo, como por exemplo, centenas de volts, ou dezenas de ampères, etc. Isso é feito pelo próprio algoritmo do *MatlabScope*, e é chamado de escalonamento dos dados.

### 3.3 Justificativa do Sistema Proposto

O uso de uma placa de som em um sistema de aquisição de dados como o sistema proposto se justifica pelo baixo custo do *hardware* empregado em relação a outros *hardwares* específicos de aquisição de dados. O efeito indesejável apresentado pela placa, como por exemplo, a atenuação em baixas frequências, foi solucionado mediante uma linearização realizada por *software*. A linearização consiste em amplificar adequadamente as componentes de frequência abaixo de 20 Hz. Na verdade, a linearização ocorre para frequências abaixo de 60 Hz, como foi visto na caracterização do sistema.

A resolução de 16 *bits* da placa, expansível até 24 *bits*, também garante erros reduzidos na precisão das medidas, conforme será visto mais adiante.

O sistema é ainda flexível no que diz respeito ao *hardware* empregado. A placa de som utilizada pode ser substituída por outro *hardware* de aquisição, bastando, para isso, efetuar a mudança de algumas linhas de código no *software* do sistema.

# 4. Desenvolvimento do Software

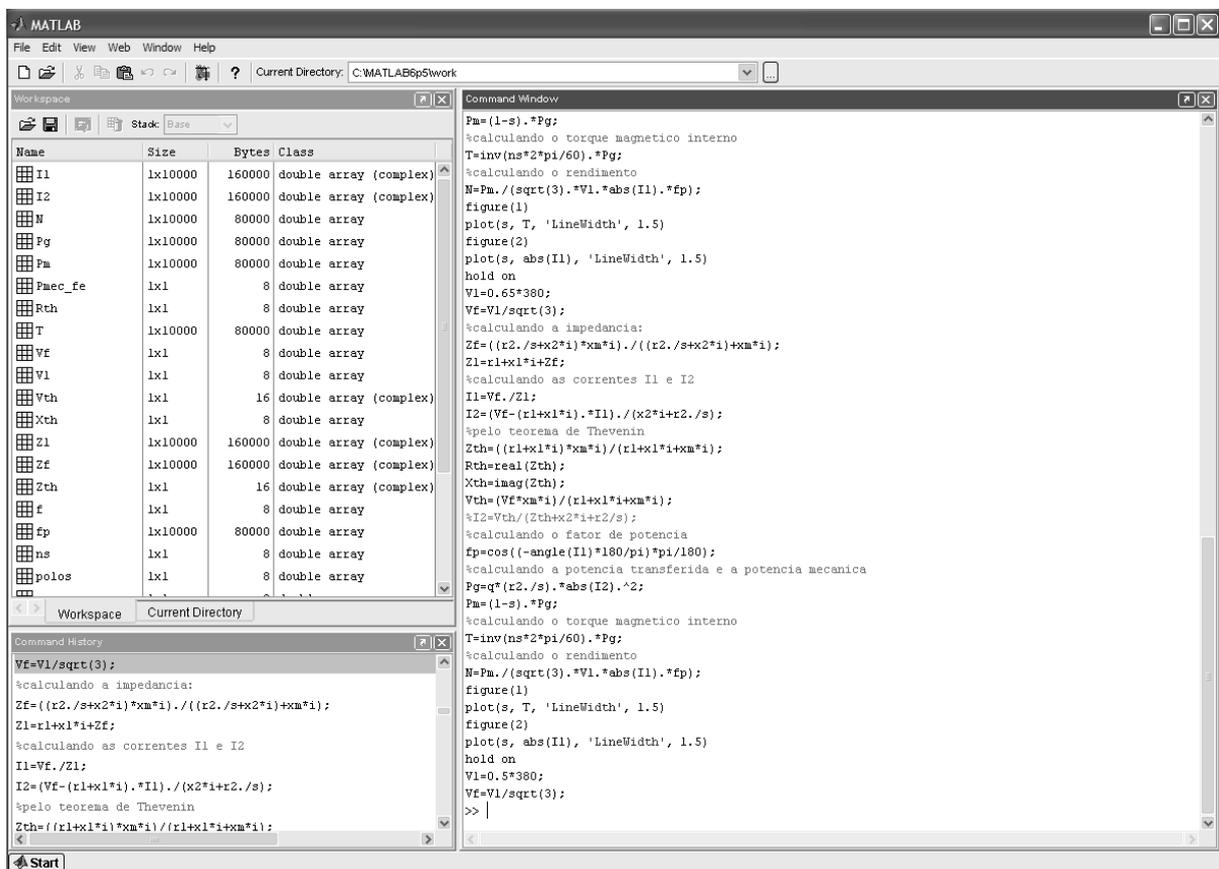
## 4.1 A linguagem Matlab

O *software* de controle do sistema foi desenvolvido utilizando a linguagem *Matlab* (*Matrix Laboratory*).

*Matlab* é um *software* para computação numérica e visualização de alto desempenho, fácil de ser usado, onde os problemas e soluções são expressos quase que da mesma forma como são escritos no papel.

O *Matlab* integra análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambiente de fácil utilização, onde problemas e soluções são expressos como eles são escritos matematicamente, ao contrário da programação tradicional. Ele contém centenas de comandos para as mais diversas finalidades.

A Fig. 4.1 mostra o ambiente de trabalho do *Matlab 6.5*, versão utilizada no desenvolvimento do sistema.



**Fig. 4.1**  
**Ambiente de trabalho do Matlab 6.5.**

O ambiente de trabalho é formado basicamente por três janelas principais. A grande área à direita na figura é a janela de comandos (*Command Window*). É onde o usuário digita os dados de entrada e usa os comandos disponíveis para o tratamento desses dados. No lado esquerdo, na parte superior, tem-se a área de trabalho (*Workspace*), onde são mostrados os tipos, os nomes e os tamanhos das variáveis atualmente usadas pelo usuário. Desse mesmo lado, na parte inferior, tem-

se a janela de histórico (*Command History*). Ela mostra um histórico dos comandos utilizados pelo usuário.

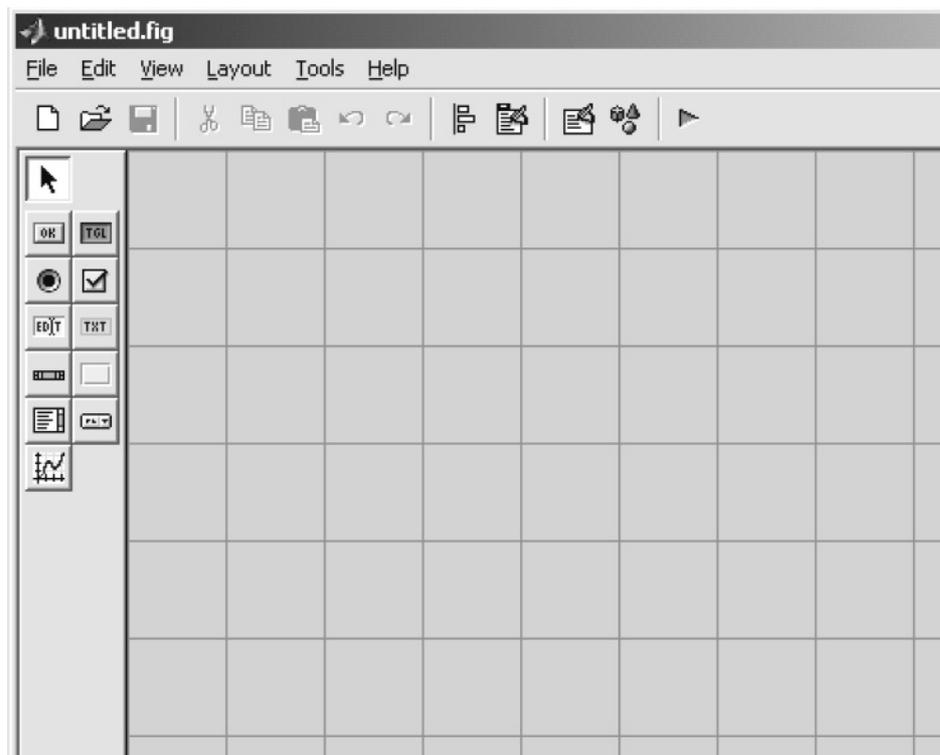
O *Matlab* possui ainda uma família de aplicativos específicos (*Toolboxes*), que são coleções de funções usadas para resolver determinados problemas tais como: otimização, manipulação algébrica, redes neurais, processamento de sinais, simulação de sistemas dinâmicos, entre outros. A principal *Toolbox* usada no desenvolvimento do software do *MatlabScope* é a *Toolbox Data Acquisition*, que possui comandos específicos para aquisição e geração de sinais analógicos e digitais.

As funções da *toolbox* podem ser executadas a partir da linha de comando, como mostrado na Fig. 4.1, ou através de programas. Os programas, no *Matlab*, são *scripts* (arquivo de texto), chamados *MFILE*, que trazem, de forma ordenada, os comandos e funções que devem ser executadas sobre um conjunto de dados.

Outra opção de programação no *Matlab* é a programação a partir de interfaces gráficas de usuário (*GUI – Graphic User Interface*). As interfaces gráficas permitem que usuários não familiarizados com a sintaxe e com as regras de programação do *Matlab* usem essa ferramenta para análise de dados.

As interfaces gráficas no *Matlab* são construídas a partir da ferramenta *GUIDE* (*Graphic User Interface Development Environment*), que é um ambiente de desenvolvimento de interfaces gráficas de usuário. Uma interface gráfica nada mais é do que uma figura com elementos de interface como menus, ícones, botões, telas, etc.

A Fig. 4.2 mostra a tela do *Guide* no *Matlab*.

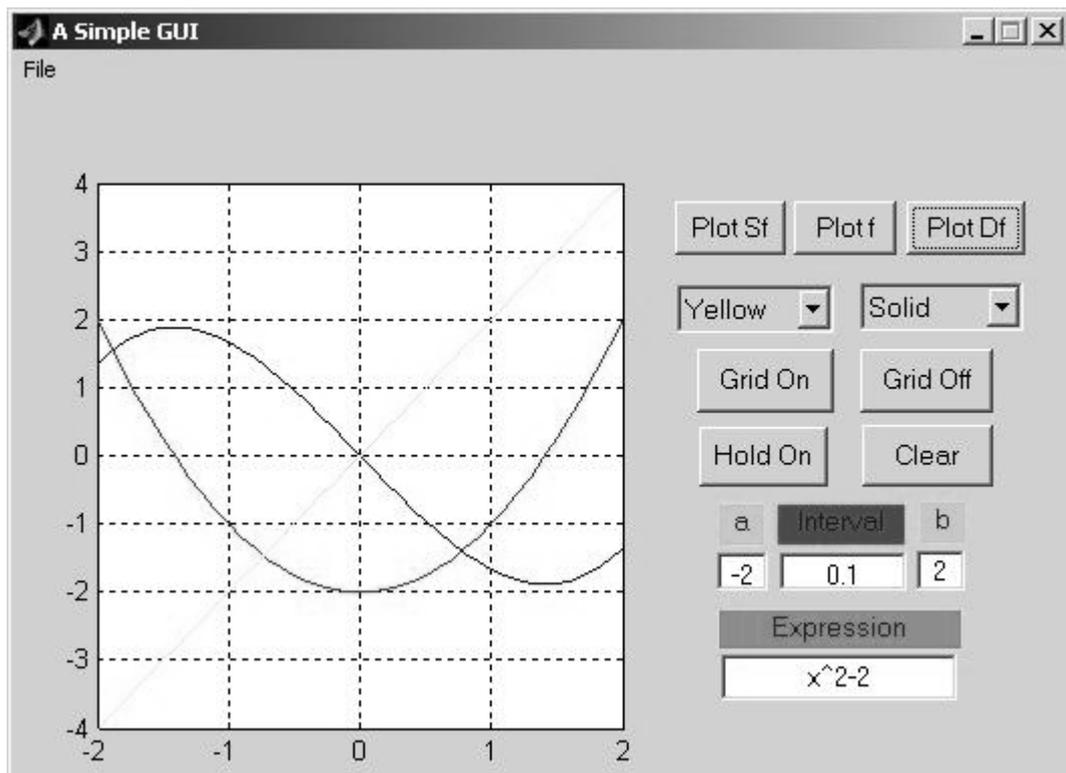


**Fig. 4.2**  
**Tela do Guide.**

A região quadriculada é o que se chama de palco, onde se desenvolve a interface. Do lado esquerdo, temos as ferramentas de desenvolvimento de interface gráfica. Essas ferramentas consistem em botões, campos de textos estáticos, campos de textos editáveis, marcadores, telas de exibição de gráficos, etc.

O desenvolvimento de uma interface gráfica consiste inicialmente em desenhar os componentes que se pretende que façam parte da interface. Isso é chamado de *layout* da interface. Nessa fase, a preocupação consiste basicamente em aspectos gráficos como cores, alinhamentos de botões e outros componentes. Cada componente possui um conjunto de propriedades que podem ser editados conforme a necessidade. A edição dessas propriedades pode ocorrer também em tempo de execução. Por exemplo, podemos especificar a cor de um botão como uma *cor C1*, mas que ao clicar com o *mouse* sobre ele, sua cor seja alterada para a *cor C2*. Essa fase produz como resultado um arquivo de imagem, que é o desenho completo da interface.

A Fig. 4.3 mostra um exemplo de interface gráfica criado no *Guide*.



**Fig. 4.3**  
**Uma simples interface gráfica criada no *Guide*.**

Posteriormente à etapa de *layout*, temos à etapa de programação. Essa etapa consiste em escrever programas que traduzam o comportamento da interface. Por exemplo, ao se pressionar um botão nomeado “*Exit Interface*”, deve-se sair da interface, atendendo à requisição do usuário. Os programas, como citado anteriormente, são *scripts* que trazem uma sequência de comandos que devem ser executados quando forem solicitados.

Finalmente, devem-se associar os programas escritos aos componentes da interface. Essa associação é feita a partir da propriedade *Callback* dos objetos gráficos inseridos na interface. Nessa propriedade especificamos o nome do *script* que deve ser executado quando ocorrer uma ação sobre o objeto gráfico associado. Fica assim concluído o processo de desenvolvimento de uma interface gráfica.

Todas as interfaces do *MatlabScope* foram desenvolvidas com o uso do *Guide* e também pela programação dos elementos da interface.

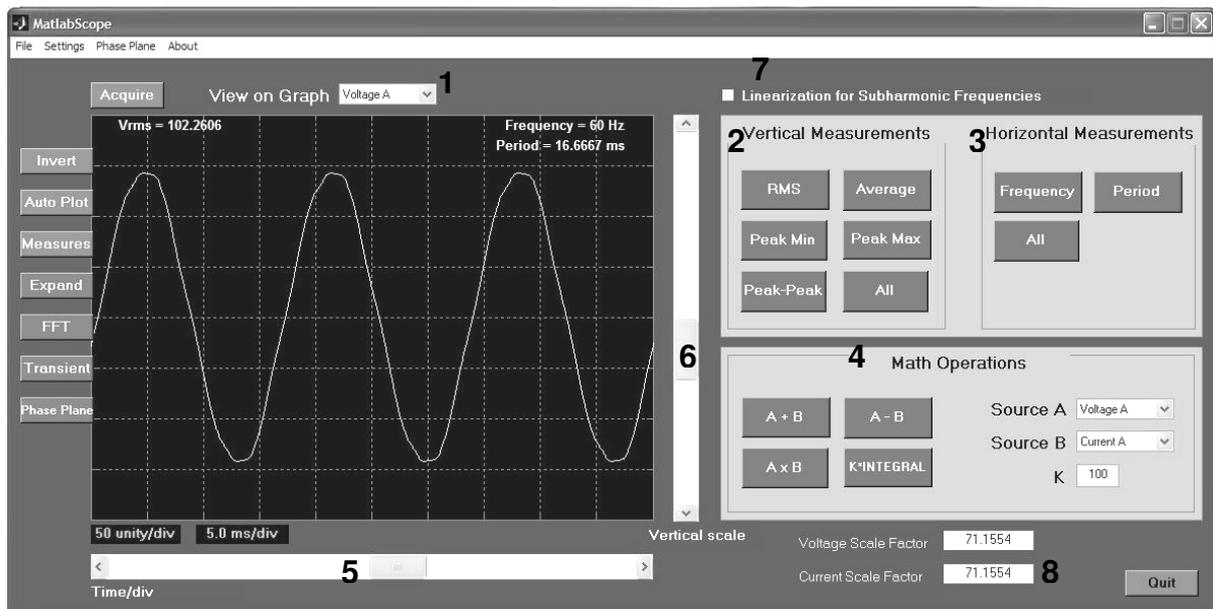
Serão discutidas as funcionalidades de cada interface e como elas interagem com os programas escritos e com o *hardware* do sistema.

## 4.2 Osciloscópio no MatlabScope

Para se iniciar o uso do *MatlabScope*, inicialmente deve-se modificar o diretório de trabalho atual para a pasta *MatlabScope*. Essa mudança é feita na área de trabalho do *Matlab*, especificando no campo *Current Directory*, na barra de ferramentas, o diretório atual da pasta. Essa pasta pode estar em qualquer lugar do computador, à escolha do usuário. Nessa pasta encontram-se todos os programas escritos e que são usados pela interface gráfica, assim como os arquivos de imagem de todas as interfaces. Assim, nenhum arquivo dentro dessa pasta pode ser alterado ou removido.

Com o diretório de trabalho atual alterado para a pasta *MatlabScope*, o usuário deve digitar “MatlabScope” na linha de comando, sem as aspas, que é o nome do primeiro *script* a ser executado. Ao se pressionar a tecla *Enter*, o *Matlab* executará o *script*. Esse é o único contato que o usuário do sistema de aquisição proposto tem com a linha de comandos do *Matlab*.

Nesse *script* temos as funções básicas que estão presentes na *toolbox Data Acquisition* e que inicializam todo o sistema de aquisição. Essas funções são usadas para se especificar o *hardware* utilizado, o número de entradas, os valores limites nas entradas e também abrir a interface principal do *MatlabScope*, que é o Osciloscópio. A Fig. 4.4 mostra a interface principal do *MatlabScope*.



**Fig. 4.4**  
**Interface Principal do MatlabScope.**

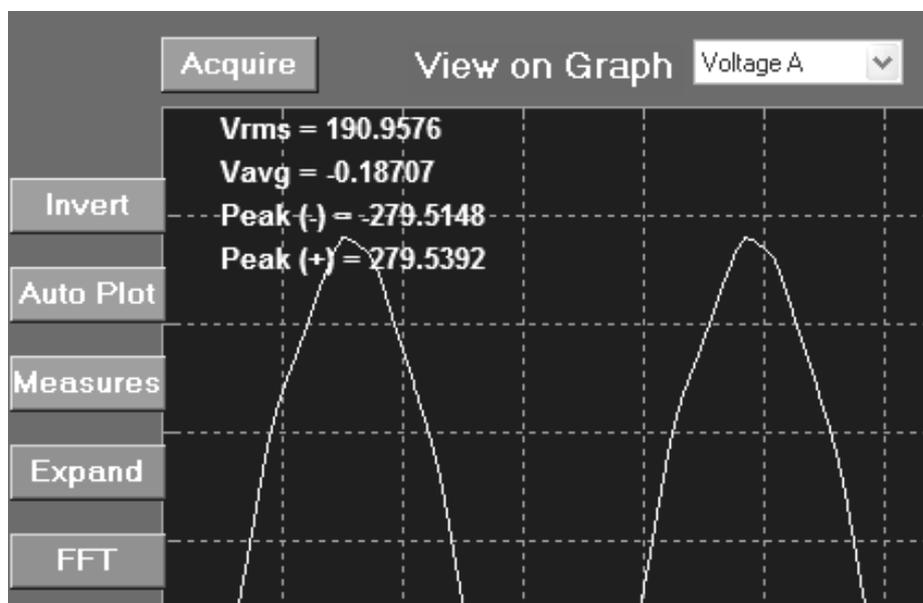
Quando a interface principal é aberta pela primeira vez, o sistema faz uma aquisição prévia de todos os canais e exibe na tela do Osciloscópio o sinal presente na entrada referente à tensão da fase A. A senóide mostrada acima na tela do Osciloscópio é o sinal referente a esta tensão.

O menu do tipo *popup* marcado na Fig. 4.4 da interface (*número 1*) e nomeado como *View on Graph* é usado para escolher qual sinal será exibido na tela do Osciloscópio. O usuário pode escolher qualquer um dos seis sinais disponíveis. Para uma rápida identificação visual do tipo de sinal exibido atualmente na tela, os sinais de tensão são exibidos em amarelo e os sinais de corrente são exibidos em vermelho.

A região marcada com o número 2 (*Vertical Measurements*) agrupa os botões referentes às medidas no eixo vertical do sinal. Como podem ser observadas na figura, essas medidas são:

*RMS*: Calcula o valor *RMS* do sinal exibido na tela;  
*Average*: Calcula o valor médio do sinal exibido na tela;  
*Peak Min*: Calcula o menor valor instantâneo do sinal;  
*Peak Max*: Calcula o maior valor instantâneo do sinal (valor de pico);  
*Peak-Peak*: Calcula o valor de pico a pico do sinal;  
*All*: Calcula todos os valores referentes às medidas do eixo vertical citados acima.

Ao clicar em algum desses botões, o *MatlabScope* executa o *script* apropriado e mostra o resultado do cálculo de forma textual, inserido na tela do Osciloscópio, junto ao sinal que está sendo observado. Isso pode ser visto na Fig. 4.5 que mostra o detalhe das medidas exibidas para um sinal de tensão.

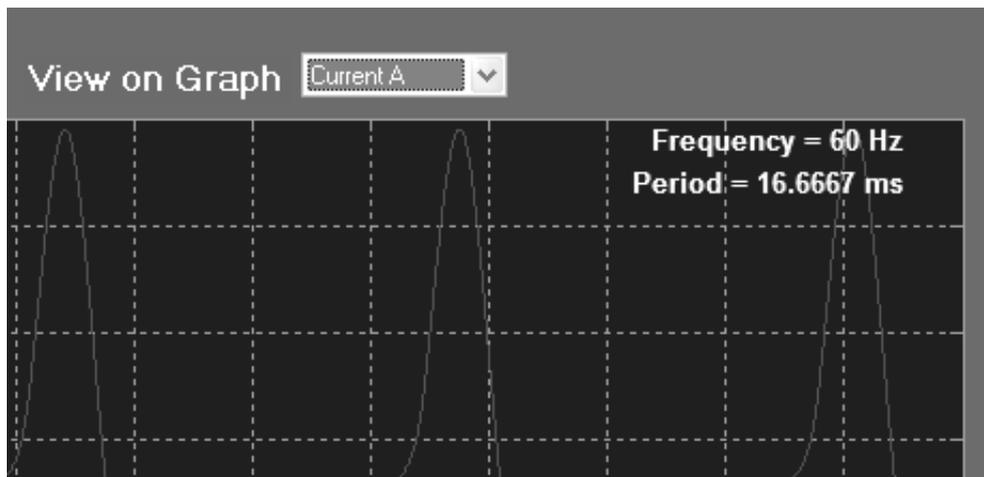


**Fig. 4.5**  
**Detalhe das medidas verticais exibidas para um sinal de tensão.**

A região marcada pelo número 3 (*Horizontal Measurements*) agrupa os botões referentes às medidas no eixo horizontal do sinal. Como podem ser observadas na figura, essas medidas são:

*Frequency*: Calcula a frequência fundamental do sinal exibido na tela;  
*Period*: Calcula o período da componente fundamental do sinal exibido na tela;  
*All*: Calcula todos os valores referentes às medidas do eixo horizontal citados acima.

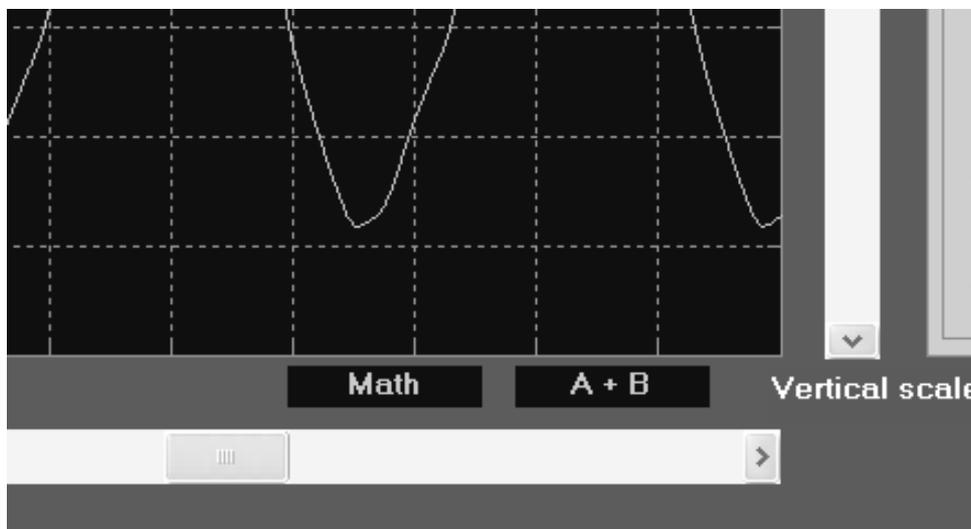
Ao clicar em algum desses botões, o *MatlabScope* executa o *script* apropriado e mostra o resultado do cálculo de forma textual, inserido na tela do Osciloscópio, junto ao sinal que está sendo observado. Isso pode ser visto na Fig. 4.6 que mostra o detalhe das medidas exibidas para um sinal de corrente.



**Fig. 4.6**  
**Detalhe das medidas horizontais exibidas para um sinal de corrente.**

A região marcada pelo número 4 (*Math Operations*) agrupa os botões referentes às operações matemáticas que podem ser realizadas sobre os sinais adquiridos. Os sinais que são usados como operandos (*fontes*) das operações matemáticas são escolhidos nos menus *popup* nomeados como *Source A* e *Source B*. O resultado de uma operação matemática desse tipo é um novo sinal que é exibido na tela do Osciloscópio do *MatlabScope*.

Para uma rápida identificação visual, os sinais resultantes de uma operação matemática são exibidos em verde e, além disso, um indicador de operação matemática e outro que mostra a operação específica são exibidos abaixo da tela do Osciloscópio, do lado direito desta, como pode ser observado na Fig. 4.7.



**Fig. 4.7**  
**Indicadores de operação matemática.**

Como podem ser observadas na Fig. 4.4, as operações disponíveis são:

*A+B*: Calcula e exibe na tela o sinal soma dos sinais das fontes A e B;  
*A-B*: Calcula e exibe na tela o sinal subtração dos sinais das fontes A e B;  
*AxB*: Calcula e exibe na tela o sinal multiplicação dos sinais das fontes A e B;  
*KxINTEGRAL*: Calcula e exibe na tela o sinal integral do sinal da fonte A, multiplicado pela constante K.

Ao clicar em algum desses botões, o *MatlabScope* executa o *script* apropriado e exibe o sinal resultante na tela.

A operação  $K \times INTEGRAL$  só atua sobre o sinal da fonte A, enquanto as demais operações atuam sobre os sinais das duas fontes, A e B. Uma outra característica da operação  $K \times INTEGRAL$  é que ela admite a entrada de um outro parâmetro: a constante  $K$ . Essa constante permite que seja aplicada uma amplificação no sinal resultante para auxiliar na sua visualização, visto que o sinal resultante de uma integração é tanto menor quanto maior for a frequência do sinal. A constante  $K$  é um parâmetro que pode ser alterado pelo usuário no campo de texto editável, nomeado como  $K$ .

Após ter sido realizada uma operação matemática, o usuário pode também usar as medidas no eixo vertical e horizontal sobre o sinal resultante para calcular as grandezas de interesse. Assim, por exemplo, pode-se calcular o valor médio do sinal resultante do produto entre a tensão e a corrente da fase A, obtendo-se assim o valor da potência ativa na fase.

O campo marcado pelo número 5 é o *slider Time/div* de ajuste para visualização do sinal no eixo horizontal. Ele funciona como o controle *Time/div* presente em um osciloscópio clássico. O *slider* permite ajustar o eixo horizontal da tela do Osciloscópio com os seguintes valores:

100 ns/div  
200 ns/div  
500 ns/div  
1 ms/div  
2 ms/div  
5 ms/div  
10 ms/div  
20 ms/div  
50 ms/div  
100 ms/div

A escala horizontal possui 10 divisões. Assim, quando o *slider Time/div* estiver na posição *100 ms/div*, por exemplo, o usuário observa na tela uma janela de tempo de um segundo do sinal.

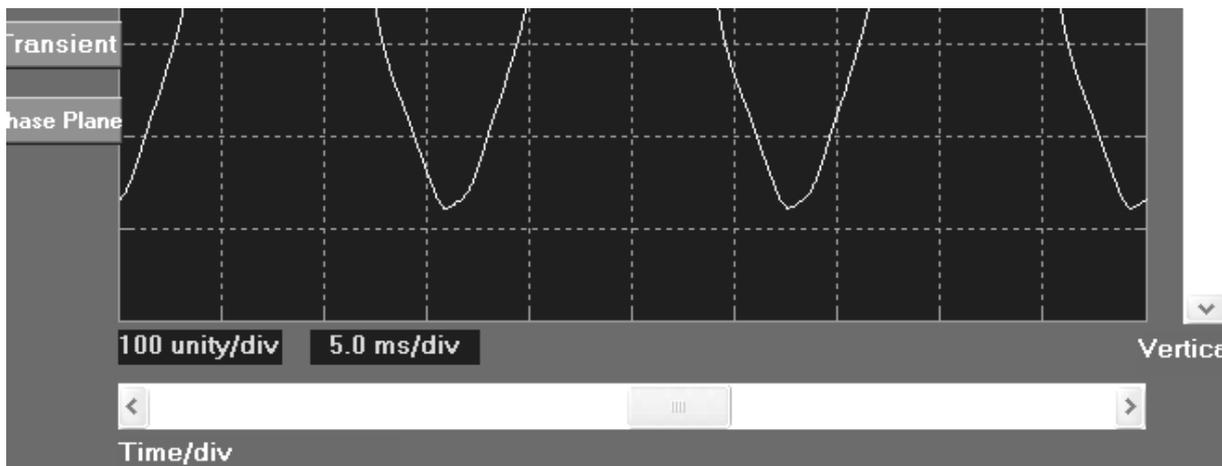
O campo marcado pelo número 6 é o *slider Vertical Scale* de ajuste para visualização do sinal no eixo vertical. Ele funciona como o controle *Volt/div* presente em um osciloscópio clássico. O *slider* permite ajustar o eixo vertical da tela do Osciloscópio com os seguintes valores:

10 unity/div  
20 unity/div  
50 unity/div  
100 unity/div  
500 unity/div

A relação *unity/div* é empregada ao invés de *volt/div*, pois o usuário pode também observar diretamente o valor das correntes medidas.

O eixo vertical possui 8 divisões. Assim, quando o *slider Vertical Scale* estiver na posição *100 unity/div*, por exemplo, o usuário poderá observar um sinal de 800 V de pico-a-pico, ou uma corrente de 800 A de pico-a-pico.

A posição atual dos *sliders* *Time/div* e *Vertical Scale* sempre é mostrada num indicador que fica abaixo da tela do Osciloscópio, do lado esquerdo desta, como pode ser observada na Fig. 4.8.



**Fig. 4.8**  
**Indicadores dos sliders *Vertical Scale* e *Time/div*.**

Para um ajuste rápido do sinal na tela do Osciloscópio, o usuário pode clicar sobre o botão *Auto Plot*, presente no conjunto de botões dispostos verticalmente do lado esquerdo da tela. Ao clicar nesse botão, é executado um *script* que calcula o período e os valores de pico do sinal sob observação. Com base nesses valores, os *sliders* *Time/div* e *Vertical Scale* são automaticamente ajustados para garantir a melhor visualização possível do sinal na tela.

Nesse mesmo conjunto de botões estão presentes os botões *Invert* e *Measures*. O botão *Invert* produz uma inversão de 180° no sinal, enquanto que o botão *Measures* exibe na tela todas as medidas dos eixos vertical e horizontal.

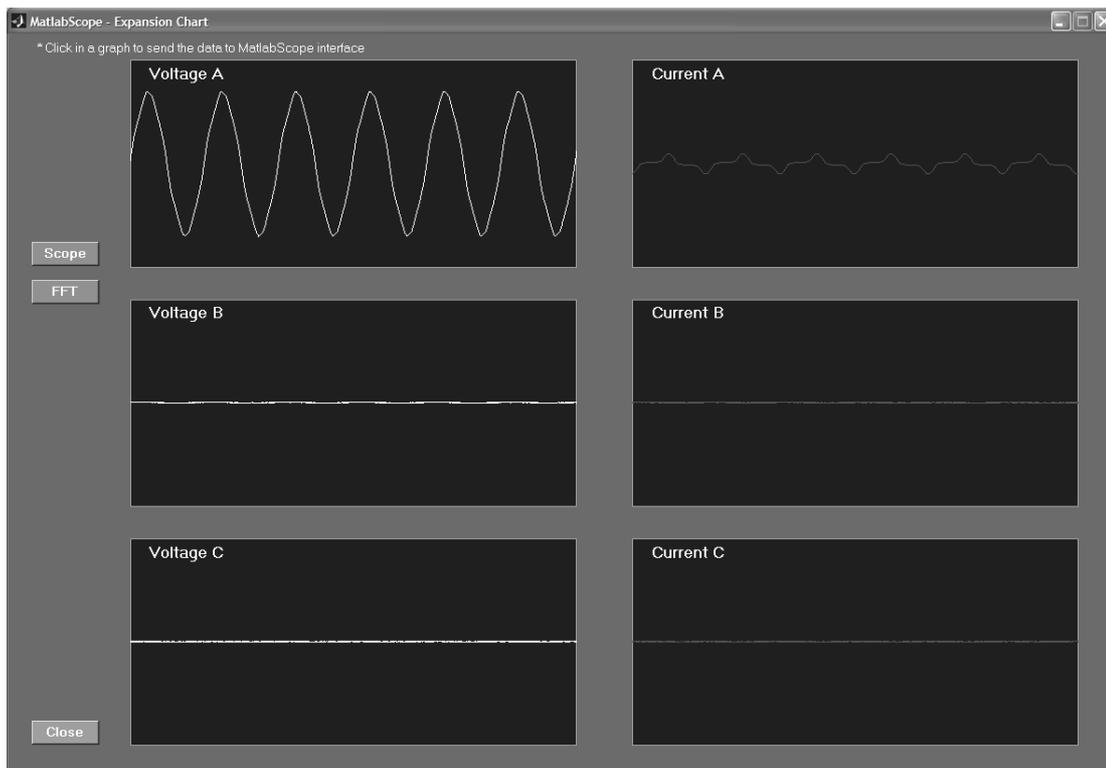
Ainda nesse conjunto de botões, estão presentes os botões *Expand*, *FFT*, *Transient* e *Phase Plane*.

O botão *Expand* abre uma tela de expansão (*expansion chart*), onde podem ser simultaneamente observados todos os sinais de tensão e corrente. Essa Tela de Expansão é mostrada na Fig. 4.9. Ao clicar, nessa tela, em qualquer um dos sinais exibidos, o sinal é automaticamente levado à interface do Osciloscópio para uma análise mais detalhada. Se o usuário desejar voltar à interface do Osciloscópio basta clicar no botão *Scope*, mostrado do lado esquerdo da tela de expansão, ou simplesmente clicar no botão *Close* para fechar esta tela.

O botão *FFT*, presente na interface do Osciloscópio, e também na Tela de Expansão, abre diretamente a interface do analisador de espectro (*Spectrum Analyzer*). As funcionalidades dessa interface serão discutidas em seguida, na seção 4.3.

O botão *Transient* abre diretamente a interface do gravador de transientes (*Transient Recorder*). As funcionalidades dessa interface serão discutidas em seguida, na seção 4.4.

O botão *Phase Plane* abre diretamente uma interface com o mesmo nome do botão. As funcionalidades dessa interface serão discutidas em seguida, na seção 4.5.



**Fig. 4.9**  
**Tela de expansão**

O botão *Acquire*, situado no lado superior esquerdo da tela do Osciloscópio é usado, a qualquer instante, para se fazer uma nova aquisição dos sinais. Ao ser clicado, sua cor é modificada para vermelho, indicando que o *MatlabScope* está em processo de aquisição, permanecendo assim durante o tempo de aquisição (*um segundo*).

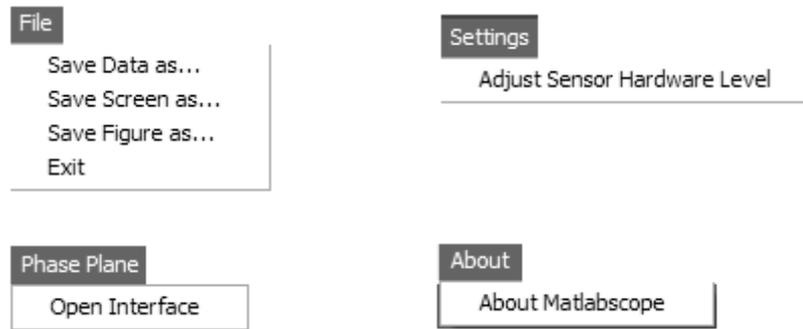
O campo marcado pelo número 7, e nomeado como *Linearization for Subharmonic Frequencies*, é um *checkbox* que faz com que os dados adquiridos quando se pressiona o botão *Acquire* sejam linearizados para frequências inferiores a 60Hz. Assim, marcando-se este *checkbox*, o usuário estará corrigindo a atenuação do sinal para baixas frequências, causada pelas limitações do *hardware* utilizado. Para fazer essa linearização, o *script* associado calcula a *FFT* (*Fast Fourier Transform – Transformada Rápida de Fourier*) do sinal e multiplica cada frequência, de 1 a 60Hz, pelas constantes mostradas na Tabela 2.2. Em seguida, o *script* recupera o sinal corrigido calculando a *FFT inversa* e exibe na tela o sinal corrigido.

No canto superior da interface do Osciloscópio encontra-se a barra de menus. A estrutura de menus no *MatlabScope* trata de operações como salvamento de arquivos, abertura de novas interfaces, ajuste de sensores empregados no sistema, entre outras.

Como pode ser visto na Fig. 4.4, a barra de menus possui os seguintes itens:

*File*: menu com opções de salvamento de dados e imagens;  
*Settings*: menu de ajuste de sensores empregados no sistema;  
*Phase Plane*: menu de abertura da interface *Phase Plane*;  
*About* : menu de informações sobre o desenvolvimento do *MatlabScope*

A Fig. 4.10 mostra os submenus referentes a cada menu acima citado.



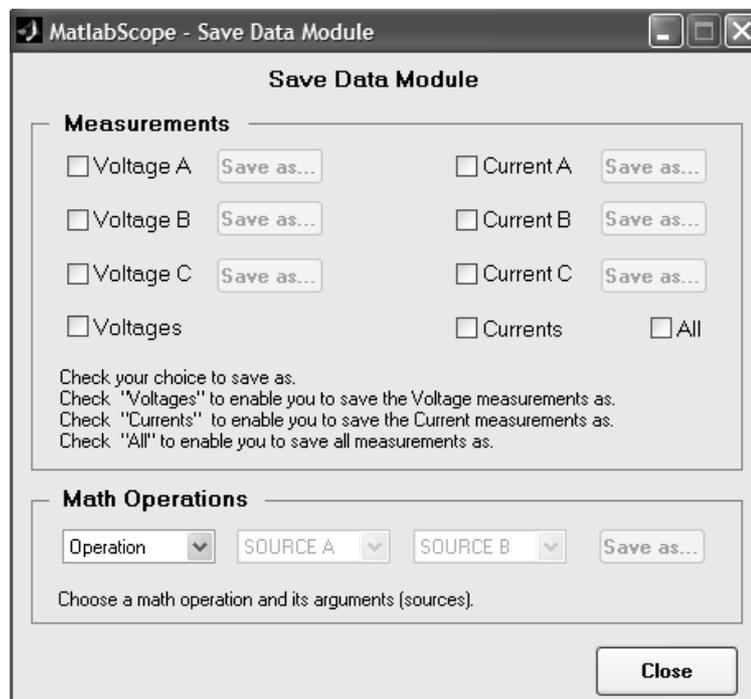
**Fig. 4.10**  
**Submenus da barra de menus do MatlabScope.**

O menu *File* possui quatro submenus.

Ao clicar no submenu *Save Data as...* é aberto o módulo de salvamento de dados (*Save Data Module*) mostrado na Fig. 4.11. Esse módulo permite ao usuário salvar os dados adquiridos da última aquisição feita. Os dados são salvos em um arquivo com extensão *.DAT* que pode posteriormente ser utilizado por outro *software* de tratamento de dados. O usuário deve escolher quais os dados a serem salvos. Para isso, basta clicar nos *checkboxes* associados a cada sinal para que o botão *Save as...* seja habilitado. Clicando no botão *Save as...* é aberta a caixa de diálogo de salvamento que leva o usuário a escolher um nome de arquivo e o diretório de destino.

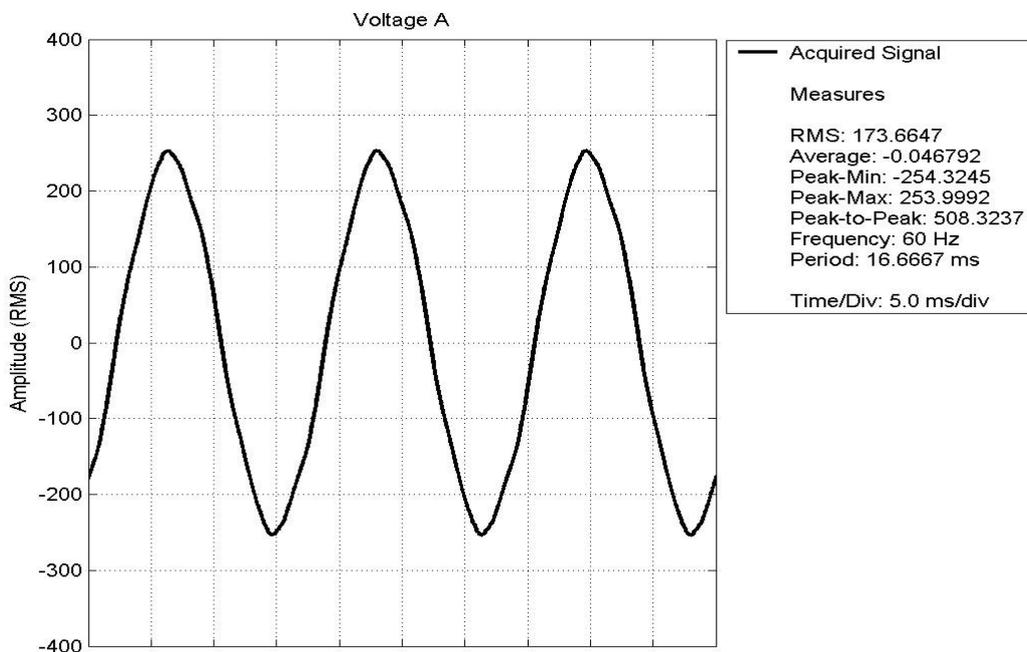
Cada sinal salvo gera um arquivo de dados. Cada arquivo de dados é composto por 32000 pontos, no formato  $(x,y)$ , onde  $x$  é o tempo e  $y$  é o valor instantâneo do sinal.

Além dos sinais de tensão e corrente, o usuário também pode salvar um arquivo de dados resultante de uma operação matemática realizada sobre as fontes A e B, mostradas também na Fig. 4.11.



**Fig. 4.11**  
**Módulo de salvamento de dados**

Ao clicar no submenu *Save Screen as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão .JPEG que mostra o sinal presente na tela do Osciloscópio assim como as grandezas medidas nos eixos vertical e horizontal, como mostrado na Fig. 4.12. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.



**Fig. 4.12**  
Arquivo de imagem obtido do submenu *Save Screen as...*

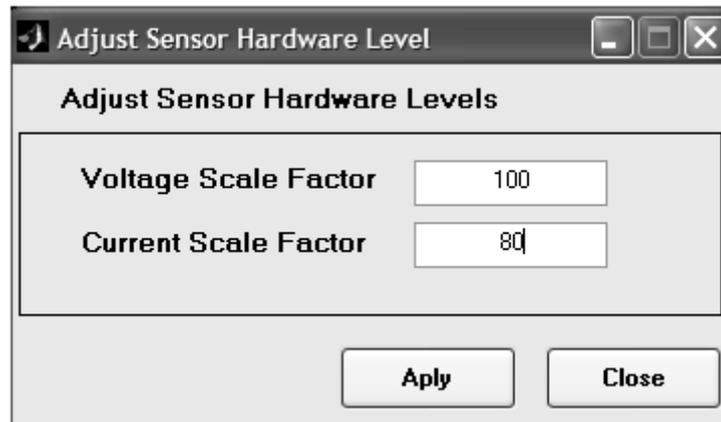
Ao clicar no submenu *Save Figure as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão .JPEG que mostra toda a interface do Osciloscópio. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.

O submenu *Exit* é usado para encerrar a sessão de aquisição de dados e fechar todas as interfaces abertas.

O menu *Settings* possui um único submenu: *Adjust Sensor Hardware Level*. Ao clicar nesse submenu, é aberta a caixa de diálogo de ajuste de níveis dos sensores de corrente e tensão, mostrada na Fig. 4.13. Essa caixa de diálogo permite ao usuário especificar qual a relação de transformação que os sensores de corrente e tensão produzem nos respectivos sinais. Todas as medidas feitas pelo *MatlabScope* se baseiam nos valores fornecidos a esses campos. As medidas feitas pelo sistema estão diretamente ligadas à precisão dos valores fornecidos aos campos *Voltage Scale Factor* (relação de transformação de tensão) e *Current Scale Factor* (relação de transformação de corrente). Se os valores fornecidos não forem corretos, todas as medidas feitas pelo *MatlabScope* também estarão incorretas.

O usuário deve fornecer dois fatores de escala, sendo um referente à tensão e outro à corrente. O fator de escala é a constante pela qual o *MatlabScope* deve multiplicar o sinal de entrada da placa para que as medições realizadas reproduzam com fidelidade os níveis reais do sinal observado (tensão ou corrente). Quando os valores estiverem sido fornecidos, o usuário deve clicar no botão *Apply* para que as alterações tenham efeito no sistema. Os valores atuais dos fatores de escala para

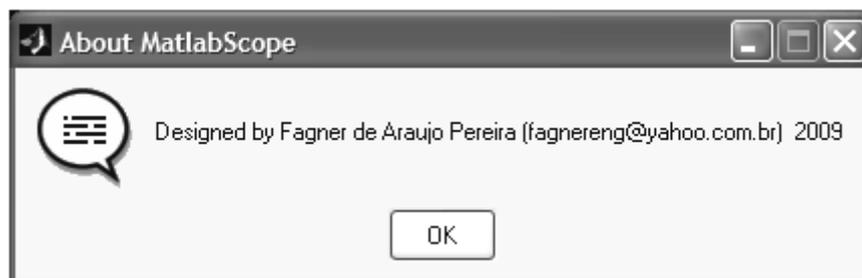
tensão e corrente são sempre exibidos na interface do Osciloscópio, no campo numerado pelo número 8 na Fig. 4.4.



**Fig. 4.13**  
**Caixa de diálogo de ajuste dos níveis dos sensores usados no sistema.**

O menu *Phase Plane* possui um único submenu: *Open Interface*. Ao clicar nesse submenu é aberta a interface do *Phase Plane*, cujas funcionalidades serão discutidas na sessão 4.5.

O menu *About* possui um único submenu: *About MatlabScope*. Ao clicar nesse submenu é aberta uma janela com informações sobre o desenvolvimento do *MatlabScope*, como mostra a Fig. 4.14.

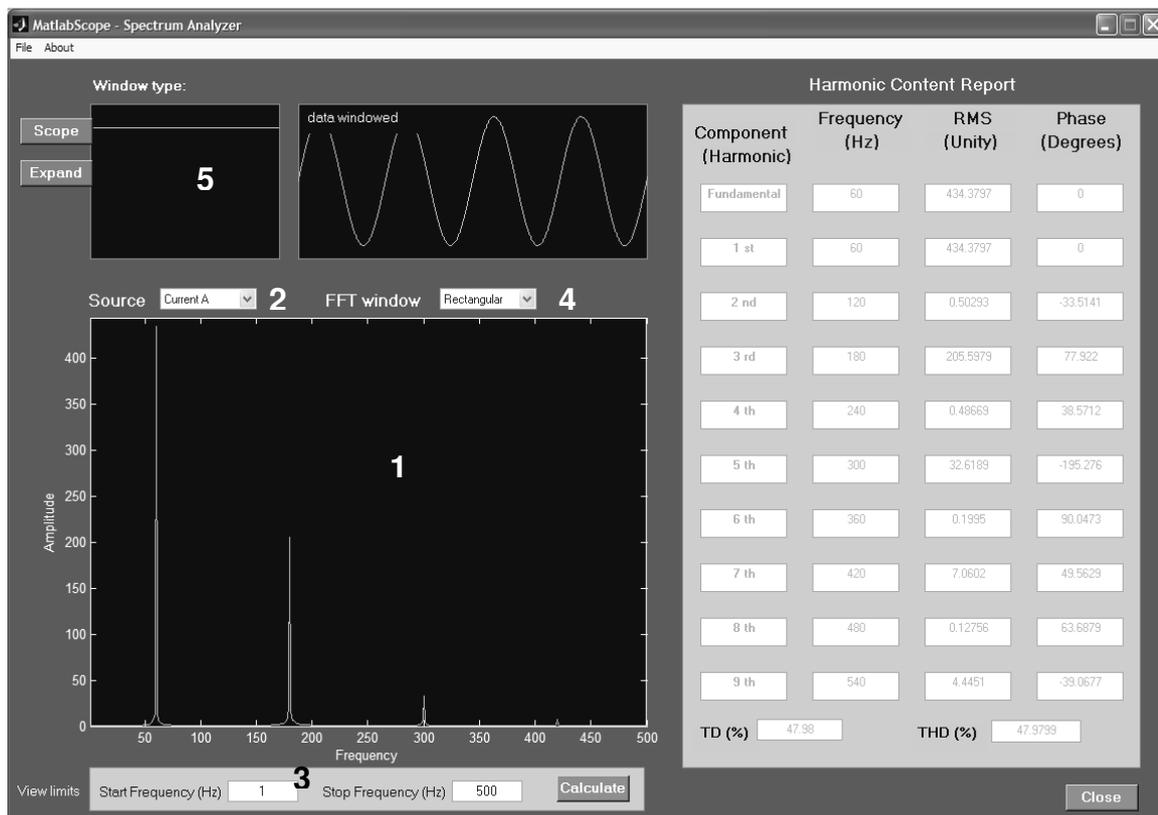


**Fig. 4.14**  
**Janela com informações sobre o desenvolvimento do MatlabScope.**

No canto inferior direito da interface do Osciloscópio há o botão *Quit*, que é usado para encerrar a sessão de aquisição de dados e fechar todas as interfaces abertas.

### 4.3 Analisador de Espectro no MatlabScope

A interface do Analisador de Espectro, aqui chamada de *Spectrum Analyzer*, é responsável pelos cálculos baseados na *FFT* (*Fast Fourier Transform – Transformada Rápida de Fourier*). Ela é aberta ao clicar no botão *FFT* presente na interface do Osciloscópio ou na tela de expansão. A Fig. 4.15 mostra a interface do *Spectrum Analyzer*.



**Fig. 4.15**  
**Interface do analisador de espectro**

A tela marcada pelo número 1 na interface da Fig. 4.15 é a tela principal do *Spectrum Analyzer*. Nela, o usuário pode observar no eixo X as frequências contidas no sinal e no eixo Y o valor de suas amplitudes, em níveis *RMS*. Por exemplo, a figura mostra que o sinal observado contém a frequência fundamental em 60Hz, bem como harmônicas em 180Hz, 300Hz e 420Hz (terceira, quinta e sétima harmônicas, respectivamente).

O usuário pode fazer a análise espectral em qualquer um dos sinais de tensão ou corrente. A escolha da fonte para análise espectral é feita no menu *popup* nomeado como *Source*, marcado pelo número 2 na Fig. 4.15. Após escolhida a fonte, o usuário deve clicar no botão *Calculate*, situado na parte inferior direita da tela principal.

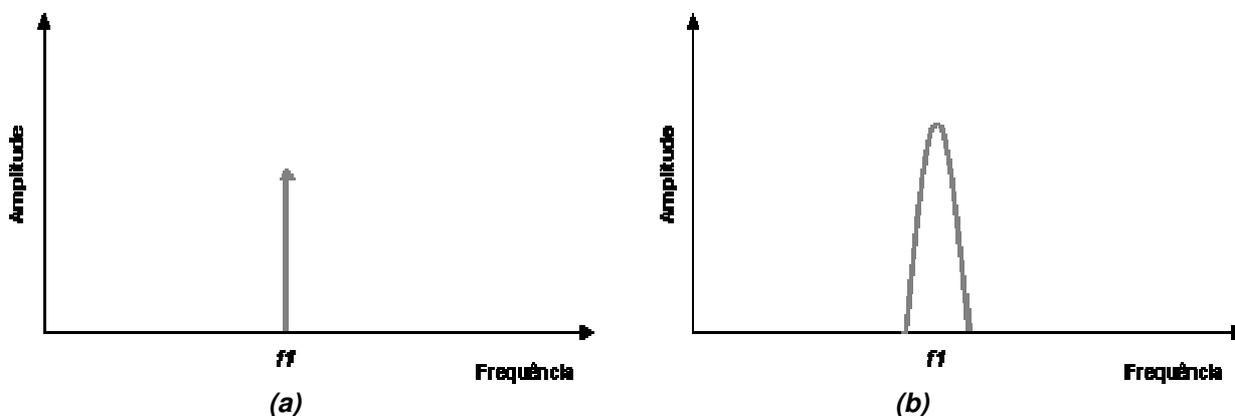
Na figura, podemos observar na tela do *Spectrum Analyzer* que o eixo das frequências mostradas (*eixo X*) vai de 0 a 500 Hz. Esses limites podem ser alterados conforme a necessidade do usuário. A mudança dos limites de visualização é feita nos campos de texto editável nomeados como *Start Frequency* (Frequência de Início) e *Stop Frequency* (Frequência de Parada) marcados na figura pelo número 3. Assim, os limites desse eixo são atualizados para os novos valores, permitindo que o usuário analise faixas específicas de frequências. Os limites de visualização máximos são aqueles já especificados para o sistema (até 3000Hz).

A área intitulada como *Harmonic Content Report* (Relatório de Conteúdo Harmônico), do lado direito da interface, exibe um resumo do conteúdo harmônico do sinal, mostrando os valores *RMS*, frequência e fase em relação à fundamental das componentes até a nona harmônica, além de calcular o valor da taxa de distorção (*TD*) e da taxa de distorção harmônica (*THD*). Esse resumo tem a intenção de apresentar um conjunto de informações básicas para uma análise superficial do sinal. As informações espectrais para toda a faixa disponível podem ser obtidas

através do salvamento dos dados disponibilizados pelo *Spectrum Analyzer* (menu *File/Save data as...*).

No lado superior esquerdo estão os botões *Scope* e *Expand*, que, quando clicados, fazem o usuário retornar à interface do Osciloscópio ou à Tela de Expansão, respectivamente.

Medições baseadas em *FFT*, como as realizadas pelo *Spectrum Analyzer*, estão sujeitas a um erro proveniente de um efeito conhecido como *leakage* (*espalhamento*). Esse efeito é caracterizado pelo espalhamento do espectro em torno das componentes de frequência presentes no sinal. A Fig. 4.16 apresenta os espectros de um mesmo sinal senoidal puro de frequência  $f_1$ . Em (a), o efeito de *leakage* não está presente, uma vez que não há espalhamento de frequências em torno da fundamental, e o espectro é um impulso situado em  $f_1$ . Em (b) o efeito é visível e constatado pelo espalhamento do espectro em torno de  $f_1$ . O efeito de *leakage* promove uma análise errônea a respeito das frequências contidas no sinal, visto que o espectro apresenta frequências que não estão contidas no sinal original.



**Fig. 4.16**  
**Espectro de uma senóide pura de frequência  $f_1$ : (a) sem efeito de leakage; (b) com efeito de leakage.**

O efeito de *leakage* ocorre quando a *FFT* é calculada sobre um bloco de dados que não é periódico. Como não é possível determinar que um conjunto de dados amostrados a partir de um instante qualquer seja periódico, funções especiais devem ser aplicadas sobre o conjunto de dados para torná-lo periódico, a fim de amenizar o erro produzido por esse fenômeno. O usuário deve escolher a função apropriada para uma aplicação específica. Quando a função não é aplicada corretamente, os erros podem ser introduzidos na amplitude da *FFT*, na frequência ou até mesmo na forma do espectro. Cada função é chamada de *janela*. Os sinais são então multiplicados pelas janelas para produzir um sinal periódico.

Muitas janelas foram propostas ao longo do tempo, cada uma com vantagens e desvantagens em relação às outras. Algumas são específicas para alguns tipos de sinal como sinais aleatórios ou senoidais. Algumas melhoram a resolução em frequência, isto é, facilitam a detecção da frequência exata de um pico no espectro. Outras melhoram a precisão na amplitude. A melhor janela deve ser escolhida de acordo com a aplicação.

As janelas disponíveis no *Spectrum Analyzer*, suas vantagens e desvantagens, assim como aplicações são apresentadas na Tabela 4.1.

**Tabela 4.1**  
**Tipos de janela FFT disponíveis no Spectrum Analyzer**

Janela	Melhor para este tipo de sinal	Atenuação do efeito de Leakage	Resolução de frequência	Precisão de amplitude
Retangular (sem janela)	Transitórios e amostragem síncrona	Pobre	Melhor	Pobre
Hanning	Aleatórios	Boa	Boa	Razoável
Hamming	Aleatórios	Razoável	Boa	Razoável
Blackman	Aleatórios ou mixados	Melhor	Pobre	Boa
Blackman-Harris	Aleatórios ou mixados	Boa	Pobre	Razoável
Flat top	Senoidais	Boa	Pobre	Melhor
Bartlett	Aleatórios	razoável	Boa	Razoável

O usuário pode escolher o tipo de *janela* a ser usada no cálculo da *FFT* através do menu do tipo popup nomeado como *FFT Window*, marcado pelo número 4 na Fig. 4.15.

O tipo de *janela* escolhido pelo usuário é mostrado graficamente através do *display* nomeado como *Window Type*, marcado pelo número 5 na Fig. 4.15. Além disso, o *display* nomeado como *data windowed*, que fica do lado direito do *display Window Type*, mostra o efeito da janela escolhida sobre um sinal senoidal puro.

No canto superior da interface do *Spectrum Analyzer* encontra-se a barra de menus. Como pode ser visto na Fig. 4.15, a barra de menus possui os seguintes itens:

*File*: menu com opções de salvamento de dados e imagens;

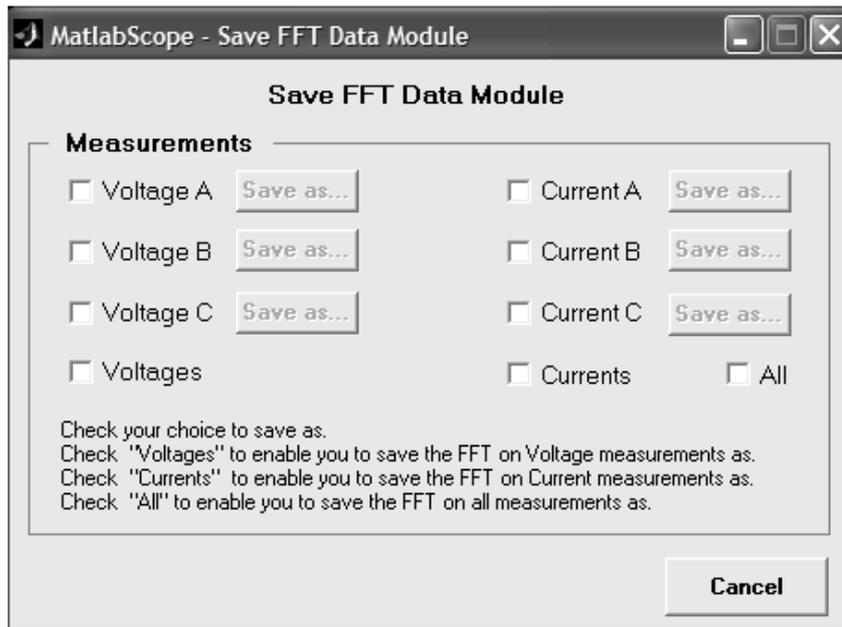
*About* : menu de informações sobre o desenvolvimento do MatlabScope.

O menu *About* é semelhante àquele apresentado na interface do Osciloscópio.

O menu *File* possui quatro submenus, semelhantes àqueles apresentados também na interface do Osciloscópio.

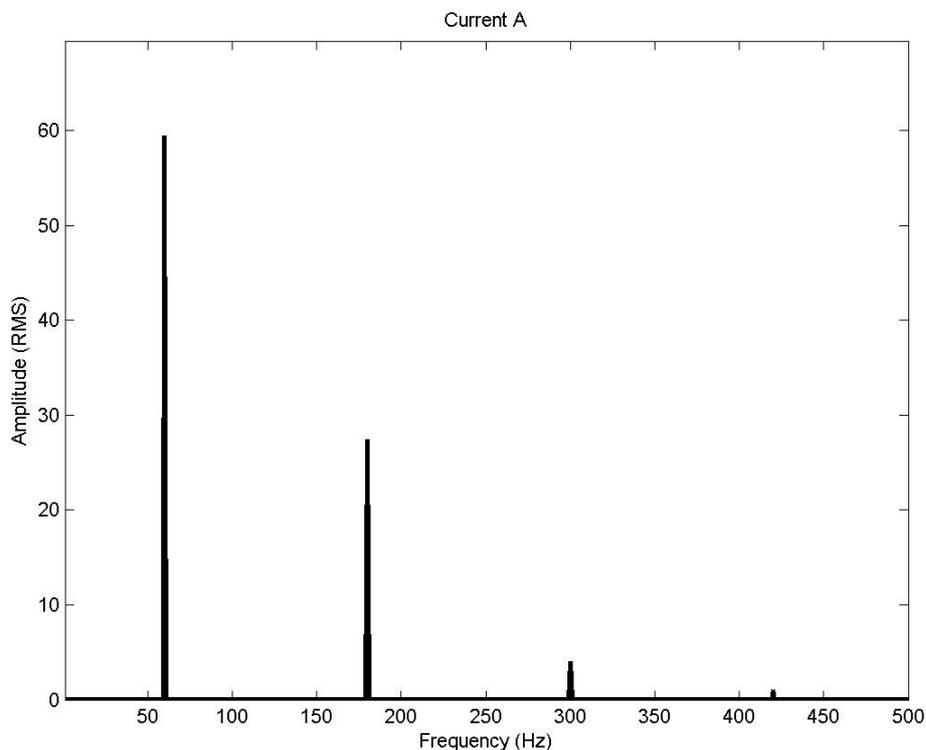
Ao clicar no submenu *Save Data as...* é aberto o módulo de salvamento de dados (*Save FFT Data Module*) mostrado na Fig. 4.17. Esse módulo permite ao usuário salvar os dados referentes à análise espectral dos sinais de interesse. Os dados são salvos em um arquivo com extensão *.DAT* que pode posteriormente ser utilizado por outro software de tratamento de dados. O usuário deve escolher quais os dados a serem salvos. Para isso, basta clicar nos *checkboxes* associados a cada sinal para que o botão *Save as...* seja habilitado.

Clicando no botão *Save as...* é aberta a caixa de diálogo de salvamento que leva o usuário a escolher um nome de arquivo e o diretório de destino. Cada sinal salvo gera um arquivo de dados. Cada arquivo de dados é composto por 3000 pontos, no formato (x,y), onde x é a frequência e y é o valor da amplitude do sinal na frequência x.



**Fig. 4.17**  
**Módulo de salvamento de dados do Spectrum Analyzer**

Ao clicar no submenu *Save Screen as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão .JPEG que mostra o sinal presente na tela do *Spectrum Analyzer* como mostrado na Fig. 4.18. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.



**Fig. 4.18**  
**Arquivo de imagem obtido do submenu *Save Screen as...* do Spectrum analyzer .**

Ao clicar no submenu *Save Figure as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão .JPEG que mostra toda a interface do *Spectrum Analyzer*. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.

O submenu *Exit* é usado para sair e fechar a interface do *Spectrum Analyzer*.

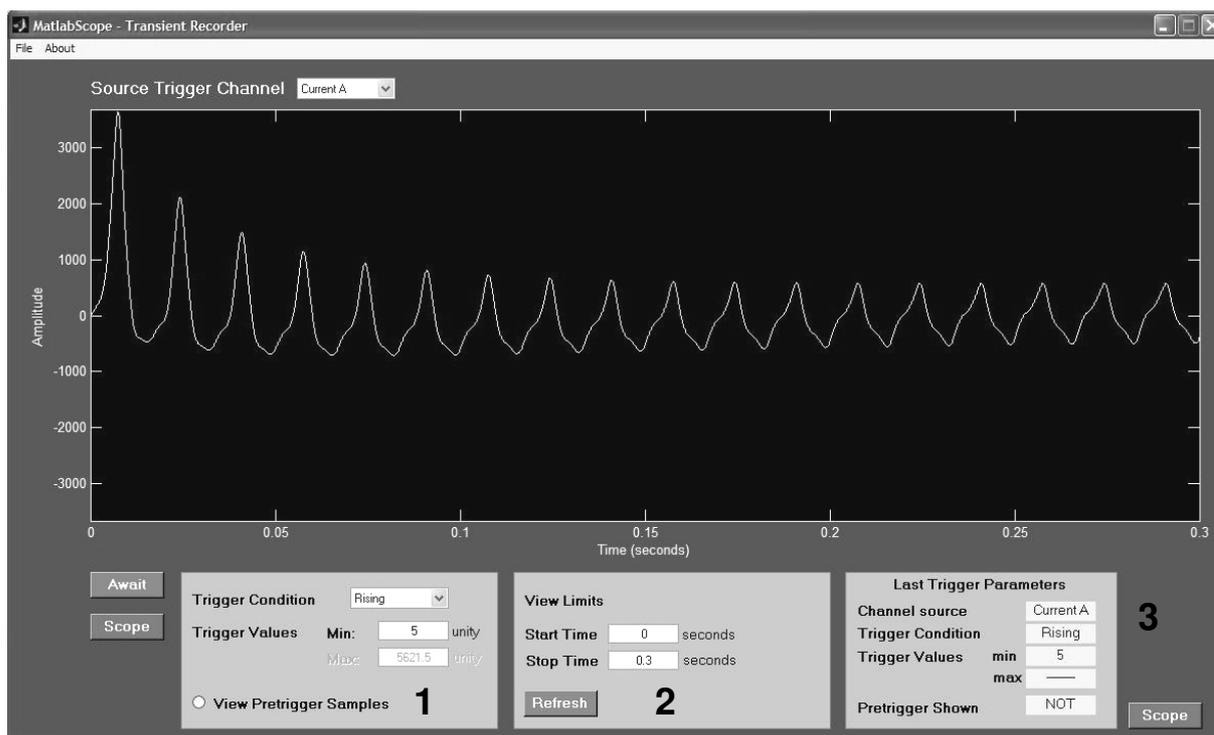
No canto inferior direito da interface do *Spectrum Analyzer* há o botão *Close*, que é usado para fechar a interface.

## 4.4 Gravador de Transientes no MatlabScope

A interface do gravador de transientes, aqui chamada de *Transient Recorder*, é responsável por gravar os sinais de tensão e corrente provenientes de transitórios programados pelo usuário, como por exemplo o fechamento ou abertura de disjuntores, curtos-circuitos programado, inserção e retirada de cargas do sistema, entre outros.

Os sinais de tensão e corrente, antes e durante o transitório, podem ser armazenados para uma posterior análise de seus comportamentos. Uma aplicação característica de tal funcionalidade é na obtenção de *correntes de inrush* em transformadores.

A Fig. 4.19 mostra a interface do *Transient Recorder*.



**Fig. 4.19**  
**Interface do Transient Recorder**

O *Transiente Recorder* funciona através do gatilhamento de um canal de sincronismo. O canal de sincronismo, ou canal fonte de gatilho, é colocado em modo de espera e ao ocorrer um transitório no sinal daquele canal o *Transient Recorder* inicia um processo de gravação do sinal.

O usuário pode definir como canal fonte de gatilho qualquer uma das correntes ou tensões do sistema. A escolha é feita pelo menu do tipo *popup*, situado

no lado superior direito da tela principal do *Transient Recorder*, e nomeado como *Source Trigger Channel*, mostrado na Fig. 4.19

O correto gatilhamento de um canal é feito a partir dos *Níveis de Gatilhamento* e da *Condição de Gatilhamento*. Os níveis e condições de gatilhamento são definidos pelo usuário na região marcada pelo número 1 na Fig. 4.19.

Nessa região, no menu do tipo popup nomeado como *Trigger Condition*, o usuário define a Condição de Gatilhamento. O *Transient Recorder* admite 4 condições de gatilhamento, definidas da seguinte forma:

*Rising*: O gatilhamento ocorre quando o sinal tiver derivada positiva quando passar pelo nível mínimo especificado;  
*Falling*: O gatilhamento ocorre quando o sinal tiver derivada negativa quando passar pelo nível mínimo especificado;  
*Entering*: O gatilhamento ocorre quando o nível do sinal estiver entrando no intervalo especificado pelos níveis mínimo e máximo;  
*Leaving*: O gatilhamento ocorre quando o nível do sinal estiver saindo do intervalo especificado pelos níveis mínimo e máximo;

Os níveis mínimos e máximos, requeridos para a completa especificação do gatilhamento, são fornecidos pelo usuário nos campos de texto editável nomeados como *Trigger Values Min* e *Max*, também situado na região 1 da Fig. 4.19.

Ainda nessa região, o *Radio Button*, nomeado como *View Pretrigger Samples* pode ser marcado caso o usuário deseje que o *Transient Recorder* também grave o sinal antes do gatilhamento.

Quando os Níveis e as Condições de Gatilhamento estiverem fornecidas o usuário deve clicar no botão *Await*, situado do lado direito da região 1, para fazer com que o *Transient Recorder* seja posto em modo de espera.

O *Transient Recorder* espera pelo gatilhamento por um tempo de 5 segundos. Caso não encontre nenhum gatilho nesse tempo, o usuário recebe uma mensagem de erro, indicando que não foi encontrado nenhum sinal de gatilhamento. Caso ocorra um gatilhamento dentro do tempo de 5 segundos, o *Transient Recorder* registra, durante o tempo de *um segundo*, o sinal da entrada especificada como fonte de gatilho e exibe o sinal na tela. Se o usuário não tiver optado pela gravação do sinal antes do gatilhamento, então a tela do *Transient Recorder* pode exibir o comportamento do sinal numa janela de tempo máxima de um segundo após a ocorrência do disparo. Se o usuário optou pela gravação do sinal antes do gatilhamento, então a janela de tempo continua sendo de um segundo, porém o sinal gravado compreende o intervalo de meio segundo antes do disparo até meio segundo após o disparo. Em qualquer caso, a janela de um segundo é suficiente para se analisar o comportamento do sinal transitório no sistema elétrico.

O eixo X da tela do *Transient Recorder* é o eixo do tempo. O usuário pode fornecer um intervalo de tempo específico para que o sinal gravado possa ser analisado. Os limites de observação do eixo do tempo são determinados na região marcada pelo número 2, e nomeada como *View Limits*. A caixa de texto editável nomeada como *Start Time* é usada para especificar o tempo inicial do intervalo, enquanto que a caixa de texto editável nomeada como *Stop Time* especifica o tempo final do intervalo. A qualquer momento, o usuário pode clicar no botão *Refresh* para que o sinal transitório seja redesenhado na tela do *Transient Recorder*.

A região marcada pelo número 3 é um quadro informativo que exibe os parâmetros do último gatilhamento realizado. Como podem ser vistos, tais parâmetros compreendem o Canal Fonte, a Condição de Gatilhamento, os Níveis

Máximos e Mínimos de Gatilhamento e a presença, ou não, do sinal antes do gatilhamento.

Os botões *Scope*, situados do lado direito e esquerdo da interface, são usados para que o usuário retorne à interface do Osciloscópio.

No canto superior da interface do *Transient Recorder* encontra-se a barra de menus. Como pode ser visto na Fig. 4.19, a barra de menus possui os seguintes itens:

*File*: menu com opções de salvamento de dados e imagens;

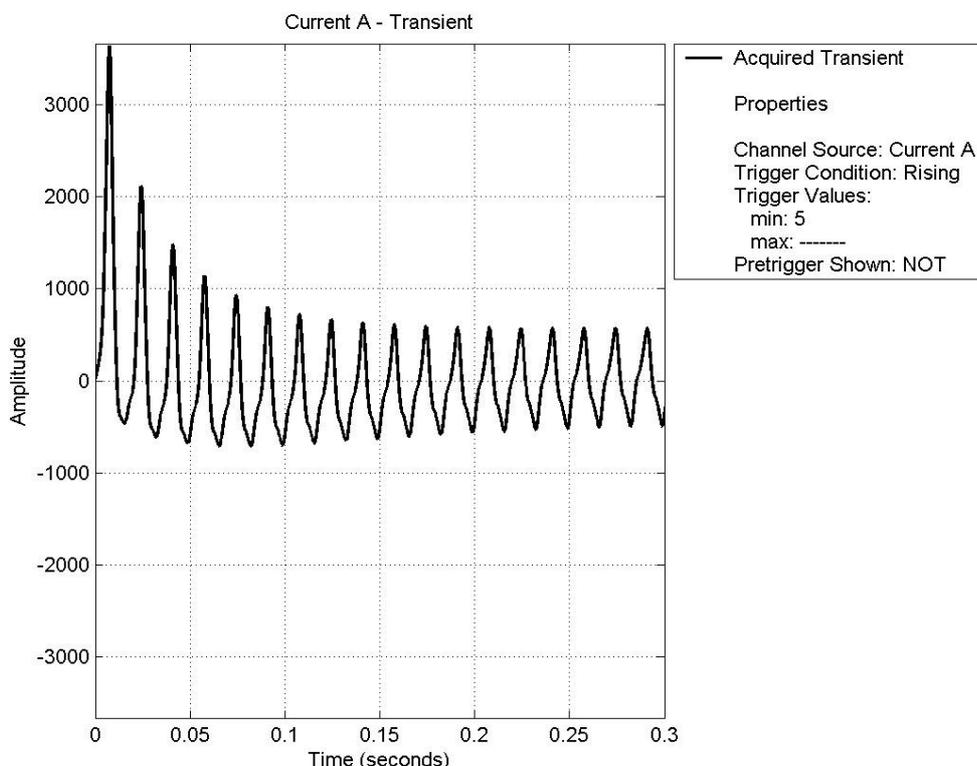
*About* : menu de informações sobre o desenvolvimento do MatlabScope.

O menu *About* é semelhante àqueles já apresentados nas interfaces citadas anteriormente.

O menu *File* possui quatro submenus, semelhantes àqueles apresentados também na interface do Osciloscópio e do Analisador de Espectro.

Ao clicar no submenu *Save Data as...* é aberta a caixa de diálogo de salvamento que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão *.DAT* que contém 32000 pontos no formato (x,y), onde x é o tempo e y é o valor da amplitude instantânea do sinal que pode posteriormente ser usado em outro programa de tratamento de dados.

Ao clicar no submenu *Save Screen as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão *.JPEG* que mostra o sinal presente na tela do *Transient Recorder*, assim como os parâmetros de gatilho, como mostrado na Fig. 4.20. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.



**Fig. 4.20**

**Arquivo de imagem obtido do submenu *Save Screen as...* do *Transient Recorder*.**

Ao clicar no submenu *Save Figure as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão

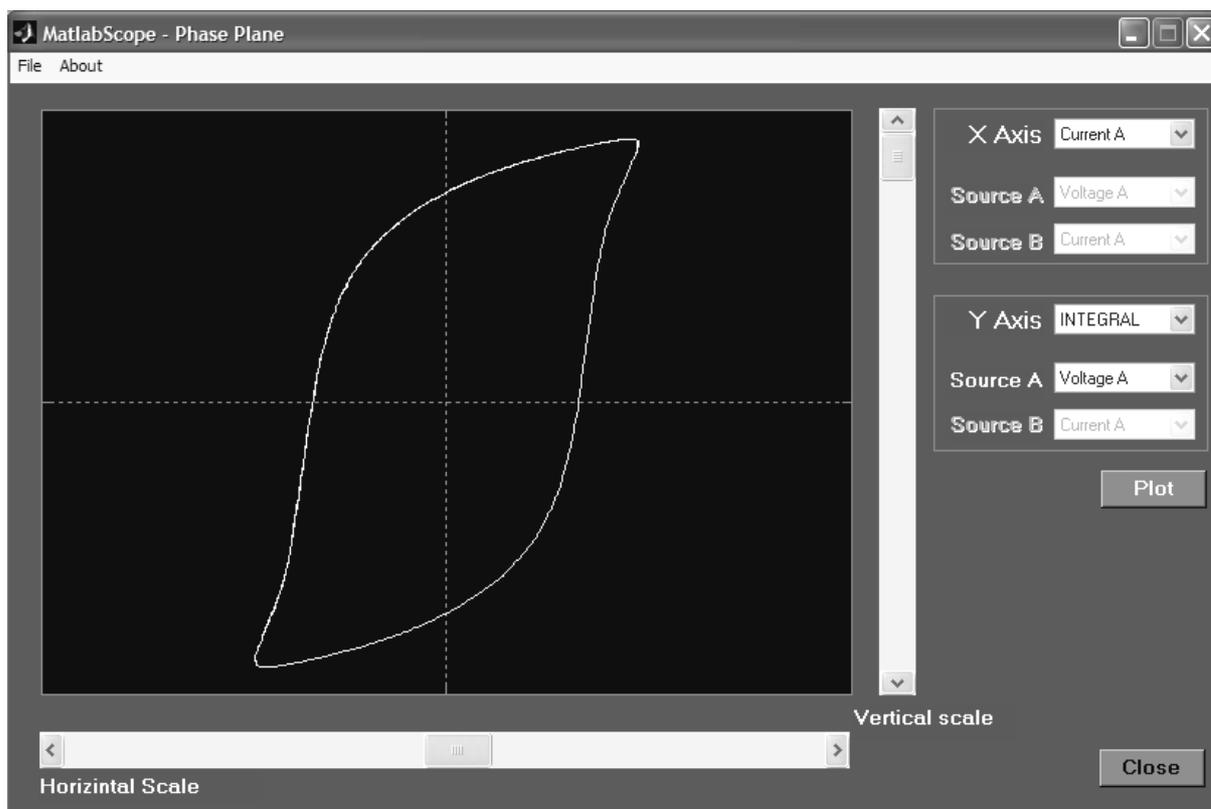
.JPEG que mostra toda a interface do *Transient Recorder*. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.

O submenu *Exit* é usado para sair e fechar a interface do *Transient Recorder*.

## 4.5 Plano de Fase no MatlabScope

Ao clicar no botão *Phase Plane*, ou através da barra de menus, ambos na interface do Osciloscópio, o usuário pode abrir a interface do Plano de Fase, aqui chamada de *Phase Plane*. Essa interface permite ao usuário traçar planos de fase a partir dos sinais adquiridos. Os planos de fase são úteis para analisar a relação de frequências e a fase relativa entre sinais periódicos.

A interface do *Phase Plane* é mostrada na Fig. 4.21.



**Fig. 4.21**  
**Interface do Phase Plane.**

Nessa interface é possível traçar planos cuja excursão nos eixos X e Y sejam provenientes de sinais de tensão, corrente, ou ainda de uma operação matemática. As fontes para os eixos X e Y são escolhidas a partir dos menus do tipo popup nomeados como *X Axis* e *Y Axis*, respectivamente, mostradas na Fig. 4.21 do lado direito da tela do *Phase Plane*. Quando a fonte do eixo é uma operação matemática, os menus popup nomeados como *Source A* e *Source B* são habilitados para que o usuário selecione as fontes (operandos) da operação matemática escolhida. Depois de selecionadas as fontes, o usuário deve clicar no botão *Plot* para que o plano de fase seja traçado.

Para uma melhor visualização do sinal na tela do *Phase Plane*, as escalas horizontal e vertical do gráfico traçado podem ser ajustadas através dos *sliders* *Horizontal Scale* e *Vertical Scale*, respectivamente. Esses *sliders* de controle estão mostrados na Fig. 4.21 e são situados abaixo e à direita da tela do *Phase Plane*.

No canto superior da interface do *Phase Plane* encontra-se a barra de menus. Como pode ser visto na Fig. 4.21, a barra de menus possui os seguintes itens:

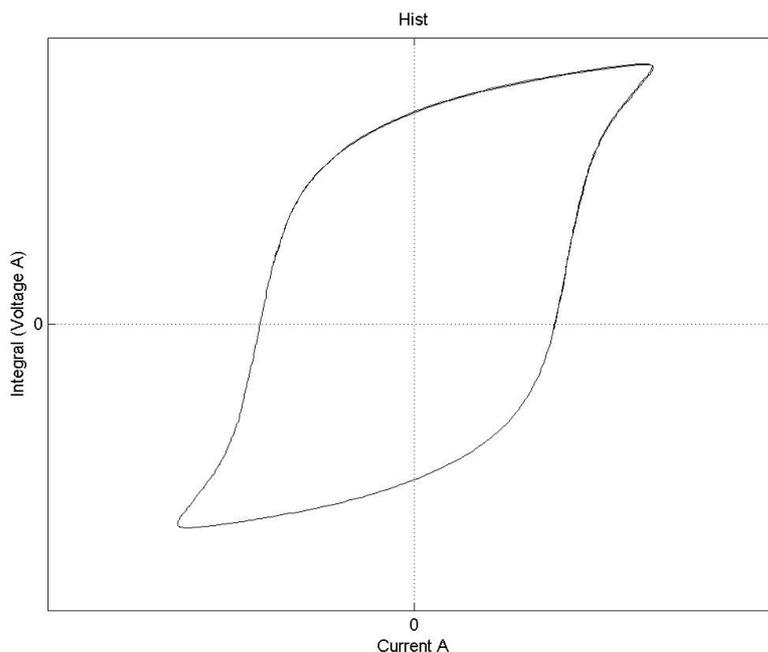
*File*: menu com opções de salvamento de dados e imagens;  
*About* : menu de informações sobre o desenvolvimento do MatlabScope.

O menu *About* é igual àqueles já apresentados nas interfaces já citadas.

O menu *File* possui quatro submenus, semelhantes àqueles apresentados também na interface do Osciloscópio.

Ao clicar no submenu *Save Data as...* é aberta a caixa de diálogo de salvamento que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão *.DAT* que contém pontos no formato (x,y), onde x é o sinal fonte do eixo x e y é o sinal fonte do eixo y. Esse arquivo pode posteriormente ser usado em outro programa de tratamento de dados.

Ao clicar no submenu *Save Screen as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão *.JPEG* que mostra o sinal presente na tela do *Phase Plane*, como mostrado na Fig. 4.22. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.



**Fig. 4.22**

**Arquivo de imagem obtido do submenu *Save Screen as...* do *Phase Plane*.**

Ao clicar no submenu *Save Figure as...* é aberta a caixa de diálogo que leva o usuário a escolher um nome e o diretório de destino para um arquivo com extensão *.JPEG* que mostra toda a interface do *Phase Plane*. Esse arquivo pode ser usado, por exemplo, para compor relatórios de experimentos.

O submenu *Exit* é usado para sair e fechar a interface do *Phase Plane*.

No canto inferior direito da interface do *Phase Plane* há o botão *Close*, que é usado para fechar a interface.

## 5. Teste do Sistema Implementado

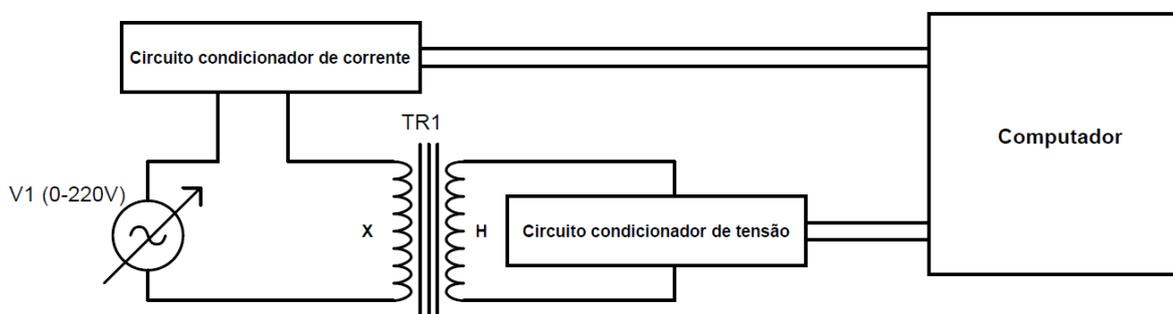
O sistema de aquisição foi testado a partir de uma série de medições realizadas sobre a montagem mostrada na Fig. 5.1.

Na figura, a fonte V1 é um autotransformador de saída variável (*varivolt*) com tensão de saída ajustável de 0 (zero) a 220V. O circuito condicionador de corrente é conectado em série com a saída do *varivolt* e tem a função de monitorar a corrente no circuito primário do transformador TR1. A saída do circuito condicionador de corrente é levada ao computador, sendo conectada a uma das entradas de corrente da placa de aquisição.

Observe, na figura, que o transformador TR1, com relação de transformação de 1:2, é alimentado pelos terminais X, ou seja, os terminais de baixa tensão. Isso é feito para que a tensão no secundário seja mais alta que no primário, a fim de analisar o comportamento do transformador em sobretensão, sendo necessário apenas aplicar uma baixa tensão nos terminais de entrada.

A presença do *varivolt* fornece um controle sobre o nível de excitação do transformador TR1, permitindo que o teste pudesse analisar o comportamento do transformador com tensão abaixo da nominal, tensão nominal e sobretensão. O transformador TR1 fornece uma isolação entre a rede de alimentação e o sistema de aquisição.

O circuito condicionador de tensão é ligado aos terminais H do transformador TR1, e tem a função de monitorar a tensão no seu secundário. A saída do circuito condicionador de tensão é levada ao computador, sendo conectada a uma das entradas de tensão da placa de aquisição.

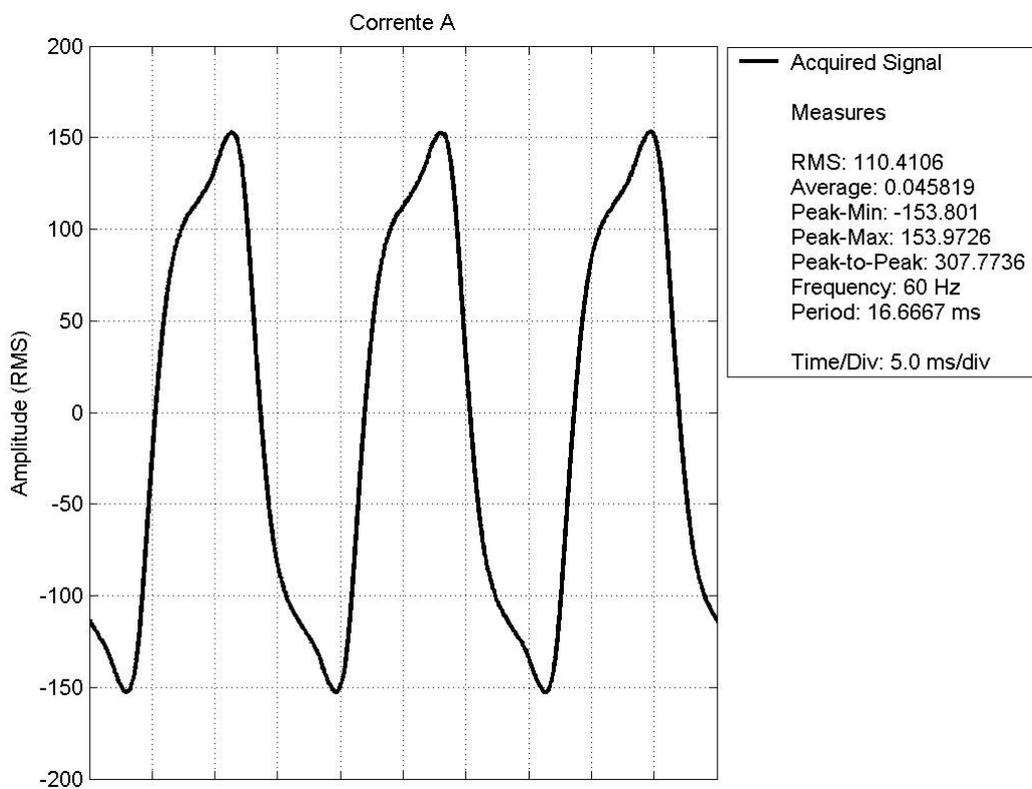
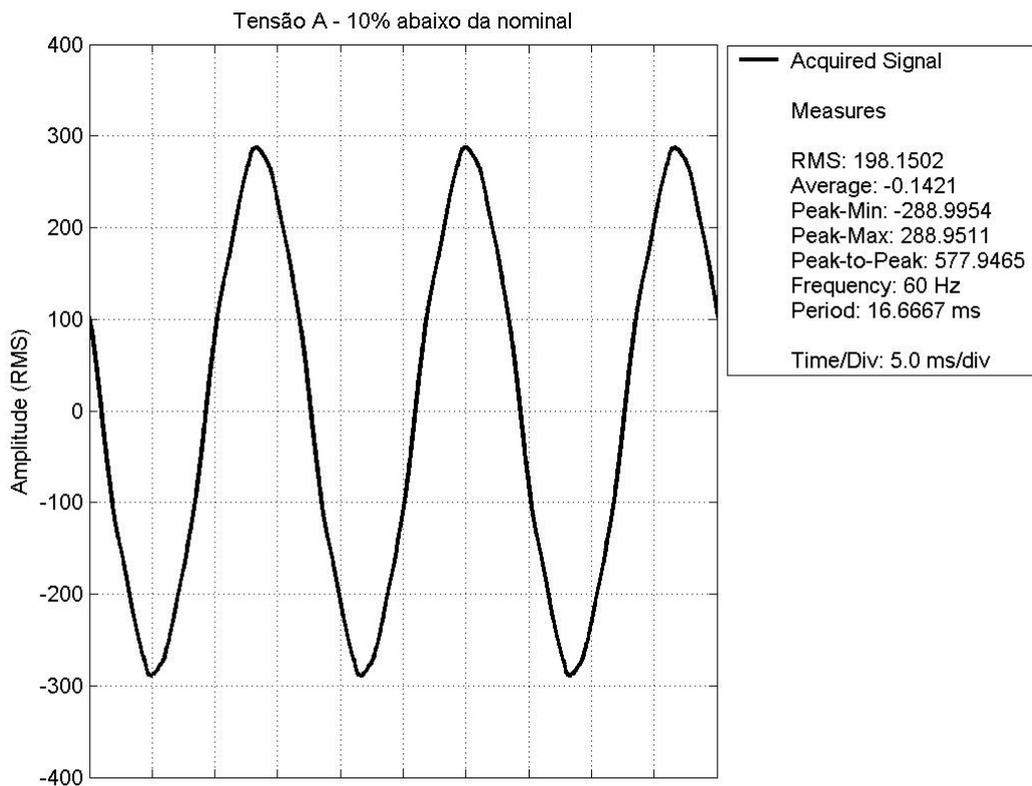


**Fig. 5.1**  
**Montagem realizada para teste do sistema de aquisição.**

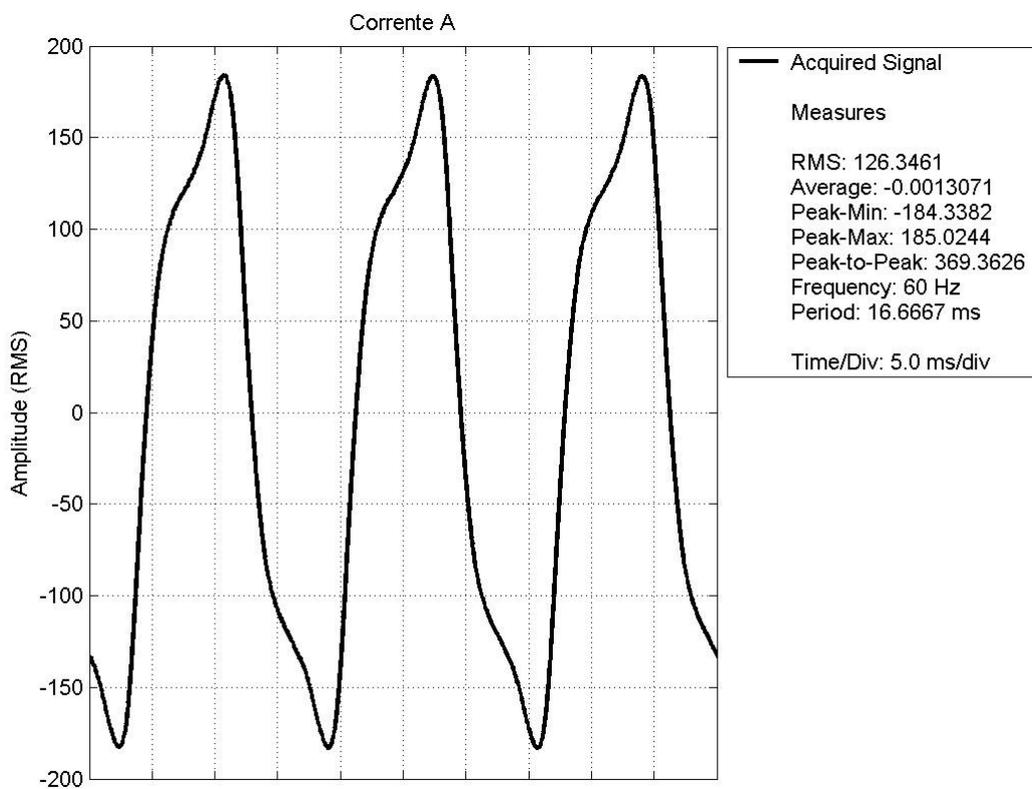
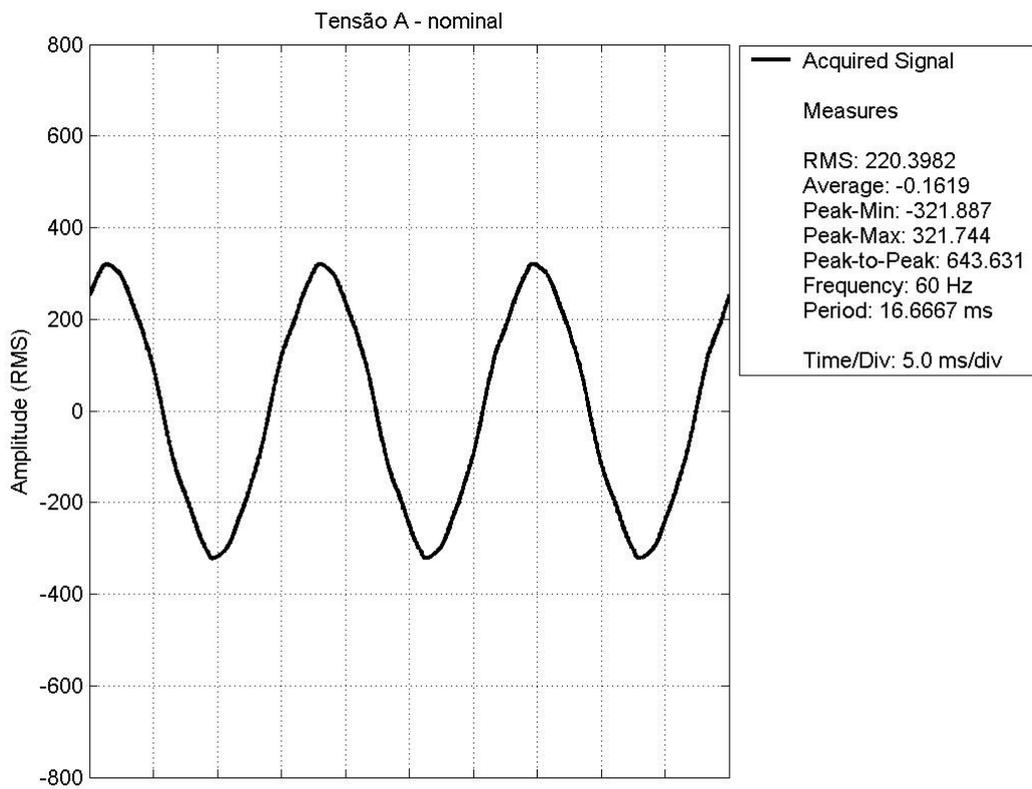
Os fatores de escala dos circuitos condicionadores foram devidamente levantados e inseridos no menu *Settings/Adjust Sensor Hardware Levels* antes das medidas serem efetuadas, a fim de fazer com que o *MatlabScope* exibisse os valores reais das grandezas medidas. Entretanto, devido à diferença da ordem de grandeza entre a tensão e a corrente monitoradas, e à disponibilidade de escalas de observação do eixo y na tela do Osciloscópio do *MatlabScope*, o fator de escala da corrente foi consideravelmente elevado para que o sinal pudesse ser melhor observado.

A Fig. 5.2 mostra os resultados das medidas de corrente no primário e tensão no secundário, feitas sobre a montagem acima na situação em que a alimentação equivale a 90% do valor de tensão nominal.

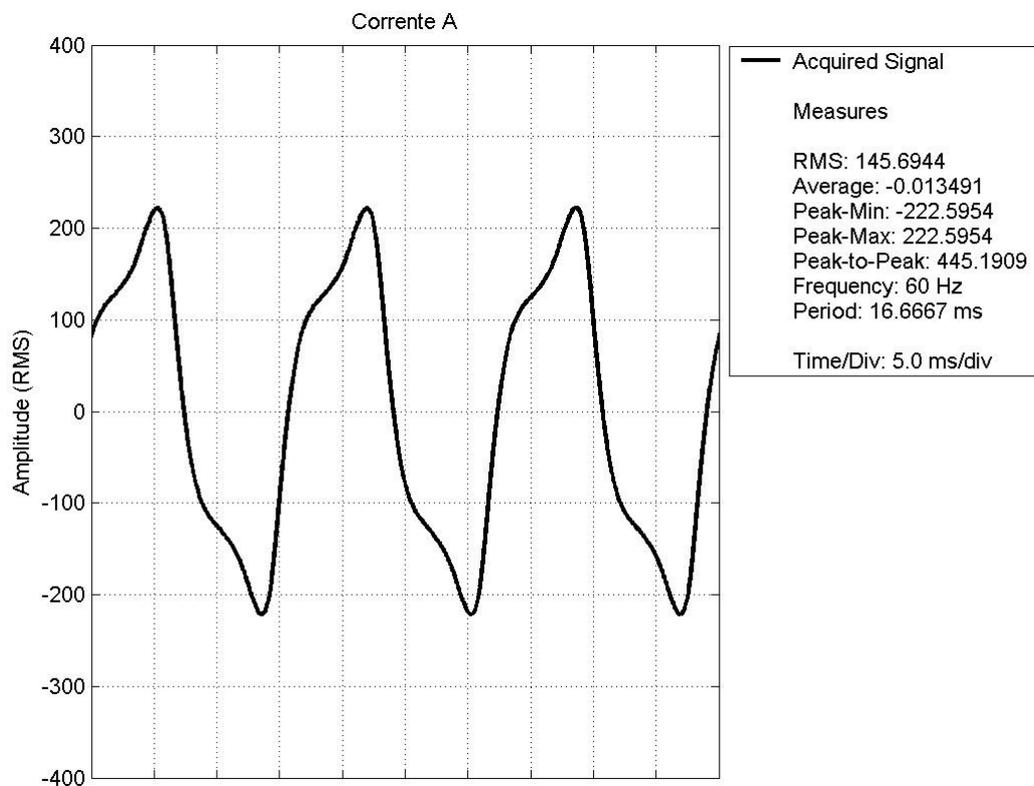
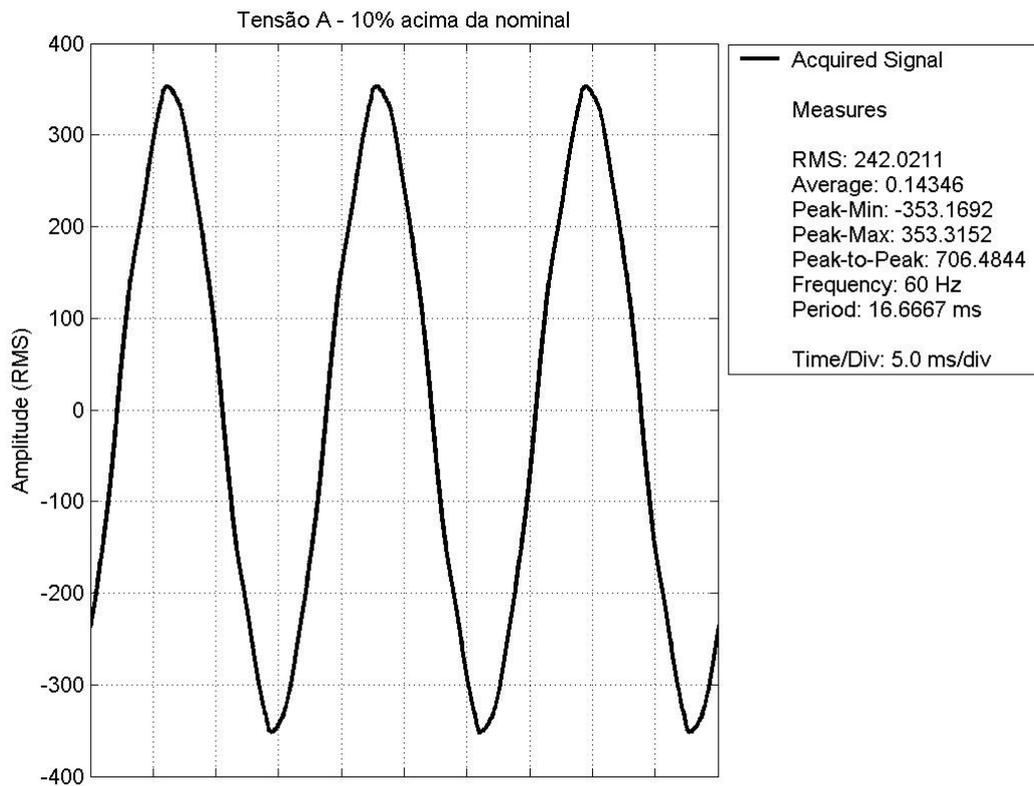
As figuras 5.3 e 5.4 são relativas aos resultados correspondentes à tensão nominal e 110% do valor de tensão nominal, respectivamente.



**Fig. 5.2**  
**Medidas de tensão no secundário e corrente no primário de TR1 na situação em que a alimentação equivale a 90% do valor de tensão nominal.**

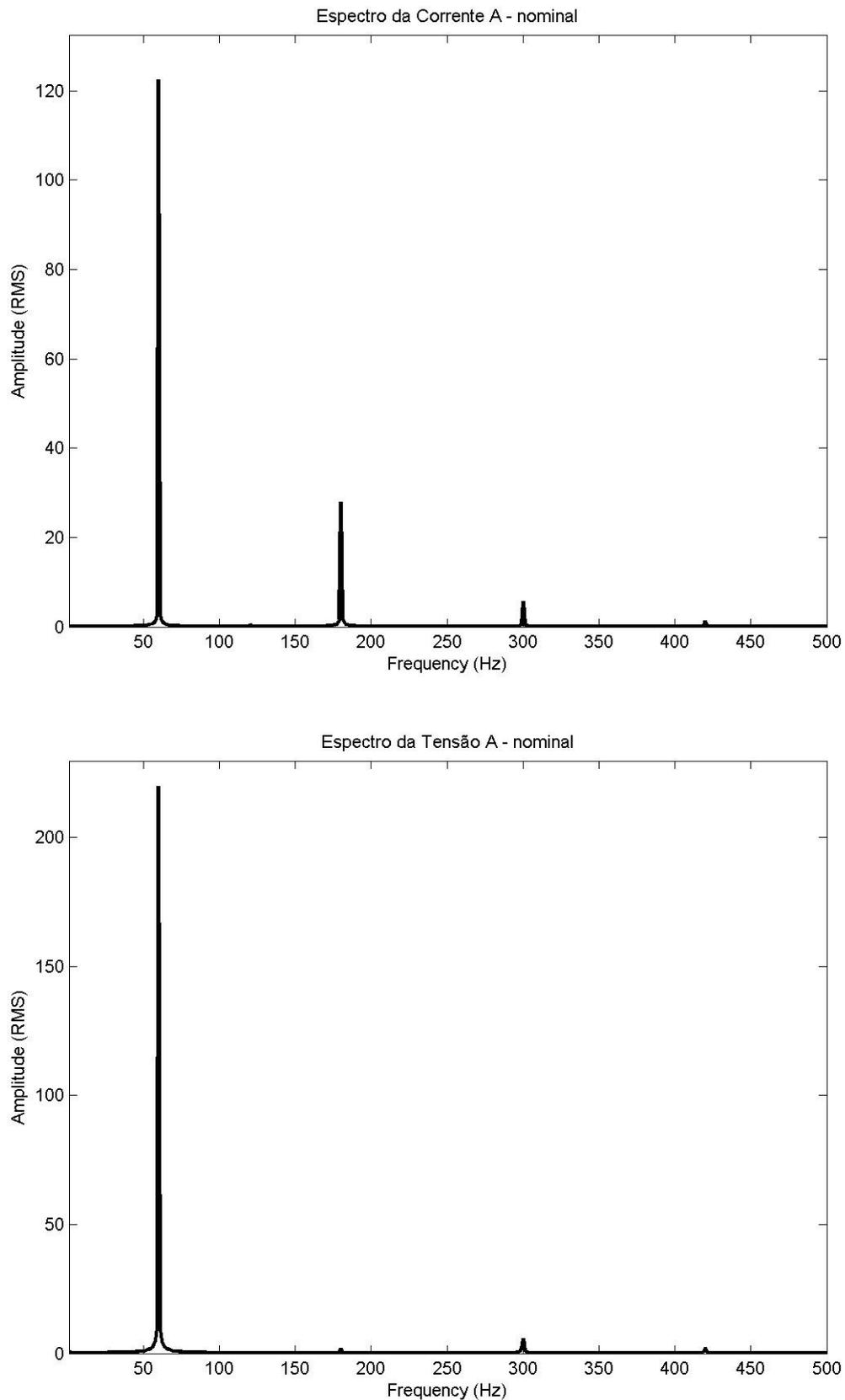


**Fig. 5.3**  
**Medidas de tensão no secundário e corrente no primário de TR1 na situação de tensão nominal.**



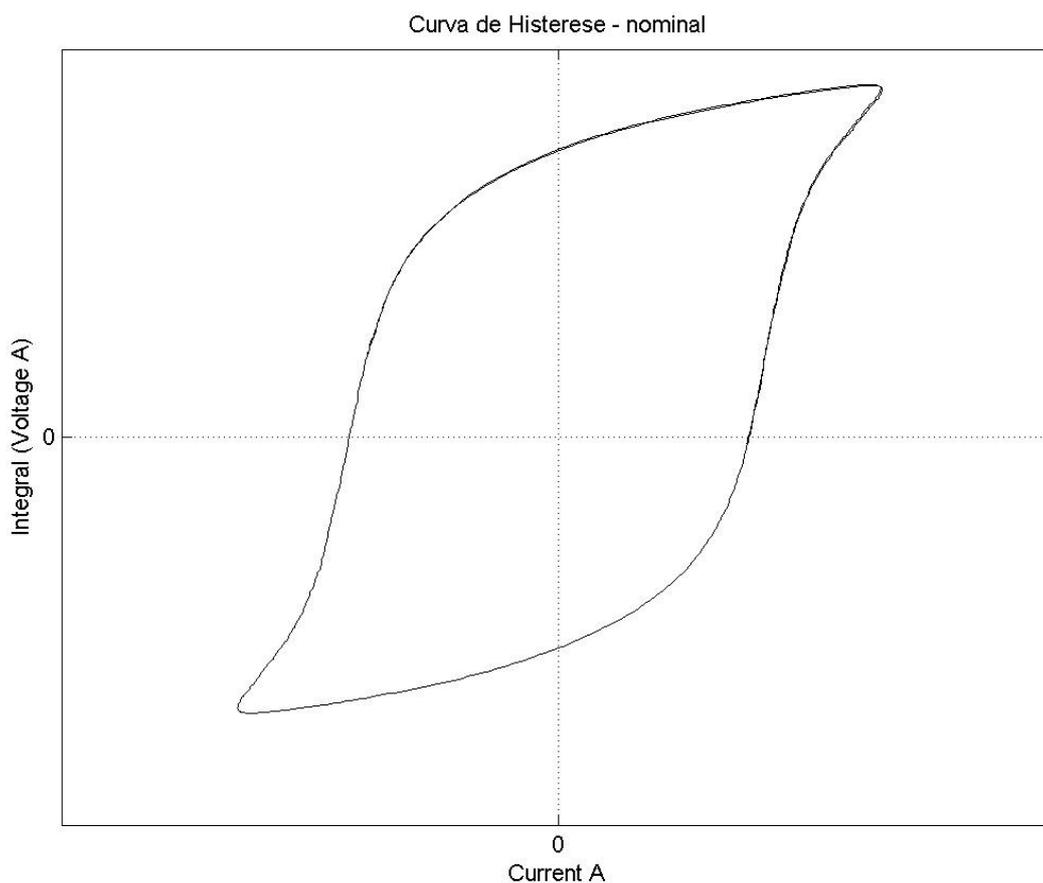
**Fig. 5.4**  
**Medidas de tensão no secundário e corrente no primário de TR1 na situação em que a alimentação equivale a 110% do valor de tensão nominal.**

A Fig. 5.5 mostra os espectros dos sinais de corrente e tensão, obtidos a partir do *Spectrum Analyzer*, para o caso de tensão nominal na alimentação.



**Fig. 5.5**  
**Espectro de frequências dos sinais de tensão e corrente na fase A para a condição de tensão nominal.**

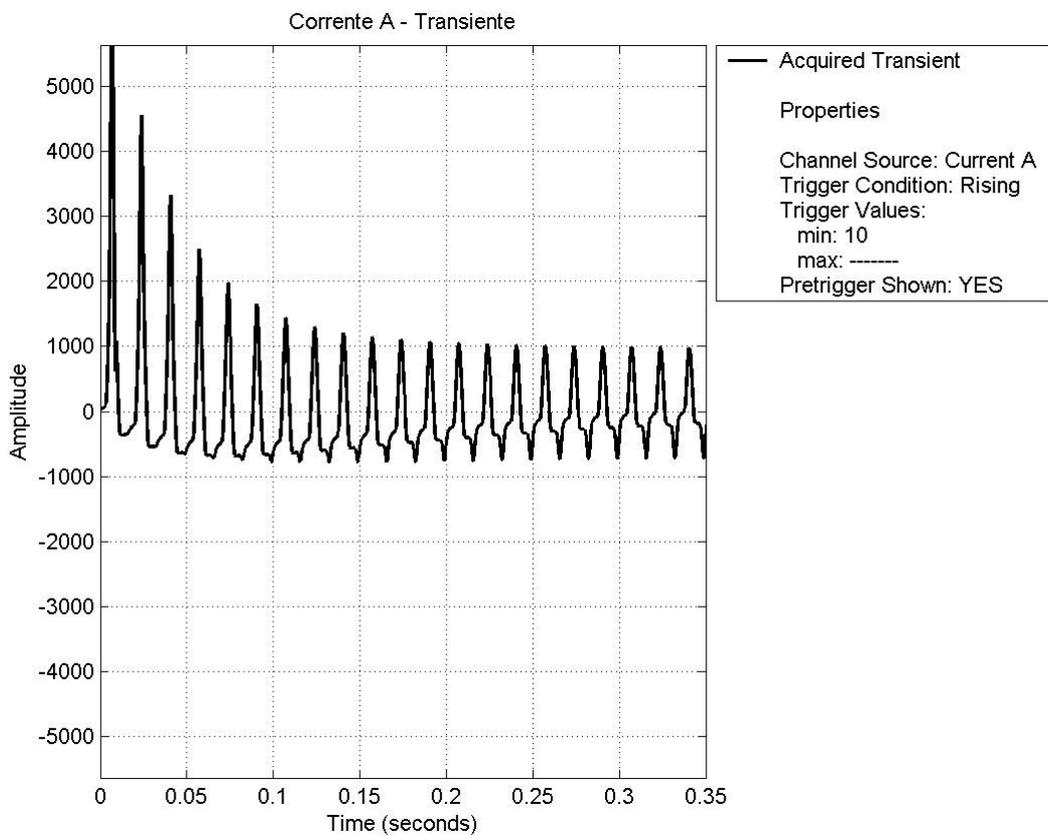
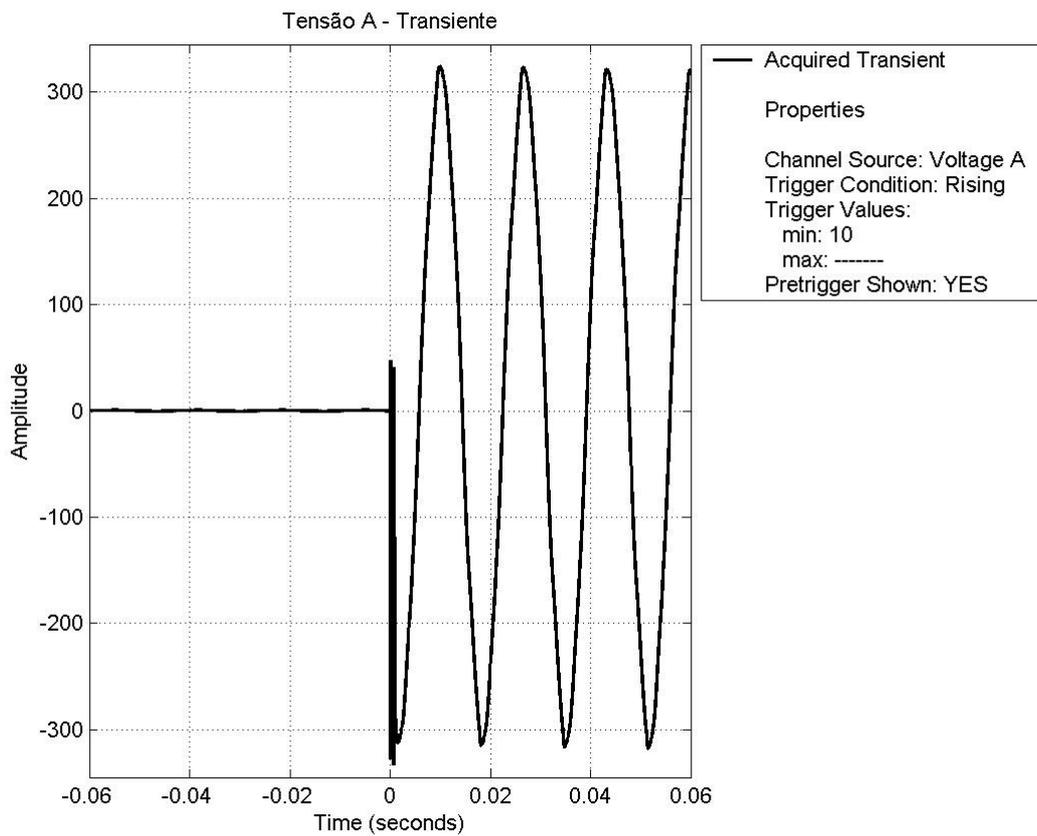
A Fig. 5.6 mostra um esboço do laço de histerese do transformador, obtido a partir do *Phase Plane*, para a condição de tensão nominal na alimentação.



**Fig. 5.6**  
**Curva de histerese do transformador TR1 para a condição de tensão nominal.**

No circuito do primário foi, em seguida, inserida uma chave para simular um transiente provocado pelo fechamento de um disjuntor de alimentação. Os sinais de tensão e corrente desse transiente foram capturados a partir do *Transient Recorder* e estão mostrados na Fig. 5.7.

Outras simulações foram feitas, a partir da síntese de sinais, utilizando geradores de sinais aleatórios de precisão para testar as demais funcionalidades do *MatlabScope*. Os resultados obtidos foram satisfatórios e consistentes com o propósito do sistema.

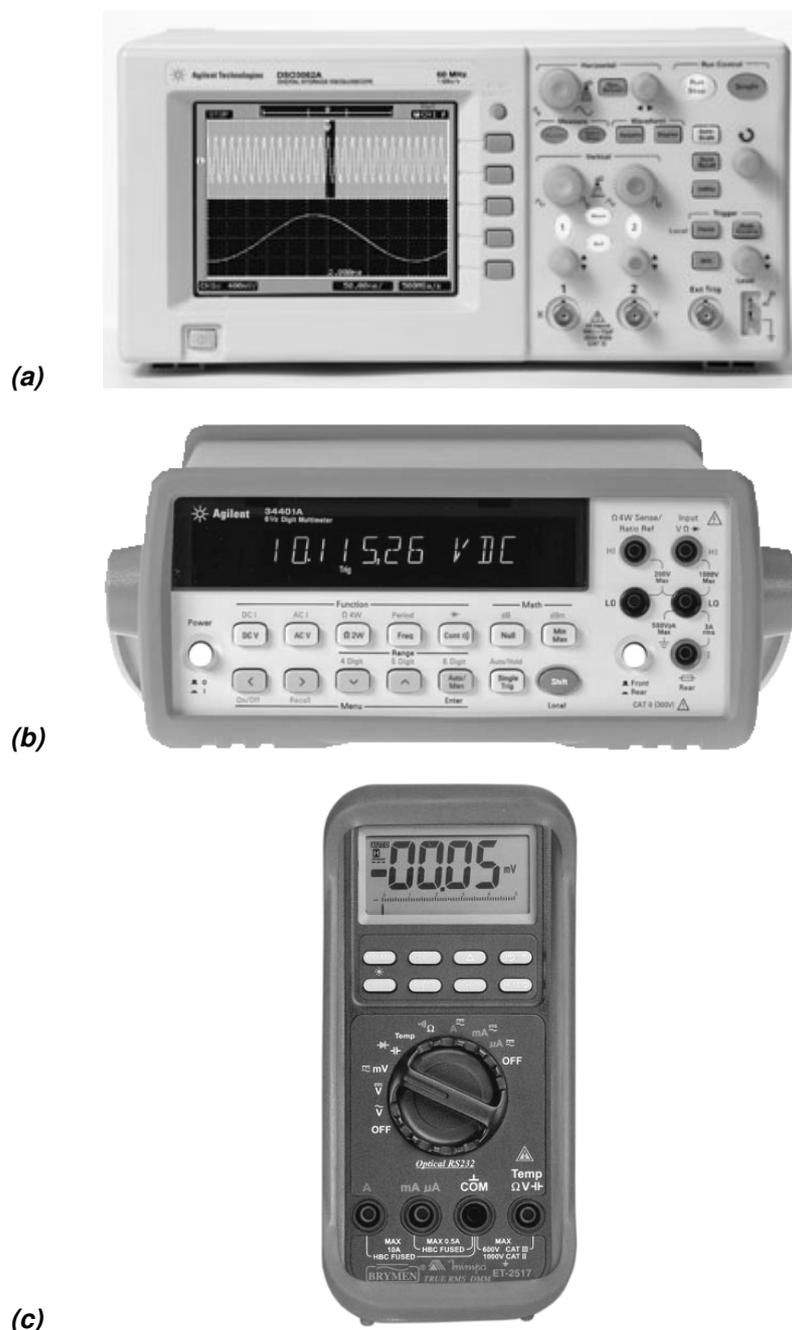


**Fig. 5.7**  
**Sinais de tensão e corrente capturados pelo Transient Recorder.**

## 6. Comparação com Outros Sistemas de Medição

Para se certificar da exatidão das medidas obtidas pelo *MatlabScope*, uma comparação com outros instrumentos de medição foi realizada. Nessa comparação, foram usados os instrumentos descritos no quadro abaixo, e mostrados na Fig. 6.1.

- \*1 Multímetro digital de mão *Minipa* modelo *ET-2517* (*True RMS*);
- \*1 Multímetro digital de bancada *Agilent* modelo *34401A* (*True RMS*);
- \*1 Osciloscópio digital *Agilent* modelo *DSO3062A*;



**Fig. 6.1**  
**Instrumentos de medição usados para comparação de medidas do MatlabScope.**  
**(a) osciloscópio digital Agilent DSO3062A; (b) multímetro de bancada Agilent 34401A;**  
**(c) multímetro de mão Minipa ET-2517.**

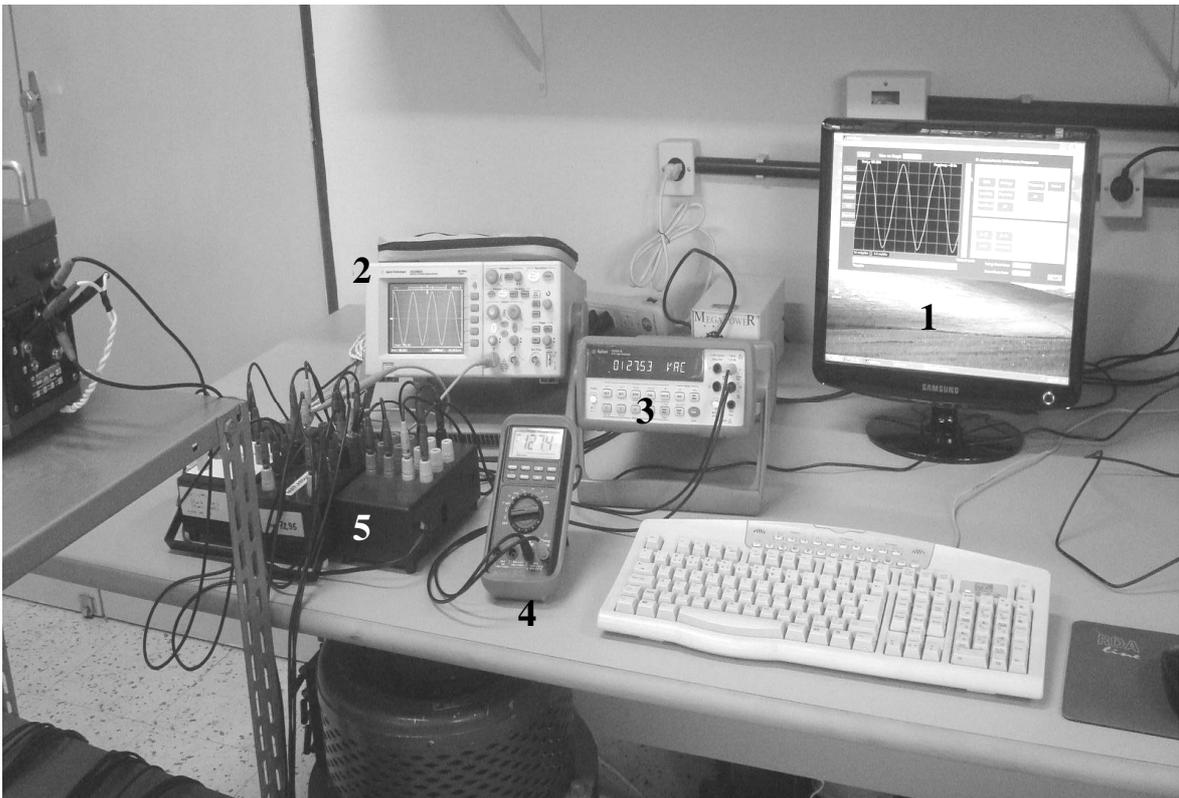
O valor *True RMS* dos sinais observados foi usado como referência de comparação, uma vez que essa grandeza era comum a todos os instrumentos. Para isso, a montagem mostrada na Fig. 5.1 foi utilizada. O transformador TR1 foi gradativamente excitado até alcançar 110% da tensão nominal. Nesse intervalo, foram obtidas algumas medidas do valor *RMS* da tensão no secundário utilizando os instrumentos acima citados, além do *MatlabScope*. As medidas e o desvio relativo ao *MatlabScope* estão mostradas na Tabela 6.1. O desvio é mostrado na tabela na coluna *Desvio Relativo Médio*, e é calculado como sendo a média dos desvios apresentados por cada instrumento em relação às medidas do *MatlabScope*.

**Tabela 6.1**  
**Comparação das medidas do *MatlabScope* com outros instrumentos de medição.**

Valores RMS no Secundário de TR1					
	MatlabScope	Osciloscópio Agilent DSO3062A	Multímetro Agilent 34401A	Multímetro Minipa ET-2517	Desvio Relativo Médio (%)
1	10,7431	10,5600	10,6700	10,8000	0,618
2	20,9671	20,9000	20,8000	20,8600	0,543
3	30,7595	30,9700	30,6330	30,7000	0,027
4	40,1787	40,1800	40,1400	40,2200	0,003
5	50,3822	50,3200	50,3180	50,3400	0,112
6	59,9130	60,0000	60,0000	60,2000	0,256
7	69,8600	69,8000	69,9400	70,1000	0,124
8	79,9565	81,2000	80,0000	80,0000	0,555
9	90,5498	90,7900	90,6810	90,9000	0,266
10	100,4515	100,7200	100,4160	100,6000	0,127
11	109,9636	110,2300	110,3500	110,1000	0,239
12	119,6757	119,7600	119,9600	120,0000	0,193
13	130,4016	130,8400	130,7800	131,2000	0,413
14	140,4770	141,1000	140,6200	140,8000	0,258
15	149,8040	150,5200	150,3400	150,6000	0,456
16	161,2369	161,7200	161,6800	161,5000	0,246
17	170,7214	170,7900	170,6800	171,1000	0,079
18	180,4290	180,8100	180,5300	181,1000	0,213
19	190,2901	191,3000	191,1800	191,5000	0,545
20	201,2199	201,9600	201,8300	202,1000	0,369
21	209,8536	210,5600	210,6000	210,8000	0,381
22	220,1269	220,6400	220,5200	220,6000	0,209
23	230,8486	231,3000	231,0000	231,8000	0,224
24	240,5876	240,8000	240,7600	240,9000	0,097

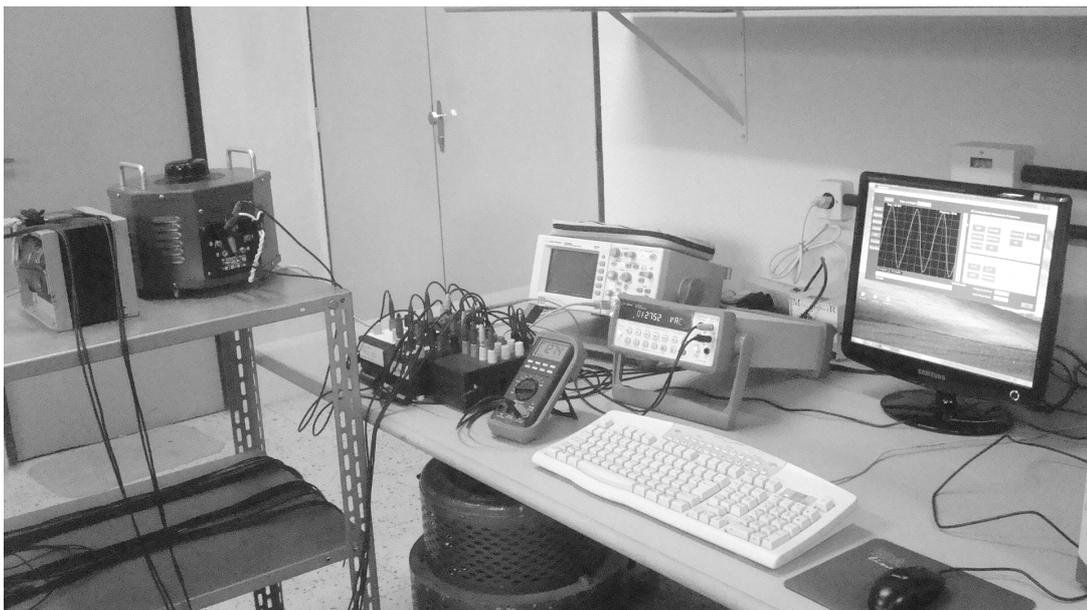
Uma característica importante que deve ser evidenciada na Tabela 6.1 é o fato do ótimo desvio relativo médio das medidas. Pode-se observar que as medidas efetuadas não apresentaram desvios relativos maiores que *um por cento*, o que confere uma boa precisão ao *MatlabScope* quando comparado aos outros instrumentos de medição.

A Fig. 6.2 mostra uma fotografia da bancada onde foram realizadas as montagens e os testes descritos.



**Fig. 6.2**  
**Bancada de montagem e testes.**

Na figura, são apresentados o monitor do computador, exibindo a interface do *MatlabScope* (1), o osciloscópio *Agilent DSO3062A* (2), o multímetro de bancada *Agilent 34401A* (3), o multímetro de mão *Minipa ET-2517* (4) e os módulos condicionadores de corrente e tensão (5). A Fig. 6.3 mostra a mesma bancada, onde ao seu lado esquerdo pode-se observar o *varivolt* e o transformador utilizados nos testes.



**Fig. 6.3**  
**Bancada de montagem e testes, com a presença do *varivolt* e do transformador utilizados.**

## 7. Conclusão

Engenheiros e pesquisadores possuem necessidades e requisitos que podem mudar rapidamente. Eles também precisam de soluções expansíveis e de fácil manutenção que possam ser utilizadas por um longo tempo. Criando-se instrumentos virtuais baseados em poderosos ambientes de desenvolvimento como o *Matlab*, projeta-se um *framework* aberto que integra *software* e *hardware*. Isto garante que a aplicação não só trabalhará bem hoje como também poderá facilmente integrar novas tecnologias, ou expandir soluções além do âmbito original, quando novos requisitos forem identificados.

Mesmo com a aplicação de um *hardware* não dedicado, o sistema desenvolvido apresentou considerável confiabilidade nas medidas realizadas, mostrando excelente grau de precisão, versatilidade e baixo custo de implementação para diversas aplicações, entre as quais podem ser citadas: estudos em materiais magnéticos, análise de fenômenos transitórios em sistemas elétricos, como correntes de *inrush*, análise harmônica, etc.

Apesar de o sistema ter sido implementado em uma arquitetura do tipo *Desktop (PC de bancada)*, a mobilidade e a portabilidade podem ser facilmente alcançadas, por exemplo, com o uso de *hardwares* com conexão *USB*, *Firewire*, ou mesmo *PCMCIA*, permitindo a sua implementação em computadores portáteis.

Uma grande vantagem do *MatlabScope* como instrumento virtual é a sua *modularidade*, a qual permite que o sistema seja ampliado pela adição de novos recursos de aquisição e análise. Isto é feito trabalhando-se apenas com novos elementos de *software*.

Finalmente, usando-se a ferramenta *Matlab Compiler*, que é parte integrante do ambiente *Matlab*, pode-se fazer com que o *MatlabScope* se torne uma aplicação executável, ficando independente da necessidade de se ter o *Matlab* instalado na máquina hospedeira.

Ao empregar soluções de instrumentação virtual, pode-se reduzir o investimento, custos de desenvolvimento do sistema e custos de manutenção, enquanto reduz-se o tempo de desenvolvimento.

## 8. Bibliografia

- B. R. Hunt, R. L. Lipsman and J. M. Rosenberg, "A guide to Matlab for Beginners and Experienced Users", 2<sup>nd</sup> Edition, Cambridge University Press, 2006.
- S. W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing", 2<sup>nd</sup> Edition, California Technical Publishing, 1999.
- H. Eren, "Electronic Portable Instruments Design and Applications", CRC Press, 2004.
- N. Kehtarnavaz and N. Kim, "Digital Signal Processing System-Level Design Using *LabVIEW*", Newnes, 2005.
- S. T. Karris, "Circuit Analysis II with Matlab Applications", Orchard Publications, 2004.
- "Understanding FFT Windows", LDS Dactron, Application Note 014, 2003.