

Este exemplar corresponde à redação final da tese defendida por **Valmir Fascio Juliano** e aprovado pela comissão julgadora em **09/02/96**.


Profª Drª Concetta Kascheres



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE QUÍMICA

Campinas - SP

1996

TESE DE DOUTORADO

**CONTROLE, AQUISIÇÃO E TRATAMENTO DE DADOS
PARA ESPECTROMETRIA DE MASSAS SEQUENCIAL EM
APARELHO PENTAQUADRUPOlar: ESPECTROS
MULTIDIMENSIONAIS MS^n ($n=1,2,3$) E MONITORAMENTO
DE ÍONS EM FUNÇÃO DO TEMPO E POTENCIAL DE
IONIZAÇÃO.**

Valmir Fascio Juliano

Orientadora: Profª Drª Concetta Kascheres

Co-orientador: Prof. Dr. Claudimir Lucio do Lago.



**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO INSTITUTO DE QUÍMICA
UNICAMP**

Juliano, Valmir Fascio

J941c Controle, aquisição e tratamento de dados para espectrometria de massas sequencial em aparelho pentaquadrupolar : espectros multidimensionais MS^n ($n=1,2,3$) e monitoramento de íons em função do tempo e potencial de ionização. / Valmir Fascio Juliano. -- Campinas, [SP : s.n.], 1996.

Orientadora: Concetta Kascheres.

Co-orientador: Claudimir Lucio do Lago.

Tese (doutorado) - Universidade Estadual de Campinas.
Instituto de Química.

1. Espectrometria de massas sequencial. 2. Espectros de massas multidimensionais. 3. Espectrômetro de massas pentaquadrupolar. I. Kascheres, Concetta. II. Lago, Claudimir Lucio do. III Universidade Estadual de Campinas. Instituto de Química. IV. Título.

UNIDADE	BC
N.º CHAMADA	T/UNICAMP
	3942c
V.	Ex.
T.MBO BC	27186
PROC.	667/96
C	☐
D	☒
PREÇO	R\$ 11,00
DATA	30/03/96
N.º CPD	

CM-00 099640-6

COMISSÃO JULGADORA

Titulares:

Ivano Gebhardt Rolf Gutz	Instituto de Química - USP
Paulo Celso Isolani	Instituto de Química - USP
Celio Pasquini	Instituto de Química - DQA - UNICAMP
Rogério Custódio	Instituto de Química - DFQ - UNICAMP

Suplentes:

Antonio L. Pires Valente	Instituto de Química - DQA - UNICAMP
Ronei de Jesus Poppi	Instituto de Química - DQA - UNICAMP

CURRICULUM VITAE

I - FORMAÇÃO ACADÊMICA

GRADUAÇÃO : Bacharel em Química com Atribuições Tecnológicas pelo Instituto de Química da Universidade Estadual de Campinas, concluído em Dezembro de 1987.

PÓS-GRADUAÇÃO:

Mestrado - Universidade Estadual de Campinas - Concluído em 08/07/91.

TÍTULO: *"Construção de Equipamento para Eletroquímica e Caracterização de Filmes de Polipirrol"*.

RESUMO: Foram desenvolvidos uma interface, que permitisse o acoplamento de um microcomputador compatível ao IBM PC/XT a um potenciostato/galvanostato, e um conjunto de programas a fim de possibilitar a aquisição e tratamento de dados para experimentos envolvendo a síntese e caracterização eletroquímica de polímeros condutores. Para avaliação do equipamento desenvolvido foram feitos ensaios com polipirrol em condições bem estabelecidas. Os ensaios demonstraram que o sistema (*Hardware + Software*) é muito útil e versátil.

II - CURSOS A NÍVEL DE PÓS-GRADUAÇÃO

1- "Introdução à Ciência dos Polímeros"

Prof. Dr. Fernando Gallembek

2- "Eletroquímica Aplicada a Polímeros Condutores"

Prof. Dr. Jean M. Pernaut

3- "Estabilização e Degradação de Polímeros"

Prof. Dr. Marco-Aurélio De Paoli

4- "Introdução à Química do Estado Sólido"

Prof. Dr. Oswaldo L. Alves

5- "Polímeros Condutores"

Prof. Dr. Marco-Aurelio De Paoli

6- "Técnicas de Análise Térmica Aplicados à Polímeros"

Prof^a Dr^a Maria Izabel Felisberti

7- "Métodos Eletroquímicos e Espectroquímicos"

Prof. Dr. Marco-Aurélio De Paoli

8- "Introdução a Espectrometria de Massas"

Prof. Dr. Marcos N. Eberlin

9- "Planejamento Experimental e Tratamento de Dados Químicos"

Prof. Dr. Roy E. Bruns

III - PARTICIPAÇÕES EM CONGRESSOS TÉCNICOS/CIENTÍFICOS

- 1- VI Seminário de Polímeros - VI SEMPOL BRASIL/FRANÇA, IMA - UFRJ, Rio de Janeiro, de 12 a 16/09/88.
- 2- I Congresso Internacional de Tintas, realizado pela ABRAFATI, São Paulo-SP., Maio de 1989.
- 3- 4^º Encontro sobre Materiais na Indústria Eletrônica e de Telecomunicações (EMIET), CPqD TELEBRAS, de 12 a 14/08/91.
- 4- VIII SIBEE - Simpósio Brasileiro de Eletroquímica e Eletroanalítica, Centro de Convenções-UNICAMP, 13 a 15/04/92.
- 5- 16^a Reunião Anual da Sociedade Brasileira de Química, Caxambu, MG, 25 a 29/05/93.
- 6- 17^a Reunião Anual da Sociedade Brasileira de Química, Caxambu, MG, 24 a 27/05/94.
- 7- 18^a Reunião Anual da Sociedade Brasileira de Química, Caxambu, MG, 31/05 a 02/06/95.
- 8- VIII Encontro Nacional de Química Analítica, Belo Horizonte, MG, 03 a 06/09/95.

IV - PUBLICAÇÕES

- 1- V.F. Juliano, A. Justino Filho e M.-A. De Paoli, "Monitoramento da degradação térmica de PVC por condutivimetria", *Quim. Nova*, **10**, 273 (1987).
- 2- J.M. Pernaut, R.C.D. Peres, V.F. Juliano e M.-A. De Paoli, "Electrochemical study of polypyrrole/dodecylsulfate", *J. Electroanal. Chem.*, **274**, 225-233 (1989).
- 3- E. Rezende Duek, V.F. Juliano, M. Guzzo, C. Kascheres e M.-A. De Paoli, "The photo-oxidation of EPDM rubber: Part II - The photo-initiation process", *Polym. Deg. & Stab.*, **28**, 235-248 (1990).
- 4- R.C.D. Peres, V.F. Juliano, M.-A. De Paoli, S. Panero e B. Scrosati, "Electrochromic properties of dodecylbenzenesulfonate doped poly(pyrrole)", *Electrochim. Acta*, **38**(7), 869-876 (1993).
- 5- Wilson A. Gazotti Jr., Valmir F. Juliano e Marco-A. De Paoli, "Thermal and photochemical degradation of dodecylsulfate doped poly(pyrrole)", *Polym. Deg. & Stab.*, **42**, 317-321 (1993).
- 6- Valmir F. Juliano, Marcos N. Eberlin, Concetta Kascheres e C. L. do Lago, "Improvements on a Magnetic Field Probe for a Double Focusing Mass Spectrometer using a Unijunction Transistor", *Lab. Microcomp.*, **13** (2-4), 83-87 (1994).
- 7- C.L. do Lago, V.F. Juliano e C. Kascheres, "Applying Moving Median Digital Filter to Mass Spectrometry and Potentiometric Titration", *Anal. Chim. Acta*, **310**, 281-288 (1995).
- 8- V. F. Juliano, F. C. Gozzo, M. N. Eberlin, C. Kascheres e C. L. do Lago, "Fast Multidimensional (3D and 4D) MS² and MS³ Scans in a High-Transmission Pentaquadrupole Mass Spectrometer", *Anal. Chem.*, **00**,00 (1996).

AGRADECIMENTOS

- À CAPES pela bolsa de doutorado;
- À FAPESP pela bolsa de doutorado adicional (processo nº 95/3196-3);
- À minha orientadora Professora Dr^a Concetta Kascheres (Connie) por ter dado a oportunidade de trabalhar neste projeto e por ter confiado em minha capacidade;
- Ao meu co-orientador Professor Dr. Claudimir Lucio do Lago e amigo de mais de dez anos pela oportunidade, incentivo e liberdade para desenvolver este trabalho;
- Ao Professor Dr. Marcos N. Eberlin por ter colaborado com idéias e sugestões que muito auxiliaram no desenvolvimento do trabalho;
- Ao Aparecido (Cidão) e à Zelinda pela agradável convivência nestes quatro anos de trabalho no Laboratório de Espectrometria de Massas;
- Aos alunos da Prof^a Connie e do Prof. Marcos que, pacientemente, testaram os programas desenvolvidos, dando sugestões para melhorá-los;
- Ao Fábio C. Gozzo pela parceria na instalação do ponto de rede internet dentro do Laboratório de Espectrometria de Massas;
- Ao Mário C. Guerreiro pelas inúmeras discussões na área de informática;
- À minha esposa Marcia pelo incentivo, colaboração, inspiração e por ter me dado uma filha maravilhosa;
- Aos meus pais pelo incentivo que sempre me deram;
- À todos que, direta ou indiretamente, colaboraram para o desenvolvimento deste trabalho;

Dedico este trabalho à duas mulheres maravilhosas a quem tenho dedicado minha vida:

- minha filha Michele que veio ao mundo para trazer muita alegria a todos que a cercam;*
- minha esposa Marcia que tem se dedicado a mim e a Michele.*

ÍNDICE

☪ Símbolos e Abreviações	1
📄 Resumo	2
📄 Abstract	3
📁 Capítulo I: Objetivos	I.1
📖 Capítulo II: Introdução	II.1
II.1. Ionização	II.1
II.2. Análise de m/z	II.2
II.3. Detecção	II.5
II.4. Espectrometria de Massas Sequencial	II.5
II.5. O Instrumento Pentaquadrupolar	II.9
II.6. Bibliografia	II.15
🖨 Capítulo III: Hardware	
III.1. Implementação	III.1
III.1.A. Endereçamento	III.6
III.1.B. Tempo de Acomodação e Transmissão nos Quadrupolos Analisadores de Massa	III.7
III.1.C. Ajuste do Potencial de Ionização	III.10
III.2. Discussão	III.11
III.3. Conclusão	III.12
III.4. Bibliografia	III.12
III.A. Apêndice: Trocador do Receptáculo de Íons	III.13
🔧 Capítulo IV: Circuito Oscilador “Aprisionador” de Íons	
IV.1. Introdução	IV.1
IV.1.A. Teoria Cinética dos Gases e as Colisões	IV.1
IV.1.B. O Movimento dos Íons	IV.3
IV.2. Implementação	IV.4
IV.3 Resultados	IV.7
IV.4. Discussão	IV.13
IV.5. Conclusão	IV.15
IV.6. Bibliografia	IV.16

IV.A. Apêndice:

IV.A.1 Experimento para Demonstrar a Ocorrência de Múltiplas Colisões IV.17

IV.A.2 Simulações do Comportamento de um Íon Dentro de q2 sem o Efeito IV.18

das Colisões Múltiplas

 Capítulo V: Software

V.1. Introdução à Programação Orientada a Objetos (POO) e Motivada por V.I

Eventos (PME)

V.2. Implementação V.6

V.3. Discussão V.16

V.4. Conclusão V.17

V.5. Bibliografia V.18

 Capítulo VI: Filtragem Digital de Sinais por Mediana Móvel

VI.1. Introdução VI.1

VI.2. Teoria e Metodologia VI.2

VI.3. Aplicações VI.7

VI.4. Discussão VI.7

VI.5. Conclusão VI.9

VI.6. Bibliografia VI.10

VI.A. Apêndice

VI.A.1. Predição do Efeito de Corte (*cut-off*) VI.12

VI.A.2. Listagem da Rotina de Filtragem Digital por Mediana Móvel com VI.14

Janela de Filtragem $w = 3$

 Capítulo VII: Aplicações

VII.1. Programa PENTA2D VII.1

VII.2. Programa PENTA3D VII.5

VII.3. Programa PENTA4D VII.7

VII.4. Programa SIMTIM VII.12

VII.5. Bibliografia VII.14

 Capítulo VIII: Avaliação Global e Conclusões VIII.1

☸ SÍMBOLOS E ABREVIACÕES

• ADC	Conversor analógico/digital;
• A/D	Analógico/digital;
• CI	Ionização química;
• CID	Dissociação induzida por colisão;
• DAC	Conversor digital/analógico;
• D/A	Digital/analógico;
• ddp	Diferença de potencial elétrico;
• DC	Corrente contínua;
• EI	Ionização por impacto de elétrons;
• GC/MS	Cromatografia gasosa/espectrometria de massas;
• H, h	Indica a base hexadecimal;
• I/O	Entrada e saída de dados;
• IEM	Monitoramento de energia de ionização;
• ME	Multiplicador de elétrons;
• MIMS	Espectrometria de massas por introdução via membrana;
• MS ⁿ	Espectrometria de massas sequencial ou de estágios múltiplos (n=1,2,3,...);
• MS/MS	Espectrometria de massas/espectrometria de massas ou MS ² ;
• m/z	Massa/carga;
• PME	Programação motivada por eventos;
• POO	Programação orientada a objetos;
• RF, rf	Rádio-frequência;
• s	Estimativa do desvio padrão σ ;
• SIM	Monitoramento de íons selecionados;
• σ	Desvio padrão;
• TIC	Corrente iônica total;
• TIM	Monitoramento de íons totais;
• u	Unidade de massa atômica
• 2D	Bidimensional;
• 3D	Tridimensional;
• 4D	Tetradimensional;

CONTROLE, AQUISIÇÃO E TRATAMENTO DE DADOS PARA ESPECTROMETRIA DE MASSAS SEQUENCIAL EM APARELHO PENTAQUADROPOLAR: ESPECTROS MULTIDIMENSIONAIS MS^n ($n=1,2,3$) E MONITORAMENTO DE ÍONS EM FUNÇÃO DO TEMPO E POTENCIAL DE IONIZAÇÃO.

Valmir Fascio Juliano, Claudimir L. do Lago e Concetta Kascheres

RESUMO

Um espectrômetro de massas pentaquadrupolar foi interfaceado a um microcomputador 486DX 50 MHz através de um cartão de interface AT-MIO-16X da National Instruments e de um outro desenvolvido em nosso laboratório.

Foi desenvolvido um conjunto de programas para controlar o instrumento e adquirir dados de experimentos MS^1 , MS^2 e MS^3 multidimensionais em espectrometria de massas sequencial. Também foi desenvolvido um programa para o monitoramento de íons em função do tempo e do potencial de ionização. O monitoramento em função do tempo permite trabalhar com a técnica de espectrometria de massas por introdução via membrana (MIMS), bem como com outros processos que gerem produtos gasosos. O potencial de ionização é controlado através de um conversor D/A, permitindo adquirir espectros enquanto é feita sua varredura. Esta técnica está sendo avaliada para emprego como ferramenta auxiliar na interpretação de espectros, assim como na eliminação de contaminações. Todo o software foi desenvolvido em Turbo Pascal 1.5 para Windows utilizando a técnica de programação orientada a objetos e programação motivada por eventos a fim de permitir um completo controle e expansão do sistema.

Com o objetivo de eliminar e impedir a interferência de sinais espúrios, foi feito um estudo do efeito de um filtro digital robusto por mediana móvel sobre um sinal analítico, comparando-o com o método de filtragem por média móvel, o qual tem sido bastante utilizado nestes processos de filtragem. Posteriormente, implementou-se o método nos programas de aquisição de dados do equipamento, obtendo-se resultados bastante satisfatórios.

Um circuito oscilador foi acoplado ao controlador do potencial elétrico da lente de focalização de saída do primeiro quadrupolo de reação a fim de aumentar o tempo de residência dos íons em seu interior e, conseqüentemente, aumentar o rendimento das reações íon-molécula. Aumentos da ordem de 30 a 260% foram observados nos experimentos testados.

CONTROL, ACQUISITION AND DATA TREATMENT FOR SEQUENTIAL MASS SPECTROMETRY IN A PENTAQUADRUPOLE INSTRUMENT: MULTIDIMENSIONAL SPECTRA MS^n ($n=1,2,3$) AND ION MONITORING AS A FUNCTION OF THE TIME AND IONIZATION POTENTIAL.

Valmir Fascio Juliano, Claudimir L. do Lago and Concetta Kascheres

 ABSTRACT

A pentaquadrupole mass spectrometer was interfaced to a 486DX 50-MHz microcomputer by an AT-MIO-16X interface card (National Instruments) and by another home made card.

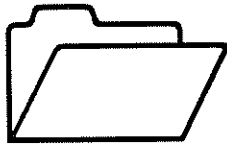
A set of programs was developed to control the instrument and to acquire data from multidimensional MS^1 , MS^2 and MS^3 experiments in sequential mass spectrometry. A program to monitorize ions as a function of the time and ionization potential was also developed. The time monitoring allows one to work with membrane inlet mass spectrometry as well as with other processes that yield gas products. The ionization potential is controlled by a DAC that allows one to acquire spectra while varying it. This technique is being evaluated as a tool for interpretation of spectra and elimination of contamination in spectra.

All programs were developed in Turbo Pascal 1.5 for Windows using object oriented programming and event-driven programming to allow complete control of the system as well as its upgrading.

The effect of a robust digital filter, based on the moving median method, was studied. This method was compared to the moving average procedure, commonly used in analytical signal filtering. Later, this method was implemented in all data acquisition programs and good results were obtained.

An oscillator circuit was connected to the potential controller of the focussing lens at the exit of the first reaction quadrupole to increase the residence time of the ions inside it and therefore increase the yield of the ion-molecule reactions. For the reactions studied the yields were increased in the range of 30 to 260%.

CAPÍTULO I



I. OBJETIVOS

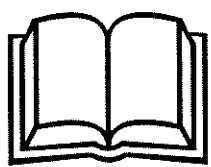
A espectrometria de massas seqüencial, embora não seja uma técnica recente, ainda tem sido alvo de muitos pesquisadores que buscam utilizá-la para desenvolver metodologias de análise, como ferramenta de trabalho na elucidação de mecanismos de reação ou como um “tubo de ensaio” para reações íon-molécula na fase gasosa.

No entanto, a espectrometria de massas seqüencial, envolvendo a obtenção de espectros multidimensionais, só é possível em espectrômetros de massas com dois ou mais analisadores de massas, quer seja um equipamento híbrido, um triploquadrupolar ou um pentaquadrupolar.

Até 1993 só existiam notícias de três equipamentos pentaquadrupolares, sendo que somente um deles gerou trabalhos publicados na literatura. Neste mesmo ano foi instalado no IQ-UNICAMP um equipamento pentaquadrupolar da Extrel. No entanto, este equipamento só possuía controles manuais. Assim, o objetivo principal deste trabalho era promover o interfaceamento do espectrômetro a um microcomputador através do desenvolvimento de circuitos eletrônicos e programas para aquisição de dados.

Devido às amplas possibilidades de controle do instrumento – indo dos elementos da fonte de ionização, passando pelos quadrupolos até o detector – achamos por bem controlar os três quadrupolos de varredura, com o objetivo de torná-lo o primeiro espectrômetro pentaquadrupolar capaz de adquirir espectros 4D. Devido ao volume e complexidade das informações, a tônica deste trabalho deveria ser sempre a busca de inovações e otimização dos processos de controle, aquisição e tratamento dos dados.

CAPÍTULO II



II. INTRODUÇÃO

No início deste século surgia uma técnica de análise que, algumas décadas depois, proporcionou grandes avanços nos estudos para determinação de estruturas de compostos desconhecidos. Era a técnica de espectrometria de massas que estava sendo introduzida pelos trabalhos de J. J. Thomson e pelos posteriores avanços feitos por F. W. Aston, A. J. Dempster, K. T. Bainbridge, A. O. Nier e outros. Aston em 1919 utilizou seu espectrógrafo de massa para a identificação de isótopos de hélio. Pouco tempo depois, Dempster em seu espectrômetro de massa, no qual o feixe de íons era registrado eletrometricamente, descobriu os isótopos de magnésio. Desde então, várias modificações foram introduzidas tanto na detecção quanto na separação e nos métodos de ionização, mas o princípio de funcionamento continua o mesmo.^[1]

Em geral, uma amostra sólida ou líquida é levada ao estado gasoso para em seguida ser ionizada por impacto com um feixe de elétrons ou com uma molécula ionizada. Estando no estado gasoso, os íons assim formados, podem ter seus movimentos controlados externamente por campos elétricos, magnéticos ou ambos. Desta forma, um íon pode ser separado de outros íons de acordo com sua relação massa/carga (m/z) para posteriormente ser analisado. Cada etapa necessária para uma análise em espectrometria de massas é discutida a seguir:

II.1. IONIZAÇÃO

Existem vários métodos que permitem a obtenção de íons positivos e negativos, além da fragmentação da molécula da amostra.^[2] Dentre eles os principais são a ionização por impacto de elétrons e ionização química, cada um com suas vantagens e desvantagens conforme é descrito a seguir:

• **Impacto de Elétrons:** Um feixe de elétrons é produzido a partir de um filamento de rênio ou tungstênio submetido a uma diferença de potencial com corrente controlada. Este filamento fica posicionado dentro de um campo elétrico o qual acelera os elétrons produzidos. Assim, as moléculas gasosas dentro da fonte de ionização, se ionizam através da colisão com os elétrons em um processo como:



Caso a energia do elétron seja superior à necessária para ocorrer este processo, poderão ocorrer outros processos, gerando uma grande quantidade de fragmentos e rearranjos com uma ou mais cargas.

A pressão na fonte de ionização, tipicamente, fica por volta de 10^{-4} a 10^{-5} torr e o tempo de permanência do íon na fonte é, aproximadamente, 10^{-6} s. Assim, colisões entre íons e entre íons e moléculas são muito pouco prováveis, significando que os íons moleculares formados com excesso de energia só poderão fragmentar-se por processos unimoleculares.

As principais desvantagens deste método são:

- O processo é bastante energético, levando, em alguns casos, a fragmentação excessiva da molécula, impedindo a identificação do peso molecular do composto analisado;
- O caráter radicalar do íon molecular promove rearranjos que, as vezes, dificulta a interpretação.

A principal vantagem, porém um pouco questionável, é a reprodutibilidade dos espectros que permite formar uma biblioteca com uma coleção extensa de espectros de massas.

•**Ionização Química:** Neste método um “gás” reagente (que pode ser o vapor de um líquido) é introduzido juntamente com a amostra, mas em uma quantidade muito maior, em uma proporção de 1 torr de gás para 10^{-4} torr de amostra. Este gás é preferencialmente ionizado pelo impacto dos elétrons e os seus íons promovem a ionização das moléculas da amostra.

A principal vantagem deste método é que a quantidade de energia transferida no processo de ionização geralmente é pequena. Se a espécie ionizante possui prótons, a energia transferida depende da diferença em afinidade protônica entre esta espécie e a molécula da amostra. Assim, quanto menor a diferença, menor será a energia transferida e menor será a formação de fragmentos (ionização mais suave). Por outro lado, se a espécie ionizante não possui prótons, como o He^+ , somente carga será transferida no processo. Desta maneira, pela utilização de gases reagentes diferentes, podemos controlar a quantidade de fragmentos e, assim, a obtenção de um espectro com a presença do pico correspondente ao íon molecular.

II.2. ANÁLISE DE m/z

Para se analisar um íon, este necessita estar fora da câmara de ionização, o que é conseguido através do controle externo por um campo elétrico. Uma partícula carregada eletricamente será acelerada por este campo elétrico em uma direção determinada pelo sinal de sua

carga. O ganho de energia cinética da partícula será igual ao produto de sua carga pela diferença de potencial elétrico ($E_c = zV$).

Uma vez fora da câmara de ionização, o íon é introduzido no analisador de massas, o qual pode ser de vários tipos: setor magnético, setor magnético/eletrostático, tempo de voo (TOF - *time of flight*), filtro quadrupolar e outros mais raros. Os mais comuns são o de setor magnético/eletrostático e de filtro quadrupolar, sendo este último o que nos interessa.

O filtro quadrupolar é constituído por quatro barras cilíndricas ou hiperbólicas perfeitamente alinhadas (figura II.1). As barras diametralmente opostas (eixo x) são submetidas ao mesmo potencial mas de sinal oposto em relação às outras duas barras (eixo y). O potencial aplicado nos pares de barras é a soma de um potencial contínuo de intensidade V_{dc} com um potencial alternado de radiofrequência de amplitude V_{rf} .

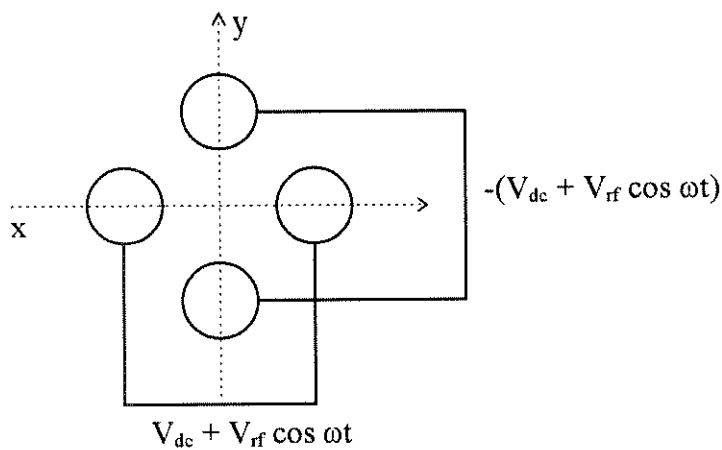


Figura II.1: Arranjo de um filtro quadrupolar. Os íons entram no filtro perpendicularmente ao plano da figura (eixo z). Os potenciais aplicados aos eixos x e y determinam a trajetória do íon e qual a relação m/z que atravessará o filtro e atingirá o detector.

Como a amplitude do sinal alternado é, em módulo, maior que o contínuo, durante uma parte do ciclo as polaridades se invertem. Assim, um íon, ao entrar na região do quadrupolo, será inicialmente atraído por um conjunto de barras que possuam polaridade oposta à sua e repelido pelo outro conjunto. Esta situação faz com que sua trajetória se alterne, provocando oscilação em seu movimento. Esta oscilação pode provocar uma instabilidade suficiente para promover a sua colisão com as barras ou o seu escape da região do quadrupolo. Esta oscilação depende da razão

massa/carga do íon, assim somente íons com uma determinada razão m/z conseguirão atravessar o analisador.

Em outras palavras, como a energia cinética de todos os íons é a mesma, pois depende apenas dos potenciais de aceleração, os íons de massas mais baixas possuem uma velocidade maior que os íons de massas mais altas. Assim, no plano XZ, se um íon é muito pesado ou a frequência é alta, o íon tende a sentir somente o efeito do potencial médio aplicado às barras. Isto é, os íons mais pesados são mais influenciados pelo potencial dc positivo, tendendo a ficarem focalizados na posição central entre as barras. Por outro lado, se os íons são muito leves, a trajetória é afetada pela rápida variação do potencial rf, fazendo com que eles colidam com as barras e se transformem em espécies neutras. Isto é, estes íons são filtrados pelo analisador com base em suas relações m/z . No plano YZ, da mesma maneira, os íons pesados somente tendem a sentir a influência do valor médio do potencial dc aplicado às barras, o qual é negativo. Isto faz com que eles sejam eliminados do feixe de íons. Por outro lado, se o íon é suficientemente leve, ele pode responder a ação focalizadora que resulta quando a porção positiva do potencial rf torna-se maior que o potencial dc negativo.

É possível demonstrar^[2,3] que para selecionar íons dentro de uma faixa estreita de m/z , aproximadamente 1 u, é necessário que as equações II.1 e II.2 sejam obedecidas:

$$a = \frac{8zV_{dc}}{mr_0^2\omega^2} \quad (II.1)$$

$$q = \frac{4zV_{rf}}{mr_0^2\omega^2} \quad (II.2)$$

onde r_0 é a metade da distância entre as superfícies das barras opostas e os valores de a e q devem ser o mais próximo possível de 0,2367 e 0,7060, respectivamente, para se obter o máximo de resolução. Sendo a e q constantes, nota-se que deve haver um compromisso entre V_{dc} e V_{rf} .

Assim, temos:

$$V_{dc} = \frac{a}{2q} V_{rf} \quad (II.3)$$

Como, em geral, a frequência e a distância entre as barras também são constantes, a razão m/z varia linearmente com a amplitude dos potenciais V_{dc} e V_{rf} em uma relação mostrada nas equações II.4 e II.5.

$$\frac{m}{z} = \frac{8}{ar_0^2 \omega^2} V_{dc}$$

(II.4)

$$\frac{m}{z} = \frac{4}{qr_0^2 \omega^2} V_{rf}$$

(II.5)

A maneira usual de se adquirir um espectro de massas é variar linearmente com o tempo os potenciais aplicados às barras, mantendo-se razão V_{rf}/V_{dc} constante, e registrar o sinal do detector.^[2-4]

 II.3. DETECÇÃO

Em virtude da quantidade de amostra utilizada em uma análise ser pequena, associada ao fato da baixa eficiência na ionização (aproximadamente uma molécula em cada dez mil é ionizada) e as perdas no processo de separação, a corrente iônica total é da ordem de 10^{-17} a 10^{-9} amperes. Para que seja possível fazer a detecção de uma espécie é necessário, portanto, a amplificação desta corrente. Um dispositivo muito utilizado tal amplificação é o Multiplicador de Elétrons (ME). Um modelo mais antigo é composto por placas de Cu-Be (dinodos), dispostas em sequência, submetidas a um gradiente de potencial elétrico da ordem de kilovolts. Um íon ao atingir a superfície do primeiro dinodo faz com que sejam emitidos, em média, dois elétrons, que por ação do gradiente de potencial caminharão na direção do segundo dinodo, repetindo o processo. Desta maneira a corrente aumenta em progressão geométrica, atingido valores detectáveis por qualquer amplificador eletrônico após pouco mais de uma dezena de dinodos.

Um modelo mais atual de ME possui dinodos contínuos, assim, os elétrons podem seguir muitos caminhos, tornando indeterminado o número de estágios de multiplicação. No entanto, a amplificação não é infinita, saturando após 10^9 elétrons em um tempo da ordem de 10^{-8} segundos. Porém, esta saturação pode ser controlada através do gradiente de potencial aplicado.

 II.4. ESPECTROMETRIA DE MASSAS SEQUENCIAL

Nos últimos vinte e sete anos foi conseguido um grande avanço na área de espectrometria de massas devido ao desenvolvimento dos equipamentos e dos métodos de medição. Foi assim que, por volta de 1968, surgiu a espectrometria de massas sequencial, com dois estágios (MS^2) e

denominada de espectrometria de massas em Tandem.^[5] A espectrometria de massas sequencial ou de estágios múltiplos envolve o acoplamento em série de analisadores de massas, com a colocação de câmaras de colisão ou reação entre os analisadores. Com dois analisadores tem-se a espectrometria de segunda ordem MS^2 , com três a de terceira ordem MS^3 e assim por diante.

Devido às várias etapas da espectrometria de massas sequencial, a transmissão dos íons desde a fonte até o detector diminui drasticamente, porém os instrumentos mais modernos, microprocessados e equipados com dispositivos capazes de detectar pequenas quantidades de íons (detectores multiplicadores de elétrons), permitiram o desenvolvimento desta técnica que tem se mostrado altamente eficiente para estudos de reações em fase gasosa e em aplicações analíticas como análise de traços ($\sim 10^{-12}$ g) em análise ambiental e em fluidos biológicos.^[6]

O desenvolvimento da espectrometria de massas sequencial é baseado em novos conceitos: a utilização de analisadores de massas acoplados em sequência como instrumentos de separação de componentes individuais presentes em misturas e a execução de uma série de operações sobre um fluxo de íons.^[7]

De uma maneira bem simples, a espectrometria de massas sequencial é uma técnica pela qual uma espécie iônica presente em uma mistura complexa é separada dessa mistura através do primeiro analisador de massas, reagindo com uma substância ou sendo fragmentada pela colisão com um gás em uma câmara de reação/colisão, de maneira que os produtos ou fragmentos resultantes possam ser analisados pelo segundo analisador de massas e assim sucessivamente. O processo representando a aquisição de um espectro de massas de um íon especificamente selecionado é ilustrado na figura II.2.

Neste sentido a MS^2 é análoga à cromatografia gasosa acoplada à espectrometria de massas (GC/MS) ou cromatografia líquida acoplada à espectrometria de massas (LC/MS) quando o íon selecionado é o íon molecular. O íon selecionado no primeiro analisador de massas, o íon pai, corresponde ao componente que elui da coluna cromatográfica. Os íons que resultam da fragmentação do íon pai na MS^2 correspondem aos íons formados a partir da ionização na fonte, do composto eluído da coluna cromatográfica em GC/MS ou LC/MS.^[8]

A espectrometria de massas sequencial caracteriza-se por uma ampla aplicabilidade, uma vez que um maior número de compostos podem ser analisados pela espectrometria de massas do que pela cromatografia gasosa, tais como compostos com baixa volatilidade, termolábeis ou com

alto peso molecular. Além da disponibilidade de novos tipos de espectros de massas, tais como perfis de peso molecular para todos os compostos com um tipo de fragmentação característica presentes em uma amostra, o acesso a diferentes constituintes das misturas é simultâneo, ao invés de ser sequencial. Isto torna o experimento total extremamente rápido. Em muitos casos não é necessária uma pré-preparação ou mesmo a preparação da amostra, seja um fragmento de tecido vegetal, fluidos biológicos, amostras geológicas, etc.^[5]

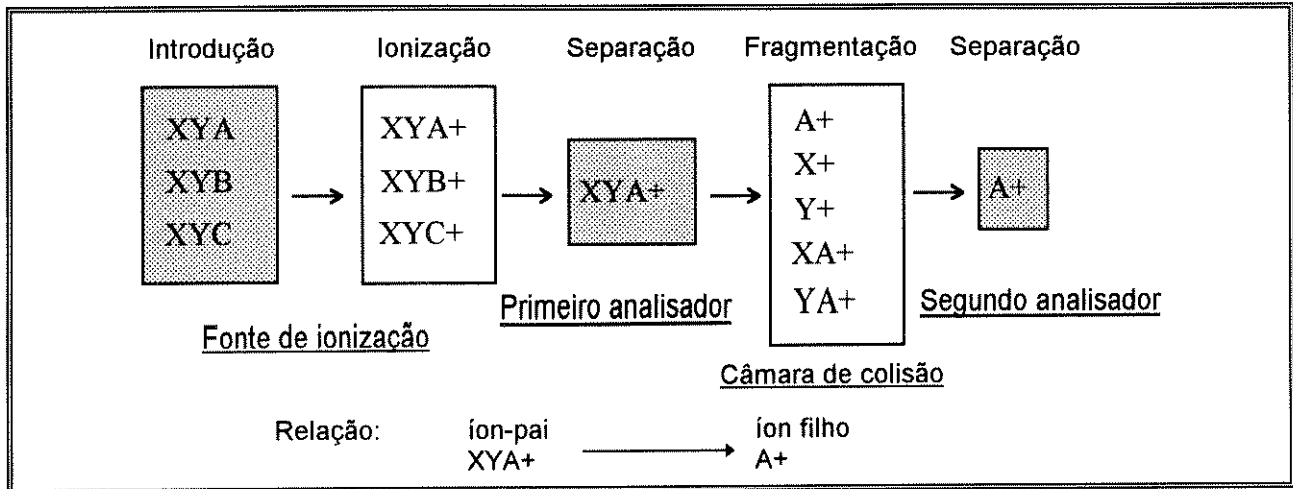


Figura II.2: Processos realizados durante a análise de uma amostra (mistura dos componentes XYA , XYB , XYC) através da espectrometria de massas sequencial de segunda ordem (MS^2), sendo que estes processos podem ser efetuados para todas as moléculas ionizadas, bem como para os fragmentos produzidos.

A espectrometria de massas sequencial tem sido executada em diversos aparelhos: triploquadrupolares (QqQ), setoriais magnético-elétrico (BE, EB, EBE, BEB, BEE, EBEB ou BEEB), quadrupolares com armadilha de íons (*ion-trap*), ressonância ciclotrônica de íons (*ICR*) e dispositivos híbridos: BEQ ou EBQ e armadilha de íons/tempo de voo.^[5,8] Os equipamentos de ressonância ciclotrônica e armadilha de íons executam os experimentos de espectrometria de massas sequencial no tempo, enquanto os demais executam no espaço.

Dentre estes equipamentos citados, o mais popular é o instrumento triploquadrupolar. Uma implementação mais sofisticada, o espectrômetro de massas pentaquadrupolar, foi introduzida por dois grupos de pesquisadores em 1986.^[9,10] No entanto, foi outro grupo^[11] que desenvolveu e demonstrou a utilização de um equipamento similar quatro anos depois.

Embora seja possível executar experimentos MS^n em um equipamento quadrupolar com armadilha de íons ($n \leq 13$),^[12] a obtenção de espectros multidimensionais só é possível nos equipamentos triplo e pentaquadrupolares ou nos híbridos (BEqQ, EBqQ ou outras combinações).

Em um equipamento do tipo armadilha de íons, a execução de um experimento MS^3 se processa da maneira descrita a seguir. Enquanto um determinado íon é aprisionado dentro do quadrupolo para ser analisado, os demais são descartados e a introdução da amostra é interrompida. Assim, neste momento, dentro do aparelho só existe uma espécie. Em seguida é introduzido a substância com a qual o íon aprisionado irá reagir. O próximo passo é aprisionar o aduto da reação eliminando as demais espécies bem como a substância reagente. Posteriormente, este aduto é excitado (acelerado), através da aplicação de um potencial de RF suplementar, para que se fragmente pela colisão com o gás hélio que, obrigatoriamente, está presente no equipamento. Finalmente, cada fragmento vai sendo liberado sequencialmente, através de uma varredura convencional do potencial RF, de modo a chegar ao detector para ser registrado.^[8]

Assim, a menos que seja implementado uma série de modificações de *hardware* e *software* de modo a fazer o controle da varredura associado ao controle de válvulas automáticas de injeção de amostra e reagente, só é possível obter espectros bidimensionais em um equipamento com armadilha de íons.

No caso de equipamentos híbridos, triplo e pentaquadrupolares, a obtenção de espectros de massa multidimensionais é possível desde que se tenha o controle sobre dois ou mais analisadores de massa simultaneamente. Por exemplo, em um experimento MS^2 , enquanto um analisador de massa estiver analisando um íon, o outro pode estar analisando os fragmentos obtidos pela colisão deste com um gás em uma câmara de reação/colisão. Se os dois analisadores executam uma varredura de massas, obtém-se um espectro de massas MS^2 tridimensional (dois eixos de massa e um de intensidade). Caso contrário, se somente um analisador executa a varredura, obtém-se um espectro de massas MS^2 bidimensional.

A obtenção de espectros MS^3 3D e 4D, no entanto, só é possível com o controle simultâneo de três analisadores de massa. Este é o caso do espectrômetro de massas pentaquadrupolar, que é descrito a seguir.

II.5. O INSTRUMENTO PENTAQUADROPOLAR (QqQqQ)

Como o próprio nome diz, este equipamento é constituído por cinco arranjos quadrupolares, conforme pode ser visto no diagrama esquemático apresentado na figura II.3.

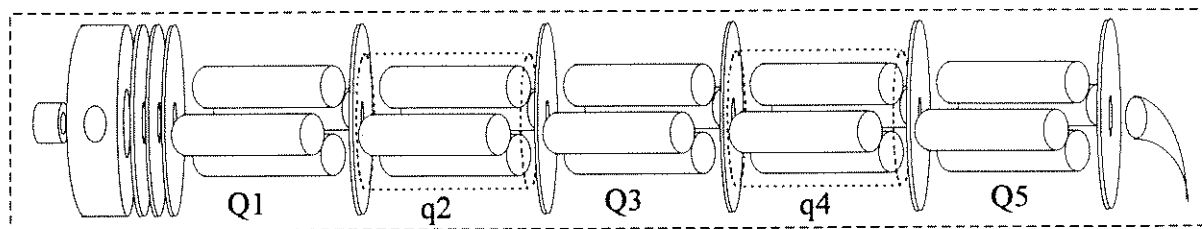


Figura II.3: Diagrama esquemático do espectrômetro de massas pentaquadrupolar (Q = quadrupolo de varredura, q = quadrupolo de reação/colisão).

Este instrumento utiliza três quadrupolos como filtros de massas (quadrupolos de varredura) e dois quadrupolos adicionais nos quais ocorrem reações ou dissociações (câmaras de colisão). Assim, experimentos MS/MS/MS (MS^3) podem ser executados gerando espectros de massas 2D, 3D e 4D, se um, dois ou três quadrupolos forem utilizados para varredura. Os espectros 4D (três dimensões em massa mais a abundância dos íons) não foram reportados até o momento porque o computador controla somente dois quadrupolos ao mesmo tempo.^[11]

Existem várias possibilidades de combinações em relação a seleção ou varredura efetuadas por cada um dos quadrupolos. Estas várias possibilidades dão origem a uma ampla variedade de modos de varredura, os quais podem fornecer as mais diversificadas informações sobre a estrutura de íons e seus processos de fragmentação ou reação.^[13]

Os modos de varredura possíveis de serem obtidos com o espectrômetro de massas pentaquadrupolar estão mostrados na figura II.4. As duas primeiras sequências (MS^1 e MS^2) também são possíveis em equipamentos com somente dois analisadores de massa, porém a última (MS^3) é exclusiva de equipamentos com três analisadores.

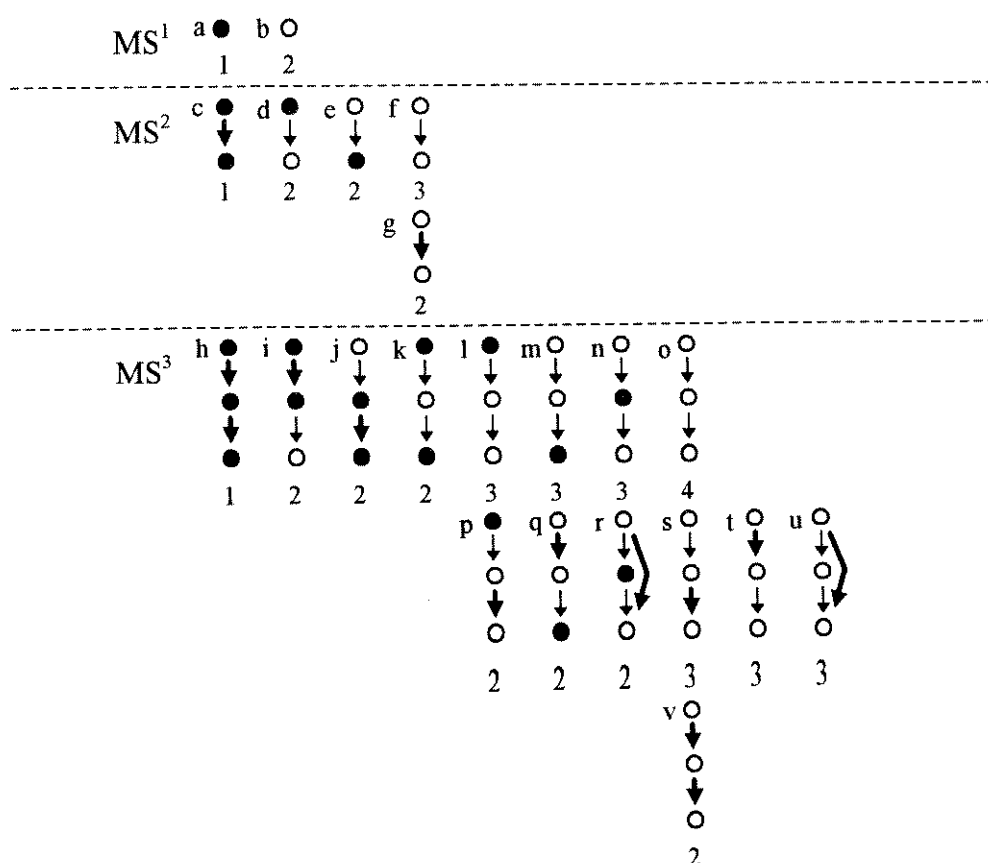


Figura II.4: Modos de varredura possíveis de serem obtidos com o espectrômetro de massas pentaquadrupolar. As letras à esquerda dos símbolos identificam o tipo de varredura e os números abaixo indicam a dimensionalidade do espectro. As setas finas (↓) indicam uma transição de massa variável, enquanto as grossas (↓) indicam uma transição de massa fixa. As circunferências vazias (O) indicam varredura de massa, enquanto as cheias (●) indicam massa fixa.^[13]

Deve ficar bem claro que os dados obtidos por varreduras de menor dimensionalidade sempre estão contidos em uma ou mais varreduras de maior dimensionalidade. Por exemplo, na figura II.5, os dados de monitoramento de reação (c) estão inclusos nos dados de monitoramento de produtos (d) e nos dados de monitoramento de íons precursores (e). Estes dois modos (d, e) e o modo de relação funcional (perda ou ganho neutro - g) são subconjuntos do modo de varredura tridimensional (f). A extração de dados de menor dimensionalidade é um processo prático e correto que envolve, simplesmente, a retirada de dados que estão incluídos em um ponto, linha, plano ou, em geral, em um subespaço de dimensão $n-1$.^[13]

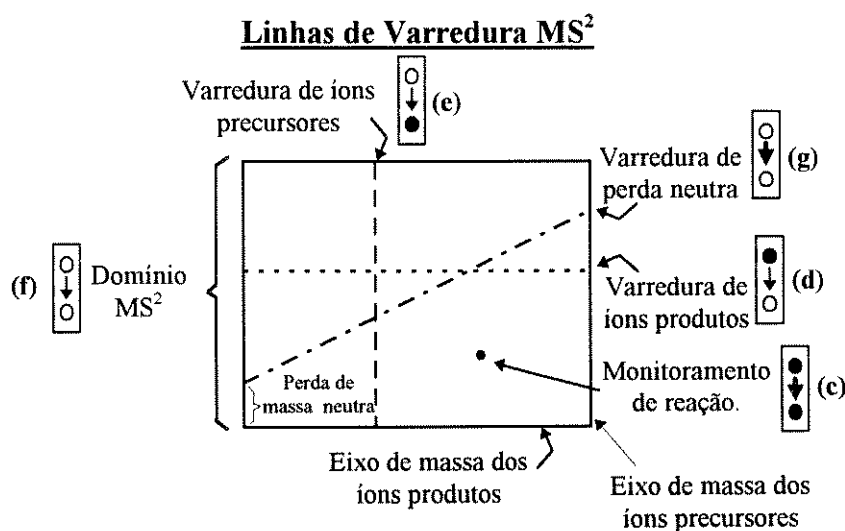


Figura II.5: Varreduras de dimensionalidade menor que podem ser extraídas de outra de maior dimensão. A figura mostra todas as cinco varreduras do domínio MS^2 (c - g).^[13]

Outra operação, chamada de projeção, também pode ser executada sobre os dados de um conjunto de dimensão n forçando-os para um outro conjunto de menor dimensionalidade ($n-1$). Conceitualmente, este processo é oposto à geração de um espectro MS^2 a partir de um espectro MS^1 . Neste processo algumas informações são perdidas em função da compactação dos dados em um conjunto menor de dados. A projeção de dados, não utilizada anteriormente em espectrometria de massas, tem-se mostrado uma operação potencialmente útil em certos casos.^[13] Por exemplo, na figura II.6, o conjunto de dados completo de MS^2 (f), pode ser projetado sobre o eixo de massa dos produtos ou dos íons precursores.

A projeção sobre o eixo de massa dos íons precursores gera um espectro de massas anterior, o qual representa uma forma do espectro de massa inicial antes de ser distorcido pela reação. Este é muito similar ao espectro de massa original, porém com algumas pequenas diferenças na abundância dos íons devido à ocorrência de processos de colisão que leva a neutralização e dissociação. A projeção sobre o eixo dos íons produtos gera um espectro de massas posterior (ou residual), o qual representa os efeitos da reação, incluindo dissociação e outros processos, sendo também ligeiramente diferente do espectro de massas original.

Estes dois métodos de projeção removem uma variável e fixam-na, reduzindo ordem dos dados e não simplesmente a sua dimensionalidade.

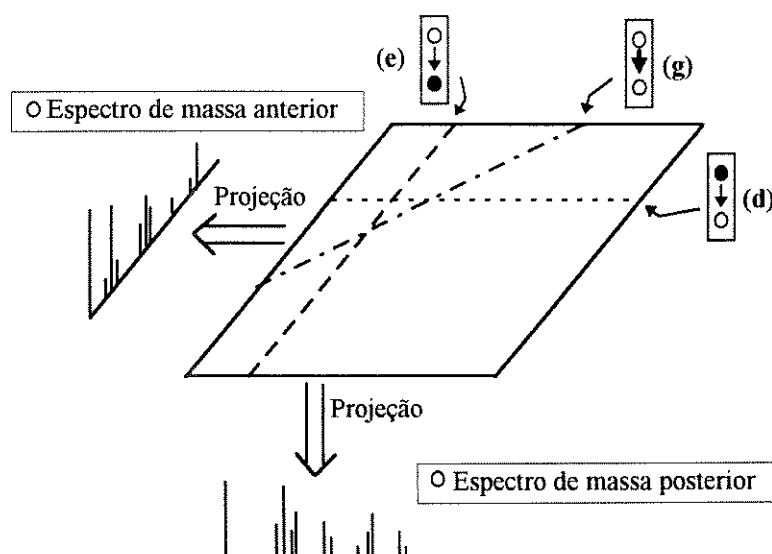


Figura II.6: Exemplos de extração e projeção sobre um domínio MS^2 .^[13]

A tabela II.1 resume os modos de varredura MS^3 , seus nomes e a dimensionalidade de cada um. O experimento de monitoramento consecutivo de reação (h) não possui dimensão em massa, ou seja, a abundância é monitorada em função do tempo. As demais combinações incluem, no mínimo, um eixo de massa. A figura II.7 mostra um domínio MS^3 do qual extrações e projeções podem ser feitas de modo a proporcionar uma investigação mais detalhada de um problema em particular.

Similarmente às extrações e projeções, um conjunto de dados MS^3 pode ser construído a partir de dados MS^2 através de operações lógicas “e” (intersecção - $\cap$). No entanto, este tipo de operação somente pode ser executada com dados de mesma dimensionalidade em massa, conforme mostrado na figura II.8.

Espectros MS^3 adquiridos em aparelhos pentaquadrupolares tem sido aplicados com grande sucesso em vários e diversificados estudos tais como mecanismo de fragmentação de hidrocarbonetos,^[14] sequenciamento de peptídeos,^[15] degradação térmica de polímeros,^[16] diferenciação de isômeros,^[17] estudo da química em fase gasosa de íons distônicos,^[18] estudo de reações orgânicas em fase gasosa,^[19] determinação de propriedades termodinâmicas tais como

afinidade por próton e por Cl^+ ,^[20] reações diagnósticas para estrutura de íons,^[21] determinação de estrutura e mecanismo de reações com íons isoméricos,^[22] entre outros.

Em nosso espectrômetro de massas pentaquadrupolar, os três quadrupolos de varredura são controlados pelo computador, o que permite obter espectros de massa MS^3 4D inéditos, além dos espectros 2D e 3D. Este controle simultâneo dos três quadrupolos permite, também, a execução inédita de experimentos MS^3 envolvendo perda neutra e ganho neutro consecutivos. Estes métodos de varredura proporcionam um estudo detalhado e completo de processos envolvendo íons e moléculas na fase gasosa.^[23]

Símbolo	Nome	Identificação	Dimensão
●→●→●	Monitoramento de reação consecutiva	h	1 (0)
●→●→○	Produtos sequenciais	i	2 (1)
○→●→●	Precursores sequenciais	j	2 (1)
●→○→●	Intermediário de reação	k	2 (1)
●→○→○	Domínio de produtos intermediários	l	3 (2)
○→○→●	Domínio de precursores intermediários	m	3 (2)
○→●→○	Varredura de precursores e produtos	n	3 (2)
○→○→○	Domínio MS^3	o	4 (3)
●→○→○	Produtos selecionados	p	2 (1)
○→○→●	Precursores selecionados	q	2 (1)
○→●→○	Perda/ganho neutra selecionada	r	2 (1)
○→○→○	Domínio de produtos selecionados	s	3 (2)
○→○→○	Domínio de precursores selecionados	t	3 (2)
○→○→○	Domínio de Perda/ganho neutra selecionada	u	3 (2)
○→○→○	Perda/ganho neutra consecutiva	v	2 (1)

Tabela II.1: Modos de varredura MS^3 . A coluna rotulada identificação refere-se a figura II.4. Os números entre parenteses na coluna dimensão refere-se ao número de eixos de massa.

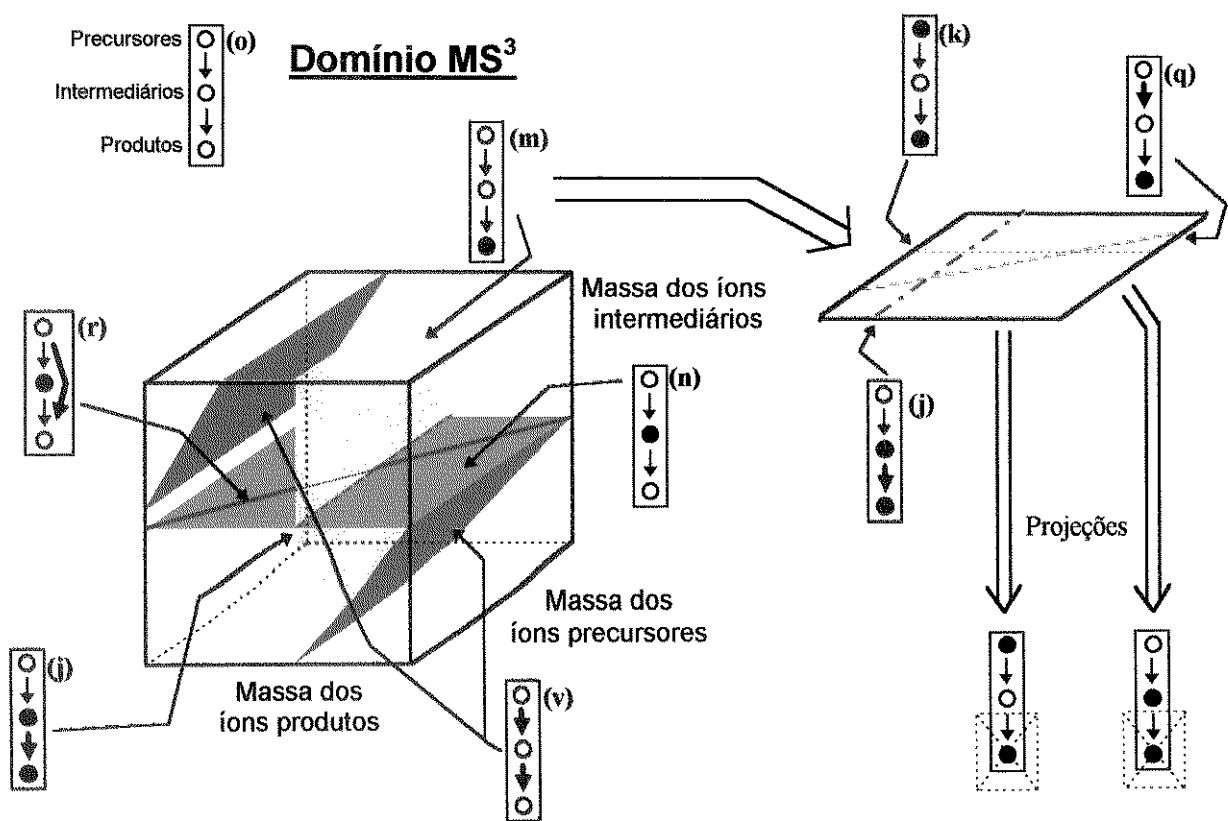


Figura II.7: Domínio MS³ (eixo de abundância não incluso) com algumas extrações e projeções.

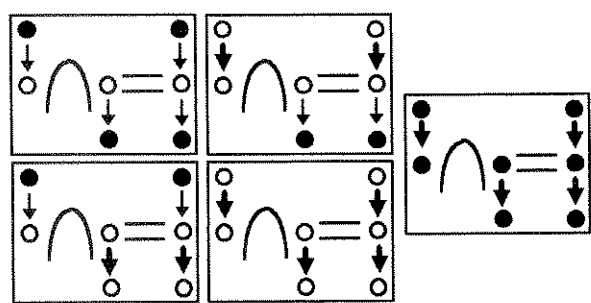


Figura II.8: Construção de um conjunto de dados MS³ a partir de dados MS².

II.6. BIBLIOGRAFIA

1. McDowell, C. A. *Mass Spectrometry*, McGraw Hill: New York, 1963, p. 45.
2. Kiser, R. W. *Introduction to Mass Spectrometry and Its Applications*, Prentice-Hall: New Jersey, 1965.
3. Miller, P. E.; Denton, M. B. *J. Chem. Ed.* **1986**, 63(7), 617.
4. Silverstein, R. M.; Bassler, G. C.; Morrill, T. C. *Identificação Espectrométrica de Compostos Orgânicos*, Editora Guanabara: Rio de Janeiro, 3ª ed., 1987, p. 8.
5. McLafferty, F. W. (ed.) *Tandem Mass Spectrometry*, Wiley: New York, 1983.
6. McLafferty, F. W. *Org. Mass Spectrom.* **1993**, 28, 1403.
7. Cooks, R. G.; Glish, G. L. *Chem. Eng. News* **1981**, 50, 40.
8. Busch, K. L.; Glish, G. L.; McLuckey, S. A. *Mass Spectrometry/Mass Spectrometry: Techniques and Applications of Tandem Mass Spectrometry*, VCH: New York, 1988.
9. Morrison, J. D.; Stanney, D. A. Tedder, J. *Proc. 34th Am. Soc. Mass Spectrom. Conf. Mass Spectrom. Allied Topics, Cincinnati, OH*, **1986**, p. 222.
10. Mestdagh, H.; Morin, N.; Rolando, C.; Beaugrand, C.; DeMaack, F. *Proc. 34th Am. Soc. Mass Spectrom. Conf. Mass Spectrom. Allied Topics, Cincinnati, OH*, **1986**, p. 799.
11. Schwartz, J. C.; Schey, K. L.; Cooks, R. G. *Int. J. Mass Spectrom. Ion Processes* **1990**, 101,1.
12. March, R. E. *Int. J. Mass Spectrom. Ion Processes* **1992**, 118/119, 71-135.
13. Schwartz, J. C.; Wade, A. P.; Enke, C. G.; Cooks, R. G. *Anal. Chem.* **1990**, 62, 1809.
14. Cooks, R. G.; Amy, J. W.; Bier, M. E.; Schwartz, J. C.; Schey, K. L. *Advances in Mass Spectrometry* **1989**, 11, 33.
15. (a) Schey, K. L.; Schwartz, J. C.; Cooks, R. G. *Rapid Commun. Mass Spectrom.* **1989**, 2, 130. (b) Schey, K. L.; Schwartz, J. C.; Cooks, R. G. *Rapid Commun. Mass Spectrom.* **1989**, 3, 305.
16. (a) Grant, E. R.; Cooks, R. G. *Science* **1990**, 250, 61. (b) Majumdar, T. K.; Eberlin, M. N.; Green, M.; Munhoz, B.; Reidy M. P.; Cooks, R. G. *J. Am. Soc. Mass Spectrom.* **1991**, 2, 130.
17. Eberlin, M. N.; Majumdar, T. K.; Cooks, R. G. *J. Am. Chem. Soc.* **1992**, 114, 2884.
18. Gozzo, F. C.; Eberlin, M. N. *J. Am. Soc. Mass Spectrom.* **1995**, 6, 554.

19. (a) Kotiaho, T.; Eberlin, M. N.; Shay, B. J.; Cooks, R. G. *J. Am. Chem. Soc.* **1993**, 115, 1004.
(b) Eberlin, M. N.; Cooks, R. G. *J. Am. Chem. Soc.* **1993**, 115, 9226. (c) Eberlin, M. N.; Morgon, N. H.; Yang, S. S.; Shay, B. J.; Cooks, R. G. *J. Am. Soc. Mass Spectrom.* **1995**, 6, 1.
20. Eberlin, M. N.; Kotiaho, T.; Cooks, R. G. *J. Am. Chem. Soc.* **1994**, 116, 2457.
21. Eberlin, M. N.; Cooks, R. G. *Org. Mass Spectrom.* **1993**, 28, 679.
22. Eberlin, M. N.; Majumdar, T. K.; Cooks, R. G. *J. Am. Chem. Soc.* **1992**, 114, 2884.
23. Juliano, V. F.; Gozzo, F. C.; Eberlin, M. N.; Kascheres, C.; Lago, C. L. *Anal. Chem.* **1996**, 00, 00.

CAPÍTULO III



III. HARDWARE

Este capítulo apresenta a parte do trabalho que envolveu o desenvolvimento, modificação e a conexão de circuitos eletrônicos necessários para acoplar o espectrômetro de massas pentaquadrupolar ao microcomputador. Espera-se que este capítulo sirva como um guia para posteriores modificações ou implementações no equipamento. Embora também faça parte da seção de implementação de *hardware*, o circuito oscilador acoplado à lente de focalização de saída de q2, bem como os resultados obtidos com a sua introdução serão discutidos separadamente no capítulo IV.

§ III.1. IMPLEMENTAÇÃO

O espectrômetro de massas foi interfaceado com um microcomputador 486DX 50MHz (com 20 Mbytes de memória RAM, monitor SVGA colorido e disco rígido de 540 Mbytes) através de um cartão multifuncional AT-MIO-16X da National Instruments e de um outro cartão adicional construído em nosso laboratório.

A figura III.1 mostra o diagrama de blocos geral do sistema. Os quadrupolos Q1, Q3 e Q5 são analisadores de massa, enquanto q2 e q4 são câmaras de reação ou colisão. O instrumento e sua eletrônica de controle foram construídos pela Extrel Mass Spectrometry (Extrel Corporation). Todos os quadrupolos têm 1,9 cm de diâmetro, 20 cm de comprimento e trabalham na frequência de 880 kHz. Q1, Q3 e Q5 operam na faixa de m/z de 1 a 1000 u. A fonte de ionização possui receptáculo de íons (*ion volume*) removível que permite realizar a ionização tanto por impacto de elétrons quanto por ionização química. Para efetuar a troca do receptáculo de íons, sem interromper as atividades do equipamento, desenvolvemos a peça mostrada no apêndice III.A. O sistema de detecção consiste de um multiplicador de elétrons (ME) modelo Galileo que permite tanto a detecção de íons positivos quanto a de íons negativos.

Para controlar adequadamente o equipamento seria necessário um cartão de interface com conversores analógico/digital (ADC) de 12 bits para leitura do ME e digital/analógico (DAC) de 14 bits para o ajuste dos valores de m/z .

O ADC de 12 bits é necessário pois como a intensidade dos dos sinais nos espectros são as mais variadas possíveis, uma resolução menor causa perda de sinais de pequena intensidade, implicando na perda de informações importantes na elucidação das estruturas dos compostos analisados. Assim quanto maior a resolução, melhor será a discriminação dos sinais.

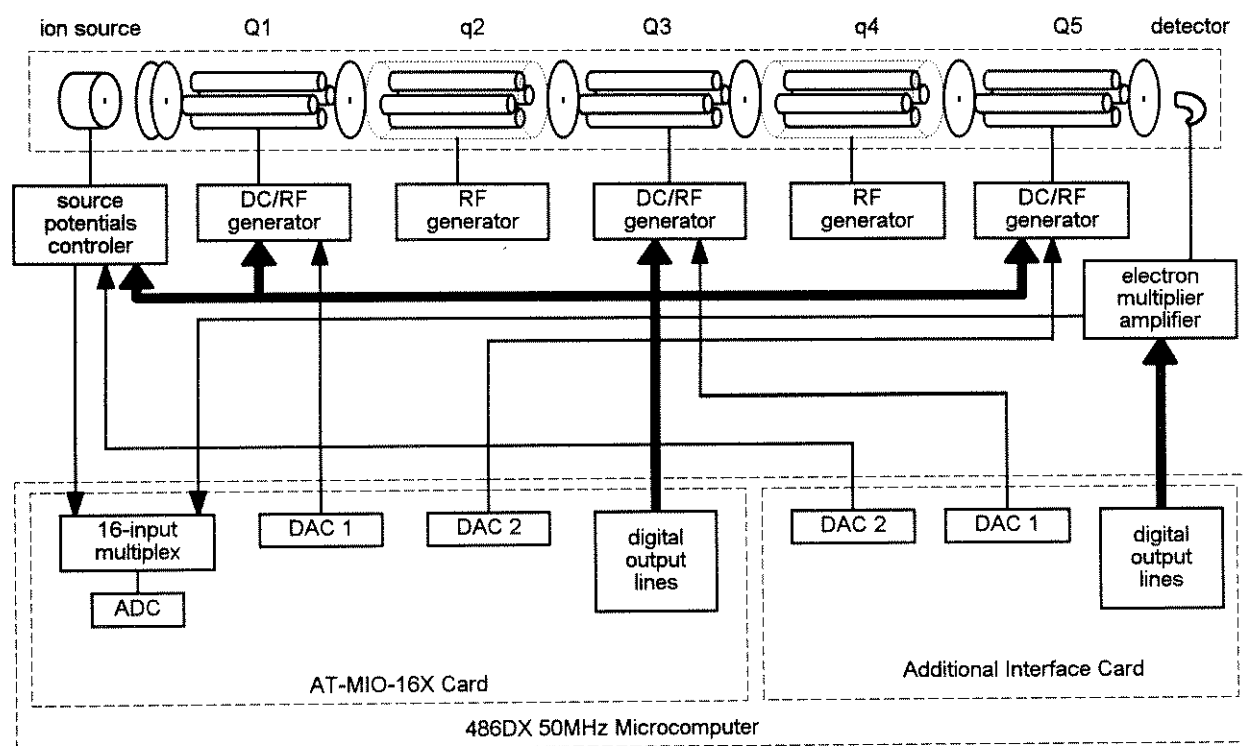


Figura III.1: Diagrama esquemático do sistema composto pelo espectrômetro de massas pentaquadrupolar, microcomputador e cartões de interface.

Para podermos ajustar valores de m/z de 0,1 u em uma faixa de 1 a 1000 u são necessários 10000 níveis de discriminação. Assim, um DAC com pelo menos 14 bits de resolução é necessário, pois possui 16384 (2^{14}) níveis. Como os DACs mais comuns encontrados são de 12 e 16 bits, só nos restou a opção do DAC de 16 bits. Assim, adquirimos o cartão AT-MIO-16X que demonstrava possuir características técnicas compatíveis ou superiores às necessárias.

Este cartão, então, foi instalado para controlar as principais funções do instrumento. Ele é constituído pelos seguintes elementos:

- Um ADC de 16 bits com 16 entradas simples ou 8 diferenciais capaz de adquirir dados a 100 kHz. Possui sete níveis de ganho: 1, 2, 5, 10, 20, 50 e 100, o que permite converter tanto sinais com intensidades altas (10 V) como com intensidades muito baixas ($\sim 3 \mu\text{V}$);
- Dois DACs independentes de 16 bits;
- Oito linhas digitais que podem ser configuradas como entrada ou saída de dados;
- Três contadores/temporizadores independentes de 16 bits com *clock* máximo de 5 MHz.

Cada um destes elementos executam uma ou mais funções, as quais são descritas a seguir:

- **DACs** - Ajustam os valores de m/z de Q1 e Q5. Trabalham na faixa de 0 a 10000 mV (resolução de 0,152 mV), sendo que para cada 1 mV tem-se 0,1 u.

- **ADC** - Um dos canais é usado para adquirir a abundância dos íons e outro para monitorar os potenciais dos elementos da fonte de ionização e a alta tensão do ME.

- **Linhas digitais** - Existem oito linhas dispostas em duas portas **A** e **B**. Todas foram configuradas para saída de dados. Três linhas da porta **A** são usadas para alternar os quadrupolos Q1, Q3 e Q5 entre os modos analisador de massa (RF/DC) e íons totais (somente-RF), enquanto as quatro linhas da porta **B** são utilizadas para selecionar os endereços de monitoração dos elementos da fonte de ionização e da alta-tensão do ME. A tabela III.1 mostra a função de cada linha digital da porta **A**, enquanto a tabela III.2 mostra os valores que devem ser enviados para a porta **B** de modo a acessar os elementos da fonte de ionização e a tensão do ME.

Para converter o valor retornado pela leitura destes elementos, as seguintes equações precisam ser utilizadas:

Para o ME: $L = 500 \times V - 105$

Para os demais elementos: $L = 40 \times V - 200$

onde: L é o valor da tensão de cada elemento e V é valor da tensão retornada pelo ADC, ambas em volts.

- **Contadores/Temporizadores** - São utilizados para contagem do tempo de estabilização dos quadrupolos. Cada quadropolo necessita de um tempo, que depende do “salto” de massa, para atingir o equilíbrio (ver item III.1.B).

LINHAS DIGITAIS DA PORTA A				
	A0	A1	A2	A3
Estado	Q1	Q3	Q5	
0	somente-RF	somente-RF	somente-RF	-
1	RF/DC	RF/DC	RF/DC	-

Tabela III.1: Identificação das linhas digitais da porta digital A.

B0	B1	B2	B3	VALOR DECIMAL	ELEMENTO MONITORADO
0	1	0	0	2	Bloco da fonte
1	0	1	0	5	Tensão do multiplicador de elétrons
0	0	0	1	8	Lente de focalização de saída de Q5
1	0	0	1	9	Repulsor
0	1	0	1	10	Lente de focalização de entrada de Q1
0	0	1	1	12	Lente extratora 1
1	0	1	1	13	Lente extratora 2

Tabela III.2: Valores de endereço para acesso dos elementos da fonte de ionização e da tensão do multiplicador de elétrons.

Como o cartão AT-MIO-16X possui somente dois DACs, um DAC de 16 bits adicional foi necessário para controlar Q3. Assim, desenvolvemos um outro cartão com o circuito integrado DAC725 da Burr-Brown^[1], conforme figura III.2. Este circuito integrado possui dois DACs independentes, o que nos levou a utilizar o DAC disponível para controlar o potencial de ionização (ver item III.1.C). Adicionalmente, o cartão contém um circuito integrado 74LS374 com capacidade de acionamento de oito linhas digitais, das quais duas foram utilizadas para ajustar o ganho do amplificador do ME em três níveis: baixo, médio e alto.

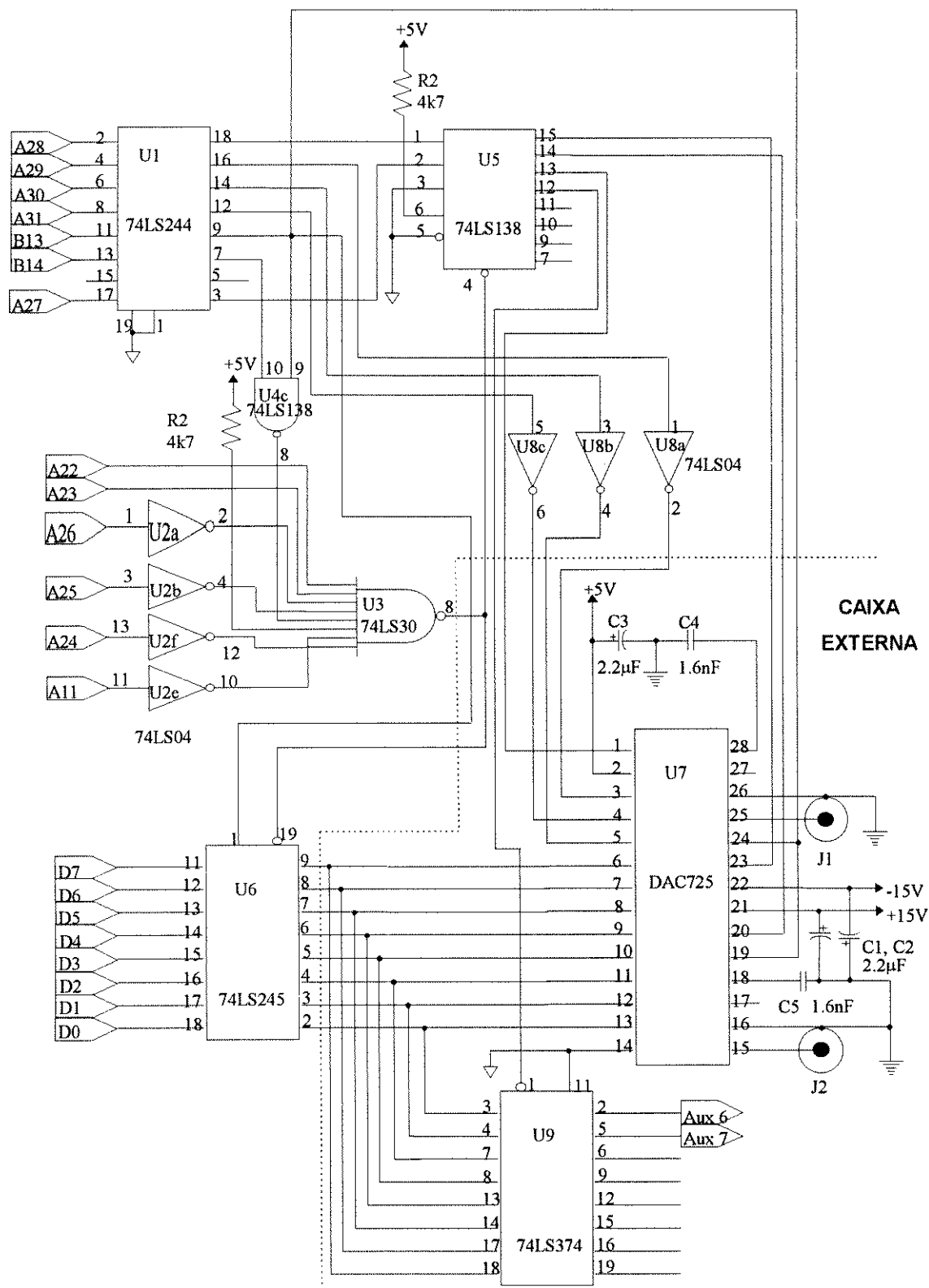


Figura III.2: Diagrama elétrico do cartão de interface adicional para controlar o terceiro quadrupolo (J1) e o potencial de ionização (J2).

A parte do circuito responsável pela decodificação de endereços do microcomputador e pelo controle de entrada e saída de dados (I/O) é composta pelos circuitos integrados U1, U2, U3, U4, U5, U6 e U8. Esta parte foi montada em um cartão separado, o qual está localizado dentro do microcomputador. A parte, contendo os circuitos integrados U7 (DAC725) e U9, foi montada em um outro cartão (linha pontilhada na figuraIII.2), o qual foi instalado em uma caixa plástica e afixado junto aos módulos de controle do equipamento.

 III.1.A ENDEREÇAMENTO

A tabela III.3 mostra os endereços de acesso das principais funções do cartão AT-MIO-16X que foram utilizados para o interfaceamento. Este cartão foi configurado para o endereço base 220H (220H a 23FH), canais de DMA 6 e 7, IRQ 12, saída analógica unipolar de 0 a 10 V e entrada analógica unipolar não diferencial de 0 a 10 V.

Deslocamento	Modo	Função
19H	Leitura (L)	Inicializa o circuito de aquisição do cartão
08H	Escrita (E)	Seleciona o canal de entrada do ADC
08H	E	Seleciona o ganho do amplificador de entrada do ADC
1DH	E	Inicia a conversão AD.
00H	L	Leitura do valor convertido pelo ADC.
10H	E	Envia o valor para o DAC 1
12H	E	Envia o valor para o DAC 2
14H e 16H	L / E	Configura, inicializa e lê os contadores/temporizadores
1CH	L / E	Linhas digitais

Tabela III.3: Endereços de I/O utilizados pelo cartão AT-MIO-16X. O deslocamento se refere ao valor que deve ser somado ao endereço base **220H** para acessar a função especificada.

A tabela III.4 mostra os endereços de I/O utilizados pelo cartão de interface adicional. Embora alguns endereços estejam sem função, eles não podem ser utilizados por outros dispositivos visto que estão reservados para o circuito integrado DAC725.

Endereço	Função
300H a 302H	Sem função
303H	Aplica o valor na saída do canal A
304H	Sem função
305H	Ajusta byte menos significativo do canal A
306H	Ajusta byte mais significativo do canal A
307H a 30AH	Sem função
30BH	Aplica o valor na saída do canal B
30CH	Sem função
30DH	Ajusta byte menos significativo do canal B
30EH	Ajusta byte mais significativo do canal B
30FH	Sem função
318H	Sensibilidade do eletrômetro do ME 1 → sensibilidade baixa 2 → sensibilidade alta 3 → sensibilidade media

Tabela III.4: Endereços de I/O utilizados pelo cartão de interface adicional. Todos os endereços são para escrita de dados.



III.1.B. TEMPO DE ACOMODAÇÃO E TRANSMISSÃO NOS QUADRUPOLOS ANALISADORES DE MASSA

Devido ao tempo de resposta do circuito de DC/RF e à capacitância do quadrupolo, não há uma correspondência instantânea entre o valor de m/z selecionado na entrada do circuito de controle e o que é efetivamente estabelecido pelo filtro quadrupolar. Segundo o fabricante, ^[2] este tempo de acomodação depende do "salto" de m/z e alguns destes valores estão plotados no gráfico da figura III.3. Este tempo de acomodação deve possuir um comportamento exponencial devido aos processos capacitivos de carga e descarga inerentes aos quadrupolos bem como do circuito de DC/RF. No entanto, para agilizar a determinação destes valores durante a aquisição de espectros, foi obtida empiricamente a equação III.1, que economiza tempo de processamento e se ajusta bem aos dados.

$$t_{\text{estab}} = \frac{13187,5 \Delta m}{16,04 + \Delta m}$$

(III.1)

onde: Δm é o salto de massa e t_{estab} é o tempo, em μs , requerido para estabilização do quadrupolo.

Esta equação nos fornece o tempo mínimo que deve ser aguardado antes de se fazer a leitura do ME. Se este tempo não for respeitado, na melhor situação, corre-se o risco de perder resolução e, na pior, os quadrupolos podem não estar selecionando a massa desejada e sim uma outra qualquer.

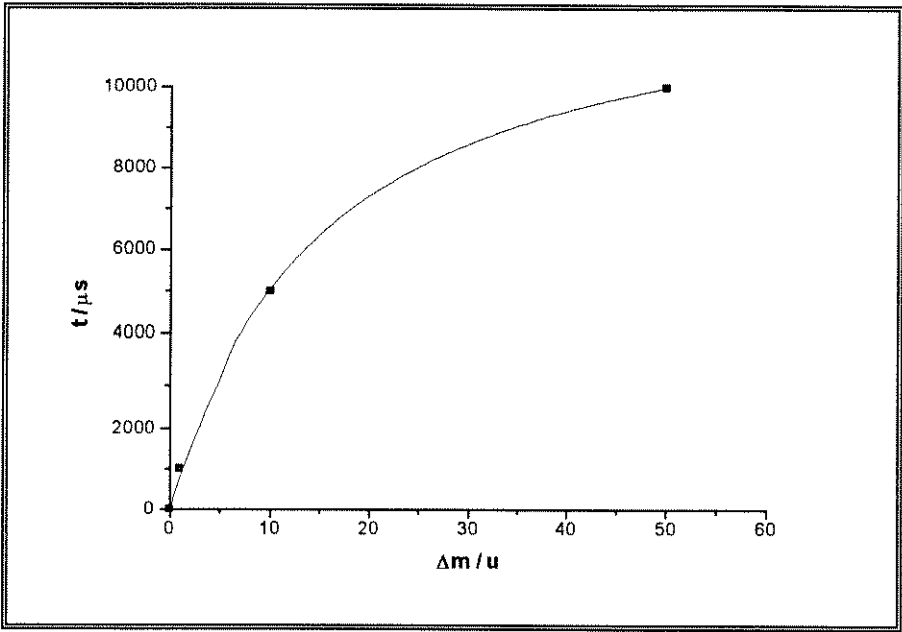


Figura III.3: Tempo de acomodação dos quadrupolos em função do “salto” de massa.

Outro parâmetro relevante é a transmissão dos íons que chegam ao detector durante uma análise. Quando se deseja varrer somente um ou dois quadrupolos, os demais devem permanecer no modo somente-RF. Nesta situação o quadrupolo possui baixíssima resolução, permitindo que íons de uma ampla faixa de m/z atravessasse por ele. No entanto, esta transmissão não é constante e depende da amplitude do sinal de RF aplicado. A figura III.4 mostra o comportamento da transmissão dos íons em função do valor de m/z selecionado no quadrupolo somente-RF (este valor está relacionado com a amplitude do sinal de RF). Observa-se que abaixo de 5 u a transmissão tende a zero. Acima de aproximadamente 9/7 do valor de m/z selecionado no quadrupolo de varredura, o mesmo acontece. O ponto de máximo de transmissão também é função do valor de m/z selecionado e alguns valores experimentais podem ser observados na

figura III.5. A equação III.2, obtida empiricamente, descreve este comportamento e a mesma é utilizada nos programas de aquisição de espectros.

$$m_{\text{off}} = \frac{91,91 m_{\text{on}}}{173,5 + m_{\text{on}}}$$

(III.2)

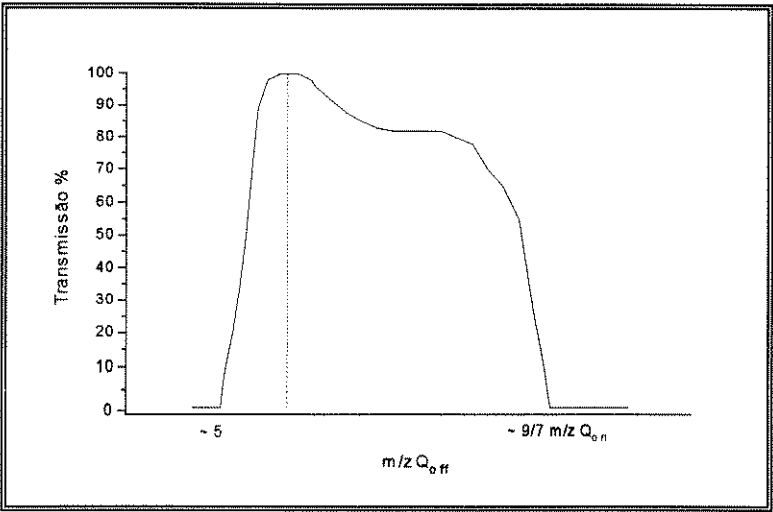


Figura III.4: Comportamento do filtro quadrupolar: Transmissão dos íons até o detector em função do valor de m/z ajustada nos quadrupolos no modo somente-RF.

m Qon	mQoff max
28	11
32	14
69	22
131	44
219	50

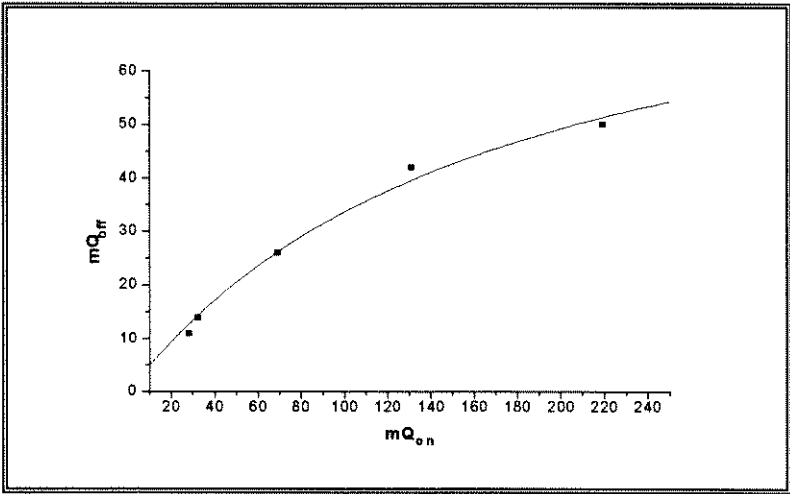


Figura III.5: Valores de m/z de transmissão máxima dos quadrupolos no modo somente-RF em função de m/z do quadrupolo no modo analisador de massa (RF/DC).



III.1.C. AJUSTE DO POTENCIAL DE IONIZAÇÃO

Durante a implementação do controle do potencial de ionização, notamos que, para um mesmo valor enviado ao conversor D/A, obtinhamos valores diferentes deste potencial em função do potencial do bloco da fonte de ionização. A tabela III.5 mostra os valores obtidos para o potencial de ionização para alguns valores enviados ao conversor D/A com o bloco da fonte de ionização ajustado para os valores de 42,0 V, 51,3 V e 70,7 V.

Valor enviado ao D/A	Tensão na saída do D/A (V)	EI (eV) p/ fonte a 42,0 V	EI (eV) p/ fonte a 51,3 V	EI (eV) p/ fonte a 70,7 V
-21500	-6,56	-	-	5,3
-20000	-6,11	-	-	9,8
-17000	-5,19	-	-	19,0
-15000	-4,58	-	5,7	25,1
-12000	-3,66	5,5	14,8	34,2
-8000	-2,44	17,7	27,0	46,3
-3000	-0,92	32,9	42,2	61,6
-100	-0,03	41,8	51,1	70,4
0	0,00	42,1	51,4	70,7
100	0,03	42,4	51,7	71,0
5000	1,52	57,3	66,6	85,9
10000	3,05	72,5	81,8	101,2
16000	4,88	90,8	101,1	-
17000	5,18	93,8	103,1	-

Tabela III.5: Valores obtidos para o potencial de ionização em função dos valores enviados para o conversor D/A e dos valores ajustados no bloco da fonte de ionização.

Observamos que há uma relação linear entre o potencial de ionização efetivo e o valor do DAC, conforme pode ser visto nas equações abaixo:

$EI = 3,05 \times 10^{-3} N_{DA} + 42,1;$

$EI = 3,04 \times 10^{-3} N_{DA} + 51,4;$

$EI = 3,05 \times 10^{-3} N_{DA} + 70,7.$

Estas equações mostram que existe um fator constante multiplicando o valor do DAC somado ao valor da tensão do bloco da fonte de ionização.

Em função desta observação, fomos obrigados a utilizar mais algumas linhas digitais (ver tabela III.2) e um outro canal do ADC do cartão AT-MIO-16X, bem como implementar o método Tionizador.Multiplex no software de aquisição (ver item V.2) para poder fazer a leitura do valor do potencial da fonte.

Com o valor do potencial do bloco da fonte (E_{fonte}) e o valor da energia de ionização desejado (EI), calcula-se o valor numérico que deve ser enviado ao conversor D/A (N_{DAC}) do cartão adicional, através da equação III.3, para gerar a tensão adequada.

$$EI = 3,0 \times 10^{-3} \cdot N_{\text{DAC}} + E_{\text{fonte}}$$

(III.3)

Entretanto, esta equação só é válida para íons positivos, que são os íons de maior interesse. Para íons negativos é necessário fazer uma nova série de medidas e determinar a nova relação.

O módulo de controle de Q1 é o mesmo que controla o detector e o filamento que emite elétrons. Assim, quer seja para controlar o valor de massa de Q1 quanto o potencial de ionização e a sensibilidade do eletrômetro, é necessário que este módulo esteja no modo controlado pelo computador.

III.2. DISCUSSÃO

A utilização do cartão AT-MIO-16X da National Instruments em associação ao cartão construído por nós proporcionou um controle bastante sofisticado do equipamento, suprimindo todas as necessidades dos grupos de pesquisa que utilizam o espectrômetro de massas pentaquadrupolar. Entretanto, ainda é possível controlar todas as funções do equipamento, bastando para isso a utilização de cartões de interface adicionais. Isto poderá ser feito sem prejudicar ou mesmo alterar o que já está em funcionamento.

A utilização dos contadores/temporizadores do cartão AT-MIO-16X no controle da estabilização dos quadrupolos proporcionou um controle mais eficiente e elegante. Cada contador faz parte de um objeto TQuad, assim cada quadrupolo pode ser ajustado independentemente do fato de um outro ter atingido a estabilidade. O que importa é que a leitura da abundância dos íons só será feita após a confirmação da estabilização dos três quadrupolos pelo método TQuad.JaEstavel (ver item V.2).

O controle do potencial de ionização foi uma implementação adicional que permitiu acrescentar mais este recurso ao sistema, dando a possibilidade de execução de estudos envolvendo o potencial de aparecimento dos íons.

Outro refinamento que permitiu melhorar o desempenho do sistema foi o ajuste do valor de m/z de transmissão máxima dos quadropolos que permanecem no modo somente-RF. Com isso conseguimos melhorar sensivelmente a transmissão dos íons e, conseqüentemente, melhorar a relação sinal/ruído.

III.3. CONCLUSÃO

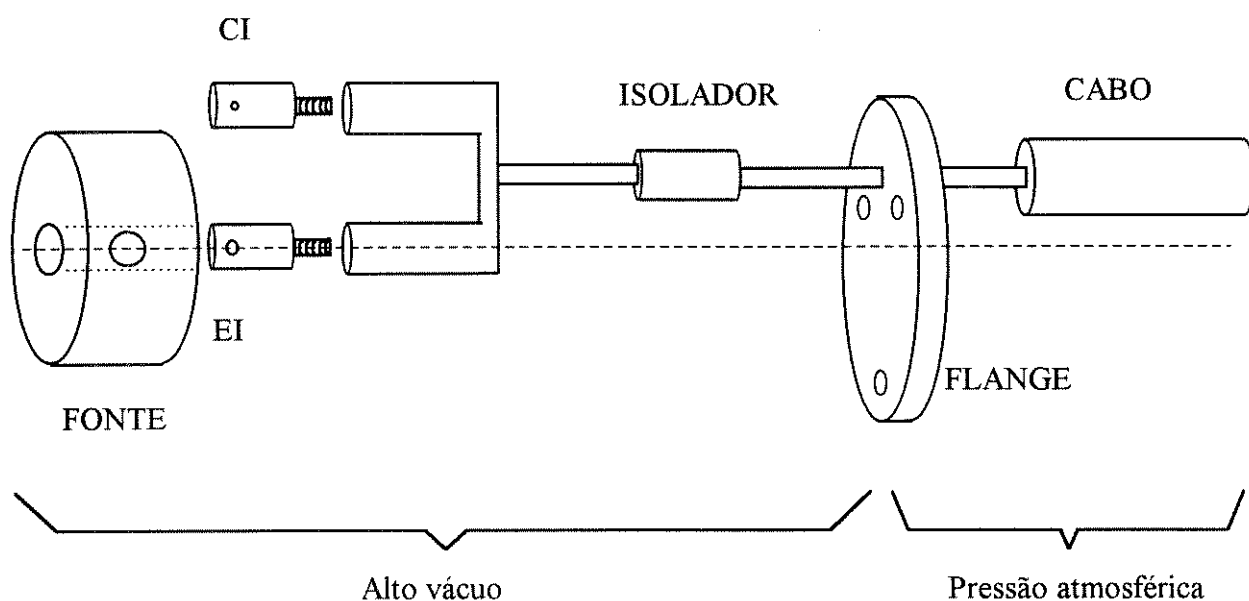
O trabalho de interfacemento foi bem sucedido, podendo-se acessar os principais controles do equipamento, bem como controlar os três quadropolos de varredura simultaneamente, o que não está disponível no equipamento similar instalado na Universidade de Purdue, Indiana, EUA.

III.4. BIBLIOGRAFIA

1. "Burr-Brown IC Data Book - Data Conversion Products", Burr-Brown Co.: Tucson, 1994, p. 3.67.
2. Extrel Mass Spectrometry, resultados não publicados.

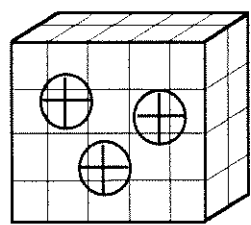
III.A. APÊNDICE: TROCADOR DO RECEPTÁCULO DE ÍONS

Em alguns estudos envolvendo reações íon-molécula é necessário efetuar a troca do receptáculo de íons durante o experimento para permitir a ionização, tanto por impacto de elétrons quanto por ionização química. Se o vácuo for “quebrado”, o equipamento precisa, no mínimo, por volta de quatro horas para estabelecer o nível adequado de pressão, de 8×10^{-7} a 1×10^{-6} torr, para experimentos MS^3 e por volta de duas horas para experimentos MS^2 ($\sim 5 \times 10^{-6}$ torr). Por esse motivo é fundamental efetuar esta troca sem interromper as atividades do equipamento. Isto nos levou a idealizar a peça mostrada abaixo.



Com esta peça efetua-se a troca do receptáculo de íons simplesmente puxando o cabo para trás, girando 180° e empurando de volta para a frente. Assim, em questão de segundos, efetua-se a troca sem “quebrar” o vácuo, não tendo o inconveniente de esperar algumas horas para a pressão atingir níveis aceitáveis para o experimento.

CAPÍTULO IV





IV. CIRCUITO OSCILADOR “APRISIONADOR” DE ÍONS

Este capítulo apresenta a parte do trabalho referente ao desenvolvimento de um circuito eletrônico para aumentar o tempo de residência dos íons no interior do quadrupolo de reação q2 objetivando aumentar o rendimento de reações íon-molécula.

IV.1. INTRODUÇÃO

IV.1.A. A TEORIA CINÉTICA DOS GASES E AS COLISÕES

São muitos os eventos que ocorrem quando uma molécula colide com outra. As reações químicas dependem destas colisões, assim como os fenômenos de transporte importantes em gases – pelos quais a energia (por condução de calor), a massa (por difusão) e o momentum (por viscosidade) são transferidos de um ponto a outro.^[1]

A seguir iremos fazer algumas suposições que nos permitam estimar o comportamento das moléculas no estado gasoso. Posteriormente, iremos admitir que os íons têm um comportamento semelhante. Informações mais detalhadas podem ser obtidas facilmente em livros textos de Físico-Química.^[1-3]

Supondo que existam dois tipos de moléculas, **A** e **B**, que interagem como esferas rígidas com diâmetros d_A e d_B , o contato entre tais moléculas ocorre sempre que a distância entre os centros se reduzir ao valor $d_{AB} = (d_A + d_B)/2$.

Suponhamos, agora, que todas as moléculas **B** estão em repouso e que a molécula **A** atravessa rapidamente o volume das moléculas estacionárias **B** com uma velocidade média $\bar{c}_A = \sqrt{8RT/\pi M}$. Podemos imaginar que a molécula **A** percorre em uma unidade de tempo um cilindro de volume $\pi d_{AB}^2 \bar{c}_A$. Se N_B/V é o número de moléculas de **B** por unidade de volume, existirão $\pi d_{AB}^2 \bar{c}_A N_B / V$ centros de moléculas **B** encontrados pela molécula **A** em seu percurso na unidade de tempo. Assim, podemos estimar a frequência de colisão para uma molécula **A** como:

$$z_{AB} = \frac{\pi d_{AB}^2 \bar{c}_A N_B}{V} \quad (IV.1)$$

Se existirem N_A/V moléculas de **A** por unidade de volume, a frequência total de colisões entre moléculas **A** e **B** será:

$$Z_{AB} = \frac{\pi d_{AB}^2 \bar{c}_A N_A N_B}{V^2} \quad (IV.2)$$

Esta estimativa despreza o efeito do volume das próprias moléculas, o que é uma aproximação razoável para baixas pressões. No entanto, a aproximação menos correta é considerar as moléculas **B** em repouso. Na verdade, naturalmente, é a velocidade de **A** em relação a **B** que determina a frequência de colisões. Esta velocidade relativa c_{AB} tem uma magnitude que depende do ângulo entre c_A e c_B ($c_{AB}^2 = c_A^2 + c_B^2 - 2c_A c_B \cos\theta$), conforme ilustra a figura IV.1. Porém, esta estimativa é mais simples e pode ser aplicada em nosso caso.

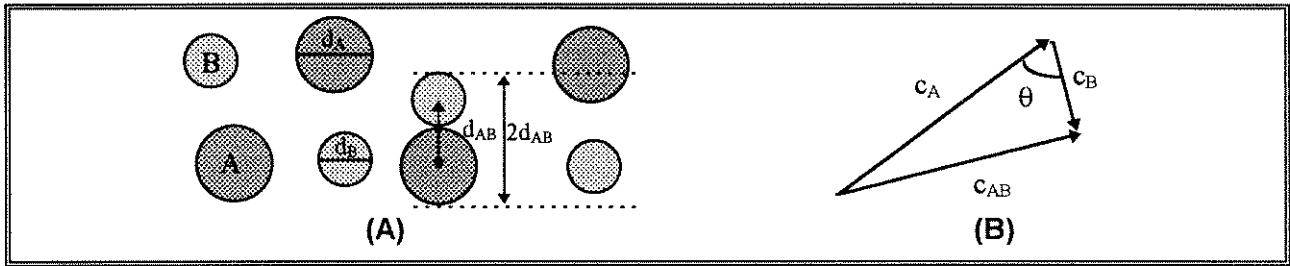


Figura IV.1: (A) Colisões moleculares e (B) velocidade relativa.

Mesmo que as moléculas não possam ser consideradas como esferas rígidas, podemos exprimir os resultados experimentais de vários processos de colisão em termos de uma secção transversal de choque efetiva σ_{AB} (para esferas rígidas $\sigma_{AB} = \pi d_{AB}^2$).

Uma quantidade importante na teoria cinética é a distância média percorrida por uma molécula entre duas colisões. Esta distância é chamada de caminho livre médio. O número médio de colisões experimentadas por uma molécula na unidade de tempo é z_{AB} . Neste tempo a molécula percorre uma distância $\bar{c}_{AB}$. O caminho livre médio λ é portanto $\bar{c}_{AB} / z_{AB}$, ou

$$\lambda = \frac{1}{\pi d_{AB}^2 (N_B / V)} = \frac{1}{\sigma_{AB} (N_B / V)} \quad (IV.3)$$

IV.1.B. O MOVIMENTO DOS ÍONS

Em um simples acelerador eletrostático, elétrons ou íons movem-se em uma direção paralela às linhas do campo elétrico E . Estes aceleradores têm duas aplicações importantes em espectrometria de massas.^[4]

A primeira delas consiste na produção de um feixe de elétrons de energia conhecida, o qual é necessário nas fontes de ionização por impacto de elétrons, onde íons são produzidos a partir de moléculas gasosas neutras pelo bombardeamento com estes elétrons (capítulo II.1).

A segunda, é acelerar os íons produzidos na fonte de ionização antes da entrada na região do analisador de massas. Uma partícula submetida à uma diferença de potencial elétrico (ddp), terá um ganho de energia cinética, E_c , igual ao produto de sua carga, ze , pela ddp, V :

$$E_c = zeV \quad (IV.4)$$

Sendo também a energia cinética (em J) de uma partícula igual a metade do produto de sua massa, m (em kg), com o quadrado de sua velocidade, v (em m/s), temos:

$$E_c = \frac{1}{2} m v_{ion}^2 \quad (IV.5)$$

Assim, a velocidade, v_{ion} , adquirida por este íon é dada por:

$$v_{ion} = \sqrt{\frac{2zeV}{m}} = 1,39 \cdot 10^4 \sqrt{\frac{zV}{M}} \quad (IV.6)$$

onde z denota o número de cargas do íon e M é sua massa atômica (em u.m.a).

Se este íon possuir uma velocidade inicial, v_0 , sua velocidade final será dada por:

$$v_{ion} = \sqrt{v_0^2 + \frac{2zeV}{m}} \quad (IV.7)$$

Uma vez que as lentes de focalização de um sistema quadrupolar estão submetidas a um potencial elétrico, a ddp entre as lentes de entrada e saída também é responsável pela aceleração ou desaceleração dos íons que entram na região do quadrupolo. Assim, podemos fazer uso das equações IV.6 e IV.7 para estimar as velocidades dos íons de diferentes massas dentro da região do quadrupolo.

Se a ddp entre as lentes for oscilante, o íon será acelerado e desacelerado na mesma frequência da oscilação. Porém, se a ddp for muito grande ou a frequência de oscilação for baixa,

o íon poderá ser expulso da região do quadrupolo, por um lado ou por outro, antes que haja a inversão do ciclo de oscilação.

No entanto, as equações IV.6 e IV.7 são, essencialmente, válidas na ausência de colisões. Como o interior do espectrômetro não é completamente despressurizado, temos que considerar as colisões com as moléculas gasosas e, assim, as velocidades finais dos íons são muito menores do que as calculadas através destas equações.

§ IV.2. IMPLEMENTAÇÃO

Na figura IV.2 é apresentado um diagrama parcial do espectrômetro mostrando os elementos de extração dos íons da fonte, as lentes de focalização e tres dos cinco quadrupolos. As lentes 1-2 e 2-3 são as lentes de focalização de entrada e saída de q2, respectivamente. A ddp entre estas lentes é responsável, também, pela aceleração ou desaceleração dos íons dentro de q2.

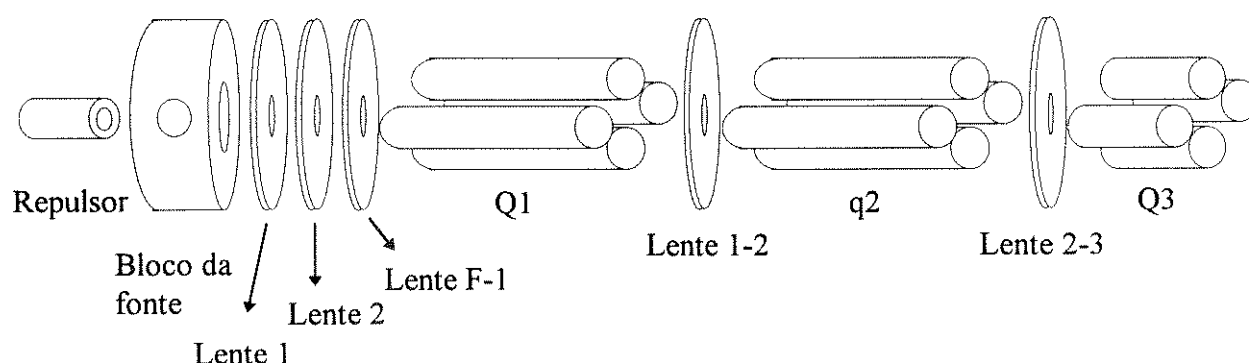


Figura IV.2: Diagrama esquemático parcial do equipamento pentaquadrupolar, mostrando os elementos extratores de íons da fonte (repulsor, lente 1 e lente2), as lentes de focalização (lentes F-1, 1-2 e 2-3) e os quadrupolos (Q1, q2 e Q3).

O circuito oscilador “aprisionador” de íons, cujo diagrama elétrico é mostrado na figura IV.3, foi acoplado ao controlador do potencial da lente de saída de q2 (Lente 2-3).

O sinal que aparece no ponto A do circuito é resultado da comparação das tensões nas entradas (pinos 2 e 3) do amplificador operacional TL084A. Assim, se a tensão na entrada não-inversora (+) é negativa, independentemente de seu valor, a tensão na saída (pino 1) é próxima do

valor da tensão de alimentação negativa (-15V). Se a tensão se inverter, a saída também se inverte, assumindo um valor próximo da tensão de alimentação positiva (+15V).

No ponto **B** do circuito temos a saída de um integrador, onde a tensão de pico, V_p , é limitada por R_1 e R_2 através da realimentação do circuito:

$$V_p = -\frac{R_2}{R_1} \cdot V_Q$$

onde V_Q é a tensão na saída do comparador ($\pm 14V$). Uma vez que este amplificador operacional (TL084B) está configurado como integrador, a qualquer instante a tensão, V_t , é dada por:

$$V_t = -\frac{1}{(R_3 + R_4) \cdot C} \int V_Q dt = -\frac{t}{(R_3 + R_4) \cdot C} \cdot V_Q$$

Assim, no instante t , no qual $V_t = V_p$, temos:

$$t = \frac{4 \cdot (R_3 + R_4) \cdot C \cdot R_2}{R_1}$$

Como os dois operacionais estão ligados entre si através da realimentação do circuito, V_Q ora é positivo, ora é negativo. Assim, no ponto **A** temos uma onda quadrada e no ponto **B** uma onda triangular, com frequência de oscilação dada por:

$$f = \frac{R_1}{4 \cdot (R_3 + R_4) \cdot C \cdot R_2}$$

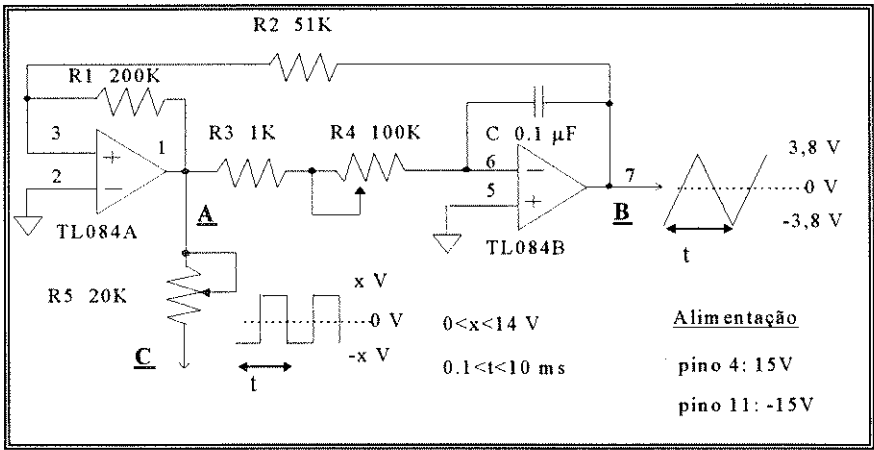


Figura IV.3: Circuito oscilador “aprisionador” de íons acoplado ao controlador de potencial da lente de saída do primeiro quadrupolo de reação (q2).

Como R_4 é um potenciômetro multi-voltas, através da alteração do seu valor de resistência, podemos ajustar o período de oscilação em um intervalo de 0,1 a 10 ms. Com isso a frequência nominal pode ser alterada de 100 Hz a 10kHz, porém de um modo não linear.

Na tabela IV.1 são apresentados os valores de frequência, medidos sobre a lente de saída de q2, obtidos pelo ajuste de R_4 .

AJUSTE NO POTENCIÔMETRO / voltas	FREQUÊNCIA / Hz
0,00	100
5,00	200
6,70	300
7,55	400
8,10	500
8,40	600
8,66	700
8,85	800
9,00	900
9,10	1000
9,20	1100
9,60	2000
9,78	3000
9,86	4000
9,92	5000
9,95	6000
9,97	7000
9,99	8000
10,00	9000

Tabela IV.1: Valores de frequência obtidos a partir do ajuste do potenciômetro R_4 . Os valores foram determinados com um osciloscópio sobre a lente de saída 2-3 de q2.

Através de medições sobre a lente 2-3, constatamos que o circuito controlador do potencial da lente não responde adequadamente ao sinal quadrado de excitação. A figura IV.4 mostra a deformação da forma de onda em função da frequência. Abaixo de 5 kHz a onda ainda mantém uma forma quadrada. Ao redor de 5 kHz já existe uma considerável deformação, diminuindo, inclusive, o valor de pico do sinal por volta de 20%. Acima de 5 kHz o sinal já tem um formato quase triangular com uma intensidade 33% menor, porém não se altera muito mais até atingir o limite de 9 kHz.

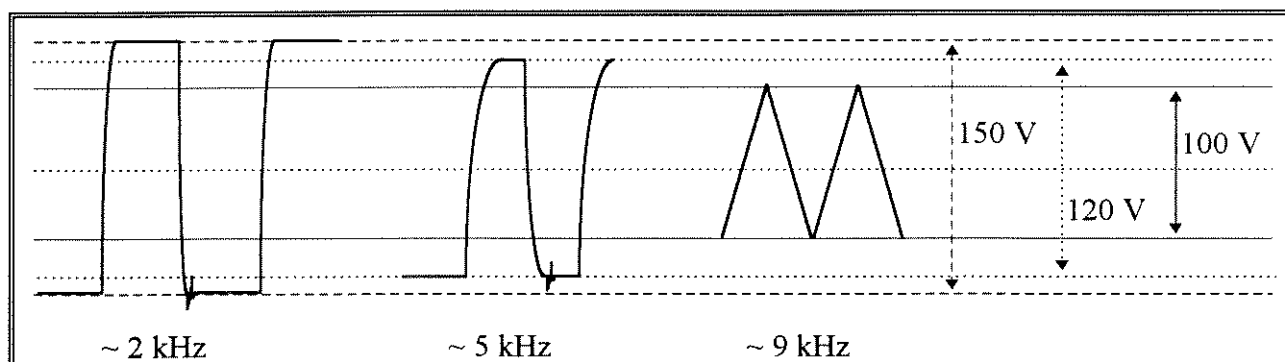


Figura IV.4: Formas da onda sobre a lente 2-3 em função do aumento da frequência de oscilação. A forma de onda na saída do circuito oscilador permanece inalterada à qualquer frequência.

Para podermos também ter controle sobre a intensidade do sinal, foi introduzido o potenciômetro multi-voltas R_5 . Com ele podemos alterar a amplitude do pulso quadrado no ponto **C** entre 0 e ± 14 V. O valor da intensidade do sinal é linear com o ajuste sobre o potenciômetro. Assim, quando o mostrador do potenciômetro estiver ajustado em 0,00, tem-se 0 V e quando estiver ajustado em 10,00 tem-se 28 V pico-a-pico (de -14 V a +14 V), ou seja, para cada volta de R_5 tem-se um aumento de 2000Ω o que equivale a um aumento $\pm 1,4$ V por volta ($2000 / 20000 \times 14 = 1,4$ V). Para cada 1 V do oscilador tem-se aproximadamente 6,7 V sobre a lente. Assim tem-se no máximo a uma oscilação de ± 93 V (nominal).

✍ IV.3. RESULTADOS

Conforme podemos ver na figura IV.3, o circuito possui dois pontos **B** e **C** de onde podemos obter as formas de onda triangular e quadrada, respectivamente. Na figura IV.5 são mostrados três espectros da reação íon-molécula entre o íon distônico $\text{CH}_2\text{OCH}_2]^+$ com a molécula de ciclopentanona. Os espectros foram obtidos sem o circuito oscilador e com as formas de onda triangular e quadrada. Notamos claramente que os sinais dos produtos aumentaram de intensidade, sendo de 107%, 41% e 32% o aumento com a onda triangular e de 80%, 41% e 81% com a onda quadrada, respectivamente para os sinais em m/z 86, 99 e 169. Podemos observar também um rendimento médio maior com a forma de onda quadrada do que com a triangular.

Entretanto, isto pode ser devido também à amplitude das ondas que são diferentes. A onda quadrada, neste caso, tem uma tensão pico-a-pico de 28 V contra 7,6 V da onda triangular.

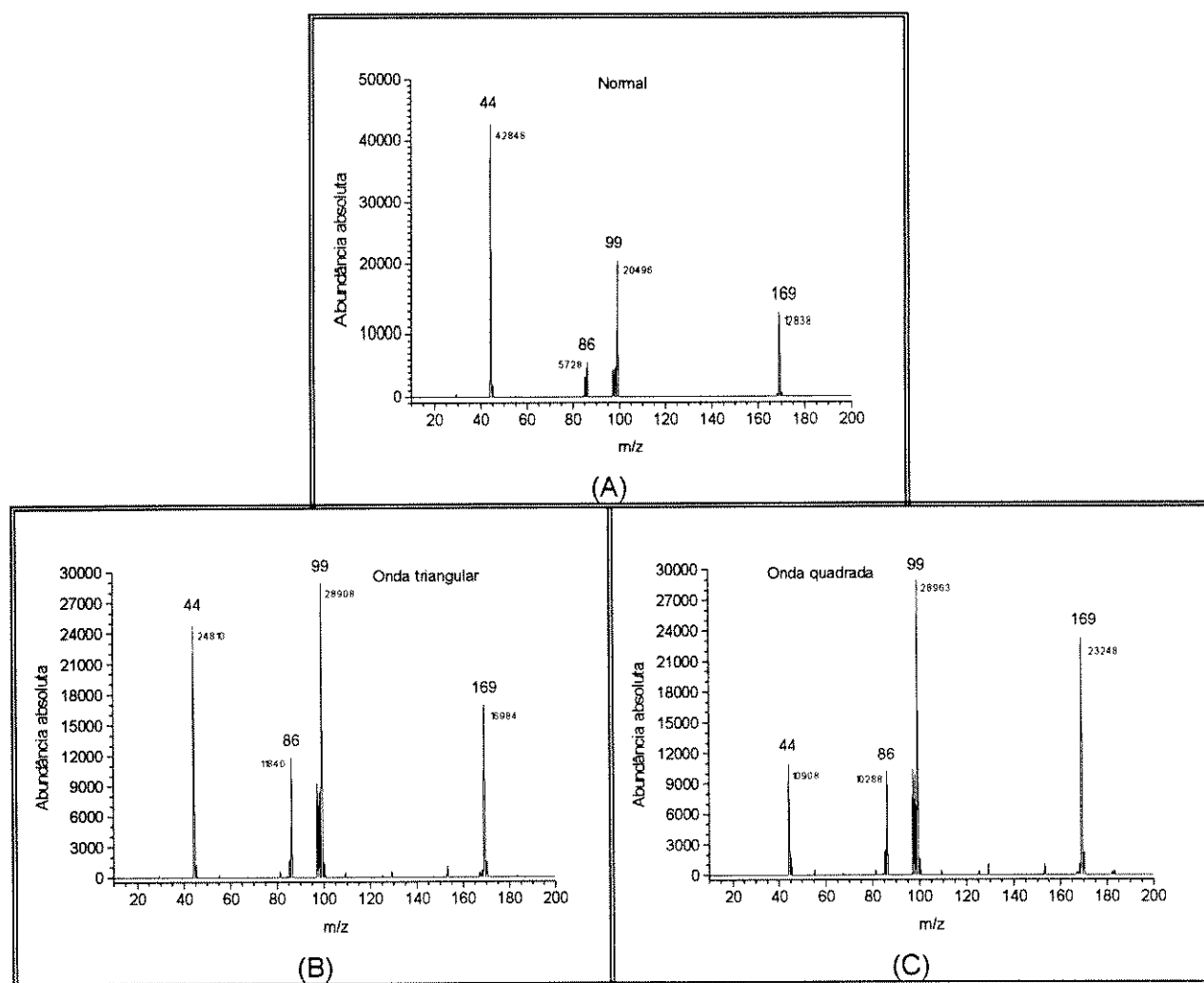


Figura IV.5: Espectros de massa da reação íon-molécula entre o íon distônico $\text{CH}_2\text{OCH}_2]^+\bullet$ (m/z 44) com a molécula de ciclopentanona (massa molar 84 g) obtidos sem o circuito oscilador (A) e com as formas de onda triangular (B) e quadrada (C), ambas com frequência de 2 kHz e amplitude de 3,8 V e 14 V, respectivamente. Os produtos da reação são vistos através dos sinais em m/z 86, 99 e 169.

Para verificar o efeito amplitude foi introduzido o potenciômetro R_5 , mostrado no esquema da figura IV.3. A figura IV.6 mostra os resultados obtidos pela variação amplitude da onda quadrada, mantendo-se a frequência fixa em 2 kHz e utilizando-se a reação entre o íon CH_3CO^+ (m/z 43) proveniente da 2-butanona com a molécula de 1,3-dioxolano (massa molar 74 g), que é uma reação com menos produtos que a anterior.

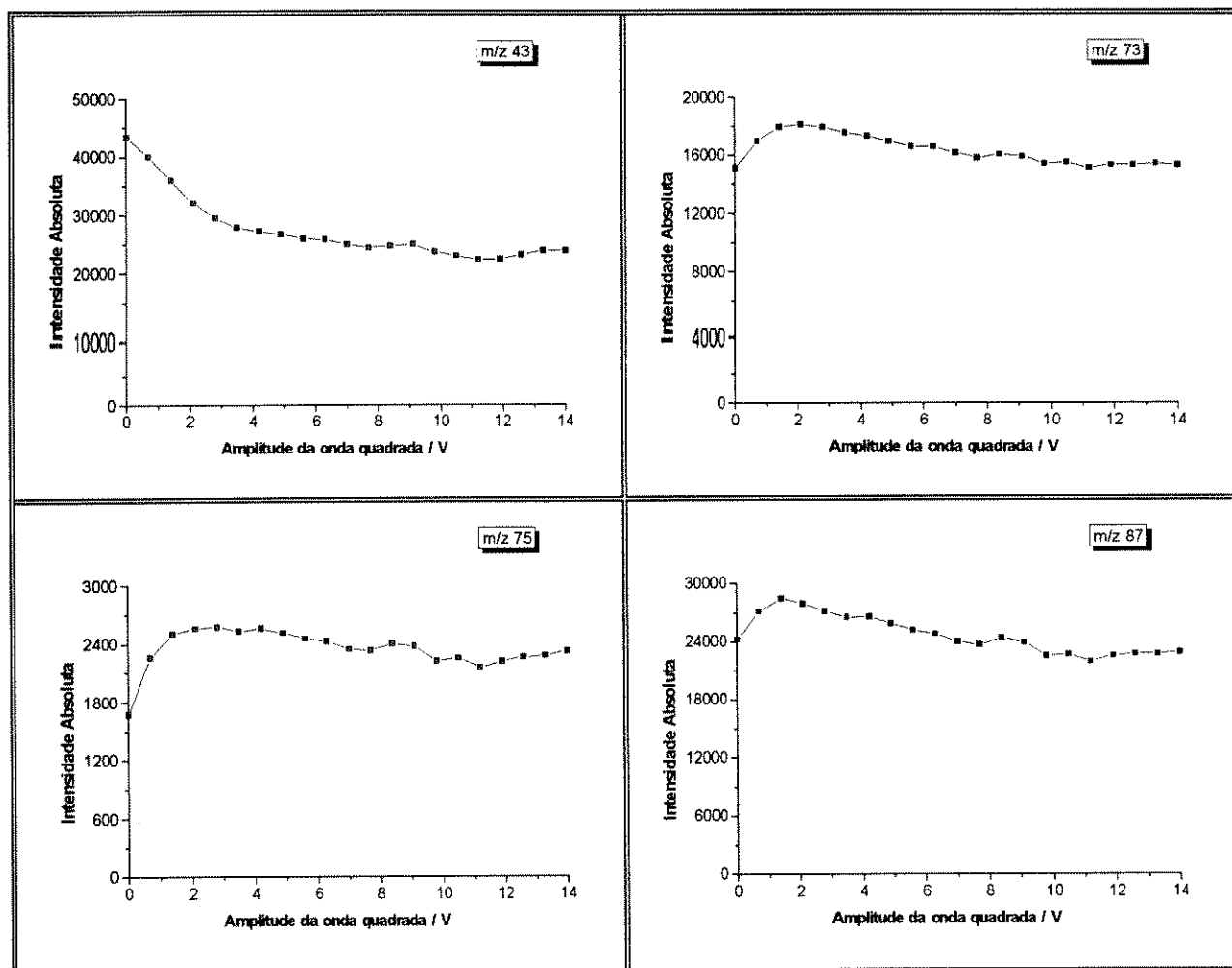


Figura IV.6: Efeito da amplitude sobre a intensidade dos produtos e reagente da reação entre o íon de m/z 43 da 2-butanona com a molécula de 1,3-dioxolano. Os picos em m/z 73, 75 e 87 são produtos da reação, correspondendo a transferência de hidreto, protonação e adição de oxirano, respectivamente. Para aquisição de dados foi utilizado o programa SIMTIM e a frequência foi mantida constante em 2 kHz.

Observando esta figura, notamos que todos os sinais dos produtos aumentam de intensidade até por volta de 2 V, sofrendo um declínio da mesma ordem do aumento após este potencial. O sinal do reagente (m/z 43), entretanto, tem um declínio bem significativo até 2 V, permanecendo constante até o limite de 14 V. Este comportamento é bastante coerente uma vez que os aumentos nas intensidades dos produtos ocorrem exatamente na mesma faixa onde ocorre redução do sinal do reagente.

Extendendo o estudo em relação à frequência, a figura IV.7 mostra os resultados obtidos em função da alteração da frequência de oscilação com a mesma reação da figura IV.6. A amplitude foi mantida constante em 14V. Constatamos que os sinais dos picos dos produtos da reação têm um aumento de intensidade até 1 kHz, tendo um ligeiro declínio após esta frequência e mantendo-se constante até o limite de 9 kHz. No entanto, acreditamos que parte desse declínio seja devido à queda no valor da amplitude do sinal que é efetivamente aplicado nas lentes, conforme discutido anteriormente (figura IV.4).

A figura IV.8 mostra outros dois espectros da mesma reação da figura IV.5 em condições diferentes, obtidos sem (A) e com (B) o circuito oscilador ativo. Através da intensidade do sinal em m/z 86 que é um dos produtos da reação, podemos observar um aumento de 3,6 vezes (260%) no rendimento da reação. Podemos observar também, através da comparação entre os espectros das figuras IV.5C e IV.8B, que o aumento da permanência do íon no quadrupolo pode alterar a competição de três processos reacionais: transferência de carga, transferência de próton e cicloadição, que correspondem aos sinais em m/z 84, 85 e 86, respectivamente.

Vamos considerar a reação íon-molécula mostrada na figura IV.9, a mesma utilizada para o estudo do efeito da frequência e amplitude mostrada nas figuras IV.6 e IV.7. Analisando os sinais em m/z 73, 75 e 87, correspondentes aos produtos da reação, notamos um aumento de 89%, 34% e 92% em relação aos sinais da reação sem a ação do circuito oscilador. Estes dados são mais um exemplo para constatar que, realmente, existe um aumento dos sinais dos produtos quando o oscilador é ativado. No entanto, estes valores podem ainda ser maiores se utilizarmos os pontos máximos observados nas figuras IV.6 e IV.7. Porém, este procedimento não é fácil de ser empregado. Assim, se após alguns ajustes é obtido um bom sinal, com intensidade suficiente para prosseguir com o experimento, não é necessário buscar o ponto ótimo.

Foi notado, em alguns casos, que o circuito oscilador também tem efeito sobre a técnica de dissociação ativada por colisão (CID). Isto é bem natural, pois tanto na dissociação quanto em uma reação íon-molécula, o íon necessita colidir com as moléculas, que neste caso é feita com as moléculas de um gás inerte.

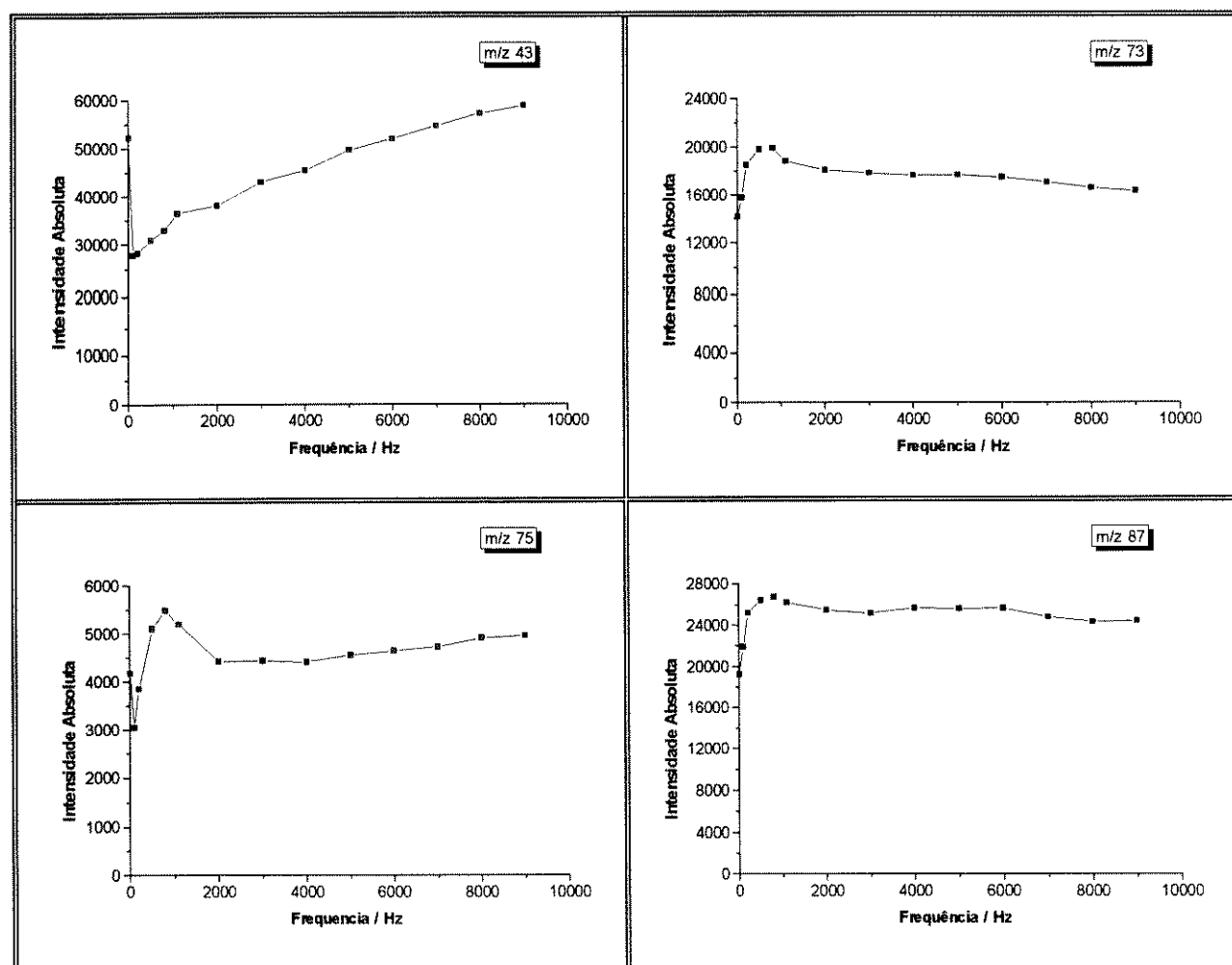


Figura IV.7: Efeito da frequência de oscilação sobre a intensidade dos produtos e reagente da reação entre o íon de m/z 43 da 2-butanona com a molécula de 1,3-dioxolano. Os picos em m/z 73, 75 e 87 são produtos da reação, correspondendo a transferência de hidreto, protonação e adição de oxirano, respectivamente. Para aquisição de dados foi utilizado o programa SIMTIM e a amplitude foi mantida constante em 14V.

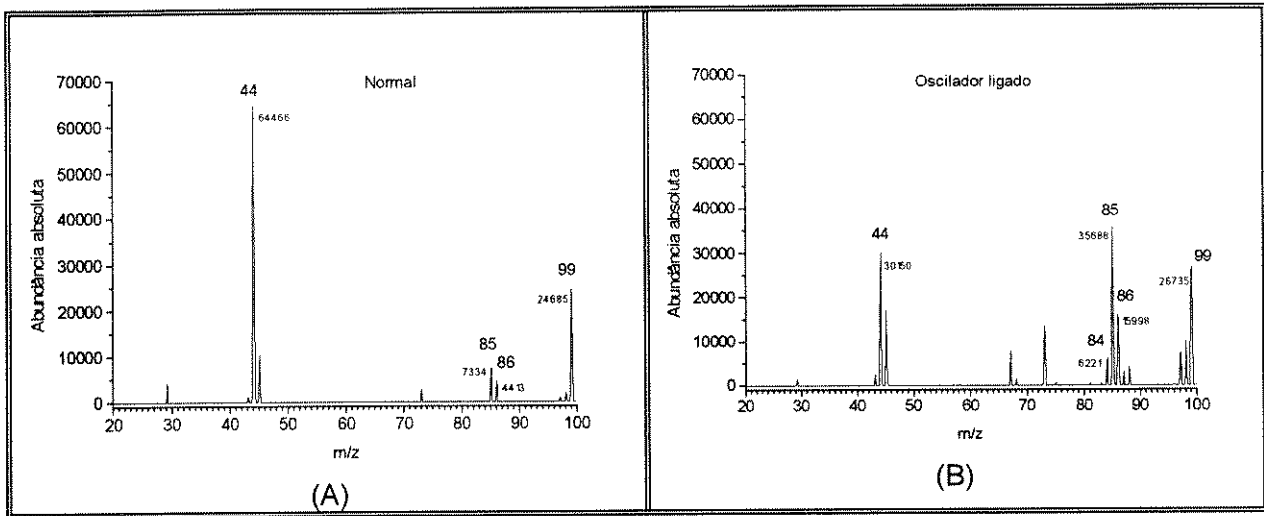


Figura IV.8: Espectros de massa da mesma reação íon-molécula da figura IV.5: (A) Com o circuito oscilador desligado e (B) com o circuito ligado ajustado para a frequência de 2 kHz e amplitude de 14 V.

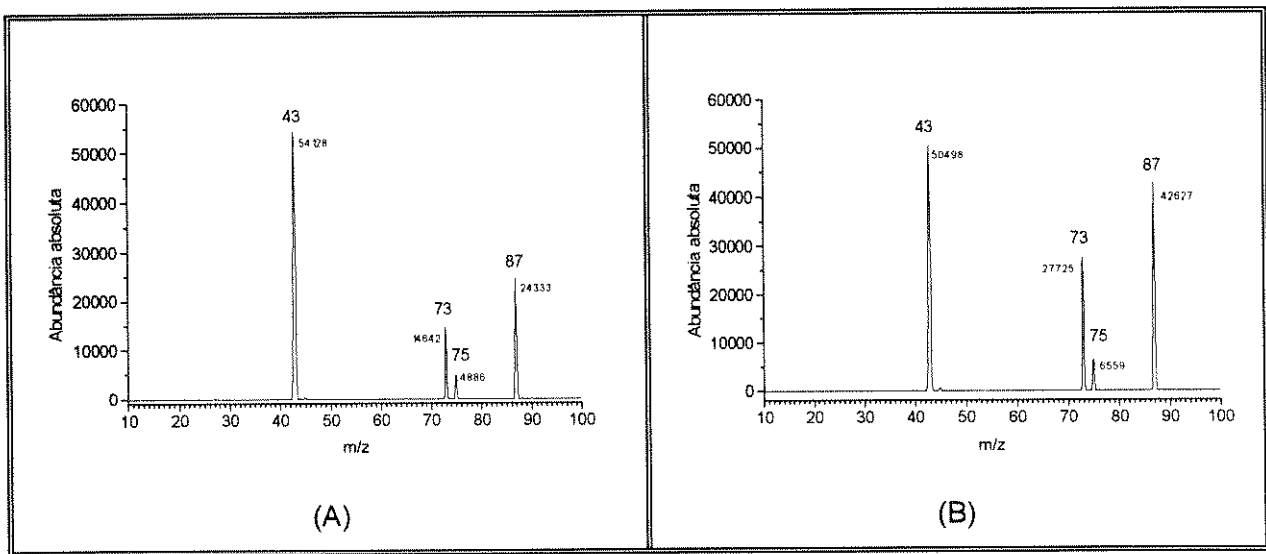


Figura IV.9: Espectros da reação íon-molécula entre o íon acílio CH_3CO^+ (m/z 43) proveniente da 2-butanona com a molécula de 1,3-dioxolano (massa molar 74 g): (A) Com o circuito oscilador desligado e (B) com o circuito ligado e ajustado para uma frequência de 2 kHz e amplitude de 14 V.

IV.4. DISCUSSÃO

O circuito oscilador implementado é muito simples, não tendo nada de especial. Contudo, a sua ação sobre a lente de focalização de saída de q2, trouxe resultados bastante interessantes e satisfatórios como mostrado anteriormente.

Para aumentar o tempo de permanência de um íon no interior do quadrupolo de reação a fim de aumentar o rendimento de reações íon-molécula, teríamos duas possibilidades: aumentar o tamanho do quadrupolo ou diminuir a velocidade do íon. A primeira alternativa é inviável por razões práticas. A segunda é inviável por razões físico-químicas relacionadas com a diminuição da energia necessária para ocorrer a reação.

Assim, a idéia básica de se utilizar o circuito oscilador é aumentar o tempo de permanência dos íons dentro do quadrupolo de reação sem reduzir a energia cinética dos mesmos, fazendo com que eles tenham uma trajetória oscilante.

No entanto, considerando um íon de m/z 44 (utilizado em um dos experimentos descritos) submetido a uma ddp oscilante de 93 V de amplitude entre as lentes de entrada e saída de q2 com frequência de 9 kHz, sua trajetória oscilante teria uma amplitude da ordem de 6 m.

O que explica, então, o fato de conseguirmos aprisionar este íon dentro de um quadrupolo com apenas 20 cm de comprimento? As múltiplas colisões dentro do quadrupolo são responsáveis por alterar a velocidade destes íons, resultando em uma amplitude de deslocamento bem menor.

Considerando o quadrupolo de reação contendo benzeno, cuja secção transversal de choque efetiva é $8,8 \times 10^{-19} \text{ m}^2$,^[2] a uma pressão da ordem de 1 torr e uma temperatura de 25 °C, a velocidade quadrática média é de 284 m/s e, assim, temos que a frequência de colisão de uma molécula é da ordem de $1,2 \times 10^7 \text{ Hz}$, ou seja, aproximadamente, uma colisão a cada 0,1 μs . Um íon com uma velocidade da ordem de $1,0 \times 10^4 \text{ m/s}$ (m/z 44 acelerado da fonte até q2 com 26 V) entrando nesta câmara reacional, caminharia aproximadamente 1 mm antes de sofrer a primeira colisão. A partir deste ponto, existem dois processos que irão determinar a velocidade e, portanto, a amplitude da trajetória oscilante: (1) campo elétrico contribuindo para o aumento da energia cinética e (2) as colisões atuando em sentido contrário.

Mesmo considerando a situação hipotética, na qual este íon nunca sofra uma colisão frontal e que sua energia cinética se mantenha constante, este colidiria aproximadamente 200 vezes antes de sair da câmara reacional. Considerando que, devido ao campo elétrico de RF, a sua

trajetória seja helicoidal com raio de 1 cm, a distância percorrida passaria de 20 para 130 cm. Logo, o número de colisões aumentaria para 1300.

Um exemplo interessante que evidencia a existência de colisões múltiplas pode ser visto no apêndice IV.A.1.

Através deste raciocínio fica claro o fato de conseguirmos “aprisionar” o íon dentro de q_2 através da oscilação do potencial da lente de saída com uma frequência da ordem de alguns kilohertz.

No apêndice IV.A.2 fazemos um outro tipo de abordagem para verificar o comportamento dos íons sem considerar a existência de colisões, bem como apresentamos uma tabela mostrando a redução da velocidade em função de uma única colisão.

Com relação ao estudo efetuado com a variação da amplitude do sinal de oscilação chegamos a seguinte conclusão: Como esperado, a medida em que a amplitude do potencial oscilante da lente de saída aumenta, a intensidade do íon reagente m/z 43 decai. Isto é devido ao aumento de probabilidade de colisão íon-molécula à medida que a amplitude da trajetória oscilante do íon reagente cresce. No entanto, a quantidade dos produtos m/z 73, 75 e 87 cresce até aproximadamente 2 V, decaindo a partir deste ponto. Isto pode ser devido à formação de produtos de m/z mais elevados, originados pela reação destes íons com a molécula neutra, produzindo por exemplo um dímero de próton, de maneira semelhante a que ocorre com a reação da figura IV.5. Estas reações são favorecidas pelo aumento da energia cinética destes íons, o que é conseguido pela transferência no momento da colisão com o íon reagente m/z 43, assim como pela atuação direta de um campo elétrico mais elevado.

Com relação à frequência, observamos que há um aumento na intensidade do sinal do íon reagente em m/z 43 em função de sua elevação. Isto provavelmente se deve a dois fatores: 1) a diminuição da amplitude do sinal efetivamente aplicado na lente de q_2 , conforme mostrado anteriormente e 2) diminuição da amplitude da trajetória oscilante do íon reagente.

Em função da atuação do circuito oscilador, notamos uma diferença no formato do sinal correspondente a um pico como mostrado na figura IV.10. Podemos notar que ora há sinal no detector, ora não, na mesma frequência do oscilador. Assim, em uma metade do ciclo de oscilação como o potencial da lente 2-3 é desfavorável, os íons não chegam ao detector. Na outra metade do ciclo de oscilação a situação se inverte.

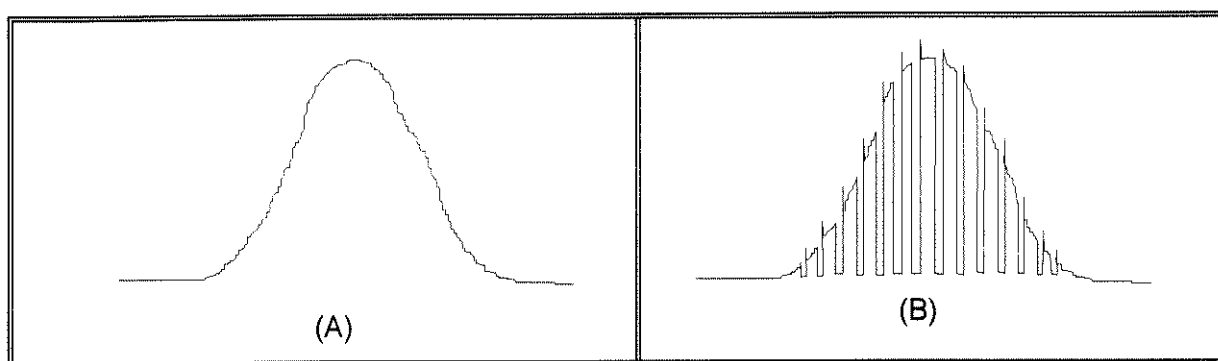


Figura IV.10: Formato do sinal correspondente a um pico que chega ao detector: **(A)** sem o oscilador e **(B)** com o oscilador ligado.

Entretanto, o formato do sinal mostrado na figura IV.10b não interfere na sua detecção. No procedimento de aquisição de dados o sinal deve ser integrado por, no mínimo, 16,7 ms (ver item V.2). O tempo de integração geralmente utilizado nas aquisições de dados de reações íon-molécula tem sido 33,4 ms. Assim, apesar do sinal passar por máximos e mínimos, o valor médio tem-se mostrado eficaz para representar a intensidade do pico, mesmo tendo um valor médio menor, já que todos os picos sofrem o mesmo efeito. Contudo, se for desejado uma maior sensibilidade, pode ser adicionado ao sistema de detecção um discriminador de sinal (*lock-in*), o qual permite que o sinal seja lido somente nos instantes favoráveis do ciclo de oscilação, separando o máximo do mínimo e, portanto, permitindo que somente o sinal de interesse seja integrado.

IV.5. CONCLUSÃO

O circuito apresentado, embora tenha uma concepção extremamente simples, apresentou resultados bastante satisfatórios na aquisição de dados de reações íon-molécula com rendimentos muito baixos. Nos dois exemplos apresentados para a reação do íon distônico $\text{CH}_2\text{OCH}_2^+\text{•}$ com a molécula de ciclopentanona, os rendimentos da reação foram aumentados da ordem de 80% e 260% para um dos produtos (m/z 86). Do estudo da influência da amplitude e frequência de oscilação, concluímos que não existem, até o momento, valores ótimos que possam ser aplicados a

qualquer reação. Estes devem ser estabelecidos empiricamente em função das massas das espécies envolvidas, assim como do tipo de reação a ser favorecida. Assim, o aumento do rendimento das reações pode ser maior ou menor dependendo não somente da reatividade dos íons e moléculas reagentes, mas também das pressões parciais dos reagentes, da temperatura e dos ajustes dos elementos da fonte de ionização e das lentes de focalização dos quadrapolos. Destes fatores, obviamente, somente a reatividade dos íons e moléculas são independentes de ajustes feitos pelo indivíduo que está realizando o experimento.

IV.6. BIBLIOGRAFIA

1. Moore, W. J. *Físico Química*; Editora Edgard Blücher Ltda: São Paulo, 1976, vol. 1, p. 106.
2. Atkins, P. W. *Physical Chemistry*; Oxford University Press: Oxford, 1994, 5th ed, p. 33-40.
3. Castellan, G. W. *Físico Química*; Livros Técnicos e Científicos: Rio de Janeiro, 1973, vol. 2, p. 764
4. Roboz, J. *Introduction to Mass Spectrometry Instrumentation and Techniques*; Interscience Publishers: New York, 1968, p. 27-29.

⊕ IV.A.1. APÊNDICE: EXPERIMENTO PARA DEMONSTRAR A OCORRÊNCIA DE COLISÕES MÚLTIPLAS

Para demonstrar a existência de colisões múltiplas dentro do quadrupolo de reação, podemos utilizar a reação íon-molécula entre o íon de m/z 70, proveniente da fragmentação da *n*-bromo-succinimida por impacto de elétrons, com a molécula neutra de isopreno (massa molar 68 g), cujo espectro é mostrado na figura IV.11.

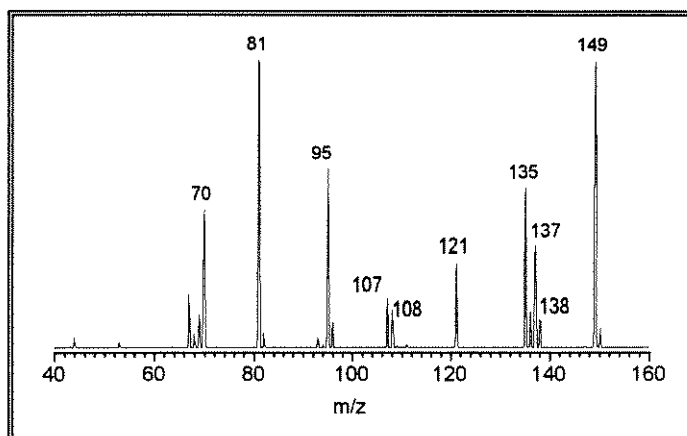


Figura IV.11: Espectro da reação íon-molécula entre o íon de m/z 70, proveniente da fragmentação da *n*-bromo-succinimida com a molécula neutra de isopreno.

O pico em m/z 138 corresponde ao produto de cicloadição do íon de m/z 70 com a molécula de isopreno. O pico em m/z 69 corresponde ao isopreno protonado. Estes dois sinais são originados de uma única colisão.

O pico em m/z 137 corresponde à reação do isopreno protonado com o isopreno neutro, em um processo envolvendo 2 colisões.

Em um processo mais complexo envolvendo 4 colisões, o íon de m/z 81, correspondendo a um fragmento de um produto de reação íon-molécula do íon de m/z 69, reage com isopreno neutro para dar origem ao produto de m/z 149.

Outros produtos de reações envolvendo colisões múltiplas também estão presentes no espectro (m/z 95 e m/z 121) e outros de maiores massas não mostrados.

⌚ IV.A.2. APÊNDICE: SIMULAÇÕES DO COMPORTAMENTO DE UM ÍON DENTRO DE q2 SEM O EFEITO DAS COLISÕES MÚLTIPLAS

Considerando a forma de onda do oscilador, como mostrado na figura IV.12, vamos analisar o que acontece com um íon positivo (m/z 44) ao entrar em q2 nos instantes de tempo 0, $T/4$, $T/2$, $3T/4$ e T , sem considerar a diminuição de sua velocidade devida às colisões com as moléculas neutras. Os dados das simulações, mostradas a seguir, foram feitas com o programa listado no final deste apêndice.

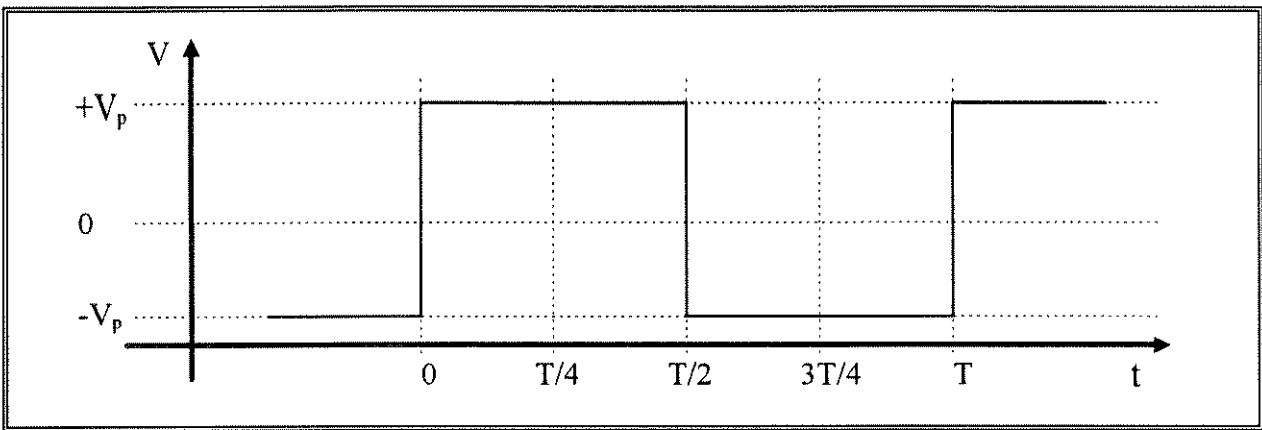


Figura IV.12: Forma de onda do oscilador aplicada ao controlador do potencial da lente de focalização de saída de q2 (Lente 2-3).

Analisando a figura IV.13 podemos ver que a energia que o íon possui, em função da ddp que o acelera, permite que ele caminhe mais ou menos dentro do quadrupolo. Se o ciclo do oscilador é desfavorável ($0 \leq t < T/2$), os íons de maior energia conseguem caminhar mais e permanecer mais tempo dentro do quadrupolo que os íons de menor energia. Se o ciclo é favorável ($T/2 \leq t < T$), os íons caminham a mesma quantidade (o tamanho do quadrupolo), porém os íons de menor energia permanecem mais tempo dentro do quadrupolo que os de maior energia. Não só o quanto o íon caminha, mas também o tempo em que permanece dentro do

quadrupolo são importantes para o aumento da probabilidade das colisões com as moléculas para promover a reação.

Assim, como pode ser observado, se não houvesse as colisões múltiplas dentro do quadrupolo, os íons não sofreriam efeito significativo do circuito oscilador incorporado ao sistema por possuírem uma velocidade muito elevada.

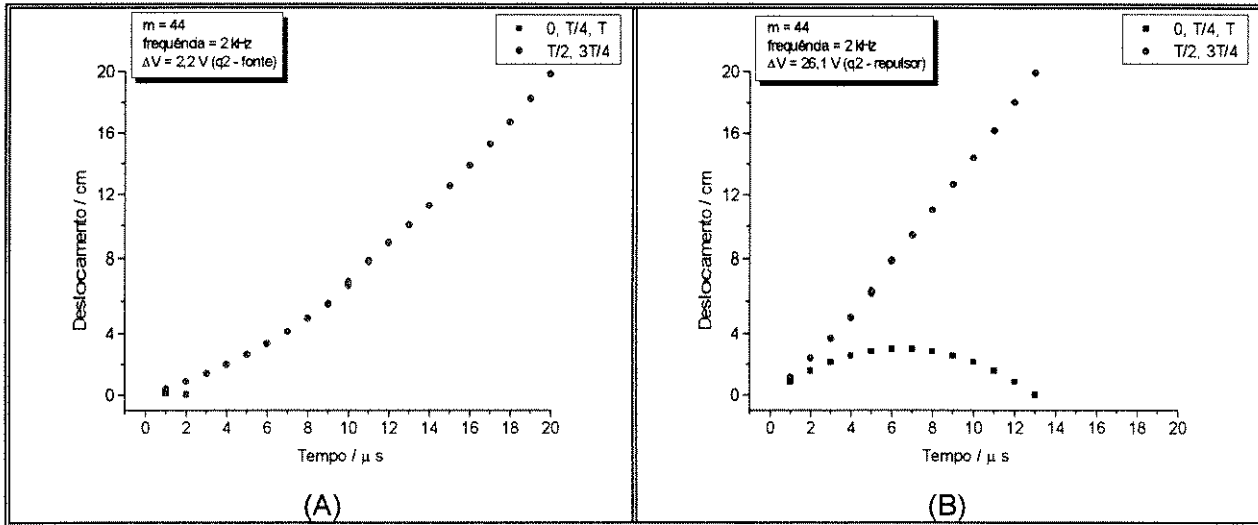


Figura IV.13: Simulações do deslocamento dos íons (m/z 44) contado a partir do instante que entra em q_2 . Os íons que entram quando o oscilador está aplicando um potencial positivo (■) são repelidos em direção à entrada, enquanto que os íons que entram no ciclo negativo (●) são atraídos em direção à saída. As simulações foram feitas para duas energias cinéticas diferentes 2,2 eV (A) e 26,1 eV (B).

A figura IV.14 mostra o comportamento destes íons (m/z 44) sem a interferência do circuito oscilador. Conforme pode ser visto, o tempo de permanência dos íons dentro do quadrupolo é ordem de 50 μs e 20 μs para os íons com energia cinética igual a 2 eV e 26 eV, respectivamente. Logo, a frequência mínima para que os íons sejam submetidos à um ciclo completo do oscilador é 20 kHz e 50 kHz, na mesma ordem. Isto acaba sendo mais uma evidência de que a velocidade dos íons não são aquelas calculadas pelas equações IV.6 e IV.7 devido à

condição de colisões múltiplas. A tabela IV.2 mostra, como exemplo, a velocidade final de um íon de m/z 44 u, acelerado da fonte até q2 com 2,2 V e 26,1 V, após uma única colisão com uma molécula neutra de massa molar 84 g. Para o cálculo da velocidade final assumimos que o choque é feito com uma molécula em repouso e utilizamos as equações derivadas de uma colisão perfeitamente elástica, na qual tem-se a conservação da energia cinética e a conservação do momento linear.

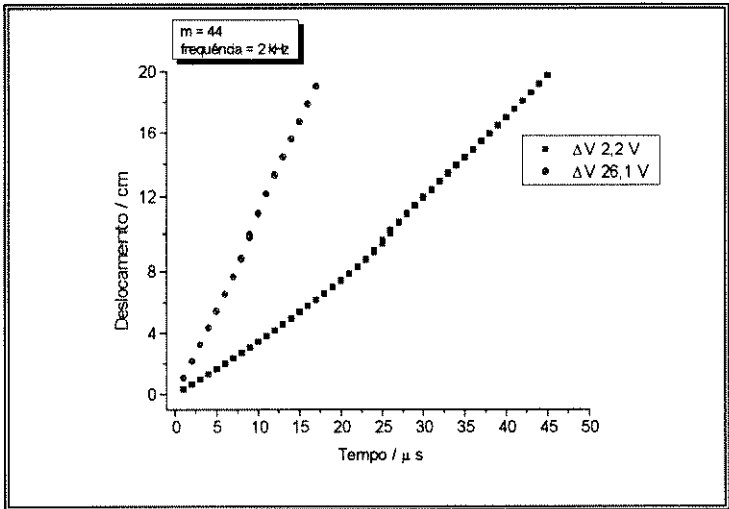


Figura IV.14: Simulações do deslocamento do íon m/z 44 contado a partir do instante em que entra em q2 sem a ação do circuito oscilador. As simulações foram feitas para duas energias cinéticas diferentes 2,2 eV (■) e 26,1 eV (●).

ÂNGULO DE SAÍDA θ_1 (°)	VELOCIDADE FINAL (m/s) - íon acelerado com 2,2 V -	VELOCIDADE FINAL (m/s) - íon acelerado com 26,1 V -	Fator de redução da velocidade (%)
5	2126,9	7325,8	68,5
10	2102,6	7242,1	67,7
15	2062,3	7103,2	66,4
20	2006,3	6910,3	64,6
25	1935,0	6664,8	62,3
30	1849,0	6368,6	59,5
35	1748,9	6023,9	56,3
40	1635,5	5633,3	52,7
45	1509,7	5199,9	48,6
50	1372,4	4726,9	44,2
55	1224,6	4218,0	39,4
60	1067,5	3676,9	34,4
65	902,3	3107,8	29,1
70	730,2	2515,1	23,5
75	552,6	1903,3	17,8
80	370,7	1277,0	11,9
85	186,1	640,9	6,0

Tabela IV.2: Diminuição da velocidade de um íon de m/z 44 u após um único choque com uma molécula de massa molar 84 g. O ângulo de saída corresponde ao desvio de trajetória sofrido pelo íon após a colisão. A velocidade do íon antes do choque é 3105,5 m/s e 10696,4 m/s quando acelerado da fonte até q2 com 2,2 V e 26,1 V, respectivamente. A molécula é considerada estar em repouso em virtude de sua velocidade ser muito menor (~297 m/s a 25 °C) em relação à velocidade inicial do íon.

LISTAGEM DO PROGRAMA UTILIZADO PARA SIMULAR O DESLOCAMENTO DOS ÍONS DENTRO DO QUADRUPOLO DE REAÇÃO (q2)

```
PROGRAM SIMULION;

Uses WinCrt;

const
  L = 0.20;

var
  acel, q2in, q2out, q2off, vp, vel0, bf, rep, ext, q1in, dt,
  v1i, a, b, c, b24ac, r1, r2 : Double;
  m, mole, i, teta1 : Integer;
  frequencia : LongInt;
  x, t, vel : Array[0..1000] of Double;
```

```
    arq : Text;

Function velo(v0, deltav : Double; u : Integer) : Double;
Begin
    velo := sqrt(v0*v0 + (1.0E03*6.02E23*1.602E-19*2*deltaV/u));
End;

Function Onda(teta,tempo,v : Double; freq : LongInt) : Double;
var
    a : Double;
Begin
    a := Sin(Pi*2*(tempo+teta)*freq);
```

APRISIONADOR DE ÍONS - IV.22

```

If a <> 0.0 then
  onda := V * a / abs(a)
else onda := 0.0;
End;

BEGIN
  Write('Frequência de oscilação (Hz) -> '); Readln(frequencia);
  Write('Tensão de pico do oscilador (V) -> '); Readln(vp);
  vp := 0.5 * vp * 200.0 / 15.0;
  Write('Massa do íon (u) -> '); Readln(m);
  Write('Massa da molécula -> '); Readln(mole);
  Writeln('Potenciais (volts):');
  Write('Bloco da fonte (#1) -> '); Readln(bf);
  Write('Repulsor (#2) -> '); Readln(rep);
  Write('Extrator (#3) -> '); Readln(ext);
  Write('Q1 in (#5) -> '); Readln(q1in);
  Write('Offset de q2 -> '); Readln(q2off);
  Write('Lente de entrada q2 -> '); Readln(q2in);
  Write('Lente de saída q2 -> '); Readln(q2out);
  Write('Leva em consideração às colisões? (S/N) -> ');
  If UpCase(ReadKey) = 'S' then
    begin
      vel0 := 0.0;
      teta1 := 0;
      v1i := velo(0,bf-q2off,m);
      a := m*m + m*mole;
      b := -2.0*m*m*v1i*cos(Pi*teta1/180.0);
      c := m*m - m*mole;
      b24ac := b*b - 4.0*a*c;
      If b24ac > 0 then
        begin
          r1 := (-b + sqrt(b24ac)) / (2.0*a);
          r2 := (-b - sqrt(b24ac)) / (2.0*a);
          If r1 > 0 then
            If r2 > 0 then
              begin
                Writeln('Existem duas soluções:',r1:0:4,' ',r2:0:4);
                Write('Qual a velocidade? -> '); Readln(vel0);
              end
            else vel0 := r1
          else
            If r2 < 0 then
              begin
                Writeln('NÃO TEM SOLUÇÃO');
                Readln;

```

```

          DoneWinCrt;
        end
      else vel0 := r2;
    end
  else
    begin
      Writeln('NÃO TEM SOLUÇÃO');
      Readln;
      DoneWinCrt;
    end;
  end
  else vel0 := velo(0,bf-q2off,m);
  Assign(arq,'simulion.dat');
  Rewrite(arq);
  vel[0] := vel0;
  t[0] := 0;
  x[0] := 0;
  i := 0;
  Repeat
    inc(i);
    t[i] := t[i-1] + dt;
    dt := t[i] - t[i-1];
    acel := -(q2out + onda(0,t[i],vp,frequencia) -
q2in)*6.02E26*1.602E-19/(m*L);
    vel[i] := vel[i-1] + acel*dt;
    x[i] := x[i-1] + vel[i]*dt + acel*dt*dt/2;
    Writeln(arq,t[i]*1.0e06,#9,x[i]*100);
  Until (x[i] >= L) or (x[i] <= 0);
  Close(arq);
  DoneWinCrt;
END.

```

CAPÍTULO V



V. SOFTWARE

Este capítulo apresenta o software desenvolvido para aquisição de dados do equipamento pentaquadrupolar, mostrando as estratégias que proporcionaram a obtenção de espectros complexos com rapidez e com excelente relação sinal/ruído.

V.1. INTRODUÇÃO À PROGRAMAÇÃO ORIENTADA A OBJETOS (POO) E MOTIVADA POR EVENTOS (PME)

O conceito de POO^[1,2] não é complicado. Os pontos básicos da POO são fáceis de aprender e não são mais difíceis de utilizar do que registros, matrizes, ponteiros ou qualquer outro assunto da programação convencional.

A diferença principal entre a programação convencional e a POO é a encapsulação. A programação na POO envolve a criação de objetos que contêm, ou encapsulam, dados e código. A porção de dados de um objeto consiste de variáveis do mesmo tipo utilizadas na maioria dos programas (*arrays*, *integers*, *strings*, etc.). A parte do código de um objeto consiste de procedimentos (*procedures*) e funções (*functions*), chamados coletivamente de métodos, que são idênticos (ou quase) aos procedimentos e funções utilizados na programação convencional.

Então o que existe de tão mágico em relação aos objetos? Pelo menos superficialmente, nada! Os objetos permitem relacionar dados com as sub-rotinas que atuam sobre estes valores de uma maneira que liga dados e código de um modo mais próximo do que seria possível sem a POO. A seguir é mostrado um exemplo de como é definido um objeto em Turbo Pascal.

```
UmObjeto = object
  X, Y : Integer;
  Procedure Ajusta(NovoX, NovoY : Integer); virtual;
  Function PegaX : Integer;
  Function PegaY : Integer;
end;
```

A razão primária para encapsular dados (X e Y no exemplo) e código (procedure Ajusta, function PegaX e function PegaY) no interior de objetos é assegurar que apenas um conjunto conhecido de métodos podem modificar uma ou mais variáveis. Em programas complexos, tal conhecimento é extremamente valioso, podendo simplificar grandemente o projeto, a programação e a depuração (*debug*) do programa. Apesar dos objetos parecerem mais complicados, no todo, e especialmente em grandes programas, estes objetos podem simplificar a programação de maneiras que não seriam possíveis com técnicas convencionais.

Uma vez definido um objeto, o passo seguinte é criar o que é chamado de instância de objeto. Todas as variáveis possuem um tipo de dado associado. Uma variável N pode ser do tipo *integer*, S do tipo *string* e assim por diante. Igualmente a *integer*, *strings* e outros tipos, os objetos também são tipos de dados e assim, pode-se criar variáveis deles. Quando os objetos são definidos como variáveis eles são chamados de instâncias. Deste modo, criam-se as instâncias da mesma forma que são criadas as variáveis de qualquer tipo de dado, como mostrado a seguir:

```
var
  UmaInstância : UmObjeto;
  OutraInstância : UmObjeto;
  A, B, C : UmObjeto;
```

Em cada instância de objeto, existe um local distinto para os campos declarados do objeto. Entretanto, existe apenas uma cópia de cada método associado com uma declaração de objeto. Assim, se existirem cinco instâncias de um UmObjeto, existirão cinco vezes a porção de dados (cinco campos X e cinco campos Y), mas somente uma porção de código (apenas uma cópia dos métodos Ajusta, PegaX e PegaY).

Além dos métodos definidos pelo programador, existem métodos com propósitos altamente refinados: **Constructor** e **Destructor**. Estes dois métodos são sempre opcionais. A diferença destes em relação à outros métodos só existe quanto se utiliza as instâncias como variáveis do tipo ponteiro (*pointer*). Neste caso deixam de ser opcionais e passam a serem obrigatórios, sendo responsáveis pela criação e destruição dos objetos na memória de alocação dinâmica (*heap*) do computador, respectivamente.

A encapsulação de dados e código é uma das características chave da POO, outra é a herança. A herança é uma característica dos objetos que permite que novas declarações de objetos herdem as propriedades de outros objetos. A herança torna possível declarar um objeto “filho” baseado no conteúdo de um objeto “pai”. O filho, também chamado de objeto descendente, recebe uma cópia dos campos de dados e declarações de métodos de seus pais. O objeto pai, frequentemente chamado de ancestral, pode, por sua vez, ser filho de um outro ancestral, que pode dever sua existência a uma longa linhagem numa árvore genealógica de declarações de objetos. No entanto, se um objeto filho é criado, mas necessita que um método seja modificado ou criado, basta declarar o método e fazer as alterações ou inclusões que o restante permanecerá inalterado. O importante é que o método alterado ou criado só é válido para este objeto filho e seus descendentes, não valendo para o pai nem para outros filhos. A única restrição é que somente métodos podem ser modificados, os campos de dados não podem ser alterados.

Em resumo, a herança em POO é perfeita, pois todas as informações genéticas dos pais são passadas inalteradas a todos os descendentes, não interessando quão distante eles forem.

Além das características de encapsulação e herança, a existência de métodos virtuais torna a POO ainda mais poderosa. As instâncias de objetos são tipicamente endereçadas por ponteiros. Por definição um ponteiro de um objeto ancestral pode endereçar quaisquer instâncias de seus descendentes. O inverso não pode ser feito: Os ponteiros dos descendentes nunca podem endereçar instâncias de objetos ancestrais.

Em função disto, um conflito potencial surge para os métodos que declaram métodos substitutivos. Dependendo de onde for feita uma declaração de instância, um método que tenha sido substituído pode não ser chamado, entrando o código do método do objeto ancestral. Tal problema ocorre com métodos estáticos que recebem endereços fixos determinados quando o programa é compilado.

Duas soluções existem para este dilema. Uma delas é mudar o tipo de declaração, forçando o endereçamento do objeto descendente. Contudo, tal abordagem requer acesso ao código fonte original, nem sempre disponível, além de que isto impossibilita o endereçamento das instâncias do objeto ancestral. A solução mais eficaz é declarar os métodos como virtuais, acrescentando a palavra-chave *virtual* no final da declaração. Isto faz com que a chamada a tais métodos seja determinada na ocasião da execução do programa, num processo conhecido como conexão

posterior. Assim, quando um ponteiro de uma instância endereçar um objeto do ancestral ou descendente, o programa determinará no momento da execução que método irá chamar. Como o método é virtual, seu endereço não é fixo, mas sim determinado pelo objeto endereçado. Se o ponteiro endereçar uma instância do ancestral, então o método original será chamado, mas se o ponteiro endereçar uma instância do descendente, o método substituto será chamado.

Ainda mais importante, esta ação ocorrerá para códigos que já estejam compilados. Se no interior de uma biblioteca de objetos existir um laço que chame um método virtual com um ponteiro P, mais tarde pode ser atribuído a P um descendente com uma nova declaração virtual que substituirá a anterior. Não será necessário recompilar a biblioteca para que isso aconteça. Com métodos virtuais, é possível prender códigos particulares no interior de objetos operacionais mesmo não possuindo o código fonte original.

A PME^[2,3] as vezes é confundida com POO, pois nela são utilizados objetos e métodos para responder aos eventos. Um evento é uma ação que ocorre a nível de sistema, quer seja a movimentação do mouse, acionamento de uma tecla ou redimensionamento de uma janela.

Em programação convencional e mesmo em POO, nos ambientes convencionais não motivados por eventos como o DOS, o fluxo de um programa é tipicamente incorporado nas sentenças deste e, conseqüentemente, segue uma sequência rígida. Quando um programa é executado, as sentenças são executadas sequencialmente a não ser que uma delas altere o fluxo do programa. O próprio programa controla quando suas subrotinas serão executadas e em que ordem.

Os ambientes motivados por eventos como o Windows e OS/2, necessitam de uma abordagem de programação diferente. Assim, na PME, embora as subrotinas contenham uma sequência rígida, o fluxo segue de acordo com a ocorrência de eventos externos a ele. Um programa motivado por eventos consiste na sua maior parte de subrotinas que respondem de maneira independente aos eventos. Como qualquer evento pode ocorrer em qualquer momento, não é mais possível prever a ordem na qual as subrotinas do programa serão executadas. Por exemplo, uma subrotina pode responder a comandos selecionados de um menu enquanto outra atualiza um relógio apresentado na tela. Como os usuários dos programas podem selecionar comandos a qualquer momento, não é possível prever a ordem na qual estes eventos ocorrerão.

De modo a programar estas duas tarefas acima de maneira convencional, teria que ser decidido se a rotina do relógio deveria monitorar o menu e responder a comandos, ou se a rotina

de menu deveria atualizar o relógio periodicamente. Qualquer uma destas duas abordagens poderia funcionar, mas se misturarmos a elas algumas dúzias de outras tarefas semelhantes juntamente com outros programas rodando em paralelo, pode-se entender porque as técnicas convencionais de programação não são aprovadas em um ambiente motivado por eventos.

Cada evento gera uma mensagem que é, prontamente, respondida pelo sistema. As mensagens podem ser originadas de muitas fontes, sendo listado abaixo três delas:

- O usuário, através de movimento ou pressionamento dos botões do mouse ou pressionando uma tecla do teclado, gera uma mensagem-de-usuário;
- Aplicações podem enviar mensagens para outras aplicações através da troca dinâmica de dados (*dynamic data exchange* - DDE);
- O ambiente de trabalho, incluindo o gerenciador de programas, envia mensagens para todas as aplicações abertas antes de fechá-las.

A resposta às mensagens-de-usuário é o meio primário para interagir com o usuário. Quando um evento ocorre, um módulo do sistema (USER.EXE, no caso do Windows), gera uma mensagem e a coloca em uma fila de resposta às aplicações. Cada aplicação é responsável por retirar sua mensagem e repondê-la. Desta maneira cada aplicação pode ser executada em conjunto com outras, cada uma recebendo a mesma atenção que as outras. Este é um dos mecanismos responsáveis pelo modo de trabalho em multitarefa. A figura V.I mostra um diagrama ilustrando o processo de comunicação entre as partes do sistema.

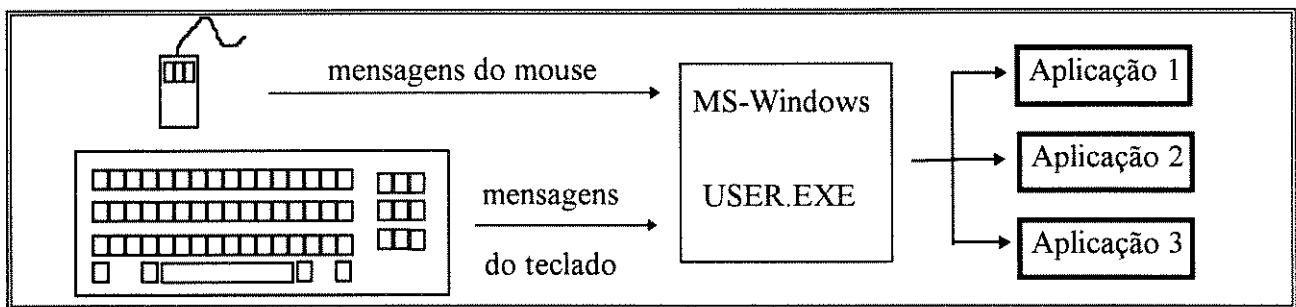


Figura V.I: Fluxo de mensagens entre aplicações no Windows.

As mensagens são as células sanguíneas que fluem pelas artérias de uma aplicação. As mensagens dão vida aos programas nestes ambientes motivados por eventos e a maioria dos

programas nestes ambientes dedicam uma grande parte do tempo enviando e respondendo várias mensagens.

§ V.2. IMPLEMENTAÇÃO

As rotinas básicas para controlar o cartão AT-MIO-16X foram fornecidas pelo fabricante. As rotinas para o cartão adicional, bem como os programas para aquisição e tratamento de dados foram desenvolvidos em Turbo Pascal 1.5 para Windows (Borland Inc.). As listagens dos programas e das rotinas de controle dos cartões encontram-se no apêndice 1 e 2, respectivamente.

Devido a complexidade do interfaceamento, nós optamos pela POO, que permite organizar as estruturas básicas e o crescimento do sistema. Para inicializar e finalizar os objetos, os métodos *constructor* e *destructor* foram implementados para todos os objetos. Além destes métodos, cada objeto tem seus próprios métodos, os quais estão relacionados na tabela V.1 e que serão discutidos a seguir.

Objetos	Métodos
TDetector	SetTempoInteg GetTempoInteg LeAD LeADAvg
TIonizador	Multiplex EI
TQuad	SetMassa GetMassa SetPoleDC JaEstavel
TPentaQuad	SetPoleDC SetMassa GetMassa Otimiza LeME LeMedia LinhaBase

Tabela V.1: Lista dos objetos e seus métodos.

• TDetector

Este objeto foi criado para adquirir a abundância dos íons. O método **TDetector.LeADAvg** minimiza o ruído e interferências sobre o sinal do ME. O sinal do ADC é adquirido pelo método **TDetector.LeAD** em um intervalo de tempo regular, ajustado pelo método

SetTempoInteg, e armazenado em um vetor. Este vetor é percorrido e a mediana é obtida para cada conjunto de três pontos. Este procedimento reduz a ocorrência de *outliers* (ver capítulo VI). O conjunto de dados resultante é somado e a média aritmética é retornada como a abundância dos íons. Normalmente o tempo de integração é ajustado para 33,4 ms (duas vezes o período da tensão alternada da rede elétrica). Este procedimento minimiza a interferência da onda senoidal da rede elétrica sobre o sinal do ME.

• **TIonizador**

Este objeto foi criado para controlar a energia de ionização, o qual contém dois métodos. O método **TIonizador.Multiplex** lê o potencial de todos os elementos da fonte de ionização, enquanto o método **TIonizador.EI** calcula e aplica, através do DAC, o novo valor da energia de ionização baseado no valor do potencial da fonte (ver item III.1.C).

• **TQuad**

Este objeto foi implementado para controlar os quadrapolos. Ele controla cada quadrapolo analisador de massa individualmente, ajustando os valores de m/z pela aplicação do método **TQuad.SetMassa**, que por sua vez aciona internamente os procedimentos DAC (para Q1 e Q5) e DAC725 (para Q3). Através do método **TQuad.SetPoleDC** cada quadrapolo de varredura é alternado entre os modos somente-RF e RF/DC.

O método **TQuad.GetMassa** fornece o valor de m/z atual ajustado em um quadrapolo. Este método é útil para outros objetos que necessitam saber qual o valor de m/z está ajustado em um determinado instante em um determinado quadrapolo.

Tanto o ADC quanto os DACs são muito rápidos, o que deixa a resposta do ME como uma função da constante de tempo de filtragem. Entretanto, o passo limitante é o tempo de acomodação dos valores de m/z nos quadrapolos analisadores de massa (ver item III.1.B - figura III.3). Enquanto o quadrapolo não tiver atingido a estabilidade, a leitura do ADC não pode ser tomada como a abundância do íons. Para contornar este inconveniente, foram usados os três contadores/temporizadores de 16 bits do cartão AT-MIO-16X, ajustados em uma base de tempo de microsegundos. Quando o método **TQuad.SetMassa** é chamado, um contador é disparado e o

método **TQuad.JaEstavel**, através do monitoramento deste contador, retorna um valor verdadeiro quando o tempo de acomodação é alcançado (equação III.1).

• **TPentaQuad**

Este é um objeto mais complexo que inclui os objetos TDetector, TIonizador e três objetos TQuad. Além disso, contém métodos para trabalhar com o instrumento como um todo (tabela V.1), discutidos a seguir:

O método **TPentaQuad.SetMassa** ajusta o valor de m/z do quadrupolo que está fazendo a varredura. Ao mesmo tempo ajusta os valores de m/z de transmissão máxima nos quadrupolos que estão no modo somente-RF conforme a equação III.2.

Os métodos **TPentaQuad.GetMassa** e **TPentaQuad.SetPoleDC** só fazem as chamadas aos métodos equivalentes em TQuad.

O método **TPentaQuad.LinhaBase** adquire o sinal da linha-base para cada valor de ganho do amplificador de entrada do ADC da placa AT-MIO-16X. Calcula média e desvio padrão de cada um, armazenando em um vetor, o qual será utilizado pelo método **TPentaQuad.LeMedia**.

O método **TPentaQuad.LeMedia** fornece a intensidade média do sinal do ME, descontando a intensidade da linha-base, através de várias chamadas ao método **TPentaQuad.LeME**. Este, por sua vez, adquire o sinal do ME, através da chamada ao método **TQuad.LeADAvg**, após a confirmação da estabilidade do quadrupolo pelo método **TQuad.JaEstavel**.

O método **TPentaQuad.Otimiza** é responsável pela localização do máximo de intensidade de um pico. Se um sinal excede o sinal da linha-base por um valor determinado, então este sinal será investigado numa varredura local ao redor da posição deste sinal.

A estrutura de **TPentaQuad** é a interface entre o hardware e os programas para aquisição de dados PENTA4D e SIMTIM.

Estes objetos de controle do equipamento, após terem sido devidamente testados, foram compilados em *units* que são utilizadas pelos programas de aquisição de dados. Primeiramente foi criada a *unit*, PENTAQUI.TPU, a qual contém elementos de controle dos quadrupolos e do detector. Posteriormente, a *unit* UPENTA4D.TPU foi criada utilizando basicamente os mesmos

objetos da primeira com a diferença de possuir um controle mais amplo e interativo, ou seja, ao controlar um quadrupolo, os demais são ajustados de modo a permitir uma melhor transmissão dos íons. Isto é conseguido através do objeto TPentaQuad.SetMassa que ajusta o valor de m/z de transmissão máxima, conforme o gráfico da figura III.5 e a equação III.2 correspondente.

Além dos objetos e métodos criados com a função de controlar uma parte específica do equipamento, foram implementados os objetos e métodos necessários para obter informações do mouse, para desenhar os espectros na tela e na impressora, bem como para copiar a porção da tela onde se encontra o espectro e converter em mapa de bit (*bitmap*). Associado a estes foram implementados os objetos e métodos que fazem os programas se tornarem motivados por eventos.

Programa PENTA2D:

Programa básico para aquisição de espectros de massas 2D. Utiliza todos os objetos descritos acima, exceto o TPentaQuad, controlando um quadrupolo por vez nas aquisições normais e dois ou três quadrupolos em aquisições de experimentos envolvendo ganho e perda neutra. Nas aquisições normais, é utilizado para auxiliar os usuários em uma aquisição de dados simples (MS^1 , MS^2 ou MS^3). Nas aquisições de dados de experimentos envolvendo somente o ganho de massa (MS^2), somente a perda de massa (MS^2) ou ganho e perda simultânea (MS^3), os quadrupolos são varridos simultaneamente com uma diferença de massa positiva ou negativa, no caso de ser ganho ou perda neutra, respectivamente.

Programa PENTA3D:

Em um segundo passo, este programa foi implementado, o qual permite varrer dois quadrupolos sequencialmente, gerando espectros de massas MS^2 e MS^3 3D. Este também não utiliza o objeto TPentaQuad. O processo de aquisição de um espectro é o seguinte: O primeiro quadrupolo (Q1 ou Q3) é varrido com incrementos de 0,1 u enquanto o segundo quadrupolo (Q3 ou Q5) permanece no modo somente-RF. Quando um valor de abundância excede o limite de detecção (dois desvios padrões do sinal da linha base), a posição do pico é registrada para posterior varredura do próximo quadrupolo. Assim, um espectro é adquirido rapidamente, dependendo apenas do tempo de integração escolhido e do número de picos encontrados na varredura prévia.

Programa PENTA4D:

Este programa é mais avançado que os anteriores. Permite adquirir espectros de massas 2D, 3D e 4D e controlar muitas funções do instrumento ao mesmo tempo. Para minimizar o tempo de varredura e a memória de armazenamento na aquisição de espectros 3D e 4D, foi desenvolvido o algoritmo descrito a seguir:

O primeiro quadrupolo é varrido com incrementos de m/z de 1 u enquanto os demais permanecem no modo somente-RF. Para cada valor inteiro de m/z , onde o sinal/ruído da abundância dos íons excede o limite de detecção, um pico é registrado. Desta maneira, um espectro 2D convencional é adquirido para este quadrupolo. Em um segundo passo, os picos com intensidades acima de dez desvios padrões são explorados pelos outros quadrupolos. A abundância é otimizada por uma varredura local em uma faixa de ± 0.2 u para encontrar a posição máxima. Este quadrupolo é, então, fixado neste valor de m/z e o processo de varredura é repetido para o próximo quadrupolo. Para um espectro 4D, o processo é executado para Q1, Q3 e Q5, mas o passo de otimização da abundância é suprimido para Q5. Este método economiza tempo e memória porque um quadrupolo é varrido somente quando um pico de intensidade apreciável é encontrado no quadrupolo anterior. O consumo de tempo e de memória depende de cada caso, mas normalmente o tempo é abaixo de dez minutos para um espectro 4D.

Por exemplo, em uma aquisição de dados onde os três quadrupolos são selecionados para varrer 200 u, se em Q1 são encontrados 50 picos, Q3 será varrido somente nestas 50 posições e, supondo que para cada posição sejam encontrados mais 50 picos, Q5 será varrido nestes 50×50 picos. Assim, se o tempo de integração for 33,4 ms, o tempo total de aquisição será:

$$t_{\text{aquis}} = (200 \times 33,4 \times 10^{-3}) + (50 \times 200 \times 33,4 \times 10^{-3}) + (50 \times 50 \times 200 \times 33,4 \times 10^{-3}) = 17040 \text{ s, ou aproximadamente, 4,7 horas.}$$

Entretanto, é importante notar que os valores foram superestimados, tanto na faixa de m/z quanto no número de picos em Q1 e Q3. Normalmente, um espectro de massas de reação íon-molécula e um espectro de fragmentação possuem, em média, 10 picos, gastando um tempo médio de 12 minutos. Em casos extremos, o tempo de integração também pode ser reduzido para 16,7 ms, o que implica em uma redução de 50% no tempo de aquisição, sem redução significativa na qualidade dos dados obtidos. Se este procedimento não fosse implementado, neste mesmo exemplo, o tempo total seria 268543 s ou aproximadamente 3 dias.

Programa SIMTIM:

Este programa foi desenvolvido tanto para o monitoramento de íon selecionado (SIM) quanto para o monitoramento total dos íons (TIM) em função do tempo, também conhecido por corrente iônica total (TIC). Adicionalmente, o monitoramento de íons totais pode ser feito em função do potencial de ionização (IEM). Este utiliza todos os objetos de controle do espectrômetro, descritos anteriormente, aliado às rotinas de programação por temporização do Windows. Desta maneira o programa tem capacidade de adquirir dados de forma totalmente independente do usuário da análise, deixando inclusive o computador livre para outras tarefas durante a aquisição de dados.

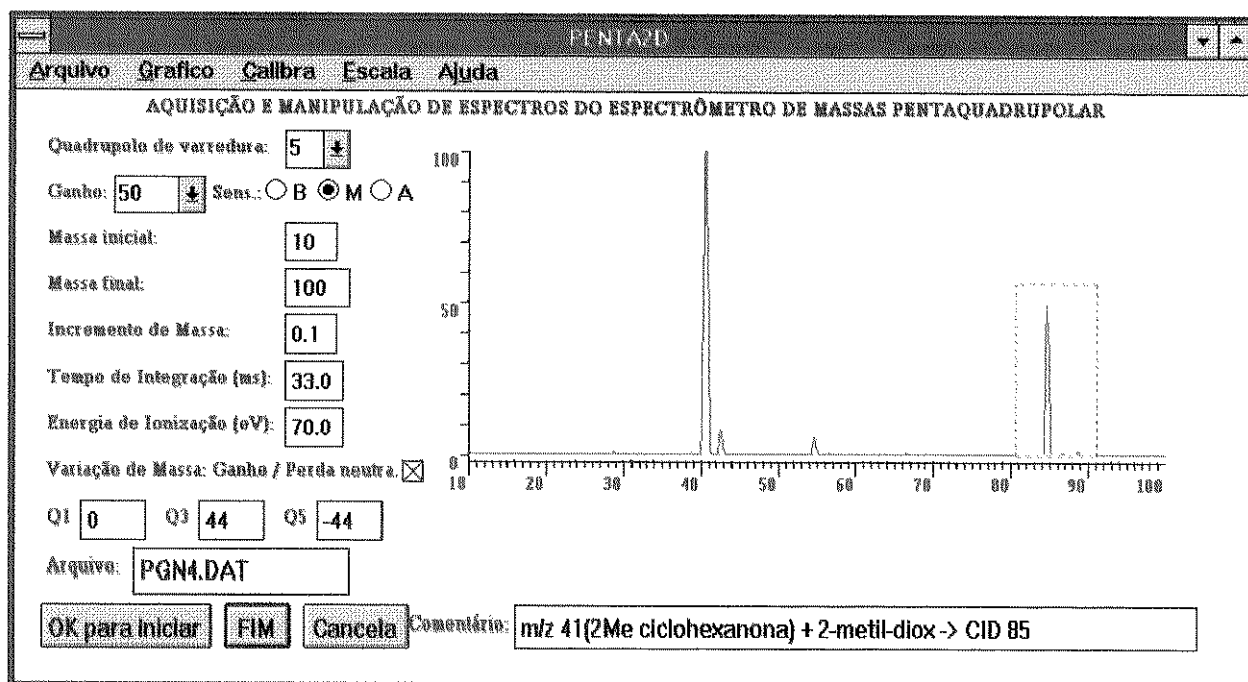
No modo SIM, embora o sistema tenha capacidade para monitorar mais de cem janelas-objeto, cada uma correspondente a um íon, o número de janelas foi restringido para seis em função de já serem raros os casos em que são necessários monitorar quatro ou mais íons de uma única vez.

Tanto no modo SIM quanto no TIM, são obtidas curvas de abundância vs tempo, as quais podem ser utilizadas para um acompanhamento cinético ou mesmo para a determinação quantitativa de uma substância. Adicionalmente, para determinação quantitativa, existem rotinas que geram curvas analíticas através da introdução de padrões.

Na parte referente ao monitoramento total dos íons, a aquisição é mostrada em tempo real no formato abundância total vs tempo ou potencial de ionização. Posteriormente, por intermédio do ponteiro do mouse, são extraídos os espectros de massas correspondentes a cada intervalo de tempo decorrido ou potencial de ionização.

Todos os programas desenvolvidos utilizam os mesmos tipos de objetos. PENTA2D e PENTA3D utilizam os mesmos objetos da mesma unidade PENTAQUI.TPU, enquanto PENTA4D e SIMTIM utilizam os mesmos objetos da unidade UPENTA4D.TPU, porém cada programa possui alguma particularidade. A seguir são mostrados as telas de apresentação de cada programa, identificando suas particularidades e características. No capítulo VII serão mostrados alguns exemplos de aplicações com cada um dos programas aqui apresentado.

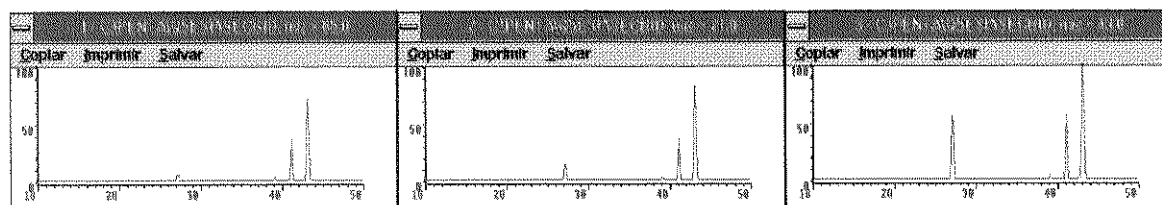
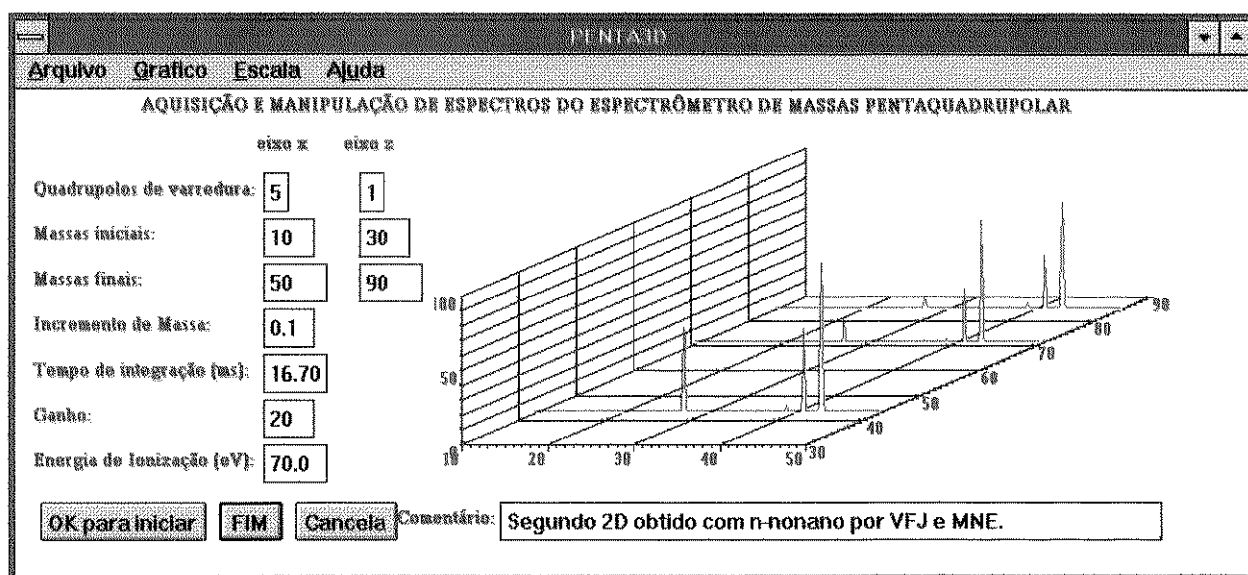
- **PENTA2D:** Este é particularizado pela sua simplicidade de operação;



Características:

- ✓ Aquisição de espectros MS^1 , MS^2 e MS^3 2D.
 - ⇒ Inclui varredura inédita para estudos de fragmentação por perda neutra e de reação íon-molécula com ganho neutro.
- ✓ Filtragem digital do sinal por mediana:
 - ⇒ O sinal (S_{MED}) é representado pela mediana de 3 leituras;
- ✓ Filtragem por média aritmética (tempo de integração):
 - ⇒ O sinal (S_{MA}) é representado pela média de n sinais S_{MED} durante o tempo de integração escolhido pelo usuário;
- ✓ Calibração semi-automática:
 - ⇒ A presença do usuário é necessária apenas para selecionar individualmente os quadrupolos que serão calibrados;
- ✓ Reescalamento do gráfico pelo mouse.
 - ⇒ Através do mouse e seus botões é selecionada a região de interesse do gráfico a ser expandida (linha tracejada na tela de apresentação).

- **PENTA3D:** Sua particularidade consiste em obter espectros 3D com rapidez;



Características:

- ✓ Aquisição de espectros MS² e MS³ 3D com varredura selecionada;
- ✓ Filtragem digital do sinal por mediana;
- ✓ Filtragem por média aritmética (tempo de integração);
- ✓ Reescalamento do gráfico pelo mouse;
- ✓ Extração de espectros 2D;



- **PENTA4D:** O controle simultâneo dos três quadrupolos e a aquisição de espectros 4D com rapidez e otimização da posição dos máximos dos picos são as suas particularidades;

1D 11DADIMENSIONAL - (C) V.F.J 1994

Arquivo Gráfico Escala Ajuda

AQUISIÇÃO E MANIPULAÇÃO DE ESPECTROS DO ESPECTRÔMETRO DE MASSAS PENTAQUADRUPOLO

Quadrupolos de varredura: ☒ 1 ☒ 3 ☒ 5

Massas iniciais: 20 20 20

Massas finais: 400 400 400

Energia de Ionização (eV): 70.0

Sensibilidade: ☐ B ☐ M ☒ A

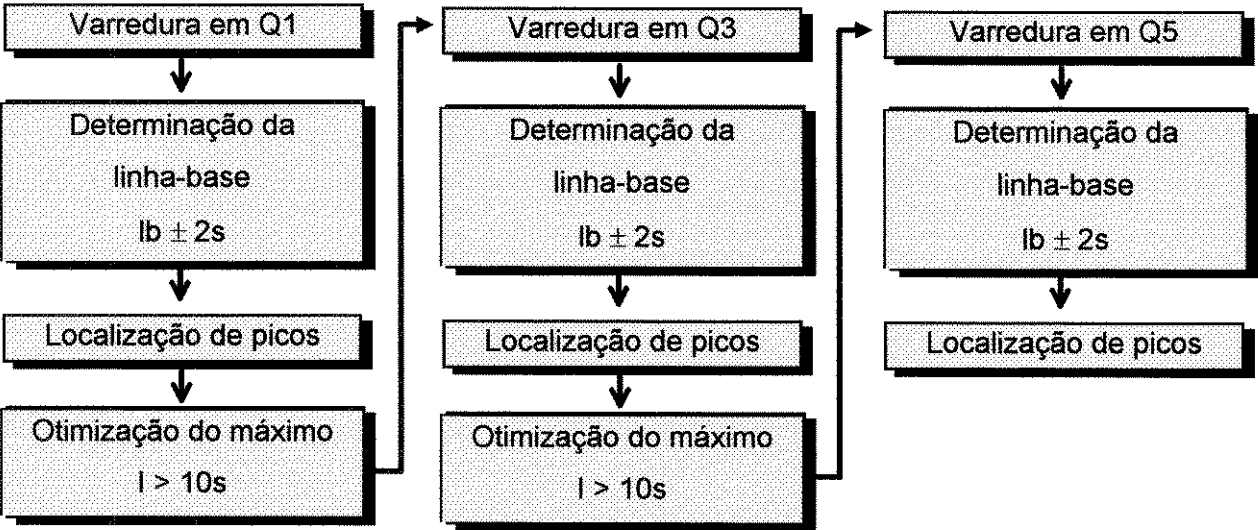
Arquivo: aquis4D

INICIA SAI CANCELA

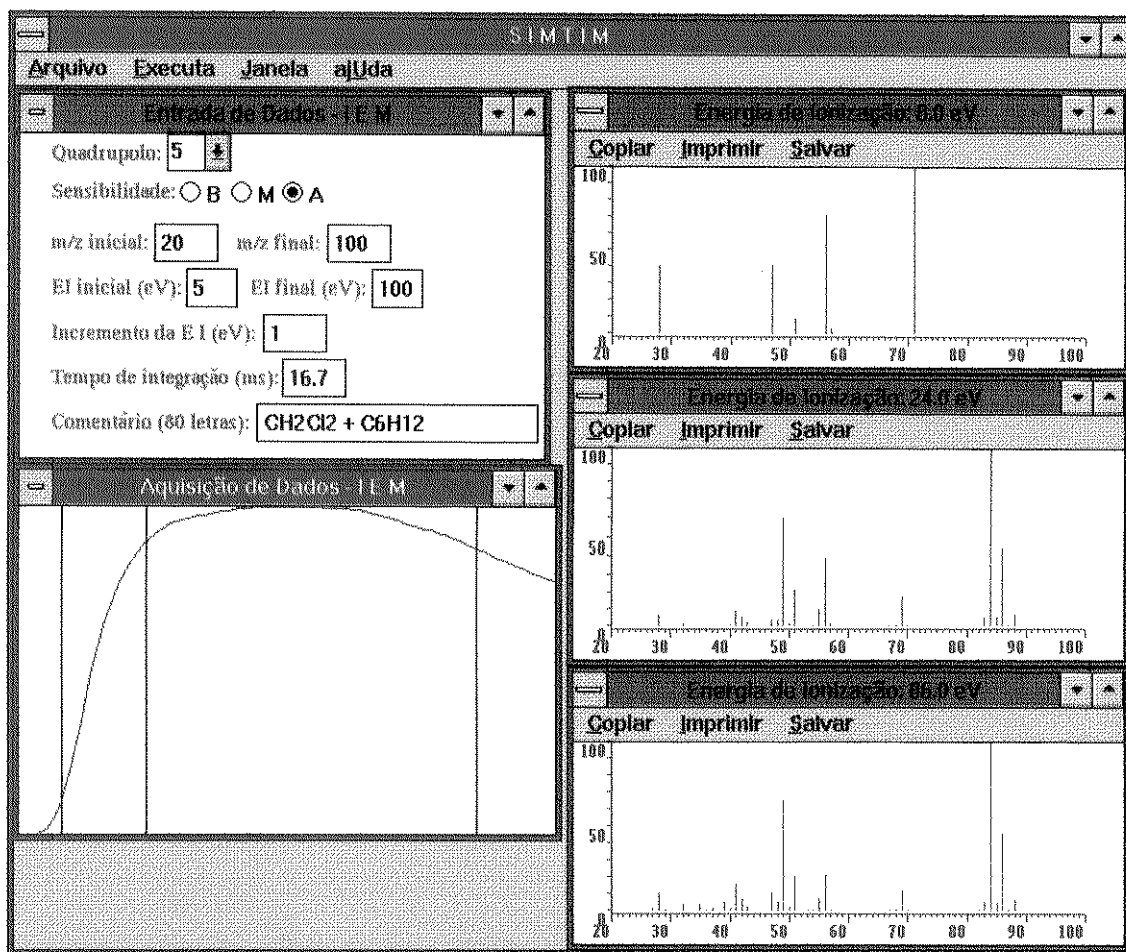
Comentário: Aquisição de dados 1,3 diox + 2-butanona

Características:

- ✓ Aquisição de espectros MS¹, MS² e MS³ 2D; MS² e MS³ 3D e MS³ 4D com varredura selecionada;
- ✓ Filtragem digital do sinal por mediana;
- ✓ Filtragem por média aritmética (tempo de integração);
- ✓ Extração de espectros 3D.



- **SIMTIM:** O método de temporização do Windows particulariza seu método de aquisição nas técnicas de SIM e TIM.



Características:

- ✓ Aquisição de dados em função do tempo:
 - ⇒ Monitoramento de íon selecionados (SIM);
 - ⇒ Monitoramento de íons totais (TIM);
- ✓ Aquisição de dados em função do potencial de ionização (IEM):
 - ⇒ Monitoramento de íons totais;
- ✓ Curvas de calibração através da introdução de padrões;
- ✓ Filtragem digital do sinal durante a aquisição:
 - ⇒ Mediana e média aritmética (tempo de integração);
- ✓ Filtragem digital do sinal após a aquisição:
 - ⇒ Mediana móvel e média móvel;

☞ V.3. DISCUSSÃO

A criação de programas no ambiente Windows proporciona ao usuário uma grande interação com outros programas que rodam neste “sistema operacional”.

A programação orientada a objetos torna o sistema muito flexível a alterações. Por exemplo, se fosse retirado ou adicionado um quadrupolo no aparelho, isto não afetaria em nada o software, pois o objeto quadrupolo já existe. Porém, se o quadrupolo adicionado tivesse características diferentes dos demais, estas características teriam que ser implementadas no objeto. Neste caso, teríamos objetos descendentes da mesma origem cada um com a sua particularidade, no entanto, a “família” seria constituída pelos pais e pelos filhos, gêmeos ou não. Enfim, é isto que torna o sistema facilmente expansível.

O método de varredura selecionada nos programas PENTA3D e PENTA4D acelera muito a aquisição de dados. No caso do programa PENTA3D, o ganho de velocidade (G) em relação a uma aquisição normal pode ser expresso por:

$$G = \Delta u_z / np;$$

onde Δu_z é a região de m/z varrida no eixo z com incrementos unitários e np é número de picos encontrados na varredura do outro quadrupolo (eixo x).

No caso de uma aquisição 4D com o programa PENTA4D, a expressão é um pouco mais complicada:

$$G = [\Delta u_1 \cdot \Delta u_3 (1 + \Delta u_5)] / [np_1 \cdot (\Delta u_3 + np_3 \cdot \Delta u_5)];$$

onde Δu_1 , Δu_3 e Δu_5 são as regiões varridas em Q1, Q3 e Q5, respectivamente, e np_1 e np_3 são os números de picos encontrados em Q1 e Q3.

Devido ao método de integração do sinal, todos os programas fornecem excelentes resultados com excelente relação sinal/ruído. Por exemplo, para o pico base de um espectro de diclorometano (m/z 49), obtido por EI a 25 eV em Q3 com Q1 e Q5 no modo somente-RF e com uma pressão de $1,4 \times 10^{-6}$ torr registrada no sensor localizado em Q5, a relação sinal ruído tem o valor aproximado de 500. Isto corresponde a um limite de detecção de picos da ordem de 0,2% da intensidade do pico base. Além disso, logo após o término da aquisição, as curvas de abundância vs tempo obtidas pelo programa SIMTIM são processadas com o método de mediana móvel e

média móvel de modo a reduzir sensivelmente os ruídos (ver discussão sobre filtragem digital de sinais no capítulo VI).

Dos programas desenvolvidos, o PENTA3D e o PENTA4D merecem alguns comentários adicionais. A partir dos espectros MS^3 3D, obtidos pelo PENTA3D, são extraídos os espectros MS^3 2D de cada posição do eixo de m/z correspondente ao primeiro quadrupolo de varredura (Q3 ou Q1). Isto proporciona uma investigação mais detalhada em cada espectro. Além disso, a partir da soma das intensidades dos sinais que aparecem em um dos eixos de m/z , pode ser reconstruído o espectro MS^2 2D de uma determinada reação íon-molécula. Este, por sua vez, contém a proporção correta das intensidades dos picos, as quais são mascaradas no espectro MS^3 3D devido às fragmentações dos íons. O mais importante disso é que este tipo de informação não é possível de ser obtida por outros meios.

Dos espectros MS^3 4D, obtidos pelo PENTA4D, podem ser extraídos os espectros MS^3 3D, MS^3 2D, MS^2 3D e MS^2 2D, bem como a reconstrução do espectro MS^2 2D, de maneira semelhante a que é feita a partir dos espectros MS^3 3D do programa PENTA3D.

➤ V.4. CONCLUSÃO

O primeiro ponto importante a ressaltar é a obtenção inédita de espectros 4D e espectros por ganho/perda neutra consecutiva, os quais contribuem para o progresso da espectrometria de massas sequencial.

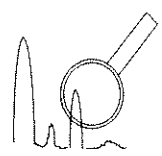
Os métodos empregados na varredura de espectros 3D e 4D são bastante eficientes, proporcionando a obtenção destes com rapidez. Os métodos de filtragem digital eliminam os interferentes sem perda significativa de qualidade do sinal analítico.

Os programas desenvolvidos para o equipamento foram muito bem aceitos pelos usuários. Em parte, a aceitação foi devida aos programas terem sido desenvolvidos para o ambiente Windows, o qual tem uma interface amigável. Outro motivo da aceitação foi a simplicidade de operação dos programas, tendo pontos de entrada de dados muito bem definidos, não deixando dúvidas a respeito da informação desejada em cada campo. Finalmente, o modo em que os dados são armazenados permite aos usuários utilizarem os programas gráficos comerciais de suas preferências para tratá-los.

V.5. BIBLIOGRAFIA

1. Budd, T. *An Introduction to Object-Oriented Programming*, Addison-Wesley: New York, 1991.
2. Swan, T. *Programando em Turbo Pascal para Windows 3.0*, vol. 1, Berkeley: Rio de Janeiro, 1992.
3. *Turbo Pascal for Windows - Windows Programming Guide*, Borland International Inc:Scotts Valey, CA, 1991.

CAPÍTULO VI



VI. FILTRAGEM DIGITAL DE SINAIS POR MEDIANA MÓVEL

“A técnica de filtragem de sinais, em alguns casos, pode ser mais complexa que a própria interpretação dos dados adquiridos”.

Este capítulo trata de um problema muito comum em aquisição de dados - o ruído. Será mostrado a seguir um procedimento de filtragem digital por mediana móvel, o qual não tem sido muito usado para tratamento de dados em química. As informações contidas neste capítulo são basicamente as mesmas publicadas recentemente.^[1]

VI.1. INTRODUÇÃO

Vamos considerar dois tipos de perturbação sobre o sinal a ser processado: *outliers* e *spikes*. O primeiro corresponde a uma perturbação no vetor de dados em uma direção sem nenhuma correlação com os dados nas posições vizinhas da mesma maneira que ocorre quando se digita manualmente um dado. Isto pode representar um erro grande, isto é, digitar 910 no lugar de 190. Em sistemas automáticos os erros podem ocorrer durante a transferência de valores adquiridos por um conversor analógico/digital. Como exemplo, um conversor A/D produz um o valor binário 11010111 (215 em decimal) e, durante a transferência, uma variação na voltagem, correspondente ao bit mais significativo, altera o valor para 01010111 (87 em decimal). O segundo tipo de ruído, o *spike*, é considerado como uma perturbação que afeta posições vizinhas de um vetor de dados. Como exemplo, a partida de um instrumento de alta potência pode causar uma perturbação eletromagnética que altera um sinal elétrico que está sendo adquirido em função do tempo. Se o *spike* tem pequena duração, comparada com a taxa de aquisição, ele pode se tornar um *outlier*. Em qualquer caso, filtros digitais devem eliminar ou, pelo menos, minimizar estas perturbações.

Por muitos anos a estatística clássica, que é baseada na distribuição normal, tem sido utilizada em tratamento de dados. De fato, na maioria dos casos, isto tem funcionado

relativamente bem para descrever dados reais, entretanto, a presença de *outliers* ou *spikes* podem invalidar esta aproximação. Neste caso, métodos de estatística robusta são necessários.^[2,3]

Muitas contribuições elaboradas com regressão linear tem sido apresentadas,^[4-6] entretanto, pouca atenção tem sido dada pelos químicos para outro tipo de procedimento usual em tratamento de dados: o filtro digital. Os procedimentos conhecidos assumem que o ruído tem uma distribuição normal (ruído branco - *white noise*).^[7-10] Caesar e Klier^[11] investigaram a eliminação de *outliers* ou *spikes* em cromatogramas, entretanto, o processo é inerentemente lento e ainda baseado na distribuição normal. Mesmo em trabalhos recentes, tratamentos matemáticos relativamente complexos tem sido usados para eliminar *outliers*.^[12] Phillips and Harris^[13] trataram da eliminação de *outliers* em espectros Raman obtidos CCD (*charge-coupled device*) com um tipo de filtro linear. Entretanto, é conhecido que programas disponíveis comercialmente utilizam um filtro por mediana, o qual não é linear, para resolver o mesmo tipo de problema.^[14]

A idéia é percorrer o vetor de dados em uma maneira similar a que é feita pela media móvel,^[7] com a substituição do valor analisado com a mediana dos valores vizinhos. Ainda que pouco usado em química, este procedimento é bastante comum em processamento de imagens,^[15-19] onde filtros lineares deformam as fronteiras em regiões de grande contraste. O uso da mediana em filtros digitais foi introduzida por Tukey^[20] e tornou-se um método padrão para eliminação de ruídos em imagens digitalizadas.^[15] Programas como Photostyler^[21] contém este recurso e tem-se informação que o programa LabVIEW 2,^[22] o qual trata dados adquiridos de vários tipos de instrumentos, também permite filtragem por mediana móvel. Várias implementações de mediana móvel, filtro por mediana^[15,23] ou híbridos do método^[24-26] podem ser encontradas na literatura. Entretanto, não temos observado seu uso em tratamento de dados para muitas das técnicas, espectroscópicas ou não, usadas em química.

Neste capítulo apresentamos e discutimos uma aproximação probabilística para filtragem digital de ruídos e extendemos o trabalho de Davies^[27] que estudou os efeitos relativos dos filtros por mediana e média móveis sobre sinais ruidosos.

✂ VI.2. TEORIA E METODOLOGIA

Analogamente à média móvel, o filtro por mediana móvel usa uma região do vetor de dados para estimar um valor central. Esta região, conhecida como janela de filtragem (*smoothing*

interval), pode ser variada em tamanho para acomodar o problema a ser resolvido. É formada por um ponto central e duas margens simétricas. Desta maneira, o conjunto de pontos tem um valor ímpar que facilita o cálculo da mediana. Este procedimento percorre o vetor de dados por completo gerando um conjunto menor de dados filtrado.

O caminho mais simples e mais usado para calcular a mediana é ordenar os pontos na janela de acordo com a intensidade do sinal. A mediana é simplesmente o valor central e, por isso, nenhum cálculo matemático é envolvido. Uma vez feita a primeira ordenação, não é mais necessário repetir o processo para o próximo ponto. Considerando que a filtragem é feita do início ao fim do vetor, a posição do vetor com o menor índice é eliminado e um novo ponto é adicionado. Para fazer isto, somente é necessário percorrer a janela em busca de uma posição apropriada para este novo ponto, permanecendo a janela ordenada.

A mediana é considerada um estimador robusto porque tem um ponto de falha (*breakdown point*) de 50%.^[2] Isto significa que é necessário o conjunto de dados conter 50% de *outliers* para proporcionar um desvio significativo na mediana. Este valor é mesmo válido para o caso menos favorável, quando *outliers* são todos maiores ou menores que a mediana.

Da mesma maneira que no filtro por média móvel, a escolha de uma janela de filtragem excessivamente larga pode causar uma deformação indesejável no conjunto de dados. Deve-se, portanto, estabelecer uma maneira bem planejada para encontrar esse tamanho. Se a probabilidade que um *outlier* possa ocorrer $P(O)$ é conhecida e os eventos não são correlacionados, então podemos considerar um modelo de probabilidade no qual cada posição no vetor é um evento independente. Neste caso, podemos avaliar a probabilidade de um conjunto de dados estar contaminado por, pelo menos, um *outlier* $P(Onf)$:

$$P(Onf) = 1 - (1 - P(O))^N \quad (VI.1)$$

onde N é o tamanho do vetor de dados. Por exemplo, se a probabilidade $P(O)$ é 0,001 e temos um vetor com 10000 posições, então a probabilidade que o vetor esteja contaminado por, pelo menos, um *outlier* é 0,99995. Assim, é certo que ele esteja contaminado.

Vamos considerar agora que o vetor seja filtrado pela mediana móvel com uma janela de 3 pontos. Por causa do ponto de falha de 50%, é necessário que, pelo menos, dois *outliers* apareçam dentro desta janela, para que a filtragem não seja bem sucedida. A probabilidade que isso ocorra $P(O_m)$ é calculado como:

$$P(O_m) = \sum_{i=m}^w \binom{w}{i} P(O)^i (1 - P(O))^{w-i} \quad (VI.2)$$

onde m é o menor número de *outliers* para o filtro falhar e w é o tamanho da janela. Podemos considerar novamente a equação VI.1 com N igual a 9998 e $P(O_m)$ no lugar de $P(O)$ para obter $P(Onf)$ igual a 0,03, isto é, em aproximadamente 100 vetores, somente 3 permanecerão contaminados. Isto é um resultado muito bom quando comparado ao primeiro caso.

Por outro lado, como acontece com outros tipos de filtros, a mediana móvel afeta não só o ruído, mas o sinal analítico também. A figura VI.1 mostra os efeitos dos filtros por média e mediana móveis sobre uma curva Gaussiana. Duas observações podem ser feitas quando o tamanho da janela de filtragem é aumentado. Primeiro, nenhum alargamento no formato do pico é observado com o filtro por mediana móvel. Já com a média móvel e outros filtros lineares em uso, o alargamento e a perda de resolução concomitante é uma ocorrência comum. Segundo, o topo do pico é cortado (*cut off*) com o filtro por mediana móvel. Isto sempre ocorre em uma região que contém um pico ou vale.

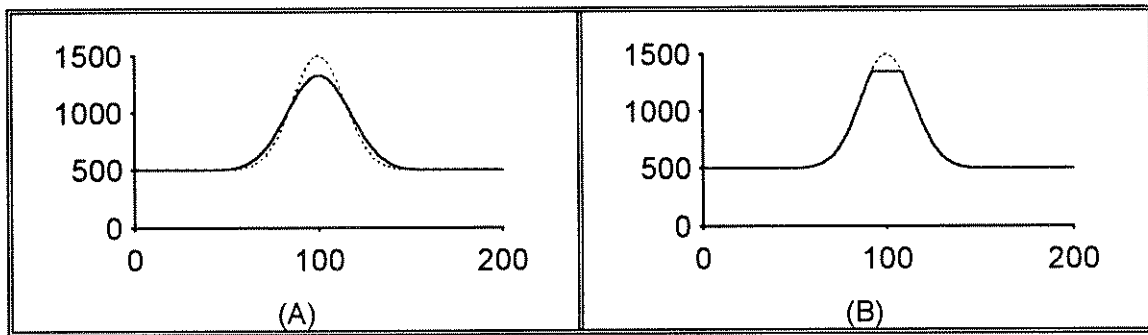


Figura VI.1: *Smoothing* de um pico Gaussiano simulado filtrado por (A) média móvel e (B) mediana móvel. A linha pontilhada representa o sinal original e a linha contínua o sinal filtrado. Em ambos os casos, foi usado um vetor de 200 posições com uma janela de filtragem de 31 pontos.

Este efeito de corte pode ser predito, permitindo ajustar o tamanho da janela de filtragem para minimizar este efeito. Para muitos picos assimétricos, os pontos próximos do centro são substituídos pelo ponto para o qual o índice é uma distância d a partir do índice do ponto central:

$$d = \text{Int}\left(\frac{w+1}{4}\right) \quad (\text{VI.3})$$

onde w é o tamanho ímpar da janela e Int denota a parte inteira do argumento (ver item VI.A).

Para uma curva Gaussiana, o conjunto de dados pode ser descrito por uma função do tipo:

$$y = y_{\max} e^{-\frac{(x-m)^2}{s^2}} \quad (\text{VI.4})$$

onde $y_{\max}$, m e s são a máxima altura, posição e largura, respectivamente. Se o intervalo Δx corresponde a duas posições vizinhas de x , então podemos relacionar a altura do pico depois da filtragem y_f com a altura original $y_{\max}$ por:

$$\frac{y_f}{y_{\max}} = e^{-\frac{(d \cdot \Delta x)^2}{s^2}} \quad (\text{VI.5})$$

ou

$$d = \frac{s}{\Delta x} \sqrt{\ln\left(\frac{y_{\max}}{y_f}\right)} \quad (\text{VI.6})$$

Desta maneira, se a largura do pico puder ser avaliada, o tamanho da janela de filtragem poderá ser ajustada em tal sentido para minimizar a redução do sinal.

A filtragem com a mediana móvel não somente elimina outliers como também proporciona uma redução substancial do ruído branco. Com a média móvel, a redução na variância do ruído é igual ao tamanho da janela de filtragem. Isto é esperado porque cada ponto do vetor é substituído pela média da janela e a variância da média é submetida à mesma redução.

Após a filtragem pela mediana, a variância (σ_{MM}^2) é dada por:

$$\sigma_{MM}^2 = \sigma^2 \left(\frac{\pi}{2(w+2)} + \frac{\pi^2}{4(w+2)(w+4)} \right) \quad (\text{VI.7})$$

ou na forma simplificada dada por Davies:^[27]

$$\sigma_{MM}^2 = \frac{1,571 \cdot \sigma^2}{w + 0,429} \quad (\text{VI.8})$$

O comportamento mostrado na figura VI.1 é muito curioso e pode servir como um guia na implementação deste tipo de filtro. Se por um lado, a largura do pico não é afetada, sua altura definitivamente é. Em muitos casos será possível se beneficiar da primeira característica sem sofrer os efeitos indesejáveis da segunda. Se tomarmos como exemplo um registro de uma

titulação potenciométrica, a qual é monotônica, o efeito de corte não ocorrerá. Consequentemente, podemos usar filtros de banda larga, os quais eliminam *outliers* e ruídos sem deformar a curva de titulação. Por outro lado, o efeito de corte pode ser útil. Em alguns casos, *spikes* têm duração tão grande quanto o intervalo de tempo para aquisição de dois pontos de dados (Δx). Neste caso, a aproximação que as ocorrências de *outliers* não são eventos correlacionados não é válida e a aproximação probabilística deve ser corrigida. Em tal caso, podemos usar o efeito de corte para reduzir o *spike* para um valor aceitável. Se o sinal de perturbação ocupa um número de pontos que é muito menor que a largura do pico, então o tamanho da janela pode ser ajustado para eliminar completamente este ruído sem afetar significativamente o sinal analítico.

Na figura VI.2 o filtro de mediana móvel foi aplicado para um pico contaminado com ruído branco, *outliers* e *spikes*. Uma vez que a largura do *spikes* ocupa 3 pontos consecutivos, podemos usar uma janela de filtragem de 7 pontos. Para comparação, esta mesma figura mostra o efeito de um filtro por média móvel de mesmo tamanho de janela. Podemos observar que a eliminação de *outliers* é muito mais eficiente com a mediana móvel.

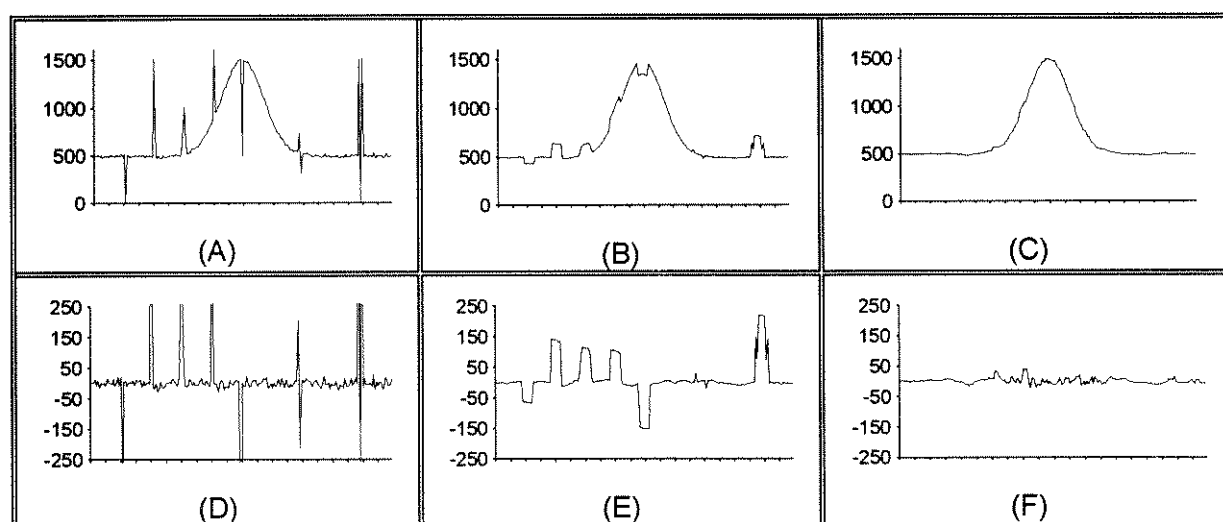


Figura VI.2: *Smoothing* de um pico Gaussiano simulado (A), onde foram introduzidos ruído branco ($\sigma=10$), *outliers* e *spikes*. A janela de filtragem para média móvel (B) e mediana móvel (C) foi de 7 pontos. Os sinais (D), (E) e (F) representam, respectivamente, as curvas (A), (B) e (C) após subtração do sinal original (linha pontilhada na figura VI.1).

Foi também observado que, no caso do sinal filtrado pela mediana móvel, o ruído residual muda de intensidade ao longo do pico (fig. VI.2F). Este comportamento levou-nos a estudar o comportamento do filtro por mediana móvel como uma função da derivada do sinal original. Após um longo número de simulações, obtivemos empiricamente a seguinte modificação da equação VI.7:

$$\sigma_{MM}^2 = \sigma^2 \left(\frac{\pi}{2(w+2)} + \frac{\pi^2}{4(w+2)(w+4)} \right) \cdot \left(\frac{4 + \alpha(w-1)}{4} \right) \quad (VI.9)$$

onde α é um fator que está relacionado com a derivada $\frac{dy}{dx}$ e com o desvio padrão do sinal, conforme a equação:

$$\alpha = \frac{dy}{dx} \cdot \frac{\Delta x}{\sigma} \quad (VI.10)$$



VI.3. APLICAÇÕES

Nós temos aplicado com sucesso o filtro por mediana em espectros de massas obtidos em um espectrômetro de massas de dupla focalização MAT311A e nos dados obtidos por um dos programas de aquisição de dados do espectrômetro de massas pentaquadrupolar.

No primeiro caso dois sinais são obtidos: um proveniente do multiplicador de elétrons, adquirido por um conversor A/D de 12 bits da Analog Devices, e outro proveniente do sensor de campo magnético.^[29,30] Para o primeiro é utilizado uma janela de 3 pontos e para o segundo uma de 21 pontos.

No segundo, o sinal do multiplicador de elétrons, adquirido por um conversor A/D de 16 bits do cartão AT-MIO-16X (ver capítulo III), é filtrado pela mediana móvel com janela de 3 pontos e em seguida é filtrado pela média móvel com a mesma janela.

VI.4. DISCUSSÃO

No caso da aquisição de dados do espectrômetro MAT311A, para o sinal do ME, um *outlier* no processo de conversão A/D pode ser confundido com um pico. Já para o sensor de campo magnético, um *outlier* pode causar uma leitura imprecisa no campo magnético e, por consequência, no valor de massa do pico. Uma janela de filtragem grande é utilizada neste

segundo caso por duas razões: (1) um erro no valor de massa pode ser um problema muito sério e (2) o sinal é monotônico, isto é, aumenta constantemente e, por isso, não está sujeito ao efeito de corte.

A figura VI.3 mostra uma pequena região dos vetores que correspondem aos sinais do ME e campo magnético. Pequenos *spikes* (fig. VI.3B) aparecem devido às perturbações causadas pelo termostato do sensor. Estas são suficientes para causar erros na conversão do campo magnético para valores de m/z . A figura VI.3D mostra este mesmo sinal após ter sido filtrado com uma janela de 21 pontos. A figura também mostra o sinal do multiplicador de elétrons antes (fig. VI.3A) e depois (fig. VI.3C) de ter sido filtrado pela mediana móvel. Uma vez que a resolução não é constante, um tamanho ótimo de filtro não pode ser utilizado para todo o espectro. Entretanto, com w igual a 3, os *outliers* são eliminados sem efeito de corte significativo sobre os picos.

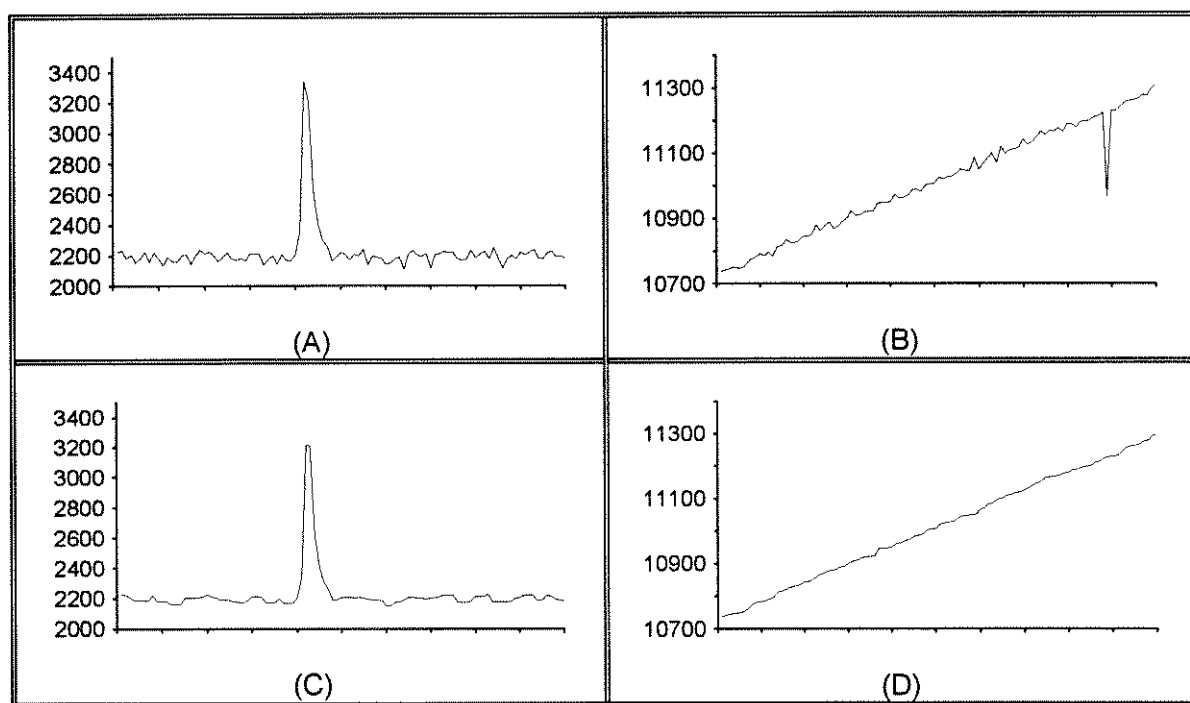


Figura VI.3: Efeito do filtro por mediana móvel sobre os sinais do multiplicador de elétrons (A e C) e do campo magnético (B e D) em um espectro de perfluorotributilamina (PFTBA) adquirido em um espectrômetro de massas MAT311A. Uma janela de 3 pontos foi usada em (C) e de 21 pontos em (D). Os gráficos mostram uma pequena região dos vetores de dados com 100 pontos.

A figura VI.4 mostra o sinal de um monitoramento de íons totais em função do potencial de ionização, obtido pelo programa SIMTIM, antes (VI.4A) e depois (VI.4B) do processo de filtragem. Tanto neste caso como no anterior, a filtragem reduz significativamente o nível de ruídos, tornando o sinal bem mais “limpo”.

No apêndice VI.A.2 encontra-se a listagem da rotina de filtragem por mediana móvel com janela de filtragem igual a 3. A execução desta rotina é extremamente rápida, tornando conveniente a sua utilização como um passo preliminar em processos de filtragem envolvendo mais de um tipo de filtro, como temos feito para os dados obtidos no monitoramento de íons em função do tempo pelo programa SIMTIM.

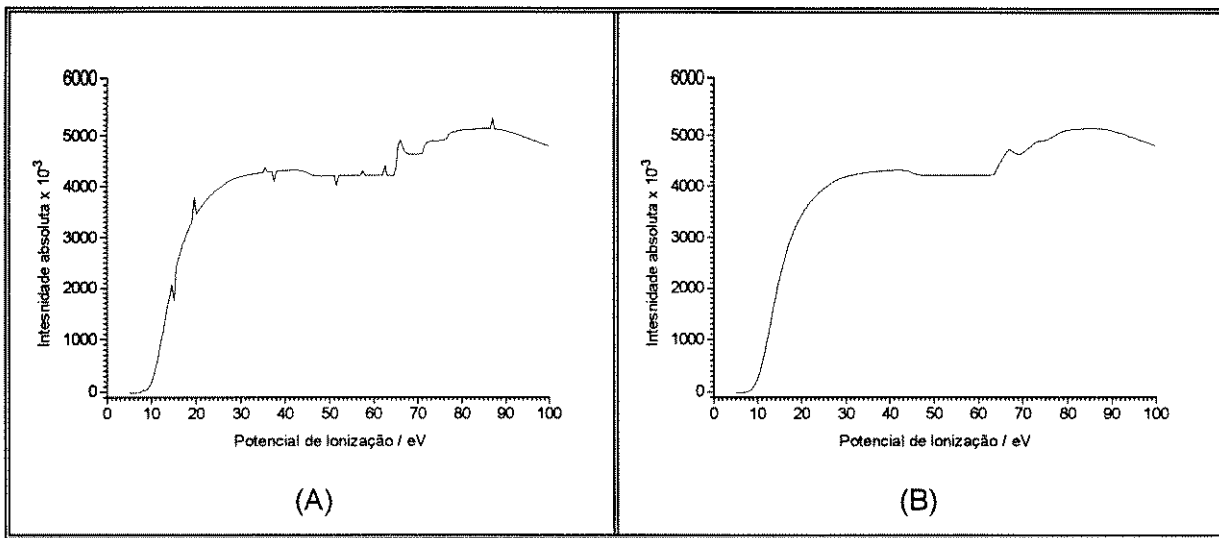


Figura VI.4: Sinais correspondentes ao monitoramento de íons totais em função do potencial de ionização obtidos com o programa SIMTIM: (A) Antes do procedimento de filtragem e (B) após a filtragem sucessiva por mediana móvel e média móvel, ambas com $w = 3$.

VI.5. CONCLUSÃO

O filtro por mediana móvel tem se mostrado eficiente na eliminação de *outliers* com pequena deformação do sinal analítico. Uma vez que bons resultados são obtidos mesmo com intervalos de modelagem pequenos, o método torna-se bastante rápido e pode ser usado como um passo preliminar em processos de filtragem que envolvam outros tipos de filtros digitais.

VI.6. BIBLIOGRAFIA

1. Lago, C. L.; Juliano, V. F.; Kascheres, C. *Anal. Chim. Acta* **1995**, 310, 281-288.
2. Rousseeuw, P. J. *J. Chemom.* **1991**, 5, 1-20.
3. Vankeerberghen, P.; Vandenbosch, C.; Smeyers-Verbeke, J.; Massart, D. L. *Chemom. Intell. Lab. Syst.* **1991**, 12, 3-13.
4. Phillips, G. R.; Eyring, E. R. *Anal. Chem.* **1983**, 55, 1134-1138.
5. Massart, D. L.; Kaufman, L.; Rousseeuw, P. J.; Leroy, A. *Anal. Chim. Acta.* **1986**, 187, 171-179.
6. Yuzhu, H.; Smeyers-Verbeke, J.; Massart, D. L. *Chemom. Intell. Lab. Syst.* **1990**, 9, 31-44.
7. Savitzky, A.; Golay, M. J. E. *Anal. Chem.* **1964**, 36, 1627-1639.
8. Rijswick, M. H. J. *Chromatographia* **1974**, 7, 491-501.
9. Bialkowski, S. E. *Anal. Chem.* **1988**, 60, 355A-361A.
10. Smit, H. C. *Chemom. Intell. Lab. Syst.* **1990**, 8, 15-27.
11. Caesar, F.; Klier, M. *Chromatographia* **1974**, 7, 526-529.
12. Xie, Y. L.; Wang, J. H. Liang, Y. Z.; Yu, R. Q. *Anal. Chim. Acta.* **1992**, 269, 307-316.
13. Phillips, G. R.; Harris, J. M. *Anal. Chem.* **1990**, 62, 2351-2357.
14. CSMA software, Princeton Instruments Inc., Trenton, NJ.
15. Davies, E. R. *Electron. Lett.* **1992**, 28, 199-201.
16. Davies, E. R. *Machine Vision: Theory, Algorithms, Practicalities*; Academic: London, 1990.
17. Davies, E. R. *Pattern Recogn. Lett.* **1988**, 7, 87.
18. Fitch, J. P.; Coyle, E. J.; Gallagher, N. C. *IEEE Trans. Acoust. Speech Signal Process.* **1985**, 33, 230.
19. Gallagher, N. C.; Wise, G. L. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, 29, 1136.
20. Tukey, J. W. *Exploratory Data Analysis*; Addison-Wesley: Menlo Park, 1971.
21. Aldus Photostyler 1.1a, Aldus Co., Seattle, WA.
22. LabVIEW2, National Instruments, Austin, TX.
23. Havlicek, J. P.; Katz, G. R.; McKeeman, J. C. *Electron. Lett.* **1992**, 28, 1258-1260.
24. Weisman, A. D.; Dougherty, E. R.; Mizes, H. A. Miller, R. J. D. *J. Appl. Phys.* **1992**, 71, 1565-1578.

25. Bauer, P. H.; Sartori, M. A.; Bryden T. M. *IEEE Trans. Signal Process.* **1992**, 40, 1068-1078.
26. Astola, J.; Heinonen, P.; Neuvo, Y. *IEEE Trans. Circ. Systems* **1989**, 36, 1430-1438.
27. Davies, E. R. *J. Mod. Optics* **1992**, 39, 103-113.
28. Kendall, M. G.; Stuart, A. *The Advanced Theory of Statistics*, vol. 1, 3th edn.; Charles Griffin: London, 1969, p.328-329.
29. Lago, C. L.; Kascheres, C. *Lab. Microcomp.* **1993**, 12, 1-4.
30. Juliano, V. F.; Eberlin, M. N.; Kascheres, C.; Lago, C. L. *Lab. Microcomp.* **1994**, 13 (2-4), 83-87.

⌚ VI.A.1. APÊNDICE: PREDIÇÃO DO EFEITO DE CORTE (*CUT-OFF*)

Ao contrário da média movel, a mediana móvel apresenta um comportamento diferente com respeito à derivada da função que descreve o conjunto de dados. Primeiro, vamos analisar o comportamento em regiões que podem ser consideradas monotônicas.

O tamanho da janela w é composto por duas margens simétricas com n elementos, então $w = 2n+1$. Para um conjunto de dados com w elementos, centrados no ponto j , e que possam ser descritos por uma função matemática, o conceito de monotonicidade pode ser entendido como:

$$y_i \leq y_{i+1} \text{ para todos } i \text{ em } \{j-n, \dots, j+n-1\} \quad (\text{VI.A.1})$$

ou

$$y_i \geq y_{i+1} \text{ para todos } i \text{ em } \{j-n, \dots, j+n-1\} \quad (\text{VI.A.2})$$

No caso, os dados estão prontamente ordenados e, por consequência, a mediana é simplesmente o valor central y_j , porque existem n valores maiores e n menores. Por esta razão não há alargamento dos picos filtrados pela mediana móvel.

Quando a função deixa de ser monotônica, a ordenação mencionada não existe mais e o que acontece pode ser melhor entendido pelo exemplo da figura VI.5. Neste caso, vamos considerar o pico simétrico, um intervalo de aquisição constante Δx e a amostragem do valor máximo. Percorrendo o vetor de dados y com a mediana móvel (onde o vetor v é uma imagem do vetor y , tal que v_0 é igual ao valor de y a ser analisado, e z é o vetor v ordenado), inicialmente existe uma região monotônica, onde v_0 é igual a z_0 . Isto ocorre até que o ponto máximo $y_{\max}$ seja igual a v_n . No passo seguinte, v_n é igual v_{n-2} e v_{n-1} é igual a $y_{\max}$. Uma vez que $v_n > v_0$, o número de valores maiores que v_0 igual ao número de valores menores. Assim, a mediana z_0 continua sendo v_0 . Esta situação permanece até $v_n = v_0$ que é o valor y_f . No próximo passo, v_{-n} , que é menor que y_f , é excluído. Contudo o novo valor de v_n é menor que y_f , então y_f continua sendo a mediana. Isto continua acontecendo até $v_{-n} = v_0 = y_f$. A partir daí, retornamos à situação na qual a mediana é igual a v_0 .

Para ambos $v_n = v_0$ e $v_{-n} = v_0$, existe um intervalo de $n+1$ pontos, no qual o ponto máximo $y_{\max}$ é o ponto central $v_{n/2}$ ou $v_{-n/2}$. Entretanto, y_f corresponde ao valor $y_{\max-d} = y_{\max+d}$, onde $d = n/2$.

Uma vez que n é par, d é um número inteiro. Se n for ímpar, y_f será determinado por $v_0 = v_{n+1}$ ou $v_0 = v_{-(n+1)}$ e assim $d = (n+1)/2$. Para qualquer valor de n temos:

$$d = \text{Int}\left(\frac{n+1}{2}\right)$$

(VI.A.3)

que é equivalente à equação VI.3. Para um pico representado pela função $f(x)$, o efeito de corte pode ser generalizado como:

$$\frac{y_f}{y_{\max}} = \frac{f(x_{\max+d})}{f(x_{\max})}$$

(VI.A.4)

onde $x_{\max}$, $y_{\max}$ representam um máximo para a função. O mesmo raciocínio pode ser aplicado para entender o efeito de corte para um vale no lugar de um pico.

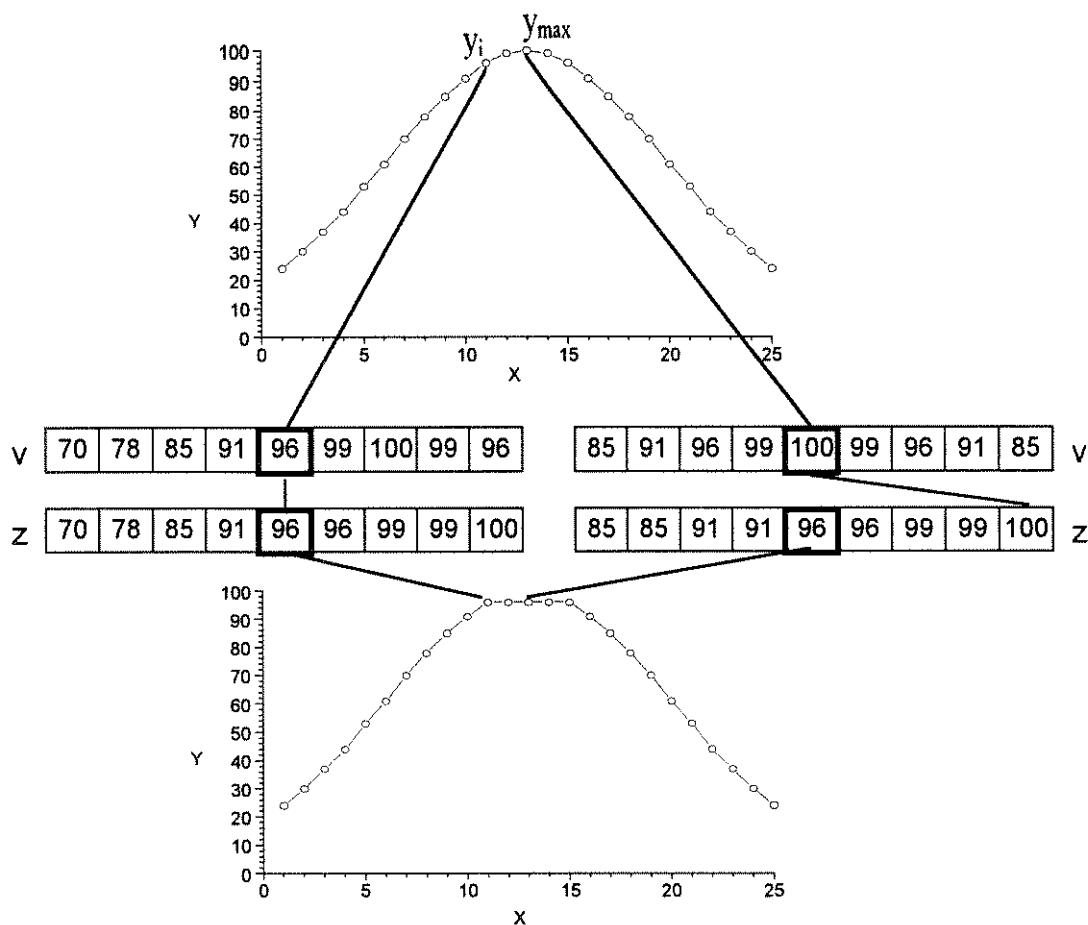


Figura VI.5: Efeito de corte. O vetor v é uma imagem de uma região com 9 pontos de y e z é o mesmo vetor v após a ordenação. Aplicando o mesmo raciocínio para outros pontos na região de corte, podemos ver que a mediana é sempre 96.

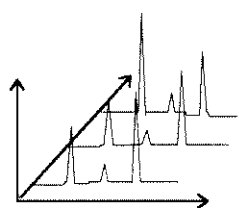
⌚ VI.A.2. APÊNDICE: LISTAGEM DA ROTINA DE FILTRAGEM DIGITAL POR MEDIANA MÓVEL COM $W = 3$

```

Procedure MedianaMovel3(var vo : TEspectro; n : Word);
var
  i : Integer;
  sai, entra, pilha : LongInt;
  d : Array[-1..1] of LongInt; { A posicao 0 e' a mediana. }
  pronto : Boolean;
Begin
  If n < 3 then Exit;      { Devem existir pelo menos 3 dados. }
  For i := -1 to 1 do d[i] := vo[i+2];
  pronto := TRUE;
  Repeat
    pronto := TRUE;
    For i := -1 to 0 do
      If d[i] > d[i+1] then
        begin
          pronto := False;
          pilha := d[i];
          d[i] := d[i+1];
          d[i+1] := pilha;
        end;
    Until pronto;
    vf[1] := d[0];
    vf[2] := d[0];
    For i := 3 to n - 1 do
      begin
        sai := vo[i - 2];
        entra := vo[i + 1];
        If sai = d[-1] then
          If entra > d[0] then
            If entra > d[1] then
              begin
                d[-1] := d[0];
                d[0] := d[1];
                d[1] := entra;
              end
            else
              begin
                d[-1] := d[0];
                d[0] := entra;
              end
            end;
          vf[i] := d[0];
        end;
      vf[n] := vf[n-1];
      For i := 1 to n do
        vo[i] := vf[i];
      End;
    
```

Obs: TEspectro = Array[1..10000] of LongInt;

CAPÍTULO VII



VII. APLICAÇÕES

Embora seja possível escrever um longo livro demonstrando todas as aplicações da Espectrometria de Massas Sequencial em um Aparelho Pentaquadrupolar, neste capítulo iremos mostrar algumas das aplicações do sistema de dados (*software + hardware*) desenvolvido. Este sistema tem sido usado em estudos de reações íon-molécula, identificação da composição elementar e da estrutura de substâncias, estudos analíticos envolvendo compostos orgânicos dissolvidos em água e outros.

VII.1. PROGRAMA PENTA2D

Este programa tem sido utilizado para obtenção de dados de experimentos MS^1 , MS^2 e MS^3 2D em estudos envolvendo reações íon-molécula, fragmentações induzidas por colisão e outros. As figuras de VII.1 a VII.3 mostram exemplos de experimentos que têm sido executados no aparelho com este programa.

A figura VII.1 mostra um espectro parcial MS^1 convencional de uma amostra de perfluorotributilamina (PFTBA), para o qual nenhum comentário adicional é necessário.

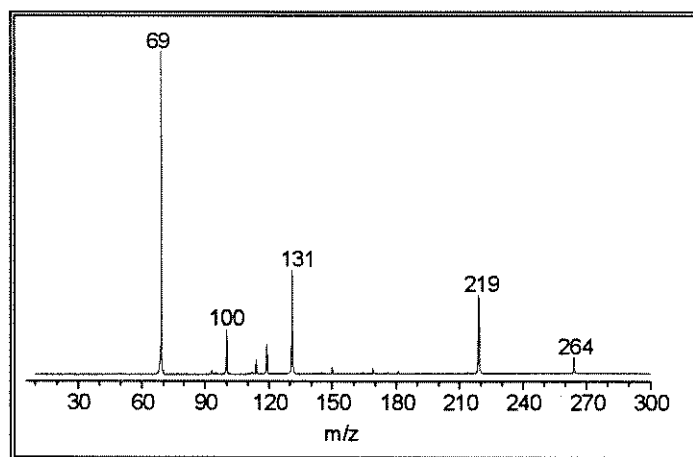


Figura VII.1: Espectro parcial MS^1 convencional de PFTBA. Os dados são arquivados em ASCII de modo que qualquer programa gráfico pode ser utilizado para gerar o espectro.

A figura VII.2 mostra o espectro MS² 2D da reação íon-molécula entre o íon acílio m/z 43 proveniente da 2-butanona com a molécula de 1,3-dioxolano (massa molecular 74 g/mol). O sinal em m/z 73 corresponde ao processo de transferência de carga seguido de perda de um radical de hidrogênio. O sinal em m/z 75 corresponde à protonação da molécula de 1,3-dioxolano e, finalmente, o sinal em m/z 87 corresponde ao produto de cicloadição, chamada de adição de oxirano.

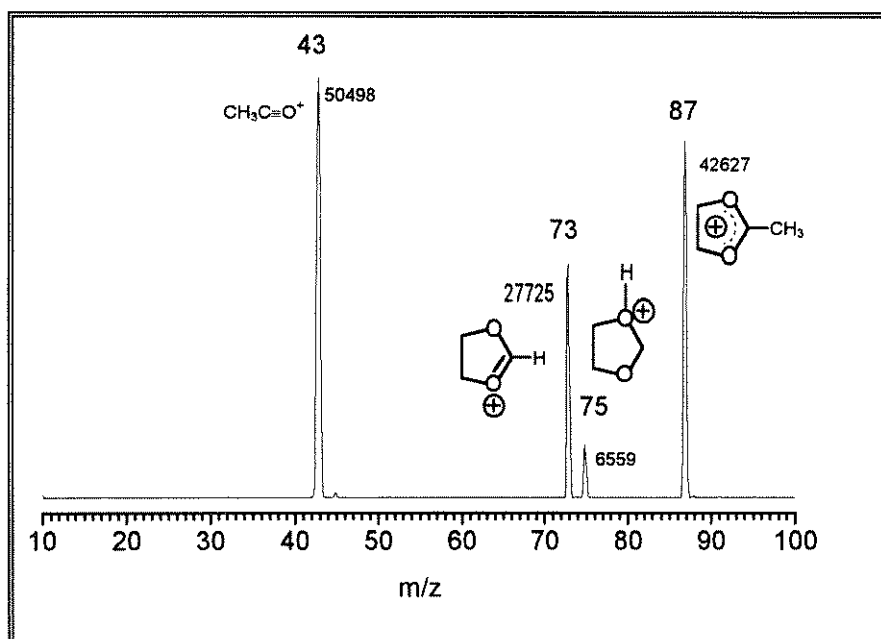


Figura VII.2: Espectro MS² 2D da reação íon-molécula entre o íon m/z 43 proveniente da 2-butanona com a molécula de 1,3-dioxolano (massa molecular 74 g/mol).

Como terceiro exemplo, a figura VII.3 mostra três espectros MS³ 2D originados de um processo consecutivo de ganho neutro, através de uma reação íon-molécula, seguido de perda neutra, através de condições de dissociação. Este método tem sido utilizado como um teste específico para identificação de íons acílios em fase gasosa.^[1] Neste processo inédito, os três quadrupolos analisadores de massa são varridos simultaneamente com uma relação de massa pré-definida entre eles. A aplicação deste modo de varredura para todos os íons produzidos a partir de etil-vinil-cetona, 2-butanona e acetofenona produz os espectros mostrados nesta figura.

Como esperado para a etil-vini-cetona (figura VII.3a), somente os íons acílios $\text{CH}_2=\text{CH}-\text{C}\equiv\text{O}^+$ (m/z 55) e $\text{CH}_3\text{CH}_2\text{C}\equiv\text{O}^+$ (m/z 57) respondem ao teste de ganho/perda neutra consecutiva de 44 u e seus sinais aparecem no espectro.

Da mesma forma, para a 2-butanona, somente os íons acílios $\text{CH}_3\text{C}\equiv\text{O}^+$ (m/z 43) e $\text{CH}_3\text{CH}_2\text{C}\equiv\text{O}^+$ (m/z 57) aparecem no espectro da figura VII.3b.

Já para a acetofenona, além dos íons acílios $\text{CH}_3\text{C}\equiv\text{O}^+$ (m/z 43) e $\text{PhC}\equiv\text{O}^+$ (m/z 105), o íon com m/z 77 também responde ao teste e aparece junto aos outros dois no espectro da figura VII.3. No entanto, um íon com tal massa não pode ser associado a nenhum íon acílio. Experimentos posteriores mostraram que o cátion fenil (Ph^+), o fragmento de m/z 77 mais provável da acetofenona, bem como seus derivados oxigenados e sulfurados, chamados de pirílio (m/z 81) e tiopirílio (m/z 97), também respondem ao teste de ganho/perda neutra consecutiva de 44 u.^[2]

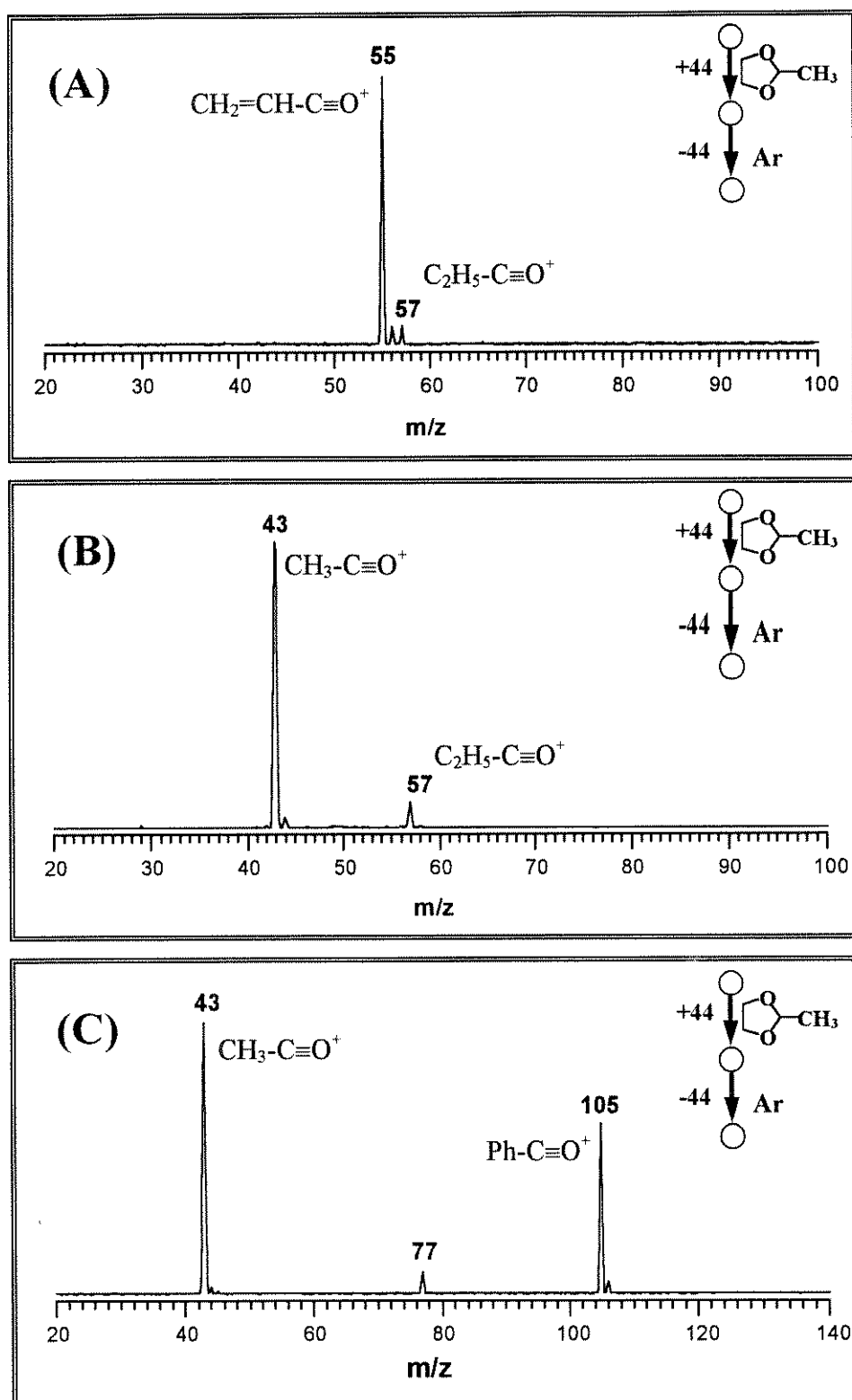


Figura VII.3: Espectros MS^3 2D de ganho/perda neutra consecutiva de 44 u aplicados para os íons produzidos a partir de (A) etil-vinil-cetona, (B) 2-butanona e (C) acetofenona. Os espectros foram obtidos através da varredura simultânea dos três quadrupolos analisadores de massa.

VII.2. PROGRAMA PENTA3D

Este programa tem sido utilizado toda vez que se deseja investigar todo o conjunto de íons produtos e precursores de um sistema em estudo. Neste caso o programa serve como uma ferramenta essencial, pois permite uma investigação detalhada através das extrações de espectros de massas MS^1 e MS^2 2D do domínio MS^2 3D do experimento.

A figura VII.4 mostra um exemplo de análise de compostos orgânicos dissolvidos em água onde, para a introdução da amostra, foi utilizado a técnica de MIMS.^[6] A ionização química permite uma correta análise qualitativa da mistura visto que somente um íon, que é a molécula protonada, é formado para cada componente individual da mistura, conforme é visto ao longo da linha tracejada na figura VII.4a, onde $m/z(Q1) = m/z(Q3)$.

Informações estruturais podem ser obtidas pelo estudo da química de dissociação de cada constituinte da mistura, as quais não poderiam ser obtidas a partir de um simples experimento CI em função de não serem produzidos fragmentos das moléculas protonadas.

Através do eixo horizontal Q3 no espectro 3D, ou na extração dos espectros 2D de produtos sequenciais (figuras VII.4b - VII.4d), são mostrados os fragmentos associados com as moléculas protonadas. A 2-pentanona (m/z 87) se fragmenta extensivamente para m/z 45 através da perda de ceteno ($CH_2=C=O$), o tolueno protonado (m/z 93) perde quase que somente uma molécula de hidrogênio para produzir $C_7H_7^+$ (m/z 91), enquanto o benzaldeído protonado (m/z 107) perde uma molécula neutra de CO para produzir, exclusivamente, benzeno protonado (m/z 79). É importante notar a qualidade de informação estrutural provida por estes padrões de fragmentação.

A projeção dos dados sobre o eixo de Q1 (figura VII.4e) permite a reconstrução de um simples espectro de massas 2D por CI, no qual as abundâncias corrigidas de cada molécula protonada pode ser usada para uma análise quantitativa.

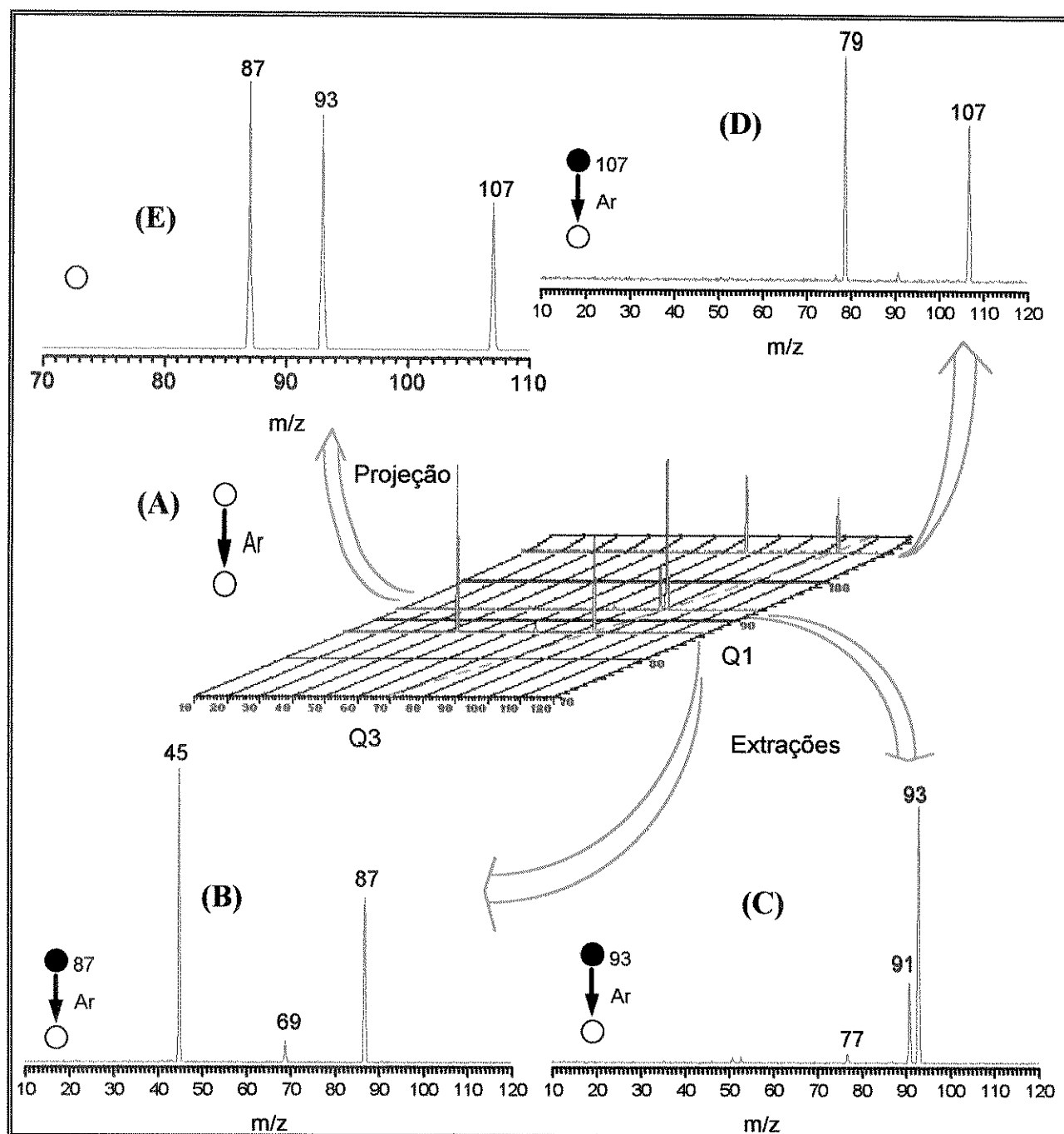


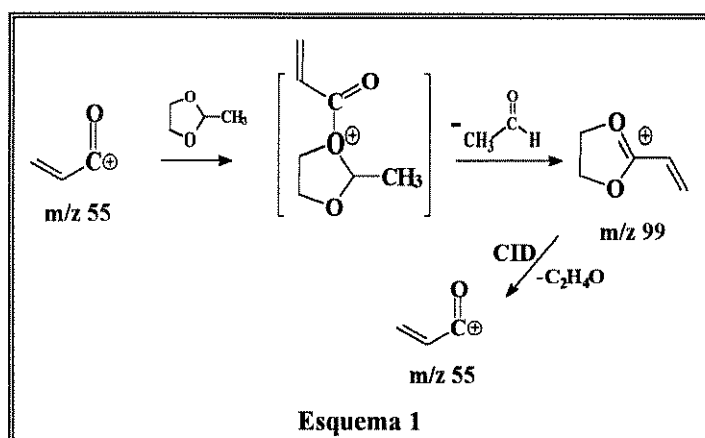
Figura VII.4: (A) Espectro MS² 3D de uma solução aquosa de 2-pentanona, tolueno e benzaldeído utilizando a técnica de MIMS. A ionização química foi feita com metano e as moléculas protonadas foram submetidas, em q2, à colisões com argônio a 15 eV. As massas selecionadas em Q1, correspondendo às moléculas protonadas são mostradas ao longo da linha tracejada, enquanto que seus produtos de dissociação correspondentes são mostrados ao longo do eixo de Q3. Espectros de produtos sequenciais MS² 2D extraídos: (B) 2-pentanona protonada, (C) tolueno protonado e (D) benzaldeído protonado. (E) Espectro CI 2D da mistura reconstruído a partir dos dados do domínio MS² 3D.

VII.3. PROGRAMA PENTA4D

Os espectros MS^3 4D^[3], embora inéditos, não têm uma aplicação direta como os demais em função da limitação na visualização dos dados. Assim, de um domínio 4D, sempre iremos extrair dados para produzir espectros 3D e 2D. Graças ao procedimento de varredura prévia, implementado no software de aquisição, um espectro 4D é adquirido em poucos minutos.

A figura VII.5 mostra o primeiro espectro 4D obtido para ciclopentanona. O controle do computador permite a seleção de todos os íons de ciclopentanona gerados na fonte de ionização. Um íon a cada vez é selecionado em Q1 e submetido à reação íon-molécula com 2-metil-1,3-dioxolano com uma energia de colisão próxima de zero em q2. Os íons produtos formados em tais reações são, um a um, selecionados em Q3 e submetidos a dissociação induzida por colisão (CID) com argônio a 15 eV em q4, enquanto Q5 analisa cada fragmento gerando todos os espectros de dissociação individualmente. Os dados 4D apresentados na figura VII.5a (o quarto eixo de abundância não pode ser mostrado) contém, então, os produtos das reações íon-molécula e suas abundâncias relativas de todos os íons da ciclopentanona, cada um diretamente associado com seus produtos de dissociação. Evidentemente, informações mais específicas, parciais e melhores para visualizar que os dados 4D, podem ser obtidas pela extração de dados de menor dimensionalidade, como exemplificado na figura VII.5b. Este espectro 3D foi obtido pela extração dos dados contidos no plano onde $m/z(Q1) = 55$ e, por isso, apresenta todos os dados relacionados especificamente à seleção do íon de m/z 55 em Q1. Isto corresponde ao modo de varredura no qual Q1 é fixado em m/z 55 enquanto Q3 e Q5 são varridos sequencialmente. Uma vista detalhada e completa de todo o processo de reação íon-molécula para o íon de m/z 55 é, então, apresentada neste espectro 3D. Ao longo da linha tracejada diagonal, na qual massas iguais são selecionadas em Q3 e Q5, o íon reagente sobrevivente (m/z 55) e todos os seus produtos de reação com 2-metil-1,3-dioxolano são mostrados (m/z 73, 87, 89 e 99). Por outro lado, os fragmentos obtidos por CID de cada produto individualmente selecionado, pode ser visto ao longo do eixo Q5. O íon de m/z 55 tem uma estrutura de íon acílio α - β insaturado $CH_2=CH-C^+=O$ ^[4] e reage extensivamente com 2-metil-1,3-dioxolano através de uma reação formal de adição de oxirano (44 u). Esta reação é diagnóstica para íons acílio, como comentado anteriormente, e produz um íon 1,3-dioxolânio em m/z 99 (esquema 1). Como pode ser visto na figura VII.5b, o produto de m/z 99 fragmenta extensivamente através da perda de molécula neutra de C_2H_4O

(44 u) em um processo que, provavelmente, regenera o íon reagente de m/z 55. A figura VII.5b também mostra que o 2-metil-1,3-dioxolano protonado (m/z 89) fragmenta extensivamente para m/z 45, enquanto que o íon 2-metil-1,3-dioxolânio (m/z 87), provavelmente formado pela transferência de hidreto, fragmenta quase que exclusivamente por perda de C_2H_4O produzindo o íon em m/z 43. É importante notar o fato de que a identidade do íon reagente também pode ser investigada pelos dados apresentados no espectro 3D. O íon m/z 55 mostra também uma extensiva fragmentação para m/z 27 por perda neutra de CO, exatamente como esperado para sua estrutura de íon acílio α - β insaturado $CH_2=CH-C^+=O$.



Posterior extracões do espectro 3D ao longo do eixo Q5, com Q3 fixado, produzem espectros MS^3 2D de produtos sequenciais, como exemplificado para o íon produto m/z 99 na figura VII.5c e para o íon reagente na figura VII.5d. Estes espectros mostram separadamente os fragmentos de cada produto da reação íon-molécula e, adicionalmente, os fragmentos do íon reagente, os quais podem ser usados para atribuições estruturais.

Projeções contra o eixo de Q5 neste espectro 3D permite reconstruir o espectro de produtos MS^2 2D do íon m/z 55 (figura VI.5e). Isto é conseguido pela soma das abundâncias individuais dos íons sobreviventes com todos os seus fragmentos. Neste espectro reconstruído, o efeito da diminuição das abundâncias relativas dos íons produtos causada pela fragmentação em q4 é cancelada e, assim, são obtidos os rendimentos corretos dos produtos da reação íon-molécula. As abundâncias reconstruídas devem ser similares à aquelas obtidas de um espectro MS^2 de produtos registrado diretamente, como é realmente observado (espectro não mostrado).

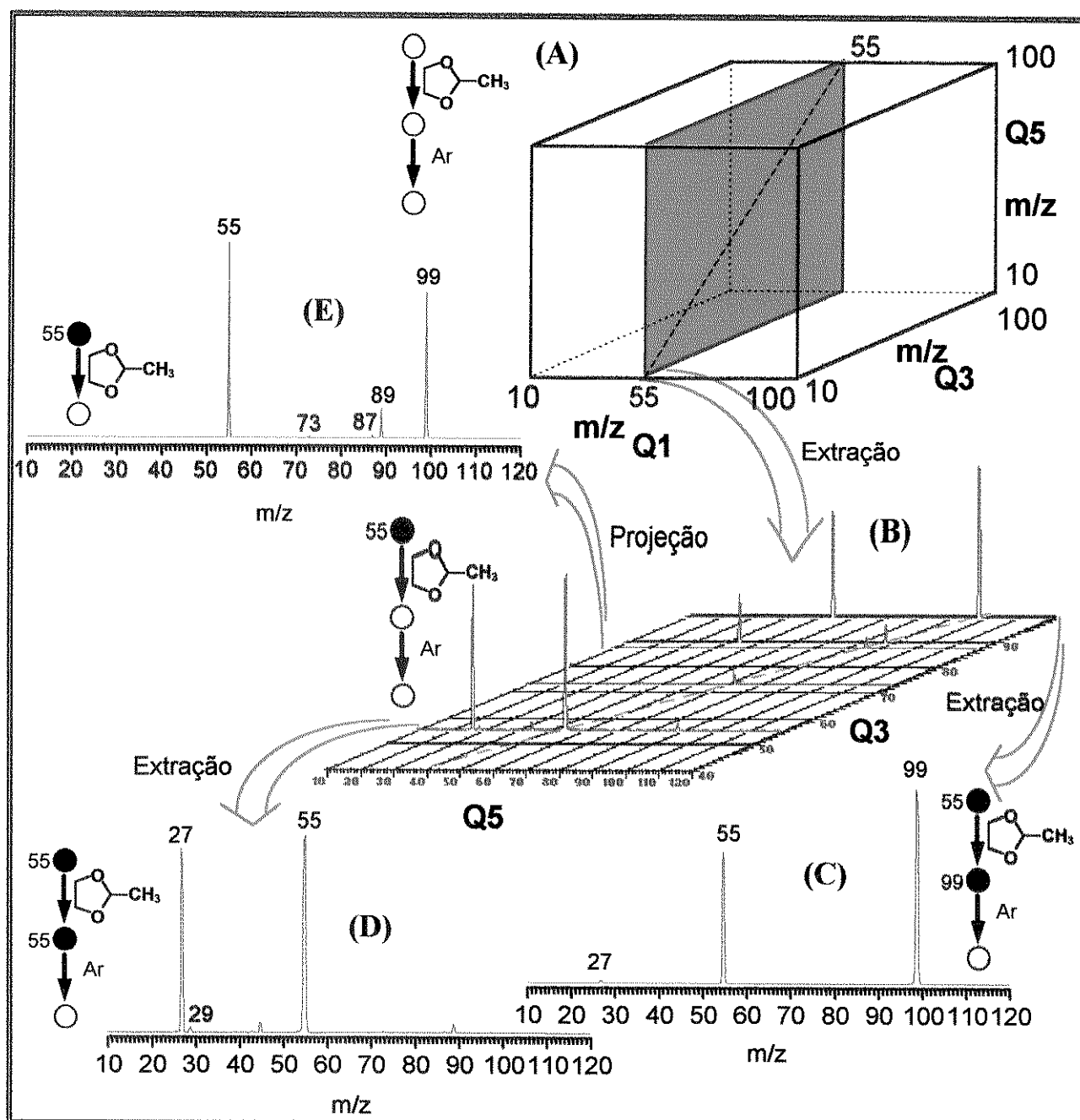


Figura VII.5: (A) Primeiro espectro MS^3 4D (eixo de abundância não incluído) obtido pela varredura sequencial dos três analisadores de massas, o qual contém o domínio de dados MS^3 completo para os íons da ciclopentanona. (B) Espectro MS^3 3D extraído do espectro 4D ao longo do plano que corta o eixo de $Q1$ em m/z 55. Todos os produtos da reação do íon m/z 55 com 2-metil-1,3-dioxolano são mostrados ao longo da linha tracejada, na qual massas iguais de $Q3$ e $Q5$ são selecionadas, enquanto que seus fragmentos correspondentes, através da colisão com argônio a 15 eV em $q4$, são mostrados ao longo do eixo horizontal de $Q5$. Espectros de produtos sequenciais MS^3 2D extraídos do espectro 3D: (C) para o produto em m/z 99 e (D) para o íon reagente sobrevivente em m/z 55. (E) Espectro de produtos MS^2 2D do íon m/z 55 reconstruído através da projeção dos sinais do espectro 3D sobre o eixo de $Q3$.

A figura VII.6 mostra outro espectro MS^3 2D extraído do espectro 4D. Os dados foram extraídos ao longo da linha definida pela intersecção de dois planos descritos pelas equações $m/z(Q3) = m/z(Q1) + 44$ e $m/z(Q5) = m/z(Q3) - 44$ (figura VII.6a). O espectro resultante (figura VII.6b) corresponde ao primeiro espectro MS^3 adquirido pela método de varredura de ganho/perda neutra consecutiva, comentado anteriormente, o qual é derivado da varredura de perda neutra consecutiva anteriormente implementado.^[5] Em tal espectro é esperado aparecer somente os sinais dos íons da ciclopentanona que reagem com 2-metil-dioxolano em q2, dando origem a produtos correspondendo ao ganho de 44 u, e que, posteriormente, se fragmentam pela colisão com argônio em q4 perdendo 44 u. Como pode ser visto na figura VII.6b, o íon acílio em m/z 55 (e a mistura com os isótopos de ^{13}C e 2H em m/z 56) sofre o ganho neutro e a perda neutra consecutiva de 44 u e, por isso, seu sinal aparece no espectro. O fato de aparecer somente este sinal no espectro é devido a este ser o único íon acílio formado a partir da ciclopentanona.^[4]

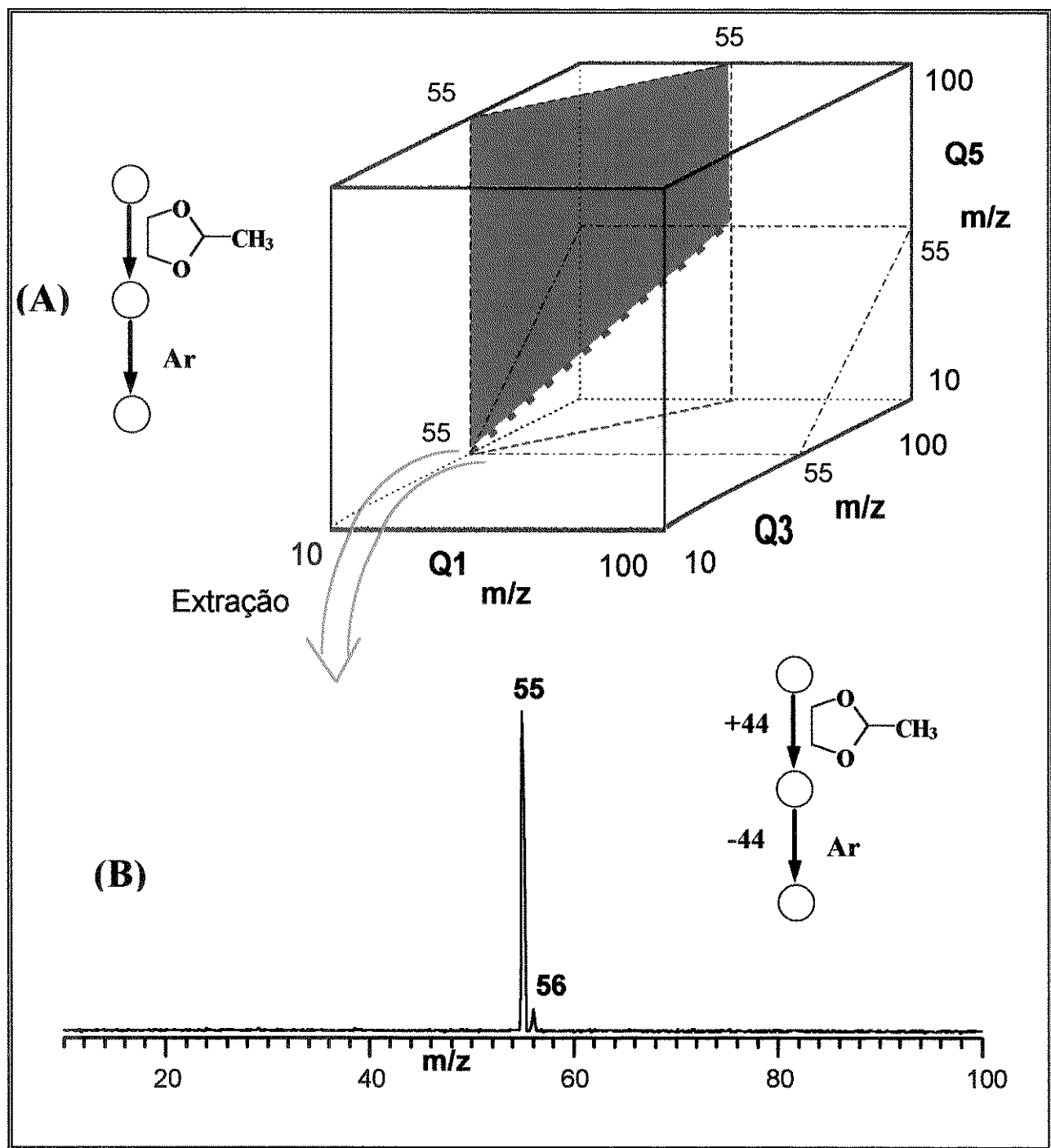


Figura VII.6: (A) Espectro MS³ 4D contendo o domínio MS³ completo para os íons da ciclopentanona, mostrando os dois planos descritos pelas equações $m/z(Q3) = m/z(Q1) + 44$ e $m/z(Q5) = m/z(Q3) - 44$. (B) Espectro MS³ 2D de ganho/perda neutra consecutiva de 44 u, o qual foi extraído do espectro 4D ao longo da linha tracejada definida pela intersecção dos dois planos. Somente o íon acílio de m/z 55 (e seu isótopo de m/z 56) responde ao teste de ganho/perda consecutiva de 44 u.

VII.4. PROGRAMA SIMTIM

Este programa, inicialmente feito com o propósito de adquirir dados somente para a técnica de MIMS, sofreu várias alterações desde a sua primeira versão, dando origem a um programa que pode ser utilizado para muitos propósitos tais como:

- Acompanhamento de processos cinéticos, como o acompanhamento da formação de produtos ou a decomposição de uma substância através do monitoramento de íons em função do tempo;
- Levantamento de curvas analíticas através do monitoramento de íons selecionados em função da concentração;
- Estudos físico-químicos, como potencial de aparecimento de picos, através da varredura da energia de ionização dos elétrons.

A figura VII.7 mostra os dados obtidos de um ensaio de monitoramento de íons em função da energia de ionização. Neste caso pretende-se utilizar a varredura de energia de ionização como uma ferramenta auxiliar na interpretação do processo de fragmentação, eliminação de contaminação do analito de interesse e também como auxiliar na eliminação da contaminação M-X na interpretação de padrões isotópicos.^[7] As figuras VII.7a e VII.7c mostram o gráfico de TIC vs energia de ionização, enquanto as figuras VII.7b e VII.7d mostram as abundâncias dos picos em m/z 84 e m/z 86 do CH_2Cl_2 e dos picos em m/z 84 e m/z 85 do C_6H_{12} vs energia de ionização, respectivamente.

Outro exemplo de aplicação do programa SIMTIM pode ser visto na figura VII.8. Nesta figura é mostrado o sinal obtido através do monitoramento de íon selecionado em função do tempo aplicado ao íon de m/z 123 (íon molecular - nitrobenzeno) em um estudo para verificar a linearidade e reprodutibilidade no monitoramento de compostos orgânicos dissolvidos em água por uma membrana de silicone através da técnica de MIMS.^[6]

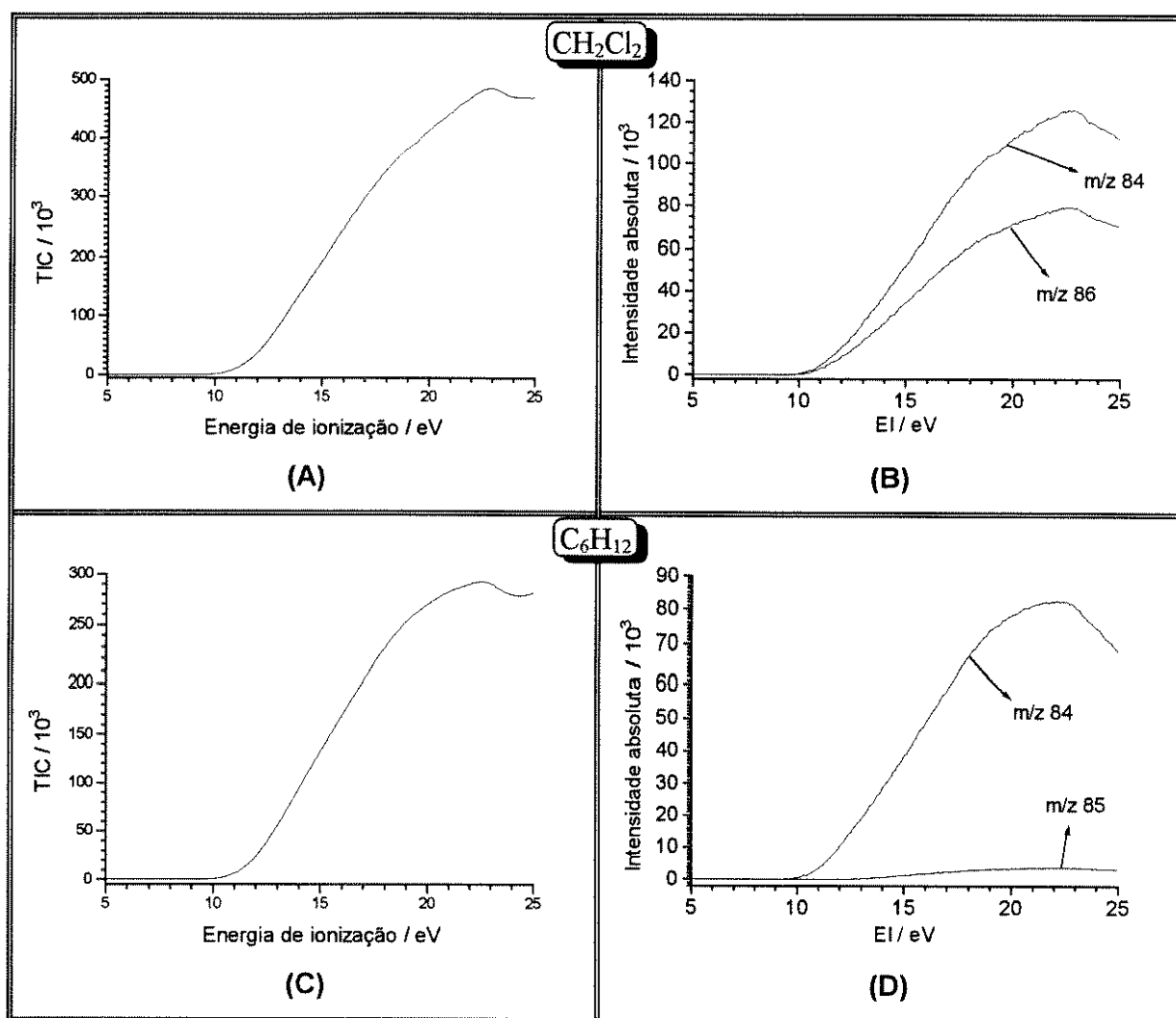


Figura VII.7: Monitoramento de íons em função da energia de ionização: (A) Corrente iônica total e (B) abundância dos picos m/z 84 e m/z 86 do CH_2Cl_2 . (C) Corrente iônica total e (D) abundância dos picos m/z 84 e m/z 85 do C_6H_{12} .

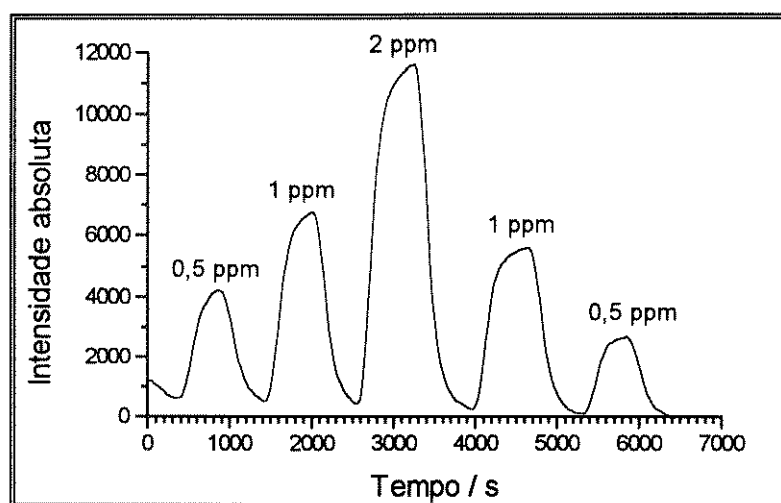


Figura VII.8: Estudo da linearidade e reprodutibilidade na resposta do sinal para o nitrobenzeno obtido através da introdução por uma membrana de silicone na técnica de MIMS.

VII.5. BIBLIOGRAFIA

1. Eberlin, M. N.; Cooks, R. G. *Org. Mass Spectrom.* **1993**, 28, 679.
2. Gozzo, F. C.; Eberlin, M. N. Resultados não publicados.
3. Juliano, V. F.; Gozzo, F. C.; Eberlin, M. N.; Kascheres, C.; Lago, C. L. *Anal. Chem.* **1996**, 00,00.
4. Budzikiewicz, H.; Djerassi, C.; Williams, D. H. *Mass Spectrometry of Organic Compounds*; Holdem-Day: San Francisco, 1967.
5. Schwartz, J. C.; Wade, A. P.; Enke, C. G.; Cooks, R. G. *Anal. Chem.* **1990**, 62, 1809.
6. Maria Anita Mendes, Tese de Doutorado em andamento, IQ - UNICAMP, sob orientação do Prof. Dr. Marcos N. Eberlin.
7. Maria Carolina B. Moraes, Dissertação de Mestrado em andamento, IQ - USP, sob orientação do Prof. Dr. Claudimir L. do Lago.

CAPITULO VIII



VIII. AVALIAÇÃO GLOBAL E CONCLUSÕES

A avaliação global que fazemos do trabalho é muito positiva, na medida que julgamos ter alcançado os objetivos ido um pouco mais além. A contribuição dada a área de desenvolvimento de hardware e, principalmente, software para espectrometria de massas sequencial foi significativa, permitindo a obtenção de dados de experimentos MS³ com rapidez e com uma qualidade muito boa.

Embora dois grupos de pesquisa tenham noticiado a existência de dois outros aparelhos pentaquadrupolares, somente tem-se notícia do aparelho em uso pelo grupo de pesquisa do professor R. G. Cooks da Universidade de Purdue, Indiana, EUA. Com isso, o nosso equipamento é o segundo no mundo em pleno funcionamento e todo esforço por nós empregado no interfaceamento foi recompensado pela originalidade no controle simultâneo dos três quadrupolos e dos métodos empregados na varredura de íons para obter espectros tri e tetradimensionais, do circuito “aprisionador” e das possibilidades do programa SIMTIM. Além disso, atualmente os programas estão sendo executados com excelente performance no sistema operacional multitarefa OS/2 Warp da IBM, dando mais flexibilidade ao trabalho de aquisição e tratamento de dados e disponibilizando os espectros para serem acessados através da rede mundial de computadores (INTERNET).

As principais conclusões que chegamos são:

- 1) O método de filtragem digital por mediana móvel aliado ao método de integração, incorporados aos programas de aquisição, ao lado da boa transmissão dos quadrupolos, proporcionaram a obtenção de dados com excelente relação sinal/ruído.
- 2) A implementação do circuito oscilador “aprisionador” de íons aumentou o rendimento de reações íon-molécula e de fragmentações via dissociação induzida por colisão (CID), melhorando a detecção dos produtos das reações e dos fragmentos. Foi observado que a amplitude e frequência de oscilação são parâmetros – ao lado de outros como pressão e energia cinética do íon reagente – que determinam o rendimento das reações, bem como o favorecimento de umas em relação às outras.

3) O conjunto de programas desenvolvidos para aquisição e tratamento dos dados em combinação com os dois cartões de interface permitiram executar experimentos inéditos MS³ 4D e MS³ 2D de ganho/perda neutra consecutiva.

PERSPECTIVAS PARA NOVOS TRABALHOS

Como perspectivas para futuros trabalhos podemos citar alguns pontos que achamos importantes:

- Permitir o controle do potencial de ionização para íons negativos.
- Construir uma fonte de ± 200 V com alta taxa de resposta para controlar a lente de saída de q2 em função do atual comprometimento entre o sinal de controle do circuito oscilador e o sinal efetivamente aplicado.
- Acoplar um circuito discriminador (*lock-in*) entre o ADC e o ME de modo a melhorar a sensibilidade quando se utiliza o circuito oscilador.
- Controlar as tensões das lentes e *offsets* dos quadropolos através de cartões adicionais com DACs de 12 bits de modo a otimizar os processos de aquisição de dados MS³ 3D e 4D.

.-.-.-.

APÊNDICE 1
LISTAGEM DOS PROGRAMAS



PENTA2D



```

Program Penta2D;
{-----}
    Programa para aquisição de espectros bidimensionais do
    espectrômetro de massas pentaquadrupolar.
    Utiliza os métodos de programação orientada a objetos e
    programação motivada por eventos.
    Desenvolvido por Valmir Fascio Juliano como parte do
    trabalho de doutoramento no Instituto de Química - UNICAMP sob
    orientação da Profa. Dra. Concetta Kascheres e do Prof. Dr. Claudimir
    Lúcio do Lago (IQ - USP).
{-----}

{ Compilar com Stack Size = 34000 e Heap Size = 8192 }

Uses Strings, WinTypes, WinDos, WinProcs, WObjects, ACESS, Matrix,
CommDlg,
    PentaAqui, WinCrt;

{$R penta.res}

Const id_EditBox = -1;
    id_PbOK = 101;
    id_PbFim = 102;
    id_PbCancela = 103;
    id_sens = 104;
    id_GroupSens = 105;
    id_perdaganho = 106;
    cm_Salva = 1;
    cm_Recupera = 2;
    cm_Copia = 3;
    cm_Print = 4;
    cm_Calibra = 5;
    cm_Expande = 6;
    cm_Normal = 7;
    cm_Sobre = 8;
    maxdados = 10000; { Número máximo de dados no vetor de dados. }
    path = 'C:\PENTA\';
    pathdados = 'Q5ESP\';

Type
TFileName = Array [0..255] of Char;

Aplicacao = object(TApplication)
    Procedure InitMainWindow; virtual;
end;

PAquisicaoGrafico = ^TAquisicaoGrafico;
TAquisicaoGrafico = object(TWindow)
    temdados, adquirendo, ButtonDown, acabou : Boolean;
    mousexi, mousexf, mouseyi, mouseyf,
    xHi, xLo, yHi : Integer;
    Dc : HDC;
    NewPen, OldPen : HPen;
    constructor Init(AParent: PWindowsObject; ATitle: PChar);
    Function GetClassName: PChar; virtual;
    Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
    Procedure WMLButtonDown(var Msg : TMessage); virtual wm_First +
wm_LButtonDown;
    Procedure WMLButtonUp(var Msg : TMessage); virtual wm_First +
wm_LButtonUp;

```

```

    Procedure WMRButtonDown(var Msg : TMessage); virtual wm_First +
wm_RButtonDown;
    Procedure WMMouseMove(var Msg : TMessage); virtual wm_First +
wm_MouseMove;
    Procedure WMLButtonDbIClk(var Msg : TMessage); virtual wm_First +
wm_LButtonDbIClk;
end;

PAquisicao = ^TAquisicao;
TAquisicao = object(TWindow)
    PbOK, PbFim, PbCancela : PButton;
    Massaln, MassaFin, m1box, m3box, m5box,
    IncrMass, TempInt, coment, arquivo, Enion : PEdit;
    quad, ganh : PComboBox;
    senslo, sensmed, senshi : PRadioButton;
    GroupSens : PGroupBox;
    perda_ganho : PCheckBox;
    grafico : PAquisicaoGrafico;
    FileName : TFileName;
    salva_recupera : Integer;
    ok_arquivo : Boolean;
    constructor Init(AParent: PWindowsObject; ATitle: PChar);
    Function GetClassName: PChar; virtual;
    Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
    Procedure SetUpWindow; virtual;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
    Procedure Executa(var Msg : TMessage); virtual id_First + id_PbOK;
    Procedure Fim(var Msg : TMessage); virtual id_First + id_PbFim;
    Procedure Cancela(var Msg : TMessage); virtual id_First +
id_PbCancela;
    Function CanClose: Boolean; virtual;
    Procedure CMFileOpen(var Msg: TMessage);
    Procedure Salvar(var Msg : TMessage); virtual cm_First + cm_Salva;
    Procedure Recuperar(var Msg : TMessage); virtual cm_First +
cm_Recupera;
    Procedure Copia(var Msg : TMessage); virtual cm_First + cm_Copia;
    Procedure Print(var Msg : TMessage); virtual cm_First + cm_Print;
    Procedure Calibra(var Msg : TMessage); virtual cm_First + cm_Calibra;
    Procedure Expandir(var Msg : TMessage); virtual cm_First +
cm_Expande;
    Procedure Normal(var Msg : TMessage); virtual cm_First + cm_Normal;
    Procedure Sobre(var Msg : TMessage); virtual cm_First + cm_Sobre;
end;
PVetori = ^TVetori;
TVetori = Array[1..maxdados] of Integer;
PVetoriW = ^TVetoriW;
TVetoriW = Array[1..maxdados] of Word;

var
    mi, mf, maxy, massainicial, massafinal,
    expansao, expmi, expmf, maximo : Word;
    m1, m3, m5, ganh, incr, ndados : Integer;
    tempo, energion : Double;
    quadrupolo : Byte;
    Aplica : Aplicacao;
    det : PDetector;
    Q1, Q3, Q5 : PQuadrupolo;
    Ion : Plonizador;
    perdaganho : Boolean;
    nome.comentario : String;
    massa : PVetori;

```

```

Intens : PVetorW;

Procedure Aplicacao.InitMainWindow;
Begin
    MainWindow := New(PAquisicao, Init(nil,'PENTA2D')); { Inicializa o
Programa. }
    ScreenSize.X := 26;
    ScreenSize.Y := 4;
    InitWinCrt;
    WriteLn('Não feche esta janela,');
    WriteLn('minimize-a se necessario.');
```

```

End;

Constructor TAquisicaoGrafico.Init(AParent: PWindowsObject; ATitle:
PChar);
Begin
    temdados := FALSE; { Sinalizador para indicar que existem dados. }
    adquirido := FALSE; { Sinalizador para evitar re-aquisição. }
    ButtonDown := FALSE;
    xHi := 580;
    xLo := 30;
    yHi := 260;
    m1 := 0;
    m3 := 0;
    m5 := 0;
    TWindow.Init(AParent, ATitle);
    With Attr do
        begin
            Style := ws_Child OR ws_Overlapped OR ws_Visible;
            X := 315; Y := 45; W := 560; H := 260;
        end;
End;

Function TAquisicaoGrafico.GetClassName: PChar;
begin
    GetClassName := 'GraphWndClass'
end;

Procedure TAquisicaoGrafico.GetWindowClass(var AWndClass:
TWndClass);
Begin
    TWindow.GetWindowClass(AWndClass);
    (* AWndClass.hbrBackground := GetStockObject(LtGray_Brush); { Fundo
gráfico cinza.} *)
    AWndClass.Style := AWndClass.Style or cs_DblClks; { Aceita click duplo
do mouse.}
End;

Procedure Adquire(var Q : PQuadrupolo; massa : Integer; var intensidade :
Word);
Begin
    Q^.SetMassa(massa/10.0);
    If tempo < 0.029 then intensidade := det^.LeAD
    else intensidade := det^.LeADAvg;
End;

Procedure TAquisicaoGrafico.Paint(PaintDC : HDC; var PaintInfo :
TPaintStruct);
var
    NewCur,
    OldCur : HCursor;
```

```

OldFont,
NewFont : HFont;
i,j,xi,incrm,posx,posy,oy : Integer;
yi,inten : Word;
s : string[95];
sc : Array[0..94] of Char;

Function Y(i : Word) : Integer;
Begin
    Y := Round(1000.0*i/65535);
End;

Function mcorte(m : Integer) : Double;
Begin
    mcorte := 91.91 * m / (1735.0 + m);
End;

Begin { TAquisicaoGrafico.Paint }
    If not temdados then EXIT;
    If not adquirido then
        begin
            det := New(PDetector,init(1,0,ganho,tempo)); { Inicializa o detector. }
            Q1 := New(PQuadrupolo,init(1,0,0,path+'Q1.CAL')); { Inicializa o
quadrupolo. }
            Q3 := New(PQuadrupolo,init(2,0,1,path+'Q3.CAL'));
            Q5 := New(PQuadrupolo,init(1,1,2,path+'Q5.CAL'));
        end;
    NewCur := LoadCursor(0,Idc_Wait); { Sinaliza com a ampulheta. }
    OldCur := SetCursor(NewCur);
    SetTextAlign(PaintDC,TA_RIGHT);
    SetTextColor(PaintDC,RGB(0,0,255));
    SetBkMode(PaintDC,transparent);
    SetMapMode(PaintDC,mm_Anisotropic);
    SetViewportOrg(PaintDC,xLo,yHi-30);
    SetViewportExt(PaintDC,xHi-xLo,-(yHi-30));
    SetWindowExt(PaintDC,mf-mi,maxy);
    SetWindowOrg(PaintDC,mi,0);
    oy := -maxy div 50; { Deslocamento da linha base em 2%. }
    NewFont := CreateFont(Round(0.06*maxy),0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
:=
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
    OldFont := SelectObject(PaintDC,NewFont);
    incrm := 100*(1 + Trunc((mf - mi)/1000.0 - 0.49)); { Incremento da escala
de massa. }
    If (mf - mi) div 10 > 80 then posX := mi - incrm div 10
    else If (mf - mi) div 10 > 60 then posX := mi - incrm div 15
        else If (mf - mi) div 10 > 40 then posX := mi - incrm div 20
            else If (mf - mi) div 10 > 20 then posX := mi - incrm div 40
                else posX := mi - incrm div 60; { Posição x p/ intensidade. }
    Str(maxy div 10,s);
    StrPCopy(sc,s); { Numera as intensidades. }
    TextOut(PaintDC,posx,maxy,sc,StrLen(sc));
    Str(maxy div 20,s);
    StrPCopy(sc,s);
    TextOut(PaintDC,posx,maxy div 2,sc,StrLen(sc));
    TextOut(PaintDC,posx,0,'0',1);
    NewPen := CreatePen(ps_Solid,0,RGB(0,0,0));
    OldPen := SelectObject(PaintDC,NewPen);
    MoveTo(PaintDC,mi,maxy);
```

```

LineTo(PaintDC,mi,oy);      { Desenha os eixos. }
LineTo(PaintDC,mf,oy);
Str(mi div 10,s);
StrPCopy(sc,s);
posy := -maxy div 15;
TextOut(PaintDC,mi,posy,sc,StrLen(sc)); { Numera a 1a massa. }
MoveTo(PaintDC,mi,posy);
LineTo(PaintDC,mi,oy);
i := 0;
xi := mi;
Repeat
  posy := -maxy div 25;
  inc(i);
  While (xi < mi + (i + 1) * increm) and (xi < mf) do
    begin
      xi := xi + increm div 10;
      MoveTo(PaintDC,xi,posy); { Subdivisões da escala de massa. }
      LineTo(PaintDC,xi,oy);
    end;
  xi := mi + i * increm; { Evita acúmulo de sobras de arredondamento. }
  Str(xi div 10,s);
  StrPCopy(sc,s);
  posy := -maxy div 15;
  TextOut(PaintDC,xi,posy,sc,StrLen(sc)); { Numera as massas. }
  MoveTo(PaintDC,xi,posy);
  LineTo(PaintDC,xi,oy);
Until xi > mf - increm;
For i := 0 to 10 do
  begin
    yi := Round(i * maxy / 10.0);
    MoveTo(PaintDC,posx,yi); { Escala de intensidade. }
    LineTo(PaintDC,mi,yi);
  end;
SetTextAlign(PaintDC,TA_CENTER);
StrPCopy(sc,comentario);
TextOut(PaintDC,mi + (mf - mi) div 2, Round(-0.25*maxy), sc, StrLen(sc));
SelectObject(PaintDC, OldFont);
DeleteObject(NewFont);
SelectObject(PaintDC, OldPen);
DeleteObject(NewPen);
If not adquirido then
  begin
    If not perdaganho then
      begin
        Q1^.SetPoleDC(off);
        Q3^.SetPoleDC(off);
        Q5^.SetPoleDC(off);
      end;
    Case quadrupolo of
      1 : Q1^.SetPoleDC(on);
      3 : Q3^.SetPoleDC(on);
      5 : Q5^.SetPoleDC(on);
    end; { case }
    NewPen := CreatePen(ps_Solid,0,RGB(255,0,255));
    OldPen := SelectObject(PaintDC,NewPen);
    MoveTo(PaintDC,mi,0);
    maximo := 0;
    i := 0;
    Repeat
      inc(i);
      massa^i[i] := mi + (i-1) * incr;

```

```

Case quadrupolo of
  1 : begin
    If perdaganho then
      begin
        Q3^.SetMassa(massa^i[i]/10.0 + m3);
        Q5^.SetMassa(massa^i[i]/10.0 + m5);
      end
    else
      begin
        Q3^.SetMassa(mcorte(massa^i[i]));
        Q5^.SetMassa(mcorte(massa^i[i]));
      end;
    Acquire(Q1, massa^i[i],intens^i[i]);
  end;
  3 : begin
    If perdaganho then
      begin
        Q1^.SetMassa(massa^i[i]/10.0 + m1);
        Q5^.SetMassa(massa^i[i]/10.0 + m5);
      end
    else
      begin
        Q1^.SetMassa(mcorte(massa^i[i]));
        Q5^.SetMassa(mcorte(massa^i[i]));
      end;
    Acquire(Q3, massa^i[i],intens^i[i]);
  end;
  5 : begin
    If perdaganho then
      begin
        Q1^.SetMassa(massa^i[i]/10.0 + m1);
        Q3^.SetMassa(massa^i[i]/10.0 + m3);
      end
    else
      begin
        Q1^.SetMassa(mcorte(massa^i[i]));
        Q3^.SetMassa(mcorte(massa^i[i]));
      end;
    Acquire(Q5, massa^i[i],intens^i[i]);
  end;
end; { case }
LineTo(PaintDC, massa^i[i], Y(intens^i[i]));
If maximo < intens^i[i] then maximo := intens^i[i];
KeyPressed;
Until (massa^i[i] >= mf) or (i >= maxdados) or acabou;
ndados := i;
MessageBeep(MB_OK); { Emite um beep após a aquisição. }
SelectObject(PaintDC, OldPen);
DeleteObject(NewPen);
end;
NewPen := CreatePen(ps_Solid,0,RGB(255,0,0));
OldPen := SelectObject(PaintDC,NewPen);
If maximo > 0 then
  begin
    MoveTo(PaintDC,mi,Word(Round(1000.0 * intens^i[1] / maximo)));
    For i := 1 to ndados do { Traça o espectro normalizando as
intensidades. }
      begin
        inten := Word(Round(1000.0 * intens^i[i] / maximo));
        If (massa^i[i] >= mi) and (massa^i[i] <= mf) then
          LineTo(PaintDC, massa^i[i],inten);

```

```

    end;
end;
SelectObject(PaintDC,OldPen);
DeleteObject(NewPen);
If not adquirido then
begin
    Dispose(det,done); { Libera o espaço de memória ocupada pelos
ponteiros }
    Dispose(Q1,done); { detector e quadrupolos. }
    Dispose(Q3,done);
    Dispose(Q5,done);
end;
SetCursor(OldCur);
adquirido := TRUE;
Dc := PaintDc;
End;

```

```

Procedure TAquisicaoGrafico.WMLButtonDown(var Msg : TMessage);
Begin
    If not ButtonDown then
    begin
        mousexi := xLo; { Valores iniciais para a posição do mouse. }
        mousexf := xHi;

        mouseyf := 0;
        ButtonDown := TRUE;
        SetCapture(HWindow);
        Dc := GetDC(HWindow);
        mousexi := Msg.LParamLo; { Obtém a nova posição do mouse. }
        mouseyi := yHi - 30;
        MoveTo(Dc,mousexi,mouseyi);

        NewPen := CreatePen(ps_Dot,0,RGB(0,128,0)); { Linha pontilhada
verde. }
        OldPen := SelectObject(Dc,NewPen);
    end;
End;

```

```

Procedure TAquisicaoGrafico.WMLButtonUp(var Msg : TMessage);
var troca : Word;
Begin
    If ButtonDown then
    begin
        LineTo(Dc,mousexf,mouseyi); { Desenha o retângulo que define a
região que }
        LineTo(Dc,mousexf,mouseyf); { poderá ser ampliada. }
        LineTo(Dc,mousexi,mouseyf);
        LineTo(Dc,mousexi,mouseyi); { Define os valores para a expansão. }
        expansao := Word(Round(maxy * (mouseyf - yHi + 30) / (30 - yHi)));
        If expansao > 1000 then expansao := 1000;
        If expansao < 50 then expansao := 50; { intensidade mínima = 5% }
        expmi := Word(10*Trunc((mi+(mousexi - xLo) * (mf - mi))/(xHi -
xLo))/10.0));
        If expmi < massainicial then expmi := massainicial;
        expmf := Word(10*Trunc((mi+(mousexf - xLo) * (mf - mi))/(xHi -
xLo))/10.0));
        If expmf > massafinal then expmf := massafinal;
        If expmf <= expmi then
        begin
            troca := expmf;
            expmf := expmi;
            expmi := troca;
        end;
    end;

```

```

    If (expmf - expmi) < 100 then expmf := expmi + 100;
    SelectObject(Dc,OldPen);
    DeleteObject(NewPen);
    ReleaseCapture;
    ReleaseDC(HWindow,Dc);
    ButtonDown := FALSE;
end;
End;

Procedure TAquisicaoGrafico.WMRButtonDown(var Msg : TMessage);
Begin
    mi := expmi;
    mf := expmf;
    maxy := expansao;
    InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro expandido.
}
End;

```

```

Procedure TAquisicaoGrafico.WMMouseMove(var Msg : TMessage);
Begin
    If ButtonDown then With Msg do
    begin
        mousexf := Integer(LParamLo); { Captura as posições do mouse em
movimento.}
        mouseyf := Integer(LParamHi);
    end;
End;

```

```

Procedure TAquisicaoGrafico.WMLButtonDbClick(var Msg : TMessage);
Begin
    maxy := 1000;
    mi := massainicial;
    mf := massafinal;
    InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro após o
duplo click. }
End;

```

```

Constructor TAquisicao.Init(AParent: PWindowsObject; ATitle: PChar);
var
    WinDir : array [0..145] of Char;
Begin
    TWindow.Init(AParent,ATitle);
    EnableKBHandler;
    with Attr do
    begin
        X := CW_USEDEFAULT; Y := CW_USEDEFAULT; W := 950; H := 500;
        Menu := LoadMenu(HInstance,'menu_1');
    end;
    Ion := New(Plonizador,Init(1,1,1,1));

    quad :=
New(PComboBox,Init(@Self,id_EditBox,205,30,50,100,cbs_DropDownList,
0));
    quad^.Attr.Style := quad^.Attr.Style and (not cbs_Sort);
    ganh := New(PComboBox,Init(@Self,
id_EditBox,75,65,70,100,cbs_DropDownList,0));
    ganh^.Attr.Style := ganh^.Attr.Style and (not cbs_Sort);
    GroupSens := New(PGroupBox, Init(@Self, id_GroupSens,"",0,0,0,0));
    senslo := New(PRadioButton, Init(@Self, id_sens,
'B',190,65,40,20,GroupSens));
    sensmed := New(PRadioButton, Init(@Self, id_sens,
'M',230,65,40,20,GroupSens));

```

```

senshi := New(PRadioButton, Init(@Self,
id_sens,'A',270,65,40,20,GroupSens));

massain :=
New(PEdit,Init(@Self,id_EditBox,'10',205,100,40,30,4,FALSE));
massafin :=
New(PEdit,Init(@Self,id_EditBox,'300',205,135,50,30,5,FALSE));
IncrMass:=
New(PEdit,Init(@Self,id_EditBox,'0.1',205,170,40,30,4,FALSE));
Tempoint :=
New(PEdit,Init(@Self,id_EditBox,'0.0',205,205,45,30,5,FALSE));
Enion := New(PEdit,Init(@Self,id_EditBox,'70.0',205,240,45,30,5,FALSE));
perda_ganho := New(PCheckBox, Init(@Self, id_perdaganho,
",295,275,30,26, nil));
m1box := New(PEdit,Init(@Self,id_EditBox,'0',50,310,50,30,5,FALSE));
m3box := New(PEdit,Init(@Self,id_EditBox,'0',140,310,50,30,5,FALSE));
m5box := New(PEdit,Init(@Self,id_EditBox,'0',230,310,50,30,5,FALSE));
arquivo := New(PEdit,Init(@Self,id_EditBox,"",90,345,165,30,9,FALSE));
coment := New(PEdit,Init(@Self,id_EditBox,"",380,380,520,30,81,FALSE));
PbOK := New(PButton,Init(@Self,id_PbOK,'OK para
iniciar',20,380,130,30,FALSE));
PbFim := New(PButton,Init(@Self,id_PbFim,'FIM',160,380,50,30,TRUE));
PbCancela :=
New(PButton,Init(@Self,id_PbCancela,'Cancela',220,380,80,30,FALSE));
New(massa);
New(intens);
Grafico := New(PAquisicaoGrafico,Init(@Self,nil));
maxy := 1000; { Correponde ao máximo para o pico base =
100.0%}
ok_arquivo := FALSE;
StrPCopy(WinDir,path+pathdados);
SetCurDir(WinDir); { Diretório Default. }
End;

Procedure TAquisicao.GetWindowClass(var AWndClass : TWndClass);
Begin
TWndClass.GetWindowClass(AWndClass);
AWndClass.HIcon := LoadIcon(HInstance,'rack');
End;

Function TAquisicao.GetClassName: PChar;
Begin
GetClassName := 'MainWndClass'
End;

Procedure TAquisicao.SetupWindow;
Begin
TWndClass.SetupWindow;
quad^.AddString('1');
quad^.AddString('3');
quad^.AddString('5');
ganh^.AddString('1');
ganh^.AddString('2');
ganh^.AddString('5');
ganh^.AddString('10');
ganh^.AddString('20');
ganh^.AddString('50');
ganh^.AddString('100');
ganh^.SetSelIndex(4);
quad^.SetSelIndex(2);
sensmed^.SetCheck(1);
End;
```

```

Procedure TAquisicao.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
OldBrush,
NewBrush : HBrush;
NewFont,
OldFont : HFont;

Begin
NewFont := CreateFont(18,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
OldFont := SelectObject(PaintDC,NewFont);
SetTextColor(PaintDC,RGB(255,0,128));
SetBkMode(PaintDC,transparent);
SetTextAlign(PaintDC,TA_LEFT);
TextOut(PaintDC,100,5,'AQUISIÇÃO E MANIPULAÇÃO DE ESPECTROS
DO ESPECTRÔMETRO DE MASSAS PENTAQUADRUPOLAR',80);
TextOut(PaintDC,25,35,'Quadrupolo de varredura:',24);
TextOut(PaintDC,25,70,'Ganho:',6);
TextOut(PaintDC,150,70,'Sens.:',6);
TextOut(PaintDC,25,105,'Massa inicial:',14);
TextOut(PaintDC,25,140,'Massa final:',12);
TextOut(PaintDC,25,175,'Incremento de Massa:',20);
TextOut(PaintDC,25,210,'Tempo de Integração (ms):',25);
TextOut(PaintDC,25,245,'Energia de Ionização (eV):',26);
TextOut(PaintDC,25,280,'Variação de Massa: Ganho / Perda neutra:',40);
TextOut(PaintDC,25,315,'Q1',2);
TextOut(PaintDC,115,315,'Q3',2);
TextOut(PaintDC,205,315,'Q5',2);
TextOut(PaintDC,25,350,'Arquivo:',8);
TextOut(PaintDC,300,385,'Comentário:',11);
SelectObject(PaintDC,OldFont);
DeleteObject(NewFont);
End;

Procedure TAquisicao.Executa(var Msg : TMessage);
var
sc : Array[0..10] of Char;
s : string;
err : Integer;
incre, m : Double;

Procedure Mostra;
Begin
Str((massainicial / 10.0):0:0,s);
StrPCopy(sc,s);
massain^.SetText(sc);
Str((massafinal / 10.0):0:0,s);
StrPCopy(sc,s);
massafin^.SetText(sc);
Str((incr / 10.0):0:1,s);
StrPCopy(sc,s);
IncrMass^.SetText(sc);
Str(tempo:0:1,s);
StrPCopy(sc,s);
Tempoint^.SetText(sc);
arquivo^.SetText(filename);
End;
```

```
Begin
    { Adquire um espectro após confirmação. }
    maxy := 1000;
    quad^.GetSelString(sc,1);
    Val(sc,quadrupolo,err);
    ganh^.GetSelString(sc,3);
    Val(sc,ganho,err);
    If sensio^.GetCheck = 1 then Port[$318] := 1;
    If sensmed^.GetCheck = 1 then Port[$318] := 3;
    If sensshi^.GetCheck = 1 then Port[$318] := 2;
    massain^.GetText(sc,10); { Captura os valores nos boxes de edição. }
    Val(sc,m,err);
    massainicial := Round(m * 10);
    If massainicial < 10 then massainicial := 10; { Limita a massa inicial = 1 u. }
}

mi := massainicial;
massafin^.GetText(sc,10);
Val(sc,m,err);
massafinal := Round(m * 10);
If massafinal < (massainicial + 100) then massafinal := massainicial + 100;
If massafinal > 10000 then massafinal := 10000; { Limita a massa final = 1000 u. }
mf := massafinal;

incrMass^.GetText(sc,10);
Val(sc,incr,err);
incr := Round(incr * 10);
If incr < 1 then incr := 1;
Enion^.GetText(sc,10);
Val(sc,energion,err);
If energion < 5.0 then energion := 5.0
else If energion > 100.0 then energion := 100.0;
Ion^.El(energion);
Tempoint^.GetText(sc,10);
Val(sc,tempo,err);
If tempo < 0 then tempo := 0
else If tempo > 33.4 then tempo := 33.4;
m1box^.GetText(sc,10);
Val(sc,m1,err);
m3box^.GetText(sc,10);
Val(sc,m3,err);
m5box^.GetText(sc,10);
Val(sc,m5,err);
perdaganho := perda_ganho^.GetCheck = 1;
StrCopy(FileName,"");
Mostra;
InvalidateRect(Grafico^.HWindow,nil,TRUE); { Limpa o grafico. }
Grafico^.temdados := TRUE; { Garante que existem dados para serem desenhados. }
Grafico^.adquirido := FALSE; { Garante que ainda não foi adquirido. }
Grafico^.acabou := FALSE;
End;
```

```
Procedure TAquisicao.CMFileOpen(var Msg: TMessage);
const
    DefExt = 'dat';
var
    OpenFN : TOpenFileName;
    Filter : array [0..100] of Char;
    FullFileName: TFileName;
Begin
    StrCopy(FullFileName,"");
```

```
FillChar(Filter, SizeOf(Filter), #0);
StrCopy(Filter, 'Espectros unidimensionais');
StrCopy(@Filter[StrLen(Filter)+1], '*.*');
FillChar(OpenFN, SizeOf(TOpenFileName), #0);
With OpenFN do
begin
    hInstance := HInstance;
    hwndOwner := HWindow;
    lpstrDefExt := DefExt;
    lpstrFile := FullFileName;
    lpstrFilter := Filter;
    lpstrFileTitle:= FileName;
    flags := ofn_FileMustExist;
    lStructSize := sizeof(TOpenFileName);
    nFilterIndex := 1; { Índice dentro da String de filtragem em lpstrFilter. }
    nMaxFile := SizeOf(FullFileName);
end;
Case salva_recupera of
    cm_salva : ok_arquivo := GetSaveFileName(OpenFN);
    cm_recupera : ok_arquivo := GetOpenFileName(OpenFN);
end;
If ok_arquivo then arquivo^.SetText(filename);
End;
```

```
Procedure TAquisicao.Salvar(var Msg : TMessage);
var
    arq : Text;
    i : Word;
    sc : Array[0..79] of Char;
Begin
    Salva_Recupera := cm_salva;
    If Grafico^.temdados then CMFileOpen(Msg);
    If not ok_arquivo then EXIT;
    nome := StrPas(FileName);
    coment^.GetText(sc,80);
    comentario := StrPas(sc);
    Assign(arq,nome);
    {$I-} Reset(arq); {$I+}
    If IOResult = 0 then
begin
        StrPCopy(sc,'Arquivo ' + nome + ' já existe. Sobrepõe?');
        If MessageBox(HWindow,sc,'Salvar', mb_defbutton2 or mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
end;
    ReWrite(arq);
    Writeln(arq,maximo);
    Writeln(arq,comentario);
    Writeln(arq,quadrupolo);
    Writeln(arq,ganho);
    Writeln(arq,tempo:0:2);
    For i := 1 to ndados do { Grava os dados separados }
        Writeln(arq,masa^[i]/10:0:1,#9,intens^[i]); { pelo caracter TAB. }
    Close(arq);
End;
```

```
Procedure TAquisicao.Recuperar(var Msg : TMessage);
var
    arq : Text;
    m,incr : Double;
    i : Word;
    s : string;
```



```

sc : Array[0..79] of Char;
err : Integer;
Begin
  Salva_Recupera := cm_recupera;
  CMFileOpen(Msg);
  If not ok_arquivo then EXIT;
  nome := StrPas(FileName);
  Assign(arq,nome);
  {$I-} Reset(arq); {$I+}
  If IOResult <> 0 then
    begin
      StrPCopy(sc,'Arquivo ' + nome + ' não encontrado. ');
      MessageBox(HWindow,sc,'Recuperar',mb_Ok or mb_IconInformation);
      EXIT;
    end;
  Readln(arq,comentario);
  Val(comentario,maximo,err);
  If err = 0 then Readln(arq,comentario)
  else maximo := 1000;
  StrPCopy(sc,comentario);
  coment^.SetText(sc);
  {$I-} Readln(arq,quadrupolo); {$I+}
  If (quadrupolo in [1,3,5]) or (IOResult = 0) then
    begin
      Str(quadrupolo,s);
      StrPCopy(sc,s);
      quad^.SetSelIndex(quad^.SetSelString(sc,-1));
      Readln(arq,ganho);
      Str(ganho,s);
      StrPCopy(sc,s);
      ganh^.SetSelIndex(ganh^.SetSelString(sc,-1));
      Readln(arq,tempo);
      Str(tempo:0:2,s);
      StrPCopy(sc,s);
      Tempoint^.SetText(sc);
    end
  else
    begin
      MessageBox(HWindow,'Arquivo de dados antigo.' + #13 +
        'Somente comentário e dados são disponíveis.',
        'Recuperar',mb_Ok or mb_IconInformation);
      Readln(arq,intens^[1]);
    end;
  i := 0;
  Repeat
    inc(i);
    Readln(arq,m,intens^[i]);
    massa^[i] := Word(Round(m*10));
  Until Eof(arq);
  Close(arq);
  ndados := i;
  massainicial := massa^[1];
  massafinal := massa^[ndados];
  incr := massa^[2] - massa^[1];
  m := massainicial / 10.0;
  Str(m:0:0,s);
  StrPCopy(sc,s); { Coloca cada valor dentro do seu respectivo }
  massain^.SetText(sc); { box de edição. }
  m := massafinal / 10.0;
  Str(m:0:0,s);
  StrPCopy(sc,s);

```

```

massafin^.SetText(sc);
incr := incr / 10.0;
Str(incr:0:1,s);
StrPCopy(sc,s);
incrmass^.SetText(sc);
mi := massainicial; { Restabelece as condições iniciais }
mf := massafinal; { para traçar o espectro total com }
maxy := 1000; { intensidade máxima de 100%. }
InvalidateRect(Grafico^.HWindow,nil,TRUE); { Limpa o grafico. }
Grafico^.temdados := TRUE;
Grafico^.adquirido := TRUE;
End;

Procedure TAquisicao.Copia(var Msg : TMessage);
var
  CopyDC : HDC; { Para o novo bitmap }
  CopyBitmap : HBitmap; { O novo bitmap }
Begin
  If OpenClipboard(HWindow) and EmptyClipboard then
    begin
      CopyDC := CreateCompatibleDC(Grafico^.DC); { Cria o novo bitmap. }
      CopyBitmap := CreateCompatibleBitmap(Grafico^.DC, 560, 260);
      CopyBitmap := SelectObject(CopyDC, CopyBitmap);
      BitBlt(CopyDC, 0, 0, 560, 260, Grafico^.DC, 0, 0, SrcCopy);
      CopyBitmap := SelectObject(CopyDC, CopyBitmap);
      SetClipboardData(cf_Bitmap, CopyBitmap); { Transfere o novo bitmap. }
      CloseClipboard;
      DeleteDC(CopyDC); { Limpa a memória. }
      MessageBox(HWindow,'Mapa de bit copiado.', 'Grafico',mb_Ok);
    end;
end;

Procedure TAquisicao.Print(var Msg : TMessage);
var Plinfo : TPaintStruct;
imp : String;
sc : Array[0..79] of Char;
NewFont, OldFont : HFont;
Begin
  coment^.GetText(sc,80);
  comentario := StrPas(sc);
  If Grafico^.temdados and (MessageBox(HWindow,'Imprime o
espectro?', 'Imprimir',
  mb_YesNo or mb_IconQuestion) = IDYes) then
    begin
      If PrnPartida('Espectro do Penta',imp) then
        begin
          { Verifica se a impressora é HP. }
          If (imp[1] = 'H') and (imp[2] = 'P') then
            begin
              Grafico^.xHi := 3 * 560; { Valores para as impressoras lasers. }
              Grafico^.yHi := 3 * 260;
              Grafico^.xLo := 100;
            end;
            NewFont := CreateFont(20,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
            OldFont := SelectObject(PDc,NewFont);
            SetTextAlign(PDc,TA_Right);

            TextOut(PDc,Grafico^.xHi,Grafico^.yHi+90,FileName,StrLen(FileName));
            SelectObject(PDc,OldFont);

```

```
DeleteObject(NewFont);
Grafico^.Paint(PDc,PInfo);
NovaPagina;
PrnFim;
Grafico^.xLo := 30; { Valores para impressoras matriciais. }
Grafico^.xHi := 560;
Grafico^.yHi := 260;
end;
end;
End;

Procedure TAquisicao.Calibra(var Msg : TMessage);
Const
    maxpico = 50;
Type
    Tdados = Array[1..maxpico] of Word;
var
    co : TNVector;
    incl,interc,a1,b1,a3,b3,a5,b5 : Double;

Procedure AjustaPol(nd : Integer; var massa,ValorDA : Tdados; ngrau :
Integer);
var M : TNMatrix;

    i,j : Integer;
    temp : Array[1..maxpico] of Double;
    err : Byte;

Function E(base : Double; expoente : Word) : Float;
var i : Word;
    px : Float;
Begin
    px := 1.0;
    For i := 1 to expoente do
        px := px * base;
    E := px;
End;

Begin { AjustaPol }
    For i := 1 to ngrau + 1 do
        begin
            co[i] := 0; { Limpa a matriz. }
            For j := 1 to ngrau + 1 do M[i,j] := 0.0;
            end;
        For i := 1 to nd do { Faz a somatoria. }
            begin
                temp[i] := massa[i] / 1000.0;
                j := 0;
                Repeat
                    Inc(j);
                    M[1,j] := M[1,j] + E(temp[i],j-1);
                    M[ngrau+1,j] := M[ngrau+1,j] + E(temp[i],ngrau+j-1);
                    co[j] := co[j] + E(temp[i],j-1) * ValorDA[i];
                Until j = ngrau+1;
            end;
        For i := 2 to ngrau do { Completa a matriz. }
            begin
                For j := 1 to ngrau do M[i,j] := M[i-1,j+1];
                M[i,ngrau+1] := M[ngrau+1,i];
            end;
        Gaussian_Elimination(ngrau+1,M,co,co,err);
    End;
```

```
Procedure SetQ(mi,mf : Integer; var valorDA : Word);
var m,Intens : Tdados;
    i,n,pos,max : Word;
Begin
    Q1^.SetMassa(mi/10.0); { Espera na massa inicial antes de prosseguir. }
    For i := 1 to 65000 do; { O tempo de estabilização dos quadrupolos é de
10 ms }
        i := 0; { para um salto maior ou igual a 50 unidades de massa }
        Repeat
            inc(i);
            m[i] := mi + (i-1) * incr;
            Adquire(Q1,m[i],Intens[i]);
        Until (m[i] >= mf);
        n := i;
        max := 0;
        For i := 1 to n do
            If max < Intens[i] then { Localiza o pico pela posição do seu máximo. }
                begin
                    max := Intens[i];
                    pos := i;
                end;
            valorDA := Word(Round(m[pos]/10.0 * incl + interc));
        End; { valorDA é calculado com incl e interc da calibração anterior.}

Procedure AjustaPicos(nomearq : String; var a,b : Double);
var
    arq : Text;
    mp,mda : Tdados;
Begin
    Assign(arq,path+nomearq);
    Reset(arq);
    Readln(arq,incl);
    Readln(arq,interc);
    Close(arq);
    mp[1] := 69; { Localiza os picos mais intensos do PFTBA. }
    mp[2] := 131;
    mp[3] := 219;
    SetQ(670,710,mda[1]);
    SetQ(1290,1330,mda[2]);
    SetQ(2170,2210,mda[3]);
    AjustaPol(3,mp,mda,1);
    a := co[2] / 1000;
    b := co[1];
End;

Procedure SalvaCal(nome : String; a, b : Double);
var
    arq : Text;
Begin
    Assign(arq,path+nome);
    ReWrite(arq);
    Writeln(arq,a:0:8);
    Writeln(arq,b:0:8);
    Close(arq);
End;

Begin { TAquisicao.Calibra }
    If MessageBox(HWWindow,'Confirma calibração?','Calibrar',
mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
```

```

    MessageBox(HWindow,'Prepare Q1 para ser calibrado...','Calibração de
Q1',mb_Ok);
    incr := 1;          { incremento de 0.1 u }
    tempo := 33.4;      { tempo de integração 33.4 ms = 2 ciclos de 60Hz }
    ganho := 100;       { ganho ideal para atenuação = 1 e sensibilidade =
med. }
    det := New(PDetector,init(1,0,ganho,tempo));
    Q1 := New(PQuadrupolo,init(1,0,0,path+'Q1.CAL'));
    AjustaPicos('Q1.CAL',a1,b1);
    Dispose(Q1,done);
    If MessageBox(HWindow,'Prepare Q3 para ser calibrado... Continua?',
        'Calibração de Q3',mb_YesNo) = IDNo then EXIT;
    Q1 := New(PQuadrupolo,init(2,0,1,path+'Q3.CAL'));
    AjustaPicos('Q3.CAL',a3,b3);
    Dispose(Q1,done);
    If MessageBox(HWindow,'Prepare Q5 para ser calibrado... Continua?',
        'Calibração de Q5',mb_YesNo) = IDNo then EXIT;
    Q1 := New(PQuadrupolo,init(1,1,2,path+'Q5.CAL'));
    AjustaPicos('Q5.CAL',a5,b5);
    Dispose(Q1,done);
    Dispose(det,done);
    If MessageBox(HWindow,'Salva as calibrações?','Calibrar',mb_YesNo)
        = IDNo then EXIT;
    SalvaCal('Q1.CAL',a1,b1);
    SalvaCal('Q3.CAL',a3,b3);
    SalvaCal('Q5.CAL',a5,b5);
End;
```

```

Procedure TAquisicao.Expandir(var Msg : TMessage);
Begin
    maxy := expansao;
    mi := expmi;
    mf := expmf;
    InvalidateRect(Grafico^,HWindow,nil,TRUE); { Redesenha o espectro
expandido. }
End;
```

```

Procedure TAquisicao.Normal(var Msg : TMessage);
Begin
    maxy := 1000;
    mi := massainicial;
    mf := massafinal;
    InvalidateRect(Grafico^,HWindow,nil,TRUE); { Redesenha o espectro
normal. }
End;
```

```

Procedure TAquisicao.Sobre(var Msg : TMessage);
Begin
    MessageBox(HWindow,'Desenvolvido por Valmir F. Juliano - 1994/95',
        'Programa PENTA2D - Versão 2.5',mb_...Ok or
mb_IconInformation);
End;
```

```

Procedure TAquisicao.Fim(var Msg : TMessage);
Begin
    CloseWindow;
End;
```

```

Procedure TAquisicao.Cancela(var Msg : TMessage);
Begin
    Grafico^.acabou := TRUE;
```

```

End;

Function TAquisicao.CanClose: Boolean;
Begin
    Dispose(intens);
    Dispose(massa);
    Dispose(lon,done);
    DoneWinCrt;
    CanClose := TRUE;
End;

BEGIN
    Aplica.Init('Aplica');
    Aplica.Run;
    Aplica.Done;
END.
```

PENTA3D



```
Program PENTA3D;
{-----}
      Programa para aquisição de espectros tridimensionais do
espectrômetro de massas pentaquadrupolar.
      Utiliza os métodos de programação orientada a objetos e
programação motivada por eventos.
      Desenvolvido por Valmir Fascio Juliano como parte do
trabalho de doutoramento no Instituto de Química - UNICAMP sob
orientação da Profa. Dra. Concetta Kascheres e do Prof. Dr. Claudimir
Lúcio do Lago (IQ - USP).
{-----}

{ Compilar com Stack Size = 10000 e Heap Size = 8182 }

Uses
Strings,WinTypes,WinDos,WinProcs,WObjects,Acess,CommDlg,PentAqui
,WinCrt;

{$R penta.res}

Const
  id_EditBox = -1;
  id_PbOK = 101;
  id_PbFim = 102;
  id_PbCancela = 103;
  cm_salva = 1;
  cm_recupera = 2;
  cm_Copia = 3;
  cm_Print = 4;
  cm_Expande = 5;
  cm_Normal = 6;
  cm_Sobre = 7;
  cm_Converte = 8;
  cm_CopiaUni = 901;
  cm_PrintUni = 902;
  cm_SalvaUni = 903;
  maxdados = 10000; { Número máximo de dados no vetor de dados. }
  maxpicos = 1000; { Número máximo de dados no vetor de picos
encontrados. }
  nomedefault = 'E2D.B.D';
  path = 'C:\PENTA\';
  pathdados = 'C:\PENTA\Q5ESP';

Type
  TFilename = Array [0..255] of Char;

  Aplicacao = object(TApplication)
    Procedure InitMainWindow; virtual;
  end;

  PAquisicaoGrafico = ^TAquisicaoGrafico;
  TAquisicaoGrafico = object(TWindow)
    adquirindo, LButtonDown, RButtonDown, acabou : Boolean;
    mousex1,mousexf,mouseyi,mouseyf, mouseyint, mousexint,
    xHi, xLo, yHi : Integer;
    Dc : HDC;
    NewPen, OldPen : HPen;
    constructor Init(AParent: PWindowsObject; ATitle: PChar);
    Function GetClassName: PChar; virtual;
    Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
```

```
    Procedure WMLButtonDown(var Msg : TMessage); virtual wm_First +
wm_LButtonDown;
    Procedure WMLButtonUp(var Msg : TMessage); virtual wm_First +
wm_LButtonUp;
    Procedure WMRButtonDown(var Msg : TMessage); virtual wm_First +
wm_RButtonDown;
    Procedure WMRButtonUp(var Msg : TMessage); virtual wm_First +
wm_RButtonUp;
    Procedure WMMouseMove(var Msg : TMessage); virtual wm_First +
wm_MouseMove;
    Procedure WMLButtonDbIClk(var Msg : TMessage); virtual wm_First +
wm_LButtonDbIClk;
    Procedure WMRButtonDbIClk(var Msg : TMessage); virtual wm_First +
wm_RButtonDbIClk;
  end;

  PAquisicao = ^TAquisicao;
  TAquisicao = object(TWindow)
    PbOK,PbFim,PbCancela : PButton;
    quad1, quad2, ganh, massain1, massain2, massafin1, massafin2,
    incmass1, Tempoint, coment, Enion : PEdit;
    grafico : PAquisicaoGrafico;
    FileName : TFileName;
    salva_recupera : Integer;
    ok_arquivo : Boolean;
    Constructor Init(AParent: PWindowsObject; ATitle: PChar);
    Function GetClassName: PChar; virtual;
    Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
    Procedure Mostra; virtual;
    Procedure Executa(var Msg: TMessage); virtual id_First + id_PbOK;
    Procedure Fim(var Msg: TMessage); virtual id_First + id_PbFim;
    Procedure Cancela(var Msg : TMessage); virtual id_First +
id_PbCancela;
    Function CanClose : Boolean; virtual;
    Procedure Salvar(var Msg : TMessage); virtual cm_First + cm_salva;
    Procedure Recuperar(var Msg : TMessage); virtual cm_First +
cm_recupera;
    Procedure Copia(var Msg : TMessage); virtual cm_First + cm_Copia;
    Procedure CMFileOpen(var Msg: TMessage; titulo,tipo,DefExt : PChar);
virtual;
    Procedure Print(var Msg : TMessage); virtual cm_First + cm_Print;
    Procedure Expandir(var Msg : TMessage); virtual cm_First +
cm_Expande;
    Procedure Normal(var Msg : TMessage); virtual cm_First + cm_Normal;
    Procedure Sobre(var Msg : TMessage); virtual cm_First + cm_Sobre;
    Procedure GraficoUni(var Msg : TMessage); virtual cm_First +
cm_converte;
  end;

  PGrafico = ^TGrafico;
  TGrafico = object(TWindow)
    ButtonDown : Boolean;
    mousex1, mousexf,mouseyi,mouseyf,
    xHi, xLo, yHi, yLo : Integer;
    Dc : HDC;
    NewPen, OldPen : HPen;
    mi,mf,maxy,ndados,expansao,expmi,expmf,
    massainicial, massafinal : Word;
    maximo : LongInt;
    massa : Array[1..maxdados] of Word;
```

```
Intens : Array[1..maxdados] of LongInt;
Constructor Init(AParent: PWindowsObject; ATitle: PChar);
Function GetClassName: PChar; virtual;
Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
Procedure WMLButtonDown(var Msg : TMessage); virtual wm_First +
wm_LButtonDown;
Procedure WMLButtonUp(var Msg : TMessage); virtual wm_First +
wm_LButtonUp;
Procedure WMRButtonDown(var Msg : TMessage); virtual wm_First +
wm_RButtonDown;
Procedure WMMouseMove(var Msg : TMessage); virtual wm_First +
wm_MouseMove;
Procedure WMLButtonDbtClk(var Msg : TMessage); virtual wm_First +
wm_LButtonDbtClk;
Procedure Copia(var Msg : TMessage); virtual cm_First + cm_CopiaUni;
Procedure Print(var Msg : TMessage); virtual cm_First + cm_PrintUni;
Procedure Salva(var Msg : TMessage); virtual cm_First + cm_SalvaUni;
Function CanClose : Boolean; virtual;
end;

PVetorW = ^TVetorW;
TVetorW = Array[1..maxdados] of Word;
PVetorW2 = ^TVetorW2;
TVetorW2 = Array[1..maxpicos] of Word;
par = Record
m1i,m1f,m2i,m2f,max : Word;
g,inc1,inc2,ni,nj : Integer;
quad1,quad2 : Byte;
tint : Double;
comen : String;
end;
reg = Record
m2 : Word;
m1,i : TVetorW;
end;

var
mi1, mi2, mf1, mf2, maxy, massainicial1, massainicial2,
massafinal1, massafinal2, expansao,
expmi, expmf, expmiz, expmfz, maximo, ndadosi, ndadosj : Word;
ganhogeral,inc1, posj, njanelas : Integer;
tempo, teta, zmax, energion : Double;
quadrupolo1,quadrupolo2 : Byte;
Aplica : Aplicacao;
det : PDetector;
Qx,Qz : PQuadrupolo;
Ion : Pionizador;
comentario,nomearq : String;
massa1,intens : PVetorW;
massa2 : PVetorW2;
parametros : par;
arqpar : File of par;
registro : reg;
arq,arq1 : File of reg;
imprimir, temdados : Boolean;

Function z(zp : Word) : Double; { Define um ponto no plano XY. }
Begin
z := zmax * (zp - mi2) / (mf2 - mi2);
End;
```

```
Function xz(z : Double) : Word; { Em unidade de massa. }
Begin
xz := Word(Round(z * cos(teta)));
End;

Function yz(z : Double) : Word; { Em unidade de Intensidade. }
Begin
yz := Word(Round(z * sin(teta)));
End;

Procedure Aplicacao.InitMainWindow;
Begin
MainWindow := New(PAquisicao,Init(nil,'PENTA3D'));
imprimir := FALSE;
ScreenSize.X := 26;
ScreenSize.Y := 4;
InitWinCrt;
WriteLn('Não feche esta janela,');
WriteLn('minimize-a se necessario.');
```

```
End;

Constructor TAquisicaoGrafico.Init(AParent: PWindowsObject; ATitle:
PChar);
Begin
temdados := FALSE; { Sinalizador para indicar que existem dados. }
adquirido := FALSE; { Sinalizador para evitar re-aquisição. }
LButtonDown := FALSE;
RButtonDown := FALSE;
xHi := 560;
xLo := 20;
yHi := 260;
TWindow.Init(AParent,ATitle);
with Attr do
begin
Style := ws_Child OR ws_Overlapped OR ws_Visible;
X := 330; Y := 45; W := 600; H := 260;
end;
End;

Function TAquisicaoGrafico.GetClassName: PChar;
begin
GetClassName := 'GraphWndClass'
end;

Procedure TAquisicaoGrafico.GetWindowClass(var AWndClass:
TWndClass);
Begin
TWindow.GetWindowClass(AWndClass);
(* AWndClass.hbrBackground := GetStockObject(LtGray_Brush);{ Fundo
gráfico cinza.}*)
AWndClass.Style := AWndClass.Style or cs_DblClks;{ Aceita click duplo
do mouse.}
End;

Procedure Adquire(var Q : PQuadrupolo; massa : Integer; var intensidade :
Word);
Begin
Q^.SetMassa(massa/10.0);
If det^.tempointeg < 0.029 then intensidade := det^.LeAD
else intensidade := det^.LeADAvg;
```

```
End;

Function LocalizaRegiaodeVarredura(var m1 : PVetorW; var m2 : PVetorW2;
var mi,mf,nj : Word) : Boolean;

const
g : Array[1..7] of Integer = (100,50,20,10,5,2,1); { Ganhos disponiveis. }
var
k : Byte;
i, j, n, np, lb, limiarinicial, limiarfinal : Word;
soma : LongInt;
slb : Double;
H : PVetorW2;
tudo_ok : Boolean;

Function Limiar(i : Word) : Word; { Limiar em função da linha base variável. }
Begin
Limiar := Word(Round(1.0*(1.0*limiarfinal-limiarinicial)*(i-1)/(n-1)+limiarinicial));
End;

Procedure MarcaPico(inicio,fim : Word);

var i, pos, max, m, ma : Word;
Begin
max := 0;
For i := inicio to fim do
If max < intens^[i] then
begin
max := intens^[i]; { Localiza ponto máximo do pico. }
pos := m1^[i];
end;
Inc(np);
m2^[np] := pos; { Vetor com todos os pontos máximos. }
H^[np] := max; { Vetor com todas as intensidades máximas. }
If np > 1 then
begin
m := m2^[np];
ma := m2^[np-1];
If m < ma + 6 then { Compara posições separadas por menos de 0.6 u. }
begin
begin
If H^[np-1] < H^[np] then
begin
H^[np-1] := H^[np]; { Ignora o pico menos intenso. }
m2^[np-1] := m2^[np];
end;
np := np - 1;
end;
end;
End;
```

```
Procedure LocPico(inreg,fimreg : Word);
var
i,ini,inicio,fim : Word;
Begin
i := inreg + 1;
While i < fimreg do
begin
inicio := i - 1;
ini := i;
```

```
Repeat { Encontra os limites do pico pela primeira derivada }
While (1.0*intens^[i+1] - intens^[i-1] >= 0) and (i < fimreg) do Inc(i);
While (1.0*intens^[i+1] - intens^[i-1] < 0) and (i < fimreg) do Inc(i);
Until (i - ini > 2) or (i >= fimreg); { 3 pto p/ indicar fim do pico }
fim := i;
If np < maxpicos then MarcaPico(inicio,fim);
Inc(i);
end;
End;
```

```
Procedure RegiaodoPico(i : Word; var j : Word);
var
k,inreg,fimreg : Word;
```

```
Function TemCerteza : Boolean;
var
TaCerto : Boolean;
Begin
TaCerto := intens^[i + j] >= Limiar(i+j);
TemCerteza := TaCerto;
If not TaCerto then
begin
k := 0;
While(intens^[i+j+k] < Limiar(i+j+k)) and (i + j + k < n) do Inc(k);
TaCerto := k > 2; { 3 pontos consecutivos < limiar indicam fim do pico. }
If not TaCerto then Inc(j,k);
TemCerteza := TaCerto;
end;
End;
```

```
Begin { RegiaodoPico }
j := 0;
While ((intens^[i + j] > Limiar(i-j)) and (i + j < n) and TemCerteza) do Inc(j);
If j > 2 then { 3 pontos consecutivos > limiar indicam um pico. }
begin
inreg := i;
fimreg := i + j;
If np < maxpicos then LocPico(inreg,fimreg);
end;
End;
```

```
Begin { LocalizaRegiaodeVarredura }
LocalizaRegiaodeVarredura := TRUE;
New(H);
np := 0; { Nro. de picos localizados. }
k := 0;
det^.init(1,0,ganhogeral,2.0); { Ajusta o tempo para 2 ms. }
Qx^.SetPoleDC(off); { Desliga POLE DC do quadrupolo de varredura do eixo x. }
Qx^.SetMassa(massainicial/10.0);
Repeat { Repete a varredura até ajustar um ganho que não sature o sinal. }
tudo_ok := TRUE;
Inc(k);
det^.SetGanho(g[k]);
i := 0;
Repeat
Inc(i);
m1^[i] := mi + (i-1); { Incrementos de 0.1 u. }
Acquire(Qz,m1^[i],intens^[i]);
If intens^[i] > 65000 then tudo_ok := FALSE; { Sinal saturado. }
```

```

Until (m1^i) >= mf) or (i >= maxdados) or (not tudo_ok);
Until (tudo_ok) or (k = 7);
det^.init(1,0,ganhogeral,tempo); { Retorna tempo e ganho aos valores
iniciais. }
Qx^.SetPoleDC(on);
If tudo_ok then
begin
n := i;

soma := 0;
For i := 1 to 20 do soma := soma + intens^[i];
lb := soma div 20; { Valor médio da linha base. }
soma := 0;
For i := 1 to 20 do soma := soma + Round(sqrt(1.0*lb - intens^[i]));
slb := sqrt(soma / 19.0); { Desvio padrão da linha base. }
limiarinicial := Round(lb + 3.0 * slb); { Limiar para detecção de um
pico. }

soma := 0;
For i := 1 to 20 do soma := soma + intens^[n-i+1];
lb := soma div 20;
soma := 0;
For i := 1 to 20 do soma := soma + Round(sqrt(1.0*lb - intens^[n-i+1]));
slb := sqrt(soma / 19.0);
limiarfinal := Round(lb + 3.0 * slb);

i := 1;
j := 0;
Repeat
If intens^[i] > Limiar(i) then
begin
RegiaoDoPico(i,j);
inc(i,j); { Incrementa a posição do vetor com j pontos de largura. }
end
else inc(i);
Until i >= n;
end
else LocalizaRegiaoDeVarredura := FALSE;
If (np > 0) and (np <= maxpicos) then
begin
mi := 100 * (m2^[1] div 100-1); { Década de massa anterior ao pico. }
mf := 100 * (m2^[np] div 100+1); { Década de massa posterior ao pico. }
nj := np;
LocalizaRegiaoDeVarredura := TRUE;
massainicial2 := mi;
massafinal2 := mf;
end
else LocalizaRegiaoDeVarredura := FALSE;
Dispose(H);
End;

```

```

Procedure TAquisicaoGrafico.Paint(PaintDC : HDC; var PaintInfo :
TPaintStruct);
var
NewCur,
OldCur : HCursor;
OldFont,
NewFont : HFont;
i,j,xi,incrm, posx : Integer;
yi,inten : Word;
s : string[95];
s1 : string[95];
sc : Array[0..199] of Char;

```

```

Function Y(i : Word) : Integer; { Converte leitura do ADC para intensidade. }
Begin
Y := Round(1000.0*i/65535);
End;

Begin { TAquisicaoGrafico.Paint }
If not temdados then EXIT;
If (not adquirido) then
begin
det := New(PDetector,init(1,0,ganhogeral,tempo));{ Inicializa o detector.
}

Case quadrupolo1 of
1 : Qx := New(PQuadrupolo,init(1,0,0,path+'Q1.CAL'));{ Inicializa o
quadrupolo1. }
3 : Qx := New(PQuadrupolo,init(2,0,1,path+'Q3.CAL'));
5 : Qx := New(PQuadrupolo,init(1,1,2,path+'Q5.CAL'));
end;

Case quadrupolo2 of
1 : Qz := New(PQuadrupolo,init(1,0,0,path+'Q1.CAL'));{ Inicializa o
quadrupolo2. }
3 : Qz := New(PQuadrupolo,init(2,0,1,path+'Q3.CAL'));
5 : Qz := New(PQuadrupolo,init(1,1,2,path+'Q5.CAL'));
end;

end;
NewCur := LoadCursor(0,rc_Wait); { Sinaliza com a ampulheta. }
OldCur := SetCursor(NewCur);
If not adquirido then { Localiza os picos. }
If not LocalizaRegiaoDeVarredura(massa1,masa2,mi2,mf2,ndados)
then
begin
MessageBox(HWWindow,'Todos os ganhos saturaram ou não existe
sinal.'

+ #13 + 'Verifique e repita a operação.',
'Aquisição',mb_Ok or mb_IconInformation);
Dispose(det,done); { Libera o espaço de memória ocupada pelos
ponteiros }
Dispose(Qz,done); { detector e quadrupolos. }
Dispose(Qx,done);
SetCursor(OldCur);
temdados := FALSE;
adquirido := FALSE;
EXIT;
end;
SetTextAlign(PaintDC,TA_RIGHT);
SetTextColor(PaintDC,RGB(0,0,255));
SetBkMode(PaintDC,transparent);
SetMapMode(PaintDC,mm_Anisotropic);
SetViewportOrg(PaintDC,xLo,yHi-30);
SetViewportExt(PaintDC,xHi-xLo,-(yHi-30));
SetWindowExt(PaintDC,2*(mf1-mi1),2*maxy); { O nro. 2 permite o
colocação de z. }
SetWindowOrg(PaintDC,mi1,0);

NewFont :=
CreateFont(Round(0.06*maxy),0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
OldFont := SelectObject(PaintDC,NewFont);
zmax := sqrt(sqrt(1.0*mf1 - mi1) + sqrt(1.0*maxy)); { Modulo de z. }
teta := ArcTan(1.0*maxy/(1.0*mf1-mi1)); { Angulo em radianos entre x e z
no plano xy. }

```



```

LineTo(PaintDC, massa1^i + xz(z(massa2^j)));
Y(intens^i) + yz(z(massa2^j)));
If maximo < intens^i then maximo := intens^i;
KeyPressed;
Until (massa1^i >= mf1) or (i >= maxdados) or acabou;
ndadosi := i;
registro.m2 := massa2^j;
registro.m1 := massa1^i;
registro.i := intens^i;
Write(arq, registro);
end;
MessageBeep(MB_OK);      { Emite um beep após o término da
aquisição. }
SelectObject(PaintDC, OldPen);
DeleteObject(NewPen);
adquirido := TRUE;
end;
Reset(arq);
NewPen := CreatePen(ps_Solid, 0, RGB(255, 0, 0));
OldPen := SelectObject(PaintDC, NewPen);
If maximo > 0 then
For j := ndadosj downto 1 do
begin
Read(arq, registro);
massa2^j := registro.m2;
massa1^i := registro.m1;
intens^i := registro.i;
If (massa2^j >= mi2) and (massa2^j <= mf2) then
begin
NewPen := CreatePen(ps_Solid, 0, RGB(255, 255, 255));
OldPen := SelectObject(PaintDC, NewPen);
For i := 1 to ndadosi do { Apaga o que está por baixo de cada pico.
}
begin
inten := Word(Round(1000.0 * intens^i / maximo));
If (massa1^i >= mi1) and (massa1^i <= mf1) then
begin
MoveTo(PaintDC, massa1^i +
xz(z(massa2^j)), 1 + yz(z(massa2^j)));
LineTo(PaintDC, massa1^i + xz(z(massa2^j)), inten +
yz(z(massa2^j)))
end;
end;
SelectObject(PaintDC, OldPen);
DeleteObject(NewPen);
inten := intens^i[1];
MoveTo(PaintDC, mi1 + xz(z(massa2^j)));
Word(Round(1000.0 * inten / maximo)) + yz(z(massa2^j)));
For i := 1 to ndadosi do { Desenha o espectro. }
begin
inten := Word(Round(1000.0 * intens^i / maximo));
If (massa1^i >= mi1) and (massa1^i <= mf1) then
LineTo(PaintDC, massa1^i + xz(z(massa2^j)),
inten + yz(z(massa2^j)));
end;
end;
end;
SelectObject(PaintDC, OldPen);
DeleteObject(NewPen);
If not adquirido then
begin

```

```

Dispose(det, done); { Libera o espaço de memória ocupada pelos
ponteiros }
Dispose(Qz, done); { detector e quadrupolos. }
Dispose(Qx, done);
end;
SetCursor(OldCur);
Close(arq);
DC := PaintDC;
End;

Procedure TAquisicaoGrafico.WMLButtonDown(var Msg : TMessage);
Begin
If not LButtonDown and temdados then
begin
mousexi := xLo;      { Valores iniciais para a posição do mouse. }
mousexf := xHi;
mouseyf := 0;
LButtonDown := TRUE;
SetCapture(HWindow);
mousexi := Integer(Msg.LParamLo); { Obtém a nova posição do
mouse. }
mouseyi := Integer(Msg.LParamHi);
NewPen := CreatePen(ps_Solid, 0, RGB(0, 0, 255)); { Linha azul. }
OldPen := SelectObject(Dc, NewPen);
end;
End;

Procedure TAquisicaoGrafico.WMLButtonUp(var Msg : TMessage);
var x : Integer;
Begin
If LButtonDown and temdados then
begin
MoveTo(Dc, mousexi, mouseyi);
LineTo(Dc, mousexf, mouseyf); { Linha no plano xz -> região a ser
ampliada. }
expmiz := Word(10*Round((1.0*mi2-((1.0*mouseyi-(yHi-
30))*2.0*maxy*(1.0*mf2-mi2))/((yHi-30)*zmax*sin(teta)))/10.0));
If (expmiz < massainicial2) or (expmiz >= massafinal2) then expmiz :=
massainicial2;
expmfz := Word(10*Round((1.0*mi2-((1.0*mouseyf-(yHi-
30))*2.0*maxy*(1.0*mf2-mi2))/((yHi-30)*zmax*sin(teta)))/10.0));
If (expmfz > massafinal2) or (expmfz <= massainicial2) then expmfz :=
massafinal2;
expmi := Word(10*Round(((mousexi-xLo)*2.0*(1.0*mf1-mi1)/(xHi-
xLo)+1.0*mi1-xz(z(expmiz)))/10.0));
If (expmi < massainicial1) or (expmi >= massafinal1) then
expmi := massainicial1;
expmf := Word(10*Round(((mousexf-xLo)*2.0*(1.0*mf1-mi1)/(xHi-
xLo)+1.0*mi1-xz(z(expmfz)))/10.0));
If expmf > massafinal1 then expmf := massafinal1;
If (expmf - expmi) < 100 then expmf := expmi + 100;
SelectObject(Dc, OldPen);
DeleteObject(NewPen);
ReleaseCapture;
LButtonDown := FALSE;
end;
End;

Procedure TAquisicaoGrafico.WMRButtonDown(var Msg : TMessage);
Begin
If not RButtonDown then

```

```

begin
  RButtonDown := TRUE;
  SetCapture(HWindow);
  mouseint := Integer(Msg.LParamHi);
  mousexint := Integer(Msg.LParamLo);
  NewPen := CreatePen(ps_solid,0,RGB(0,0,255)); { Linha azul. }
  OldPen := SelectObject(Dc,NewPen);
end;
End;

Procedure TAquisicaoGrafico.WMRButtonUp(var Msg : TMessage);
Begin
  If RButtonDown then
    begin
      SetBkMode(Dc,transparent);
      SetMapMode(Dc,mm_Anisotropic);
      SetViewportOrg(Dc,xLo,yHi-30);
      SetViewportExt(Dc,xHi-xLo,-(yHi-30));
      SetWindowExt(Dc,2*(mf1-mi1),2*maxy);
      SetWindowOrg(Dc,mi1,0);
      expansao := Word(Round(-2.0*maxy*(mouseint-(yHi-30))/(yHi-30)));
      If expansao > 1000 then expansao := 1000;
      If expansao < 50 then expansao := 50; { intensidade minima = 5% }
      MoveTo(Dc,expmi+xz(z(expmiz)),yz(z(expmiz)));
      LineTo(Dc,expmf+xz(z(expmiz)),yz(z(expmiz)));
      LineTo(Dc,expmf+xz(z(expmfz)),yz(z(expmfz)));
      LineTo(Dc,expmf+xz(z(expmfz)),expansao + yz(z(expmfz)));
      LineTo(Dc,expmi+xz(z(expmfz)),expansao + yz(z(expmfz)));
      LineTo(Dc,expmi+xz(z(expmfz)),yz(z(expmfz)));
      LineTo(Dc,expmi+xz(z(expmiz)),yz(z(expmiz)));
      SelectObject(Dc,OldPen);
      DeleteObject(NewPen);
      ReleaseCapture;
      RButtonDown := FALSE;
    end;
  End;

Procedure TAquisicaoGrafico.WMMouseMove(var Msg : TMessage);
Begin
  If LButtonDown then With Msg do
    begin
      mousexf := Integer(LParamLo); { Captura as posições do mouse em movimento. }
      mouseyf := Integer(LParamHi);
    end;
  End;

Procedure TAquisicaoGrafico.WMLButtonDownDbkClk(var Msg : TMessage);
Begin
  maxy := 1000; { Estabelece as condições do espectro normal. }
  mi1 := massainicial1;
  mf1 := massafinal1;
  mi2 := massainicial2;
  mf2 := massafinal2;
  InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro após o duplo click. }
End;

Procedure TAquisicaoGrafico.WMRButtonDownDbkClk(var Msg : TMessage);
Begin
  mi1 := expmi;

```

```

mf1 := expmf;
mi2 := expmiz;
mf2 := expmfz;
maxy := expansao;
  InvalidateRect(HWindow, nil, TRUE); { Desenha o espectro expandido após dbclk. }
End;

Constructor TAquisicao.Init(AParent: PWindowsObject; ATitle: PChar);
var
  WinDir : array [0..145] of Char;
Begin
  TWindow.Init(AParent,ATitle);
  EnableKBHandler;
  with Attr do
    begin
      X := CW_USEDEFAULT; Y := CW_USEDEFAULT; W := 980; H := 440;
      Menu := LoadMenu(HInstance,'menu_2');
    end;
  Ion := New(Plonizador.Init(1,1,1));
  quad1 := New(PEdit,Init(@Self,id_EditBox,'5',195,65,20,30,2,FALSE));
  quad2 := New(PEdit,Init(@Self,id_EditBox,'1',270,65,20,30,2,FALSE));
  massain1 :=
  New(PEdit,Init(@Self,id_EditBox,'10',195,100,40,30,4,FALSE));
  massafin1 :=
  New(PEdit,Init(@Self,id_EditBox,'100',195,135,50,30,5,FALSE));
  IncrMass1:=
  New(PEdit,Init(@Self,id_EditBox,'0.1',195,170,40,30,4,FALSE));
  massain2 :=
  New(PEdit,Init(@Self,id_EditBox,'10',270,100,40,30,4,FALSE));
  massafin2 :=
  New(PEdit,Init(@Self,id_EditBox,'100',270,135,50,30,5,FALSE));
  Tempoint :=
  New(PEdit,Init(@Self,id_EditBox,'0.0',195,205,50,30,5,FALSE));
  ganh := New(PEdit,Init(@Self,id_EditBox,'20',195,240,40,30,4,FALSE));
  Enion := New(PEdit,Init(@Self,id_EditBox,'70.0',195,275,50,30,5,FALSE));
  coment := New(PEdit,Init(@Self,id_EditBox,',',380,320,520,30,81,FALSE));
  PbOK := New(PButton,Init(@Self,id_PbOK,'OK para
iniciar',20,320,130,30,FALSE));
  PbFim := New(PButton,Init(@Self,id_PbFim,'FIM',160,320,50,30,TRUE));
  PbCancela :=
  New(PButton,Init(@Self,id_PbCancela,'Cancela',220,320,80,30,FALSE));
  New(massa1);
  New(massa2);
  New(intens);
  Grafico := New(PAquisicaoGrafico,Init(@Self,nil));
  maxy := 1000; { Correponde ao máximo para o pico base =
100.0%}
  ok_arquivo := FALSE;
  StrPCopy(WinDir,pathdados);
  SetCurDir(WinDir); { Diretório Default. }
End;

Procedure TAquisicao.GetWindowClass(var AWndClass : TWndClass);
Begin
  TWindow.GetWindowClass(AWndClass);
  AWndClass.HIcon := LoadIcon(HInstance,'bid');
End;

Function TAquisicao.GetClassName: PChar;
begin

```

```

GetClassName := 'MainWndClass'
end;

Procedure TAquisicao.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
    OldBrush,
    NewBrush : HBrush;
    NewFont,
    OldFont : HFont;

Begin
    NewFont := CreateFont(18,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
        OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
        VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
    OldFont := SelectObject(PaintDC,NewFont);
    SetTextColor(PaintDC,RGB(255,0,128));
    SetBkMode(PaintDC,transparent);
    SetTextAlign(PaintDC,TA_LEFT);
    TextOut(PaintDC,100,5,'AQUISIÇÃO E MANIPULAÇÃO DE ESPECTROS
DO ESPECTRÔMETRO DE MASSAS PENTAQUADRUPOLAR',80);
    TextOut(PaintDC,190,35,'eixo x eixo z',19);
    TextOut(PaintDC,15,70,'Quadrupolos de varredura:',25);
    TextOut(PaintDC,15,105,'Massas iniciais:',16);
    TextOut(PaintDC,15,140,'Massas finais:',14);
    TextOut(PaintDC,15,175,'Incremento de Massa:',20);
    TextOut(PaintDC,15,210,'Tempo de integração (ms):',25);
    TextOut(PaintDC,15,245,'Ganho:',6);
    TextOut(PaintDC,15,280,'Energia de ionização (eV):',26);
    TextOut(PaintDC,300,325,'Comentário:',11);
    SelectObject(PaintDC,OldFont);
    DeleteObject(NewFont);
End;

```

```

Procedure TAquisicao.Mostra;

```

```

var
    s : string;
    sc : Array[0..79] of Char;

Begin
    Str(quadrupolo1,s);
    StrPCopy(sc,s);
    quad1^.SetText(sc);
    Str(quadrupolo2,s);
    StrPCopy(sc,s);
    quad2^.SetText(sc);
    Str(ganhogeral,s);
    StrPCopy(sc,s);
    ganh^.SetText(sc);
    Str((massainicial1 / 10.0):0:0,s);
    StrPCopy(sc,s);
    massain1^.SetText(sc);
    Str((massafinal1 / 10.0):0:0,s);
    StrPCopy(sc,s);
    massafin1^.SetText(sc);
    Str((incr1 / 10.0):0:1,s);
    StrPCopy(sc,s);
    IncrMass1^.SetText(sc);
    Str((massainicial2 / 10.0):0:0,s);
    StrPCopy(sc,s);
    massain2^.SetText(sc);
    Str((massafinal2 / 10.0):0:0,s);

```

```

StrPCopy(sc,s);
massafin2^.SetText(sc);
Str(tempo:0:2,s);
StrPCopy(sc,s);
Tempoint^.SetText(sc);
StrPCopy(sc,comentario);
coment^.SetText(sc);
End;

```

```

Procedure TAquisicao.Executa(var Msg: TMessage);

```

```

var
    sc : Array[0..199] of Char;
    s : String[80];
    s1 : String[3];
    err : Integer;
    incre, m : Double;

```

```

Begin
    { Adquire um espectro após confirmação. }
    quad1^.GetText(sc,10); { Captura os valores nos boxes de edição. }
    Val(sc,quadrupolo1,err);
    If not (quadrupolo1 in [1,3,5]) then quadrupolo1 := 1;
    quad2^.GetText(sc,10);
    Val(sc,quadrupolo2,err);
    If not (quadrupolo2 in [1,3,5]) then quadrupolo2 := 5;
    ganh^.GetText(sc,10);
    Val(sc,ganhogeral,err);
    If not (ganhogeral in [1,2,5,10,20,50,100]) then ganhogeral := 50;
    massain1^.GetText(sc,10);
    Val(sc,m,err);
    massainicial1 := Round(m * 10);
    If massainicial1 < 10 then massainicial1 := 10; { Limita a massa inicial = 1
u. }
    mi1 := massainicial1;
    massafin1^.GetText(sc,10);
    Val(sc,m,err);
    massafinal1 := Round(m * 10);
    If massafinal1 < (massainicial1 + 100) then massafinal1 := massainicial1
+ 100;
    If massafinal1 > 10000 then massafinal1 := 10000;{ Limita a massa final
= 1000 u.}
    mf1 := massafinal1;
    IncrMass1^.GetText(sc,10);
    Val(sc,incre,err);
    incr1 := Round(incre * 10);
    If incr1 < 1 then incr1 := 1;
    massain2^.GetText(sc,10);
    Val(sc,m,err);
    massainicial2 := Round(m * 10);
    If massainicial2 < 10 then massainicial2 := 10; { Limita a massa inicial = 1
u. }
    mi2 := massainicial2;
    massafin2^.GetText(sc,10);
    Val(sc,m,err);
    massafinal2 := Round(m * 10);
    If massafinal2 < (massainicial2 + 100) then massafinal2 := massainicial2
+ 100;
    If massafinal2 > 10000 then massafinal2 := 10000;{ Limita a massa final
= 1000 u.}
    mf2 := massafinal2;
    Tempoint^.GetText(sc,10);
    Val(sc,tempo,err);

```

```

If tempo < 0 then tempo := 0
else If tempo > 33.4 then tempo := 33.4;
Enion^.GetText(sc,10);
Val(sc,energion,err);
If energion < 5.0 then energion := 5.0
else If energion > 100.0 then energion := 100.0;
lon^.El(energion);
coment^.GetText(sc,80);
comentario := StrPas(sc);
Mostra;
Str(quadrupolo2,s);
Str(quadrupolo1,s1);
s := 'Coloque Q' + s + ' e Q' + s1 + ' no modo "computer"';
StrPCopy(sc,s);
MessageBox(HWindow,sc,'Aquisição',mb_Ok or mb_IconInformation);
InvalidateRect(Grafico^.HWindow,nil,TRUE); { Limpa o grafico. }
temdados := TRUE; { Garante que existem dados para serem
desenhados. }
Grafico^.adquirido := FALSE; { Garante que ainda não foi adquirido. }
Grafico^.acabou := FALSE;
StrCopy(FileName,"");
maxy := 1000;
nomearq := pathdados + nomedefault;
End;

```

```

Procedure TAquisicao.CMFileOpen(var Msg: TMessage; titulo,tip,DefExt :
PChar);
var
    OpenFN : TOpenFileName;
    Filter : array [0..100] of Char;
    FullFileName: TFilename;
Begin
    StrCopy(FullFileName, "");
    FillChar(Filter, SizeOf(Filter), #0);
    StrCopy(Filter, titulo);
    StrCopy(@Filter[StrLen(Filter)+1], tip);
    FillChar(OpenFN, SizeOf(TOpenFileName), #0);
    With OpenFN do
    begin
        hInstance := HInstance;
        hwndOwner := HWindow;
        lpstrDefExt := DefExt;
        lpstrFile := FullFileName;
        lpstrFilter := Filter;
        lpstrFileTitle := FileName;
        flags := ofn_FileMustExist;
        IStructSize := sizeof(TOpenFileName);
        nFilterIndex := 1; { Índice dentro da String de filtragem em lpstrFilter. }
        nMaxFile := SizeOf(FullFileName);
    end;
    Case salva_recupera of
        cm_salva : ok_arquivo := GetSaveFileName(OpenFN);
        cm_recupera : ok_arquivo := GetOpenFileName(OpenFN);
    end; { case }
End;

```

```

Procedure TAquisicao.Salvar(var Msg: TMessage);
var
    sc : Array[0..79] of char;
Begin
    Salva_Recupera := cm_salva;

```

```

        If temdados then CMFileOpen(Msg,'Espectros
Tridimensionais','*.BID','bid');
        If not ok_arquivo then EXIT;
        coment^.GetText(sc,80);
        comentario := StrPas(sc);
        nomearq := StrPas(filename);
        nomearq := Copy(nomearq,1,pos('.',nomearq)-1);
        nomearq := nomearq + '.BID';
        Assign(arq,nomearq);
        ($I-) Reset(arq); {$I+}
        If IOResult = 0 then
            begin
                StrPCopy(sc,'Arquivo ' + nomearq + ' já existe. Sobrepor?');
                If MessageBox(HWindow,sc,'Salvar', mb_defbutton2 or
                    mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
            end;
        Rewrite(arq);
        Assign(arq1,pathdados + nomedefault);
        Reset(arq1);
        Repeat
            Read(arq1,registro);
            Write(arq,registro);
        Until EOF(arq1);
        Close(arq);
        Close(arq1);
        nomearq := Copy(nomearq,1,pos('.',nomearq)-1);
        nomearq := nomearq + '.PAR';
        With parametros do
            begin
                m1i := massainicial1;
                m1f := massafinal1;
                m2i := massainicial2;
                m2f := massafinal2;
                g := ganhogeral;
                inc1 := inc1;
                ni := ndadosi;
                nj := ndadosj;
                quad1 := quadrupolo1;
                quad2 := quadrupolo2;
                tint := tempo;
                max := maximo;
                comen := comentario;
            end;
        Assign(arqpar,nomearq);
        Rewrite(arqpar);
        Write(arqpar,parametros);
        Close(arqpar);
        nomearq := pathdados + nomedefault;
    End;

```

```

Procedure TAquisicao.Recuperar(var Msg: TMessage);
var dir : Array[0..80] of Char;
Begin
    Salva_Recupera := cm_recupera;
    CMFileOpen(Msg,'Espectros Tridimensionais','*.BID','bid');
    If not ok_arquivo then EXIT;
    GetCurDir(dir,0);
    nomearq := StrPas(dir);
    If nomearq[Length(nomearq)] = '\' then nomearq := nomearq +
StrPas(filename)
    else nomearq := nomearq + '\' + StrPas(filename);

```

```

nomearq := Copy(nomearq,1,pos('.',nomearq)-1);
nomearq := nomearq + '.PAR';
Assign(arqpar,nomearq);
ReSet(arqpar);
Read(arqpar,parametros);
With parametros do
begin
    massainicial1 := m1i;
    mi1 := m1i;
    massafinal1 := m1f;
    mf1 := m1f;
    massainicial2 := m2i;
    mi2 := m2i;
    massafinal2 := m2f;
    mf2 := m2f;
    ganhogeral := g;
    incr1 := inc1;
    ndadosi := ni;
    ndadosj := nj;
    quadrupolo1 := quad1;
    quadrupolo2 := quad2;
    tempo := tint;
    maximo := max;
    comentario := comen;
end;
Close(arqpar);
nomearq := Copy(nomearq,1,pos('.',nomearq)-1);
nomearq := nomearq + '.BID';
Mostra;
temdados := TRUE;
Grafico^.adquirido := TRUE;
InvalidateRect(HWindow, nil, TRUE);
End;

Procedure TAquisicao.Copia(var Msg : TMessage);
Begin
    CopiaBitmap(HWindow,Grafico^.DC,560,260);
End;

Procedure TAquisicao.Print(var Msg : TMessage);
var PlInfo : TPaintStruct;
    imp : String;
    sc : Array[0..79] of Char;
    NewFont, OldFont : HFont;
Begin
    coment^.GetText(sc,80);
    comentario := StrPas(sc);
    If temdados and (MessageBox(HWindow,'Imprime o espectro?', 'Imprimir',
        mb_YesNo or mb_IconQuestion) = IDYes) then
    begin
        imprimir := TRUE;
        If PrmPartida('Espectro Tridimensional',imp) then
        begin
            { Verifica se a impressora é HP. }
            If (imp[1] = 'H') and (imp[2] = 'P') then
            begin
                Grafico^.xLo := 3 * 560; { Valores para as impressoras lasers. }
                Grafico^.yHi := 3 * 260;
                Grafico^.xLo := 100;
            end;
            NewFont := CreateFont(20,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,

```

```

OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
    VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
    OldFont := SelectObject(PDc,NewFont);
    SetTextAlign(PDc,TA_Right);

    TextOut(PDc,Grafico^.xHi,Grafico^.yHi+90,filename,StrLen(filename));
    SelectObject(PDc,OldFont);
    DeleteObject(NewFont);
    Grafico^.Paint(PDc,PlInfo);
    NovaPagina;
    PrmFim;
    Grafico^.xLo := 20; { Valores para impressoras matriciais. }
    Grafico^.xHi := 560;
    Grafico^.yHi := 260;
end;
end;
imprimir := FALSE;
End;

Procedure TAquisicao.Expandir(var Msg : TMessage);
Begin
    maxy := expansao;
    mi1 := expmi;
    mf1 := expmf;
    mi2 := expmiz;
    mf2 := expmfz;
    InvalidateRect(Grafico^.HWindow,nil,TRUE); { Redesenha o espectro
    expandido. }
End;

Procedure TAquisicao.Normal(var Msg : TMessage);
Begin
    maxy := 1000;
    mi1 := massainicial1;
    mf1 := massafinal1;
    mi2 := massainicial2;
    mf2 := massafinal2;
    InvalidateRect(Grafico^.HWindow,nil,TRUE); { Redesenha o espectro
    normal. }
End;

Procedure TAquisicao.Sobre(var Msg : TMessage);
Begin
    MessageBox(HWindow,'Desenvolvido por Valmir F. Juliano - 1994/95',
        'Programa PENTA3D - Versão 1.5',mb_Ok or mb_IconInformation);
End;

Procedure TAquisicao.Fim(var Msg : TMessage);
Begin
    CloseWindow;
End;

Procedure TAquisicao.Cancela(var Msg : TMessage);
Begin
    Grafico^.acabou := TRUE;
End;

Function TAquisicao.CanClose: Boolean;
Begin
    Dispose(intens);

```

```

Dispose(massa2);
Dispose(massa1);
Dispose(lon,done);
DoneWinCrit;
CanClose := TRUE;
End;

Procedure TAquisicao.GraficoUni(var Msg : TMessage);
var
  JF : Array[1..200] of PGrafico;
  titulo : Array[0..20] of Char;
  s1,s2 : String[5];
  j,k : Word;
  nome : String;
Begin
  If not temdados then EXIT;
  temdados := FALSE;
  nome := Copy(nomearq,1,pos('.',nomearq)-1);
  Assign(arqpar,nome + '.PAR');
  Reset(arqpar);
  Read(arqpar,parametros);
  Close(arqpar);
  Assign(arq,nome + '.BID');
  Reset(arq);
  posj := -30;
  For j := 1 to parametros.nj do
    begin
      inc(posj,30);
      Read(arq,registro);
      Str(j,s1);
      Str(registro.m2/10.0:0:1,s2);
      StrPCopy(titulo,s1 + ' - ' + nome + ': m/z = ' + s2);
      JF[j] := PGrafico(Aplica.MakeWindow(New(PGrafico,init(@Self,titulo))));
    end;
  Close(arq);
End;

```

```

Constructor TGrafico.Init(AParent: PWindowsObject; ATitle: PChar);
var
  k : Word;
Begin
  ButtonDown := FALSE;
  xHi := 400;
  xLo := 30;
  yHi := 160;
  yLo := 30;
  maxy := 1000;
  TWindow.Init(AParent,ATitle);
  With Attr do
    begin
      Style := ws_PopUpWindow OR ws_Caption OR ws_Visible;
      X := 40 * (posj div 600); Y := posj mod 600;
      W := xHi + xLo + 10; H := yHi + yLo + 30;
      Menu := LoadMenu(HInstance,'menu_3');
    end;
  With parametros do
    begin
      njanelas := nj;
      ndados := ni;
      maximo := max;
      expansao := 1000;

```

```

mi := m1i;
massainicial := m1i;
mf := m1f;
massafinal := m1f;
For k := 1 to ni do
  begin
    massa[k] := registro.m1[k];
    Intens[k] := registro.i[k];
  end;
end;
End;

Function TGrafico.GetClassName: PChar;
begin
  GetClassName := 'EspectroUni'
end;

Procedure TGrafico.GetWindowClass(var AWndClass: TWndClass);
Begin
  TWindow.GetWindowClass(AWndClass);
  AWndClass.Style := AWndClass.Style or cs_DblClks;{ Aceita click duplo
do mouse.}
  AWndClass.HIcon := LoadIcon(HInstance,'uni');
End;

Procedure TGrafico.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
  NewCur, OldCur : HCursor;
  OldFont,
  NewFont : HFont;
  i,j,xi,increm,posx,posy,oy : Integer;
  yi,inten : Word;
  s : string[95];
  sc : Array[0..94] of Char;

Function Y(i : Longint) : Integer;
Begin
  Y := Round(1000.0*/maximo);
End;

Begin { TGrafico.Paint }
  NewCur := LoadCursor(0,idc_Wait); { Sinaliza com a ampulheta. }
  OldCur := SetCursor(NewCur);
  SetTextAlign(PaintDC,TA_RIGHT);
  SetTextColor(PaintDC,RGB(0,0,255));
  SetBkMode(PaintDC,transparent);
  SetMapMode(PaintDC,mm_Anisotropic);
  SetViewportOrg(PaintDC,xLo,yHi-yLo);
  SetViewportExt(PaintDC,xHi-xLo,yLo-yHi);
  SetWindowExt(PaintDC,mf-mi,maxy);
  SetWindowOrg(PaintDC,mi,0);
  oy := -maxy div 50; { Deslocamento da linha base em 2%. }
  NewFont :=
CreateFont(Round(0.06*maxy),0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,Tms Rmn');
  OldFont := SelectObject(PaintDC,NewFont);
  increm := 100*(1 + Trunc((mf - mi)/1000.0 - 0.49)); { Incremento da escala
de massa. }
  If (mf - mi) div 10 > 80 then posX := mi - increm div 10

```

```

else If (mf - mi) div 10 > 60 then posx := mi - increm div 15
    else If (mf - mi) div 10 > 40 then posx := mi - increm div 20
        else If (mf - mi) div 10 > 20 then posx := mi - increm div 40
            else posx := mi - increm div 60; { Posição x p/ intensidade. }
Str(maxy div 10,s);
StrPCopy(sc,s);          { Numera as intensidades. }
TextOut(PaintDC,posx,maxy,sc,StrLen(sc));
Str(maxy div 20,s);
StrPCopy(sc,s);
TextOut(PaintDC,posx,maxy div 2,sc,StrLen(sc));
TextOut(PaintDC,posx,0,'1');
NewPen := CreatePen(ps_Solid,0,RGB(0,0,0));
OldPen := SelectObject(PaintDC,NewPen);
MoveTo(PaintDC,mi,maxy);
LineTo(PaintDC,mi,oy);    { Desenha os eixos. }
LineTo(PaintDC,mf,oy);
Str(mi div 10,s);
StrPCopy(sc,s);
posy := -maxy div 15;
TextOut(PaintDC,mi,posy,sc,StrLen(sc)); { Numera a 1a massa. }
MoveTo(PaintDC,mi,posy);
LineTo(PaintDC,mi,oy);
i := 0;
xi := mi;
Repeat
    posy := -maxy div 25;
    inc(i);
    While (xi < mf + (i + 1) * increm) and (xi < mf) do
        begin
            xi := xi + increm div 10;
            MoveTo(PaintDC,xi,posy); { Subdivisões da escala de massa. }
            LineTo(PaintDC,xi,oy);
        end;
    xi := mi + i * increm; { Evita acúmulo de sobras de arredondamento. }
    Str(xi div 10,s);
    StrPCopy(sc,s);
    posy := -maxy div 15;
    TextOut(PaintDC,xi,posy,sc,StrLen(sc)); { Numera as massas. }
    MoveTo(PaintDC,xi,posy);
    LineTo(PaintDC,xi,oy);
Until xi > mf - increm;
SetTextAlign(PaintDC,TA_CENTER);
StrPCopy(sc,comentario);
TextOut(PaintDC,mi + (mf - mi) div 2,Round(-0.25*maxy),sc,StrLen(sc));
For i := 0 to 10 do
    begin
        yi := Round(i * maxy / 10.0);
        MoveTo(PaintDC,posx,yi); { Escala de intensidade. }
        LineTo(PaintDC,mi,yi)
    end;
SelectObject(PaintDC,OldFont);
DeleteObject(NewFont);
SelectObject(PaintDC,OldPen);
DeleteObject(NewPen);
NewPen := CreatePen(ps_Solid,0,RGB(255,0,0));
OldPen := SelectObject(PaintDC,NewPen);
MoveTo(PaintDC,mi,0);
If maximo > 0 then
    For i := 1 to ndados do { Traça o espectro normalizando as
intensidades. }
        If (massa[i] >= mi) and (massa[i] <= mf) then

```

```

        LineTo(PaintDC,massa[i],Y(intens[i]));
SetCursor(OldCur);
Dc := PaintDC;
End;

Procedure TGráfico.WMLButtonDown(var Msg : TMessage);
Begin
    If not ButtonDown then
        begin
            mousexi := xLo;    { Valores iniciais para a posição do mouse. }
            mousexf := xHi;
            mouseyf := 0;
            SetCapture(HWindow); { Habilita a janela a receber informações do
mouse. }
            Dc := GetDC(HWindow); { Abre um manipulador de contexto. }
            mousexi := Msg.LParamLo; { Obtém a nova posição do mouse. }
            mouseyi := yHi - yLo;
            MoveTo(Dc,mousexi,mouseyi);
            NewPen := CreatePen(ps_Dot,0,RGB(0,128,0)); { Linha pontilhada
verde. }
            OldPen := SelectObject(Dc,NewPen);
            ButtonDown := TRUE;
        end;
End;

```

```

Procedure TGráfico.WMLButtonUp(var Msg : TMessage);
var troca : Word;
Begin
    If ButtonDown then
        begin
            LineTo(Dc,mousexf,mouseyi); { Desenha o retângulo que define a
região que }
            LineTo(Dc,mousexf,mouseyf); { poderá ser ampliada. }
            LineTo(Dc,mousexi,mouseyf);
            LineTo(Dc,mousexi,mouseyi); { Define os valores para a expansão. }
            expansao := Word(Round(maxy * (mouseyf - yHi + yLo) / (yLo - yHi)));
            If expansao > 1000 then expansao := 1000;
            If expansao < 50 then expansao := 50; { intensidade mínima = 5% }
            expmi := Word(10*Trunc((mi+(mousexi - xLo) * (mf - mi))/(xHi -
xLo))/10.0));
            If expmi < massainicial then expmi := massainicial;
            expmf := Word(10*Trunc((mi+(mousexf - xLo) * (mf - mi))/(xHi -
xLo))/10.0));
            If expmf > massafinal then expmf := massafinal;
            If expmf <= expmi then
                begin
                    troca := expmf;
                    expmf := expmi;
                    expmi := troca;
                end;
            If (expmf - expmi) < 100 then expmf := expmi + 100;
            SelectObject(Dc,OldPen);
            DeleteObject(NewPen);
            ReleaseCapture; { Libera a captura do mouse para outras janelas. }
            ReleaseDC(HWindow,Dc); { Libera o manipulador de contexto. }
            ButtonDown := FALSE;
        end;
End;

```

```

Procedure TGráfico.WMRButtonDown(var Msg : TMessage);
Begin

```



```

mi := expmi;
mf := expmf;
maxy := expansao;
InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro expandido.
}
End;

```

```

Procedure TGráfico.WMMouseMove(var Msg : TMessage);
Begin
If ButtonDown then With Msg do
begin
mousexf := Integer(LParamLo); { Captura as posições do mouse em
movimento.}
mouseyf := Integer(LParamHi);
end;
End;

```

```

Procedure TGráfico.WMLButtonDbClk(var Msg : TMessage);
Begin
maxy := 1000;
mi := massainicial;
mf := massafinal;
InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro após o
duplo click. }
End;

```

```

Procedure TGráfico.Copia(var Msg : TMessage);
Begin
InvalidateRect(HWindow, nil, TRUE);
CopiaBitmap(HWindow, DC, 410, 155)
End;

```

```

Procedure TGráfico.Print(var Msg : TMessage);
var Pinfo : TPaintStruct;
imp : String;
NewFont, OldFont : HFont;
Begin
If MessageBox(HWindow, 'Imprime o espectro?', 'Imprimir',
mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
If PrnPartida('Espectro Unidimensional', imp) then
begin
{ Verifica se a impressora é HP. }
If (imp[1] = 'H') and (imp[2] = 'P') then
begin
xLo := 100;
xHi := 1500; { Valores para as impressoras lasers. }
yHi := 600;
end;
NewFont := CreateFont(20, 0, 0, 0, FW_BOLD, 0, 0, 0, ANSI_CHARSET,

```

```

OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN, 'Tms Rmn');
OldFont := SelectObject(PDc, NewFont);
SetTextAlign(PDc, TA_Right);
TextOut(PDc, xHi, yHi+90, Attr.Title, StrLen(Attr.Title));
SelectObject(PDc, OldFont);
DeleteObject(NewFont);
Paint(PDc, Pinfo);
NovaPagina;
PmFim;
xLo := 30; { Valores para o video e para as impressoras matriciais. }
xHi := 400;

```

```

yHi := 160;
end;
End;

```

```

Procedure TGráfico.Salva(var Msg : TMessage);
var
i : Word;
arqtext : Text;
nome : String;
sc : Array[0..100] of Char;
Begin
nome := StrPas(Attr.Title);
nome := Copy(nomearq, 1, pos('.', nomearq)-3) +
Copy(nome, 1, pos('-', nome)-2) + '.DAT';
If MessageBox(HWindow, 'Salva o espectro?', 'Salvar',
mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
Assign(arqtext, nome);
{$I-} Reset(arqtext); {$I+}
If IOResult = 0 then
begin
StrPCopy(sc, 'Arquivo ' + nome + ' já existe. Sobrepor?');
If MessageBox(HWindow, sc, 'Salvar', mb_defbutton2 or
mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
end;
ReWrite(arqtext);
WriteLn(arqtext, maximo);
WriteLn(arqtext, StrPas(Attr.Title));
WriteLn(arqtext, quadrupolo2);
WriteLn(arqtext, ganhogeral);
WriteLn(arqtext, tempo:0:2);
For i := 1 to ndados do WriteLn(arqtext, massa[i]/10.0:0:1, #9, intens[i]);
Close(arqtext);
nome := StrPas(Attr.title) + #13 + 'foi salvo como: ' + nome;
StrPCopy(sc, nome);
MessageBox(HWindow, sc, 'Salvar', mb_Ok or mb_IconInformation);
End;

```

```

Function TGráfico.CanClose;
Begin
Dec(njanelas);
If njanelas = 0 then temdados := TRUE;
End;

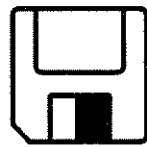
```

```

BEGIN
Aplica.Init('Aplica');
Aplica.Run;
Aplica.Done;
END.

```

PENTA4D



```
Program PENTA4D;
{-----}
    Programa para aquisição de espectros bi, tri e
    tetradimensionais do espectrômetro de massas pentaquadrupolar.
    Utiliza os métodos de programação orientada a objetos e
    programação motivada por eventos.
    Desenvolvido por Valmir Fascio Juliano como parte do
    trabalho de doutoramento no Instituto de Química - UNICAMP sob
    orientação da Profa. Dra. Concetta Kascheres e do Prof. Dr. Claudimir
    Lúcio do Lago (IQ - USP).
{-----}
```

```
{ Compilar com Stack Size = 8192 e Heap Size = 8192 }
```

```
Uses UPenta4D, WinProcs, WObjects, WinTypes, Strings, WinDos,
MPrint;
```

```
($R Penta)
```

```
Const
    path = 'C:\PENTA\';
    pathdados = path + 'QSESP\';
    id_EditBox = -1;
    id_quad = 101;
    id_g = 102;
    id_PbOk = 103;
    id_PbFim = 104;
    id_PbCancela = 105;
    id_sens = 106;
    id_GroupSens = 107;
    cm_salva = 1;
    cm_recupera = 2;
    cm_Copia = 3;
    cm_Print = 4;
    cm_Expande = 5;
    cm_Normal = 6;
    cm_Sobre = 7;
```

```
Type
    PentaQuadrupolo = Object(TApplication)
        Procedure InitMainWindow; virtual;
    end;
    PPrincipal = ^TPrincipal;
    TPrincipal = Object(TWindow)
        g : PComboBox;
        quadru : Array[um..cinco] of PCheckBox;
        senslo, sensmed, senshi : PRadioButton;
        GroupSens : PGroupBox;
        coment, Enlon : PEdit;
        massain, massafin : Array[um..cinco] of PEdit;
        PbOk, PbFim, PbCancela : PButton;
        Constructor Init(AParent: PWindowsObject; ATitle: PChar);
        Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
        Function GetClassName: PChar; virtual;
        Procedure SetupWindow; virtual;
        Procedure Acquire(var Msg: TMessage); virtual id_First + id_PbOk;
        Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
        Procedure Grafico(var Msg: TMessage); virtual cm_First + cm_copia;
        Function CanClose: Boolean; virtual;
        Procedure Fim(var Msg: TMessage); virtual id_First + id_PbFim;
    end;
```

```
PGrafico = ^TGrafico;
TGrafico = object(TWindow)
    ButtonDown, converteu: Boolean;
    mousexi, mousexf, mouseyi, mouseyf,
    xHi, xLo, yHi: Integer;
    Dc: HDC;
    NewPen, OldPen: HPen;
    mi, mf, maxy, ndados, expansao, expmi, expmf,
    massainicial, massafinal: Word;
    maximo: LongInt;
    massa: Array[1..1000] of Word;
    Intens: Array[1..1000] of LongInt;
    Constructor Init(AParent: PWindowsObject; ATitle: PChar);
    Function GetClassName: PChar; virtual;
    Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
    Procedure Converte;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
    Procedure WMLButtonDown(var Msg: TMessage); virtual wm_First +
    wm_LButtonDown;
    Procedure WMLButtonUp(var Msg: TMessage); virtual wm_First +
    wm_LButtonUp;
    Procedure WMRButtonDown(var Msg: TMessage); virtual wm_First +
    wm_RButtonDown;
    Procedure WMMouseMove(var Msg: TMessage); virtual wm_First +
    wm_MouseMove;
    Procedure WMLButtonDbIClk(var Msg: TMessage); virtual wm_First +
    wm_LButtonDbIClk;
    end;
    Tident = Array[0..9] of Char;
    TPico = Record
        Case Integer of
            0 : (m: Word; h, v: Single);
            1 : (mq: Array[um..cinco] of Integer; npicos: Word);
            2 : (ident: Tident);
            3 : (ndesvio, ndexplora: Integer);
        end;
    TESpectro = Array[1..1000] of TPico;
    PESpectro = ^TESpectro;
```

```
var
    Penta: PentaQuadrupolo;
    PentaQ: PPentaQuad;
    mi, mf: Array[um..cinco] of Double;
    varreQ: Array[um..cinco] of Boolean;
    nomearq: String[8];
    Energialon: Double;
    cabeca,
    identidade: TPico;
    temdados: Boolean;
    arquivo: PEdit;

    Procedure PentaQuadrupolo.InitMainWindow;
    Begin
        { Inicializa o Programa. }
        MainWindow := New(PPrincipal, Init(nil, "TETRADIMENSIONAL - (C) VFJ
        1994"));
    End;

    Constructor TPrincipal.Init(AParent: PWindowsObject; ATitle: PChar);
    var
        WinDir: array [0..145] of Char;
        q: Quadrupolos;
```

```
Begin
TWindow.Init(AParent,ATitle);
EnableKBHandler;
with Attr do
begin
X := CW_USEDEFAULT; Y := CW_USEDEFAULT; W := 950; H := 440;
Menu := LoadMenu(HInstance,'menu_2');
end;
{ g := New(PComboBox, Init(@Self, id_g,
190,75,70,100,cbs_DropDownList,0));
g^.Attr.Style := g^.Attr.Style and (not cbs_Sort); }
quadru[um] := New(PCheckBox, Init(@Self, id_quad, '1',210,30,30,26,
nil));
quadru[tres] := New(PCheckBox, Init(@Self, id_quad, '3',250,30,30,26,
nil));
quadru[cinco] := New(PCheckBox, Init(@Self, id_quad, '5',290,30,30,26,
nil));
massain[um] :=
New(PEdit,Init(@Self,id_EditBox,'20',185,70,40,30,4,FALSE));
massafin[um] :=
New(PEdit,Init(@Self,id_EditBox,'200',185,110,50,30,5,FALSE));
massain[tres] :=
New(PEdit,Init(@Self,id_EditBox,'20',245,70,40,30,4,FALSE));
massafin[tres] :=
New(PEdit,Init(@Self,id_EditBox,'200',245,110,50,30,5,FALSE));
massain[cinco] :=
New(PEdit,Init(@Self,id_EditBox,'20',305,70,40,30,4,FALSE));
massafin[cinco] :=
New(PEdit,Init(@Self,id_EditBox,'200',305,110,50,30,5,FALSE));
Enlon := New(PEdit,Init(@Self,id_EditBox,'70.0',205,150,45,30,5,FALSE));
GroupSens := New(PGroupBox, Init(@Self, id_GroupSens,"0,0,0,0));
senslo := New(PRadioButton, Init(@Self, id_sens,
'B',190,190,40,20,GroupSens));
sensmed := New(PRadioButton, Init(@Self, id_sens,
'M',230,190,40,20,GroupSens));
senshi := New(PRadioButton, Init(@Self,
id_sens,'A',270,190,40,20,GroupSens));
arquivo := New(PEdit,Init(@Self,id_EditBox,"",90,240,165,30,9,FALSE));
PbOK :=
New(PButton,Init(@Self,id_PbOK,'INICIA',20,320,110,30,FALSE));
PbFim := New(PButton,Init(@Self,id_PbFim,'SAI',140,320,50,30,TRUE));
PbCancela :=
New(PButton,Init(@Self,id_PbCancela,'CANCELA',200,320,100,30,FALSE
));
coment := New(Pedit,Init(@Self,id_EditBox,"",360,320,520,30,61,FALSE));
StrPCopy(WinDir,path+pathdados);
SetCurDir(WinDir);
New(PentaQ,Init(path,'Q1.CAL','Q3.CAL','Q5.CAL'));
For q := um to cinco do
begin
mi[q] := 20.0;
mf[q] := 180.0;
end;
varreQ[um] := FALSE;
varreQ[tres] := FALSE;
varreQ[cinco] := TRUE;
temdados := FALSE;
nomearq := 'sem_nome';
cabeca.ndesvio := 2;
cabeca.ndexplora := 10;
identidade.ident[0] := #0;
```

```
End;

Procedure TPrincipal.GetWindowClass(var AWndClass : TWndClass);
Begin
TWindow.GetWindowClass(AWndClass);
AWndClass.HIcon := LoadIcon(HInstance,'trid');
End;

Function TPrincipal.GetClassName: PChar;
Begin
GetClassName := 'MainWndClass'
End;

procedure TPrincipal.SetupWindow;
var
sz : Array[0..8] of Char;
Begin
TWindow.SetupWindow;
{ g^.AddString('1');
g^.AddString('2');
g^.AddString('5');
g^.AddString('10');
g^.AddString('20');
g^.AddString('50');
g^.AddString('100');
g^.SetSelIndex(5);
quadru[cinco]^^.SetCheck(1);
sensmed^.SetCheck(1);
StrPCopy(sz,nomearq);
arquivo^.SetText(sz);
End;

Procedure TPrincipal.Acquire(var Msg : TMessage);
var i, l : Word;
s1,s2 : String;
sc : Array[0..79] of Char;
arq : File of TPico;
DC: HDC;
PS: TPaintStruct;

Function GetParam : Boolean;
var
erro : Integer;
q : Quadruolos;
Begin
GetParam := TRUE;
With PentaQ^do
begin
Enlon^.GetText(sc,10);
Val(sc,Energialon,erro);
If Energialon < 5.0 then Energialon := 5.0
else If Energialon > 100.0 then Energialon := 100.0;
Ionizador.El(Energialon);
If senslo^.GetCheck = 1 then Port[$318] := 1;
If sensmed^.GetCheck = 1 then Port[$318] := 3;
If senshi^.GetCheck = 1 then Port[$318] := 2;
arquivo^.GetText(sc,9);
nomearq := StrPas(sc);
If nomearq = "" then nomearq := 'sem_nome';
For q := um to cinco do
begin
```

```

varreQ[q] := quadru[q]^^.GetCheck = 1;
massain[q]^^.GetText(sc,4);
Val(sc,mi[q],erro);
massafin[q]^^.GetText(sc,5);
Val(sc,mf[q],erro);
end;
{ g^..GetSelString(sc,3);
Val(sc,det.ganho,erro);}
If not varreQ[um] and not varreQ[tres] and not varreQ[cinco] then
begin
MessageBeep(MB_ICONEXCLAMATION);
MessageBox(HWindow,'Não foi selecionado nenhum quadrupolo.',
'ERRO NA AQUISIÇÃO',MB_OK or MB_ICONINFORMATION);
GetParam := FALSE;
end;
end;
End;

```

```

Procedure Scan(q : Quadrupolos); { Procedimento recursivo para varredura.
}
var
mz, mzi, mzf, np : Word;
pico : PEspectro;
pre : TPico;
u : Quadrupolos;

```

```

Procedure RepeteScan;
var
mz : Word;
Begin
np := 0;
For mz := mzi to mzf do With pico^[mz] do With PentaQ^ do
begin
SetMassa(q,mz);
m := mz;
LeMedia(5,h,v);
If saturado then EXIT;
Case q of
um : TextOut(DC,500,180,'Q1 - Varrendo...',16);
tres : TextOut(DC,500,200,'Q3 - Varrendo...',16);
cinco : TextOut(DC,500,220,'Q5 - Varrendo...',16);
end;
If h > cabeca.ndesvio * sqrt(v) then inc(np);
end;
End;

```

```

Begin { Scan }
New(pico);
If VarreQ[q] then with PentaQ^do
begin
SetPoleDC(q,TRUE);
mzi := Round(mi[q]);
mzf := Round(mf[q]);
ganho := 7;
det.ganho := ganhos[ganho];
Repeat { Repete a varredura até encontrar um }
If saturado then { ganho que não sature o ADC. }
begin
dec(ganho);
det.ganho := ganhos[ganho];
end;

```

```

RepeteScan;
Until (not saturado) or (det.ganho = 1);
SetMassa(q,mi[q]);
With pre do
begin
npicos := np;
For u := um to cinco do
If not poleDC[u] then mq[u] := -1 else { m=-1 -> poleDC em off. }
If u <> q then mq[u] := Round(GetMassa(u)){ m=x -> massa fixa. }
else mq[u] := 0; { m=0 -> varrendo. }
end;
Write(arq,pre);
For mz := mzi to mzf do
If pico^[mz].h > cabeca.ndesvio * sqrt(pico^[mz].v) then
Write(arq,pico^[mz]);
If q < cinco then
For mz := mzi to mzf do
If pico^[mz].h > cabeca.ndexplora * sqrt(pico^[mz].v) then
begin
SetMassa(q,mz); { Estabelece a m/z a ser explorada. }
Otimiza(q,mz,0.25); { Localiza o máximo. }
Scan(Succ(q)); { Varre o próximo quadrupolo. }
end;
end
else
If q < cinco then Scan(Succ(q));
Dispose(pico);
End;

```

```

Begin { Adquire }
temdados := FALSE;
If not GetParam then EXIT;
With PentaQ^ do
begin
While not LinhaBase do
begin
Str(lbmedia[7]:0:1,s1);
Str(sqrt(lbvar[7]):0:1,s2);
s1 := 'Linha Base muito baixa: média=' + s1 + ', desvio=' + s2 + '#13
+
'Utilize "COMP0" no painel de Q1 para elevar a Linha Base.'
+ '#13 + 'CONTINUA?';
StrPCopy(@sc,s1);
MessageBeep(MB_ICONEXCLAMATION);
If MessageBox(HWindow,@sc,'ERRO NA AQUISIÇÃO',
MB_YESNO or MB_ICONSTOP) = IDNO then EXIT;
end;
SetPoleDC(um,FALSE);
SetPoleDC(tres,FALSE);
SetPoleDC(cinco,FALSE);
Assign(arq,pathdados + nomearq + '.tri');
ReWrite(arq);
Write(arq,identidade);
Write(arq,cabeca);
InvalidateRect(HWindow, nil, FALSE);
DC := BeginPaint(HWindow, PS);
Scan(um);
EndPaint(HWindow, PS);
Close(arq);
InvalidateRect(HWindow,nil, TRUE);
MessageBeep(MB_ICONINFORMATION);

```

```

    MessageBox(HWindow,'Aquisição terminada.','AQUISIÇÃO',MB_OK or
MB_ICONINFORMATION);
    temdados := TRUE;
    end;
End;

Procedure TPrincipal.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
    NewFont;
    OldFont : HFont;
Begin
    NewFont := CreateFont(18,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
    VARIABLE_PITCH OR FF_ROMAN,Tms Rmn');
    OldFont := SelectObject(PaintDC,NewFont);
    SetTextColor(PaintDC,RGB(0,0,128));
    SetBkMode(PaintDC,transparent);
    SetTextAlign(PaintDC,TA_LEFT);
    TextOut(PaintDC,100,5,'AQUISIÇÃO E MANIPULAÇÃO DE ESPECTROS
DO ESPECTRÔMETRO DE MASSAS PENTAQUADROPOLAR',80);
    TextOut(PaintDC,25,35,'Quadrupolos de varredura:',25);
    TextOut(PaintDC,25,75,'Massas iniciais:',16);
    TextOut(PaintDC,25,115,'Massas finais:',14);
    TextOut(PaintDC,25,155,'Energia de ionização (eV):',26);
    TextOut(PaintDC,25,195,'Sensibilidade: ',15);
    TextOut(PaintDC,25,245,'Arquivo:',8);
    TextOut(PaintDC,300,325,'Comentário:',11);
    SelectObject(PaintDC,OldFont);
    DeleteObject(NewFont);
End;

Procedure TPrincipal.Grafico(var Msg : TMessage);
var
    JF : PGrafico;
    sc : Array[0..80] of Char;
Begin
    arquivo^.GetText(sc,9);
    nomearq := StrPas(sc);
    StrPCopy(sc,nomearq);
    JF := PGrafico(Penta.MakeWindow(New(PGrafico,Init(@Self,sc))));
End;

Function TPrincipal.CanClose : Boolean;
Begin
    Dispose(PentaQ.Done);
    CanClose := TRUE;
End;

Procedure TPrincipal.Fim(var Msg : TMessage);
Begin
    CloseWindow;
End;

Constructor TGrafico.Init(AParent: PWindowsObject; ATitle: PChar);
Begin
    ButtonDown := FALSE;
    xHi := 560;
    xLo := 30;
    yHi := 260;
    maxy := 1000;

```

```

    converteu := FALSE;
    TWindow.Init(AParent,ATitle);
    With Attr do
        begin
            Style := ws_PopUp OR ws_overlappedWindow OR ws_Visible;
            X := 330; Y := 45; W := 600; H := 300;
            { Menu := LoadMenu(HInstance,'menu_2');}
        end;
    End;

Function TGrafico.GetClassName: PChar;
begin
    GetClassName := 'GraphWndClass'
end;

Procedure TGrafico.GetWindowClass(var AWndClass: TWndClass);
Begin
    TWindow.GetWindowClass(AWndClass);
    AWndClass.Style := AWndClass.Style or cs_DblClks;{ Aceita click duplo
do mouse }
    AWndClass.HIcon := LoadIcon(HInstance,'trid');
End;

Procedure TGrafico.Converte;
var
    pico : PEspectro;
    arq : File of TPico;
    cabeca;
    identidade,pre : TPico;
    i : Word;
    sc : Array[0..8] of Char;
Begin
    New(pico);
    Assign(arq,pathdados + nomearq + '.P4D');
    Reset(arq);
    Read(arq,identidade);
    Read(arq,cabeca);
    Read(arq,pre);
    maximo := 0;
    For i := 1 to pre.npico do
        begin
            Read(arq,pico^[i]);
            With pico^[i] do
                begin
                    massa[i] := 10*m;
                    intens[i] := Round(h);
                    If maximo < intens[i] then maximo := intens[i];
                end;
            end;
        end;
    Close(arq);
    ndados := pre.npico;
    mi := 100 * (massa[1] div 100) ;
    mf := 100 * (massa[ndados] div 100 + 1); {??? <- deu pau aqui}
    massainicial := mi;
    massafinal := mf;
    Dispose(pico);
    converteu := TRUE;
End;

Procedure TGrafico.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var

```

```

NewCur,
OldCur : HCursor;
OldFont,
NewFont : HFont;
i,j,xi,increm,posx,posy,oy : Integer;
yi,inten : Word;
s : string[95];
sc : Array[0..94] of Char;

Function Y(i : LongInt) : Integer;
Begin
    Y := Round(1000.0*i/maximo);
End;

Begin { TGráfico.Paint }
{ If not termdados then EXIT;}
If not converteu then Converte;
NewCur := LoadCursor(0, IDC_WAIT); { Sinaliza com a ampulheta. }
OldCur := SetCursor(NewCur);
SetTextAlign(PaintDC, TA_RIGHT);
SetTextColor(PaintDC, RGB(0,0,255));
SetBkMode(PaintDC, transparent);
SetMapMode(PaintDC, MM_ANISOTROPIC);
SetViewportOrg(PaintDC, xLo, yHi-30);
SetViewportExt(PaintDC, xHi-xLo, -(yHi-30));
SetWindowExt(PaintDC, mf-mi, maxy);
SetWindowOrg(PaintDC, mi, 0);
oy := -maxy div 50; { Deslocamento da linha base em 2%. }
NewFont :=
CreateFont(Round(0.06*maxy), 0, 0, 0, FW_BOLD, 0, 0, 0, ANSI_CHARSET,

OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN, 'Tms Rmn');
OldFont := SelectObject(PaintDC, NewFont);
increm := 100*(1 + Trunc((mf - mi)/1000.0 - 0.49)); { Incremento da escala
de massa. }
If (mf - mi) div 10 > 80 then posx := mi - increm div 10
else If (mf - mi) div 10 > 60 then posx := mi - increm div 15
    else If (mf - mi) div 10 > 40 then posx := mi - increm div 20
        else If (mf - mi) div 10 > 20 then posx := mi - increm div 40
            else posx := mi - increm div 60; { Posição x p/ intensidade. }

Str(maxy div 10, s);
StrPCopy(sc, s); { Numera as intensidades. }
TextOut(PaintDC, posx, maxy, sc, StrLen(sc));
Str(maxy div 20, s);
StrPCopy(sc, s);
TextOut(PaintDC, posx, maxy div 2, sc, StrLen(sc));
TextOut(PaintDC, posx, 0, '0', 1);
NewPen := CreatePen(ps_Solid, 0, RGB(0,0,0));
OldPen := SelectObject(PaintDC, NewPen);
MoveTo(PaintDC, mi, maxy);
LineTo(PaintDC, mi, oy); { Desenha os eixos. }
LineTo(PaintDC, mf, oy);
Str(mi div 10, s);
StrPCopy(sc, s);
posy := -maxy div 15;
TextOut(PaintDC, mi, posy, sc, StrLen(sc)); { Numera a 1a massa. }
MoveTo(PaintDC, mi, posy);
LineTo(PaintDC, mi, oy);
i := 0;
xi := mi;

```

```

Repeat
    posy := -maxy div 25;
    inc(i);
    While (xi < mi + (i + 1) * increm) and (xi < mf) do
        begin
            xi := xi + increm div 10;
            MoveTo(PaintDC, xi, posy); { Subdivisões da escala de massa. }
            LineTo(PaintDC, xi, oy);
        end;
    xi := mi + i * increm; { Evita acúmulo de sobras de arredondamento. }
    Str(xi div 10, s);
    StrPCopy(sc, s);
    posy := -maxy div 15;
    TextOut(PaintDC, xi, posy, sc, StrLen(sc)); { Numera as massas. }
    MoveTo(PaintDC, xi, posy);
    LineTo(PaintDC, xi, oy);
Until xi > mf - increm;
For i := 0 to 10 do
    begin
        yi := Round(i * maxy / 10.0);
        MoveTo(PaintDC, posx, yi); { Escala de intensidade. }
        LineTo(PaintDC, mi, yi)
    end;
SelectObject(PaintDC, OldFont);
DeleteObject(NewFont);
SelectObject(PaintDC, OldPen);
DeleteObject(NewPen);
NewPen := CreatePen(ps_Solid, 0, RGB(255,0,0));
OldPen := SelectObject(PaintDC, NewPen);
If maximo > 0 then
    For i := 1 to ndados do { Traça o espectro normalizando as
intensidades. }
        begin
            MoveTo(PaintDC, massa[i], 0);
            If (massa[i] >= mi) and (massa[i] <= mf) then
                LineTo(PaintDC, massa[i], Y(intens[i]));
            end;
        end;
SelectObject(PaintDC, OldPen);
DeleteObject(NewPen);
SetCursor(OldCur);
End;

Procedure TGráfico.WMLButtonDown(var Msg : TMessage);
Begin
    If not ButtonDown then
        begin
            mousexi := xLo; { Valores iniciais para a posição do mouse. }
            mousexf := xHi;
            mouseyf := 0;
            ButtonDown := TRUE;
            SetCapture(HWindow);
            Dc := GetDC(HWindow);
            mousexi := Msg.LParamLo; { Obtém a nova posição do mouse. }
            mouseyi := yHi - 30;
            MoveTo(Dc, mousexi, mouseyi);
            NewPen := CreatePen(ps_Dot, 0, RGB(0,128,0)); { Linha pontilhada
verde. }
            OldPen := SelectObject(Dc, NewPen);
        end;
    End;

```

```
Procedure TGráfico.WMLButtonUp(var Msg : TMessage);
var troca : Word;
Begin
  If ButtonDown then
    begin
      LineTo(Dc,mousexf,mouseyf); { Desenha o retângulo que define a
região que }
      LineTo(Dc,mousexf,mouseyf); { poderá ser ampliada. }
      LineTo(Dc,mousexi,mouseyf);
      LineTo(Dc,mousexi,mouseyi); { Define os valores para a expansão. }
      expansao := Word(Round(maxy * (mouseyf - yHi + 30) / (30 - yHi)));
      If expansao > 1000 then expansao := 1000;
      If expansao < 50 then expansao := 50; { intensidade mínima = 5% }
      expmi := Word(10*Trunc((mi+(mousexi - xLo) * (mf - mi)/(xHi -
xLo))/10.0));
      If expmi < massainicial then expmi := massainicial;
      expmf := Word(10*Trunc((mi+(mousexf - xLo) * (mf - mi)/(xHi -
xLo))/10.0));
      If expmf > massafinal then expmf := massafinal;
      If expmf <= expmi then
        begin
          troca := expmf;
          expmf := expmi;
          expmi := troca;
        end;
      If (expmf - expmi) < 100 then expmf := expmi + 100;
      SelectObject(Dc,OldPen);
      DeleteObject(NewPen);
      ReleaseCapture;
      ReleaseDC(HWindow,Dc);
      ButtonDown := FALSE;
    end;
End;
```

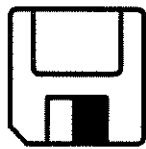
```
Procedure TGráfico.WMRButtonDown(var Msg : TMessage);
Begin
  mi := expmi;
  mf := expmf;
  maxy := expansao;
  InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro expandido.
}
End;
```

```
Procedure TGráfico.WMMouseMove(var Msg : TMessage);
Begin
  If ButtonDown then With Msg do
    begin
      mousexf := integer(LParamLo); { Captura as posições do mouse em
movimento.}
      mouseyf := Integer(LParamHi);
    end;
End;
```

```
Procedure TGráfico.WMLButtonDownClick(var Msg : TMessage);
Begin
  maxy := 1000;
  mi := massainicial;
  mf := massafinal;
  InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro após o
duplo click. }
End;
```

```
BEGIN
  Penta.Init('PENTA4D');
  Penta.Run;
  Penta.Done;
END.
```


SIMTIM



```
Program SIMTIM;
{ ----- }

    Programa para adquirir dados através do monitoramento de
    ions selecionados (SIM) e monitoramento de ions totais (TIM) em função
    do tempo, monitoramento de ions totais em função da energia de
    ionização (IEM) e para gerar curvas analíticas a partir da introdução de
    padrões.

    Utiliza os métodos de programação orientada a objetos,
    programação motivada por eventos e programação por temporização.

    Desenvolvido por Valmir Fascio Juliano como parte do
    trabalho de doutoramento no Instituto de Química-UNICAMP sob
    orientação da Profa. Dra. Concetta Kascheres e do Prof. Dr. Claudimir
    Lúcio do Lago (IQ - USP).
{ ----- }

{ Compilar com Stack Size = 30000 e Heap Size = 8192 }

Uses UPenta4D, WinProcs, WinTypes, WObjects, Strings, WinDos,
CommDlg, Access;

{$R SIMTIM.RES}
var
    numion, controle : Byte;
    numgraficosim : ShortInt;
    temcalibracao, temTIM, temIEM, esta_adquirindo : Boolean;
    WinDir : Array[0..145] of Char;

Const
    EditBoxID = -1;
    pbokID = 1;
    GrupoSensID = 2;
    SensID = 3;
    MenuID = 100;
    simID = 101;
    SalvarID = 102;
    SairID = 103;
    PrintID = 104;
    CalibraID = 105;
    timID = 106;
    iemID = 107;
    InicialID = 200;
    SobreID = 301;
    cm_CopiaUni = 901;
    cm_PrintUni = 902;
    cm_SalvaUni = 903;
    MaxPontos = 2000;
    path = 'C:\PENTA\';
    pathdados = path + 'Q5ESP';

Type
    TTitulo = Array[0..80] of Char;

{ -----TSIMTIMBasico----- }

Type
    PSIMTIMBasico = ^TSIMTIMBasico;
    TSIMTIMBasico = Object(TWindow)
        Procedure Temporizador; virtual;
    End;

Procedure TSIMTIMBasico.Temporizador;
Begin
End;
```

```
{ -----TJanelaSemicone----- }

Type
    PJanelaSemicone = ^TJanelaSemicone;
    TJanelaSemicone = Object(TSIMTIMBasico)
        Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
        Function GetClassName: PChar; virtual;
    End;

Procedure TJanelaSemicone.GetWindowClass(var AWndClass:
TWndClass);
Begin
    TSIMTIMBasico.GetWindowClass(AWndClass);
    AWndClass.hbrBackground := GetStockObject(White_Brush);
    AWndClass.hIcon := 0;
    AWndClass.Style := AWndClass.Style or cs_DblClks;
End;

Function TJanelaSemicone.GetClassName: PChar;
Begin
    GetClassName := 'JanelaSemicone';
End;

{ -----TSIM----- }

Type
    TDadoRec = Record
        tempo,abundancia,varianca : LongInt;
    end;
    PESpectro = ^TESpectro;
    TESpectro = array[1..MaxPontos] of TDadoRec;
    PListaEspectro = ^TListaEspectro;
    TListaEspectro = Object(TObject)
        Espectro : PESpectro;
        PentaQ : PPentaQuad;
        vf : Array[1..MaxPontos] of LongInt;
        quad : Array[um..cinco] of Boolean;
        Xmax, Ymax, cor, maxy : LongInt;
        pontoatual, fim, maxdados, bx, ty, ndados : Integer;
        acabou : Boolean;
        t0 : Double;
        tempointeg, enion : Double;
        mz : Word;
        intaquis : LongInt;
        sensibilidade : Byte;
        comentario : String[80];
        Constructor Init(Max : Integer; q,sens : Byte; mi,mf : Word;
            ti : Double; ia : Word; ei : Double; coment : String);
        Destructor Done; virtual;
        Procedure MedianaMovel(var vo : PESpectro; n : Word);
        Procedure MediaMovel(var vo : PESpectro; n : Word; janela : Integer);
        Procedure LeAbundancia(var abunda, vari : LongInt);
        Procedure Linha(DC: HDC; a1, b1, a2, b2, corlinha: LongInt);
        Procedure Desenha(DC: HDC; Indice: Word); virtual;
        Procedure Redesenha(DC: HDC);
        Procedure Adquire(DC: HDC);
    End;

PSIM = ^TSIM;
TSIM = Object(TJanelaSemicone)
    GraficoEspectro : PListaEspectro;
```

```

    indice : Byte;
    titulo : TTitulo;
    Constructor Init(AParent: PWindowsObject; ATitle: PChar;
        q,s : Byte; mi,mf : Word; ti : Double; ia : Word;
        ei : Double; coment : String);
    Destructor Done; virtual;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
    Procedure WMLButtonDown(var Msg : TMessage); virtual wm_First +
wm_LButtonDown;
    Procedure WMLButtonDbIClk(var Msg : TMessage); virtual wm_First +
wm_LButtonDownDbIClk;
    Procedure WMSize(var Msg: TMessage); virtual wm_First + wm_Size;
    Procedure Temporizador; virtual;
End;

Constructor TListaEspectro.Init(Max : Integer; q,sens : Byte; mi,mf : Word;
    ti : Double; ia : Word; ei : Double;
    coment : String);

var
    h,m,s,cs : Word;
Begin
    TObject.Init;
    If Max > MaxPontos then Max := MaxPontos;
    maxdados := Max;
    cor := RGB(0,0,255);
    GetMem(Espectro, SizeOf(TDadoRec) * Maxdados);
    quad[um] := q = 1;
    quad[tres] := q = 3;
    quad[cinco] := q = 5;
    sensibilidade := sens;
    mz := mi;
    tempointeg := ti;
    intaquis := ia;
    enion := ei;
    comentario := coment;
    New{PentaQ,init(path,'Q1.CAL','Q3.CAL','Q5.CAL')};
    If not PentaQ^.LinhaBase then
        MessageBox(GetFocus,'Problemas com a linha-base, interrompa a
aquisição.',
        'SIM',mb_OK);
    pontoatual := 2;
    fim := 0;
    ndados := 0;
    Xmax := 60000;
    Ymax := 20000;
    maxy := 0;
    acabou := FALSE;
    With Espectro^[1] do
        begin
            tempo := 0;
            LeAbundancia(abundancia,varianca);
        end;
    GetTime(h,m,s,cs);
    t0 := 3600000.0 * h + 60000.0 * m + 1000.0 * s + 10.0 * cs;
End;

Destructor TListaEspectro.Done;
Begin
    FreeMem(Espectro, SizeOf(TDadoRec) * Maxdados);
    TObject.Done;
End;
```

```

Procedure TListaEspectro.MedianaMovel(var vo : PEspectro; n : Word);
var
    i : Integer;
    sai, entra, pilha : LongInt;
    d : Array[-1..1] of LongInt; { A posicao 0 e' a mediana. }
    pronto : Boolean;
Begin
    If n < 3 then Exit; { Devem existir pelo menos 3 dados. }
    For i := -1 to 1 do d[i] := vo^[i+2].abundancia;
    pronto := TRUE;
    Repeat
        pronto := TRUE;
        For i := -1 to 0 do
            If d[i] > d[i+1] then
                begin
                    pronto := False;
                    pilha := d[i];
                    d[i] := d[i+1];
                    d[i+1] := pilha;
                end;
        Until pronto;
        vf[1] := d[0];
        vf[2] := d[0];
        For i := 3 to n - 1 do
            begin
                sai := vo^[i - 2].abundancia;
                entra := vo^[i + 1].abundancia;
                If sai = d[-1] then
                    If entra > d[0] then
                        If entra > d[1] then
                            begin
                                d[-1] := d[0];
                                d[0] := d[1];
                                d[1] := entra;
                            end
                        else
                            begin
                                d[-1] := d[0];
                                d[0] := entra;
                            end
                        else d[-1] := entra
                    else
                        If sai = d[0] then
                            If entra > d[-1] then
                                If entra > d[1] then
                                    begin
                                        d[0] := d[1];
                                        d[1] := entra;
                                    end
                                else d[0] := entra
                            else
                                begin
                                    d[0] := d[-1];
                                    d[-1] := entra;
                                end
                            else
                                If entra >= d[-1] then
                                    If entra >= d[0] then d[1] := entra
                                else
                                    begin
```

```

        d[1] := d[0];
        d[0] := entra;
    end
else
begin
    d[1] := d[0];
    d[0] := d[-1];
    d[-1] := entra;
end;
vf[i] := d[0];
end;
vf[n] := vf[n-1];
For i := 1 to n do
    vo^[i].abundancia := vf[i];
End;

```

```

Procedure TListaEspectro.MediaMovei(var vo : PESpectro; n : Word; janela
: Integer);
var i : Word;
    j : Integer;
    soma : Double;
Begin
    If n < (2 * janela + 1) then EXIT;
    soma := 0.0;
    i := janela + 1;
    For j := -janela to janela do
        soma := soma + 1.0 * vo^[i+j].abundancia;
    vf[j] := Round(soma / (2.0 * janela + 1));
    Repeat
        Inc(i);
        soma := soma - 1.0 * vo^[i - janela - 1].abundancia
            + 1.0 * vo^[i + janela].abundancia;
        vf[j] := Round(soma / (2.0 * janela + 1));
    Until i = n - janela;
    For i := janela + 1 to n - janela do
        vo^[i].abundancia := vf[i];
    End;

```

```

Procedure TListaEspectro.LeAbundancia(var abunda,vari : LongInt);
var q,quadrup : Quadrupolos;
    i : Integer;
    fator,ab,va : Double;
Begin
    PentaQ^.det.SetTempoInteg(tempointeg); { Ajusta sempre todos os
parâmetros, }
    PentaQ^.Ionizador.El(enion);           { pois algum outro objeto pode estar }
    Port[$318] := sensibilidade;           { manipulando os elementos do espec-
}
    For q := um to cinco do                { trômetro.                }
        If quad[q] then                    {                          }
            begin                          {                          }
                PentaQ^.SetPoleDC(q,TRUE); { Coloca os quadrupolos que não
es- }
                quadrup := q;              { tão varrendo com o poleDC em off. }
            end                            {                          }
        else PentaQ^.SetPoleDC(q,FALSE);   {                          }
    PentaQ^.SetMassa(quadrup,mz);          { Ajusta a massa do quadrupolo.
}
    i := 8;
    Repeat
        Dec(i);

```

```

        PentaQ^.det.ganho := ganhos[i];
        PentaQ^.LeMedia(3,ab,va);
    Until (not PentaQ^.det.saturado) or (i = 1);
    If ab < 0 then ab := 0;
    abunda := Round(ab);
    vari := Round(va);
    If maxy < abunda then maxy := abunda;
End;

Procedure TListaEspectro.Linha(DC: HDC; a1,b1,a2,b2,corlinha: Longint);
Function XP(v : Double) : Integer;
Begin
    XP := Integer(Round(v * bx / Xmax));
End;

```

```

Function YP(v : Double) : Integer;
Begin
    YP := Integer(Round(-ty*(v - Ymax) / Ymax));
End;

var
    OldPen: HPen;
Begin
    OldPen := SelectObject(DC, CreatePen(PS_SOLID, 1, corlinha));
    MoveTo(DC, XP(a1), YP(b1));
    LineTo(DC, XP(a2), YP(b2));
    DeleteObject(SelectObject(DC, OldPen));
End;

```

```

Procedure TListaEspectro.Desenha(DC: HDC; Indice: Word);
Begin
    Linha(DC,Espectro^[Indice-1].tempo,Espectro^[Indice-1].abundancia,
        Espectro^[Indice].tempo,Espectro^[Indice].abundancia,cor);
End;

```

```

Procedure TListaEspectro.Redesenha(DC: HDC);
var i : Word;
Begin
    For i := 2 to fim do Desenha(DC,i);
End;

```

```

Procedure TListaEspectro.Acquire(DC: HDC);
var
    h,m,s,cs : Word;
    dt : Double;
    st : String[40];
    c : Array[0..40] of Char;
Begin
    GetTime(h,m,s,cs);
    dt := 3600000.0 * h + 60000.0 * m + 1000.0 * s + 10.0 * cs; { dt em ms. }
    If dt < t0 then dt := dt + 24 * 3600000.0 - t0
    else dt := dt - t0;
    If dt < (Espectro^[pontoatual-1].tempo + intaquis) then EXIT;
    With Espectro^[pontoatual] do
        begin
            tempo := Round(dt);           { tempo em ms. }
            LeAbundancia(abundancia,varianca);
        end;
    fim := pontoatual - 1;
    Desenha(DC, pontoatual);
    Inc(pontoatual);

```

```
If pontoatual > Maxdados then
begin
  acabou := TRUE;
  Dec(pontoatual);
  ndados := pontoatual;
  fim := ndados;
  Xmax := Espectro^[ndados].tempo;
  Ymax := maxy;
  MedianaMovel(Espectro,ndados);
  MediaMovel(Espectro,ndados,3);
  Str(mz,st);
  st := 'Aquisição de m/z ' + st;
  StrPCopy(c,st);
  MessageBeep(mb_ok);
  MessageBox(GetFocus,'Aquisição encerrada.',c,mb_OK);
end;
End;

Constructor TSIM.Init(AParent: PWindowsObject; ATitle: PChar;
  q,s: Byte; mi,mt: Word; ti: Double; ia: Word;
  ei: Double; coment: String);

Begin
  TJanelaSemlcone.Init(AParent, ATitle);
  GraficoEspectro := New(PListaEspectro,
    Init(MaxPontos,q,s,mi,mt,ti,ia,ei,coment));
  GraficoEspectro^.bx := Attr.W;
  GraficoEspectro^.ty := Attr.H;
End;

Destructor TSIM.Done;

Begin
  indice := 0;
  Dec(numgraficosim);
  If numgraficosim <= 0 then esta_adquirindo := FALSE;
  Dispose(GraficoEspectro^.PentaQ,Done);
  Dispose(GraficoEspectro,Done);
  TJanelaSemlcone.Done;
End;

Procedure TSIM.WMSize(var Msg: TMessage);
var
  i: Integer;
Begin
  TJanelaSemlcone.WMSize(Msg);
  With GraficoEspectro^ do
    begin
      i := pontoatual;
      If (i > 2) and not acabou then
        begin
          If Xmax < Espectro^[i-1].tempo then Xmax := 2 * Espectro^[i-1].tempo;
          If Ymax < Espectro^[i-1].abundancia then Ymax := 2 * Espectro^[i-1].abundancia;
          end;
          tx := Msg.LParamLo; {Attr.W;}
          ty := Msg.LParamHi; {Attr.H;}
          end;
          InvalidateRect(HWindow, nil, TRUE);
        End;

Procedure TSIM.WMLButtonDown(var Msg: TMessage);
```

```
var
  i: Integer;
Begin
  With GraficoEspectro^ do
    begin
      i := pontoatual;
      If Xmax < Espectro^[i-1].tempo then Xmax := 2 * Espectro^[i-1].tempo;
      If Ymax < Espectro^[i-1].abundancia then Ymax := 2 *
        Espectro^[i-1].abundancia;
      end;
      InvalidateRect(HWindow, nil, TRUE);
    End;

Procedure TSIM.WMLButtonDbIClk(var Msg: TMessage);
var
  s: String[40];
  c: Array[0..40] of Char;
Begin { WMLButtonDbIClk }
  With GraficoEspectro^ do
    begin
      If acabou then EXIT;
      acabou := TRUE;
      Dec(pontoatual);
      Dec(numgraficosim);
      If numgraficosim <= 0 then esta_adquirindo := FALSE;
      ndados := pontoatual;
      fim := ndados;
      Xmax := Espectro^[ndados].tempo;
      Ymax := maxy;
      MedianaMovel(Espectro,ndados);
      MediaMovel(Espectro,ndados,3);
      end;
      InvalidateRect(HWindow,nil,TRUE);
      Str(indice,s);
      s := 'Aquisição ' + s;
      StrPCopy(c,s);
      MessageBeep(mb_ok);
      MessageBox(HWindow,'Aquisição encerrada.',c,mb_OK);
    End;

Procedure TSIM.Paint(PaintDC: HDC; var PaintInfo: TPaintStruct);
Begin
  TJanelaSemlcone.Paint(PaintDC, PaintInfo);
  GraficoEspectro^.Redesenha(PaintDC);
End;

Procedure TSIM.Temporizador; { Executa o conteúdo do procedure
periodicamente. }
var
  DC: HDC;
Begin
  If not GraficoEspectro^.acabou then
    begin
      DC := GetDC(HWindow);
      GraficoEspectro^.Adquire(DC);
      ReleaseDC(HWindow, DC);
    end;
  End;

{ -----TEntraSIM----- }
Type
```

```
PEntraSIM = ^TEntaSIM;
TEntaSIM = Object(TJanelaSemicone)
  mz,tempoint,coment, intaquis, enion : PEdit;
  senslo, sensmed, senshi : PRadioButton;
  GrupoSens : PGroupBox;
  quad : PComboBox;
  indice : Byte;
  Constructor Init(AParent : PWindowsObject; ATitle : PChar);
  Procedure SetupWindow; virtual;
  Destructor Done; virtual;
  Procedure Paint(PaintDC : HDC; var Paintinfo : TPaintStruct); virtual;
End;

Constructor TEntaSIM.Init(AParent : PWindowsObject; ATitle : PChar);
Begin
  TJanelaSemicone.Init(AParent, ATitle);
  With Attr do
    begin
      X := CW_USEDEFAULT; Y := CW_USEDEFAULT; W := 420; H := 290;
    end;

    quad :=
New(PComboBox,Init(@Self,EditBoxID,115,0,50,100,cbs_DropDownList,0))
;

    quad^.Attr.Style := quad^.Attr.Style and (not cbs_Sort);
    GrupoSens := New(PGroupBox, Init(@Self, GrupoSensID,"",0,0,0,0));
    senslo := New(PRadioButton, Init(@Self, sensID,
'B',125,35,40,20,GrupoSens));
    sensmed := New(PRadioButton, Init(@Self, sensID,
'M',165,35,40,20,GrupoSens));
    senshi := New(PRadioButton, Init(@Self,
sensID,'A',205,35,40,20,GrupoSens));
    mz := New(PEdit,Init(@Self,EditBoxID,'28',60,70,50,30,5,FALSE));
    tempoint :=
New(PEdit,Init(@Self,EditBoxID,'33.4',205,105,50,30,5,FALSE));
    intaquis := New(PEdit,Init(@Self,EditBoxID,'50',215,140,60,30,6,FALSE));
    enion := New(PEdit,Init(@Self,EditBoxID,'70.0',210,175,50,30,5,FALSE));
    coment := New(PEdit,Init(@Self,EditBoxID,"",185,210,220,30,81,FALSE));
  End;

Procedure TEntaSIM.SetupWindow;
Begin
  TJanelaSemicone.SetupWindow;
  quad^.AddString('1');
  quad^.AddString('3');
  quad^.AddString('5');
  quad^.SetSelIndex(2); { 0,1,2 } { Default quad = 5. }
  sensmed^.SetCheck(1); { 0,1,2 } { Default sensibilidade = media. }
End;

Destructor TEntaSIM.Done;
Begin
  indice := 0;
  TJanelaSemicone.Done;
End;

Procedure TEntaSIM.Paint(PaintDC : HDC; var Paintinfo : TPaintStruct);
var
  OldBrush,
  NewBrush : HBrush;
  NewFont,
  OldFont : HFont;
```

```
OldPen: HPen;
i : Byte;
Begin
  If IsIconic(HWindow) then
    begin
      OldPen := SelectObject(PaintDC, CreatePen(PS_SOLID, 1,
RGB(255,0,0)));
      i := 0;
      Repeat
        inc(i,2);
        Rectangle(PaintDC,i,i,80-i,80-i);
      Until i = 50;
      DeleteObject(SelectObject(PaintDC, OldPen));
    end
  else
    begin
      NewFont := CreateFont(20,0,0,0,FW_NORMAL,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
      OldFont := SelectObject(PaintDC,NewFont);
      SetTextColor(PaintDC,RGB(255,0,0));
      SetBkMode(PaintDC,transparent);
      SetTextAlign(PaintDC,TA_LEFT);
      TextOut(PaintDC,25,5,'Quadrupolo:',11);
      TextOut(PaintDC,25,35,'Sensibilidade:',14);
      TextOut(PaintDC,25,75,'m/z:',4);
      TextOut(PaintDC,25,110,'Tempo de integração (ms):',25);
      TextOut(PaintDC,25,145,'Intervalo de aquisição (ms):',28);
      TextOut(PaintDC,25,180,'Energia de ionização (eV):',26);
      TextOut(PaintDC,25,215,'Comentário (80 letras):',23);
      SelectObject(PaintDC,OldFont);
      DeleteObject(NewFont);
    end;
  End;

{ -----TCalibra----- }
Type
  Dado = Record
    abundancia, varianca : LongInt;
  end;
  PCalibra = ^TCalibra;
  TCalibra = Object(TJanelaSemicone)
    PentaQ : PPentaQuad;
    mz, coment, concentracoes, enion, leituras : PEdit;
    senslo, sensmed, senshi : PRadioButton;
    GrupoSens : PGroupBox;
    quad : PComboBox;
    pbOK : PButton;
    firm, firmnz, nleituras, ndesconhecido : Integer;
    massa : Array[1..10] of Integer;
    conc : Array[1..50] of Double;
    abunda, desconhecido : Array[1..50,1..10] of Dado;
    enion : Double;
    comentario : String;
  Constructor Init(AParent : PWindowsObject; ATitle : PChar);
  Procedure SetupWindow; virtual;
  Destructor Done; virtual;
  Procedure Paint(PaintDC : HDC; var Paintinfo : TPaintStruct); virtual;
  Procedure Regressao(var incl, inter : Double; n,j : Integer);
  Procedure Executa(var Msg : TMessage); virtual id_First + pbokID;
```

```

End;

Constructor TCalibra.Init(AParent : PWindowsObject; ATitle : PChar);
Begin
  TJanelaSemicone.Init(AParent, ATitle);
  With Attr do
    begin
      X := CW_USEDEFAULT; Y := CW_USEDEFAULT; W := 420; H := 320;
    end;

    quad :=
New(PComboBox,Init(@Self,EditBoxID,115,0,50,100,cbs_DropDownList,0))
;
quad^.Attr.Style := quad^.Attr.Style and (not cbs_Sort);
GrupoSens := New(PGroupBox, Init(@Self, GrupoSensID,"",0,0,0,0));
sensio := New(PRadioButton, Init(@Self, sensID,
'B',125,35,40,20,GrupoSens));
sensmed := New(PRadioButton, Init(@Self, sensID,
'M',165,35,40,20,GrupoSens));
senshi := New(PRadioButton, Init(@Self,
sensID,'A',205,35,40,20,GrupoSens));
mz := New(PEdit,Init(@Self,EditBoxID,"",70,70,200,30,81,FALSE));
enion := New(PEdit,Init(@Self,EditBoxID,'70.0',210,105,50,30,5,FALSE));
concentracoes :=
New(PEdit,Init(@Self,EditBoxID,"",130,140,260,30,81,FALSE));
leituras := New(PEdit,Init(@Self,EditBoxID,'10',235,175,40,30,4,FALSE));
coment := New(PEdit,Init(@Self,EditBoxID,"",185,210,220,30,81,FALSE));
pbOK := New(PButton,Init(@Self,pbokID,'Iniciar',180,250,80,30,FALSE));
End;

Procedure TCalibra.SetupWindow;
Begin
  TJanelaSemicone.SetupWindow;
  quad^.AddString('1');
  quad^.AddString('3');
  quad^.AddString('5');
  quad^.SetSelIndex(2); { 0,1,2 } { Default quad = 5. }
  sensmed^.SetCheck(1); { 0,1,2 } { Default sensibilidade = media. }
End;

Destructor TCalibra.Done;
Begin
  TJanelaSemicone.Done;
  temcalibracao := FALSE;
End;

Procedure TCalibra.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
  OldBrush,
  NewBrush : HBrush;
  NewFont,
  OldFont : HFont;
  OldPen : HPen;
  i : Byte;
Begin
  If IsIconic(HWindow) then
    begin
      OldPen := SelectObject(PaintDC, CreatePen(PS_SOLID,
1,RGB(0,0,255)));
      i := 0;
      Repeat
        inc(i,2);

```

```

      Rectangle(PaintDC,i,i,80-i,80-i);
      Until i = 50;
      DeleteObject(SelectObject(PaintDC, OldPen));
    end
  else
    begin
      NewFont := CreateFont(20,0,0,0,FW_NORMAL,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
      OldFont := SelectObject(PaintDC,NewFont);
      SetTextColor(PaintDC,RGB(0,0,255));
      SetBkMode(PaintDC,transparent);
      SetTextAlign(PaintDC,TA_LEFT);
      TextOut(PaintDC,25,5,'Quadrupolo:',11);
      TextOut(PaintDC,25,40,'Sensibilidade:',14);
      TextOut(PaintDC,25,75,'m/z"s:',6);
      TextOut(PaintDC,25,110,'Energia de Ionização (eV):',26);
      TextOut(PaintDC,25,145,'Concentrações:',15);
      TextOut(PaintDC,25,180,'Número de leituras por ponto:',30);
      TextOut(PaintDC,25,215,'Comentário (80 letras):',23);
      SelectObject(PaintDC,OldFont);
      DeleteObject(NewFont);
    end;
  End;

Procedure TCalibra.Regressao(var incl,inter : Double; n,j : Integer);
var
  sx,sx2,sy2,sxy,ym,xm,correl : Double;
  i : Word;
  a : Longint;
  c : Array[0..255] of Char;
  s : String;
  s1,s2 : String[30];
Begin
  sx := 0.0; sy := 0.0;
  sx2 := 0.0; sy2 := 0.0;
  sxy := 0.0;
  For i := 1 to n do
    begin
      a := abunda[i,j],abundancia;
      sx := sx + conc[i];
      sy := sy + a;
      sx2 := sx2 + conc[i] * conc[i];
      sy2 := sy2 + 1.0 * a * a;
      sxy := sxy + conc[i] * a;
    end;
  xm := sx / n; ym := sy / n;
  incl := (sxy - sx * sy / n) / (sx2 - sx * sx / n);
  inter := ym - incl * xm;
  correl := incl*sqrt(sx2/n-xm*xm)/sqrt(sy2/n-ym*ym);
  Str(incl:13,s1);
  s := 'Inclinação: ' + s1 + #10;
  Str(inter:13,s1);
  s := s + 'Intercepto: ' + s1 + #10;
  Str(correl:0.5,s1);
  s := s + 'Correlação: ' + s1;
  StrPCopy(c,s);
  MessageBox(HWindow,c,'Ajuste Linear',mb_Ok or mb_IconInformation);
End;

```

```
Procedure TCalibra.Executa(var Msg : TMessage);
var
  s : String;
  s1 : String[30];
  c : Array[0..255] of Char;
  vi, err, i, j : Integer;
  a, v : Double;
  quadrup : Array[um..cinco] of Boolean;
  sensibilidade : Byte;
  q,quadrupol : quadrupolos;
  inclinacao, intercepto : Array[1..10] of Double;
Begin
  If esta_adquirindo then EXIT;
  If MessageBox(HWindow,'Adquire dados?','Calibração',
    mb_YesNo or mb_IconQuestion) = idNo then EXIT;
  New(PentaQ,init(path,'Q1.CAL','Q3.CAL','Q5.CAL'));
  quad^a.GetSelString(c,1);
  s := StrPas(c);
  Val(s,vi,err);
  quadrup[um] := vi = 1;
  quadrup[tres] := vi = 3;
  quadrup[cinco] := vi = 5;
  For q := um to cinco do
    If quadrup[q] then
      begin
        PentaQ^.SetPoleDC(q,TRUE);
        quadrupol := q;
      end
    else PentaQ^.SetPoleDC(q,FALSE);
  If senslo^.GetCheck = 1 then sensibilidade := 1;
  If sensmed^.GetCheck = 1 then sensibilidade := 3;
  If senshi^.GetCheck = 1 then sensibilidade := 2;
  If not PentaQ^.LinhaBase then
    MessageBox(GetFocus,'Problemas com a linha-base, interrompa a
aquisição.',
      'Calibração',mb_OK);
  mz^.GetText(c,255);
  s := StrPas(c);
  fimmz := 0;
  While pos(', ',s) > 0 do
    begin
      Inc(fimmz);
      s1 := Copy(s,1,pos(', ',s)-1);
      Val(s1,masa[fimmz],err);
      Delete(s,1,pos(', ',s));
    end;
  Inc(fimmz);
  Val(s,masa[fimmz],err);
  If err <> 0 then
    begin
      MessageBox(HWindow,'Erro nos valores de m/z','Leitura de Dados',
        mb_ok or mb_IconHand);
      Dispose(PentaQ,Done);
      EXIT;
    end;
  enion^.GetText(c,5);
  s := StrPas(c);
  Val(s,enerion,err);
  PentaQ^.Ionizador.El(enerion);
  PentaQ^.det.SetTempoInteg(33.4);
  leituras^.GetText(c,4);

  s := StrPas(c);
  Val(s,nleituras,err);
  If err <> 0 then nleituras := 1;
  If nleituras < 2 then nleituras := 2;
  coment^a.GetText(c,80);
  comentario := StrPas(c);
  concentracoes^.GetText(c,255);
  s := StrPas(c);
  fim := 0;
  While pos(', ',s) > 0 do
    begin
      Inc(fim);
      s1 := Copy(s,1,pos(', ',s)-1);
      Val(s1,conc[fim],err);
      Delete(s,1,pos(', ',s));
    end;
  Inc(fim);
  Val(s,conc[fim],err);
  If (fim < 2) or (err <> 0) then
    begin
      MessageBox(HWindow,'Erro nas concentrações','Leitura de Dados',
        mb_ok or mb_IconHand);
      Dispose(PentaQ,Done);
      EXIT;
    end;
  Port[$318] := sensibilidade;
  For i := 1 to fim do
    begin
      Str(conc[i]:13,s);
      s := 'Introduza o padrão de concentração ' + s + #10;
      s := s + 'na fonte do aparelho e "clic"';
      StrPCopy(c,s);
      MessageBox(HWindow,c,'Leitura de Dados',mb_ok or mb_IconHand);
      For j := 1 to fimmz do
        begin
          PentaQ^.SetMassa(quadrupol,masa[j]);
          PentaQ^.LeMedia(nleituras,a,v);
          abunda[i,j].abundancia := Round(a);
          abunda[i,j].varianca := Round(v);
        end;
      end;
      For j := 1 to fimmz do Regressao(inclinacao[j],intercepto[j],fim,j);
      If MessageBox(HWindow,'Continua para fazer leituras de
desconhecidos?',
        'Leitura de Dados',mb_YesNo or mb_IconQuestion) = idNo then
        begin
          ndesconhecido := 0;
          Dispose(PentaQ,Done);
          EXIT;
        end;
      i := 0;
      While MessageBox(HWindow,'Introduza o desconhecido e "clic"',
        'Leitura de Dados',mb_OkCancel or mb_IconHand) = idOk do
        begin
          inc(i);
          Str(i,s1);
          For j := 1 to fimmz do
            begin
              s1 := " ";
              PentaQ^.SetMassa(quadrupol,masa[j]);
              s := 'Concentração do desconhecido ' + s1 + ' em m/z ';
```



```

PentaQ^.LeMedia(nleituras,a,v);
desconhecido[i,j].abundancia := Round(a);
desconhecido[i,j].varianca := Round(v);
Str(massa[j],s1);
s := s + s1 + ' ';
Str((desconhecido[i,j].abundancia - intercepto[j]) / inclinacao[j]:13,s1);
s := s + s1;
StrPCopy(c,s);
MessageBox(HWindow,c,'Leitura de Dados',mb_Ok or
mb_IconInformation);
end;
end;
ndesconhecido := i;
Dispose(PentaQ,Done);
End;

```

```

{ -----TTIM----- }

```

Type

```

TDadoTIM = Record
    tim,tempo : LongInt;
end;
TRegistro = Record
    tempo : LongInt;
    abundancia : Array[1..1000] of LongInt;
    varianca : Array[1..1000] of LongInt;
end;
PEspectroTIM = ^TEspectroTIM;
TEspectroTIM = Array[1..MaxPontos] of TDadoTIM;
PListaTIM = ^TListaTIM;
TListaTIM = Object(TObject)
    Espectro : PEspectroTIM;
    registro : TRegistro;
    arq : File of TRegistro;
    PentaQ : PPentaQuad;
    quad : Array[um..cinco] of Boolean;
    Xmax, Ymax, cor, maxy : LongInt;
    pontoatual, fim, maxdados, tx, ty, ndados : Integer;
    acabou : Boolean;
    tempointeg, enion : Double;
    t0 : Double;
    mzi, mzf : Word;
    intaquis : LongInt;
    sensibilidade : Byte;
    quadrup : Quadrupolos;
    comentario : String[80];
    vf : Array[1..MaxPontos] of LongInt;
    Constructor Init(Max : Integer; q,sens : Byte; mi,mf : Word;
        ti : Double; ia : Word; ei : Double; c : String);
    Destructor Done; virtual;
    Procedure MedianaMovet(var vo : PEspectroTIM; n : Word);
    Procedure MediaMovet(var vo : PEspectroTIM; n : Word; janela : Integer);
    Function LeAbundancia : LongInt;
    Procedure Linha(DC: HDC; a1, b1, a2, b2, corlinha: LongInt);
    Procedure Desenha(DC: HDC; Indice: Word); virtual;
    Procedure Redesenha(DC: HDC);
    Procedure Adquire(DC: HDC);
End;

PGrafico = ^TGrafico;
TGrafico = object(TJanelaSemlcone)
    ButtonDown : Boolean;

```

```

mousexi,mousexf,mouseyi,mouseyf,
xHi, xLo, yHi, yLo : Integer;
Dc : HDC;
NewPen, OldPen : HPen;
mi,mf,maxy,numdados,expansao,expmi,expmf,
massainicial, massafinal : Word;
maximo : LongInt;
massa : Array[1..1000] of Word;
intens : Array[1..1000] of LongInt;
    Constructor Init(AParent: PWindowsObject; ATitle: PChar);
    Destructor Done; virtual;
    Function GetClassName: PChar; virtual;
    Procedure GetWindowClass(var AWndClass: TWndClass); virtual;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
    Procedure WMLButtonDown(var Msg : TMessage); virtual wm_First +
wm_LButtonDown;
    Procedure WMRButtonDown(var Msg : TMessage); virtual wm_First +
wm_RButtonDown;
    Procedure WMLButtonUp(var Msg : TMessage); virtual wm_First +
wm_LButtonUp;
    Procedure WMMouseMove(var Msg : TMessage); virtual wm_First +
wm_MouseMove;
    Procedure WMLButtonDbClick(var Msg : TMessage); virtual wm_First +
wm_LButtonDbClick;
    Procedure Copia(var Msg : TMessage); virtual cm_First + cm_CopiaUni;
    Procedure Print(var Msg : TMessage); virtual cm_First + cm_PrintUni;
    Procedure Salva(var Msg : TMessage); virtual cm_First + cm_SalvaUni;
end;

PTIM = ^TTIM;
TTIM = Object(TJanelaSemlcone)
    GraficoEspectro : PListaTIM;
    JF : PGrafico;
    Constructor Init(AParent: PWindowsObject; ATitle: PChar;
        quad, s : Byte; mi,mf : Word; ti : Double; ia : Word;
        ei : Double; coment : String);
    Destructor Done; virtual;
    Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
    Procedure WMLButtonDown(var Msg : TMessage); virtual wm_First +
wm_LButtonDown;
    Procedure WMLButtonDbClick(var Msg : TMessage); virtual wm_First +
wm_LButtonDbClick;
    Procedure WMSize(var Msg: TMessage); virtual wm_First + wm_Size;
    Procedure Temporizador; virtual;
End;

```

```

    Constructor TListaTIM.Init(Max : Integer; q,sens : Byte; mi,mf : Word;
        ti : Double; ia : Word; ei : Double; c : String);
var
    h,m,s,cs : Word;
    quadr : quadrupolos;
Begin
    TObject.Init;
    If Max > MaxPontos then Max := MaxPontos;
    maxdados := Max;
    cor := RGB(0,0,255);
    GetMem(Espectro, SizeOf(TDadoRec) * Maxdados);
    quad[um] := q = 1;
    quad[tres] := q = 3;
    quad[cinco] := q = 5;
    sensibilidade := sens;

```

```
mzi := mi;
mzf := mf;
tempointeg := ti;
intaquis := ia;
enion := ei;
comentario := c;
New(PentaQ,Init(path,'Q1.CAL','Q3.CAL','Q5.CAL'));
PentaQ^.det.SetTempInteg(tempointeg);
PentaQ^.Ionizador.El(enion);
Port{$318} := sensibilidade;
For quadr := um to cinco do
  If quad[quadr] then
    begin
      PentaQ^.SetPoleDC(quadr,TRUE);
      quadrup := quadr;
    end
  else PentaQ^.SetPoleDC(quadr,FALSE);
If not PentaQ^.LinhaBase then
  MessageBox(GetFocus,'Problemas com a linha-base, interrompa a
aquisição.',
  'TIM',mb_OK);
pontoatual := 2;
fim := 0;
ndados := 0;
Xmax := 60000;
Ymax := 20000;
maxy := 0;
acabou := FALSE;
Assign(arq,pathdados + '\dados.tim');
ReWrite(arq);
With Espectro^[1] do
  begin
    tempo := 0;
    tim := LeAbundancia;
  end;
  GetTime(h,m,s,cs);
  t0 := 3600000.0 * h + 60000.0 * m + 1000.0 * s + 10.0 * cs;
End;

Destructor TListaTIM.Done;
Begin
  FreeMem(Espectro, SizeOf(TDadoRec) * Maxdados);
  TObject.Done;
End;

Procedure TListaTIM.MedianaMove(var vo : PEspectroTIM; n : Word);
var
  i : Integer;
  sai, entra, pilha : LongInt;
  d : Array[-1..1] of LongInt; { A posicao 0 e' a mediana. }
  pronto : Boolean;
Begin
  If n < 3 then Exit; { Devem existir pelo menos 3 dados. }
  For i := -1 to 1 do d[i] := vo^[i+2].tim;
  pronto := TRUE;
  Repeat
    pronto := TRUE;
    For i := -1 to 0 do
      If d[i] > d[i+1] then
        begin
          pronto := False;
```

```
        pilha := d[i];
        d[i] := d[i+1];
        d[i+1] := pilha;
        end;
  Until pronto;
  vf[1] := d[0];
  vf[2] := d[0];
  For i := 3 to n - 1 do
    begin
      sai := vo^[i - 2].tim;
      entra := vo^[i + 1].tim;
      If sai = d[-1] then
        If entra > d[0] then
          If entra > d[1] then
            begin
              d[-1] := d[0];
              d[0] := d[1];
              d[1] := entra;
            end
          else
            begin
              d[-1] := d[0];
              d[0] := entra;
            end
          else d[-1] := entra
        else
          If sai = d[0] then
            If entra > d[-1] then
              If entra > d[1] then
                begin
                  d[0] := d[1];
                  d[1] := entra;
                end
              else d[0] := entra
            else
              begin
                d[0] := d[-1];
                d[-1] := entra;
              end
            else
              If entra >= d[-1] then
                If entra >= d[0] then d[1] := entra
              else
                begin
                  d[1] := d[0];
                  d[0] := entra;
                end
              else
                begin
                  d[1] := d[0];
                  d[0] := d[-1];
                  d[-1] := entra;
                end;
              vf[i] := d[0];
            end;
          vf[n] := vf[n-1];
          For i := 1 to n do
            vo^[i].tim := vf[i];
          End;
```

```
Procedure TListaTIM.MediaMoveI(var vo : PEspectroTIM; n : Word; janela : Integer);
var i : Word;
    j : Integer;
    soma : Double;
Begin
    If n < (2 * janela + 1) then EXIT;
    soma := 0.0;
    i := janela + 1;
    For j := -janela to janela do
        soma := soma + 1.0 * vo^[i+j].tim;
    v[i] := Round(soma / (2.0 * janela + 1));
    Repeat
        Inc(i);
        soma := soma - 1.0 * vo^[i - janela - 1].tim
            + 1.0 * vo^[i + janela].tim;
        v[i] := Round(soma / (2.0 * janela + 1));
    Until i = n - janela;
    For i := janela + 1 to n - janela do
        vo^[i].tim := v[i];
    End;

Function TListaTIM.LeAbundancia : LongInt;
var
    i, j : Integer;
    soma : LongInt;
    abunda, varianca : Double;
Begin
    soma := 0;
    For i := mzi to mzf do
        begin
            PentaQ^.SetMassa(quadrup,i);
            j := 8;
            Repeat
                Dec(j);
                PentaQ^.det.ganho := ganhos[j];
                PentaQ^.LeMedia(3,abunda,varianca);
            Until (not PentaQ^.det.saturado) or (j = 1);
            If abunda < 0 then abunda := 0;
            registro.abundancia[i] := Round(abunda);
            registro.varianca[i] := Round(varianca);
            soma := soma + registro.abundancia[i];
        end;
    If maxy < soma then maxy := soma;
    LeAbundancia := soma;
End;

Procedure TListaTIM.Linha(DC: HDC; a1,b1,a2,b2,corinha: Longint);
Function XP(v : Double) : Integer;
Begin
    XP := Integer(Round(v * tx / Xmax));
End;

Function YP(v : Double) : Integer;
Begin
    YP := Integer(Round(-ty*(v - Ymax) / Ymax));
End;

var
    OldPen: HPen;
Begin
```

```
    OldPen := SelectObject(DC, CreatePen(PS_SOLID, 1, corinha));
    MoveTo(DC, XP(a1), YP(b1));
    LineTo(DC, XP(a2), YP(b2));
    DeleteObject(SelectObject(DC, OldPen));
End;

Procedure TListaTIM.Desenha(DC: HDC; Indice: Word);
Begin
    Linha(DC,Espectro^[Indice-1].tempo,Espectro^[Indice-1].tim,
        Espectro^[Indice].tempo,Espectro^[Indice].tim,cor);
End;

Procedure TListaTIM.Redesenha(DC: HDC);
var i : Word;
Begin
    For i := 2 to fim do Desenha(DC,i);
End;

Procedure TListaTIM.Acquire(DC: HDC);
var
    h,m,s,cs : Word;
    dt : Double;
Begin
    GetTime(h,m,s,cs);
    dt := 3600000.0 * h + 60000.0 * m + 1000.0 * s + 10.0 * cs; { dt em ms. }
    If dt < t0 then dt := dt + 24 * 3600000.0 - t0
    else dt := dt - t0;
    If dt < (Espectro^[pontoatual-1].tempo + intaquis) then EXIT;
    With Espectro^[pontoatual] do
        begin
            tempo := Round(dt); { tempo em ms. }
            tim := LeAbundancia;
            registro.tempo := tempo;
            Write(arq,registro);
        end;
    fim := pontoatual - 1;
    Desenha(DC, pontoatual);
    Inc(pontoatual);
    If pontoatual > Maxdados then
        begin
            MessageBeep(mb_ok);
            acabou := TRUE;
            esta_adquirindo := FALSE;
            Dec(pontoatual);
            ndados := pontoatual;
            fim := ndados;
            Xmax := Espectro^[ndados].tempo;
            Ymax := maxy;
            Close(arq);
            MedianaMoveI(Espectro,ndados);
            MediaMoveI(Espectro,ndados,3);
            MessageBeep(mb_ok);
            MessageBox(GetFocus,'Aquisição encerrada.','TIM',mb_OK);
        end;
    End;

Constructor TTIM.Init(AParent: PWindowsObject; ATitle: PChar;
    quad, s : Byte; mi,mf : Word; ti : Double; ia : Word;
    ei : Double; coment : String);
Begin
    TJanelaSemIcone.Init(AParent, ATitle);
```

```

        GraficoEspectro      :=      New(PListaTIM,
Init(MaxPontos,quad,s,mi,mf,ti,ia,ei,coment));
With GraficoEspectro^ do
begin
    tx := Attr.W;
    ty := Attr.H;
end;
End;

Destructor TTIM.Done;
var
    i : Byte;
Begin
    temTIM := FALSE;
    esta_adquirindo := FALSE;
    Dispose(GraficoEspectro^.PentaQ,Done);
    Dispose(GraficoEspectro,Done);
    TJanelaSemicone.Done;
End;

Procedure TTIM.WMSize(var Msg: TMessage);
var
    i : Integer;
Begin
    TJanelaSemicone.WMSize(Msg);
    With GraficoEspectro^ do
begin
    i := pontoatual;
    If (i > 2) and (not acabou) then
begin
        If Xmax < Espectro^[i-1].tempo then Xmax := 2 * Espectro^[i-
1].tempo;
        If Ymax < Espectro^[i-1].tim then Ymax := 2 * Espectro^[i-1].tim;
        end;
        tx := Msg.LParamLo; {Attr.W;}
        ty := Msg.LParamHi; {Attr.H;}
        end;
        InvalidateRect(HWindow, nil, TRUE);
    End;

Procedure TTIM.WMLButtonDown(var Msg: TMessage);
var
    mousex : Word;
    t : LongInt;
    i,k : Integer;
    titulo : Array[0..80] of Char;
    s : String[80];
    Dc : HDC;
Begin
    If not GraficoEspectro^ acabou then
begin
        With GraficoEspectro^ do
begin
            i := pontoatual;
            If Xmax < Espectro^[i-1].tempo then Xmax := 2 * Espectro^[i-
1].tempo;
            If Ymax < Espectro^[i-1].tim then Ymax := 2 * Espectro^[i-1].tim;
            tx := Attr.W;
            ty := Attr.H;
            end;
            InvalidateRect(HWindow, nil, TRUE);

```

```

end
else With GraficoEspectro^ do
begin
    SetCapture(HWindow);
    Dc := GetDC(HWindow); { Obtém o manipulador gráfico. }
    mousex := Integer(Msg.LParamLo); { Obtém posição x do mouse. }
    ReleaseCapture;
    t := Round((1.0 * mousex * Xmax) / (1.0 * tx)); { Converte em tempo. }
    MoveTo(Dc,mousex,0);
    LineTo(Dc,mousex,ty);
    ReleaseDC(HWindow, Dc);
    Assign(arq,pathdados + '\dados.tim');
    Reset(arq);
    Repeat
        Read(arq,registro);
    Until (registro.tempo >= t) or EOF(arq);
    Close(arq);
    Str(registro.tempo/1000.0:2,s);
    s := 'Tempo: ' + s + ' s';
    StrPCopy(titulo,s);
JF      :=
PGrafico(Application^.MakeWindow(New(PGrafico,Init(@Self,titulo))));
With JF^ do
begin
    mi := 10 * mzi;
    massainicial := mi;
    mf := 10 * mzf;
    massafinal := mf;
    maximo := 0;
    numdados := mzf - mzi + 1;
    For k := mzi to mzf do
begin
        i := k - mzi + 1;
        massa[i] := 10 * k;
        intens[i] := registro.abundancia[k];
        If maximo < intens[i] then maximo := intens[i];
        end;
    end;
end;
End;

Procedure TTIM.WMLButtonDbiClk(var Msg: TMessage);
Begin
    With GraficoEspectro^ do
begin
        If acabou then EXIT;
        acabou := TRUE;
        esta_adquirindo := FALSE;
        Dec(pontoatual);
        Close(arq);
        ndados := pontoatual;
        fim := ndados;
        Xmax := Espectro^[ndados].tempo;
        Ymax := maxy;
        MedianaMovel(Espectro,ndados);
        MediaMovel(Espectro,ndados,3);
        end;
        InvalidateRect(HWindow,nil,TRUE);
        MessageBeep(mb_ok);
        MessageBox(HWindow,'Aquisição encerrada.','TIM',mb_OK);
    End;

```

```
Procedure TTIM.Paint(PaintDC: HDC; var PaintInfo: TPaintStruct);
Begin
  TJanelaSemicone.Paint(PaintDC, PaintInfo);
  GraficoEspectro^.Redesenha(PaintDC);
End;

Procedure TTIM.Temporizador; { Executa o conteúdo do procedure periodicamente. }
var
  DC: HDC;
Begin
  If not GraficoEspectro^.acabou then
  begin
    DC := GetDC(HWindow);
    GraficoEspectro^.Adquire(DC);
    ReleaseDC(HWindow, DC);
  end;
End;

Constructor TGrafico.Init(AParent: PWindowsObject; ATitle: PChar);
var
  k : Word;
Begin
  ButtonDown := FALSE;
  xHi := 400;
  xLo := 30;
  yHi := 160;
  yLo := 30;
  maxy := 1000;
  TJanelaSemicone.Init(AParent, ATitle);
  With Attr do
  begin
    Style := ws_OverlappedWindow or ws_Visible;
    X := CW_USEDEFAULT; Y := CW_USEDEFAULT;
    W := xHi + xLo + 10; H := yHi + yLo + 30;
    Menu := LoadMenu(HInstance, 'menu_tim');
  end;
  expansao := 1000;
End;

Destructor TGrafico.Done;
Begin
  TJanelaSemicone.Done;
End;

Function TGrafico.GetClassName: PChar;
begin
  GetClassName := 'GraphWndClass'
end;

Procedure TGrafico.GetWindowClass(var AWndClass: TWndClass);
Begin
  TJanelaSemicone.GetWindowClass(AWndClass);
  AWndClass.HIcon := LoadIcon(HInstance, 'massa');
End;

Procedure TGrafico.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
  NewCur,
  OldCur : HCursor;
```

```
OldFont,
NewFont : HFont;
i,j,xi,incrm,posx,posy,oy : Integer;
yi,inten : Word;
s : string[95];
sc : Array[0..94] of Char;

Function Y(i : LongInt) : Integer;
Begin
  Y := Round(1000.0*i/maximo);
End;

Begin { TGrafico.Paint }
  NewCur := LoadCursor(0, idc_Wait); { Sinaliza com a ampulheta. }
  OldCur := SetCursor(NewCur);
  SetTextAlign(PaintDC, TA_RIGHT);
  SetTextColor(PaintDC, RGB(0,0,255));
  SetBkMode(PaintDC, transparent);
  SetMapMode(PaintDC, mm_Anisotropic);
  SetViewportOrg(PaintDC, xLo, yHi-30);
  SetViewportExt(PaintDC, xHi-xLo, -(yHi-30));
  SetWindowExt(PaintDC, mf-mi, maxy);
  SetWindowOrg(PaintDC, mi, 0);

  oy := -maxy div 50; { Deslocamento da linha base em 2%. }

  NewFont :=
CreateFont(Round(0.06*maxy), 0, 0, 0, FW_BOLD, 0, 0, 0, ANSI_CHARSET,

OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
  VARIABLE_PITCH OR FF_ROMAN, Tms Rmn);
  OldFont := SelectObject(PaintDC, NewFont);
  incrm := 100*(1 + Trunc((mf - mi)/1000.0 - 0.49)); { Incremento da escala de massa. }
  If (mf - mi) div 10 > 80 then posX := mi - incrm div 10
  else If (mf - mi) div 10 > 60 then posX := mi - incrm div 15
    else If (mf - mi) div 10 > 40 then posX := mi - incrm div 20
      else If (mf - mi) div 10 > 20 then posX := mi - incrm div 40
        else posX := mi - incrm div 60; { Posição x p/ intensidade. }

  Str(maxy div 10, s);
  StrPCopy(sc, s); { Numera as intensidades. }
  TextOut(PaintDC, posX, maxy, sc, StrLen(sc));
  Str(maxy div 20, s);
  StrPCopy(sc, s);
  TextOut(PaintDC, posX, maxy div 2, sc, StrLen(sc));
  TextOut(PaintDC, posX, 0, '0', 1);
  NewPen := CreatePen(ps_Solid, 0, RGB(0,0,255));
  OldPen := SelectObject(PaintDC, NewPen);
  MoveTo(PaintDC, mi, maxy);
  LineTo(PaintDC, mi, oy); { Desenha os eixos. }
  LineTo(PaintDC, mf, oy);
  Str(mi div 10, s);
  StrPCopy(sc, s);
  posy := -maxy div 15;
  TextOut(PaintDC, mi, posy, sc, StrLen(sc)); { Numera a 1a massa. }
  MoveTo(PaintDC, mi, posy);
  LineTo(PaintDC, mi, oy);
  i := 0;
  xi := mi;
  Repeat
    posy := -maxy div 25;
    inc(i);
  While (xi < mi + (i + 1) * incrm) and (xi < mf) do
```

```

begin
    xi := xi + increm div 10;
    MoveTo(PaintDC,xi,posy); { Subdivisões da escala de massa. }
    LineTo(PaintDC,xi,oy);
end;
xi := mi + i * increm; { Evita acúmulo de sobras de arredondamento. }
Str(xi div 10,s);
StrPCopy(sc,s);
posy := -maxy div 15;
TextOut(PaintDC,xi,posy,sc,StrLen(sc)); { Numera as massas. }
MoveTo(PaintDC,xi,posy);
LineTo(PaintDC,xi,oy);
Until xi > mf - increm;
For i := 0 to 10 do
begin
    yi := Round(i * maxy / 10.0);
    MoveTo(PaintDC,posx,yi); { Escala de intensidade. }
    LineTo(PaintDC,mi,yi)
end;
SelectObject(PaintDC,OldFont);
DeleteObject(NewFont);
SelectObject(PaintDC,OldPen);
DeleteObject(NewPen);
NewPen := CreatePen(ps_Solid,0,RGB(255,0,0));
OldPen := SelectObject(PaintDC,NewPen);
If maximo > 0 then
    For i := 1 to numdados do { Traça o espectro normalizando as
intensidades. }
        begin
            MoveTo(PaintDC,massa[i],0);
            If (massa[i] >= mi) and (massa[i] <= mf) then
                LineTo(PaintDC,massa[i],Y(intens[i]));
            end;
        end;
    SelectObject(PaintDC,OldPen);
    DeleteObject(NewPen);
    SetCursor(OldCur);
End;

Procedure TGráfico.WMLButtonDown(var Msg : TMessage);
Begin
    If not ButtonDown then
        begin
            mousexi := xLo; { Valores iniciais para a posição do mouse. }
            mousexf := xHi;
            mouseyf := 0;
            ButtonDown := TRUE;
            SetCapture(HWindow);
            Dc := GetDC(HWindow); { Obtém o manipulador gráfico. }
            mousexi := Msg.LParamLo; { Obtém a nova posição do mouse. }
            mouseyi := yHi - 30;
            MoveTo(Dc,mousexi,mouseyi);
            NewPen := CreatePen(ps_Dot,0,RGB(0,128,0)); { Linha pontilhada
verde. }
            OldPen := SelectObject(Dc,NewPen);
            end;
        End;

Procedure TGráfico.WMLButtonUp(var Msg : TMessage);
var troca : Word;
Begin
    If ButtonDown then

```

```

begin
    LineTo(Dc,mousexf,mouseyi); { Desenha o retângulo que define a
região que }
    LineTo(Dc,mousexf,mouseyf); { poderá ser ampliada. }
    LineTo(Dc,mousexi,mouseyf);
    LineTo(Dc,mousexi,mouseyi); { Define os valores para a expansão. }
    expansao := Word(Round(maxy * (mouseyf - yHi + 30) / (30 - yHi)));
    If expansao > 1000 then expansao := 1000;
    If expansao < 50 then expansao := 50; { intensidade mínima = 5% }
    expmi := Word(10*Trunc((mi+(mousexi - xLo) * (mf - mi))/(xHi -
xLo))/10.0));
    If expmi < massainicial then expmi := massainicial;
    expmf := Word(10*Trunc((mi+(mousexf - xLo) * (mf - mi))/(xHi -
xLo))/10.0));
    If expmf > massafinal then expmf := massafinal;
    If expmf <= expmi then
        begin
            troca := expmf;
            expmf := expmi;
            expmi := troca;
        end;
    If (expmf - expmi) < 100 then expmf := expmi + 100;
    SelectObject(Dc,OldPen);
    DeleteObject(NewPen);
    ReleaseCapture;
    ReleaseDC(HWindow,Dc);
    ButtonDown := FALSE;
end;
End;

Procedure TGráfico.WMRButtonDown(var Msg : TMessage);
Begin
    mi := expmi;
    mf := expmf;
    maxy := expansao;
    InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro expandido.
}
End;

Procedure TGráfico.WMMouseMove(var Msg : TMessage);
Begin
    If ButtonDown then With Msg do
        begin
            mousexf := Integer(LParamLo); { Captura as posições do mouse em
movimento.}
            mouseyf := Integer(LParamHi);
            end;
        End;

Procedure TGráfico.WMLButtonDbClick(var Msg : TMessage);
Begin
    maxy := 1000;
    mi := massainicial;
    mf := massafinal;
    InvalidateRect(HWindow, nil, TRUE); { Redesenha o espectro após o
duplo click. }
End;

Procedure TGráfico.Copia(var Msg : TMessage);
Begin
    InvalidateRect(HWindow,nil,TRUE);

```

```

CopiaBitmap(HWindow,DC,410,155)
End;

Procedure TGráfico.Print(var Msg : TMessage);
var PInfo : TPaintStruct;
    imp : String;
    NewFont, OldFont : HFont;
Begin
    If MessageBox(HWindow,'Imprime o espectro?', 'Imprimir',
        mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
    If PrnPartida('Espectro Unidimensional',imp) then
        begin
            { Verifica se a impressora é HP. }
            If (imp[1] = 'H') and (imp[2] = 'P') then
                begin
                    xLo := 100;
                    xHi := 1500; { Valores para as impressoras lasers. }
                    yHi := 600;
                end;
            NewFont := CreateFont(20,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
    VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
            OldFont := SelectObject(PDc,NewFont);
            SetTextAlign(PDc,TA_Right);
            TextOut(PDc,xHi,yHi+90,Attr.Title,StrLen(Attr.Title));
            SelectObject(PDc,OldFont);
            DeleteObject(NewFont);
            Paint(PDc,PInfo);
            NovaPagina;
            PrnFim;
            xLo := 30; { Valores para o video e para as impressoras matriciais. }
            xHi := 400;
            yHi := 160;
        end;
    End;

Procedure TGráfico.Salva(var Msg : TMessage);
var
    i : Word;
    arqtext : Text;
    nome : String;
    sc : Array[0..100] of Char;
Begin
    nome := StrPas(Attr.Title);
    nome := 't'+ Copy(nome,pos(':',),nome)+2,Length(nome)-10) + '.DAT';
    If MessageBox(HWindow,'Salva o espectro?', 'Salvar',
        mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
    Assign(arqtext,nome);
    {$I-} Reset(arqtext); {$I+}
    If IOResult = 0 then
        begin
            StrPCopy(sc,'Arquivo ' + nome + ' já existe. Sobrepõe?');
            If MessageBox(HWindow,sc,'Salvar', mb_defbutton2 or
                mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
        end;
    Rewrite(arqtext);
    WriteIn(arqtext,maximo);
    WriteIn(arqtext,StrPas(Attr.Title));
    For i := 1 to numdados do WriteIn(arqtext,masa[i]/10.0:0:1,#9,intens[i]);
    Close(arqtext);
    nome := StrPas(Attr.title) + #13 + 'foi salvo como:' + nome;

```

```

StrPCopy(sc,nome);
MessageBox(HWindow,sc,'Salvar',mb_Ok or mb_IconInformation);
End;

{ -----TEntraTIM----- }
Type
    PEntraTIM = ^TEntraTIM;
    TEntraTIM = Object(TJanelaSemicone)
        mzi,mzf,tempoint,coment, intaquis, enion : PEdit;
        sensio, sensmed, senshi : PRadioButton;
        GrupoSens : PGroupBox;
        quad : PComboBox;
        Constructor Init(AParent : PWindowsObject; ATitle : PChar);
        Procedure SetupWindow; virtual;
        Destructor Done; virtual;
        Procedure Paint(PaintDC : HDC; var PaintInfo : TPaintStruct); virtual;
    End;

Constructor TEntraTIM.Init(AParent : PWindowsObject; ATitle : PChar);
Begin
    TJanelaSemicone.Init(AParent, ATitle);
    With Attr do
        begin
            X := CW_USEDEFAULT; Y := CW_USEDEFAULT; W := 420; H := 290;

                                quad :=
New(PComboBox,Init(@Self,EditBoxID,115,0,50,100,cbs_DropDownList,0))
;
            quad^.Attr.Style := quad^.Attr.Style and (not cbs_Sort);
            GrupoSens := New(PGroupBox, Init(@Self, GrupoSensID,"",0,0,0,0));
                sensio := New(PRadioButton, Init(@Self, sensioID,
'B',125,35,40,20,GrupoSens));
                sensmed := New(PRadioButton, Init(@Self, sensioID,
'M',165,35,40,20,GrupoSens));
                senshi := New(PRadioButton, Init(@Self,
sensioID,'A',205,35,40,20,GrupoSens));
            mzi := New(PEdit,Init(@Self,EditBoxID,'20',105,70,50,30,5,FALSE));
            mzf := New(PEdit,Init(@Self,EditBoxID,'300',240,70,50,30,5,FALSE));

                                tempoint :=
New(PEdit,Init(@Self,EditBoxID,'1.0',205,105,50,30,5,FALSE));
            intaquis := New(PEdit,Init(@Self,EditBoxID,'50',215,140,60,30,6,FALSE));
            enion := New(PEdit,Init(@Self,EditBoxID,'70.0',210,175,50,30,5,FALSE));
            coment := New(PEdit,Init(@Self,EditBoxID,"",185,210,220,30,81,FALSE));
        end;

Procedure TEntraTIM.SetupWindow;
Begin
    TJanelaSemicone.SetupWindow;
    quad^.AddString('1');
    quad^.AddString('3');
    quad^.AddString('5');
    quad^.SetSelIndex(2); { 0,1,2 } { Default quad = 5. }
    sensmed^.SetCheck(1); { 0,1,2 } { Default sensibilidade = media. }
End;

Destructor TEntraTIM.Done;
Begin
    temTIM := FALSE;
    TJanelaSemicone.Done;
End;

```

```
Procedure TEntraTIM.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
  OldBrush,
  NewBrush : HBrush;
  NewFont,
  OldFont : HFont;
  OldPen : HPen;
  i : Byte;
Begin
  If IsIconic(HWindow) then
    begin
      OldPen := SelectObject(PaintDC, CreatePen(PS_SOLID, 1,
RGB(0,128,0)));
      i := 0;
      Repeat
        inc(i,2);
        Rectangle(PaintDC,i,i,80-i,80-i);
      Until i = 50;
      DeleteObject(SelectObject(PaintDC, OldPen));
    end
  else
    begin
      NewFont := CreateFont(20,0,0,0,FW_NORMAL,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
      VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
      OldFont := SelectObject(PaintDC,NewFont);
      SetTextColor(PaintDC,RGB(0,128,0));
      SetBkMode(PaintDC,transparent);
      SetTextAlign(PaintDC,TA_LEFT);
      TextOut(PaintDC,25,5,'Quadrupolo:',11);
      TextOut(PaintDC,25,35,'Sensibilidade:',14);
      TextOut(PaintDC,25,75,'m/z inicial:',12);
      TextOut(PaintDC,170,75,'m/z final:',10);
      TextOut(PaintDC,25,110,'Tempo de integração (ms):',25);
      TextOut(PaintDC,25,145,'Intervalo de aquisição (ms):',28);
      TextOut(PaintDC,25,180,'Energia de ionização (eV):',26);
      TextOut(PaintDC,25,215,'Comentário (80 letras):',23);
      SelectObject(PaintDC,OldFont);
      DeleteObject(NewFont);
    end;
End;

{-----TIEM-----}
Type
TDadoIEM = Record
  IEM : LongInt;
  enion : Double;
end;
TRegistrolEM = Record
  enion : Double;
  abundancia : Array[1..1000] of LongInt;
  varianca : Array[1..1000] of LongInt;
end;
PEspectrolEM = ^TEspectrolEM;
TEspectrolEM = Array[1..MaxPontos] of TDadoIEM;
PListalEM = ^TListalEM;
TListalEM = Object(TObject)
  Espectro : PEspectrolEM;
  registro : TRegistrolEM;
  arq : File of TRegistrolEM;
```

```
PentaQ : PPentaQuad;
quad : Array[um..cinco] of Boolean;
Xmin, Xmax, Ymax, cor, maxy : LongInt;
pontoatual, fim, maxdados, tx, ty, ndados : Integer;
acabou : Boolean;
tempointeg, enerion, intervenion : Double;
mzi, mzf : Word;
sensibilidade : Byte;
quadrup : Quadrupolos;
comentario : String[80];
vf : Array[1..MaxPontos] of LongInt;
Constructor Init(Max : Integer; q,sens : Byte; mi,mf : Word;
  ti,iei,eii,eif : Double; c : String);
Destructor Done; virtual;
Procedure MedianaMovl(var vo : PEspectrolEM; n : Word);
  Procedure MediaMovl(var vo : PEspectrolEM; n : Word; janela :
Integer);
  Function LeAbundancia : LongInt;
  Procedure Linha(DC: HDC; a1, b1, a2, b2 : Double; corlinha: LongInt);
  Procedure Desenha(DC: HDC; Indice: Word); virtual;
  Procedure Redesenha(DC: HDC);
  Procedure Adquire(DC: HDC);
End;

PIEM = ^TIEM;
TIEM = Object(TJanelaSemicone)
  GraficoEspectro : PListalEM;
  JF : PGrafico;
  Constructor Init(AParent: PWindowsObject; ATitle: PChar;
    quad, s : Byte; mi,mf : Word; ti, iei,eii,eif : Double;
    coment : String);
  Destructor Done; virtual;
  Procedure Paint(PaintDC: HDC; var PaintInfo: TPaintStruct); virtual;
  Procedure WMLButtonDown(var Msg : TMessage); virtual wm_First +
wm_LButtonDown;
  Procedure WMLButtonDbiCik(var Msg : TMessage); virtual wm_First +
wm_LButtonDbiCik;
  Procedure WMSize(var Msg: TMessage); virtual wm_First + wm_Size;
  Procedure Temporizador; virtual;
End;

Constructor TListalEM.Init(Max : Integer; q,sens : Byte; mi,mf : Word;
  ti,iei,eii,eif : Double; c : String);
var
  h,m,s,cs : Word;
  quadr : quadrupolos;
Begin
  TObject.Init;
  If Max > MaxPontos then Max := MaxPontos;
  maxdados := Max;
  cor := RGB(0,0,255);
  GetMem(Espectro, SizeOf(TDadoRec) * Maxdados);
  quad[um] := q = 1;
  quad[tres] := q = 3;
  quad[cinco] := q = 5;
  sensibilidade := sens;
  mzi := mi;
  mzf := mf;
  tempointeg := ti;
  intervenion := iei;
  enerion := eii;
  comentario := c;
```



```
New(PentaQ,Init(path,'Q1.CAL','Q3.CAL','Q5.CAL'));
PentaQ^.det.SetTempInteg(tempointeg);
PentaQ^.Ionizador.El(enerion);
Port{$318} := sensibilidade;
For quadr := um to cinco do
  If quad[quadr] then
    begin
      PentaQ^.SetPoleDC(quadr,TRUE);
      quadrup := quadr;
    end
  else PentaQ^.SetPoleDC(quadr,FALSE);
If not PentaQ^.LinhaBase then
  MessageBox(GetFocus,'Problemas com a linha-base, interrompa a
aquisição.',
  'IEM',mb_OK);
pontoatual := 2;
fim := 0;
ndados := 0;
Xmin := Round(eii); { Xmin e Xmax servem para o tracar o gráfico e
armazenar }
Xmax := Round(eif); { os valores das energias de ionização inicial e final. }
Ymax := 20000;
maxy := 0;
acabou := FALSE;
Assign(arq,pathdados + '\dados.IEM');
ReWrite(arq);
With Espectro^[1] do
  begin
    enion := enerion;
    IEM := LeAbundancia;
    registro.enion := enerion;
    Write(arq,registro);
  end;
End;

Destructor TListalIEM.Done;
Begin
  FreeMem(Espectro, SizeOf(TDadoRec) * Maxdados);
  TObject.Done;
End;

Procedure TListalIEM.MedianaMoveI(var vo : PESpectroIEM; n : Word);
var
  i : Integer;
  sai, entra, pilha : LongInt;
  d : Array[-1..1] of LongInt; { A posicao 0 e' a mediana. }
  pronto : Boolean;
Begin
  If n < 3 then Exit; { Devem existir pelo menos 3 dados. }
  For i := -1 to 1 do d[i] := vo^[i+2].IEM;
  pronto := TRUE;
  Repeat
    pronto := TRUE;
    For i := -1 to 0 do
      If d[i] > d[i+1] then
        begin
          pronto := False;
          pilha := d[i];
          d[i] := d[i+1];
          d[i+1] := pilha;
        end;
    end;
```

```
Until pronto;
vf[1] := d[0];
vf[2] := d[0];
For i := 3 to n - 1 do
  begin
    sai := vo^[i - 2].IEM;
    entra := vo^[i + 1].IEM;
    If sai = d[-1] then
      If entra > d[0] then
        If entra > d[1] then
          begin
            d[-1] := d[0];
            d[0] := d[1];
            d[1] := entra;
          end
        else
          begin
            d[-1] := d[0];
            d[0] := entra;
          end
        else d[-1] := entra
      else
        If sai = d[0] then
          If entra > d[-1] then
            If entra > d[1] then
              begin
                d[0] := d[1];
                d[1] := entra;
              end
            else d[0] := entra
          else
            begin
              d[0] := d[-1];
              d[-1] := entra;
            end
          else
            If entra >= d[-1] then
              If entra >= d[0] then d[1] := entra
            else
              begin
                d[1] := d[0];
                d[0] := entra;
              end
            else
              begin
                d[1] := d[0];
                d[0] := d[-1];
                d[-1] := entra;
              end;
            end;
          vf[i] := d[0];
        end;
    vf[n] := vf[n-1];
    For i := 3 to n-3 do { O loop For i := 1 to n do achata as pontas }
      vo^[i].IEM := vf[i];
    End;

Procedure TListalIEM.MedianaMoveI(var vo : PESpectroIEM; n : Word; janela :
Integer);
var i : Word;
    j : Integer;
    soma : Double;
```

```
Begin
If n < (2 * janela + 1) then EXIT;
soma := 0.0;
i := janela + 1;
For j := -janela to janela do
    soma := soma + 1.0 * vo^[i+j].IEM;
vf[i] := Round(soma / (2.0 * janela + 1));
Repeat
    Inc(i);
    soma := soma - 1.0 * vo^[i - janela - 1].IEM
        + 1.0 * vo^[i + janela].IEM;
    vf[i] := Round(soma / (2.0 * janela + 1));
Until i = n - janela;
For i := janela + 1 to n - janela do
    vo^[i].IEM := vf[i];
End;

Function TListaIEM.LeAbundancia : LongInt;
var
    i,j : Integer;
    soma : LongInt;
    abunda, varianca : Double;
Begin
    soma := 0;
    For i := mzi to mzf do
        begin
            PentaQ^.SetMassa(quadруп,i);
            j := 8;
            Repeat
                Dec(j);
                PentaQ^.det.ganho := ganhos[j];
                PentaQ^.LeMedia(3,abunda,varianca);
            Until (not PentaQ^.det.saturado) or (j = 1);
            If abunda < 0 then abunda := 0;
            registro.abundancia[i] := Round(abunda);
            registro.varianca[i] := Round(varianca);
            soma := soma + registro.abundancia[i];
        end;
    If maxy < soma then maxy := soma;
    LeAbundancia := soma;
End;

Procedure TListaIEM.Linha(DC: HDC; a1,b1,a2,b2 : Double; corlinha:
LongInt);
Function XP(v : Double) : Integer;
Begin
    XP := Integer(Round((v - Xmin) * tx / (Xmax - Xmin)));
End;

Function YP(v : Double) : Integer;
Begin
    YP := Integer(Round(-ty*(v - Ymax) / Ymax));
End;

var
    OldPen: HPen;
Begin
    OldPen := SelectObject(DC, CreatePen(PS_SOLID, 1, corlinha));
    MoveTo(DC, XP(a1), YP(b1));
    LineTo(DC, XP(a2), YP(b2));
    DeleteObject(SelectObject(DC, OldPen));
```

```
End;

Procedure TListaIEM.Desenha(DC: HDC; Indice: Word);
Begin
    Linha(DC,Espectro^[Indice-1].enion,Espectro^[Indice-1].IEM,
        Espectro^[Indice].enion,Espectro^[Indice].IEM,cor);
End;

Procedure TListaIEM.Redesenha(DC: HDC);
var i : Word;
Begin
    For i := 2 to fim do Desenha(DC,i);
End;

Procedure TListaIEM.Acquire(DC: HDC);
Begin
    enerion := enerion + intervenion;
    PentaQ^.Ionizador.El(enerion);
    With Espectro^[pontoatual] do
        begin
            IEM := LeAbundancia;
            enion := enerion;
            registro.enion := enerion;
            Write(arq,registro);
        end;
    fim := pontoatual - 1;
    Desenha(DC, pontoatual);
    Inc(pontoatual);
    If (pontoatual > Maxdados) or (enerion >= Xmax) then
        begin
            acabou := TRUE;
            esta_adquirindo := FALSE;
            Dec(pontoatual);
            ndados := pontoatual;
            fim := ndados;
            Xmax := Round(Espectro^[ndados].enion);
            Ymax := maxy;
            Close(arq);
            MedianaMovel(Espectro,ndados);
            MediaMovel(Espectro,ndados,3);
            MessageBeep(mb_ok);
            MessageBox(GetFocus,'Aquisição encerrada.','IEM',mb_OK);
        end;
End;

Constructor TIEM.Init(AParent: PWindowsObject; ATitle: PChar;
    quad, s : Byte; mi,mf : Word; ti,lei,eii,eif : Double;
    coment : String);
Begin
    TJanelaSemicone.Init(AParent, ATitle);
        GraficoEspectro := New(PListaIEM,
Init(MaxPontos,quad,s,mi,mf,ti,lei,eii,eif,coment));
    With GraficoEspectro^ do
        begin
            tx := Attr.W;
            ty := Attr.H;
        end;
End;

Destructor TIEM.Done;
var
```

```

i : Byte;
Begin
temIEM := FALSE;
esta_adquirindo := FALSE;
Dispose(GraficoEspectro^.PentaQ,Done);
Dispose(GraficoEspectro,Done);
TJanelaSemicone.Done;
End;

Procedure TIEM.WMSize(var Msg: TMessage);
var
i : Integer;
Begin
TJanelaSemicone.WMSize(Msg);
With GraficoEspectro^ do
begin
i := pontoatual;
If (i > 2) and (not acabou) then
If Ymax < Espectro^[i-1].IEM then Ymax := 2 * Espectro^[i-1].IEM;
tx := Msg.LParamLo; {Attr.W;}
ty := Msg.LParamHi; {Attr.H;}
end;
InvalidateRect(HWindow, nil, TRUE);
End;

Procedure TIEM.WMLButtonDown(var Msg : TMessage);
var
mousex : Word;
t : Double;
i,k : Integer;
titulo : Array[0..80] of Char;
s : String[80];
Dc : HDC;
Begin
If not GraficoEspectro^.acabou then
begin
With GraficoEspectro^ do
begin
i := pontoatual;
If Ymax < Espectro^[i-1].IEM then Ymax := 2 * Espectro^[i-1].IEM;
tx := Attr.W;
ty := Attr.H;
end;
InvalidateRect(HWindow, nil, TRUE);
end
else With GraficoEspectro^ do
begin
SetCapture(HWindow);
Dc := GetDC(HWindow); { Obtém o manipulador gráfico. }
mousex := Integer(Msg.LParamLo); { Obtém posição x do mouse. }
ReleaseCapture;
t := (1.0 * mousex * Xmax) / (1.0 * tx); { Converte em enion. }
MoveTo(Dc,mousex,0);
LineTo(Dc,mousex,ty);
ReleaseDC(HWindow, Dc);
Assign(arq,pathdados + '\dados.iem');
Reset(arq);
Repeat
Read(arq,registro);
Until (registro.enion >= t) or EOF(arq);
Close(arq);

```

```

Str(registro.enion:0:1,s);
s := 'Energia de Ionização: ' + s + ' eV';
StrPCopy(titulo,s);
JF :=
PGrafico(Application^.MakeWindow(New(PGrafico,Init(@Self,titulo))));
With JF^ do
begin
mi := 10 * mzi;
massainicial := mi;
mf := 10 * mzf;
massafinal := mf;
maximo := 0;
numdados := mzf - mzi + 1;
For k := mzi to mzf do
begin
i := k - mzi + 1;
massa[i] := 10 * k;
intens[i] := registro.abundancia[k];
If maximo < intens[i] then maximo := intens[i];
end;
end;
End;

Procedure TIEM.WMLButtonDbClick(var Msg : TMessage);
Begin
With GraficoEspectro^ do
begin
If acabou then EXIT;
acabou := TRUE;
esta_adquirindo := FALSE;
Dec(pontoatual);
Close(arq);
ndados := pontoatual;
fim := ndados;
Xmax := Round(Espectro^[ndados].enion);
Ymax := maxy;
MedianaMovel(Espectro,ndados);
MediaMovel(Espectro,ndados,3);
end;
InvalidateRect(HWindow,nil,TRUE);
MessageBeep(mb_ok);
MessageBox(HWindow,'Aquisição encerrada.','IEM',mb_OK);
End;

Procedure TIEM.Paint(PaintDC: HDC; var PaintInfo: TPaintStruct);
Begin
TJanelaSemicone.Paint(PaintDC, PaintInfo);
GraficoEspectro^.Redesenha(PaintDC);
End;

Procedure TIEM.Temporizador; { Executa o conteúdo do procedure periodicamente. }
var
DC: HDC;
Begin
If not GraficoEspectro^.acabou then
begin
DC := GetDC(HWindow);
GraficoEspectro^.Acquire(DC);
ReleaseDC(HWindow, DC);

```

```

end;
End;

{ -----TEntalEM----- }
Type
PEntalEM = ^TEntalEM;
TEntalEM = Object(TJanelaSemlcone)
    mzi,mzf,eii,eif,tempoint,coment, inteI, enion : PEdit;
    senslo, sensmed, senshi : PRadioButton;
    GrupoSens : PGroupBox;
    quad : PComboBox;
    Constructor Init(AParent : PWindowsObject; ATitle : PChar);
    Procedure SetupWindow; virtual;
    Destructor Done; virtual;
    Procedure Paint(PaintDC : HDC; var PaintInfo : TPaintStruct); virtual;
End;

Constructor TEntalEM.Init(AParent : PWindowsObject; ATitle : PChar);
Begin
    TJanelaSemlcone.Init(AParent, ATitle);
    With Attr do
        begin
            X := CW_USEDEFAULT; Y := CW_USEDEFAULT; W := 420; H := 290;
        end;
        quad :=
New(PComboBox,Init(@Self,EditBoxID,115,0,50,100,cbs_DropDownList,0))
;
        quad^.Attr.Style := quad^.Attr.Style and (not cbs_Sort);
        GrupoSens := New(PGroupBox, Init(@Self, GrupoSensID,"0,0,0,0));
        senslo := New(PRadioButton, Init(@Self, sensID,
'B',125,35,40,20,GrupoSens));
        sensmed := New(PRadioButton, Init(@Self, sensID,
'M',165,35,40,20,GrupoSens));
        senshi := New(PRadioButton, Init(@Self,
sensID,'A',205,35,40,20,GrupoSens));
        mzi := New(PEdit,Init(@Self,EditBoxID,'20',105,70,50,30,5,FALSE));
        mzf := New(PEdit,Init(@Self,EditBoxID,'300',240,70,50,30,5,FALSE));
        eii := New(PEdit,Init(@Self,EditBoxID,'5',130,105,40,30,4,FALSE));
        eif := New(PEdit,Init(@Self,EditBoxID,'100',275,105,40,30,4,FALSE));
        inteI := New(PEdit,Init(@Self,EditBoxID,'5.0',190,140,50,30,5,FALSE));
        tempoint :=
New(PEdit,Init(@Self,EditBoxID,'1.0',205,175,50,30,5,FALSE));
        coment := New(PEdit,Init(@Self,EditBoxID,"",185,210,220,30,81,FALSE));
    End;

    Procedure TEntalEM.SetupWindow;
    Begin
        TJanelaSemlcone.SetupWindow;
        quad^.AddString('1');
        quad^.AddString('3');
        quad^.AddString('5');
        quad^.SetSelIndex(2); { 0,1,2 } { Default quad = 5. }
        sensmed^.SetCheck(1); { 0,1,2 } { Default sensibilidade = media. }
    End;

    Destructor TEntalEM.Done;
    Begin
        termEM := FALSE;
        TJanelaSemlcone.Done;
    End;

```

```

Procedure TEntalEM.Paint(PaintDC : HDC; var PaintInfo : TPaintStruct);
var
    OldBrush,
    NewBrush : HBrush;
    NewFont,
    OldFont : HFont;
    OldPen : HPen;
    i : Byte;
Begin
    If IsIconic(HWindow) then
        begin
            OldPen := SelectObject(PaintDC, CreatePen(PS_SOLID, 1,
RGB(0,128,128)));
            i := 0;
            Repeat
                inc(i,2);
                Rectangle(PaintDC,i,i,80-i,80-i);
            Until i = 50;
            DeleteObject(SelectObject(PaintDC, OldPen));
        end
    else
        begin
            NewFont := CreateFont(20,0,0,0,FW_NORMAL,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
            OldFont := SelectObject(PaintDC,NewFont);
            SetTextColor(PaintDC,RGB(0,128,128));
            SetBkMode(PaintDC,transparent);
            SetTextAlign(PaintDC,TA_LEFT);
            TextOut(PaintDC,25,5,'Quadrupolo:',11);
            TextOut(PaintDC,25,35,'Sensibilidade:',14);
            TextOut(PaintDC,25,75,'m/z inicial:',12);
            TextOut(PaintDC,170,75,'m/z final:',10);
            TextOut(PaintDC,25,110,'EI inicial (eV):',16);
            TextOut(PaintDC,180,110,'EI final (eV):',14);
            TextOut(PaintDC,25,145,'Incremento da E I (eV):',23);
            TextOut(PaintDC,25,180,'Tempo de integração (ms):',25);
            TextOut(PaintDC,25,215,'Comentário (80 letras):',23);
            SelectObject(PaintDC,OldFont);
            DeleteObject(NewFont);
        end;
    End;
End;

{ -----TPrincipal----- }
Type
TFileName = Array [0..255] of Char;
PPrincipal = ^TPrincipal;
TPrincipal = Object(TMDIWindow)
    GraficoSIM : Array[1..6] of PSIM;
    DadosSIM : Array[1..6] of PEntraSIM;
    DadosTIM : PEntraTIM;
    GraficoTIM : PTIM;
    DadosIEM : PEntalEM;
    GraficoIEM : PIEM;
    DadosCalibracao : PCalibra;
    ok_arquivo : Boolean;
    FileName : TFileName;
    Procedure SetupWindow; virtual;
    Procedure EntraSIM(var Msg: TMessage); virtual cm_First + simID;

```

```

    Procedure EntraCalibracao(var Msg : TMessage); virtual cm_First +
calbralID;

    Procedure EntraTIM(var Msg : TMessage); virtual cm_First + timID;
    Procedure EntralEM(var Msg : TMessage); virtual cm_First + iemID;
    Procedure Executa(var Msg : TMessage); virtual cm_First + InicialD;
    Procedure CMFileOpen(var Msg : TMessage);
    Procedure Salvar(var Msg : TMessage); virtual cm_First + SalvarID;
    Procedure Print(var Msg : TMessage); virtual cm_First + PrintID;
    Procedure Sair(var Msg : TMessage); virtual cm_First + SairID;
    Procedure Sobre(var Msg : TMessage); virtual cm_First + SobreID;
    Procedure WMTimer(var Msg : TMessage); virtual wm_First +
wm_Timer;
    Procedure WMDestroy(var Msg : TMessage); virtual wm_First +
wm_Destroy;
End;

```

```

Procedure TPrincipal.SetupWindow;
var
    Result: Integer;
Begin
    TMDIWindow.SetupWindow;
    Result := IDRetry;
    While (SetTimer(hWindow, 0, 1, nil) = 0) and (Result = IDRetry) do
        Result := MessageBox(GetFocus,'Não é possível inicializar', 'Programa
SIMTIM',
            mb_RetryCancel);
    If Result = IDCCancel then PostQuitMessage(0);
End;

```

```

Procedure TPrincipal.EntraSIM(var Msg : TMessage);
var
    s : String[40];
    c : Array[0..40] of Char;
Begin
    If temTIM then EXIT;
    If numion = 6 then EXIT;
    StrCopy(FileName, "");
    Inc(numion);
    Str(numion, s);
    s := 'SIM: Entrada de Dados - ' + s;
    StrPCopy(c, s);
    DadosSIM[numion] := PEntraSIM(Application^.MakeWindow(
        New(PEntraSIM, Init(@Self, c))));
    DadosSIM[numion]^Indice := numion;
End;

```

```

Procedure TPrincipal.EntraCalibracao(var Msg : TMessage);
Begin
    If temcalibracao or temTIM then EXIT;
    DadosCalibracao := PCalibra(Application^.MakeWindow(
        New(PCalibra, Init(@Self, 'Calibração')));
    temcalibracao := TRUE;
End;

```

```

Procedure TPrincipal.EntraTIM(var Msg : TMessage);
Begin
    If temTIM or temIEM then EXIT;
    DadosTIM := PEntraTIM(Application^.MakeWindow(New(PEntraTIM,
        Init(@Self, 'Entrada de Dados - T I M')));
    temTIM := TRUE;
End;

```

```

Procedure TPrincipal.EntralEM(var Msg : TMessage);
Begin
    If temIEM or temTIM then EXIT;
    DadosIEM := PEntralEM(Application^.MakeWindow(New(PEntralEM,
        Init(@Self, 'Entrada de Dados - I E M')));
    temIEM := TRUE;
End;

```

```

Procedure TPrincipal.Executa(var Msg : TMessage);
var c : TTitulo;
    st, coment : String[80];
    i, err : Integer;
    q, s : Byte;
    mi, mf, ia : Word;
    eii, eif, iei, ti, ei : Double;
Begin
    If temTIM then
        begin
            If esta_adquirindo then EXIT;
            DadosTIM^.quad^.GetSelString(c, 1);
            st := StrPas(c);
            Val(st, q, err);
            If DadosTIM^.senslo^.GetCheck = 1 then s := 1;
            If DadosTIM^.sensmed^.GetCheck = 1 then s := 3;
            If DadosTIM^.senshi^.GetCheck = 1 then s := 2;
            DadosTIM^.mzi^.GetText(c, 4);
            st := StrPas(c);
            Val(st, mi, err);
            DadosTIM^.mzf^.GetText(c, 4);
            st := StrPas(c);
            Val(st, mf, err);
            DadosTIM^.tempoint^.GetText(c, 6);
            st := StrPas(c);
            Val(st, ti, err);
            DadosTIM^.intaquis^.GetText(c, 6);
            st := StrPas(c);
            Val(st, ia, err);
            DadosTIM^.enion^.GetText(c, 10);
            st := StrPas(c);
            Val(st, ei, err);
            DadosTIM^.coment^.GetText(c, 80);
            coment := StrPas(c);
            esta_adquirindo := TRUE;
            GraficoTIM := PTIM(Application^.MakeWindow(New(PTIM,
                Init(@Self, 'Aquisição de Dados - T I M',
                    q, s, mi, mf, ti, ia, ei, coment))));
        end;
    If temIEM then
        begin
            If esta_adquirindo then EXIT;
            DadosIEM^.quad^.GetSelString(c, 1);
            st := StrPas(c);
            Val(st, q, err);
            If DadosIEM^.senslo^.GetCheck = 1 then s := 1;
            If DadosIEM^.sensmed^.GetCheck = 1 then s := 3;
            If DadosIEM^.senshi^.GetCheck = 1 then s := 2;
            DadosIEM^.mzi^.GetText(c, 4);
            st := StrPas(c);
            Val(st, mi, err);
            DadosIEM^.mzf^.GetText(c, 4);

```

```

st := StrPas(c);
Val(st,mf,err);
DadosIEM^.tempoint^.GetText(c,6);
st := StrPas(c);
Val(st,ti,err);
DadosIEM^.eii^.GetText(c,10);
st := StrPas(c);
Val(st,eii,err);
DadosIEM^.eif^.GetText(c,10);
st := StrPas(c);
Val(st,eif,err);
DadosIEM^.intei^.GetText(c,6);
st := StrPas(c);
Val(st,iei,err);
DadosIEM^.coment^.GetText(c,80);
coment := StrPas(c);
esta_adquirindo := TRUE;
GraficoIEM := PIEM(Application^.MakeWindow(New(PIEM,
    Init(@Self,'Aquisição de Dados - I E M',
    q,s,mi,mf,ti,iei,eii,eif,coment))));
end;
If (numion > 6) or (esta_adquirindo) then EXIT;
numgraficosim := 0;
For i := 1 to numion do If DadosSIM[i]^indice = i then
begin
    Inc(controle);
    Inc(numgraficosim);
    DadosSIM[i]^quad^.GetSelString(c,1);
    st := StrPas(c);
    Val(st,q,err);
    If DadosSIM[i]^senslo^.GetCheck = 1 then s := 1;
    If DadosSIM[i]^sensmed^.GetCheck = 1 then s := 3;
    If DadosSIM[i]^senshi^.GetCheck = 1 then s := 2;
    DadosSIM[i]^mz^.GetText(c,4);
    st := StrPas(c);
    Val(st,mi,err);
    DadosSIM[i]^tempoint^.GetText(c,6);
    st := StrPas(c);
    Val(st,ti,err);
    DadosSIM[i]^intaquis^.GetText(c,6);
    st := StrPas(c);
    Val(st,ia,err);
    DadosSIM[i]^enion^.GetText(c,10);
    st := StrPas(c);
    Val(st,ei,err);
    DadosSIM[i]^coment^.GetText(c,80);
    coment := StrPas(c);
    DadosSIM[i]^mz^.GetText(c,4);
    Str(i,st);
    st := 'Aquisição '+ st;
    st := st + ' - m/z: ' + StrPas(c);
    StrPCopy(c,st);
    GraficoSIM[i] := PSIM(Application^.MakeWindow(New(PSIM,
        Init(@Self,c,q,s,mi,mf,ti,ia,ei,coment))));
    GraficoSIM[i]^titulo := c;
    GraficoSIM[i]^indice := i;
    esta_adquirindo := TRUE;
end;
End;

```

```

Procedure TPrincipal.CMFileOpen(var Msg: TMessage);

```

```

const
    DefExt = 'dat';
var
    OpenFN      : TOpenFileName;
    Filter       : array [0..100] of Char;
    FullFileName: TFileName;
Begin
    ok_arquivo := FALSE;
    StrCopy(FullFileName, "");
    FillChar(Filter, SizeOf(Filter), #0);
    StrCopy(Filter, 'Dados de SIMTIM');
    StrCopy(@Filter[StrLen(Filter)+1], '*.dat');
    FillChar(OpenFN, SizeOf(TOpenFileName), #0);
    With OpenFN do
        begin
            hInstance := HInstance;
            hwndOwner  := HWindow;
            lpstrDefExt := DefExt;
            lpstrFile   := FullFileName;
            lpstrFilter  := Filter;
            lpstrFileTitle := FileName;
            flags       := ofn_FileMustExist;
            lStructSize := sizeof(TOpenFileName);
            nFilterIndex := 1; {Índice dentro da String de filtragem em lpstrFilter.}
            nMaxFile    := SizeOf(FullFileName);
        end;
    ok_arquivo := GetSaveFileName(OpenFN);
End;

Procedure TPrincipal.Salvar(var Msg: TMessage);
var
    ftim : File of TRegistro;
    blocotim : TRegistro;
    fiem : File of TRegistroiem;
    blocoiem : TRegistroiem;
    arq : Text;
    nome : String;
    c : Array[0..255] of Char;
    i,j : Integer;
Begin
    If ((numion > 0) and (controle > 0)) or temcalibracao or temtim or temIEM
then
    CMFileOpen(Msg) else EXIT;
    If not ok_arquivo then EXIT;
    nome := StrPas(FileName);
    Assign(arq,nome);
    {$I-} Reset(arq); {$I+}
    If IOResult = 0 then
        begin
            StrPCopy(c,'Arquivo ' + nome + ' já existe. Sobrepor?');
            If MessageBox(HWindow,c,'Salvar', mb_defbutton2 or
                mb_YesNo or mb_IconQuestion) = IDNo then EXIT;
        end;
    Rewrite(arq);
    If temtim then With GraficoTIM^ do
        begin
            WriteLn(arq,'m/z inicial: ',GraficoEspectro^.mzi,#9,
                'm/z final: ',GraficoEspectro^.mzf);
            WriteLn(arq,'tempo de integração: ',GraficoEspectro^.tempointeg:0:1,
                'ms');
            WriteLn(arq,'Intervalo de aquisição: ',GraficoEspectro^.intaquis,' ms');

```

```

Writeln(arq,'energia de ionização: ',GraficoEspectro^.enion:0:1,' eV');
Writeln(arq,'comentario: ',GraficoEspectro^.comentario);
Writeln(arq,'t (ms)',#9,'abundância absoluta');
For i := 1 to GraficoEspectro^.ndados do
    Writeln(arq,GraficoEspectro^.Espectro^[i].tempo,#9,
        GraficoEspectro^.Espectro^[i].tim);
Writeln(arq);
Assign(ftim,pathdados + '\dados.tim');
Reset(ftim);
Repeat
    Read(ftim,blocotim);
    Writeln(arq,blocotim.tempo);
    For i := GraficoEspectro^.mzi to GraficoEspectro^.mzf do

Writeln(arq,i,#9,blocotim.abundancia[i],#9,sqrt(Round(blocotim.varianca[i]))
);
    Writeln(arq);
    Until EOF(ftim);
    Close(ftim);
end;
If temIEM then With GraficoIEM^ do
begin
    Writeln(arq,'m/z inicial: ',GraficoEspectro^.mzi,#9,
        'm/z final: ',GraficoEspectro^.mzf);
    Writeln(arq,'tempo de integração: ',GraficoEspectro^.tempointeg:0:1,'
ms');
    Writeln(arq,'Energia de ionização inicial: ',GraficoEspectro^.Xmin:5,#9,
        'Energia de ionização final: ',GraficoEspectro^.Xmax:5);
    Writeln(arq,'Incremento de EI: ',GraficoEspectro^.intervenion:0:1,' eV');
    Writeln(arq,'comentario: ',GraficoEspectro^.comentario);
    Writeln(arq,'EI (eV)',#9,'TIC');
    For i := 1 to GraficoEspectro^.ndados do
        Writeln(arq,GraficoEspectro^.Espectro^[i].enion:0:1,#9,
            GraficoEspectro^.Espectro^[i].iem);
    Writeln(arq);
    Assign(fiem,pathdados + '\dados.iem');
    Reset(fiem);
    Repeat
        Read(fiem,blocoiem);
        Writeln(arq,'Energia de ionização: ',blocoiem.enion:0:1);
        Writeln(arq,'m/z',#9,'Int.abs.',#9,'desvio padrão');
        For i := GraficoEspectro^.mzi to GraficoEspectro^.mzf do

Writeln(arq,i,#9,blocoiem.abundancia[i],#9,Round(sqrt(blocoiem.varianca[i]
)));
        Writeln(arq);
        Until EOF(fiem);
        Close(fiem);
    end;
    If temcalibracao then With DadosCalibracao^ do
begin
    Write(arq,'m/z"s: ');
    For i := 1 to fimz do Write(arq,masa[i],', ');
    Writeln(arq);
    Writeln(arq,'energia de ionização: ',enerion:0:1,' eV');
    Writeln(arq,'comentario: ',comentario);
    Writeln(arq,'número de leituras por ponto: ',nleituras);
    For j := 1 to fimz do
begin
        Writeln(arq,'m/z: ',masa[j]);
        Writeln(arq,'conc.',#9,'abundância absoluta',#9,'desvio padrão');

```

```

For i := 1 to fim do
    Writeln(arq,conc[i]:13,#9,abunda[i,j].abundancia,#9,
        Round(Sqrt(abunda[i,j].varianca)));
end;
If ndesconhecido > 0 then
For j := 1 to fimz do
begin
    Writeln(arq,'m/z: ',masa[j]);
    For i := 1 to ndesconhecido do
        Writeln(arq,'Desconhecido ',i,' ',desconhecido[i,j].abundancia,#9,
            Round(Sqrt(desconhecido[i,j].varianca)));
    end;
end;
For i := 1 to numion do
begin
    If GraficoSIM[i]^indice = i then With GraficoSIM[i]^ do
begin
        Writeln(arq,'m/z: ',GraficoEspectro^.mz);
        Writeln(arq,'tempo de integração: ',GraficoEspectro^.tempointeg:0:1,'
ms');
        Writeln(arq,'Intervalo de aquisição: ',GraficoEspectro^.intaquis,' ms');
        Writeln(arq,'energia de ionização: ',GraficoEspectro^.enion:0:2,' eV');
        Writeln(arq,'comentario: ',GraficoEspectro^.comentario);
        Writeln(arq,'t (s)',#9,'abundância absoluta',#9,'desvio padrão');
        For j := 1 to GraficoEspectro^.ndados do
            Writeln(arq,GraficoEspectro^.Espectro^[j].tempo/1000:0:2,#9,
                GraficoEspectro^.Espectro^[j].abundancia,#9,
                Round(Sqrt(GraficoEspectro^.Espectro^[j].varianca)));
        Writeln(arq,' ');
    end;
end;
Close(arq);
End;

Procedure TPrincipal.Print(var Msg : TMessage);
var Pinto : TPaintStruct;
    imp : String;
    NewFont, OldFont : HFont;
    OldPen : HPen;
    i, fonte : Byte;
    x, y, incr : Integer;
    s : String[20];
    c : Array[0..20] of Char;
    t : Array[0..60] of Char;

Procedure ImprimeSIM(var p : PSIM);
var j : Byte;
Begin
    If PmPartida('Dados - SIM',imp) then
begin
    x := p^.GraficoEspectro^.tx;
    y := p^.GraficoEspectro^.ty;
    If (imp[1] = 'H') and (imp[2] = 'P') then { Verifica se é HP. }
begin
        fonte := 40;
        p^.GraficoEspectro^.tx := 2000; { Valores para HP. }
        p^.GraficoEspectro^.ty := 1000;
    end
    else
begin
        fonte := 20;

```

```

    p^ GraficoEspectro^ tx := 500; { Valores para matriciais. }
    p^ GraficoEspectro^ ty := 200;
end;
NewFont := CreateFont(fonte,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
    VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
OldFont := SelectObject(PDc,NewFont);
SetTextAlign(PDc,TA_Right);
With p^ do With GraficoEspectro^ do
begin
    TextOut(PDc,tx,ty+90,FileName,StrLen(FileName));
    Linha(PDc,0,0,Xmax,0,cor);
    SetTextAlign(PDc,TA_Left);
    TextOut(PDc,0,ty+90,titulo,StrLen(titulo));
    StrPCopy(t,comentario);
    TextOut(PDc,0,ty+150,t,StrLen(t));
    incr := tx div 10;
    For j := 0 to 10 do
        begin
            MoveTo(PDc,j*incr,ty+10);
            LineTo(PDc,j*incr,ty);
            Str(j * Xmax / 10000.0 :0:0,s);
            StrPCopy(c,s);
            TextOut(PDc,j*incr,ty+30,c,StrLen(c));
        end;
    end;
p^.Paint(PDc,PlInfo);
SelectObject(PDc,OldFont);
DeleteObject(NewFont);
NovaPagina;
PrnFim;
p^ GraficoEspectro^ tx := x;
p^ GraficoEspectro^ ty := y;
end;
End;

```

```

Procedure ImprimeTIM(var p : PTIM);
var j : Byte;
Begin
    If PrnPartida('Dados - TIM',imp) then
        begin
            x := p^ GraficoEspectro^ tx;
            y := p^ GraficoEspectro^ ty;
            If (imp[1] = 'H') and (imp[2] = 'P') then { Verifica se é HP. }
                begin
                    fonte := 40;
                    p^ GraficoEspectro^ tx := 2000; { Valores para HP. }
                    p^ GraficoEspectro^ ty := 1000;
                end
            else
                begin
                    fonte := 20;
                    p^ GraficoEspectro^ tx := 500; { Valores para matriciais. }
                    p^ GraficoEspectro^ ty := 200;
                end;
            NewFont := CreateFont(fonte,0,0,0,FW_BOLD,0,0,0,ANSI_CHARSET,
OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
    VARIABLE_PITCH OR FF_ROMAN,'Tms Rmn');
            OldFont := SelectObject(PDc,NewFont);

```

```

SetTextAlign(PDc,TA_Right);
With p^ do With GraficoEspectro^ do
begin
    TextOut(PDc,tx,ty+90,FileName,StrLen(FileName));
    Linha(PDc,0,0,Xmax,0,cor);
    SetTextAlign(PDc,TA_Left);
    StrPCopy(t,comentario);
    TextOut(PDc,0,ty+150,t,StrLen(t));
    incr := tx div 10;
    For j := 0 to 10 do
        begin
            MoveTo(PDc,j*incr,ty+10);
            LineTo(PDc,j*incr,ty);
            Str(j * Xmax / 10000.0 :0:0,s); { Transforma em segundos. }
            StrPCopy(c,s);
            TextOut(PDc,j*incr,ty+30,c,StrLen(c));
        end;
    end;
p^.Paint(PDc,PlInfo);
SelectObject(PDc,OldFont);
DeleteObject(NewFont);
NovaPagina;
PrnFim;
p^ GraficoEspectro^ tx := x;
p^ GraficoEspectro^ ty := y;
end;
End;

```

```

Begin { Print }
If (MessageBox(HWindow,'Imprime o(s) grafico(s)?','Imprimir',
    mb_YesNo or mb_IconQuestion) = IDYes) then
    If temtim then ImprimeTIM(GraficoTIM)
    else For i := 1 to numion do If GraficoSIM[i]^ indice = i then
        ImprimeSIM(GraficoSIM[i]);
End;

```

```

Procedure TPrincipal.Sair(var Msg: TMessage);
var
    i : Byte;
Begin
    If temTIM then
        begin
            Dispose(DadosTIM,Done);
            Dispose(GraficoTIM,Done);
        end;
    If temcalibracao then Dispose(DadosCalibracao,Done);
    For i := 1 to numion do
        begin
            If DadosSIM[i]^ indice = i then
                begin
                    Dispose(DadosSIM[i],Done);
                    If controle > 0 then
                        If GraficoSIM[i]^ indice = i then Dispose(GraficoSIM[i],Done);
                end;
            end;
        end;
    CloseWindow;
End;

```

```

Procedure TPrincipal.WMTimer(var Msg: TMessage);

```

```

Procedure Temporizador_Filhas(PJanelaFilha: PSIMTIMBasico); Far;

```



```
Begin
    PJanelaFilha^.Temporizador;
End;

Begin {TPPrincipal.WMTimer}
    ForEach(@Temporizador_Filhas);
End;

Procedure TPPrincipal.WMDestroy(var Msg: TMessage);
Begin
    KillTimer(HWindow, 0);
    TMDIWindow.WMDestroy(Msg);
End;

Procedure TPPrincipal.Sobre(var Msg : TMessage);
Begin
    MessageBox(GetFocus,'Desenvolvido por Valmir F. Juliano - IQ -
UNICAMP - 1995',
    'Programa SIMTIM versão 1.0',mb_OK or mb_IconInformation);
End;

{ -----TSIMTIMAplicacao----- }
Type
    TSIMTIMAplicacao = Object(TApplication)

    Procedure InitMainWindow; virtual;
End;

Procedure TSIMTIMAplicacao.InitMainWindow;
Begin
    MainWindow := New(PPPrincipal,
        Init('S I M T I M', LoadMenu(HInstance,MakeIntResource(MenuID))));
End;

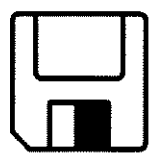
var
    SIMTIMAplicacao: TSIMTIMAplicacao;

BEGIN
    numion := 0;
    controle := 0;
    numgraficosim := 0;
    temcalibracao := FALSE;
    temTIM := FALSE;
    esta_adquirindo := FALSE;
    StrPCopy(WinDir,pathdados);
    SetCurDir(WinDir);
    SIMTIMAplicacao.Init('SIMTIM');
    SIMTIMAplicacao.Run;
    SIMTIMAplicacao.Done;
END.
```

APÊNDICE 2
LISTAGEM DAS *UNITS*



PENTAQUI



```
Unit PentAqui;
{-----}
Esta unidade contém todas as rotinas de acesso da interface AT-MIO-16X
da NATIONAL e da placa adicional para aquisição de dados do
PENTAQUADRUPOLO com os programas PENTA2D e PENTA3D.
{-----}

Interface

Uses WinTypes;

Const
    max = 1024;    { Número máximo de dados adquiridos para a média. }
    on = 1;
    off = 0;

Type
    reg = Record
        v : Array[1..3] of Word;
    end;
    buff = Array[1..max] of reg;
    Plonizador = ^Tlonizador;
    Tlonizador = Object
        Constructor Init(p,c,g,po : Integer);
        Destructor Done;
        Function Multiplex(l : Byte) : Double;
        Procedure El(e : Double);
        Private
            placa, canal, ganho, porta : Integer;
    end; { Tlonizador }
    PDetector = ^TDetector;
    TDetector = Object
        tempointeg : Word;
        Constructor Init(nro_da_placa,canal_ME,g : Integer; t : Double);
        Destructor Done;
        Procedure SetGanho(novoganho : Integer);
        Function LeAD : Word;
        Function LeADAvg : Word;
        Function JaLeu : Boolean;
        Private
            placa, canal, ganho : Integer;
            nb : Word;
            buf : ^buff;
    end;
    PQuadrupolo = ^TQuadrupolo;
    TQuadrupolo = Object
        Constructor Init(nro_da_placa,canal_DA,bpdo : Integer; nome : String);
        Destructor Done;
        Procedure DAC725(canal_DA,valor : Integer);
        Procedure DAC(nro_da_placa,canal_DA,valor : Integer);
        Procedure SetMassa(massa : Double);
        Procedure SetPoleDC(onoff : Byte);
        Private
            placa, canal, BitPoleDC : Integer;
            incl, interc : Double;
    end;

Implementation

{$I wdaq_tp.inc}
```

```
Constructor Tlonizador.Init(p,c,g,po : Integer);
Begin
    placa := p;
    canal := c;
    ganho := g;
    porta := po;
End;

Destructor Tlonizador.Done;
Begin
    El(70.0);
End;

Function Tlonizador.Multiplex(l : Byte) : Double;
const offset : Array[1..6] of Single = (198.4,198.7,198.9,198.4,198.6,198.3);
var tensao : Double;
    valor : Integer;
Begin
    Case l of
        0 : valor := 5;    { Multiplicador de Elétrons. }
        1 : valor := 2;    { Bloco da Fonte. }
        2 : valor := 9;    { Repeller. }
        3 : valor := 12;   { Lente 1. }
        4 : valor := 13;   { Lente 2. }
        5 : valor := 10;   { Q1 - Lente de Entrada. }
        6 : valor := 8;    { Q5 - Lente de Saída. }
        else valor := 0;
    end; { case }
    Dig_Out_Port(placa,porta,valor);    { Seleciona o elemento a ser
monitorado. }
    For valor := -32000 to 32000 do;    { Aguarda estabilização do sinal. }
        Al_VRead(placa,canal,ganho,tensao); { Le o sinal do Analog Multiplexer. }
        If l = 0 then Multiplex := 500.0 * tensao - 105.0
        else Multiplex := 40.0 * tensao - offset[l];
    End;

Procedure Tlonizador.El(e : Double);
var
    v : Integer;
    i : Byte;
Begin
    v := Round((e - Multiplex(1)) / 0.00304); { Ajusta a energia dos elétrons. }
    Port[$30E] := Hi(v);
    For i := 1 to 2 do;
        Port[$30D] := Lo(v);
    For i := 1 to 2 do;
        Port[$30B] := 0;
    End;

Constructor TDetector.Init(nro_da_placa,canal_ME,g : Integer; t : Double);
Begin
    New(buf);
    ganho := g;    { 1, 2, 5, (10), 20, 50, 100 } { Sens. média e Atenuação 10 }
    placa := nro_da_placa;
    canal := canal_ME;
    DAQ_Clear(placa);    { Inicializa o circuito de aquisição da placa. }
    CTR_Config(1,1,0,0,0,0);    { Configura o contador 1 da placa 1. }
    tempointeg := Round(t*1000); { Tempo em microsegundos. }
End;

Destructor TDetector.Done;
```

```
Begin
    DAQ_Clear(placa);
    CTR_Reset(1,1,0); { "Reseta" o contador 1. }
    Dispose(buf);
End;

Procedure TDetector.SetGanho(novoganho : Integer);
Begin
    ganho := novoganho;
End;

Function TDetector.LeAD : Word;
var valor : Integer;
Begin
    AI_Read(placa,canal,ganho,valor); { Le um único dado. }
    LeAD := Word(valor);
End;

Function TDetector.JaLeu : Boolean;
var ovf : Integer;
    cont : Word;
Begin
    CTR_EvRead(1,1,ovf,cont); { Le a contagem. }
    JaLeu := (cont >= tempointeg) or (ovf = 1);
End;

Function TDetector.LeADAvg : Word;
var
    i, med : Word;
    soma : Longint;
Begin
    CTR_EvCount(1,1,1,0); { Inicia o contador 1 em 1MHz e para com overflow. }
    i := 0;
    Repeat
        inc(i);
        buf^i.v[1] := LeAD;
        buf^i.v[2] := LeAD;
        buf^i.v[3] := LeAD;
    Until JaLeu;
    nb := i;
    soma := 0;
    For i := 1 to nb do
        begin
            With buf^i do
                begin { Encontra a mediana dos 3 valores lidos em cada posição de buf. }
                    med := v[1];
                    If (v[2] > v[1]) and (v[2] < v[3]) or (v[2] < v[1]) and (v[2] > v[3])
                    then med := v[2] else
                        If (v[3] > v[1]) and (v[3] < v[2]) or (v[3] < v[1]) and (v[3] > v[2])
                        then med := v[3];
                    end;
                    Inc(soma,med);
                end;
            end;
    LeADAvg := soma div nb; { Faz a média dos valores lidos. }
End;

Constructor TQuadrupolo.Init(nro_da_placa,canal_DA,bpdc : Integer; nome : String);
var arq : Text;
```

```
Begin
    Assign(arq,nome); { Le os dados da calibração. }
    ReSet(arq);
    Readln(arq,incl);
    Readln(arq,interc);
    Close(arq);
    placa := nro_da_placa;
    canal := canal_DA;
    BitPoleDC := bpdc; { Bit correspondente ao quad. na porta digital. }
    DAC(placa,canal,10000); { Inicializa a saída do D/A com aprox. m/z 150 }
    SetPoleDC(on); { Ajusta os quadrupolos com POLE DC em on. }
End;

Destructor TQuadrupolo.Done;
Begin
    DAC(placa,canal,10000); { Finaliza com aproximadamente m/z 150. }
    SetPoleDC(on); { Deixa os quadrupolos com POLE DC em on. }
End;

Procedure TQuadrupolo.DAC725(canal_DA,valor : Integer);
var i : Byte;
Begin
    Case canal_DA of
        0 : begin
            Port[$306] := Hi(valor); { Byte mais significativo. }
            For i := 1 to 2 do;
                Port[$305] := Lo(valor); { Byte menos significativo. }
                For i := 1 to 2 do;
                    Port[$303] := 0; { Habilita o latch do DA. }
                end;
            end;
        1 : begin
            Port[$30E] := Hi(valor); { Byte mais significativo. }
            For i := 1 to 2 do;
                Port[$30D] := Lo(valor); { Byte menos significativo. }
                For i := 1 to 2 do;
                    Port[$30B] := 0; { Habilita o latch do DA. }
                end;
            end;
    end; { case }
    For i := 1 to 20 do; { Tempo de acomodação do sinal. }
    End;

Procedure TQuadrupolo.DAC(nro_da_placa,canal_DA,valor : Integer);
Begin
    Case nro_da_placa of
        1 : AO_Write(nro_da_placa,canal_DA,valor);
        2 : DAC725(canal_DA,valor);
    end; { case }
End;

Procedure TQuadrupolo.SetMassa(massa : Double);
var valor : Integer;
Begin
    valor := Integer(Round(massa * incl + interc));
    DAC(placa,canal,valor);
    For valor := 1 to 10 do; { Cada salto de massa unitária não deve ser feito }
    End; { em um tempo menor que 1 ms. }

Procedure TQuadrupolo.SetPoleDC(onoff : Byte);
Begin
```

Case onoff of

0 : Dig_Out_Line(1,0,BitPoleDC,0); { placa=1, port=0, bit=bitpoledc, off }

1 : Dig_Out_Line(1,0,BitPoleDC,1); { placa=1, port=0, bit=bitpoledc, on }

end;

End;

BEGIN

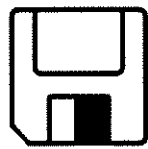
Dig_Prt_Config(1,0,0,1); { Configura as porta 0 e 1 para saída na placa 1.

}

Dig_Prt_Config(1,1,0,1);

END.

UPENTA4D



```
Unit UPenta4D;
{-----}
Esta unidade contém todas as rotinas de acesso da interface AT-MIO-16X
da NATIONAL e da placa adicional para aquisição de dados do
PENTAQUDRUPOLO com os programas PENTA4D e SIMTIM.
{-----}
```

interface

```
Uses WinTypes;
Const
T60 = 1000.0 / 60.0; { Período do sinal da rede em ms. }
maxmedia = 34; { Número máximo de aquisições do ME para média }
ganhos : Array[1..7] of Integer = (1,2,5,10,20,50,100); { Ganhos
disponíveis. }
```

Type

```
reg = Record
v : Array[1..3] of Word;
end;
Pbuf = ^Tbuf;
Tbuf = Array[1..1333] of reg;
Tionizador = Object
Constructor Init(p,c,g,po : Integer);
Destructor Done;
Function Multiplex(l : Byte) : Double;
Procedure El(e : Double);
Private
placa, canal, ganho, porta : Integer;
end; { Tionizador }
TDetector = Object
buf : Pbuf;
ganho : Integer;
saturado : Boolean;
Constructor Init(nro_da_placa,canal_ME,g : Integer);
Destructor Done;
Procedure SetTempointeg(t : Double);
Function GetTempointeg : Double;
Function LeAD : Word;
Function LeADAvg : Word;
Private
placa, canal : Integer;
tempointeg : Double;
nb : Word;
end; { TDetector }
TQuad = Object
Constructor Init(nro_da_placa,canal_DA,ncont, bpdc : Integer; arqcal :
String);
Destructor Done;
Procedure DAC725(canal_DA, valor : Integer);
Procedure DAC(nro_da_placa,canal_DA, valor : Integer);
Procedure SetMassa(massa : Double);
Function GetMassa : Double;
Function JaEstavel : Boolean;
Procedure SetPoleDC(on : Boolean);
Private
settime : Word;
placa, canal, BitPoleDC, nro_cont : Integer;
incl, interc, mzatual : Double;
end; { TQuad }
```

```
Quadrupolos = (um, tres, cinco);
PPentaQuad = ^TPentaQuad;
TPentaQuad = Object
quad : Array[um..cinco] of TQuad;
det : TDetector;
ionizador : Tionizador;
poleDC : Array[um..cinco] of Boolean;
lbmedia, lbvar : Array[1..7] of Double;
ganho : Integer;
saturado : Boolean;
Constructor Init(path,arqcal1,arqcal3,arqcal5 : String);
Procedure SetPoleDC(q : Quadrupolos; liga : Boolean);
Procedure SetMassa(q : Quadrupolos; m : Double);
Function GetMassa(q : Quadrupolos) : Double;
Function Otimize(q : Quadrupolos; m0, dm : Double) : Double;
Function LeME : Word;
Procedure LeMedia(n : Word; var media, varianca : Double);
Function LeMediaSimples(n : Word) : LongInt;
Function LinhaBase : Boolean;
Destructor Done;
end; { TPentaQuad }
```

implementation

```
{ $I wdaq_tp.inc }

Constructor Tionizador.Init(p,c,g,po : Integer);
Begin
placa := p;
canal := c;
ganho := g;
porta := po;
End;

Destructor Tionizador.Done;
Begin
El(70.0);
End;

Function Tionizador.Multiplex(l : Byte) : Double;
const offset : Array[1..6] of Double = (198.4,198.7,198.9,198.4,198.6,198.3);
var tensao : Double;
valor : Integer;
Begin
Case l of
0 : valor := 5; { Multiplicador de Elétrons. }
1 : valor := 2; { Bloco da Fonte. }
2 : valor := 9; { Repeller. }
3 : valor := 12; { Lente 1. }
4 : valor := 13; { Lente 2. }
5 : valor := 10; { Q1 - Lente de Entrada. }
6 : valor := 8; { Q5 - Lente de Saída. }
else valor := 0;
end; { case }
Dig_Out_Port(placa,porta,valor); { Seleciona o elemento a ser
monitorado. }
For valor := -32000 to 32000 do; { Aguarda estabilização do sinal. }
AL_VRead(placa,canal,ganho,tensao); { Le o sinal do Analog Multiplexer. }
If l = 0 then Multiplex := 500.0 * tensao - 105.0
else Multiplex := 40.0 * tensao - offset[l];
End;
```



```
Procedure TIonizador.El(e : Double);
var
  v : Integer;
  i : Byte;
Begin
  v := Round((e - Multiplex{1}) / 0.00305); { Ajusta a energia dos elétrons. }
  Port[$30E] := Hi(v);
  For i := 1 to 2 do;
  Port[$30D] := Lo(v);
  For i := 1 to 2 do;
  Port[$30B] := 0;
End;

Constructor TDetector.Init(nro_da_placa,canal_ME,g : Integer);
Begin
  ganho := g; { 1, 2, 5, (10) , 20, 50, 100 } { Sens. media e Aten.= 10 }
  placa := nro_da_placa;
  canal := canal_ME;
  SetTempointeg(T60);
  DAQ_Clear(placa); { Inicializa o circuito de aquisição da placa. }
  New(buf);
  saturado := FALSE;
End;

Destructor TDetector.Done;
Begin
  DAQ_Clear(placa);
  Dispose(buf);
End;

Procedure TDetector.SetTempointeg(t : Double);
Begin
  tempointeg := 1000.0 * t; { Tempo em microssegundos. }
End;

Function TDetector.GetTempointeg : Double;
Begin
  GetTempointeg := tempointeg;
End;

Function TDetector.LeAD : Word;
var valor : Integer;
Begin
  Al_Read(placa,canal,ganho,valor); { Le um único dado. }
  LeAD := Word(valor);
End;

Function TDetector.LeADAvg : Word;

Function JaLeu : Boolean;
var ovf : Integer;
  cont : Word;
Begin
  CTR_EvRead(1,1,ovf,cont); { Le a contagem e retorna "true" }
  JaLeu := (cont >= tempointeg) or (ovf = 1); { se já excedeu o tempointeg. }
End;

var
  i, med : Word;
  soma : LongInt;
```

```
Begin
  saturado := FALSE;
  CTR_EvCount(1,1,1,0); { Inicia o contador 1 em 1MHz e para com overflow. }
  i := 0;
  Repeat
    inc(i);
    buf^[i].v[1] := LeAD;
    buf^[i].v[2] := LeAD;
    buf^[i].v[3] := LeAD;
  Until JaLeu;
  nb := i;
  soma := 0;
  For i := 1 to nb do
    begin
      With buf^[i] do
        begin { Encontra a mediana dos 3 valores lidos em cada posição de buf. }
          med := v[1];
          If (v[2] > v[1]) and (v[2] < v[3]) or (v[2] < v[1]) and (v[2] > v[3])
          then med := v[2] else
          If (v[3] > v[1]) and (v[3] < v[2]) or (v[3] < v[1]) and (v[3] > v[2])
          then med := v[3];
        end;
        If med >= 65530 then saturado := TRUE;
        Inc(soma,med);
      end;
    LeADAvg := soma div nb; { Faz a média dos valores lidos. }
  End;

Constructor TQuad.Init(nro_da_placa,canal_DA,ncont,bpdc : Integer; arqcal : String);
var arq : Text;
Begin
  Assign(arq,arqcal); { Le os dados da calibração. }
  ReSet(arq);
  Readln(arq,incl);
  Readln(arq,interc);
  Close(arq);
  placa := nro_da_placa;
  canal := canal_DA;
  nro_cont := ncont;
  BitPoleDC := bpdc; { Bit correspondente ao quad. na porta digital. }
  CTR_Config(1,nro_cont,0,0,0); { Configura o contador nro_cont da placa 1. }
  mztatual := 1000.0; { Só para garantir ... }
  SetMassa(30.0); { Inicializa a saída do D/A com m/z=30. }
  SetPoleDC(FALSE); { Ajusta os quadrupolos com POLE DC em off. }
End;

Destructor TQuad.Done;
Begin
  SetMassa(30.0); { Finaliza com m/z 30. }
  SetPoleDC(FALSE); { Deixa os quadrupolos com POLE DC em off. }
  CTR_Reset(1,nro_cont,0); { "Reseta" o contador. }
End;

Procedure TQuad.DAC725(canal_DA,valor : Integer);
var i : Byte;
Begin
```

```

If canal_DA = 0 then
begin
Port[$306] := Hi(valor); { Byte mais significativo. }
Port[$305] := Lo(valor); { Byte menos significativo. }
Port[$303] := 0;      { Habilita o latch do DA. }
end
else
begin
Port[$30E] := Hi(valor); { Byte mais significativo. }
Port[$30D] := Lo(valor); { Byte menos significativo. }
Port[$30B] := 0;      { Habilita o latch do DA. }
end;
End;

Procedure TQuad.DAC(nro_da_placa,canal_DA,valor : Integer);
Begin
If nro_da_placa = 1 then AO_Write(nro_da_placa,canal_DA,valor)
else DAC725(canal_DA,valor);
End;

Procedure TQuad.SetMassa(massa : Double);
var valor : Integer;
salto : Word;
Begin      { Converte massa em valor de 16 bits para o DA. }
valor := Integer(Round(massa * incl + interc));
DAC(placa,canal,valor);
CTR_EvCount(1,nro_cont,1,0); { Contador em 1MHz e parada com
overflow. }
salto := Round(10.0*Abs(massa - mzatual));
settime := Word(13188 * LongInt(salto) div (160 + salto)); { tempo em us. }
mzatual := massa;
End;

Function TQuad.GetMassa : Double;
Begin
GetMassa := mzatual;
End;

Function TQuad.JaEstavel : Boolean;
var ovf : Integer;
cont : Word;
Begin
CTR_EvRead(1,nro_cont,ovf,cont);
JaEstavel := (cont >= settime) or (ovf = 1);
End;

Procedure TQuad.SetPoleDC(on : Boolean);
Begin
If on then Dig_Out_Line(1,0,BitPoleDC,1) { placa=1, port=0, bit=bitpoledc,
on }
else Dig_Out_Line(1,0,BitPoleDC,0); { placa=1, port=0, bit=bitpoledc,
off }
End;

Constructor TPentaQuad.Init(path,arqcal1,arqcal3,arqcal5 : String);
var i : Quadrupolos;
Begin
quad[um].Init(1,0,1,0,path+arqcal1); { placa, canal_DA, contador e
bitpoleDC. }
quad[tres].Init(2,0,2,1,path+arqcal3);
quad[cinco].Init(1,1,5,2,path+arqcal5);

```

```

For i := um to cinco do poleDC[i] := FALSE;
det.Init(1,0,50); { placa, canal e ganho. }
ionizador.Init(1,1,1,1); { placa, canal, ganho, porta. }
saturado := FALSE;
End;

Procedure TPentaQuad.SetPoleDC(q : Quadrupolos; liga : Boolean);
Begin
poleDC[q] := liga;
quad[q].SetPoleDC(poleDC[q]);
End;

Procedure TPentaQuad.SetMassa(q : Quadrupolos; m : Double);
var
qs : Quadrupolos;
mqoff : Double;
Begin
mqoff := 91.91 * m / (173.5 + m);
quad[q].SetMassa(m);
For qs := um to cinco do
If not poleDC[qs] then quad[qs].SetMassa(mqoff);
End;

Function TPentaQuad.GetMassa(q : Quadrupolos) : Double;
Begin
GetMassa := quad[q].GetMassa;
End;

Function TPentaQuad.Otimiza(q : Quadrupolos; m0, dm : Double) : Double;
var
l : Word;
m, mmax, incr, somai, somami : Double;
Begin
m := m0 - dm;
quad[q].SetMassa(m);
incr := 2.0 / 5.0 * dm; { 5 ptos. }
somai := 0.0;
somami := 0.0;
While (m <= m0 + dm) do
begin
quad[q].SetMassa(m);
l := det.LeADAvg;
somai := somai + l;
somami := somami + l * m;
m := m + incr;
end;
mmax := somami / somai; { Centróide do pico. }
quad[q].SetMassa(mmax);
otimiza := mmax;
End;

Function TPentaQuad.LeME : Word;
Begin
Repeat
Until quad[um].JaEstavel and quad[tres].JaEstavel and
quad[cinco].JaEstavel;
LeME := det.LeADAvg;
End;

Destructor TPentaQuad.Done;
Begin

```

```

quad[um].Done;
quad[tres].Done;
quad[cinco].Done;
det.Done;
ionizador.Done;
End;

```

```

Procedure TPentaQuad.LeMedia(n : Word; var media, varianca : Double);
var

```

```

i : Word;
v : Array[1..maxmedia] of Word;
soma, me, vari, fator : Double;
Begin
fator := 100.0 / det.ganho;
saturado := FALSE;
If n > maxmedia then n := maxmedia;
v[1] := LeME;
For i := 2 to n do
begin
v[i] := det.LeADAvg;
If det.saturado then saturado := TRUE;
end;
soma := v[1];
For i := 2 to n do soma := soma + v[i];
me := fator * soma / n;
soma := 0.0;
For i := 1 to n do soma := soma + sqrt(fator*v[i] - me);
vari := soma / (n * (n - 1)); { Variança da média. }
media := me - lbmedia[ganho];
varianca := vari + lbvar[ganho];
End;

```

```

Function TPentaQuad.LeMediaSimples(n : Word) : LongInt;
var

```

```

i : Byte;
j : Word;
fator : Double;
soma : LongInt;
Begin
i := 8;
Repeat
Dec(i);
det.ganho := ganhos[i];
fator := 100.0 / det.ganho;
soma := LeME;
Until (not det.saturado) or (i = 1);
For j := 1 to n - 1 do soma := soma + det.LeADAvg;
LeMediaSimples := Round(fator * soma / n);
End;

```

```

Function TPentaQuad.LinhaBase : Boolean;
var

```

```

pilhabool : Array[um..cinco] of Boolean;
q : Quadrupolos;
i : Integer;
Begin
For q := um to cinco do { Verifica o estado de cada quadrupolo. }
begin
pilhabool[q] := PoleDC[q];
SetPoleDC(q, TRUE);
end;

```

```

SetMassa(um, 10.0);

```

```

SetMassa(tres, 400.0);

```

```

SetMassa(cinco, 10.0);

```

```

For i := 1 to 7 do

```

```

begin

```

```

ganho := i; { Variável para apontar o ganho selecionado. }

```

```

det.ganho := ganhos[ganho];

```

```

lbmedia[ganho] := 0.0;

```

```

lbvar[ganho] := 0.0;

```

```

LeMedia(maxmedia, lbmedia[ganho], lbvar[ganho]);

```

```

end;

```

```

For q := um to cinco do SetPoleDC(q, pilhabool[q]); { Retorna ao estado inicial. }

```

```

LinhaBase := lbmedia[7] > 3.0 * sqrt(lbvar[7]); { Verifica lb no maior ganho. }

```

```

}

```

```

End;

```

```

BEGIN

```

```

Dig_Prt_Config(1, 0, 0, 1); { Configura as porta 0 e 1 para saída na placa 1. }

```

```

}

```

```

Dig_Prt_Config(1, 1, 0, 1);

```

```

END.

```