

DIRCEU SONO ALVES DE AZEVEDO

LUCIANO BARBOSA CHIECO

**SISTEMA DE ANÁLISE**  
**CINEMÁTICA DA MARCHA HUMANA**

Projeto de Formatura apresentado à  
Escola Politécnica da Universidade de São Paulo –  
Departamento de Engenharia Mecânica  
– Hab. em Automação e Sistemas

Orientador:

Prof. Dr. Raul Gonzalez

São Paulo

1998

10,0 (dez)  
hm

## **DEDICATÓRIA**

Aos meus familiares, amigos, mestres e colegas da Escola Politécnica que sempre incentivaram e apoiaram meu aperfeiçoamento técnico e profissional.

## AGRADECIMENTOS

Ao amigo e orientador Prof. Dr. Raul Gonzalez pelas diretrizes seguras e permanente incentivo.

Ao Dr. Marcelo Saad pelos conhecimentos técnicos passados e conselhos valiosos.

Aos mestres da Escola Politécnica sempre dispostos a auxiliar e transmitir instruções preciosas.

Aos amigos e colegas, fundamentais durante todo os anos da graduação, consistindo na principal fonte de ajuda e conselhos preciosos para toda a vida profissional e pessoal.

A todos que direta ou indiretamente colaboraram na execução deste trabalho.

# SUMARIO

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| DEDICATÓRIA.....                                                                              | 1  |
| AGRADECIMENTOS .....                                                                          | 2  |
| SUMARIO.....                                                                                  | 3  |
| LISTAS DE FIGURAS.....                                                                        | 6  |
| LISTAS DE TABELAS.....                                                                        | 7  |
| LISTA DE SÍMBOLOS.....                                                                        | 8  |
| 1. OBJETIVOS .....                                                                            | 10 |
| 2. MOTIVAÇÃO.....                                                                             | 11 |
| 3. FUNDAMENTOS TEÓRICOS .....                                                                 | 12 |
| 3.1. REPRESENTAÇÃO DE UMA IMAGEM DIGITAL .....                                                | 12 |
| 3.2. AMOSTRAGEM E QUANTIZAÇÃO.....                                                            | 12 |
| 3.3. FORMAÇÃO DA IMAGEM .....                                                                 | 13 |
| 3.4. MODELAGEM DA CÂMERA .....                                                                | 17 |
| 3.5. CALIBRAÇÃO DA CÂMERA .....                                                               | 20 |
| 3.6. ALGORITMOS DE SEGMENTAÇÃO DE IMAGENS.....                                                | 21 |
| 3.6.1. Algoritmo de Detecção de Descontinuidades .....                                        | 21 |
| 3.6.2. Algoritmo de Thresholding .....                                                        | 22 |
| 3.6.3. Algoritmo de Segmentação Orientada por Regiões.....                                    | 23 |
| 3.6.4. Algoritmos de Segmentação através do Movimento .....                                   | 24 |
| 3.7. ALGORITMOS DE DETECÇÃO DE PADRÕES.....                                                   | 25 |
| 3.7.1. Métodos em Domínio Espacial .....                                                      | 25 |
| 3.7.2. Método em Domínio de Frequências .....                                                 | 27 |
| 3.7.3. Geração de máscaras espaciais a partir de especificações no domínio de frequência..... | 27 |

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| 3.8. ALGORITMO DE FILTRAGEM DOS RESULTADOS .....                    | 29        |
| 3.9. MÉTODO NUMÉRICO DE NEWTON-RAPSHON .....                        | 34        |
| <b>4. ARQUITETURA DO SISTEMA.....</b>                               | <b>36</b> |
| 4.1. ESTRUTURA DO SISTEMA DE AQUISIÇÃO DE IMAGENS.....              | 38        |
| 4.2. MARCADORES .....                                               | 40        |
| 4.3. CALIBRAÇÃO DO SISTEMA .....                                    | 41        |
| 4.4. ESTRUTURA DO SISTEMA DE PROCESSAMENTO DE IMAGENS.....          | 43        |
| 4.5. CONVERSÃO DE ARQUIVOS AVI EM BMP .....                         | 44        |
| 4.6. ALGORITMO DE SEGMENTAÇÃO DE IMAGENS.....                       | 44        |
| 4.7. ALGORITMO DE IDENTIFICAÇÃO E RASTREAMENTO DOS MARCADORES ..... | 46        |
| 4.8. ALGORITMO DE FILTRAGEM DE RESULTADOS .....                     | 48        |
| 4.9. RECONSTRUÇÃO CINEMÁTICA DA TRAJETÓRIA .....                    | 50        |
| <b>5. PRECISÃO E ACURACIDADE DO SISTEMA .....</b>                   | <b>51</b> |
| <b>6. DISCUSSÃO.....</b>                                            | <b>52</b> |
| <b>7. CONCLUSÕES.....</b>                                           | <b>53</b> |
| <b>ANEXO I .....</b>                                                | <b>54</b> |
| MANIPULAÇÃO DE BITMAPS E PALHETAS DE CORES.....                     | 54        |
| <b>ANEXO II.....</b>                                                | <b>56</b> |
| TRANSFORMADA DE FOURIER .....                                       | 56        |
| <b>ANEXO III .....</b>                                              | <b>57</b> |
| CONVOLUÇÃO E CORRELAÇÃO.....                                        | 57        |
| a. Convolução .....                                                 | 57        |
| b. Correção .....                                                   | 58        |
| <b>ANEXO IV .....</b>                                               | <b>60</b> |
| <b>ANEXO V .....</b>                                                | <b>73</b> |

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| GRÁFICOS RESULTADOS DO PROCESSAMENTO .....                          | 73         |
| <b>ANEXO VI .....</b>                                               | <b>120</b> |
| CÓDIGO FONTE DAS PRINCIPAIS PROCEDURES DOS APLICATIVOS DELPHI ..... | 120        |
| <i>a. Conversor AVI para BMPs (Delphi 3)</i> .....                  | 120        |
| <i>b. Identificador de Padrões em BMPs</i> .....                    | 122        |
| <b>ANEXO VIII .....</b>                                             | <b>129</b> |
| TELAS DOS PROGRAMAS .....                                           | 129        |
| <i>a. Conversor AVI para BMPs</i> .....                             | 129        |
| <i>b. Identificador de Padrões em BMPs</i> .....                    | 129        |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>                             | <b>130</b> |

## LISTAS DE FIGURAS

|                                                                                              |     |
|----------------------------------------------------------------------------------------------|-----|
| FIGURA 1. MODELO BÁSICO DO PROCESSO DE FORMAÇÃO DE IMAGENS                                   | 13  |
| FIGURA 2. GEOMETRIA DE AQUISIÇÃO DE IMAGENS COM DOIS SISTEMAS DE COORDENADAS                 | 18  |
| FIGURA 3. HISTOGRAMAS DE NÍVEIS DE CINZA QUE PODEM SER PARTICIONADOS PELO<br>THRESHOLDING.   | 23  |
| FIGURA 4. VERSOR NORMAL                                                                      | 31  |
| FIGURA 5. ORIENTAÇÃO DO VERSOR NORMAL                                                        | 31  |
| FIGURA 6. FLUXOGRAMA DO SISTEMA                                                              | 36  |
| FIGURA 7. AQUISIÇÃO DAS IMAGENS DE CALIBRAÇÃO                                                | 42  |
| FIGURA 8. HISTOGRAMA CORRESPONDENTE A IMAGEM ORIGINAL                                        | 46  |
| FIGURA 9. IMAGENS ORIGINAL E PROCESSADA COM O ALGORITMO DE THRESHOLDING                      | 46  |
| FIGURA 10. MÁSCARA DE CONVOLUÇÃO ESPACIAL                                                    | 47  |
| FIGURA 11. ESQUEMA DO PROCESSO DE IDENTIFICAÇÃO E RASTREAMENTO DE MARCADORES                 | 48  |
| FIGURA 12. ILUSTRAÇÃO GRÁFICA DA CONVOLUÇÃO                                                  | 58  |
| FIGURA 13. ILUSTRAÇÃO GRÁFICA DA CORRELAÇÃO                                                  | 59  |
| FIGURA 14 - TELA DO PROGRAMA CONVERSOR AVI PARA BMPS                                         | 129 |
| FIGURA 15 - TELA DO IDENTIFICADOR DE PADRÕES EM BMPS (ANTES/DEPOIS DE PROCESSAR A<br>IMAGEM) | 129 |

## LISTAS DE TABELAS

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| TABELA 1. HEADER DO FORMATO DE ARQUIVO WINDOWS' BMP                                    | 54 |
| TABELA 2. CONSTANTES E RESTRIÇÕES GEOMÉTRICAS                                          | 60 |
| TABELA 3. COORDENADAS CARTESIANAS DOS MARCADORES (PIXELS) – VALORES ORIGINAIS          | 61 |
| TABELA 4. COORDENADAS CARTESIANAS DOS MARCADORES (MM) – VALORES ORIGINAIS              | 62 |
| TABELA 5. DIMENSÕES GEOMÉTRICAS – VALORES ORIGINAIS                                    | 63 |
| TABELA 6. VELOCIDADES LINEARES DOS MARCADORES – VALORES ORIGINAIS                      | 64 |
| TABELA 7. ACELERAÇÕES LINEARES DOS MARCADORES (MM/S <sup>2</sup> ) – VALORES ORIGINAIS | 65 |
| TABELA 8. DESLOCAMENTOS, VELOCIDADES E ACELERAÇÕES ANGULARES ORIGINAIS                 | 66 |
| TABELA 9. COORDENADAS CARTESIANAS DOS MARCADORES (PIXELS) – VALORES FILTRADOS          | 67 |
| TABELA 10. COORDENADAS CARTESIANAS DOS MARCADORES (MM) – VALORES FILTRADOS             | 68 |
| TABELA 11. DIMENSÕES GEOMÉTRICAS – VALORES FILTRADOS                                   | 69 |
| TABELA 12. VELOCIDADES LINEARES DOS MARCADORES – VALORES FILTRADOS                     | 70 |
| TABELA 13. VELOCIDADES LINEARES DOS MARCADORES – VALORES FILTRADOS                     | 71 |
| TABELA 14. DESLOCAMENTOS, VELOCIDADES E ACELERAÇÕES ANGULARES FILTRADOS                | 72 |

## LISTA DE SÍMBOLOS

|                |                                                                         |
|----------------|-------------------------------------------------------------------------|
| $\psi$         | : Restrição geométrica                                                  |
| $\lambda$      | : Distância focal da lente                                              |
| $\mathfrak{F}$ | : Transformada de Fourier                                               |
| $c$            | : Ponto no sistema de coordenadas da câmera cartesiana                  |
| $d_{ij}$       | : Imagem resultante da análise de segmentação através de movimento      |
| $e$            | : Função erro, diferença entre um valor correto e outro estimado        |
| $r$            | : Distância da câmera ao centro do plano da imagem                      |
| $w$            | : Ponto no sistema de coordenadas global cartesiano                     |
| $A$            | : Marcador fixo na articulação do quadril                               |
| $B$            | : Marcador fixo na articulação do joelho                                |
| $C$            | : Marcador fixo na articulação do tornozelo                             |
| $D$            | : Marcador fixo a ponta do pé                                           |
| $F$            | : Comprimento do fêmur                                                  |
| $G$            | : Matriz de translação para modelagem da câmera                         |
| $H$            | : Máscara de convolução espacial                                        |
| $I$            | : Intensidade de nível de pixel                                         |
| $J$            | : Índice de performance                                                 |
| $P$            | : Comprimento do pé                                                     |
| $P$            | : Matriz de transformação perspectiva                                   |
| $R$            | : Matriz de rotação para modelagem da câmera                            |
| $T$            | : Comprimento da tíbia                                                  |
| $T$            | : Nível de tolerância                                                   |
| $x_A$          | : Componente horizontal filtrada da coordenada cartesiana do marcador A |
| $y_A$          | : Componente vertical filtrada da coordenada cartesiana do marcador A   |
| $x_B$          | : Componente horizontal filtrada da coordenada cartesiana do marcador B |
| $y_B$          | : Componente vertical filtrada da coordenada cartesiana do marcador B   |

- $x_C$  : Componente horizontal filtrada da coordenada cartesiana do marcador C
- $y_C$  : Componente vertical filtrada da coordenada cartesiana do marcador C
- $x_D$  : Componente horizontal filtrada da coordenada cartesiana do marcador D
- $y_D$  : Componente vertical filtrada da coordenada cartesiana do marcador D
- $x_A^*$  : Componente horizontal capturada da coordenada cartesiana do marcador A
- $y_A^*$  : Componente vertical capturada da coordenada cartesiana do marcador A
- $x_B^*$  : Componente horizontal capturada da coordenada cartesiana do marcador B
- $y_B^*$  : Componente vertical capturada da coordenada cartesiana do marcador B
- $x_C^*$  : Componente horizontal capturada da coordenada cartesiana do marcador C
- $y_C^*$  : Componente vertical capturada da coordenada cartesiana do marcador C
- $x_D^*$  : Componente horizontal capturada da coordenada cartesiana do marcador D
- $y_D^*$  : Componente vertical capturada da coordenada cartesiana do marcador D
- CGx : Componente horizontal do centro de massa do marcador
- CGy : Componente vertical do centro de massa

## 1. OBJETIVOS

O objetivo deste projeto foi o desenvolvimento de sistema para análise cinemática do movimento da marcha humana, através descrição quantitativa de todas as grandezas de interesse à análise.

A marcha humana compreende uma seqüência de eventos rápidos e complexos. É difícil, à observação clínica, analisar estes fenômenos, e quantificar seu grau de afastamento da normalidade. Tais limitações levaram médicos, biólogos, engenheiros e estudiosos do movimento a desenvolver recursos de alta tecnologia e precisão para o registro e análise da marcha.

O projeto pode ser inicialmente dividido em duas grandes frentes. Uma primeira responsável fundamentalmente pelo processo de aquisição das imagens do movimento de interesse, o que engloba toda a especificação dos equipamentos e componentes de hardware. E uma segunda, envolvendo basicamente todas as etapas de processamento de imagens, através das quais o movimento seria identificado, rastreado ao longo do tempo e quantificado em todas as suas grandezas tempo-espaciais de interesse à análise, podendo-se citar o comprimento do passo, deslocamentos, velocidades e acelerações lineares e angulares das articulações da perna.

## 2. MOTIVAÇÃO

A análise de marcha tem papel fundamental no estudo e tratamento de patologias que envolvem o aparelho locomotor. Esta sistemática fornece dados que contribuem para a compreensão dos mecanismos causadores da alteração de marcha, a classificação da gravidade do acometimento, o direcionamento do tratamento, a prescrição e adequação de órteses, próteses e auxiliares de marcha, a comparação pré e pós-intervenção e a melhoria de desempenho para o caso de atletas de elite.

Como pode-se notar a análise de movimentos detém papel importante em diferentes áreas das ciências humanas e biomédicas. Na neurofisiologia, auxilia um melhor entendimento dos mecanismos básicos de controle dos movimentos; na ortopedia e reabilitação motora, contribui com diagnósticos funcionais mais detalhados e quantificados; em neurologia, proporciona a detecção de pequenos desvios, não evidentes através de simples inspeção visual, permitindo rápido diagnóstico de lesões cerebrais; e na medicina esportiva, a performance atlética pode ser aprimorada através da análise de movimentos elementares.

Em nosso país, considerando-se que ainda é incompleta a assistência primária em saúde, pode-se entender porque tão poucos serviços e profissionais de saúde se dediquem ao estudo da marcha. A montagem de um laboratório de análise de marcha completo e moderno pode incluir um gasto exorbitante, e estes custos estão fora do orçamento da maioria das instituições.

Por tais razões, muito esforço está sendo despendido no desenvolvimento de sistemas dedicados à análise cinemática da marcha humana. Mais recentemente a tecnologia baseada em aquisição do movimento através de câmeras de vídeo e processamento digital dessas informações vem recebendo grande destaque. Tal fato deve-se significativamente devido ao grande aprimoramento observado nesta área o que possibilita velocidade, precisão e custo de processamento atraentes à aplicação.

### 3. FUNDAMENTOS TEÓRICOS

Nesta parte discorre-se idéias, fundamentos, problemas, sugestões dos vários autores pertinentes e selecionados. O levantamento aqui apresentado serviu como base à execução do trabalho proposto.

Os conceitos dispostos nessa seção são frutos de profunda pesquisa e revisão de literaturas em várias fontes documentais disponíveis.

#### 3.1. REPRESENTAÇÃO DE UMA IMAGEM DIGITAL

Uma imagem monocromática é representada por uma função bidimensional de intensidade de luz,  $f(x, y)$ , onde  $x$  e  $y$  são as coordenadas espaciais e o valor da função  $f$  em cada ponto  $(x, y)$  proporcional à intensidade de nível de cinza da imagem naquele ponto.

Uma imagem digital é uma imagem  $f(x, y)$  com os valores das coordenadas espaciais e de intensidade de nível de cinza discretizadas. Uma possível e usual maneira de representação da imagem é a matricial, onde as linhas e colunas identificam as coordenadas espaciais e o valor das células, o nível de cinza no ponto. Nesta representação cada elemento da matriz é denominado "picture element", ou pixel.

#### 3.2. AMOSTRAGEM E QUANTIZAÇÃO

Como já mencionado, a imagem digital  $f(x, y)$  deve sofrer discretização das coordenadas espaciais e da amplitude. Este processo de digitalização da imagem influi diretamente no processamento de imagem.

Uma imagem contínua  $f(x, y)$  pode ser amostrada por uma matriz  $N \times N$ , onde cada elemento da matriz é um valor discreto. Este processo de digitalização da imagem demanda uma especificação de  $N$  e do número de discretizações de nível de cinza, o que implica diretamente na resolução e no número de bits necessários ao

armazenamento da imagem. É prática comum em processamento de imagens digitais deste valores serem números inteiros de potência de dois.

A determinação das constantes de amostragem deve considerar tanto a resolução, quão mais próximo a imagem digital está da original, e os requisitos de processamento de imagem. Porém uma tentativa de melhorar a resolução implica em maiores exigências para o processamento de imagem, e vice-versa. Desta forma verifica-se que uma definição de uma configuração ótima está intimamente relacionada à aplicação.

### 3.3. FORMAÇÃO DA IMAGEM

A transformação perspectiva projeta pontos tridimensionais em um plano, exercendo papel central na processamento de imagens, já que ela propicia uma aproximação no modo em que a imagem é formada captando informações tridimensionais.

Um modelo de formação de imagem é mostrado na figura 1. Define-se o sistema de coordenadas da câmera ( $x, y, z$ ) como tendo o plano de imagem coincidente com o plano  $xy$  e o eixo óptico, determinado pelo centro da lente, ao longo do eixo  $z$ . Então o centro do plano da imagem fica na origem e o centro da lente na coordenada  $(0, 0, \lambda)$ . Assume-se assim que o sistema de coordenadas da câmera está alinhado ao sistema de coordenadas global  $(X, Y, Z)$ .

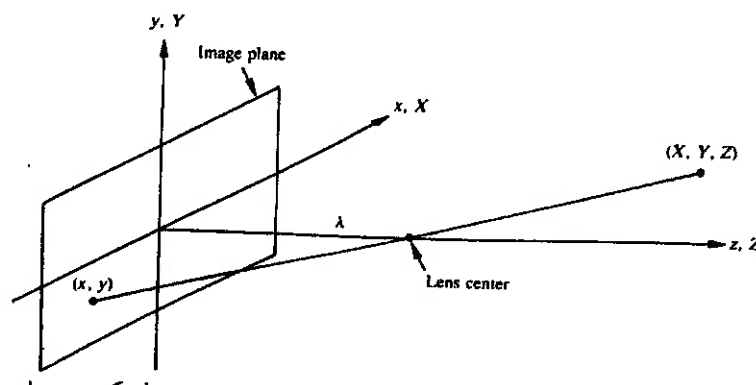


FIGURA 1. MODELO BÁSICO DO PROCESSO DE FORMAÇÃO DE IMAGENS

Se a câmera possui o foco para objetos distantes,  $\lambda$  é o comprimento focal da lente. Assume-se todos os pontos de interesse estão a frente da lente, isto é,  $Z > \lambda$ . Utilizando-se da semelhança de triângulos, obtém-se uma relação entre as coordenadas  $(x, y)$  da projeção do ponto  $(X, Y, Z)$  no plano da imagem.

$$x = \frac{\lambda X}{\lambda - Z} \quad y = \frac{\lambda Y}{\lambda - Z} \quad (3.3.1)$$

Nota-se que essas equações são não-lineares porque elas envolvem divisão pela variável  $Z$ . Pode-se rescrever essas equações segundo a representação matricial usando coordenadas homogêneas. As coordenadas homogêneas de um ponto escrito em coordenadas cartesianas  $(X, Y, Z)$  são definidas como  $(kX, kY, kZ, k)$ , onde  $k$  é uma constante arbitrária não nula. Um ponto no sistema de coordenadas global cartesiano pode ser expresso pelo vetor:

$$w = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.3.2)$$

e sua transformação homogênea é dada por

$$w_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \quad (3.3.3)$$

A matriz de transformação perspectiva é definida da seguinte forma:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \quad (3.3.4)$$

As coordenadas da câmera na forma homogênea é dada pelo produto  $P.w_h$ , resultando no vetor  $c_h$ .

$$c_h = P.w_h = \begin{bmatrix} kX \\ kY \\ kZ \\ \frac{-kZ}{\lambda} + k \end{bmatrix} \quad (3.3.5)$$

Essas coordenadas podem ser convertidas para a forma cartesiana dividindo cada termo pelo quarto termo de  $c_h$ . Portanto as coordenadas cartesianas de qualquer ponto no sistema de coordenadas da câmera são dados pelo vetor:

$$c = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix} \quad (3.3.6)$$

Os dois primeiros termos de  $c$  são as coordenadas  $(x, y)$  no plano de imagem do ponto  $(X, Y, Z)$  projetado. A terceira componente não é de interesse nos termos do modelo mostrado na figura 1.

A transformação perspectiva inversa mapeia um ponto na imagem em 3D.

$$w_h = P^{-1}c_h \quad (3.3.7)$$

onde  $P^{-1}$  é dado por:

$$P^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix} \quad (3.3.8)$$

Suponha que um dado ponto da imagem possua as coordenadas  $(x_0, y_0, 0)$ , para um plano de imagem em  $z=0$ . Este ponto pode ser expresso por um vetor homogêneo

$$c_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \quad (3.3.9)$$

Aplicando-se a equação (3.3.7), tem-se o vetor de coordenadas globais homogêneo

$$w_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \quad (3.3.10)$$

ou em coordenadas cartesianas,

$$w = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix} \quad (3.3.11)$$

Isto não é obviamente o esperado, uma vez que resulta  $Z=0$  para qualquer ponto 3D.

O ponto da imagem  $(x_0, y_0)$  corresponde a um conjunto de pontos 3D colineares que estão sobre a linha determinada pelos pontos  $(x_0, y_0, 0)$  e  $(0, 0, \lambda)$ . As equações desta linha no sistema de coordenadas globais são obtidas das equações (3.3.1),

$$X = \frac{x_0}{\lambda}(\lambda - Z) \quad Y = \frac{y_0}{\lambda}(\lambda - Z)$$

Estas equações mostram que a menos que se saiba algo sobre o ponto 3D que produziu a imagem, não se pode recuperar totalmente o ponto 3D da imagem. Então usando a componente  $z$  de  $c_h$  como variável livre, ao invés de 0. Então tem-se:

$$c_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix}$$

da equação (3.3.11)

$$w_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ \frac{kz}{\lambda} + k \end{bmatrix}$$

que, convertendo para coordenadas cartesianas resulta

$$w = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{\lambda x_0}{\lambda + z} \\ \frac{\lambda y_0}{\lambda + z} \\ \frac{\lambda z}{\lambda + z} \end{bmatrix}$$

Substituindo o último termo nos demais, tem-se

$$X = \frac{x_0}{\lambda}(\lambda - Z) \quad Y = \frac{y_0}{\lambda}(\lambda - Z)$$

### 3.4. MODELAGEM DA CÂMERA

As equações (3.3.5) e (3.3.6) caracterizam a formação de uma imagem através da projeção de pontos tridimensionais em um plano de imagem. Estas duas equações constituem portanto um modelo matemático básico de uma câmera. Este modelo é baseado na hipótese de que os sistemas de coordenadas global e da câmera são coincidentes. Considera-se agora um problema mais genérico em que os dois sistemas de coordenadas podem estar separados. Porém, o objetivo continua sendo a obtenção das coordenadas da imagem no plano de qualquer ponto tridimensional.

A situação é mostrada na figura 2, que apresenta o sistema de coordenadas globais (X, Y, Z) usados para localizar a câmera e o ponto 3D, denominado w. A figura também mostra o sistema de coordenadas da câmera (x, y, z) e pontos de imagem, denominados c. Assume-se que a câmera possua os movimentos de rotação em torno dos eixos X e Z. A distância da câmera à origem do sistema de coordenadas globais é denominada pelo vetor  $w_0$ , e a distância da câmera ao centro do plano da imagem, r.

A técnica para obtenção do modelo para a câmera é alinhar os sistemas de coordenadas global e da câmera aplicando um conjunto de transformações. Depois que isso é obtido, basta aplicar a transformação perspectiva dada em (3.3.4) para obter as coordenadas do plano da imagem de um dado ponto 3D.

Supõem-se, inicialmente, que a câmera está em uma posição normal, no sentido que a câmera e a origem do plano de imagem estão na origem do sistema de coordenadas globais, e todos demais eixos estão alinhados. Começando da posição normal, a configuração apresentada na figura 2 pode ser atingida de diversas maneiras. Para aplicação da transformação perspectiva, deve-se aplicar as mesmas transformações aplicadas à câmera, também para todos os pontos 3D.

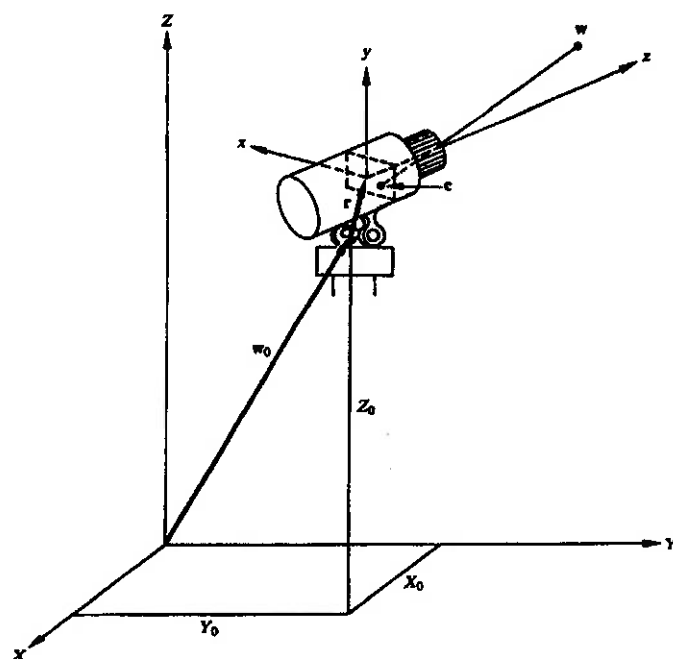


FIGURA 2. GEOMETRIA DE AQUISIÇÃO DE IMAGENS COM DOIS SISTEMAS DE COORDENADAS

Translação da origem do sistema de coordenadas globais para a posição da câmera é conseguida usando a matriz de transformação dada pela seguinte equação:

$$G = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4.1)$$

Em outras palavras, um ponto  $w_h$  com coordenadas  $(X_0, Y_0, Z_0)$  está na origem do novo sistema de coordenadas após a transformação  $Gw_h$ .

As rotações dos eixos  $x$  e  $z$  são conseguidas aplicando a matriz de transformação concatenada  $R = R_\alpha R_\theta$  dada pela equação:

$$R = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta \cdot \cos \alpha & \cos \theta \cdot \cos \alpha & \sin \alpha & 0 \\ \sin \theta \cdot \sin \alpha & -\cos \theta \cdot \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4.2)$$

Finalmente, o deslocamento da origem do plano da imagem representada pelo vetor  $r$  é obtida pela matriz de transformação:

$$C = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4.3)$$

Então aplicando ao vetor  $w_h$  a série de transformações  $CRGw_h$ , tem-se os sistemas de coordenadas global e da câmera coincidentes. As coordenadas no plano da imagem de um ponto  $w_h$  são finalmente obtidas pela equação Eq.(3.3.5). Assim tem-se a equação que representa a transformação perspectiva envolvendo dois sistemas de coordenadas:

$$c_h = PCRGw_h \quad (3.4.4)$$

Expandindo a equação Eq.(3.4.4) e convertendo para coordenadas cartesianas tem-se as coordenadas de imagem do ponto  $w$  cujas coordenadas globais são  $(X, Y, Z)$ :

$$x = \lambda \frac{(X - X_0) \cos \theta + (Y - Y_0) \sin \theta - r_1}{-(X - X_0) \sin \theta \cdot \sin \alpha + (Y - Y_0) \cos \theta \cdot \sin \alpha - (Z - Z_0) \cos \alpha + r_3 + \lambda} \quad (3.4.5)$$

$$y = \lambda \frac{(X - X_0) \sin \theta \cdot \cos \alpha + (Y - Y_0) \cos \theta \cdot \cos \alpha + (Z - Z_0) \sin \alpha - r_2}{-(X - X_0) \sin \theta \cdot \sin \alpha + (Y - Y_0) \cos \theta \cdot \sin \alpha - (Z - Z_0) \cos \alpha + r_3 + \lambda}$$

### 3.5. CALIBRAÇÃO DA CÂMERA

A implementação das equações para as coordenadas de um ponto 3D na imagem requer o conhecimento da distância focal e disposição linear e angular da câmera. Como estes parâmetros podem ser medidos diretamente, é freqüentemente mais conveniente determinar um ou mais parâmetros usando a câmera como instrumento de medição. Isto requer um conjunto de imagens de pontos cujas coordenadas globais são conhecidas, e o procedimento computacional utilizado para obter os parâmetros da câmera usando estes pontos conhecidos é comumente conhecido como calibração da câmera.

Com relação a Eq.(3.4.4), fixa-se  $A=PCRG$ . O elemento  $A$  contém todos os parâmetros da câmera, e sabe-se da Eq.(3.4.4) que  $c_h = Aw_h$ . Fazendo  $k=1$  na representação homogênea, têm-se:

$$\begin{bmatrix} c_{h.1} \\ c_{h.2} \\ c_{h.3} \\ c_{h.4} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.5.1)$$

Sabe-se que as coordenadas da câmera na forma cartesiana são dadas por

$$\begin{aligned} x &= c_{h.1}/c_{h.4} \\ y &= c_{h.2}/c_{h.4} \end{aligned} \quad (3.5.2)$$

Substituindo esse resultado na Eq.(3.5.1) e expandindo o produto matricial tem-se:

$$\begin{aligned} x.c_{h.4} &= a_{11}.X + a_{12}.Y + a_{13}.Z + a_{14} \\ y.c_{h.4} &= a_{21}.X + a_{22}.Y + a_{23}.Z + a_{24} \\ c_{h.4} &= a_{41}.X + a_{42}.Y + a_{43}.Z + a_{44} \end{aligned} \quad (3.5.3)$$

onde a expansão de  $c_{h.3}$  foi ignorada porque é relacionada com  $z$

Substituindo  $c_{h.4}$  nas duas primeiras equações tem-se duas equações com 12 coeficientes desconhecidos

$$\begin{aligned} a_{11}X + a_{12}Y + a_{13}Z - a_{41}xX - a_{42}xY - a_{43}xZ - a_{44}x + a_{14} &= 0 \\ a_{21}X + a_{22}Y + a_{23}Z - a_{41}yX - a_{42}yY - a_{43}yZ - a_{44}y + a_{24} &= 0 \end{aligned} \quad (3.5.4)$$

O procedimento de calibração consiste em obter  $m \geq 6$  pontos 3D (são 2 equações) com coordenadas conhecidas  $(X_i, Y_i, Z_i)$ ,  $i = 1, 2, \dots, m$ , capturando estes pontos com a câmera em uma dada posição para obter os pontos de imagem correspondentes  $(x_i, y_i)$ ,  $i = 1, 2, \dots, m$  e usando esses resultados nas equações Eq.(3.5.4) para resolver os coeficientes desconhecidos.

### 3.6. ALGORITMOS DE SEGMENTAÇÃO DE IMAGENS

O processo de segmentação consiste em subdividir a imagem em objetos de interesse, os marcadores, e demais partes constituintes. Esta fase é fundamental para a automação da análise de imagens, pois é neste momento que as entidades objetos de estudo são extraídas da imagem para subsequente processamento.

Os algoritmos de segmentação são baseados na análise de algumas propriedades da intensidade dos pixels: descontinuidade e similaridade. Algoritmos de segmentação por descontinuidade do nível de cinza dos pixels são de vasta aplicação na detecção de pontos isolados, linhas e limites de uma imagem. Entre os algoritmos baseados em similaridade pode-se citar o thresholding e segmentação orientada por regiões. Tais soluções são aplicáveis tanto para imagens estáticas quanto dinâmicas, porém no caso de imagens que sofrem alterações no tempo pode-se aprimorar consideravelmente a performance do algoritmo através do uso de algoritmos que se utilizam do movimento para a segmentação.

#### 3.6.1. Algoritmo de Detecção de Descontinuidades

Este método geralmente detecta pontos, linhas e limites baseados em pequenas máscaras espaciais.

O método é bastante simples, supõem-se uma máscara geral 3 x 3, com coeficientes  $w_1, w_2, \dots, w_9$  e  $x_1, x_2, \dots, x_9$  representam o nível de cinza dos pixels sobre a máscara quando a mesma está em uma determinada posição na imagem.

|       |       |       |
|-------|-------|-------|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

Calcula-se então o produto escalar entre os vetores  $w$  e  $x$ :

$$w'x = w_1.x_1 + w_2.x_2 + \dots + w_9.x_9$$

A detecção é obtida quando observa-se um formato semelhante a máscara, que ultrapasse um valor de tolerância.

$$|w'x| > T$$

A máscara deve ser ajustada de acordo com o objeto que se deseja detectar.

### 3.6.2. Algoritmo de Thresholding

Uma das técnicas mais simples e difundidas de segmentação de imagens, baseadas em uma propriedade básica da imagem, a descontinuidade de intensidade dos pixels.

Suponha uma imagem,  $f(x, y)$ , composta por objetos claros e fundo escuro, de forma a apresentar um histograma de nível de cinza, conforme figura X, onde os pixels dos objetos e do fundo dispõem de níveis de cinza agrupados em duas regiões dominantes. A técnica de thresholding extrai os objetos do fundo, separando essas regiões, e criando uma imagem binária. Qualquer ponto  $(x, y)$  na qual  $f(x, y) > T$  é declarado ponto do objeto; os demais são pontos do fundo, onde  $T$  é nível de intensidade conveniente.

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases}$$

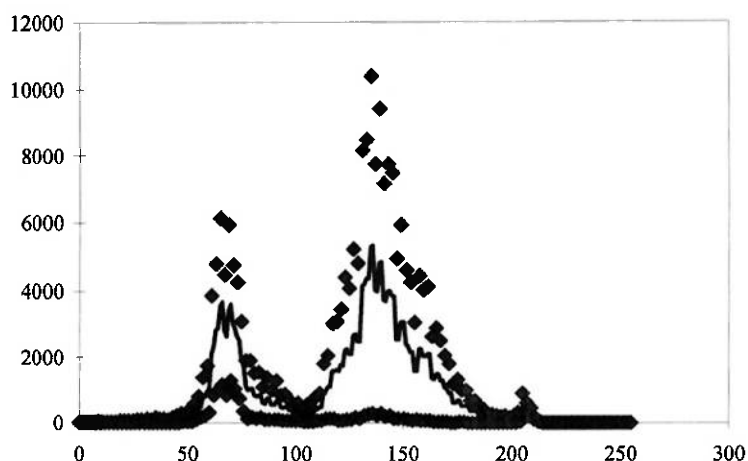


FIGURA 3. HISTOGRAMAS DE NÍVEIS DE CINZA QUE PODEM SER PARTICIONADOS PELO THRESHOLDING.

A grande vantagem deste método consiste na simplicidade, porém apresenta certas limitações de discriminação, uma vez que a detecção dos objetos de interesse é baseado simplesmente na intensidade de brilho. Problemas são encontrados caso a imagem a ser analisada apresenta superfícies refletoras, áreas brancas, luz, etc.

### 3.6.3. Algoritmo de Segmentação Orientada por Regiões

A segmentação tem por objetivo particionar a imagem em regiões, e esta técnica visa definir estas regiões diretamente, o que difere da técnica comentada anteriormente, o thresholding, que realiza a segmentação baseada na distribuição das propriedades dos pixels.

Uma região pode ser definida por: Seja uma imagem  $R$ , e o processo de segmentação que particiona  $R$  em  $n$  sub-regiões,  $R_1, R_2, \dots, R_n$ , tal que:

A união de todas as  $n$  sub-regiões,  $R_1, R_2, \dots, R_n$ , definem a imagem  $R$ ;

$R_i$  é uma região conectada,  $i = 1, 2, \dots, n$ ;

$R_i \cap R_j = \emptyset$  para todo  $i$  e  $j$ , com  $i \neq j$ ;

$P(R_i) = \text{TRUE}$  para  $i = 1, 2, \dots, n$ , onde  $P(R_i)$  é uma função lógica definida sobre os pontos do conjunto  $R_i$ ;

$P(R_i \cup R_j) = \text{FALSE}$  para  $i \neq j$ ;

Estas condições garantem uma segmentação completa, com regiões conectadas e disjuntas, e que cada região são definidas pelo atendimento de forma única a uma propriedade, que pode ser a intensidade de nível de cinza.

A formulação dessas regiões pode ser baseadas em dois métodos principais. A primeira, mais simples, baseia-se na agregação de pixels, onde começa-se com um conjunto de pontos "sementes" e a partir destes formam-se as regiões pelo anexação de pontos vizinhos que apresentam propriedades semelhantes (nível de cinza, textura, cor). A segunda alternativa baseia-se na subdivisão da imagem em um conjunto arbitrário e disjunto de regiões e então fundir e/ou separar as regiões tentando satisfazer as condições definidas para regiões.

#### 3.6.4. Algoritmos de Segmentação através do Movimento

O uso do movimento produzido pelo deslocamento relativo entre o sistema de aquisição de imagens e a cena capturada pode aprimorar o processo de segmentação de imagens e pode ser aplicado em ambos domínios, espacial e em frequência.

Utilizando-se técnicas espaciais, detecta-se alterações entre dois frames  $f(x, y, t_i)$  e  $f(x, y, t_j)$  tomados em tempos  $t_i$  e  $t_j$ , respectivamente, comparando-se as duas imagens pixel a pixel, produzindo uma imagem de diferenças definida por:

$$d_{ij}(x, y) = \begin{cases} 1 & \text{se } |f(x, y, t_i) - f(x, y, t_j)| > \theta \\ 0 & \text{caso contrário} \end{cases}$$

onde  $\theta$  é o critério de seleção.

Na análise de imagem dinâmica, todos os pixels em  $d_{ij}(x, y)$  com valor 1 são considerados resultado da movimentação do objeto. Este método é aplicável somente se as duas imagens são adquiridas e a iluminação é relativamente constante dentro dos limites estabelecidos por  $\theta$ . Na prática alguns valores 1 em  $d_{ij}(x, y)$  sempre resultam de ruídos. Porém tipicamente estes serão pontos isolados na imagem de diferença, o que pode ser facilmente corrigido formando regiões formadas por pixels,

isto é, considerando a menor unidade da imagem um conjunto de pixels. Isto pode resultar na ignora  o de pequenos e/ou lentos objetos, mas real  a as chances das demais entradas na imagem de diferen  a serem verdadeiramente devidas ao movimento.

Considerando o problema sob o dom  nio de freq  ncias, tem-se uma seq  ncia  $f(x, y, t)$ ,  $t=0, 1, \dots, T-1$ , de  $T$  imagens digitais de tamanho  $M \times N$  gerados por uma c  mera estacion  ria. Assume-se inicialmente que todos os frames possuem fundo homog  neo com intensidade nula, com uma   nica exce   o de um pixel-objeto de intensidade unit  ria que est   se movendo com velocidade constante  $v$ .

Aplicando a transformada de Fourier para a projec  o do plano da imagem no eixo  $x$ , para todos os frames, multiplicado por  $\exp[j2\pi k_1 x \Delta t]$ ,  $x = 0, 1, \dots, M-1$ , obt  m-se dois picos, um na freq  ncia  $v_1 k_1$  e outra em  $T - v_1 k_1$ . Este   ltimo pico    devido ao fen  meno de folding, sendo portanto ignorado. Assim o pico procurado pelo espectro de Fourier resultará  $v_1 k_1$ . Dividindo-se esse valor por  $k_1$  tem-se  $v_1$ , que    a componente da velocidade na dire  o  $x$  desde que assume-se que a taxa de aquisi  o de imagens seja conhecida. Para a determina  o da componente da velocidade na dire  o  $y$ , usa-se um racioc  nio an  logo.

### **3.7. ALGORITMOS DE DETEC  O DE PADR  ES**

Este procedimento deve ser capaz de identificar informa  o   til para os algoritmos de reconstru  o. Buscando maximizar a efici  ncia e acuracidade do processo deve-se ter o cuidado para n  o incluir falsos objetos, provenientes de ru  dos, o que pode ocasionar na aceita  o de um falso marcador ou na rejei  o de um verdadeiro.

#### **3.7.1. M  todos em Dom  nio Espacial**

Os m  todos em dom  nio espacial operam diretamente sobre os elementos b  sicos constituintes de uma imagem, os pixels. As fun  o es de processamento de imagens em dom  nio espacial podem ser descritas na forma:

$$g(x, y) = T[f(x, y)],$$

onde  $f(x, y)$  refere-se a imagem de entrada,  $g(x, y)$  a imagem processada, e  $T$  um operador sobre  $f(x, y)$ , definida em uma região próxima de  $(x, y)$ .

O método mais utilizado, devido a grande facilidade e velocidade de implementação, para a definição de uma vizinhança de  $(x, y)$  consiste no uso de uma subimagem quadrada centrada em  $(x, y)$ . O centro dessa subimagem é movimentado pixel a pixel aplicando o operador a cada posição  $(x, y)$  de forma a obter o valor de  $g$  em cada posição da imagem.

Sob sua forma mais simples assume-se um operador  $T$  definido em uma vizinhança  $1 \times 1$ . Neste caso a função  $g$  depende apenas do valor de  $f(x, y)$  e  $T$  torna-se uma função de transformação de nível de cinza. Vizinhanças mais largas permitem uma variedade de funções de processamento, tendo como método geral a definição de  $g$  em determinada coordenada como função dos valores de  $f$  em uma vizinhança pré-definida.

O método de máscaras, também denominado window, ou filtro, consistem em um dos principais usos dessa formulação. Basicamente uma máscara é uma pequena matriz bidimensional, cujos coeficientes são determinados para detectar uma dada propriedade da imagem.

Outra método para a detecção de marcadores fundamenta-se no conceito de centro de massa de um objeto plano. A determinação do centro geométrico de um marcador numa imagem pode ser feita usando algoritmos similares aos de identificação do centro de massa de um corpo, ou seja, fazendo uma somatória ponderada de cada posição  $(x, y)$  da imagem, usando-se a intensidade do pixel como peso:

$$CG_x = \frac{\sum I_i \cdot x_i}{\sum I_i} \quad CG_y = \frac{\sum I_i \cdot y_i}{\sum I_i}$$

### 3.7.2. Método em Domínio de Frequências

O fundamento das técnicas em domínio de frequências é o teorema da convolução. Assume-se  $g(x, y)$  uma imagem formada pela convolução de uma imagem  $f(x, y)$  e um operador invariante na posição  $h(x, y)$ , resumindo:

$$g(x, y) = h(x, y) * f(x, y)$$

Então, do teorema da convolução, obtém-se a relação de domínio de frequências:

$$G(u, v) = H(u, v) \cdot F(u, v)$$

onde  $G$ ,  $H$  e  $F$  são as transformadas de Fourier de  $g$ ,  $h$  e  $f$ , respectivamente. A transformada  $H(u, v)$  é às vezes denominada função de transferência do processo. O mesmo processo descrito em domínio espacial é obtido em domínio em frequência da seguinte forma:

$$g(x, y) = \mathcal{F}^{-1} [ H(u, v) \cdot F(u, v) ]$$

Em aplicações de processamento de imagens  $f(x, y)$  corresponderia a imagem dada e a saída desejada  $g(x, y)$  exibiria alguma característica destacada de  $f(x, y)$ , após a seleção de uma função  $H(u, v)$ . É interessante notar que limites em  $f(x, y)$  podem ser realçados usando a função de  $H(u, v)$  que enfatize as componentes de alta-frequência de  $F(u, v)$ .

### 3.7.3. Geração de máscaras espaciais a partir de especificações no domínio de frequência

Características como alta velocidade e simplicidade de implantação realçam a importância do uso de máscaras espaciais para o processamento de imagens, por outro lado certos filtros são mais convenientemente especificados sob o domínio de frequência.

O processo de filtragem no domínio das frequências é baseado na equação:

$$G(u, v) = H(u, v) \cdot F(u, v),$$

onde  $F(u, v)$  e  $G(u, v)$  são as transformadas de Fourier das imagens de entrada e saída, respectivamente, e  $H(u, v)$  é a função de transferência do filtro. A partir do teorema da convolução (3.3.8), sabe-se que a função acima pode ser descrita no domínio espacial pela expressão:

$$g(x, y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} h(x-i, y-k) \cdot f(i, k)$$

com  $x = 0, 1, 2, \dots, N-1$  e  $y = 0, 1, 2, \dots, N-1$ . A função  $h$  é a representação espacial do filtro, isto é, a transformada inversa de Fourier de  $H(u, v)$  também denominada máscara de convolução espacial,  $f$  é a imagem de entrada e  $g$  a imagem capturada.

Desde que  $H$  é a transformada de Fourier de  $h$ , segue que:

$$H(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} h(x, y) \exp[-j2\pi(ux + vy) / N]$$

para  $u, v = 0, 1, 2, \dots, N-1$ . Supõem-se, contudo, que restrinja-se  $h(x, y)$  para ser nulo para valores de  $x > n$  e  $y > n$ , com  $n < N$ , o que cria uma máscara de convolução  $mc$  com a transformada de Fourier:

$$MC(u, v) = \frac{1}{N} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} mc(x, y) \exp[-j2\pi(ux + vy) / N]$$

para  $u, v = 0, 1, 2, \dots, N-1$ . O objetivo é encontrar os coeficientes de  $mc(x, y)$  de forma que a magnitude do erro, descrito abaixo, é minimizado.

$$e^2 = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |MC(u, v) - H(u, v)|^2$$

A equação da máscara de convolução pode ser simplificada para a forma:

$$MC = C \cdot mc$$

onde  $MC$  é um vetor coluna de ordem  $N^2$  contendo os termos para  $MC(u, v)$ ,  $mc$  é um vetor coluna de ordem  $n^2$  contendo elementos de  $mc(x, y)$  e  $C$  é uma matriz  $N^2 \times n^2$  de termos exponenciais cujas posições são determinadas pela ordem de  $MC$  e  $mc$ .

Usando notação matricial, o erro pode ser descrito na forma:

$$e^2 = (MC - H) * (MC - H) = \|MC - H\|^2 = \|C.mc - H\|^2$$

Encontra-se o mínimo erro  $e^2$  com respeito a  $mc$  tomando-se a derivada parcial e equacionando-a para o vetor nulo:

$$\frac{\partial e^2}{\partial mc} = 2C * (C.mc - H) = 0$$

$$mc = (C * C)^{-1} C * H = C\#H$$

Assim consegue-se necessariamente os coeficientes de mínimo erro para a reconstrução da máscara de convolução  $n \times n$  a partir de um filtro  $H(u, v)$   $N \times N$ , especificado no domínio de frequências.

### 3.8. ALGORITMO DE FILTRAGEM DOS RESULTADOS

O método de otimização adotado consiste no método do Multiplicador de Lagrange aplicado em uma situação de minimização de erros sujeitos às restrições de igualdade.

Considere o problema de minimizar uma função ou um índice de performance:

$$J = \phi(x_1, x_2)$$

onde as variáveis  $x_1$  e  $x_2$  não são independentes, mas precisam satisfazer restrição:

$$\psi(x_1, x_2) = 0$$

Este problema é idêntico a minimizar:

$$J' = \phi(x_1, x_2) + \nu \psi(x_1, x_2) = G(x_1, x_2)$$

sujeito à

$$\psi(x_1, x_2) = 0$$

a menos da escolha de uma constante indeterminada  $\nu$  que é chamada multiplicador de Lagrange.

A primeira derivada resulta em

$$\delta J' = G_{x_1} \delta x_1 + G_{x_2} \delta x_2 = (\phi_{x_1} + \nu \psi_{x_1}) \delta x_1 + (\phi_{x_2} + \nu \psi_{x_2}) \delta x_2$$

Uma vez que  $\nu$  é arbitrária, escolhe-se seu valor de forma que o coeficiente do termo de variação dependente  $\delta x_2$  se anula. Assim, a primeira derivada contém apenas variações independentes e por isso deve ser nula, isto é, o coeficiente de  $\delta x_1$  deve ser zero. Portanto, as seguintes equações devem ser satisfeitas em um ponto extremo.

$$G_{x_1} = \phi_{x_1} + \nu \psi_{x_1} = 0, G_{x_2} = \phi_{x_2} + \nu \psi_{x_2} = 0, \psi = 0$$

Constituindo assim em um sistema de três equações e três incógnitas  $x_1$ ,  $x_2$  e  $\nu$ .

Obtém-se então a segunda derivada, dada por:

$$\delta^2 J' = G_{x_1 x_1} \delta x_1^2 + 2G_{x_1 x_2} \delta x_1 \delta x_2 + G_{x_2 x_2} \delta x_2^2 + G_{x_2} \delta^2 x_2$$

que deve ser maior ou igual a zero para um ponto de mínimo, assim:

$$\frac{\delta^2 J'}{\delta x_1^2} = G_{x_1 x_1} - 2G_{x_1 x_2} \frac{\psi_{x_1}}{\psi_{x_2}} + G_{x_2 x_2} \left( \frac{\psi_{x_1}}{\psi_{x_2}} \right)^2 \geq 0$$

Resumindo, para minimizar o índice de performance  $J = \phi(x_1, x_2)$  sujeito à restrição

$\psi(x_1, x_2) = 0$ , constrói-se a função  $J' = \phi + \nu \psi = G$  e usar as condições:  $G_{x_1} = 0$ ,  $G_{x_2} = 0$ ,  $\psi = 0$  e a condição de ponto de mínimo.

O uso do multiplicador de Lagrange tem o efeito de tornar todas derivadas independentes na primeira derivada. Contudo, isto não é verdadeiro na segunda derivada, e a derivada dependente deve ser eliminada.

A condição de otimização tem uma interessante interpretação geométrica. Um conjunto de contornos para a função  $\psi = \psi(x_1, x_2)$ , isto é,  $\psi = cte$ , é mostrada abaixo.

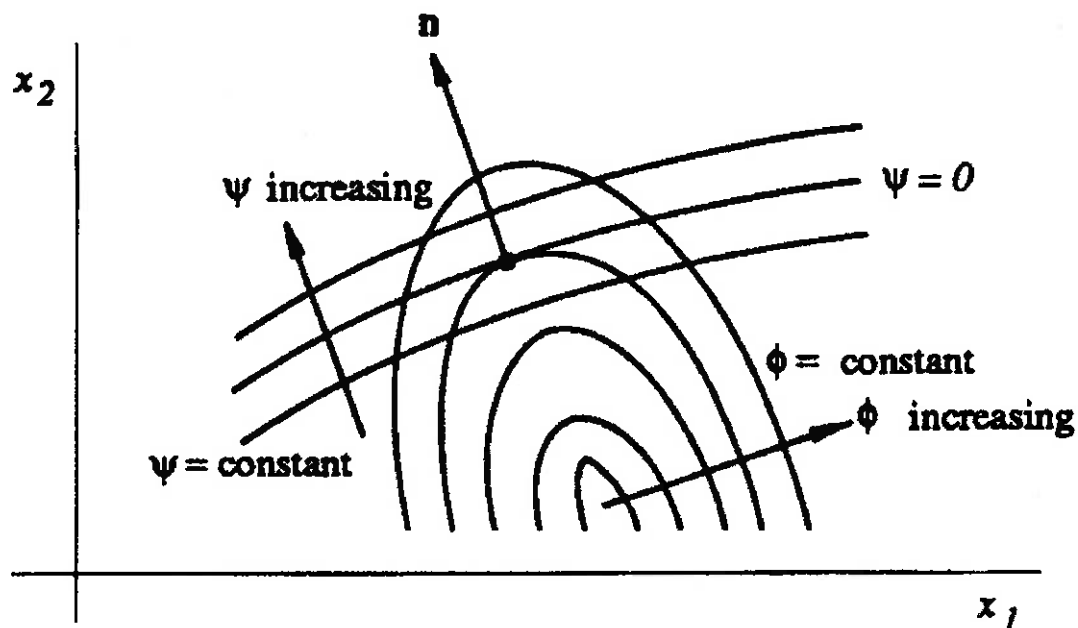


FIGURA 4. VETOR NORMAL

No ponto P sobre o contorno de  $\psi=0$ , existe um vetor normal ao contorno com a direção de  $\psi$  crescente. Um elemento linear  $\delta s$  de  $\psi=0$  sobre P é mostrado abaixo, onde a orientação de  $\delta s$  é determinado por  $\delta x_1$  e  $\delta x_2$ .

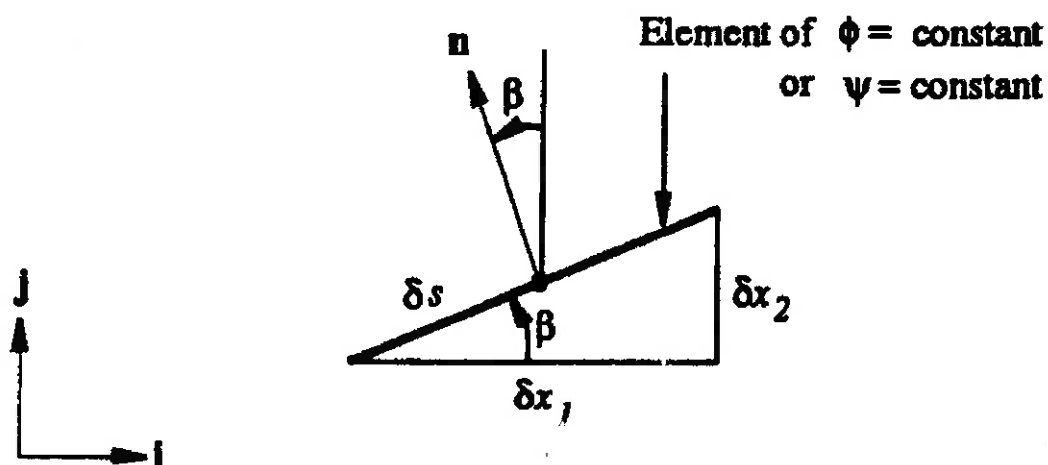


FIGURA 5. ORIENTAÇÃO DO VETOR NORMAL

Uma vez que  $\psi=0$  e  $\delta\psi = \psi_{x1}\delta x_1 + \psi_{x2}\delta x_2 = 0$ . Se  $i$  e  $j$  são versores nas direções  $x_1$  e  $x_2$ , o vetor normal é dado por:

$$n = -\sin\beta i + \cos\beta j$$

da figura, verifica-se que:

$$\sin \beta = \frac{\delta x_2}{(\delta x_1^2 + \delta x_2^2)^{1/2}}$$

$$\cos \beta = \frac{\delta x_1}{(\delta x_1^2 + \delta x_2^2)^{1/2}}$$

ou em termos das derivadas de  $\psi$

$$\sin \beta = -\frac{\psi_{x1}/\psi_{x2}}{[1 + (\psi_{x1}/\psi_{x2})^2]^{1/2}}$$

$$\cos \beta = \frac{1}{[1 + (\psi_{x1}/\psi_{x2})^2]^{1/2}}$$

Portanto o versor normal é

$$n_\psi = \frac{(\psi_{x1}/\psi_{x2})i + j}{[1 + (\psi_{x1}/\psi_{x2})^2]^{1/2}}$$

Um contorno do índice de performance ( $\phi = \text{cte}$ ) também passa através de P e pela mesma razão obtém-se a expressão para o versor normal:

$$n_\phi = \frac{(\phi_{x1}/\phi_{x2})i + j}{[1 + (\phi_{x1}/\phi_{x2})^2]^{1/2}}$$

Devido a condição de otimização, que é obtida na solução do sistema de equações descrito no método do multiplicador de Lagrange, que é:

$$\frac{\psi_{x1}}{\psi_{x2}} = \frac{\phi_{x1}}{\phi_{x2}}$$

verifica-se que ambos vetores normais são iguais. Isto significa que os contornos de  $\psi$  e  $\phi$  através de P são tangentes em P.

O termo  $\delta x_2$  incluído na segunda derivada, apesar de seu coeficiente se anular em um ponto extremo e não ser utilizado para problemas que podem ser resolvidos usando apenas a primeira e segunda derivada, é necessário para o desenvolvimento de derivadas de ordens maiores e, talvez, para métodos iterativos numéricos.

Outra observação interessante é que o multiplicador de Lagrange é mantido constante durante o processo de derivação. Isto é possível porque todas as derivadas são avaliadas no ponto mínimo, e todos os termos que são multiplicados por  $\delta u$ ,  $\delta^2 u$ , ... são zeros.

O problema genérico de minimização com restrições consiste em minimizar o índice de performance:

$$J = \phi(x_1, x_2, \dots, x_n)$$

sujeito às restrições:

$$\psi_1(x_1, x_2, \dots, x_n)$$

$$\psi_2(x_1, x_2, \dots, x_n)$$

$$\psi_p(x_1, x_2, \dots, x_n)$$

Cada restrição é multiplicada pelo multiplicador de Lagrange e adicionada ao índice de performance para formar a função:

$$J' = \phi(x_1, \dots, x_n) + \sum_{j=1}^p \nu_j \psi_j(x_1, \dots, x_n) = G(x_1, \dots, x_n)$$

A primeira derivada é dada por:

$$\delta J' = \sum_{k=1}^n G_{xk} \delta x_k$$

onde, contudo, as derivadas  $\delta x_k$  não são independentes. Se os multiplicadores de Lagrange  $p$  são escolhidos de forma que os coeficientes das derivadas dependentes de  $p$  (que podem ser arbitrariamente especificada) se anulem, então os  $n-p$  coeficientes restantes devem ser zeros por que as derivadas correspondentes são independentes, e a primeira derivada precisa ser zero para derivadas independentes.

Portanto, as condições necessárias para a primeira derivada são dadas por:

$$G_{xk} = 0, \quad k = 1 \rightarrow n$$

A segunda derivada da função é:

$$\delta^2 J' = \sum_{k=1}^n \sum_{l=1}^n G_{xkxl} \delta_{xk} \delta_{xl} \geq 0$$

onde o termo envolvendo  $G_{xk}$  foi omitido devido a condição necessária para primeira derivada. O relacionamento entre as derivadas são obtidas a partir de  $\delta\psi = 0$  e são dadas por:

$$\sum_{k=1}^n \psi_{xk}^j \cdot \delta_{xk} = 0, \quad j = 1 \rightarrow p$$

Estas  $p$  equações são resolvidas para as derivadas dependentes de  $p$ , que são eliminadas para formar uma nova função quadrática em termos das derivadas independente.

Através da aplicação do conteúdo teórico exposto na resolução da questão de aprimoramento dos valores das coordenadas cartesianas das articulações da perna obtidos com o processamento digital de imagens, verifica-se que um índice de performance adequado para a implementação do método do multiplicador de Lagrange consiste no quadrado da distância entre o valor da coordenada cartesiana aprimorada e o valor da coordenada cartesiana medida, isto é, o quadrado do erro entre o valor aprimorado, apresentando uma precisão suficiente para ser considerado o valor real, e o valor medido.

A restrição que a função índice de performance está sujeita é diferença nula entre o quadrado do valor aprimorado da distância entre os marcadores fixos nas articulações e o valor real da distância entre os marcadores, obtido pela medição prévia através de uma fita métrica e posteriormente transformada na unidade de processamento (pixels). Esta transformação de unidades é possível devido ao processo de calibração.

### 3.9. MÉTODO NUMÉRICO DE NEWTON-RAPSHON

A resolução do sistema de equações não lineares foi realizada através do método numérico de Newton-Raphson. Este método possui um modo eficiente de convergir

para as raízes das equações, caso a estimativa inicial seja suficientemente boa. O método pode também falhar grandemente na convergência, indicando que as raízes não existem nas proximidades da estimativa inicial.

Um problema típico apresenta  $N$  relações funcionais a serem zeradas, envolvendo variáveis  $x_i$ ,  $i = 1, 2, \dots, N$ :

$$F_i(x_1, x_2, \dots, x_N) = 0 \quad i = 1, 2, \dots, N.$$

Denominaremos  $x$  o vetor com todos os valores  $x_i$  e  $F$  o vetor com as funções  $F_i$ .

Nas proximidades de  $x$ , cada função  $F_i$  pode ser expandida em séries de Taylor

$$F_i(x + \delta x) = F_i(x) + \sum_{j=1}^N \frac{\partial F_i}{\partial x_j} \delta x_j + O(\delta x^2)$$

A matriz de derivadas parciais surgida na equação acima é a matriz Jacobiano  $J$

$$J_{ij} = \frac{\partial F_i}{\partial x_j}$$

A equação pode ser rescrita em notação matricial para a forma:

$$F(x + \delta x) = F(x) + J \cdot \delta x + O(\delta x^2)$$

Desprezando os termos de segunda ordem  $\delta x^2$  e estabelecendo  $F(x + \delta x) = 0$ , obtém-se um conjunto de equações lineares para as correções de  $\delta x$  que movem cada função para próximo do zero simultaneamente

$$J \cdot \delta x = -F$$

Esta equação matricial pode ser resolvida pelo método de decomposição LU. As correções são então adicionadas ao vetor solução,

$$x_{\text{new}} = x_{\text{old}} + \delta x$$

e o processo é iterado até a convergência. Em geral é aconselhável checar o grau de convergência das funções e variáveis. Uma vez que uma atinja a acuracidade da máquina, a outra não irá mudar.

#### 4. ARQUITETURA DO SISTEMA

A proposta de solução para o problema da análise do movimento de marcha consiste em um projeto completo de um sistema responsável por todas as etapas necessárias a uma análise cinemática da marcha humana. O fluxograma abaixo contribui para uma melhor compreensão do sistema, descrevendo as principais etapas e módulos constituintes bem como toda interação entre os mesmos e o meio externo.

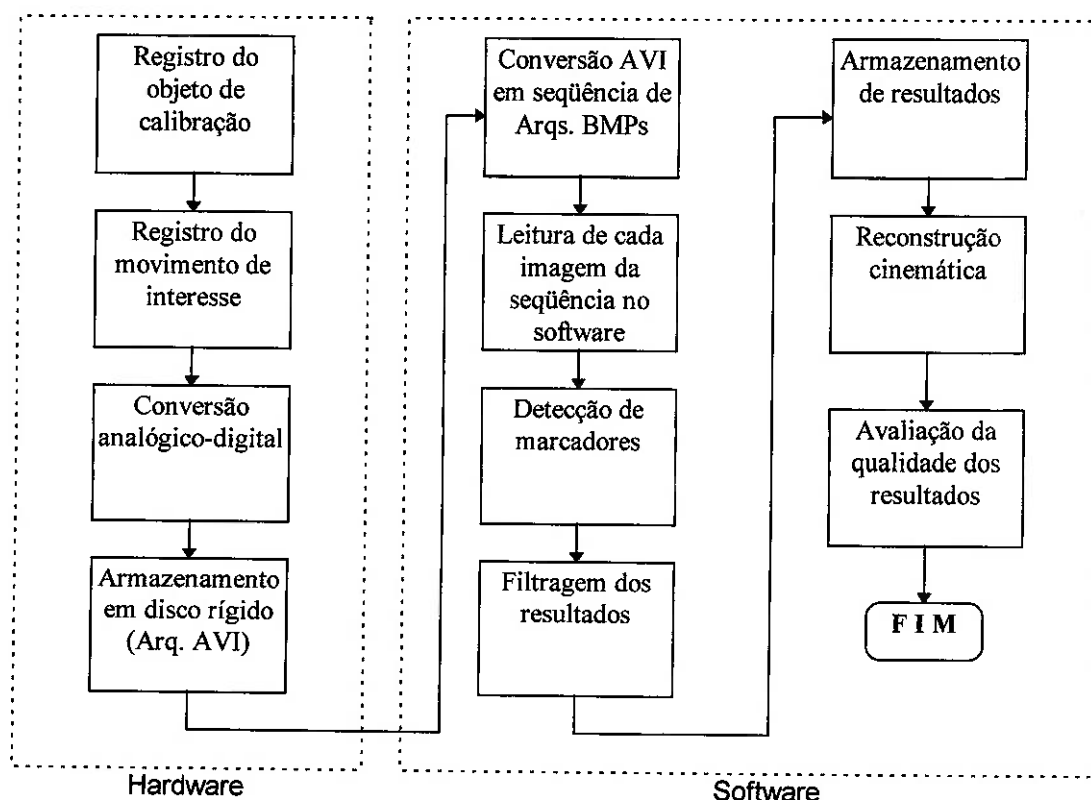


FIGURA 6. FLUXOGRAMA DO SISTEMA

Como já mencionado anteriormente o sistema pode ser dividido em duas grandes frentes de trabalho. A primeira frente de atuação engloba todo o processo de aquisição de imagens, consistindo na interface entre o meio externo e o processamento computacional, foco principal do projeto. Fazendo-se uma analogia, essa etapa consiste no fornecimento de matéria-prima para um futuro processamento. A grande dificuldade, no entanto, é garantir a qualidade das imagens adquiridas. O sistema requer imagens de um objeto de calibração para obtenção de um modo para

relacionar grandezas externas as grandezas processadas e exige também imagens do movimento completo da marcha humana, sem que a posição, orientação ou foco da câmera tenha sido alterado. Essas imagens são capturadas por uma câmera CCD diretamente acoplada a uma placa de vídeo, que são responsáveis pela digitalização das imagens que são armazenadas em meio magnético no formato AVI.

A segunda frente de atuação envolve toda a etapa de processamento de imagens e geração dos resultados. Em linhas gerais o processo tem início com conversão do arquivo AVI contendo todo o movimento em uma coleção de arquivos bitmap contendo cada imagem elementar do movimento. Na prática toma-se o movimento completo e executa-se uma análise do mesmo, quebrando-o em seus constituintes elementares, sobre os quais todo o processamento será executado. Em seguida faz-se a leitura dessas imagens bitmap e armazena-as na memória do computador. Nesse instante o sistema está apto a executar seus algoritmos de processamento de imagens.

Cada imagem é submetida a um processo de segmentação, onde os objetos de interesse são separados do restante da imagem. Em seguida executa-se o processo de identificação e rastreamento dos marcadores, onde cada marcador é identificado como sendo uma determinada articulação e rastreado ao longo de todo o movimento, isto é, cada marcador tem suas coordenadas em relação ao sistema de coordenadas da imagem computadas durante todo o movimento da marcha. Esses valores passam então por um processo de filtragem, que consiste em algoritmo de minimização de erro, de forma a se garantir resultados com maior acurácia e precisão. Em seguida o sistema fornece a reconstrução cinemática da trajetória, através do cálculo de todas as grandezas cinemáticas de interesse ao estudo da marcha humana.

O sistema apresenta algumas exigências de hardware:

- Câmera de Vídeo, com controle de tempo de abertura (shutter) e com a possibilidade de fixação de foco;

- Tripé para câmera de vídeo
- Conversor analógico-digital para sinal de vídeo, preferencialmente capaz de adquirir o sinal de vídeo em tempo real.
- Microcomputador PC 486 ou superior, mínimo de 8 Mbytes de memória RAM e sistema operacional Windows for 32 bits (Windows 95 ou NT);

Durante toda etapa de desenvolvimento, implementação e testes do sistema, a configuração de hardware utilizada foi a seguinte:

- Câmera: CCD Câmera Module B/W 3.8mm F2
- Adaptador de Vídeo: ATI 3D Charger (Português) (DirectDraw), Fabricante: ATI Technologies Inc. – Enhanced, Versão de Hardware: 154
- Controlador de Vídeo: ATI Multimedia Video Driver (Fabric.: ATI Technologies Inc)
- Microcomputador: Notebook Pentium 166Mhz MMX com 32Mbytes de RAM, sistema operacional Windows 95.

A câmera utilizada durante todo projeto apresenta como vantagens alta resolução e armazenamento digital das imagens em formato AVI e como principais desvantagens o foco variável, a lente esférica, o controle de shutter insuficiente(tempo de abertura), além de exigir uma placa de vídeo.

#### **4.1. ESTRUTURA DO SISTEMA DE AQUISIÇÃO DE IMAGENS**

A interface do sistema geral de análise de marcha humana com o ambiente externo é obtida através de um sensor de imagem, uma câmera acoplada a uma placa conversora de sinal analógico-digital captam a imagem da marcha de uma pessoa. A câmera é disposta a uma distância suficiente do ponto de interesse de forma a conseguir a aquisição de todas as informações necessárias, isto é, o movimento completo de um passo humano com o mínimo de distorção possível.

Essa posição da câmera é determinante para o sucesso do processo de aquisição de imagens. Uma distância grande possibilita a obtenção de todo o movimento de interesse à análise de marcha dentro de um pequeno ângulo de abertura. Esse ângulo é formado entre o eixo central da câmera e o segmento que une a origem do eixo central e o ponto de interesse mais distante da câmera. Isso possibilita um menor erro ocasionado devido a distância de um ponto 3D à câmera, já que haverá uma diferença no tamanho dos marcadores dependendo da posição deste em relação à câmera. Desta forma quando a pessoa estiver perfeitamente em frente a câmera, isto é, quando interceptar o eixo central da câmera, os marcadores atingirão seu tamanho máximo, pois essa é a posição mais próxima da câmera. No entanto uma distância excessiva compromete a captação de imagens, uma vez que os marcadores ficarão com dimensões muito reduzidas podendo no processo de segmentação de imagens serem confundidos com ruídos ou reflexos presentes na imagem adquirida.

Com o intuito de obter uma imagem mais apropriada e conveniente ao processamento de imagens o cenário onde ocorre a aquisição das imagens foi devidamente preparado. Desejou-se fundamentalmente controlar a fonte de luz e diminuir a presença de reflexos. Isso demandou uma completa preparação da sala de filmagens, o que foi razoavelmente conseguida através de diversos testes de iluminação e instalação de telas anti-reflexivas para o fundo.

Os testes de iluminação foram realizados através de filmagens sujeitas a diferentes configurações das fontes de luz. Dispunha-se de três fontes de luz fixas e os testes foram realizados com sete configurações diferentes.

Devido a precariedade dos equipamentos de preparação do cenário o resultado obtido não pode ser considerado ótimo, o que irá acarretar em uma dificuldade extra ao processamento digital de imagens. Nas imagens obtidas verifica-se ainda a presença de reflexos e a fonte de luz não está na posição mais apropriada, o que provoca a ocorrência de sombras indesejáveis nos marcadores.

O indivíduo sujeito à análise também sofreu um processo de preparação com o objetivo de reduzir os ruídos presentes na imagem. Isso é satisfatoriamente obtido através da utilização de vestimentas pretas sobre as quais são anexadas os marcadores reflexivos. O uso de calças e calçados pretos auxiliam no aumento de contraste obtido entre o cenário e os marcadores.

## **4.2. MARCADORES**

Existem basicamente dois tipos de marcadores distintos, marcadores ativos que necessitam de suprimento de energia para emissão de luz permitindo grande contraste com respeito ao demais objetos constituintes da imagem, e marcadores passivos compostos por material altamente reflexivo.

Marcadores ativos consistem usualmente de LEDs, light-emitting diodes, e apresentam algumas características desfavoráveis à aplicação. Dificultam a movimentação livre devido à presença dos fios alimentadores de energia e da unidade de sincronização e controle. Possuem um ângulo de emissão de luz restrito, dificilmente excedendo 50° da faixa com metade da potência emissora, o que acarreta em problemas quando há rotação do marcador durante o movimento e dificuldades adicionais para detecção simultânea por dois sensores diferentes para análise 3D. Além disso esses marcadores normalmente exigem time-multiplexed lighting, o que limita o número de marcadores.

Marcadores passivos consistem basicamente de suportes plásticos revestidos por papel reflexivo e são fixos ao corpo humano por meio de adesivos, o que permite uma análise com total liberdade de movimentos. Um marcador esférico é visível em qualquer direção pelos sensores e sua imagem não é alterada com rotações em seus eixos de simetria. Porém este tipo de marcador exige um processamento de imagem mais refinado devido ao rastreamento e identificação de cada marcador.

Nessa aplicação de análise de marcha humana, devido as razões expostas, definiu-se o uso de marcadores reflexivos. Diversos testes foram realizados com o intuito de

definir-se um formato e tamanho do marcador. Marcadores planos de diversos tamanhos foram testados, porém esses apresentavam excessiva deformação quando localizados em posições cujo ângulo formado pelo segmento que une o local do marcador e a origem da câmera e o eixo central da câmera fosse superior a  $30^\circ$  ou causadas por rotações em seus eixos de simetria. Assim optou-se definitivamente por marcadores esféricos reflexivos, implementados através de bolinhas de tênis de mesa fixas às articulações da perna, quadril, joelho, tornozelo e a ponta do pé através de fitas adesivas. O resultado final obtido confirmou a escolha do marcador, que garantiu um contraste suficiente com todo o restante da imagem, permitindo assim uma maior precisão na geração dos resultados.

#### **4.3. CALIBRAÇÃO DO SISTEMA**

O processo de calibração do sistema consiste na aquisição de imagens de objetos com dimensões conhecidas, o que foi implementado fixando quatro marcadores em uma estrutura retangular disposto a mesma distância da câmera, na qual foi realizada o movimento para análise. Assim consegue-se computar a relação entre as dimensões reais e as obtidas via algoritmos processamento de imagens necessária para uma correta análise do movimento.

A determinação da posição do objeto de calibração é fundamental para uma correta calibração da câmera. Escolheu-se em princípio posicionar o objeto de calibração na mesma distância da câmera em que o movimento foi capturado e disposto perpendicularmente ao eixo central da câmera, isto é, colocou-se o objeto perfeitamente em frente a câmera.

Com o intuito de corrigir as distorções provocadas pelo formato da lente esférica, que são visualmente verificados por uma deformação da imagem na região próxima às bordas, optou-se por realizar o processo de calibração supracitado também com o objeto de calibração posicionado nessas regiões problemáticas.

Assim a relação de calibração é efetuada a partir de três posições do objeto de calibração, considerando tanto as questões de resolução horizontal e vertical quanto as distorções devido à lente esférica da câmera.

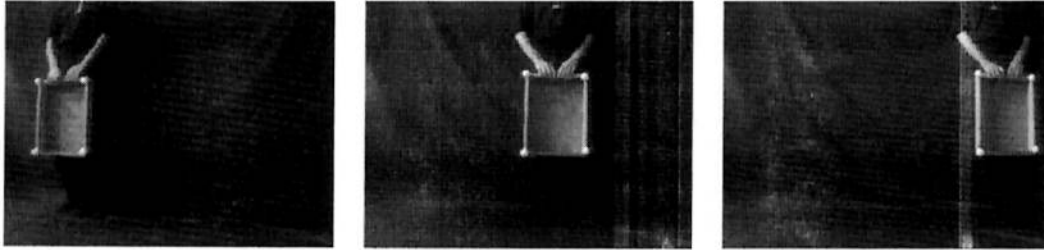


FIGURA 7. AQUISIÇÃO DAS IMAGENS DE CALIBRAÇÃO

Criteriosamente deveria-se calcular diferentes constantes de calibração para grandezas horizontais e verticais e também diferentes constantes de acordo com a posição do objeto de interesse à análise de movimento devido à distorção de imagens. Tal solução acarretaria em um esforço computacional demasiado, já que dependendo da posição do marcador em relação à câmera deveria-se utilizar duas determinadas relações de calibração, uma para resolução horizontal e outra para vertical.

Por esses motivos optou-se por uma simplificação retirando-se, para cada configuração da posição do objeto de calibração, valores médios das grandezas horizontais e verticais. Assim para cada configuração teria-se um valor para a relação de calibração, que considerava completamente a questão de resolução horizontal e vertical. De posse destas constantes de calibração, determina-se a relação de calibração final através do cálculo de uma média ponderada entre as diversas configurações, onde a configuração central recebe o dobro do peso das configurações marginais.

Através desse método obteve-se a seguinte relação de configuração que é implementada indiferentemente para qualquer posição do objeto de análise em movimento em relação à câmera.

- Relação de Calibração = 0,179 pixels / mm

#### 4.4. ESTRUTURA DO SISTEMA DE PROCESSAMENTO DE IMAGENS

O sistema de processamento de imagens pode ser estruturado em cinco módulos ou algoritmos principais, responsáveis por todo processamento de imagens e a reconstrução cinemática de toda a trajetória, com todas as grandezas têmporo-espaciais interessantes à análise, dentre as quais destaca-se: o comprimento do passo, deslocamentos, velocidades e acelerações lineares e angulares das articulações da perna durante todo o movimento de marcha.

Este sistema que têm como entrada as imagens do movimento completo da marcha humana e gera como saída os valores supracitados sob a forma de tabelas e gráficos.

O sistema é composto pelos seguintes módulos:

- Conversão de arquivos AVI em BMP
- Algoritmo de segmentação de imagens
- Algoritmo de identificação e rastreamento dos marcadores
- Algoritmo de filtragem de resultados
- Reconstrução cinemática da trajetória

Os três primeiros módulos foram implementados em linguagem Delphi, sendo o primeiro em Delphi 3 (32 bits) e os outros dois em Delphi 1 (16 bits). O Delphi, próprio para ambientes gráficos, o que permitiu a criação de uma interface amigável com o usuário. Essa característica é muito valorizada pelos usuários e merece especial destaque caso este projeto tenha como meta final a criação de um produto comercial.

O algoritmo de filtragem de resultados foi implementado em linguagem C, propícia para a resolução equações matemáticas mais complexas. Já o módulo responsável pela reconstrução cinemática da trajetória foi desenvolvido e implementado no aplicativo de planilha de cálculo MS-Excel 95.

A interação entre os diversos módulos é efetuada através de transferência de arquivos do tipo texto. Assim os dados provenientes dos algoritmos em Delphi são passados para o filtro descrito em linguagem C e posteriormente a transferência de informações

para a planilha MS-Excel 95 é executada através de arquivos de formato TXT. Este método apresenta como vantagens maior facilidade de implementação e maior segurança na operação, porém tem como principal revés uma queda de performance considerável.

#### **4.5. CONVERSÃO DE ARQUIVOS AVI EM BMP**

Este módulo permite a conversão de arquivos AVI contendo o movimento objeto de análise e subdivide em seus elementos básicos constituintes, isto é, em uma seqüência de imagens elementares, os frames, sob os quais será efetuado todo o processamento de imagens.

Esse subsistema permite a seleção do arquivo AVI que se deseja analisar. Após seu carregamento em memória é possível ainda selecionar ainda o trecho exato do movimento dentro do filme. Através dessa opção pode-se selecionar visualmente um passo completo, que consiste no movimento de interesse à análise de marcha, marcando-se dentro da seqüência original o início e o fim do passo da pessoa sob análise.

Após a seleção do movimento de interesse é ativada a conversão de formatos de arquivos, tendo como produto uma seqüência de arquivos bitmaps com nomeação automática padronizada.

#### **4.6. ALGORITMO DE SEGMENTAÇÃO DE IMAGENS**

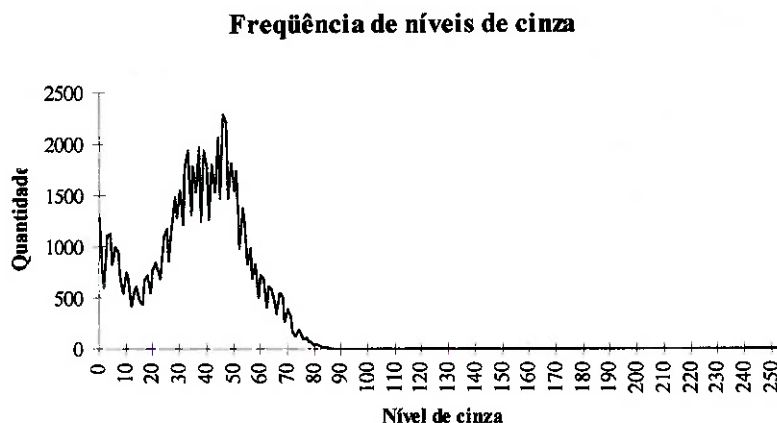
O algoritmo de segmentação selecionado foi o método de thresholding, devido fundamentalmente a simplicidade e velocidade de implementação.

O sistema permite operar tanto com imagens elementares como com seqüências completas oriundas do módulo de conversão de arquivos AVI em BMP. Dado o nome de um arquivo origem e um destino e escolhendo-se um nível de threshold, o software grava no arquivo destino uma imagem binarizada contendo somente pixel pretos (nível 0) ou brancos (nível 255), a partir do nível escolhido, ou seja: para cada pixel da

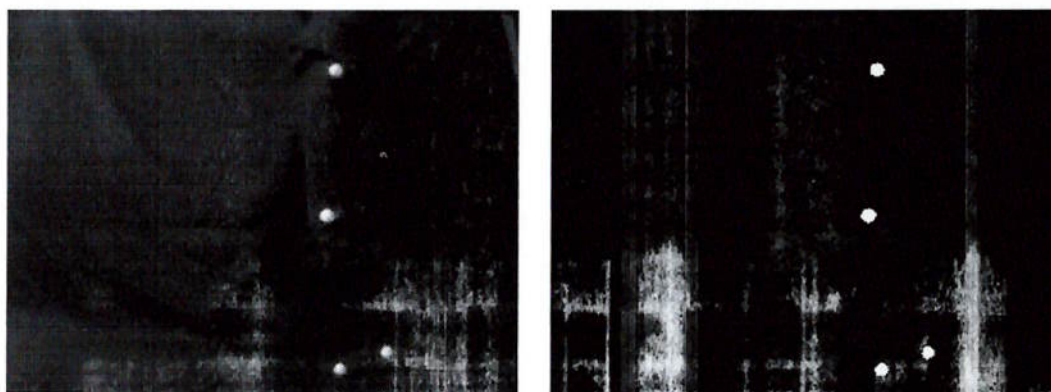
imagem original, compara com o valor escolhido. Se for menor, passa para preto; se for maior ou igual, para branco. Para fazer tal função o programa lê o arquivo original do disco para a memória, e, analisando o header do arquivo bitmap, verifica se é um arquivo válido (bitmap; níveis de cinza em 8 bits; não compactado) e o tamanho da imagem (linhas e colunas). Então, copia o header para outra posição de memória que armazenará o arquivo destino. Depois, num loop de "linhas x colunas", varre a imagem original e preenche a destino com o valor adequado. Finalmente, salva a imagem destino em disco.

A definição de um correto nível de thresholding é determinante para um menor esforço computacional do processamento de detecção de padrões. Caso opte-se para um gatilho de thresholding inferior ao valor ideal, ter-se-á como resultado uma imagem com excesso de ruídos acoplados ao objeto de interesse, caso contrário, com um gatilho superior ao valor ideal, perder-se-á informações muitas vezes importantes, deformando o objeto de interesse. A análise de um histograma é fundamental para a correta seleção do nível de thresholding. Neste gráfico tem-se a frequência de cada intensidade de nível de cinza dos pixels constituintes da imagem. Este procedimento foi implementado varrendo-se a imagem original e acrescentando-se num vetor estático [0..255] os valores de intensidade correspondente. Ao final do processamento, tem-se um vetor com a frequência de cada intensidade no vetor. Salva-se então este vetor num arquivo texto, que serve de entrada para uma planilha Excel, onde os gráficos são traçados.

Aplicando esse conceito para o caso em estudo, verifica-se que a quase a totalidade dos pixels são pertencentes ao cenário e menos de 1% dos pixels da imagem correspondem aos marcadores. Assim um valor próximo de 100 corresponderia a uma boa estimativa para o nível de thresholding, o que pode ser verificado através da imagem original e a imagem processada abaixo.



*FIGURA 8. HISTOGRAMA CORRESPONDENTE A IMAGEM ORIGINAL*



*FIGURA 9. IMAGENS ORIGINAL E PROCESSADA COM O ALGORITMO DE THRESHOLDING*

#### **4.7. ALGORITMO DE IDENTIFICAÇÃO E RASTREAMENTO DOS MARCADORES**

O algoritmo de identificação e rastreamento dos marcadores selecionado corresponde ao método de convolução no domínio espacial. Esta técnica de reconhecimento de padrões baseada tanto no formato quanto nas dimensões do marcador é baseado no conceito de convolução ou correlação de uma máscara espacial.

Desta forma a função de correlação é maximizada quando a imagem se aproxima da forma esperada, e é minimizada quando a imagem diferir significativamente. O funcionamento deste algoritmo consiste no rastreamento por toda imagem, pixel por pixel, da máscara checando a função de correlação. Ao se achar uma porção da imagem com uma porcentagem de coincidência com a máscara superior a uma

tolerância pré-definida, identifica-se esta porção como um marcador e, assim, tem-se as coordenadas do marcador.

A definição da máscara é determinante para o algoritmo de identificação dos marcadores. A estipulação de um formato e dimensão para a máscara exige uma bateria de testes que em conjunto com uma boa estimativa de tolerância garantem a detecção de todos os marcadores ao longo de todo o movimento. Tal tarefa merece especial atenção devido a deformações da imagem dos marcadores devido a incorreta iluminação, provocando sombras inconvenientes ao processamento de imagens, velocidades excessivas dos marcadores e distância variável dos marcadores à câmera.

A solução implementada visa atender as especificações estipuladas, identificando todos os marcadores durante todo o ciclo de marcha. Devido a grande diversidade de formatos e dimensões encontradas ao longo das imagens do movimento de marcha, estipulou-se uma máscara média e uma tolerância pré-definido, mostrada abaixo:

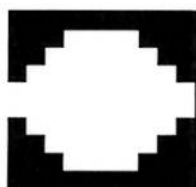


FIGURA 10. MÁSCARA DE CONVOLUÇÃO ESPACIAL

Com essa configuração o sistema identificava todos os marcadores, porém alguns são identificados mais de uma vez. Tal problema foi resolvido via software, onde para a determinação de um marcador, faz-se através da comparação dos marcadores identificados em coordenadas muito próximas, selecionando apenas aquele que melhor atende às especificações da máscara espacial.

O processo de rastreamento dos marcadores é obtido nomeando os identificados na primeira imagem, no caso em questão: o marcador A corresponde ao quadril, B ao joelho, C ao tornozelo e D a ponto do pé.

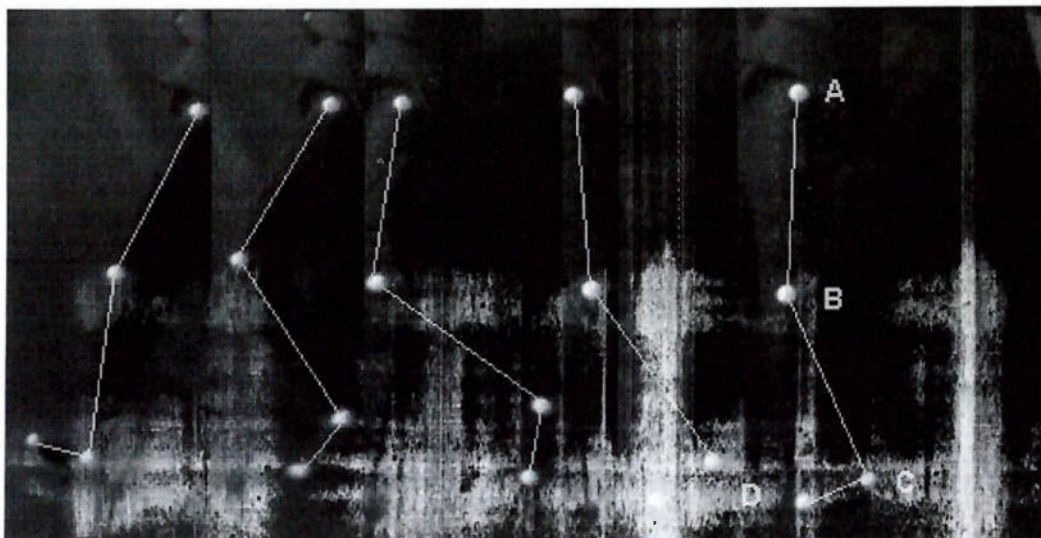


FIGURA 11. ESQUEMA DO PROCESSO DE IDENTIFICAÇÃO E RASTREAMENTO DE MARCADORES

Este processo é repetido ao longo de todos os frames que compõem o ciclo de marcha. Durante grande parte do movimento a ordem dos marcadores descrita é obtida na ordem inversa em que os marcadores são identificados, já que o processo de identificação foi implementado para a varredura da imagem de baixo para cima e da esquerda para direita. Porém em certos momentos do ciclo de marcha o tornozelo fica abaixo da ponta do pé, o que compromete o processo de rastreamento dos marcadores. Tal questão também é solucionada via software, uma vez que conhece-se a forma da perna e o movimento em análise. Assim sempre que o primeiro e o segundo marcador forem detectados, o sistema verifica qual possui menor coordenada horizontal, isto é, qual possui a coordenada horizontal situada mais a esquerda. Isto restringe que o movimento seja executado da direita para a esquerda. Caso queira-se analisar uma marcha da esquerda para direita basta alterar esta comparação.

#### 4.8. ALGORITMO DE FILTRAGEM DE RESULTADOS

Com o intuito de aprimorar o resultado obtido através do processamento digital de imagens, os valores resultantes, isto é, os valores das coordenadas cartesianas das articulações de interesse na análise da marcha humana passarão agora por um novo

processamento visando a minimização de erros e conseqüentemente uma maior precisão e acurácia dos dados de análise.

A principal fonte de discrepância entre o valor real e o valor medido é a baixa resolução da câmera utilizada, tempo insuficientemente pequeno entre um frame e outro e imperfeições na iluminação. O fator resolução motiva diretamente o erro no cômputo da posição dos marcadores. O fator velocidade acarreta em um erro significativo no cálculo de coordenadas que possuem um velocidade suficientemente alta para produzirem uma imagem desfocada. Já o fator iluminação provoca consideráveis distorções nos marcadores afetando diretamente sua identificação e conseqüente determinação de suas coordenadas.

Os valores processados até então são fornecidos ao algoritmo através de arquivo em formato texto. Esses valores consistem nas coordenadas cartesianas dos marcadores medidos no sistema de coordenadas da câmera. O filtro atua sobre estes valores retornando novas coordenadas aprimoradas, já que estas satisfazem restrições geométricas invariáveis no tempo.

O método adotado baseia-se na técnica de minimização de uma função índice de performance sujeita à restrições geométricas invariantes no tempo.

O índice de performance selecionado pode ser descrito na seguinte forma:

$$\begin{aligned} & (x_A - x_A^*)^2 + (y_A - y_A^*)^2 + \\ & (x_B - x_B^*)^2 + (y_B - y_B^*)^2 + \\ & (x_B - x_B^*)^2 + (y_B - y_B^*)^2 + \\ & (x_B - x_B^*)^2 + (y_B - y_B^*)^2 = 0 \end{aligned}$$

A restrição geométrica estabelecida corresponde a:

$$\begin{aligned} & \{[(x_A - x_B)^2 + (y_A - y_B)^2 - F^2] + \\ & [(x_B - x_C)^2 + (y_B - y_C)^2 - T^2] + \\ & [(x_C - x_D)^2 + (y_C - y_D)^2 - P^2]\} = 0 \end{aligned}$$

Aplicando o método de minimização de erros através da técnica do multiplicador de Lagrange, atinge-se o seguinte sistema de equações não lineares, conforme previsto no fundamentos teóricos na seção 3.8:

A função escrita em linguagem C que realiza o cálculo do sistema de equações não lineares executa um número pré-definido de iterações,  $n_{trial}$ , começando da estimativa inicial para o vetor solução  $x[1...n]$ . As iterações finalizam se a soma das magnitudes das funções  $F_i$  for inferior a tolerância  $tolf$ , ou a soma dos valores absolutos das correções de  $\delta x_i$  for inferior a tolerância  $tolx$ . A rotina chama uma função que provê os valores da função  $F$  e a matriz Jacobiano  $J$ . A informação da estimativa inicial, isto é, as coordenadas cartesianas obtidas pelo processamento de imagens, está organizada em arquivo texto. A saída da rotina, o vetor solução  $x$  correspondendo às coordenadas ortogonais reais, também será estruturada em arquivo texto para posterior processamento.

#### **4.9. RECONSTRUÇÃO CINEMÁTICA DA TRAJETÓRIA**

O módulo de reconstrução cinemática é responsável pelo cálculo de todas as grandezas cinemáticas de interesse à análise de marcha. Os valores das coordenadas cartesianas de todos os marcadores para todo o movimento consiste na entrada deste módulo realizado através de arquivo formato texto. Ao abri-lo na planilha desenvolvida e implementada para a reconstrução cinemática da trajetória, automaticamente serão efetuados os cálculos dos parâmetros têmporo-espaciais: comprimento do passo, deslocamentos, velocidades e acelerações lineares e angulares das articulações da perna durante todo o movimento de marcha.

Instantaneamente serão elaborados gráficos de todas essas grandezas, possibilitando analisar seu comportamento durante todo movimento da marcha. Exemplos desses gráficos são apresentados em Anexo IV.

## 5. PRECISÃO E ACURACIDADE DO SISTEMA

A medição do grau de precisão e acuracidade do sistema, isto é, a estimativa do erro entre as grandezas cinemáticas reais e as calculadas apresenta grande dificuldade de ser dimensionada.

A comparação dos resultados obtidos através do sistema de processamento de imagens e sistemas profissionais completos de análise de marcha humana, como exemplo cita-se o sistema Vicom.

Outra maneira possível, porém que exige grande esforço manual consiste na verificação das medidas geométricas de cada frame, obtendo-se assim as coordenadas dos marcadores em relação ao sistema de coordenadas do frame. Essa alteração do sistema de coordenadas não implicará em qualquer problema, uma vez que o interesse é voltado fundamentalmente aos deslocamentos relativos lineares e angulares. Para a verificação de grandezas como velocidades e acelerações lineares e angulares faz-se necessária a medição das grandezas supracitadas em dois ou mais frames, obtendo-se uma estimativa satisfatória velocidade e aceleração.

Um modo diferente para a verificação da precisão e acurácia do sistema é a comparação qualitativa dos gráficos gerados pelo sistema para o movimento de uma pessoa normal em marcha com ciclos padrões de marcha humana.

## 6. DISCUSSÃO

Através da análise dos resultados obtidos verifica-se que os valores das grandezas têmporo-espaciais envolvidas na reconstrução cinemática da trajetória são qualitativamente satisfatórias. Esta análise baseou-se na comparação dos gráficos confeccionados para o ângulo de giro das articulações do joelho e tornozelo com um modelo padrão denominado ciclo de marcha normal.

Eventuais desvios observados são justificados pela aquisição de uma marcha humana não natural, pelo controle de abertura (shutter) insuficiente para movimentos em alta velocidades e falta de precisão nas medições geométricas. Uma maior quantidade de testes provavelmente eliminaria grande parte destes problemas, possibilitando uma análise mais criteriosa.

No entanto a constatação da eficácia do algoritmo de filtragem dos resultados através da análise de gráficos e da medição direta das coordenadas relativas dos marcadores nas imagens elementares do movimento garante um aprimoramento considerável na acurácia e precisão do sistema.

Atentando-se a questão de performance observa-se que os algoritmos de processamento de imagens levam aproximadamente 19 minutos para a análise de um movimento composto por 72 frames. Tal resultado demonstrou-se altamente satisfatório, comprovando a eficácia dos métodos de processamento em domínio espacial para a aplicação em questão.

## 7. CONCLUSÕES

Pelos resultados apresentados pode-se concluir que o sistema é totalmente satisfatório para descrever quantitativamente a marcha de um sujeito estudado, com a precisão necessária. Por outro lado, a simplicidade do sistema permite a realização em pouco tempo das análises desejadas, no caso estudado cerca de 30 minutos entre filmagens e a obtenção dos resultados, quando a filmagem é feita em condições adequadas. O custo ínfimo do sistema se comparado com os sistemas profissionais disponíveis comercialmente é outro ponto altamente positivo. Estima-se o custo do hardware do sistema aqui apresentado seja inferior a 5% do custo de um sistema completo de análise de marcha profissional.

Como futuros aprimoramentos ao sistema identifica-se a implementação de visão estéreo, onde utilizaria-se duas ou mais câmeras permitiria a reconstrução 3D da trajetória, permitindo maior acurácia e precisão. Outro aprimoramento ao sistema atual seria a implementação de um algoritmo de estimativa estatística para a determinação de marcadores que ficam ocultos por alguns segundos devido a sobreposição do braço.

O sistema poderia também ser adaptado para análise de diversas patologias, tais como desvios de coluna, desvios na marcha frontal, etc., ampliando de maneira significativa o leque de aplicações para o sistema que estaria apto assim a análise de qualquer tipo de movimento humano. Melhorias no hardware e equipamentos fazer-se-ia necessária. A utilização de câmeras com melhor controle de shutter, lente plana, e maior resolução e fontes de luz apropriadas é altamente recomendada para garantia de resultados mais confiáveis. Garantia essa fundamental para tomada de decisões terapêuticas que envolvem a locomoção, aumentado-se assim as chances de sucesso de um tratamento.

## ANEXO I

### MANIPULAÇÃO DE BITMAPS E PALHETAS DE CORES

O termo bitmap provém do tempo quando os monitores apresentavam apenas um a cor além do preto, denominados monitores monocromáticos. Para uma imagem monocromática cada bit representava um pixel. Um valor 1 significava pixel ativo e, por sua vez, um valor 0 significava pixel desativado. Portanto verifica-se que uma imagem monocromática consiste em um mapa de bits.

Há muitos formatos de arquivos para armazenamento de bitmaps, incluindo RLE, JPEG, TIFF, TGA, PCX, BMP, PNG, PCD e GIF. Apresenta-se aqui bitmaps de 256 cores, onde oito bits representam um pixel. Um dos formatos de arquivos bitmap de 256 cores mais simples é o Windows' Bitmap. Este formato pode ser armazenado de modo não comprimido, por isso a leitura de arquivos BMP é razoavelmente simples; a maioria dos outros formatos são comprimidos.

Há também alguns poucos sub-tipos de formatos de arquivo BMP. O formato apresentado aqui é o formato Windows' RGB-encoded BMP. Para bitmaps de 256 cores, este possui um header de 54 bytes (ver tabela I) seguida de uma tabela de palheta decore de 1024 bytes. Depois disso está o bitmap propriamente dito, que começa no canto inferior esquerdo.

**Tabela 1. Header do formato de arquivo Windows' BMP**

| Dado               | Descrição                                     |
|--------------------|-----------------------------------------------|
| WORD Type;         | Tipo de arquivo. Setado como "BM."            |
| DWORD Size;        | Tamanho em DWORDs do arquivo .                |
| DWORD Reserved;    | Reservado. Setado como zero.                  |
| DWORD Offset;      | Offset dos dados.                             |
| DWORD headerSize;  | Tamanho do resto do header. Setado como 40.   |
| DWORD Width;       | Largura do bitmap em pixels.                  |
| DWORD Hight;       | Altura do bitmap em pixels.                   |
| WORD Planes;       | Número de planos. Setado como 1.              |
| WORD BitsPerPixel; | Número de bits por pixel.                     |
| DWORD Compression; | Tipo de compressão. Usualmente setado como 0. |

|                        |                                           |
|------------------------|-------------------------------------------|
| DWORD SizeImage;       | Tamanho em bytes do bitmap.               |
| DWORD XpixelsPerMeter; | Pixels na horizontal por metro.           |
| DWORD YpixelsPerMeter; | Pixels na vertical por metro.             |
| DWORD ColorUsed;       | Número de cores utilizadas.               |
| DWORD ColorsImportant; | Número de cores consideradas importantes. |

## ANEXO II

### TRANSFORMADA DE FOURIER

Assuma  $f(x)$  uma função contínua de variável real  $x$ . A transformada de Fourier de  $f(x)$ , denominada por  $\mathfrak{F}\{f(x)\}$ , é definida pela equação (All.1)

$$\mathfrak{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) \exp[-j2\pi ux] dx$$

onde  $j = (-1)^{0.5}$

Dado  $F(u)$ ,  $f(x)$  pode ser obtida através da transformada inversa de Fourier, definida pela equação (All.2)

$$\mathfrak{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) \exp[j2\pi ux] du$$

equações podem ser solucionadas se  $f(x)$  for contínua e integrável e  $F(u)$  integrável.

A transformada de Fourier de uma função real é geralmente complexa, isto é:

$$F(u) = R(u) + jI(u)$$

onde  $R(u)$  e  $I(u)$  são respectivamente as componentes reais e imaginárias de  $F(u)$ .

Outro modo conveniente de expressar é sob a forma exponencial:

$$F(u) = |F(u)| e^{j\phi(u)}$$

onde

$$|F(u)| = [R^2(u) + I^2(u)]^{0.5}$$

$$\phi(u) = \tan^{-1} \left[ \frac{I(u)}{R(u)} \right]$$

A magnitude da função  $|F(u)|$  é denominada espectro de Fourier de  $f(x)$ ,  $\phi(u)$  o ângulo de fase, e  $u$  a frequência.

## ANEXO III

### CONVOLUÇÃO E CORRELAÇÃO

Essas duas relações das transformadas de Fourier constituem uma ligação básica entre os domínios espaciais e em frequência. Essas relações, denominadas convolução e correlação, são de fundamental importância no desenvolvimento de técnicas de processamento de imagens baseadas na transformada de Fourier.

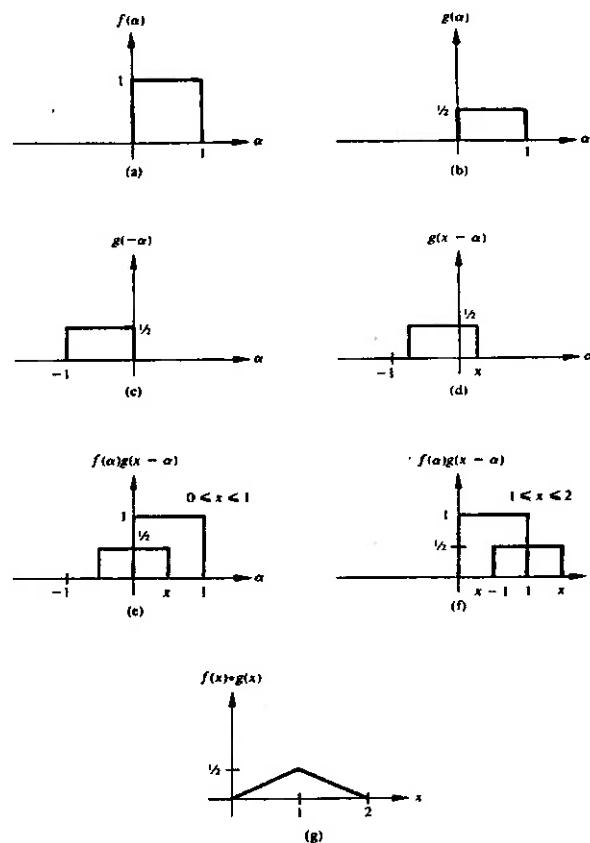
#### a. Convolução

A convolução de duas funções  $f(x)$  e  $g(x)$ , denotadas por  $f(x)*g(x)$ , é definida pela integral

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha)g(x - \alpha)d\alpha$$

onde  $\alpha$  é a variável de integração.

Como a mecânica da integral de convolução não particularmente de fácil visualização, seu entendimento é aprimorado através da ilustração gráfica.



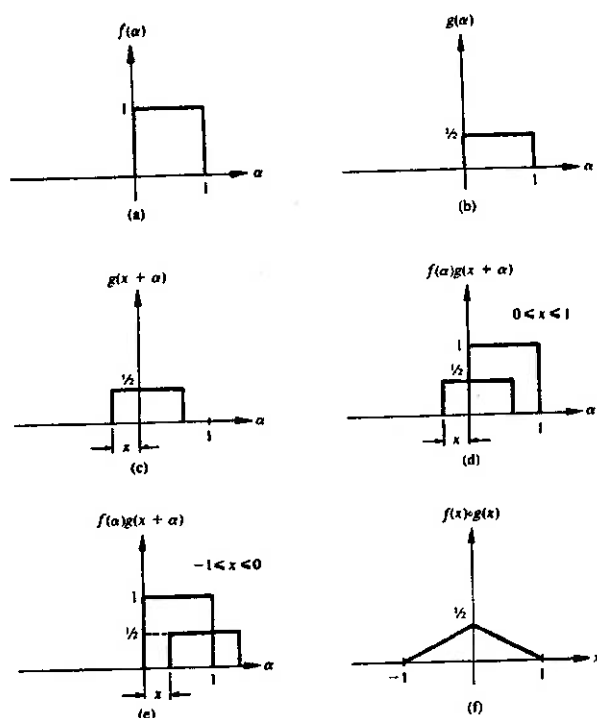
**Figura 12. Ilustração gráfica da convolução**

## b. Correlação

A correlação entre duas funções  $f(x)$  e  $g(x)$ , denotadas por  $f(x) \circ g(x)$ , é definida pela integral

$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f^*(\alpha) g(x + \alpha) d\alpha$$

onde  $*$  é o complexo conjugado. A diferença entre as definições de correlação e convolução é que a função  $g(x)$  não é dobrada sobre a origem. Assim para executar a correlação simplesmente desliza-se  $g(x)$  sobre  $f(x)$  e integra-se o produto de  $-\infty$  a  $\infty$  para cada valor de deslocamento  $x$ . O procedimento é ilustrado na figura 13, que deve ser comparado com a figura 12.



**Figura 13. Ilustração gráfica da correlação**

Uma das principais aplicações da correlação em processamento de imagens é na área de detecção de padrões, onde o problema é encontrar a mais fiel identificação entre uma dada imagem desconhecida e um conjunto de imagens de origem conhecida. Um método é computar a correlação entre a imagem desconhecida e cada imagem conhecida. A identificação mais fiel pode então ser verificada selecionando-se a imagem que a função correlação atinge seu valor máximo. Uma vez que o resultado da correlação são funções bi-dimensionais, isto envolve a procura pela maior amplitude de cada função. Como no caso da convolução discreta, o cálculo de  $f_e(x, y) \circ g_e(x, y)$  é freqüentemente mais eficientemente obtido no domínio de freqüências usando um algoritmo FFT, Fast Fourier Transform, para a obtenção das transformadas de Fourier.

**TABELA 2. CONSTANTES E RESTRIÇÕES GEOMÉTRICAS**

[illegible]

**Tabela 3. Coordenadas Cartesianas dos Marcadores (pixels) – Valores Originais**

| Ax (px) | Ay (px) | Bx (px) | By (px) | Cx (px) | Cy (px) | Dx (px) | Dy (px) |
|---------|---------|---------|---------|---------|---------|---------|---------|
| 285,000 | 192,000 | 256,000 | 119,000 | 242,000 | 31,000  | 213,000 | 29,000  |
| 283,000 | 192,000 | 254,000 | 118,000 | 241,000 | 30,000  | 212,000 | 28,000  |
| 281,000 | 192,000 | 252,000 | 118,000 | 241,000 | 29,000  | 212,000 | 27,000  |
| 276,000 | 192,000 | 249,000 | 116,000 | 239,000 | 28,000  | 210,000 | 24,000  |
| 276,000 | 192,000 | 249,000 | 116,000 | 239,000 | 28,000  | 210,000 | 23,000  |
| 274,000 | 193,000 | 246,000 | 116,000 | 239,000 | 28,000  | 210,000 | 23,000  |
| 271,000 | 193,000 | 243,000 | 116,000 | 239,000 | 28,000  | 209,000 | 21,000  |
| 269,000 | 194,000 | 241,000 | 116,000 | 238,000 | 28,000  | 209,000 | 21,000  |
| 266,000 | 195,000 | 239,000 | 116,000 | 238,000 | 28,000  | 208,000 | 20,000  |
| 263,000 | 196,000 | 237,000 | 116,000 | 238,000 | 27,000  | 208,000 | 20,000  |
| 260,000 | 197,000 | 235,000 | 116,000 | 238,000 | 27,000  | 208,000 | 20,000  |
| 257,000 | 197,000 | 233,000 | 115,000 | 238,000 | 27,000  | 207,000 | 20,000  |
| 255,000 | 198,000 | 231,000 | 115,000 | 238,000 | 27,000  | 207,000 | 20,000  |
| 252,000 | 199,000 | 230,000 | 115,000 | 238,000 | 27,000  | 207,000 | 19,000  |
| 250,000 | 200,000 | 229,000 | 115,000 | 238,000 | 27,000  | 207,000 | 19,000  |
| 247,000 | 200,000 | 226,000 | 114,000 | 238,000 | 27,000  | 207,000 | 19,000  |
| 245,000 | 201,000 | 226,000 | 114,000 | 238,000 | 27,000  | 207,000 | 19,000  |
| 243,000 | 201,000 | 225,000 | 114,000 | 238,000 | 27,000  | 207,000 | 19,000  |
| 241,000 | 202,000 | 224,000 | 114,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 240,000 | 202,000 | 224,000 | 114,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 238,000 | 202,000 | 223,000 | 114,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 236,000 | 203,000 | 222,000 | 114,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 234,000 | 203,000 | 221,000 | 114,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 233,000 | 203,000 | 220,000 | 114,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 231,000 | 203,000 | 219,000 | 113,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 229,000 | 203,000 | 217,000 | 113,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 227,000 | 203,000 | 215,000 | 113,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 225,000 | 203,000 | 214,000 | 113,000 | 237,000 | 27,000  | 207,000 | 19,000  |
| 222,000 | 203,000 | 213,000 | 113,000 | 237,000 | 27,000  | 207,000 | 18,000  |
| 220,000 | 203,000 | 211,000 | 112,000 | 237,000 | 27,000  | 207,000 | 18,000  |
| 218,000 | 202,000 | 209,000 | 112,000 | 237,000 | 27,000  | 207,000 | 18,000  |
| 216,000 | 202,000 | 208,000 | 112,000 | 237,000 | 27,000  | 207,000 | 18,000  |
| 213,000 | 202,000 | 205,000 | 112,000 | 237,000 | 28,000  | 207,000 | 18,000  |
| 210,000 | 202,000 | 204,000 | 112,000 | 237,000 | 28,000  | 207,000 | 18,000  |
| 207,000 | 202,000 | 202,000 | 112,000 | 237,000 | 28,000  | 207,000 | 18,000  |
| 204,000 | 201,000 | 199,000 | 112,000 | 236,000 | 29,000  | 207,000 | 19,000  |
| 201,000 | 201,000 | 198,000 | 112,000 | 236,000 | 29,000  | 207,000 | 19,000  |
| 199,000 | 201,000 | 195,000 | 111,000 | 236,000 | 29,000  | 207,000 | 19,000  |
| 196,000 | 201,000 | 194,000 | 111,000 | 236,000 | 29,000  | 207,000 | 19,000  |
| 193,000 | 200,000 | 193,000 | 111,000 | 236,000 | 30,000  | 208,000 | 19,000  |
| 190,000 | 200,000 | 191,000 | 111,000 | 236,000 | 30,000  | 208,000 | 19,000  |

TABELA 4. COORDENADAS CARTESIANAS DOS MARCADORES (MM) – VALORES ORIGINAIS

| Ax (mm)   | Bx (mm)   | Cx (mm)   | Dx (mm)   | Ay (mm)   | By (mm) | Cy (mm) | Dy (mm) |
|-----------|-----------|-----------|-----------|-----------|---------|---------|---------|
| 1.592,179 | 1.430,168 | 1.351,955 | 1.189,944 | 1.072,626 | 664,804 | 173,184 | 162,011 |
| 1.581,006 | 1.418,994 | 1.346,369 | 1.184,358 | 1.072,626 | 659,218 | 167,598 | 156,425 |
| 1.569,832 | 1.407,821 | 1.346,369 | 1.184,358 | 1.072,626 | 659,218 | 162,011 | 150,838 |
| 1.541,899 | 1.391,061 | 1.335,196 | 1.173,184 | 1.072,626 | 648,045 | 156,425 | 134,078 |
| 1.541,899 | 1.391,061 | 1.335,196 | 1.173,184 | 1.072,626 | 648,045 | 156,425 | 128,492 |
| 1.530,726 | 1.374,302 | 1.335,196 | 1.173,184 | 1.078,212 | 648,045 | 156,425 | 128,492 |
| 1.513,966 | 1.357,542 | 1.335,196 | 1.167,598 | 1.078,212 | 648,045 | 156,425 | 117,318 |
| 1.502,793 | 1.346,369 | 1.329,609 | 1.167,598 | 1.083,799 | 648,045 | 156,425 | 117,318 |
| 1.486,034 | 1.335,196 | 1.329,609 | 1.162,011 | 1.089,385 | 648,045 | 156,425 | 111,732 |
| 1.469,274 | 1.324,022 | 1.329,609 | 1.162,011 | 1.094,972 | 648,045 | 150,838 | 111,732 |
| 1.452,514 | 1.312,849 | 1.329,609 | 1.162,011 | 1.100,559 | 648,045 | 150,838 | 111,732 |
| 1.435,754 | 1.301,676 | 1.329,609 | 1.156,425 | 1.100,559 | 642,458 | 150,838 | 111,732 |
| 1.424,581 | 1.290,503 | 1.329,609 | 1.156,425 | 1.106,145 | 642,458 | 150,838 | 111,732 |
| 1.407,821 | 1.284,916 | 1.329,609 | 1.156,425 | 1.111,732 | 642,458 | 150,838 | 106,145 |
| 1.396,648 | 1.279,330 | 1.329,609 | 1.156,425 | 1.117,318 | 642,458 | 150,838 | 106,145 |
| 1.379,888 | 1.262,570 | 1.329,609 | 1.156,425 | 1.117,318 | 636,872 | 150,838 | 106,145 |
| 1.368,715 | 1.262,570 | 1.329,609 | 1.156,425 | 1.122,905 | 636,872 | 150,838 | 106,145 |
| 1.357,542 | 1.256,983 | 1.329,609 | 1.156,425 | 1.122,905 | 636,872 | 150,838 | 106,145 |
| 1.346,369 | 1.251,397 | 1.324,022 | 1.156,425 | 1.128,492 | 636,872 | 150,838 | 106,145 |
| 1.340,782 | 1.251,397 | 1.324,022 | 1.156,425 | 1.128,492 | 636,872 | 150,838 | 106,145 |
| 1.329,609 | 1.245,810 | 1.324,022 | 1.156,425 | 1.128,492 | 636,872 | 150,838 | 106,145 |
| 1.318,436 | 1.240,223 | 1.324,022 | 1.156,425 | 1.134,078 | 636,872 | 150,838 | 106,145 |
| 1.307,263 | 1.234,637 | 1.324,022 | 1.156,425 | 1.134,078 | 636,872 | 150,838 | 106,145 |
| 1.301,676 | 1.229,050 | 1.324,022 | 1.156,425 | 1.134,078 | 636,872 | 150,838 | 106,145 |
| 1.290,503 | 1.223,464 | 1.324,022 | 1.156,425 | 1.134,078 | 631,285 | 150,838 | 106,145 |
| 1.279,330 | 1.212,291 | 1.324,022 | 1.156,425 | 1.134,078 | 631,285 | 150,838 | 106,145 |
| 1.268,156 | 1.201,117 | 1.324,022 | 1.156,425 | 1.134,078 | 631,285 | 150,838 | 106,145 |
| 1.256,983 | 1.195,531 | 1.324,022 | 1.156,425 | 1.134,078 | 631,285 | 150,838 | 106,145 |
| 1.240,223 | 1.189,944 | 1.324,022 | 1.156,425 | 1.134,078 | 631,285 | 150,838 | 100,559 |
| 1.229,050 | 1.178,771 | 1.324,022 | 1.156,425 | 1.134,078 | 625,698 | 150,838 | 100,559 |
| 1.217,877 | 1.167,598 | 1.324,022 | 1.156,425 | 1.128,492 | 625,698 | 150,838 | 100,559 |
| 1.206,704 | 1.162,011 | 1.324,022 | 1.156,425 | 1.128,492 | 625,698 | 150,838 | 100,559 |
| 1.189,944 | 1.146,251 | 1.324,022 | 1.156,425 | 1.128,492 | 625,698 | 156,425 | 100,559 |
| 1.173,184 | 1.139,665 | 1.324,022 | 1.156,425 | 1.128,492 | 625,698 | 156,425 | 100,559 |
| 1.156,425 | 1.128,492 | 1.324,022 | 1.156,425 | 1.128,492 | 625,698 | 156,425 | 100,559 |
| 1.139,665 | 1.111,732 | 1.318,436 | 1.156,425 | 1.122,905 | 625,698 | 162,011 | 106,145 |
| 1.122,905 | 1.106,145 | 1.318,436 | 1.156,425 | 1.122,905 | 625,698 | 162,011 | 106,145 |
| 1.111,732 | 1.089,385 | 1.318,436 | 1.156,425 | 1.122,905 | 620,112 | 162,011 | 106,145 |
| 1.094,972 | 1.083,799 | 1.318,436 | 1.156,425 | 1.122,905 | 620,112 | 162,011 | 106,145 |
| 1.078,212 | 1.078,212 | 1.318,436 | 1.162,011 | 1.117,318 | 620,112 | 167,598 | 106,145 |
| 1.061,453 | 1.067,069 | 1.318,436 | 1.162,011 | 1.117,318 | 620,112 | 167,598 | 106,145 |

TABELA 5. DIMENSÕES GEOMÉTRICAS – VALORES ORIGINAIS

| distAB (px) | distBC (px) | distAC (px) | distCD (px) | distBD (px) | F (mm)  | T (mm)  | P (mm)  |
|-------------|-------------|-------------|-------------|-------------|---------|---------|---------|
| 78,549      | 89,107      | 166,643     | 29,069      | 99,745      | 438,823 | 497,803 | 162,396 |
| 79,480      | 88,955      | 167,356     | 29,069      | 99,318      | 444,020 | 496,956 | 162,396 |
| 79,480      | 89,677      | 167,836     | 29,069      | 99,403      | 444,020 | 500,990 | 162,396 |
| 80,654      | 88,566      | 168,122     | 29,275      | 99,925      | 450,579 | 494,784 | 163,445 |
| 80,654      | 88,566      | 168,122     | 29,428      | 100,846     | 450,579 | 494,784 | 164,402 |
| 81,933      | 88,278      | 168,671     | 29,428      | 99,725      | 457,726 | 493,173 | 164,402 |
| 81,933      | 88,091      | 168,074     | 30,806      | 100,901     | 457,726 | 492,128 | 172,300 |
| 82,873      | 88,051      | 168,870     | 29,833      | 100,245     | 462,980 | 491,906 | 166,664 |
| 83,487      | 88,006      | 169,331     | 31,048      | 100,881     | 466,405 | 491,652 | 173,354 |
| 84,119      | 89,006      | 170,839     | 30,806      | 100,285     | 469,938 | 497,238 | 172,300 |
| 84,770      | 89,051      | 171,418     | 30,806      | 99,725      | 473,577 | 497,489 | 172,300 |
| 85,440      | 88,142      | 171,058     | 31,780      | 98,494      | 477,319 | 492,413 | 177,345 |
| 86,400      | 88,278      | 171,843     | 31,780      | 97,985      | 482,683 | 493,173 | 177,345 |
| 86,833      | 88,363      | 172,569     | 32,016      | 98,717      | 485,102 | 493,647 | 178,338 |
| 87,556      | 88,459      | 173,416     | 32,016      | 98,489      | 489,138 | 494,185 | 178,338 |
| 88,527      | 87,824      | 173,234     | 32,016      | 96,881      | 494,563 | 490,635 | 178,338 |
| 89,051      | 87,824      | 174,141     | 32,016      | 96,881      | 497,489 | 490,635 | 178,338 |
| 88,843      | 87,966      | 174,072     | 32,016      | 96,690      | 496,327 | 491,430 | 178,338 |
| 89,627      | 87,966      | 175,046     | 31,048      | 96,509      | 500,710 | 491,430 | 173,354 |
| 89,443      | 87,966      | 175,026     | 31,048      | 96,509      | 499,680 | 491,430 | 173,354 |
| 89,269      | 88,119      | 175,003     | 31,048      | 96,338      | 498,711 | 492,286 | 173,354 |
| 90,094      | 88,284      | 176,003     | 31,048      | 96,177      | 503,321 | 493,205 | 173,354 |
| 89,944      | 88,459      | 176,026     | 31,048      | 96,026      | 502,483 | 494,185 | 173,354 |
| 89,944      | 88,645      | 176,045     | 31,048      | 95,885      | 502,483 | 495,225 | 173,354 |
| 90,796      | 87,864      | 176,102     | 31,048      | 94,763      | 507,243 | 490,858 | 173,354 |
| 90,796      | 88,295      | 176,182     | 31,048      | 94,530      | 507,243 | 493,268 | 173,354 |
| 90,796      | 88,769      | 176,284     | 31,048      | 94,340      | 507,243 | 495,918 | 173,354 |
| 90,670      | 89,022      | 176,409     | 31,048      | 94,260      | 506,535 | 497,332 | 173,354 |
| 90,449      | 89,286      | 176,638     | 31,321      | 95,189      | 505,301 | 498,805 | 174,377 |
| 91,444      | 88,888      | 176,819     | 31,321      | 94,085      | 510,860 | 496,579 | 174,377 |
| 90,449      | 89,493      | 176,028     | 31,321      | 94,021      | 505,301 | 499,961 | 174,377 |
| 90,355      | 89,811      | 176,255     | 31,321      | 94,005      | 504,776 | 501,737 | 174,377 |
| 90,355      | 89,889      | 175,647     | 31,623      | 94,021      | 504,776 | 502,172 | 176,394 |
| 90,200      | 90,250      | 176,082     | 31,623      | 94,048      | 503,909 | 504,188 | 176,394 |
| 90,139      | 91,000      | 176,567     | 31,623      | 94,133      | 503,569 | 508,380 | 176,394 |
| 89,140      | 90,874      | 174,951     | 30,676      | 93,343      | 497,991 | 507,673 | 171,375 |
| 89,051      | 91,285      | 175,525     | 30,676      | 93,434      | 497,489 | 509,974 | 171,375 |
| 90,089      | 91,679      | 175,935     | 30,676      | 92,779      | 503,290 | 512,172 | 171,375 |
| 90,022      | 92,130      | 176,590     | 30,676      | 92,914      | 502,917 | 514,695 | 171,375 |
| 89,000      | 91,706      | 175,354     | 30,983      | 93,215      | 497,207 | 512,324 | 168,355 |
| 89,006      | 92,661      | 176,114     | 30,983      | 93,557      | 497,238 | 517,657 | 168,355 |
| 88,142      | 91,924      | 174,909     | 30,971      | 93,941      | 492,413 | 513,541 | 172,300 |

TABELA 6. VELOCIDADES LINEARES DOS MARCADORES – VALORES ORIGINAIS

| Vax (mm/s) | Vbx (mm/s) | Vcx (mm/s) | Vdx (mm/s) | Vay (mm/s) | Vby (mm/s) | Vcy (mm/s) | Vdy (mm/s) |
|------------|------------|------------|------------|------------|------------|------------|------------|
| -335,196   | -335,196   | -167,598   | -167,598   | 0,000      | -167,598   | -167,598   | -167,598   |
| -335,196   | -335,196   | 0,000      | 0,000      | 0,000      | 0,000      | -167,598   | -167,598   |
| -837,989   | -502,793   | -335,196   | -335,196   | 0,000      | -335,196   | -167,598   | -502,793   |
| 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | -167,598   |
| -335,196   | -502,793   | 0,000      | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -502,793   | -502,793   | 0,000      | -167,598   | 0,000      | 0,000      | 0,000      | -335,196   |
| -335,196   | -335,196   | -167,598   | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -502,793   | -335,196   | 0,000      | -167,598   | 167,598    | 0,000      | 0,000      | -167,598   |
| -502,793   | -335,196   | 0,000      | 0,000      | 167,598    | 0,000      | -167,598   | 0,000      |
| -502,793   | -335,196   | 0,000      | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -502,793   | -335,196   | 0,000      | -167,598   | 0,000      | -167,598   | 0,000      | 0,000      |
| -335,196   | -335,196   | 0,000      | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -502,793   | -167,598   | 0,000      | 0,000      | 167,598    | 0,000      | 0,000      | -167,598   |
| -335,196   | -167,598   | 0,000      | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -502,793   | -502,793   | 0,000      | 0,000      | 0,000      | -167,598   | 0,000      | 0,000      |
| -335,196   | 0,000      | 0,000      | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | -167,598   | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | 0,000      | 0,000      | 167,598    | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -167,598   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | 0,000      | 0,000      | 0,000      | -167,598   | 0,000      | 0,000      |
| -335,196   | -335,196   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -335,196   | -335,196   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -502,793   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | -167,598   |
| -335,196   | -335,196   | 0,000      | 0,000      | 0,000      | -167,598   | 0,000      | 0,000      |
| -335,196   | -335,196   | 0,000      | 0,000      | -167,598   | 0,000      | 0,000      | 0,000      |
| -335,196   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -502,793   | -502,793   | 0,000      | 0,000      | 0,000      | 0,000      | 167,598    | 0,000      |
| -502,793   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -502,793   | -335,196   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -502,793   | -502,793   | -167,598   | 0,000      | -167,598   | 0,000      | 167,598    | -167,598   |
| -502,793   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -335,196   | -502,793   | 0,000      | 0,000      | 0,000      | -167,598   | 0,000      | 0,000      |
| -502,793   | -167,598   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -502,793   | -167,598   | 0,000      | 167,598    | -167,598   | 0,000      | 167,598    | 0,000      |
| -502,793   | -335,196   | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      | 0,000      |
| -1.173,184 | -502,793   | -167,598   | -167,598   | -167,598   | 0,000      | 335,196    | 0,000      |

TABELA 7. ACELERAÇÕES LINEARES DOS MARCADORES (MM/S<sup>2</sup>) – VALORES ORIGINAIS

| Aax<br>(mm/s <sup>2</sup> ) | Abx<br>(mm/s <sup>2</sup> ) | Acx<br>(mm/s <sup>2</sup> ) | Adx<br>(mm/s <sup>2</sup> ) | Aay<br>(mm/s <sup>2</sup> ) | Aby<br>(mm/s <sup>2</sup> ) | Acy<br>(mm/s <sup>2</sup> ) | Ady (mm/s <sup>2</sup> ) |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|--------------------------|
|                             |                             |                             |                             |                             |                             |                             |                          |
|                             |                             |                             |                             |                             |                             |                             |                          |
| 0,000                       | 0,000                       | 5.027,933                   | 5.027,933                   | 0,000                       | 5.027,933                   | 0,000                       | 0,000                    |
| -15.083,799                 | -5.027,933                  | -10.055,866                 | -10.055,866                 | 0,000                       | -10.055,866                 | 0,000                       | -10.055,866              |
| 25.139,665                  | 15.083,799                  | 10.055,866                  | 10.055,866                  | 0,000                       | 10.055,866                  | 5.027,933                   | 10.055,866               |
| -10.055,866                 | -15.083,799                 | 0,000                       | 0,000                       | 5.027,933                   | 0,000                       | 0,000                       | 5.027,933                |
| -5.027,933                  | 0,000                       | 0,000                       | -5.027,933                  | -5.027,933                  | 0,000                       | 0,000                       | -10.055,866              |
| 5.027,933                   | 5.027,933                   | -5.027,933                  | 5.027,933                   | 5.027,933                   | 0,000                       | 0,000                       | 10.055,866               |
| -5.027,933                  | 0,000                       | 5.027,933                   | -5.027,933                  | 0,000                       | 0,000                       | 0,000                       | -5.027,933               |
| 0,000                       | 0,000                       | 0,000                       | 5.027,933                   | 0,000                       | 0,000                       | -5.027,933                  | 5.027,933                |
| 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 5.027,933                   | 0,000                    |
| 0,000                       | 0,000                       | 0,000                       | -5.027,933                  | -5.027,933                  | -5.027,933                  | 0,000                       | 0,000                    |
| 5.027,933                   | 0,000                       | 0,000                       | 5.027,933                   | 5.027,933                   | 5.027,933                   | 0,000                       | 0,000                    |
| -5.027,933                  | 5.027,933                   | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | -5.027,933               |
| 5.027,933                   | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 5.027,933                |
| -5.027,933                  | -10.055,866                 | 0,000                       | 0,000                       | -5.027,933                  | -5.027,933                  | 0,000                       | 0,000                    |
| 5.027,933                   | 15.083,799                  | 0,000                       | 0,000                       | 5.027,933                   | 5.027,933                   | 0,000                       | 0,000                    |
| 0,000                       | -5.027,933                  | 0,000                       | 0,000                       | -5.027,933                  | 0,000                       | 0,000                       | 0,000                    |
| 0,000                       | 0,000                       | -5.027,933                  | 0,000                       | 5.027,933                   | 0,000                       | 0,000                       | 0,000                    |
| 5.027,933                   | 5.027,933                   | 5.027,933                   | 0,000                       | -5.027,933                  | 0,000                       | 0,000                       | 0,000                    |
| -5.027,933                  | -5.027,933                  | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                    |
| 0,000                       | 0,000                       | 0,000                       | 0,000                       | 5.027,933                   | 0,000                       | 0,000                       | 0,000                    |
| 0,000                       | 0,000                       | 0,000                       | 0,000                       | -5.027,933                  | 0,000                       | 0,000                       | 0,000                    |
| 5.027,933                   | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                    |
| -5.027,933                  | 0,000                       | 0,000                       | 0,000                       | 0,000                       | -5.027,933                  | 0,000                       | 0,000                    |
| 0,000                       | -5.027,933                  | 0,000                       | 0,000                       | 0,000                       | 5.027,933                   | 0,000                       | 0,000                    |
| 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                    |
| 0,000                       | 5.027,933                   | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                    |
| -5.027,933                  | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | -5.027,933               |
| 5.027,933                   | -5.027,933                  | 0,000                       | 0,000                       | 0,000                       | -5.027,933                  | 0,000                       | 5.027,933                |
| 0,000                       | 0,000                       | 0,000                       | 0,000                       | -5.027,933                  | 5.027,933                   | 0,000                       | 0,000                    |
| 0,000                       | 5.027,933                   | 0,000                       | 0,000                       | 5.027,933                   | 0,000                       | 0,000                       | 0,000                    |
| -5.027,933                  | -10.055,866                 | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 5.027,933                   | 0,000                    |
| 0,000                       | 10.055,866                  | 0,000                       | 0,000                       | 0,000                       | 0,000                       | -5.027,933                  | 0,000                    |
| 0,000                       | -5.027,933                  | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                       | 0,000                    |
| 0,000                       | -5.027,933                  | -5.027,933                  | 0,000                       | -5.027,933                  | 0,000                       | 5.027,933                   | 5.027,933                |
| 0,000                       | 10.055,866                  | 5.027,933                   | 0,000                       | 5.027,933                   | 0,000                       | -5.027,933                  | -5.027,933               |
| 5.027,933                   | -10.055,866                 | 0,000                       | 0,000                       | 0,000                       | -5.027,933                  | 0,000                       | 0,000                    |
| -5.027,933                  | 10.055,866                  | 0,000                       | 0,000                       | 0,000                       | 5.027,933                   | 0,000                       | 0,000                    |
| 0,000                       | 0,000                       | 0,000                       | 5.027,933                   | -5.027,933                  | 0,000                       | 5.027,933                   | 0,000                    |
| 0,000                       | -5.027,933                  | 0,000                       | -5.027,933                  | 5.027,933                   | 0,000                       | -5.027,933                  | 0,000                    |

TABELA 8. DESLOCAMENTOS, VELOCIDADES E ACELERAÇÕES ANGULARES ORIGINAIS

| Alfa (graus) | Beta (graus) | Valpha (graus/s) | Vbeta (graus/s) | Aalpha (graus/s <sup>2</sup> ) | Abeta (graus/s <sup>2</sup> ) |
|--------------|--------------|------------------|-----------------|--------------------------------|-------------------------------|
| 12,626       | -12,985      |                  |                 |                                |                               |
| 12,996       | -12,349      | 11,099           | 19,083          |                                |                               |
| 14,354       | -10,991      | 40,728           | 40,728          | 888,875                        | 649,341                       |
| 13,075       | -14,336      | -38,366          | -100,363        | -2.372,819                     | -4.232,731                    |
| 13,075       | -16,266      | 0,000            | -57,873         | 1.150,977                      | 1.274,704                     |
| 15,435       | -14,330      | 70,797           | 58,051          | 2.123,898                      | 3.477,713                     |
| 17,381       | -15,737      | 58,364           | -42,184         | -372,966                       | -3.007,050                    |
| 17,794       | -15,523      | 12,413           | 6,409           | -1.378,528                     | 1.457,799                     |
| 18,218       | -15,582      | 12,708           | -1,786          | 8,845                          | -245,858                      |
| 18,648       | -12,490      | 12,899           | 92,766          | 5,721                          | 2.836,562                     |
| 19,083       | -11,203      | 13,053           | 38,605          | 4,620                          | -1.624,824                    |
| 19,566       | -9,472       | 14,484           | 51,931          | 42,918                         | 399,765                       |
| 20,676       | -8,176       | 33,295           | 38,883          | 564,328                        | -391,435                      |
| 19,871       | -9,276       | -24,144          | -32,987         | -1.723,155                     | -2.156,084                    |
| 19,717       | -8,631       | -4,614           | 19,352          | 585,897                        | 1.570,154                     |
| 21,576       | -6,617       | 55,758           | 60,415          | 1.811,150                      | 1.231,882                     |
| 20,173       | -6,617       | -42,086          | 0,000           | -2.935,297                     | -1.812,439                    |
| 20,188       | -5,972       | 0,455            | 19,357          | 1.276,220                      | 580,721                       |
| 19,432       | -6,433       | -22,667          | -13,834         | -693,650                       | -995,732                      |
| 18,803       | -6,433       | -18,869          | 0,000           | 113,924                        | 415,011                       |
| 18,815       | -5,790       | 0,348            | 19,292          | 576,505                        | 578,773                       |
| 18,722       | -5,149       | -2,791           | 19,223          | -94,153                        | -2,083                        |
| 18,731       | -4,511       | 0,271            | 19,149          | 91,837                         | -2,215                        |
| 19,367       | -3,875       | 19,071           | 19,071          | 564,015                        | -2,344                        |
| 19,416       | -3,110       | 1,483            | 22,952          | -527,633                       | 116,437                       |
| 20,687       | -1,840       | 38,112           | 38,112          | 1.098,867                      | 454,797                       |
| 21,944       | -0,582       | 37,723           | 37,723          | -11,669                        | -11,669                       |
| 21,941       | 0,041        | -0,085           | 18,706          | -1.134,254                     | -570,506                      |
| 21,303       | -1,106       | -19,132          | -34,437         | -571,397                       | -1.594,292                    |
| 22,656       | 0,309        | 40,584           | 42,454          | 1.791,467                      | 2.306,726                     |
| 23,943       | 1,533        | 38,607           | 36,736          | -59,310                        | -171,533                      |
| 23,918       | 2,139        | -0,751           | 18,178          | -1.180,740                     | -556,742                      |
| 25,934       | 2,419        | 60,481           | 8,410           | 1.836,961                      | -293,060                      |
| 25,262       | 3,013        | -20,168          | 17,798          | -2.419,452                     | 281,663                       |
| 25,800       | 4,185        | 16,137           | 35,164          | 1.089,125                      | 520,965                       |
| 27,242       | 5,001        | 43,269           | 24,480          | 813,971                        | -320,530                      |
| 26,530       | 5,574        | -21,348          | 17,199          | -1.938,518                     | -218,430                      |
| 29,110       | 7,539        | 77,384           | 58,958          | 2.961,982                      | 1.252,779                     |
| 28,394       | 8,096        | -21,466          | 16,688          | -2.965,500                     | -1.268,108                    |
| 27,962       | 6,514        | -12,963          | -47,436         | 255,079                        | -1.923,709                    |

TABELA 9. COORDENADAS CARTESIANAS DOS MARCADORES (PIXELS) – VALORES FILTRADOS

| Ax (px) - f | Ay (px) - f | Bx (px) - f | By (px) - f | Cx (px) - f | Cy (px) - f | Dx (px) - f | Dy (px) - f |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 283,090     | 186,996     | 256,780     | 118,076     | 241,220     | 36,426      | 214,910     | 29,503      |
| 280,988     | 186,640     | 254,876     | 117,089     | 240,124     | 35,719      | 214,012     | 28,552      |
| 278,900     | 186,365     | 253,038     | 116,981     | 239,962     | 35,049      | 214,100     | 27,605      |
| 274,002     | 186,096     | 249,990     | 115,150     | 237,868     | 33,987      | 212,141     | 24,767      |
| 273,974     | 186,010     | 250,000     | 115,133     | 237,856     | 33,999      | 212,171     | 23,857      |
| 271,814     | 186,624     | 247,319     | 115,161     | 237,605     | 34,291      | 212,261     | 23,925      |
| 268,788     | 186,489     | 244,550     | 115,133     | 237,296     | 34,269      | 211,366     | 22,109      |
| 266,725     | 187,191     | 242,668     | 115,178     | 236,252     | 34,468      | 211,354     | 22,164      |
| 263,688     | 187,673     | 240,818     | 115,197     | 235,930     | 34,774      | 210,564     | 21,356      |
| 260,627     | 188,003     | 239,004     | 115,139     | 235,636     | 34,431      | 210,733     | 21,427      |
| 257,671     | 188,670     | 237,133     | 115,197     | 235,405     | 34,642      | 210,791     | 21,491      |
| 254,798     | 188,654     | 235,198     | 114,373     | 235,160     | 34,508      | 209,845     | 21,466      |
| 252,721     | 189,164     | 233,448     | 114,427     | 234,883     | 34,840      | 209,948     | 21,569      |
| 249,850     | 189,717     | 232,428     | 114,481     | 234,682     | 35,040      | 210,040     | 20,762      |
| 247,893     | 190,271     | 231,492     | 114,549     | 234,486     | 35,322      | 210,130     | 20,858      |
| 244,929     | 190,201     | 228,768     | 113,737     | 234,212     | 35,219      | 210,091     | 20,842      |
| 243,087     | 190,806     | 228,637     | 113,818     | 234,106     | 35,452      | 210,169     | 20,924      |
| 241,213     | 190,868     | 227,627     | 113,820     | 234,019     | 35,405      | 210,140     | 20,907      |
| 239,282     | 191,495     | 226,590     | 113,905     | 233,025     | 35,618      | 210,104     | 20,982      |
| 238,398     | 191,558     | 226,483     | 113,908     | 233,033     | 35,570      | 210,086     | 20,964      |
| 236,521     | 191,598     | 225,478     | 113,909     | 232,938     | 35,540      | 210,063     | 20,953      |
| 234,592     | 192,059     | 224,550     | 113,993     | 232,699     | 35,874      | 210,159     | 21,074      |
| 232,717     | 192,079     | 223,548     | 113,993     | 232,594     | 35,859      | 210,140     | 21,069      |
| 231,727     | 192,047     | 222,653     | 113,992     | 232,484     | 35,883      | 210,136     | 21,078      |
| 229,851     | 191,962     | 221,652     | 113,191     | 232,378     | 35,784      | 210,119     | 21,063      |
| 227,869     | 191,874     | 219,868     | 113,190     | 232,148     | 35,849      | 210,115     | 21,087      |
| 225,886     | 191,771     | 218,089     | 113,188     | 231,910     | 35,925      | 210,115     | 21,116      |
| 224,009     | 191,744     | 217,096     | 113,188     | 231,788     | 35,944      | 210,107     | 21,124      |
| 221,241     | 191,597     | 216,025     | 113,174     | 231,610     | 35,950      | 210,124     | 20,279      |
| 219,262     | 191,360     | 214,257     | 112,377     | 231,352     | 35,956      | 210,128     | 20,306      |
| 217,299     | 190,769     | 212,400     | 112,278     | 231,264     | 35,733      | 210,037     | 20,220      |
| 215,423     | 190,697     | 211,417     | 112,278     | 231,121     | 35,785      | 210,039     | 20,241      |
| 212,462     | 190,745     | 208,714     | 112,368     | 230,831     | 36,557      | 209,993     | 20,330      |
| 209,701     | 190,649     | 207,638     | 112,368     | 230,664     | 36,625      | 209,997     | 20,359      |
| 206,844     | 190,426     | 205,796     | 112,367     | 230,346     | 36,782      | 210,014     | 20,425      |
| 203,858     | 190,272     | 202,827     | 112,371     | 229,582     | 37,159      | 209,733     | 21,198      |
| 201,095     | 190,120     | 201,770     | 112,370     | 229,387     | 37,267      | 209,748     | 21,242      |
| 199,035     | 189,636     | 199,278     | 111,573     | 228,904     | 37,447      | 209,783     | 21,344      |
| 196,288     | 189,454     | 198,228     | 111,575     | 228,685     | 37,573      | 209,800     | 21,398      |
| 193,487     | 189,166     | 196,978     | 111,559     | 228,966     | 37,054      | 210,568     | 21,322      |
| 190,655     | 188,819     | 195,178     | 111,562     | 228,568     | 38,192      | 210,600     | 21,426      |
| 184,073     | 188,416     | 191,853     | 111,636     | 227,581     | 39,469      | 209,493     | 21,480      |

TABELA 10. COORDENADAS CARTESIANAS DOS MARCADORES (MM) – VALORES FILTRADOS

| Ax (mm) - f | Bx (mm) - f | Cx (mm) - f | Dx (mm) - f | Ay (mm) - f | By (mm) - f | Cy (mm) - f | Dy (mm) - f |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1.581,507   | 1.434,528   | 1.347,595   | 1.200,616   | 1.044,669   | 659,641     | 203,497     | 164,820     |
| 1.569,763   | 1.423,886   | 1.341,477   | 1.195,600   | 1.042,681   | 654,129     | 199,546     | 159,510     |
| 1.558,100   | 1.413,622   | 1.340,568   | 1.196,090   | 1.041,147   | 653,525     | 195,805     | 154,215     |
| 1.530,736   | 1.396,590   | 1.328,870   | 1.185,145   | 1.039,641   | 643,295     | 189,873     | 138,365     |
| 1.530,581   | 1.396,647   | 1.328,803   | 1.185,311   | 1.039,164   | 643,202     | 189,941     | 133,280     |
| 1.518,516   | 1.381,670   | 1.327,404   | 1.185,817   | 1.042,591   | 643,356     | 191,568     | 133,659     |
| 1.501,611   | 1.366,200   | 1.325,676   | 1.180,815   | 1.041,840   | 643,199     | 191,445     | 123,516     |
| 1.490,085   | 1.355,688   | 1.319,846   | 1.180,750   | 1.045,757   | 643,451     | 192,560     | 123,819     |
| 1.473,117   | 1.345,351   | 1.318,044   | 1.176,337   | 1.048,454   | 643,557     | 194,266     | 119,309     |
| 1.456,015   | 1.335,216   | 1.316,404   | 1.177,281   | 1.050,295   | 643,235     | 192,351     | 119,705     |
| 1.439,505   | 1.324,764   | 1.315,113   | 1.177,601   | 1.054,023   | 643,559     | 193,530     | 120,061     |
| 1.423,451   | 1.313,955   | 1.313,742   | 1.172,316   | 1.053,930   | 638,953     | 192,784     | 119,919     |
| 1.411,852   | 1.304,180   | 1.312,193   | 1.172,893   | 1.056,783   | 639,255     | 194,638     | 120,497     |
| 1.395,812   | 1.298,479   | 1.311,071   | 1.173,409   | 1.059,873   | 639,560     | 195,754     | 115,987     |
| 1.384,875   | 1.293,251   | 1.309,976   | 1.173,908   | 1.062,964   | 639,941     | 197,328     | 116,527     |
| 1.368,319   | 1.278,036   | 1.308,444   | 1.173,693   | 1.062,577   | 635,402     | 196,757     | 116,438     |
| 1.358,027   | 1.277,304   | 1.307,857   | 1.174,131   | 1.065,956   | 635,853     | 198,058     | 116,892     |
| 1.347,561   | 1.271,660   | 1.307,370   | 1.173,968   | 1.066,301   | 635,867     | 197,795     | 116,797     |
| 1.336,768   | 1.265,865   | 1.301,816   | 1.173,763   | 1.069,802   | 636,343     | 198,984     | 117,216     |
| 1.331,831   | 1.265,269   | 1.301,862   | 1.173,664   | 1.070,156   | 636,355     | 198,717     | 117,118     |
| 1.321,348   | 1.259,651   | 1.301,332   | 1.173,535   | 1.070,382   | 636,363     | 198,546     | 117,056     |
| 1.310,572   | 1.254,468   | 1.299,993   | 1.174,073   | 1.072,953   | 636,831     | 200,416     | 117,734     |
| 1.300,098   | 1.248,870   | 1.299,409   | 1.173,969   | 1.073,065   | 636,834     | 200,332     | 117,702     |
| 1.294,562   | 1.243,871   | 1.298,795   | 1.173,945   | 1.072,886   | 636,829     | 200,466     | 117,753     |
| 1.284,084   | 1.238,279   | 1.298,200   | 1.173,850   | 1.072,412   | 632,353     | 199,912     | 117,670     |
| 1.273,009   | 1.228,311   | 1.296,917   | 1.173,829   | 1.071,924   | 632,346     | 200,271     | 117,806     |
| 1.261,932   | 1.218,375   | 1.295,589   | 1.173,824   | 1.071,346   | 632,338     | 200,696     | 117,967     |
| 1.251,449   | 1.212,826   | 1.294,906   | 1.173,780   | 1.071,197   | 632,335     | 200,806     | 118,009     |
| 1.235,986   | 1.206,842   | 1.293,910   | 1.173,877   | 1.070,375   | 632,258     | 200,838     | 113,288     |
| 1.224,929   | 1.196,967   | 1.292,472   | 1.173,900   | 1.069,049   | 627,807     | 200,873     | 113,444     |
| 1.213,961   | 1.186,593   | 1.291,976   | 1.173,392   | 1.065,750   | 627,253     | 199,625     | 112,959     |
| 1.203,479   | 1.181,103   | 1.291,180   | 1.173,401   | 1.065,346   | 627,250     | 199,916     | 113,076     |
| 1.186,938   | 1.166,001   | 1.289,561   | 1.173,143   | 1.065,614   | 627,756     | 204,227     | 113,576     |
| 1.171,513   | 1.159,989   | 1.288,626   | 1.173,167   | 1.065,076   | 627,754     | 204,607     | 113,736     |
| 1.155,556   | 1.149,696   | 1.286,847   | 1.173,265   | 1.063,834   | 627,748     | 205,484     | 114,107     |
| 1.138,871   | 1.133,109   | 1.282,582   | 1.171,695   | 1.062,974   | 627,768     | 207,593     | 118,425     |
| 1.123,433   | 1.127,207   | 1.281,492   | 1.171,778   | 1.062,122   | 627,767     | 208,197     | 118,673     |
| 1.111,929   | 1.113,283   | 1.278,791   | 1.171,975   | 1.059,417   | 623,313     | 209,201     | 119,241     |
| 1.096,579   | 1.107,418   | 1.277,569   | 1.172,066   | 1.058,402   | 623,323     | 209,906     | 119,542     |
| 1.080,935   | 1.100,437   | 1.279,140   | 1.176,360   | 1.056,793   | 623,233     | 212,031     | 119,116     |
| 1.065,109   | 1.090,378   | 1.276,913   | 1.176,538   | 1.054,856   | 623,254     | 213,363     | 119,700     |

TABELA 11. DIMENSÕES GEOMÉTRICAS – VALORES FILTRADOS

| distAB (px) - f | distBC (px) - f | distAC (px) - f | distCD (px) - f | distBD (px) - f | F (mm) - f | T (mm) - f | P (mm) - f |
|-----------------|-----------------|-----------------|-----------------|-----------------|------------|------------|------------|
| 73,771          | 83,119          | 156,283         | 27,205          | 97,971          | 412,128    | 464,354    | 151,983    |
| 74,291          | 82,697          | 156,355         | 27,078          | 97,512          | 415,034    | 461,992    | 151,271    |
| 74,047          | 82,969          | 156,246         | 26,912          | 97,490          | 413,672    | 463,513    | 150,345    |
| 74,899          | 82,063          | 156,342         | 27,329          | 97,987          | 418,432    | 458,451    | 152,676    |
| 74,822          | 82,038          | 156,243         | 27,615          | 98,805          | 418,000    | 458,311    | 154,274    |
| 75,545          | 81,451          | 156,127         | 27,382          | 97,739          | 422,038    | 455,035    | 152,972    |
| 75,361          | 81,189          | 155,444         | 28,639          | 98,765          | 421,012    | 453,567    | 159,997    |
| 75,925          | 80,964          | 155,733         | 27,773          | 98,144          | 424,162    | 452,313    | 155,154    |
| 75,999          | 80,571          | 155,399         | 28,696          | 98,597          | 424,577    | 450,120    | 160,311    |
| 76,005          | 80,778          | 155,592         | 28,094          | 97,883          | 424,606    | 451,276    | 156,948    |
| 76,290          | 80,574          | 155,629         | 27,907          | 97,338          | 426,200    | 450,133    | 155,907    |
| 76,823          | 79,864          | 155,391         | 28,478          | 96,304          | 429,180    | 446,169    | 159,093    |
| 77,183          | 79,600          | 155,352         | 28,246          | 95,785          | 431,188    | 444,690    | 157,801    |
| 77,227          | 79,473          | 155,419         | 28,479          | 96,356          | 431,436    | 443,985    | 159,102    |
| 77,477          | 79,284          | 155,528         | 28,327          | 96,096          | 432,832    | 442,929    | 158,251    |
| 78,153          | 78,706          | 155,352         | 28,080          | 94,754          | 436,612    | 439,698    | 156,873    |
| 78,333          | 78,556          | 155,613         | 28,001          | 94,712          | 437,613    | 438,860    | 156,431    |
| 78,236          | 78,675          | 155,629         | 27,936          | 94,545          | 437,074    | 439,525    | 156,067    |
| 78,620          | 78,551          | 156,002         | 27,196          | 94,375          | 439,220    | 438,834    | 151,933    |
| 78,559          | 78,611          | 156,080         | 27,202          | 94,379          | 438,877    | 439,166    | 151,964    |
| 78,470          | 78,724          | 156,100         | 27,131          | 94,225          | 438,382    | 439,797    | 151,567    |
| 78,709          | 78,542          | 156,196         | 26,964          | 94,026          | 439,716    | 438,783    | 150,639    |
| 78,622          | 78,656          | 156,219         | 26,888          | 93,887          | 439,229    | 439,418    | 150,210    |
| 78,580          | 78,725          | 156,165         | 26,808          | 93,754          | 438,993    | 439,805    | 149,763    |
| 79,196          | 78,146          | 156,198         | 26,686          | 92,847          | 442,437    | 436,573    | 149,086    |
| 79,090          | 78,310          | 156,084         | 26,521          | 92,618          | 441,844    | 437,487    | 148,160    |
| 78,968          | 78,490          | 155,963         | 26,351          | 92,417          | 441,164    | 438,494    | 147,210    |
| 78,860          | 78,629          | 155,994         | 26,263          | 92,329          | 440,558    | 439,267    | 146,720    |
| 78,596          | 78,781          | 155,992         | 26,594          | 93,083          | 439,085    | 440,119    | 148,568    |
| 79,141          | 78,310          | 155,873         | 26,370          | 92,164          | 442,127    | 437,485    | 147,321    |
| 78,644          | 78,836          | 155,664         | 26,291          | 92,089          | 439,349    | 440,423    | 146,878    |
| 78,521          | 78,990          | 155,705         | 26,193          | 92,047          | 438,667    | 441,283    | 146,332    |
| 78,466          | 78,972          | 155,279         | 26,411          | 92,047          | 438,358    | 441,185    | 147,549    |
| 78,308          | 79,166          | 155,444         | 26,301          | 92,039          | 437,474    | 442,268    | 146,930    |
| 78,066          | 79,472          | 155,432         | 26,094          | 92,038          | 436,126    | 443,978    | 145,776    |
| 77,909          | 79,829          | 155,259         | 25,470          | 91,434          | 435,244    | 445,970    | 142,292    |
| 77,752          | 80,020          | 155,449         | 25,347          | 91,476          | 434,371    | 447,038    | 141,604    |
| 78,063          | 79,827          | 155,092         | 24,998          | 90,838          | 436,106    | 445,962    | 139,651    |
| 77,903          | 80,024          | 155,298         | 24,865          | 90,916          | 435,214    | 447,063    | 138,911    |
| 77,686          | 80,255          | 155,319         | 24,801          | 91,265          | 433,998    | 448,354    | 138,553    |
| 77,389          | 80,611          | 155,325         | 24,575          | 91,446          | 432,341    | 450,340    | 137,288    |
| 77,173          | 80,527          | 155,172         | 25,510          | 91,866          | 431,134    | 449,873    | 142,515    |

TABELA 12. VELOCIDADES LINEARES DOS MARCADORES – VALORES FILTRADOS

| Vax (mm/s) | Vbx (mm/s) | Vcx (mm/s) | Vdx (mm/s) | Vay (mm/s) | Vby (mm/s) | Vcy (mm/s) | Vdy (mm/s) |
|------------|------------|------------|------------|------------|------------|------------|------------|
| -352,304   | -319,250   | -183,540   | -150,489   | -59,635    | -165,351   | -118,521   | -159,284   |
| -349,910   | -307,929   | -27,266    | 14,715     | -46,014    | -18,114    | -112,231   | -158,838   |
| -820,916   | -510,959   | -350,946   | -328,352   | -45,186    | -306,915   | -177,967   | -475,519   |
| -4,644     | 1,711      | -2,028     | 4,959      | -14,324    | -2,772     | 2,044      | -152,546   |
| -361,935   | -449,301   | -41,948    | 15,198     | 102,820    | 4,602      | 48,818     | 11,358     |
| -507,151   | -464,109   | -51,860    | -150,065   | -22,540    | -4,713     | -3,677     | -304,264   |
| -345,793   | -315,350   | -174,904   | -1,941     | 117,530    | 7,561      | 33,440     | 9,064      |
| -509,023   | -310,105   | -54,044    | -132,416   | 80,899     | 3,188      | 51,190     | -135,276   |
| -513,064   | -304,052   | -49,198    | 28,325     | 55,236     | -9,660     | -57,448    | 11,874     |
| -495,300   | -313,571   | -38,733    | 9,618      | 111,843    | 9,726      | 35,348     | 10,681     |
| -481,634   | -324,250   | -41,137    | -158,568   | -2,785     | -138,176   | -22,377    | -4,261     |
| -347,980   | -293,270   | -46,467    | 17,326     | 85,584     | 9,065      | 55,613     | 17,337     |
| -481,181   | -171,022   | -33,667    | 15,482     | 92,701     | 9,126      | 33,477     | -135,305   |
| -328,106   | -156,824   | -32,826    | 14,963     | 92,724     | 11,440     | 47,221     | 16,212     |
| -496,694   | -456,462   | -45,968    | -6,465     | -11,608    | -136,177   | -17,130    | -2,684     |
| -308,763   | -21,970    | -17,612    | 13,152     | 101,368    | 13,541     | 39,056     | 13,632     |
| -313,983   | -169,296   | -14,615    | -4,900     | 10,342     | 0,419      | -7,900     | -2,860     |
| -323,764   | -173,873   | -166,606   | -6,148     | 105,043    | 14,292     | 35,678     | 12,586     |
| -148,121   | -17,884    | 1,366      | -2,959     | 10,613     | 0,359      | -8,026     | -2,945     |
| -314,478   | -168,526   | -15,912    | -3,877     | 6,785      | 0,228      | -5,135     | -1,878     |
| -323,284   | -155,504   | -40,150    | 16,147     | 77,124     | 14,027     | 56,106     | 20,339     |
| -314,241   | -167,917   | -17,518    | -3,120     | 3,368      | 0,095      | -2,518     | -0,945     |
| -166,058   | -149,970   | -18,441    | -0,726     | -5,386     | -0,152     | 4,027      | 1,512      |
| -314,348   | -167,764   | -17,830    | -2,852     | -14,201    | -134,282   | -16,634    | -2,481     |
| -332,239   | -299,037   | -38,483    | -0,629     | -14,661    | -0,202     | 10,791     | 4,072      |
| -332,316   | -298,089   | -39,849    | -0,141     | -17,326    | -0,247     | 12,736     | 4,837      |
| -314,494   | -166,483   | -20,489    | -1,327     | -4,480     | -0,067     | 3,291      | 1,255      |
| -463,899   | -179,520   | -29,898    | 2,926      | -24,643    | -2,307     | 0,959      | -141,607   |
| -331,697   | -296,229   | -43,127    | 0,662      | -39,792    | -133,544   | 1,075      | 4,662      |
| -329,048   | -311,241   | -14,878    | -15,224    | -98,979    | -16,611    | -37,465    | -14,541    |
| -314,471   | -164,698   | -23,888    | 0,263      | -12,117    | -0,112     | 8,737      | 3,493      |
| -496,231   | -453,040   | -48,564    | -7,751     | 8,058      | 15,193     | 129,326    | 15,020     |
| -462,730   | -180,361   | -28,036    | 0,736      | -16,134    | -0,063     | 11,413     | 4,784      |
| -478,729   | -308,809   | -53,392    | 2,941      | -37,271    | -0,183     | 26,317     | 11,138     |
| -500,533   | -497,597   | -127,944   | -47,111    | -25,814    | 0,611      | 63,273     | 129,527    |
| -463,142   | -177,050   | -32,693    | 2,493      | -25,543    | -0,027     | 18,114     | 7,457      |
| -345,128   | -417,736   | -81,034    | 5,910      | -81,142    | -133,621   | 30,113     | 17,051     |
| -460,497   | -175,953   | -36,672    | 2,734      | -30,458    | 0,297      | 21,139     | 9,020      |
| -469,325   | -209,423   | 47,145     | 128,808    | -48,285    | -2,706     | 63,767     | -12,773    |
| -474,762   | -301,771   | -66,800    | 5,345      | -58,098    | 0,626      | 39,951     | 17,518     |
| -1.103,139 | -557,249   | -165,286   | -185,497   | -67,629    | 12,305     | 213,964    | 8,957      |

TABELA 13. VELOCIDADES LINEARES DOS MARCADORES – VALORES FILTRADOS

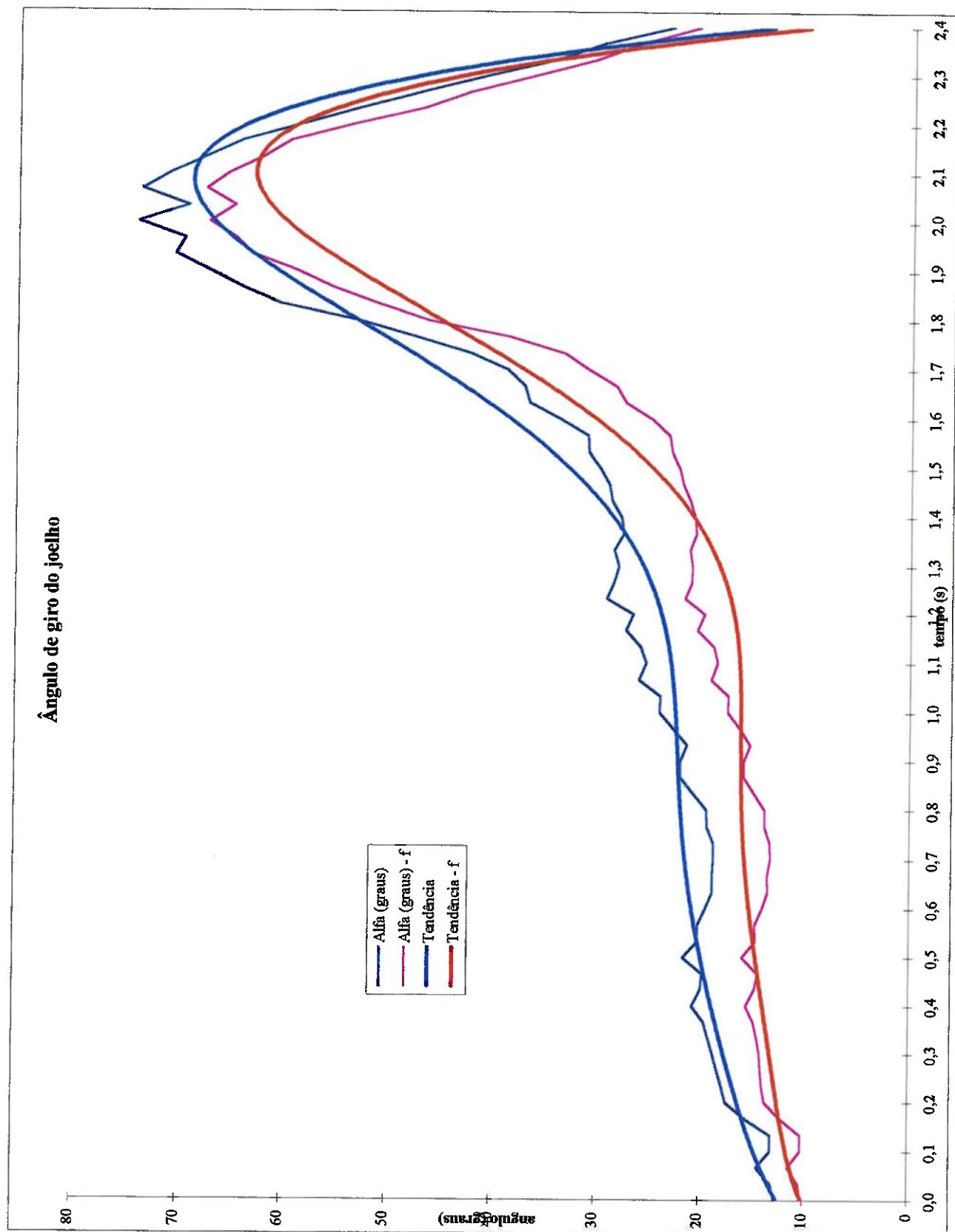
| Aax<br>(mm/s <sup>2</sup> ) | Abx<br>(mm/s <sup>2</sup> ) | Acx<br>(mm/s <sup>2</sup> ) | Adx<br>(mm/s <sup>2</sup> ) | Aay<br>(mm/s <sup>2</sup> ) | Aby<br>(mm/s <sup>2</sup> ) | Acy<br>(mm/s <sup>2</sup> ) | Ady<br>(mm/s <sup>2</sup> ) |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
|                             |                             |                             |                             |                             |                             |                             |                             |
|                             |                             |                             |                             |                             |                             |                             |                             |
| 71,809                      | 339,637                     | 4.688,221                   | 4.956,124                   | 408,610                     | 4.417,125                   | 188,713                     | 13,374                      |
| -14.130,171                 | -6.090,893                  | -9.710,397                  | -10.292,008                 | 24,863                      | -8.664,049                  | -1.972,096                  | -9.500,430                  |
| 24.488,165                  | 15.380,100                  | 10.467,548                  | 9.999,322                   | 925,854                     | 9.124,296                   | 5.400,337                   | 9.689,209                   |
| -10.718,734                 | -13.530,374                 | -1.197,603                  | 307,187                     | 3.514,324                   | 221,219                     | 1.403,211                   | 4.917,102                   |
| -4.356,478                  | -444,213                    | -297,367                    | -4.957,889                  | -3.760,823                  | -279,453                    | -1.574,839                  | -9.468,639                  |
| 4.840,733                   | 4.462,743                   | -3.691,312                  | 4.443,707                   | 4.202,120                   | 368,221                     | 1.113,521                   | 9.399,831                   |
| -4.896,890                  | 157,349                     | 3.625,793                   | -3.914,256                  | -1.098,945                  | -131,194                    | 532,493                     | -4.330,202                  |
| -121,218                    | 181,594                     | 145,383                     | 4.822,240                   | -769,882                    | -385,441                    | -3.259,146                  | 4.414,505                   |
| 532,896                     | -285,551                    | 313,944                     | -561,208                    | 1.698,199                   | 581,576                     | 2.783,886                   | -35,799                     |
| 409,993                     | -320,380                    | -72,121                     | -5.045,576                  | -3.438,830                  | -4.437,035                  | -1.731,771                  | -448,245                    |
| 4.009,626                   | 929,388                     | -159,883                    | 5.276,811                   | 2.651,068                   | 4.417,205                   | 2.339,703                   | 647,940                     |
| -3.996,045                  | 3.667,450                   | 383,983                     | -55,312                     | 213,506                     | 1,840                       | -664,054                    | -4.579,270                  |
| 4.592,242                   | 425,951                     | 25,240                      | -15,577                     | 0,699                       | 69,431                      | 412,311                     | 4.545,503                   |
| -5.057,628                  | -8.989,145                  | -394,265                    | -642,836                    | -3.129,959                  | -4.428,518                  | -1.930,545                  | -566,864                    |
| 5.637,937                   | 13.034,755                  | 850,676                     | 588,520                     | 3.389,269                   | 4.491,548                   | 1.685,584                   | 489,479                     |
| -156,590                    | -4.419,774                  | 89,915                      | -541,569                    | -2.730,781                  | -393,657                    | -1.408,666                  | -494,784                    |
| -293,450                    | -137,328                    | -4.559,712                  | -37,438                     | 2.841,028                   | 416,172                     | 1.307,318                   | 463,384                     |
| 5.269,294                   | 4.679,698                   | 5.039,135                   | 95,666                      | -2.832,893                  | -417,972                    | -1.311,094                  | -465,928                    |
| -4.990,721                  | -4.519,277                  | -518,320                    | -27,538                     | -114,848                    | -3,952                      | 86,712                      | 32,028                      |
| -264,153                    | 390,660                     | -727,155                    | 600,717                     | 2.110,183                   | 413,980                     | 1.837,237                   | 666,508                     |
| 271,287                     | -372,399                    | 678,977                     | -578,011                    | -2.212,683                  | -417,967                    | -1.758,711                  | -638,537                    |
| 4.445,467                   | 538,416                     | -27,704                     | 71,814                      | -262,614                    | -7,406                      | 196,326                     | 73,720                      |
| -4.448,690                  | -533,811                    | 18,342                      | -63,759                     | -264,454                    | -4.023,895                  | -619,803                    | -119,801                    |
| -536,737                    | -3.938,204                  | -619,592                    | 66,675                      | -13,812                     | 4.022,397                   | 822,731                     | 196,607                     |
| -2,298                      | 28,463                      | -40,973                     | 14,646                      | -79,939                     | -1,337                      | 58,359                      | 22,937                      |
| 534,655                     | 3.948,169                   | 580,777                     | -35,593                     | 385,366                     | 5,405                       | -283,344                    | -107,447                    |
| -4.482,141                  | -391,113                    | -282,253                    | 127,584                     | -604,865                    | -67,203                     | -69,969                     | -4.285,855                  |
| 3.966,049                   | -3.501,282                  | -396,880                    | -67,897                     | -454,490                    | -3.937,128                  | 3,494                       | 4.388,058                   |
| 79,482                      | -450,347                    | 847,458                     | -476,583                    | -1.775,609                  | 3.507,989                   | -1.156,218                  | -576,106                    |
| 437,309                     | 4.396,299                   | -270,292                    | 464,611                     | 2.605,877                   | 494,965                     | 1.386,050                   | 541,036                     |
| -5.452,813                  | -8.650,282                  | -740,268                    | -240,436                    | 605,242                     | 459,166                     | 3.617,683                   | 345,806                     |
| 1.005,034                   | 8.180,372                   | 615,831                     | 254,630                     | -725,767                    | -457,673                    | -3.537,397                  | -307,086                    |
| -479,961                    | -3.853,428                  | -760,681                    | 66,137                      | -634,098                    | -3,605                      | 447,124                     | 190,634                     |
| -654,114                    | -5.663,635                  | -2.236,545                  | -1.501,567                  | 343,709                     | 23,822                      | 1.108,684                   | 3.551,661                   |
| 1.121,722                   | 9.616,409                   | 2.857,515                   | 1.488,137                   | 8,130                       | -19,141                     | -1.354,782                  | -3.662,110                  |
| 3.540,409                   | -7.220,589                  | -1.450,237                  | 102,499                     | -1.667,972                  | -4.007,826                  | 359,995                     | 287,834                     |
| -3.461,078                  | 7.253,502                   | 1.330,864                   | -95,284                     | 1.520,517                   | 4.017,530                   | -269,221                    | -240,939                    |
| -264,836                    | -1.004,108                  | 2.514,504                   | 3.782,222                   | -534,816                    | -90,065                     | 1.278,835                   | -653,792                    |
| -163,106                    | -2.770,446                  | -3.418,346                  | -3.703,892                  | -294,380                    | 99,960                      | -714,494                    | 908,744                     |

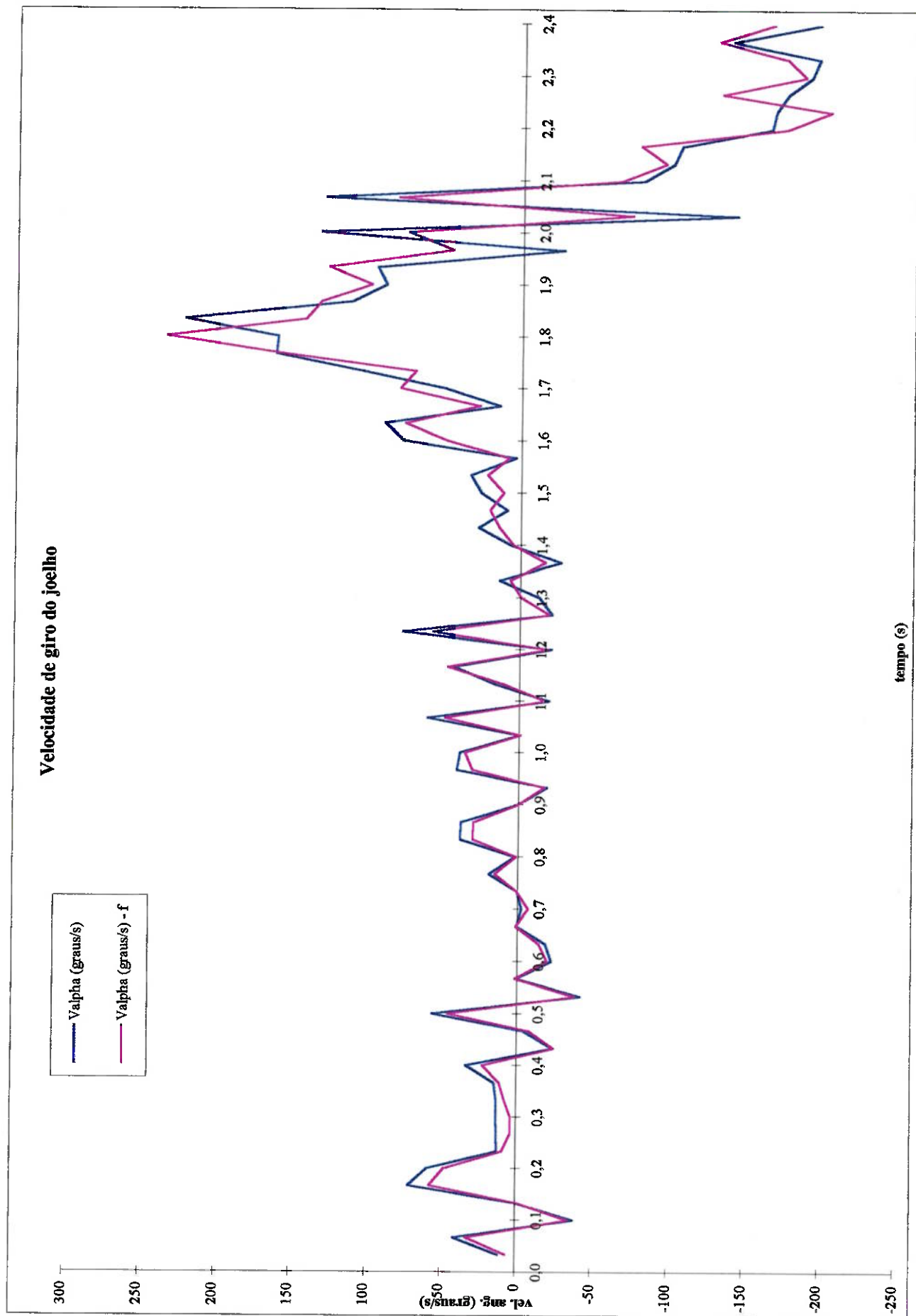
TABELA 14. DESLOCAMENTOS, VELOCIDADES E ACELERAÇÕES ANGULARES FILTRADOS

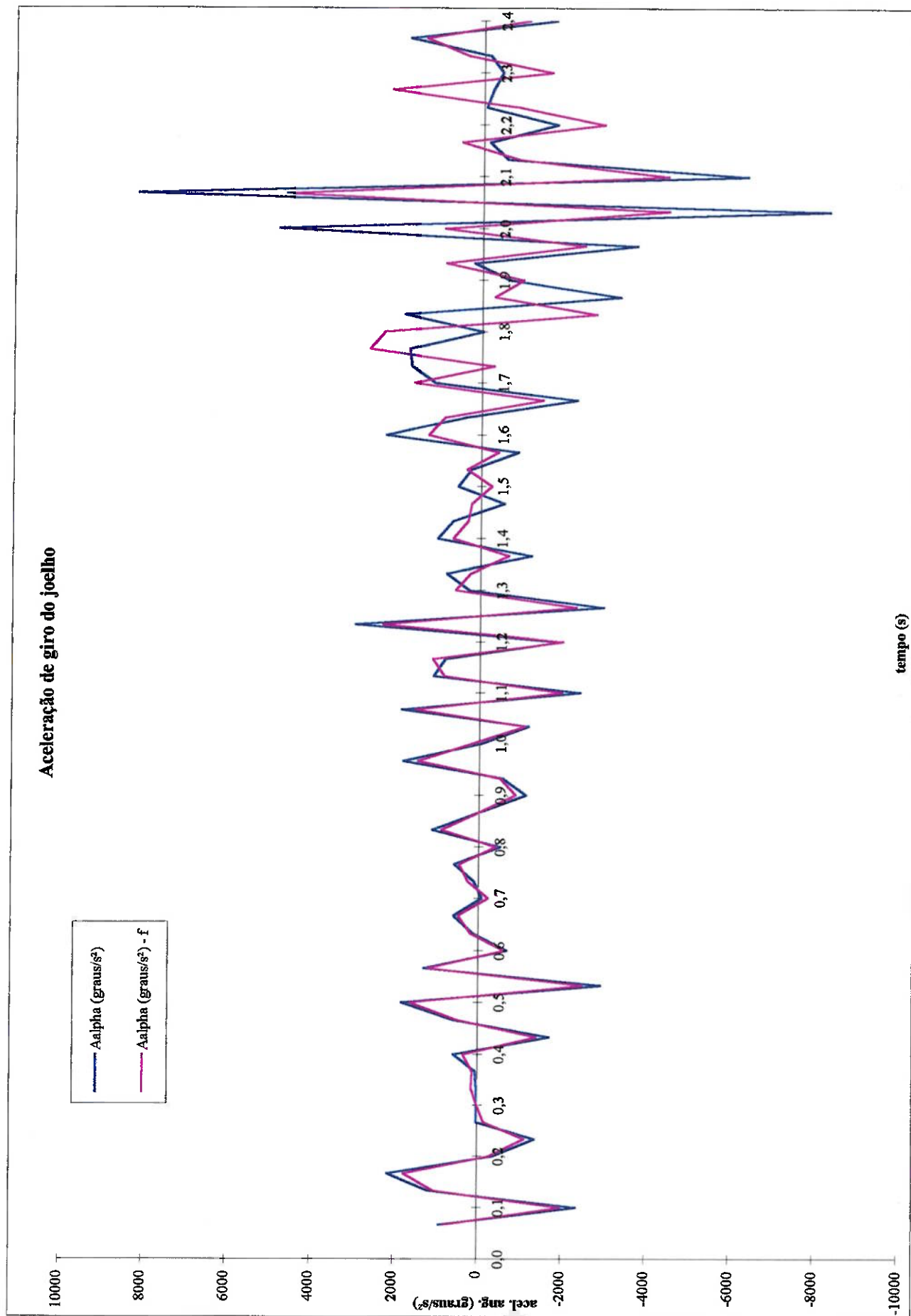
| Alfa (graus) | Beta (graus) | Valpha (graus/s) | Vbeta (graus/s) | Aalpha (graus/s <sup>2</sup> ) | Abeta (graus/s <sup>2</sup> ) |
|--------------|--------------|------------------|-----------------|--------------------------------|-------------------------------|
| 10,104       | -25,533      |                  |                 |                                |                               |
| 10,303       | -25,622      | 5,978            | -2,671          |                                |                               |
| 11,374       | -25,127      | 32,129           | 14,851          | 784,541                        | 525,678                       |
| 10,204       | -28,211      | -35,086          | -92,519         | -2.016,470                     | -3.221,109                    |
| 10,175       | -30,060      | -0,869           | -55,483         | 1.026,525                      | 1.111,074                     |
| 12,071       | -29,094      | 56,873           | 28,998          | 1.732,244                      | 2.534,431                     |
| 13,636       | -30,249      | 46,942           | -34,657         | -297,914                       | -1.909,650                    |
| 13,928       | -30,844      | 8,755            | -17,844         | -1.145,604                     | 504,407                       |
| 14,035       | -31,355      | 3,225            | -15,332         | -165,924                       | 75,332                        |
| 14,140       | -29,961      | 3,135            | 41,805          | -2,699                         | 1.714,132                     |
| 14,389       | -29,343      | 7,491            | 18,558          | 130,696                        | -697,404                      |
| 14,754       | -27,286      | 10,930           | 61,717          | 103,159                        | 1.294,750                     |
| 15,493       | -26,991      | 22,175           | 8,835           | 337,350                        | -1.586,451                    |
| 14,663       | -28,465      | -24,881          | -44,205         | -1.411,688                     | -1.591,216                    |
| 14,385       | -28,539      | -8,352           | -2,224          | 495,874                        | 1.259,436                     |
| 15,899       | -26,832      | 45,426           | 51,217          | 1.613,358                      | 1.603,234                     |
| 14,622       | -27,264      | -38,321          | -12,970         | -2.512,419                     | -1.925,615                    |
| 14,661       | -26,605      | 1,158            | 19,771          | 1.184,375                      | 982,233                       |
| 13,989       | -27,861      | -20,140          | -37,679         | -638,939                       | -1.723,494                    |
| 13,503       | -27,697      | -14,583          | 4,905           | 166,711                        | 1.277,525                     |
| 13,529       | -27,086      | 0,770            | 18,351          | 460,579                        | 403,384                       |
| 13,286       | -27,335      | -7,286           | -7,468          | -241,684                       | -774,570                      |
| 13,302       | -26,769      | 0,484            | 16,957          | 233,097                        | 732,737                       |
| 13,805       | -26,351      | 15,078           | 12,553          | 437,841                        | -132,128                      |
| 13,831       | -25,591      | 0,800            | 22,807          | -428,346                       | 307,630                       |
| 14,828       | -24,799      | 29,912           | 23,758          | 873,365                        | 28,535                        |
| 15,808       | -24,051      | 29,396           | 22,437          | -15,504                        | -39,627                       |
| 15,799       | -23,585      | -0,276           | 13,960          | -890,157                       | -254,327                      |
| 15,216       | -24,696      | -17,498          | -33,326         | -516,643                       | -1.418,583                    |
| 16,235       | -23,794      | 30,591           | 27,074          | 1.442,650                      | 1.812,020                     |
| 17,415       | -22,317      | 35,397           | 44,313          | 144,196                        | 517,150                       |
| 17,369       | -21,957      | -1,400           | 10,789          | -1.103,926                     | -1.005,698                    |
| 19,001       | -21,643      | 48,984           | 9,438           | 1.511,539                      | -40,526                       |
| 18,419       | -21,295      | -17,481          | 10,428          | -1.993,956                     | 29,683                        |
| 18,764       | -20,823      | 10,346           | 14,151          | 834,799                        | 111,703                       |
| 20,341       | -19,222      | 47,322           | 48,045          | 1.109,277                      | 1.016,813                     |
| 19,692       | -19,024      | -19,481          | 5,929           | -2.004,073                     | -1.263,484                    |
| 21,607       | -18,319      | 57,466           | 21,161          | 2.308,393                      | 456,973                       |
| 20,944       | -18,210      | -19,909          | 3,273           | -2.321,223                     | -536,646                      |
| 20,914       | -18,625      | -0,899           | -12,458         | 570,294                        | -471,938                      |

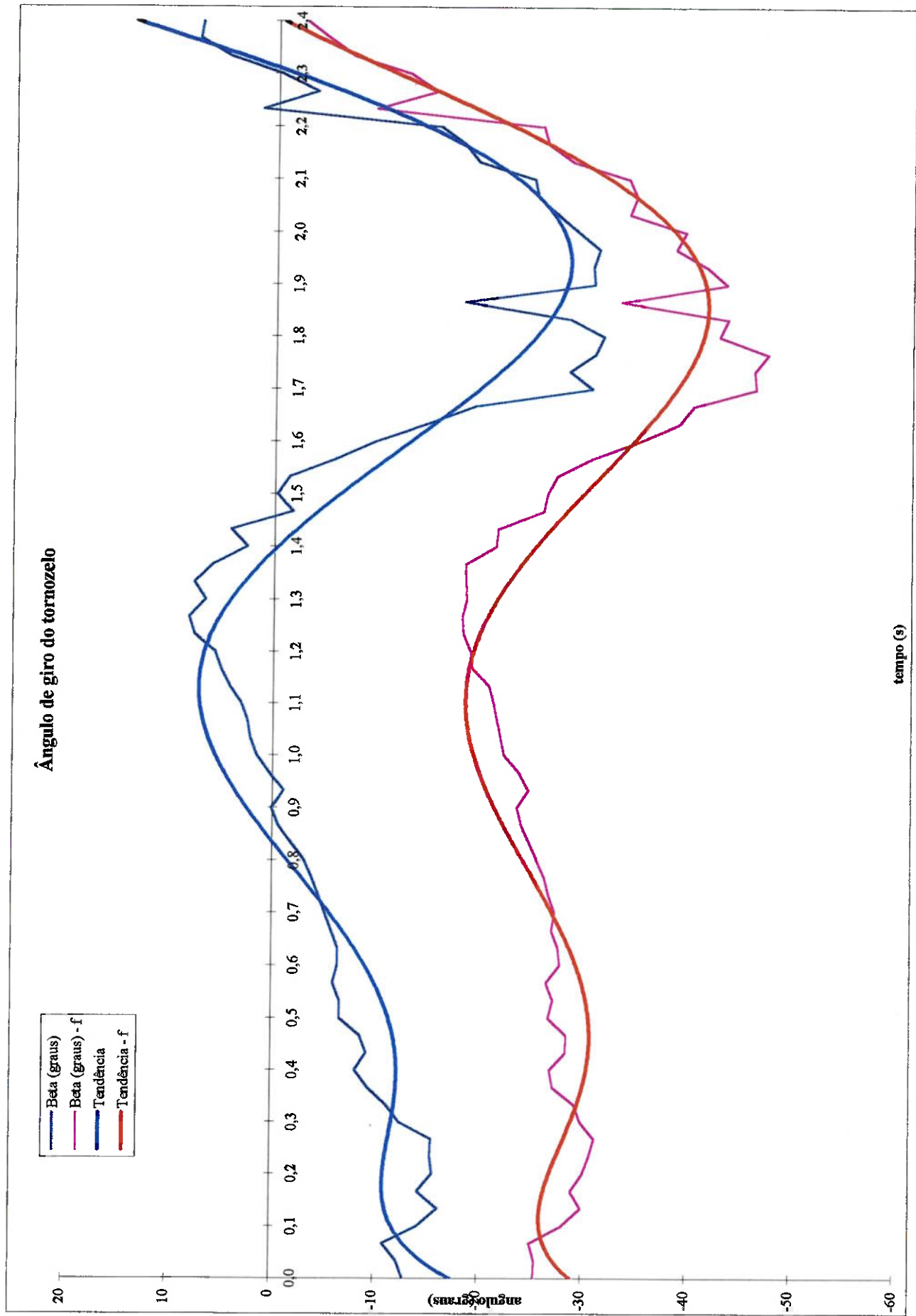
## ANEXO V

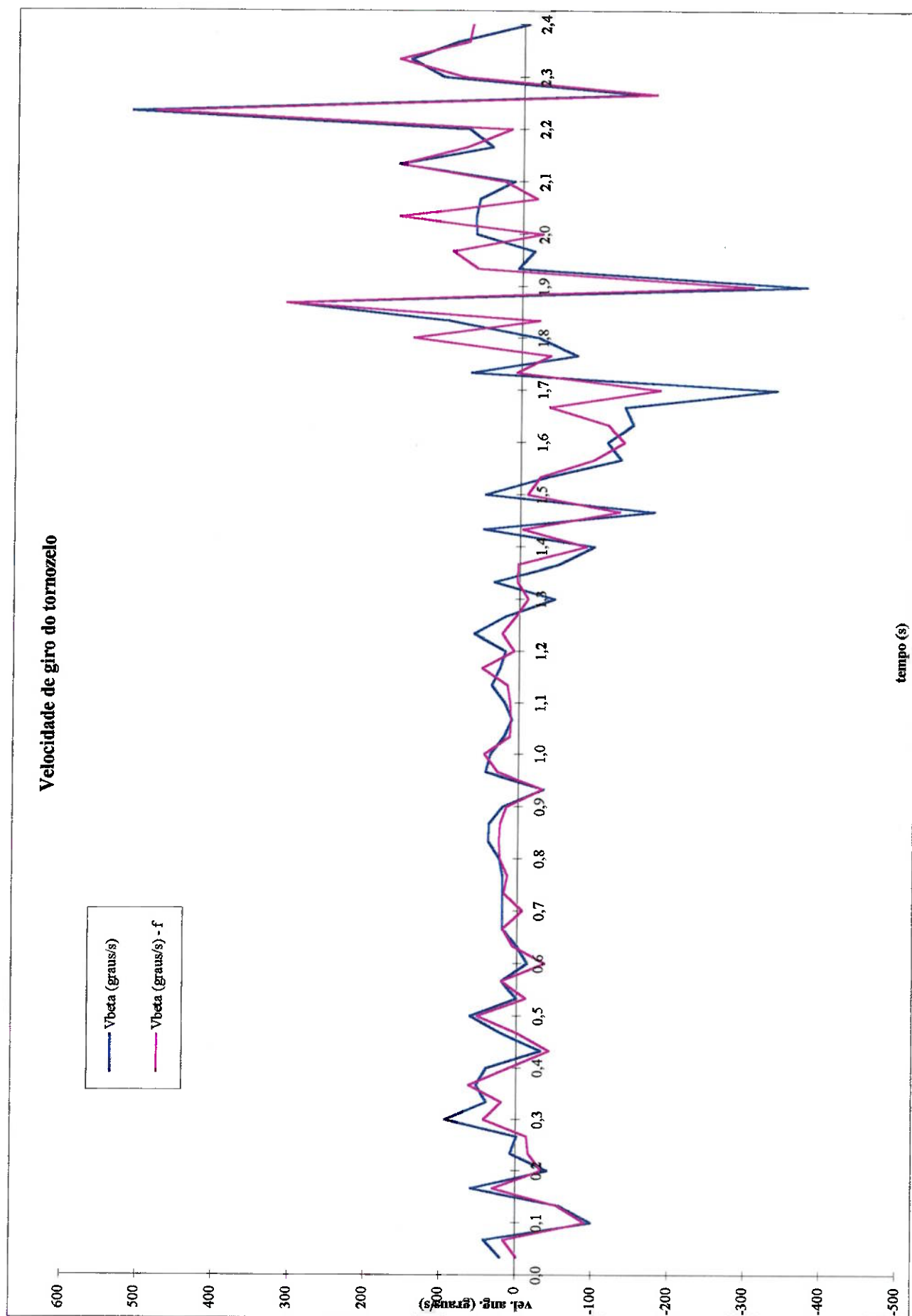
## GRÁFICOS RESULTADOS DO PROCESSAMENTO

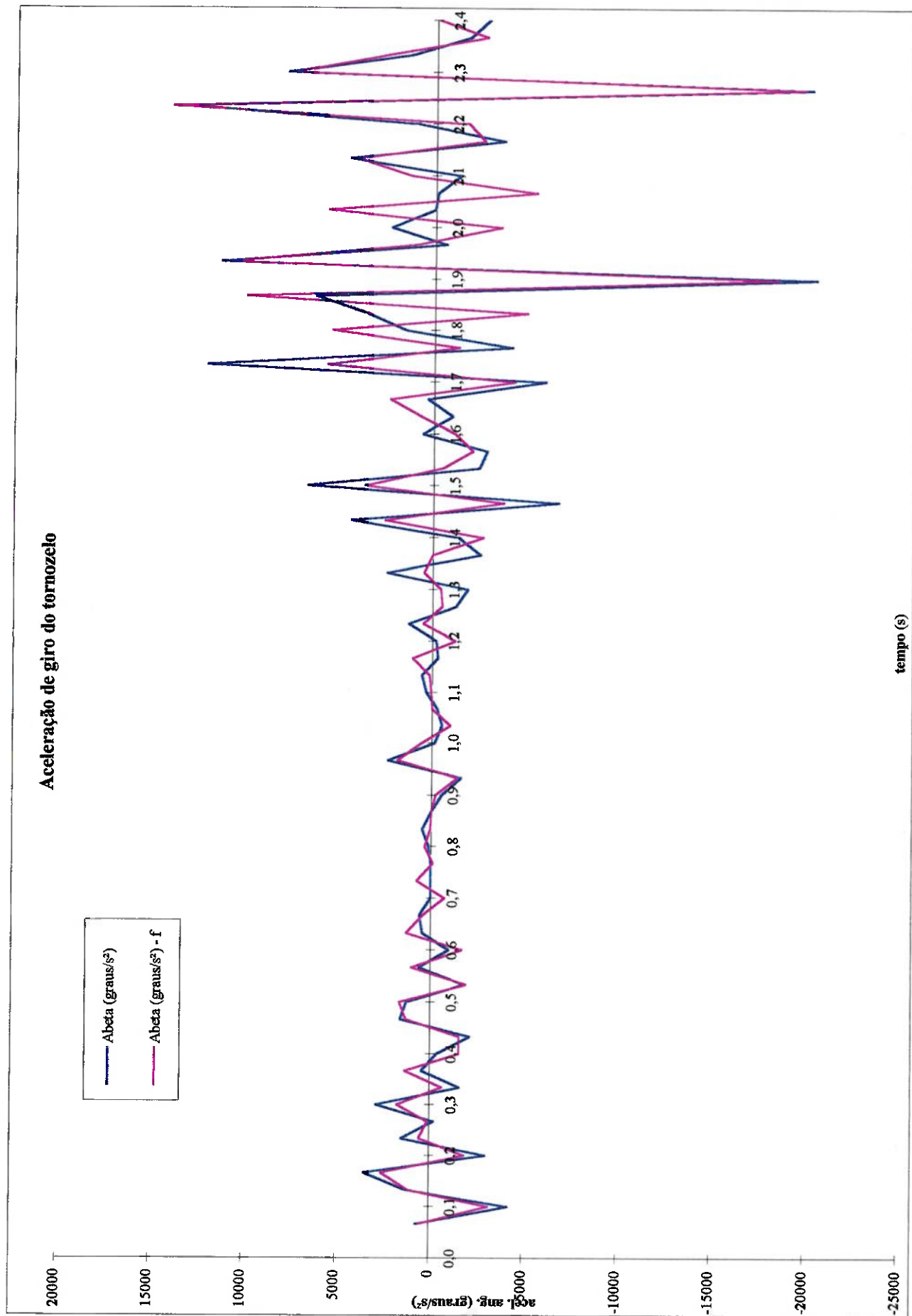


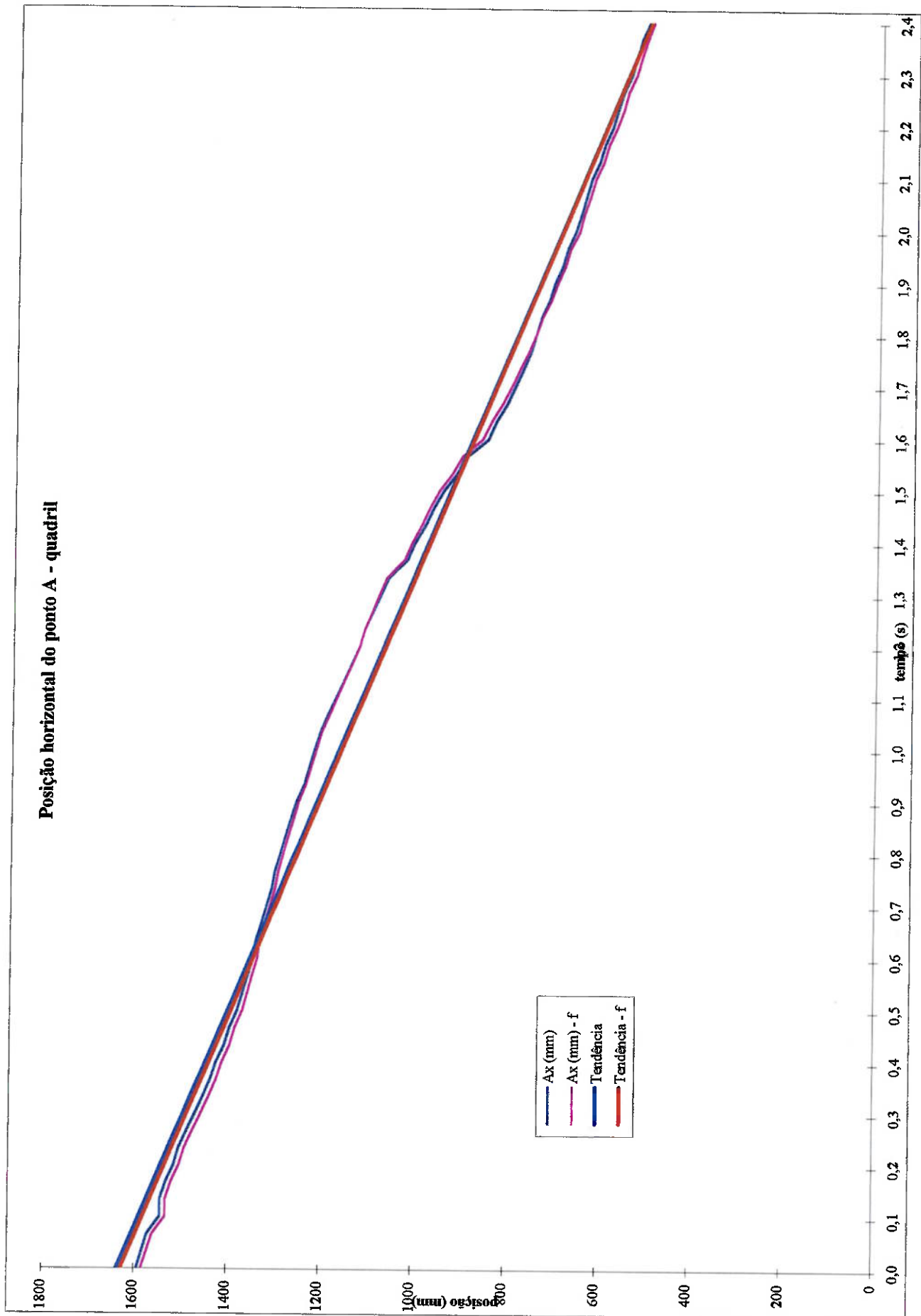


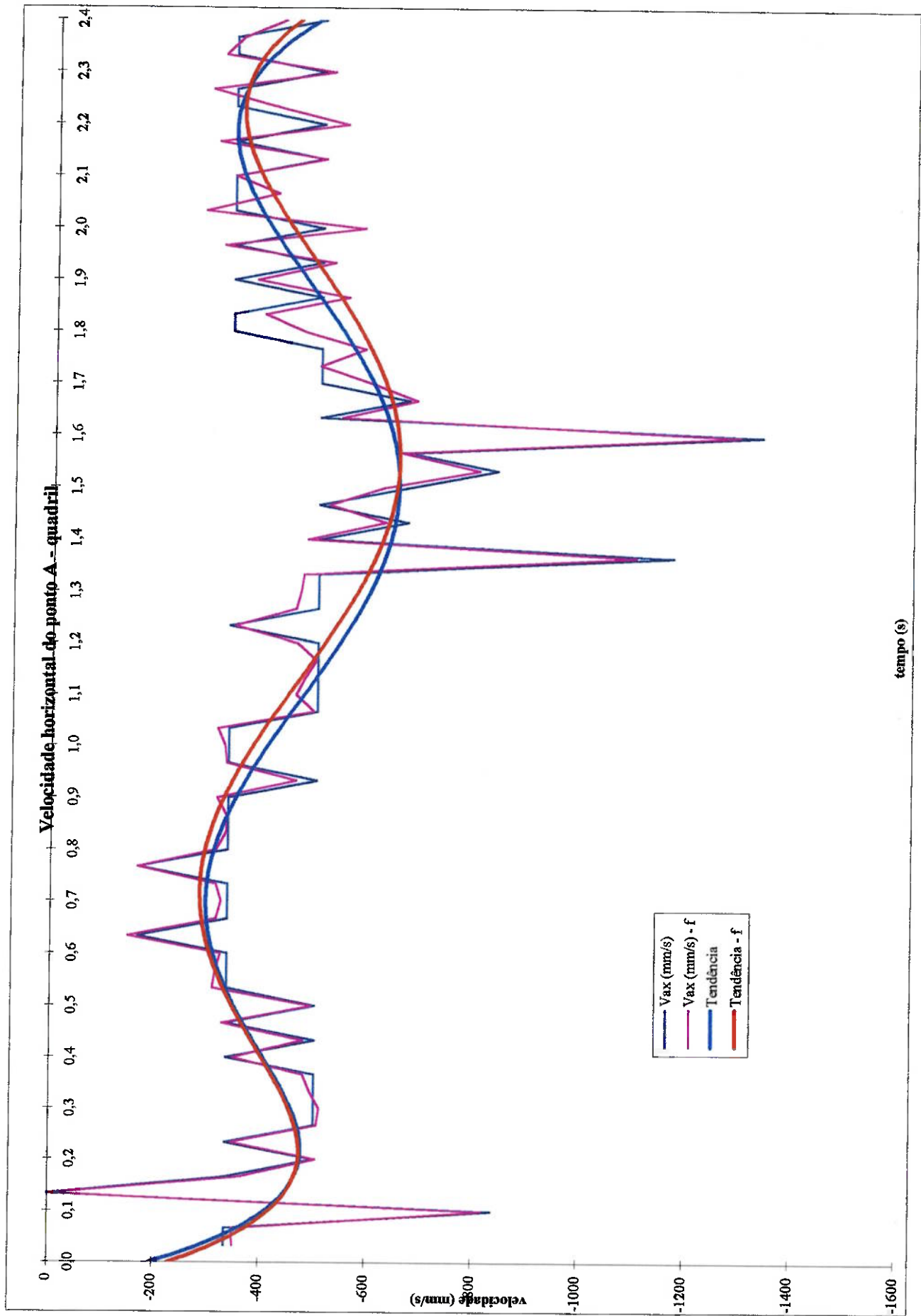




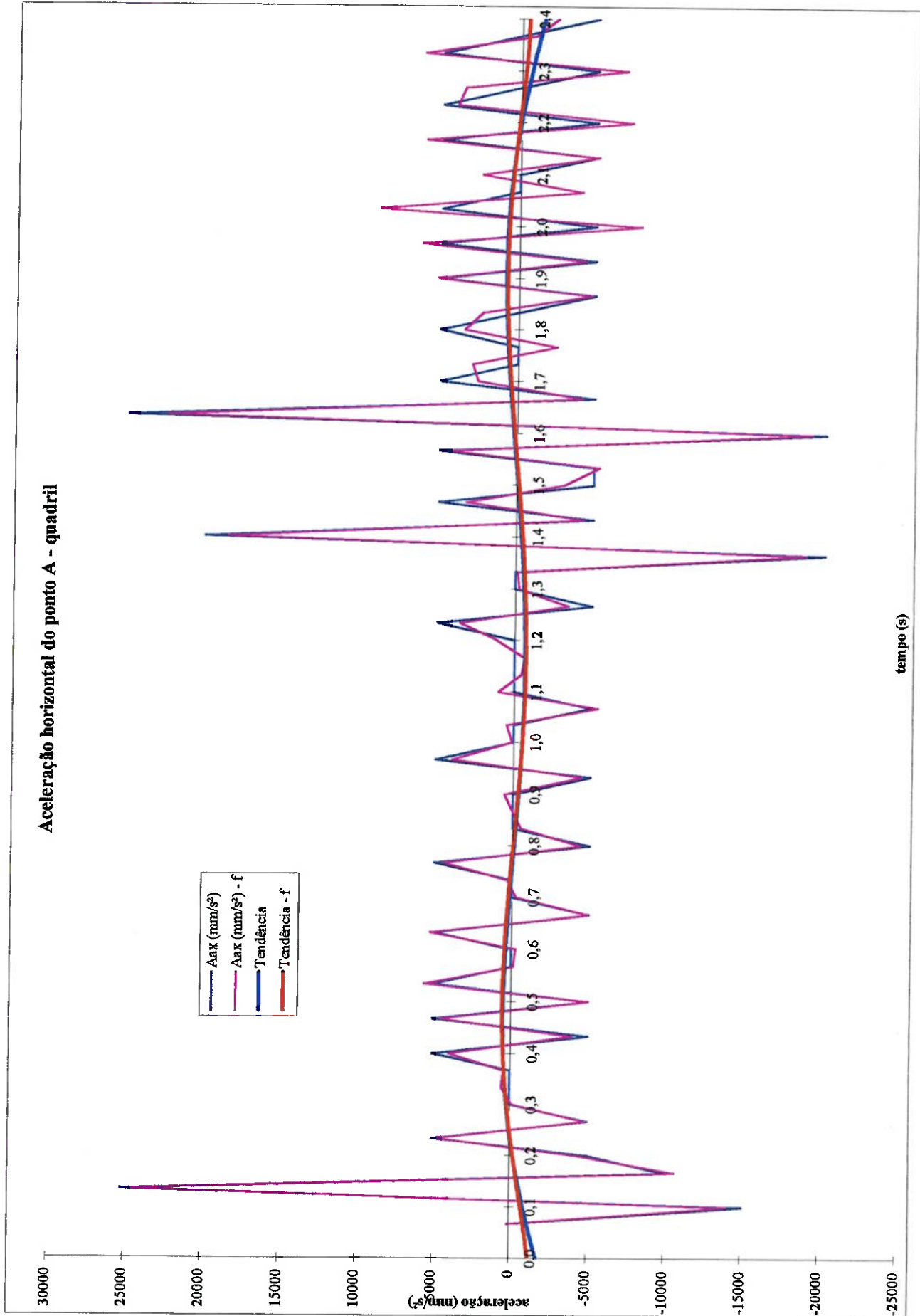




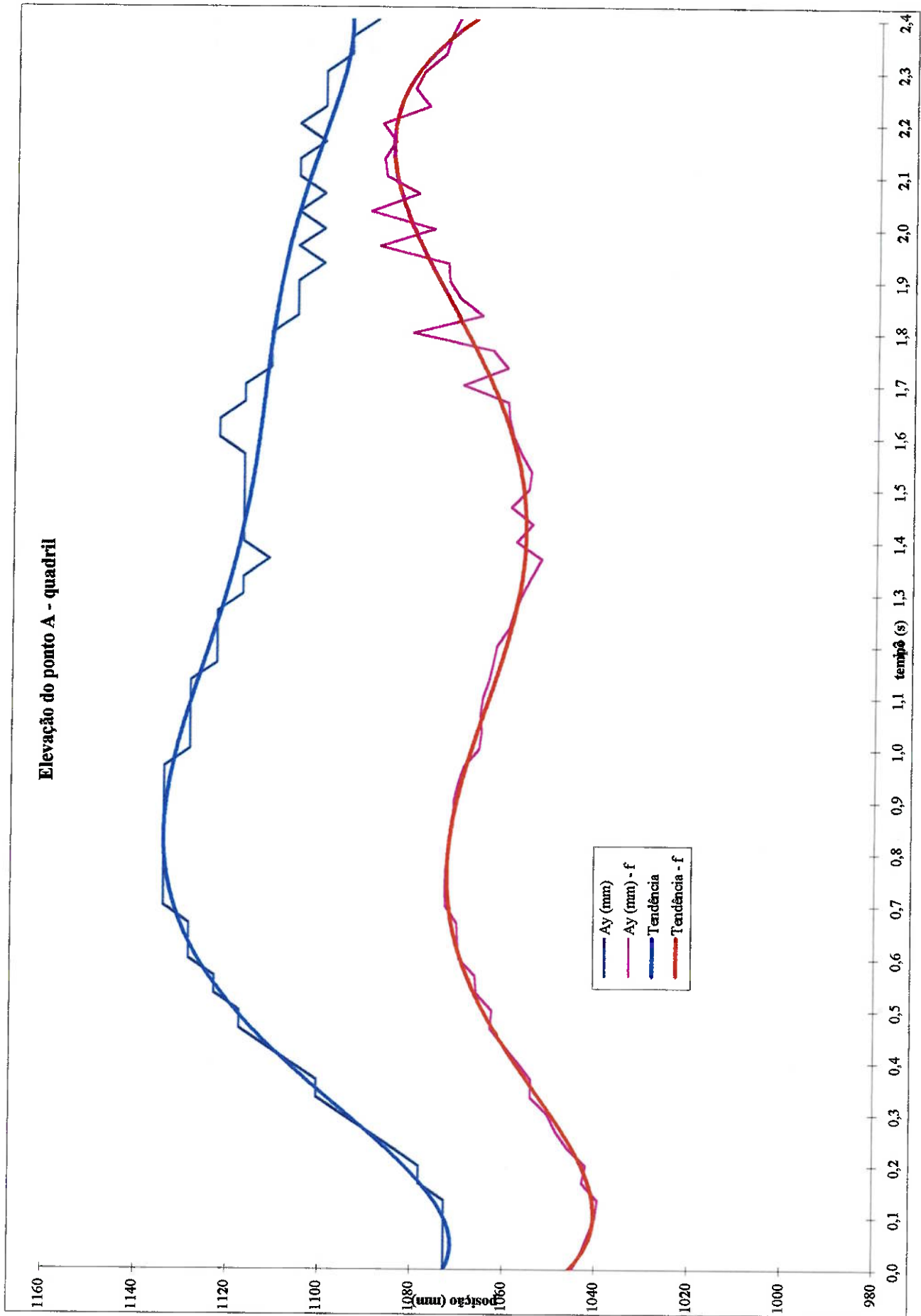




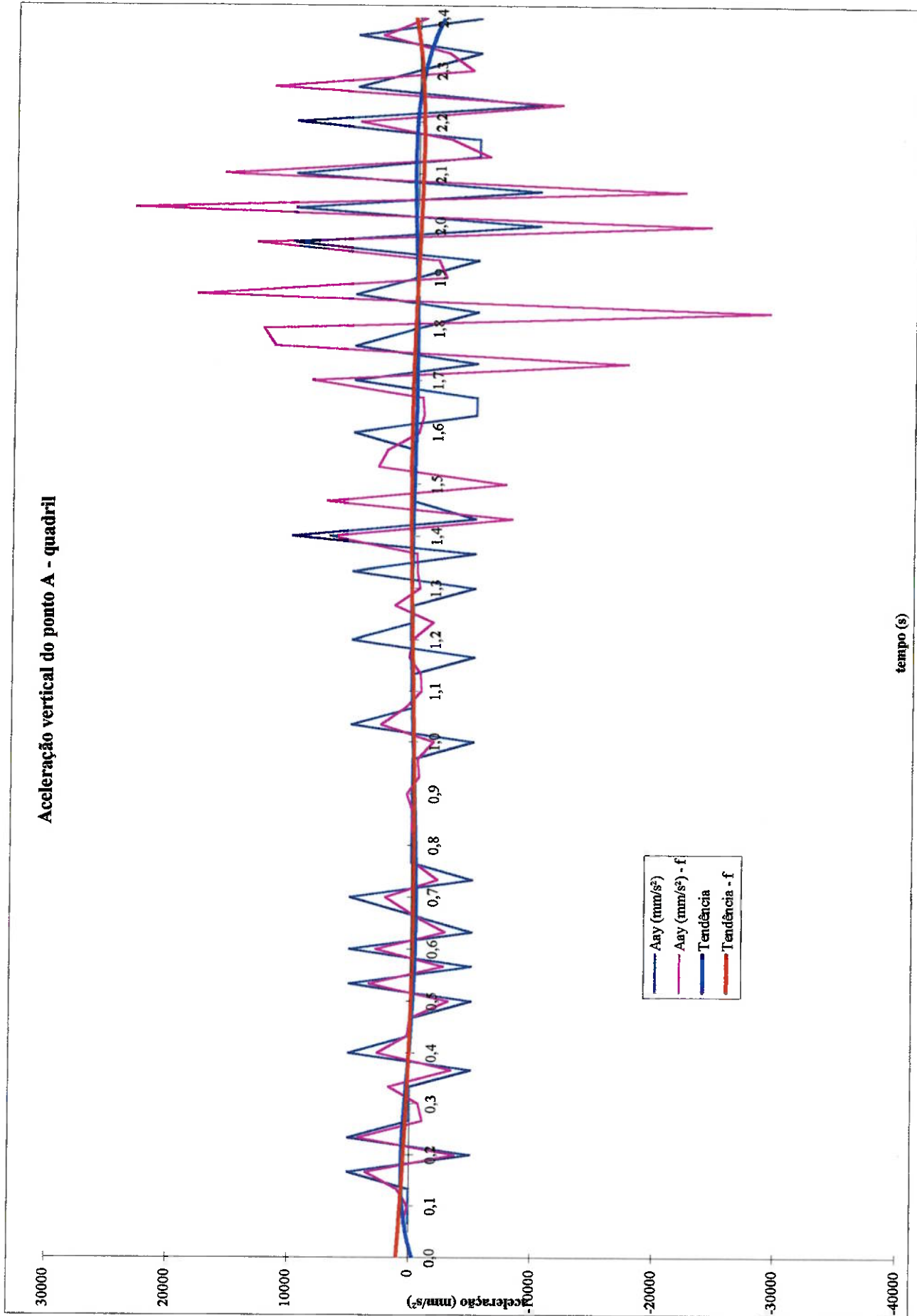
Aceleração horizontal do ponto A - quadril

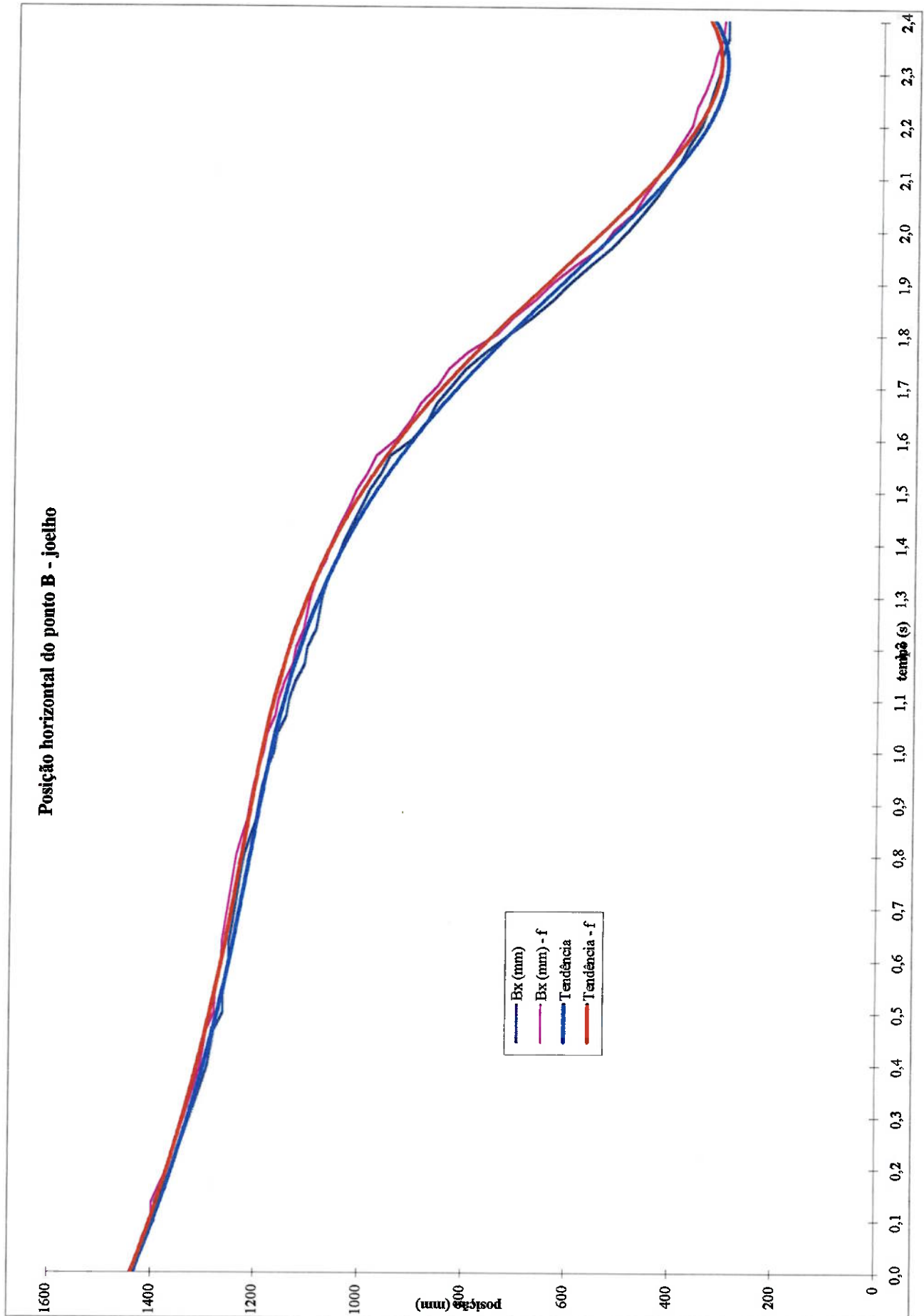


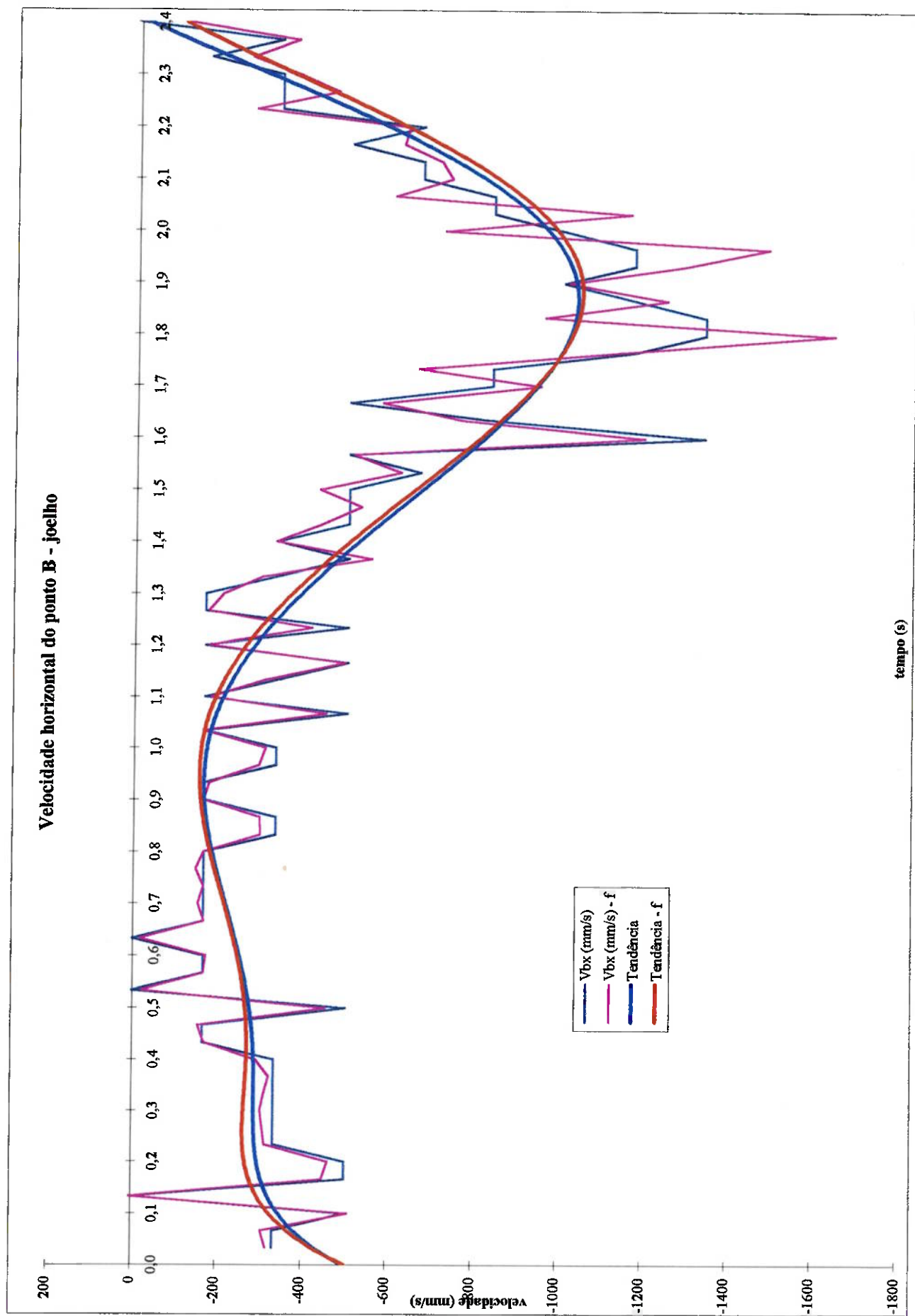
Elevação do ponto A - quadril



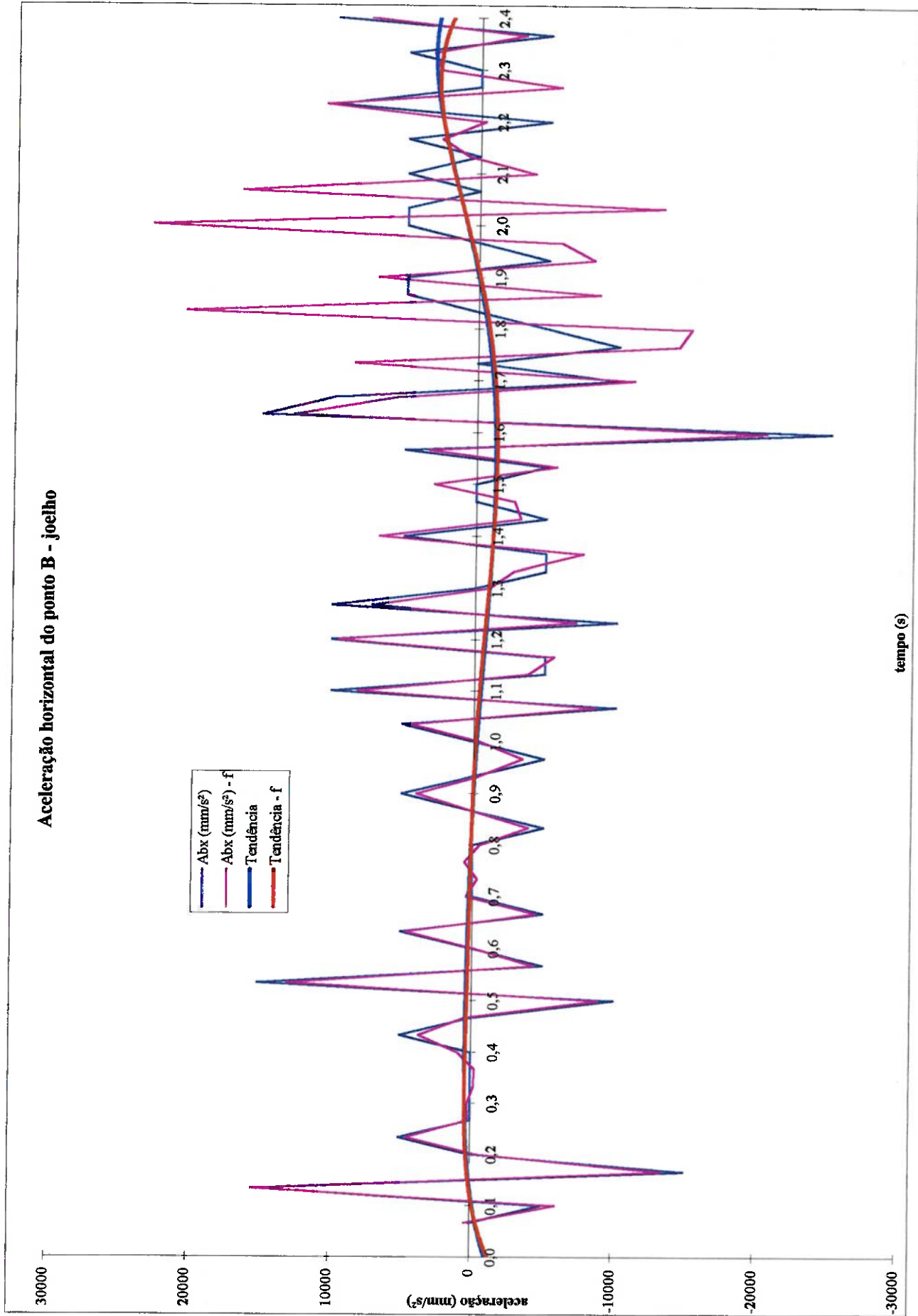
Aceleração vertical do ponto A - quadril



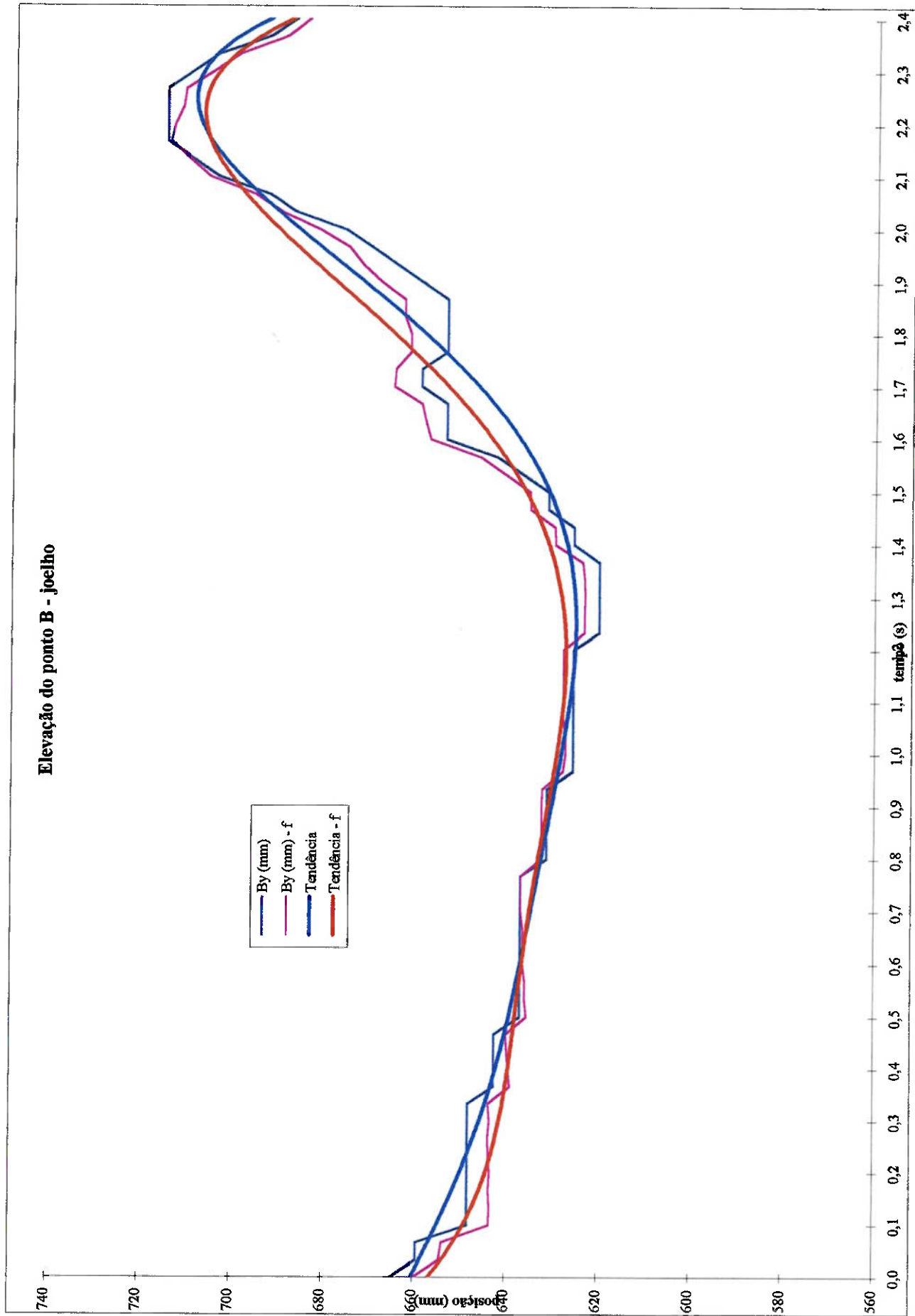


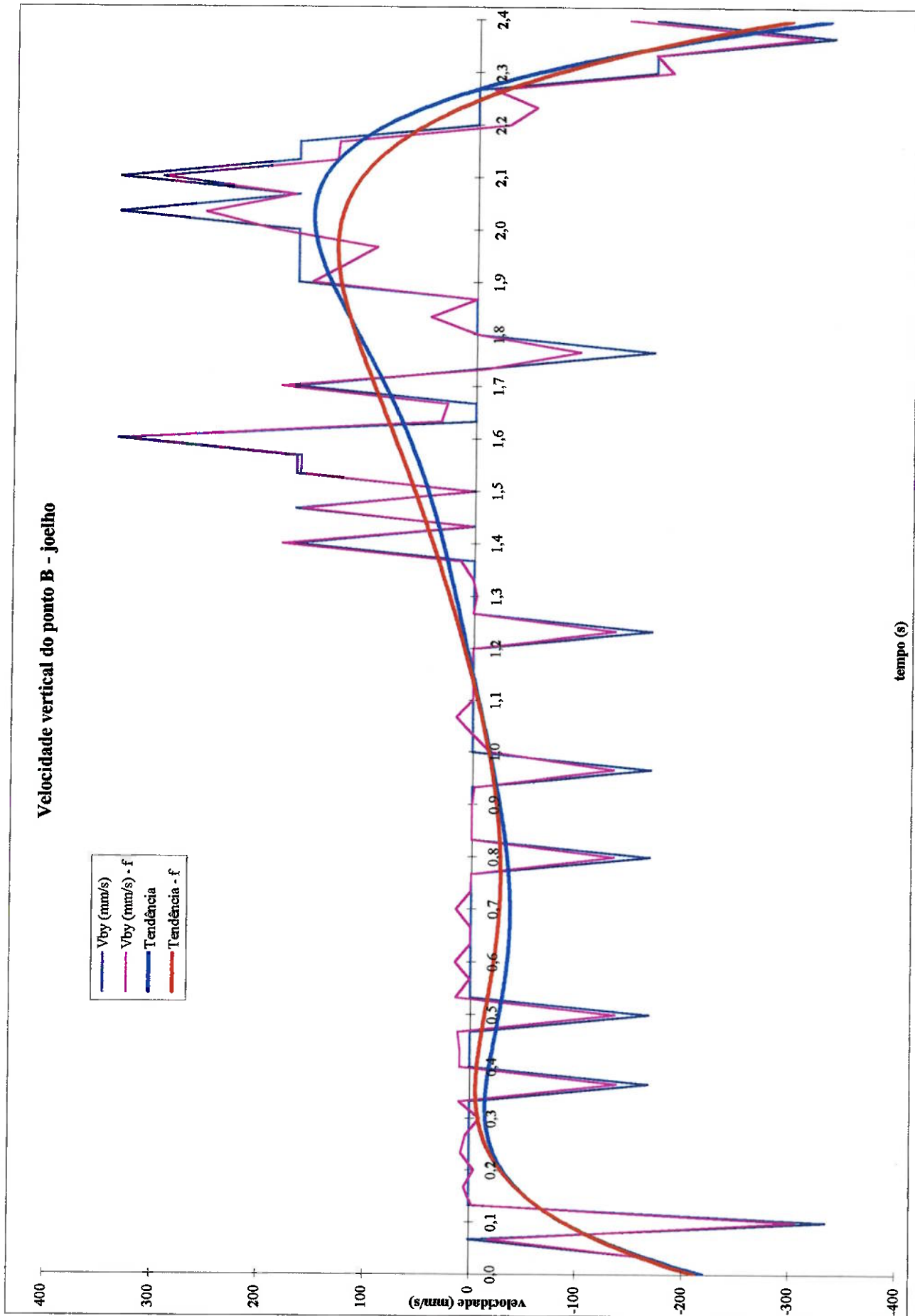


Aceleração horizontal do ponto B - joelho

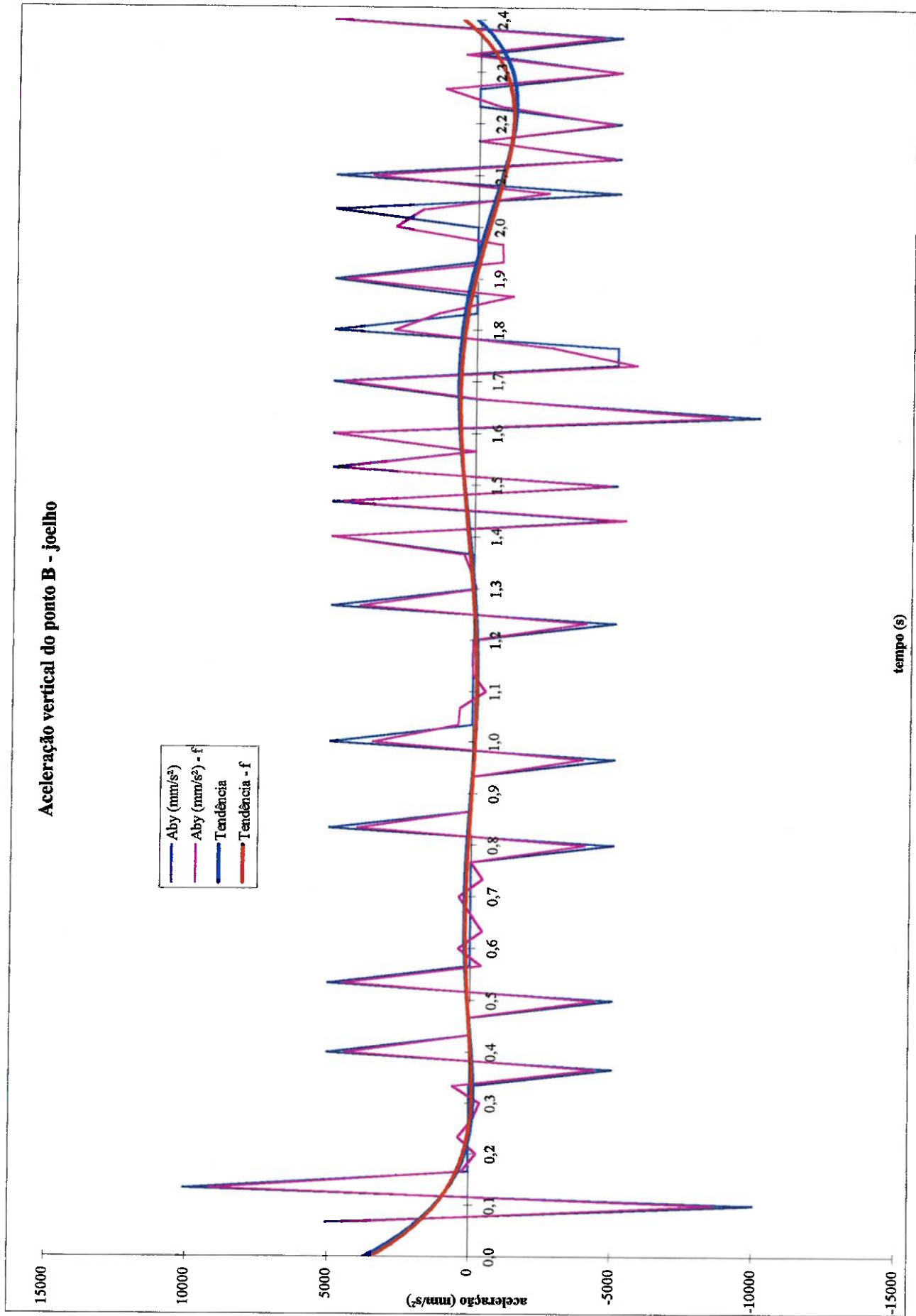


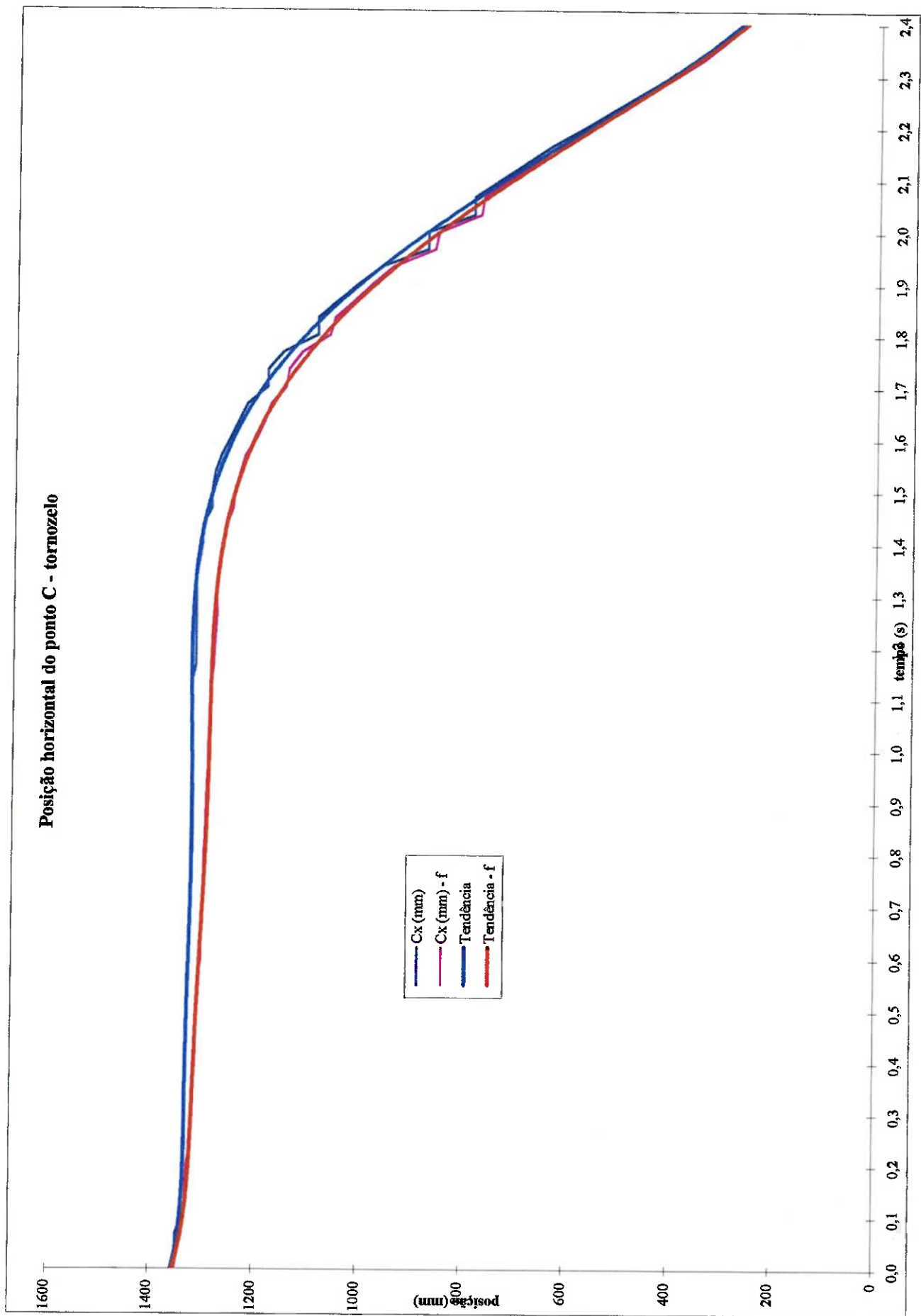
Elevação do ponto B - joelho

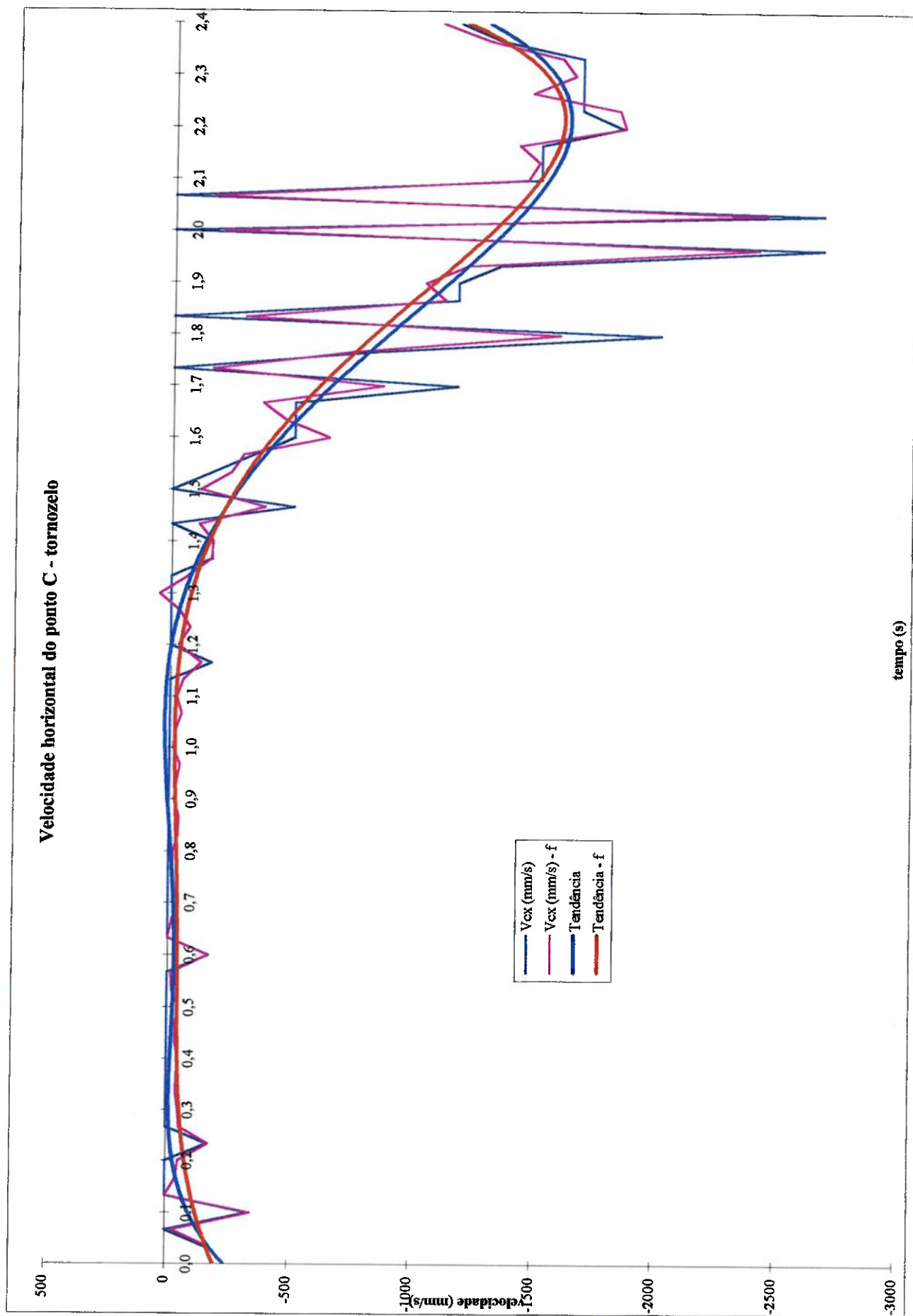




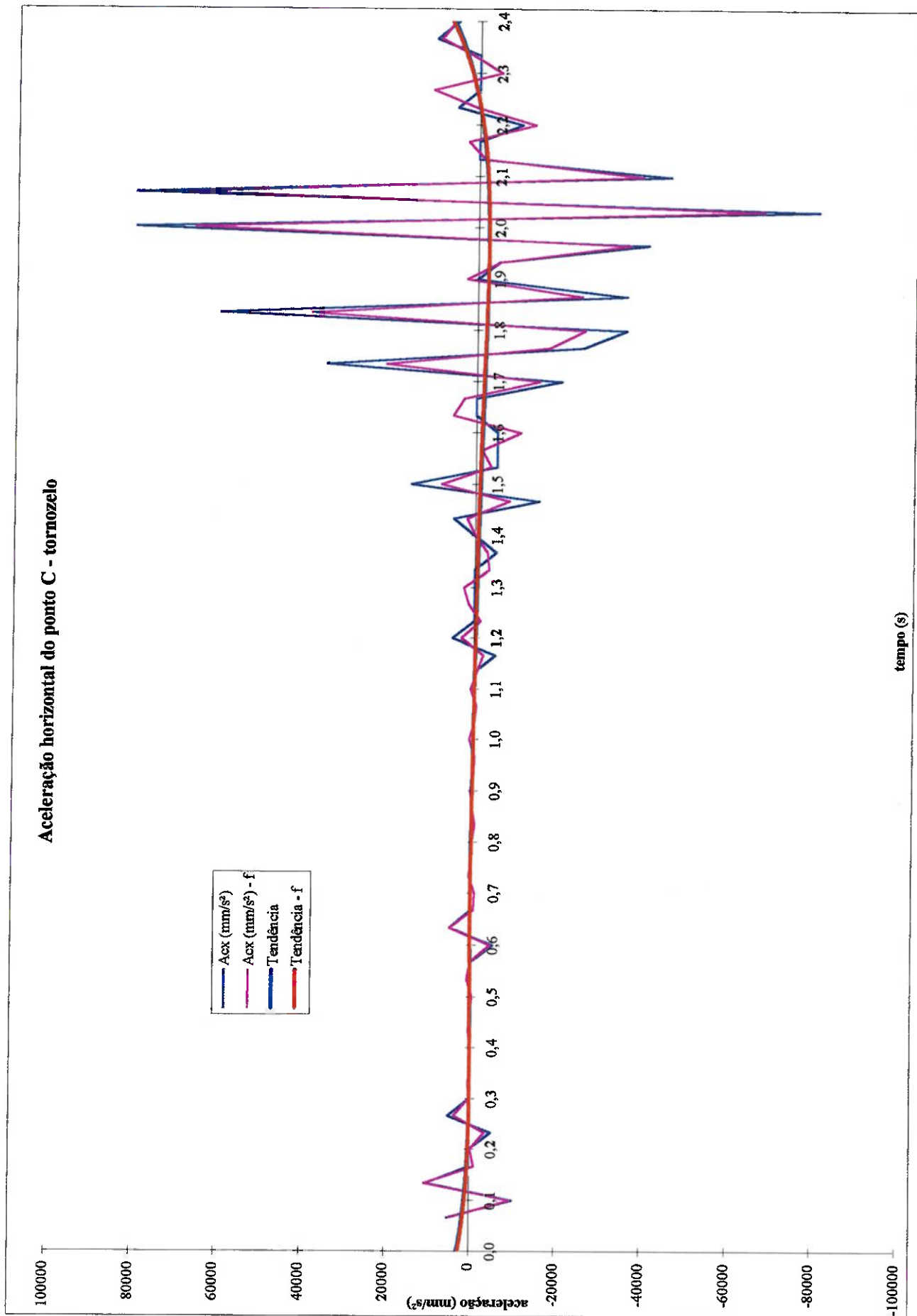
Aceleração vertical do ponto B - joelho



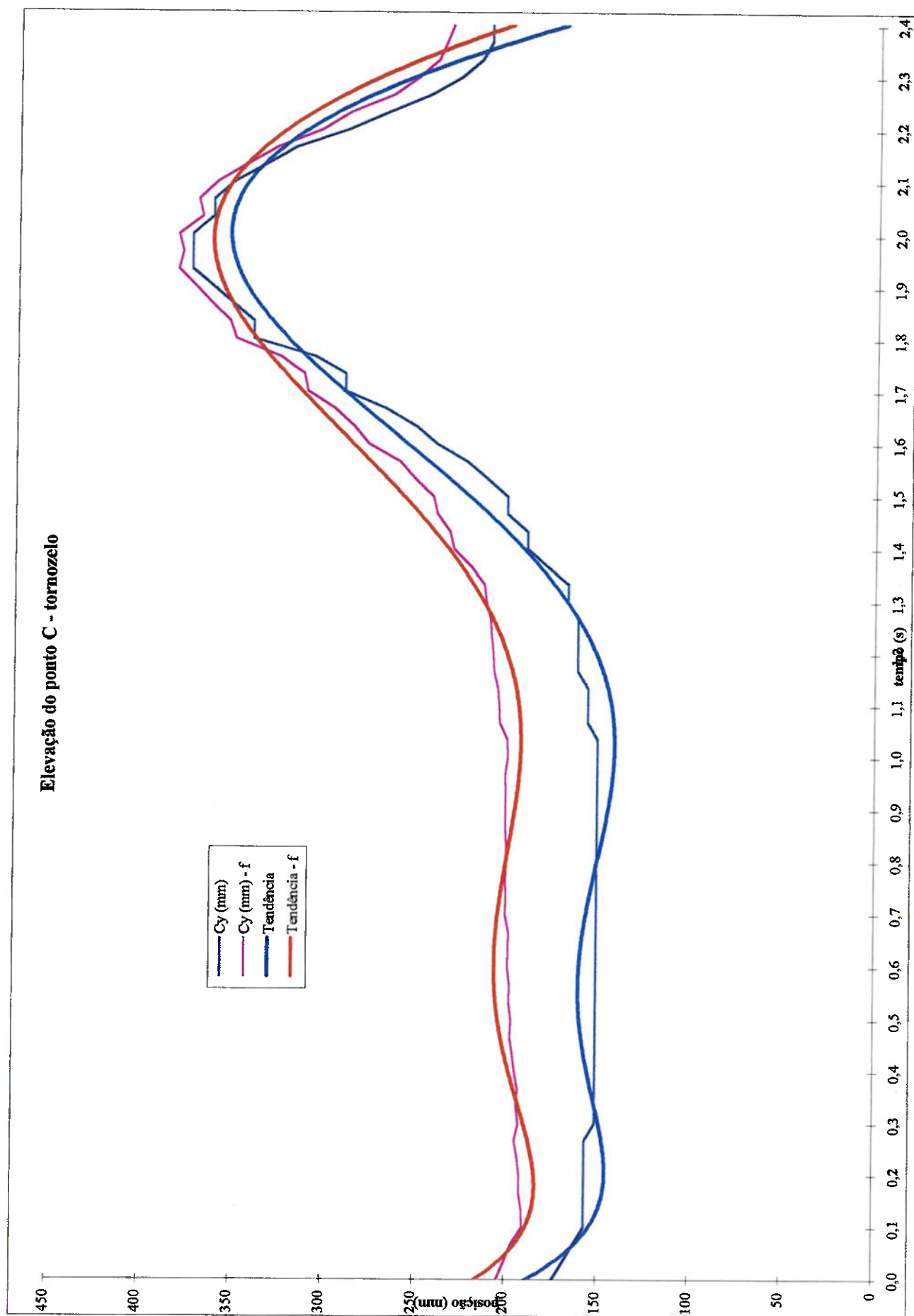




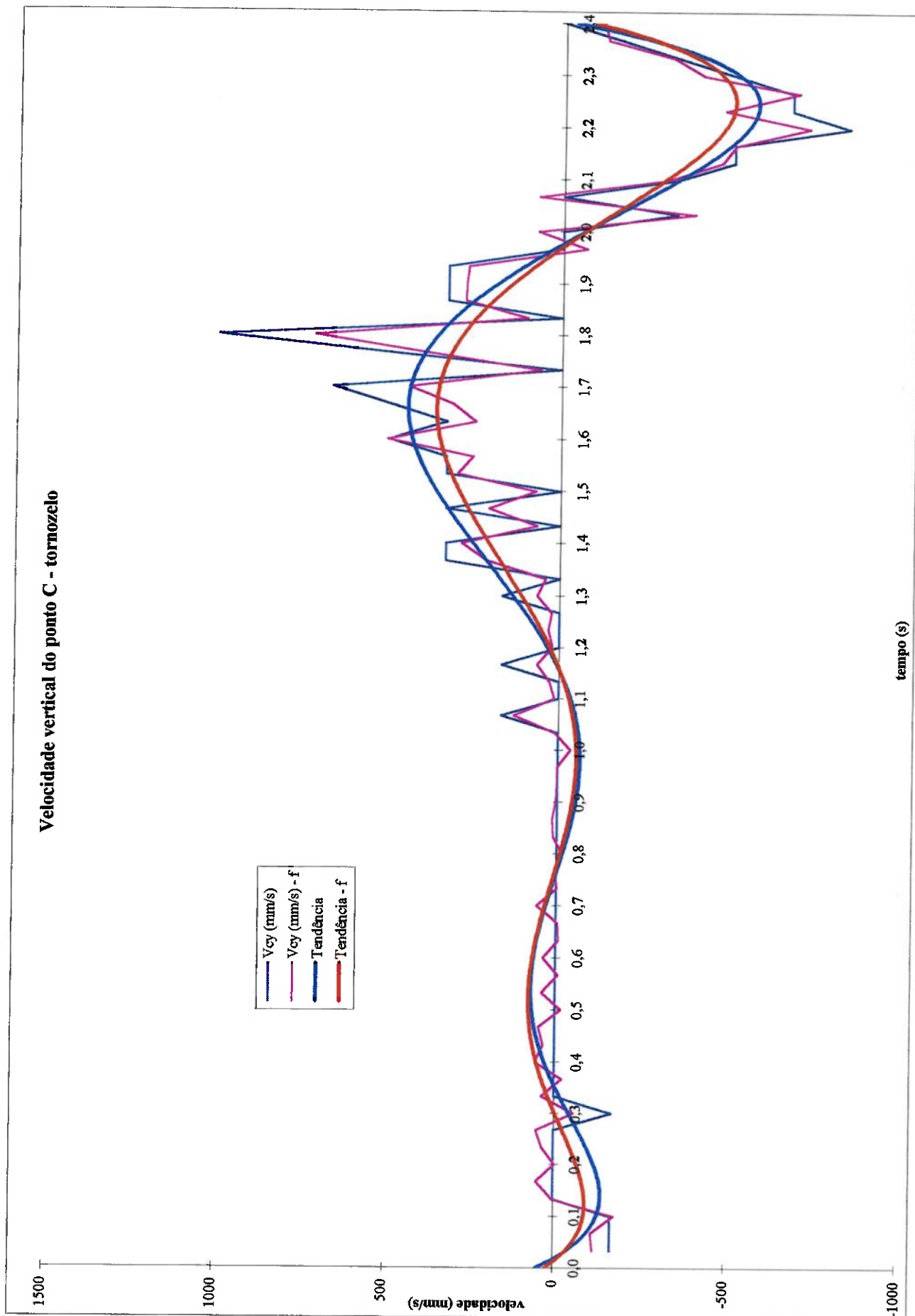
## Aceleração horizontal do ponto C - tornozelo

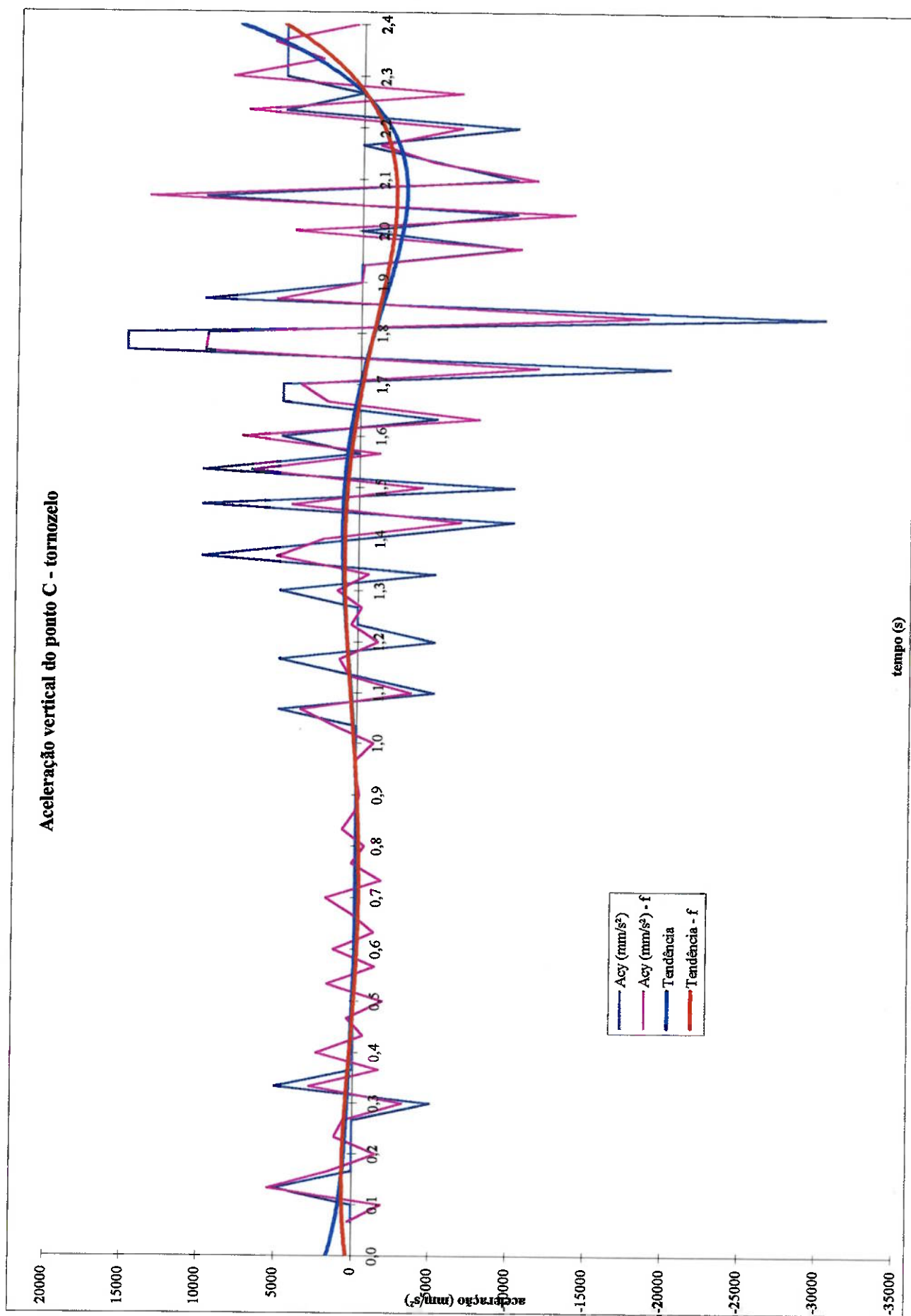


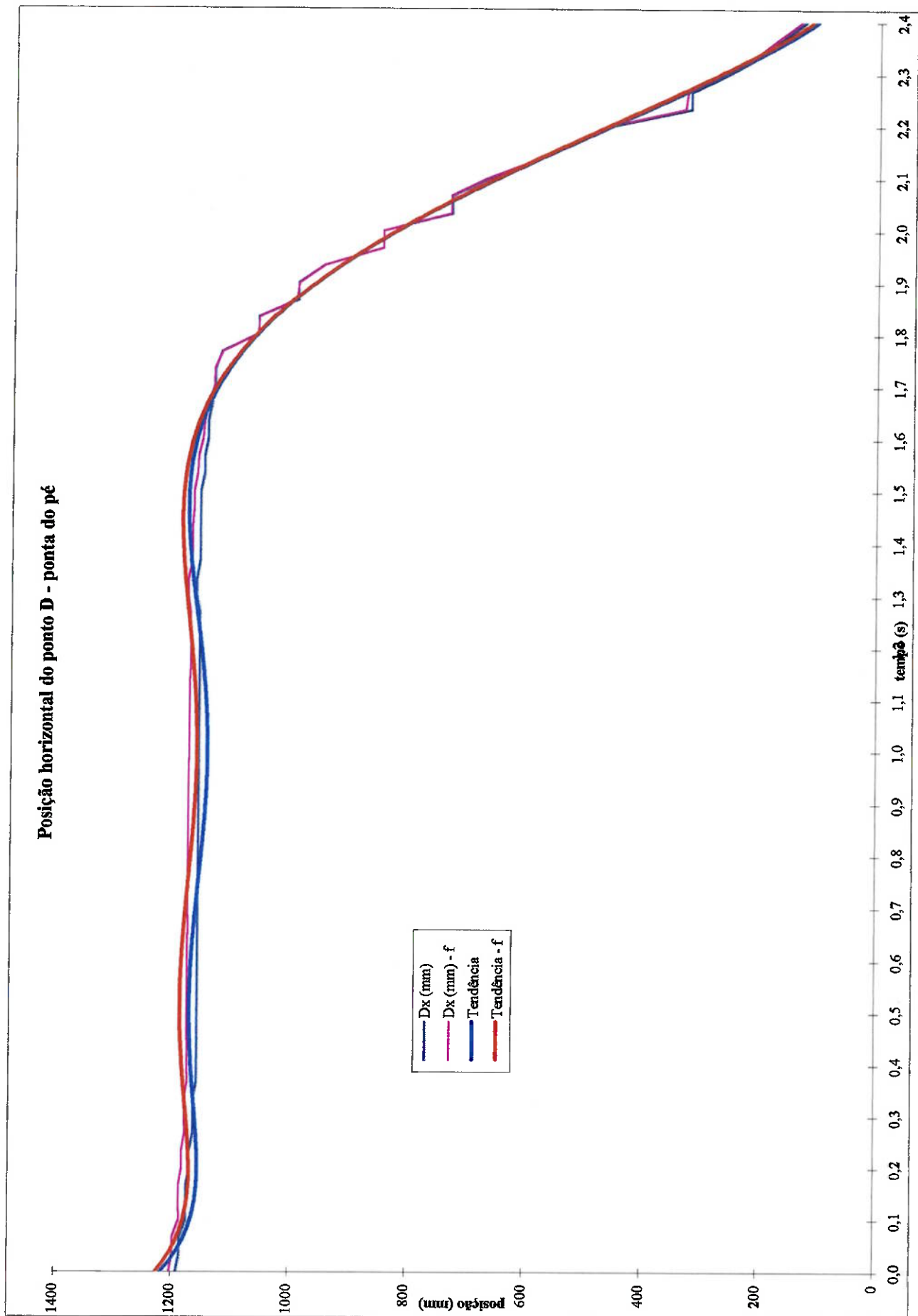
Elevação do ponto C - tornozelo

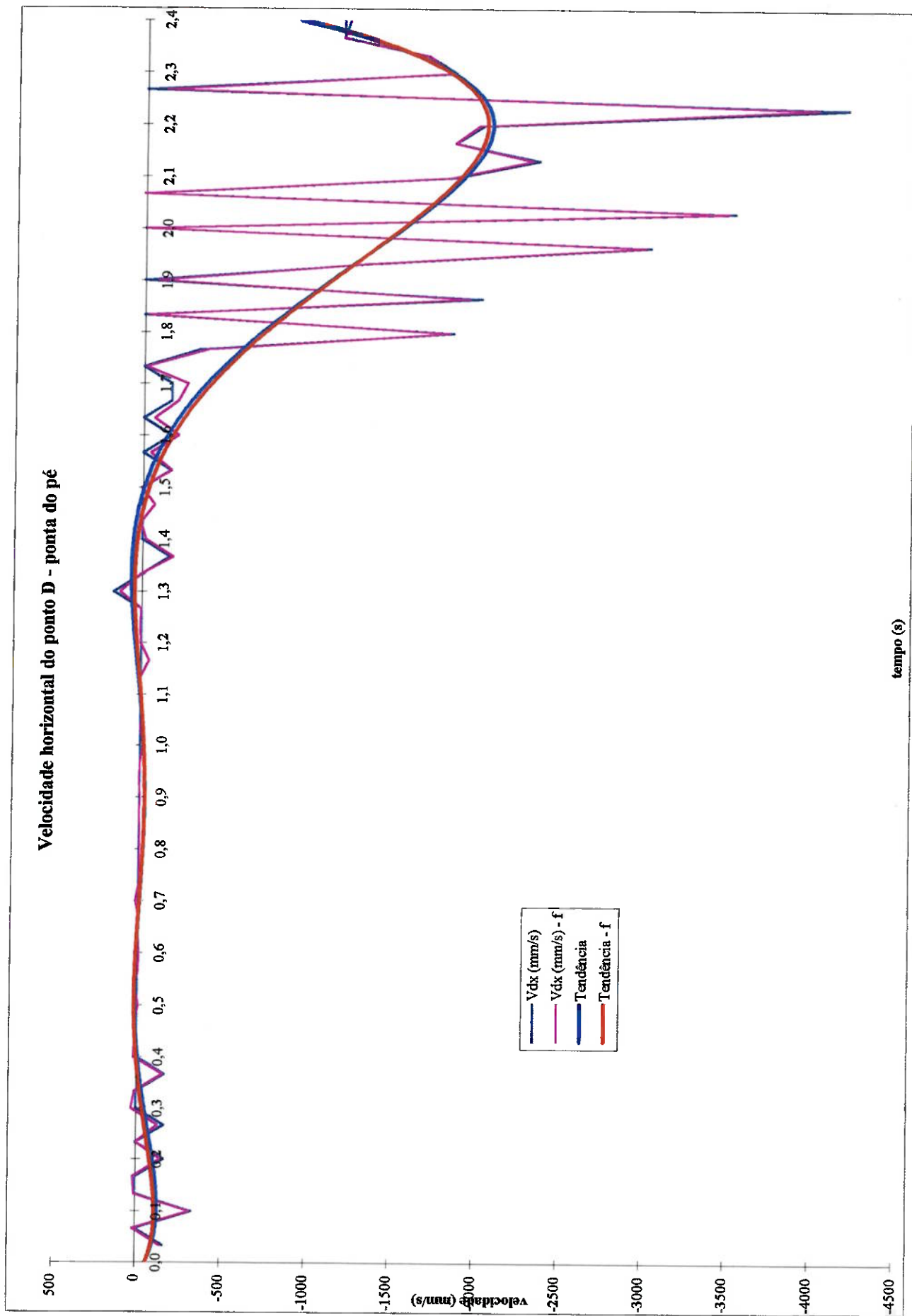


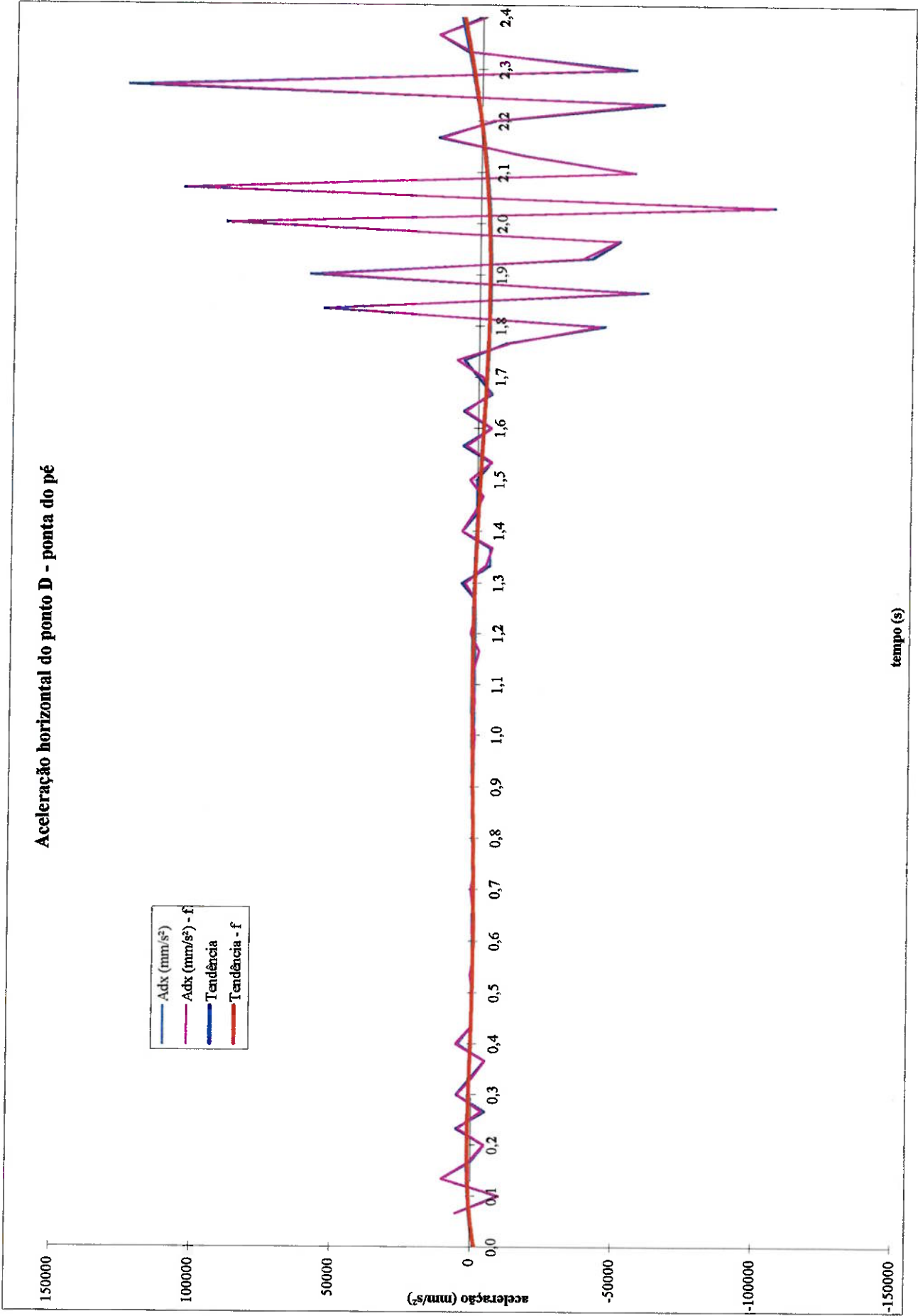
Velocidade vertical do ponto C - tornozelo



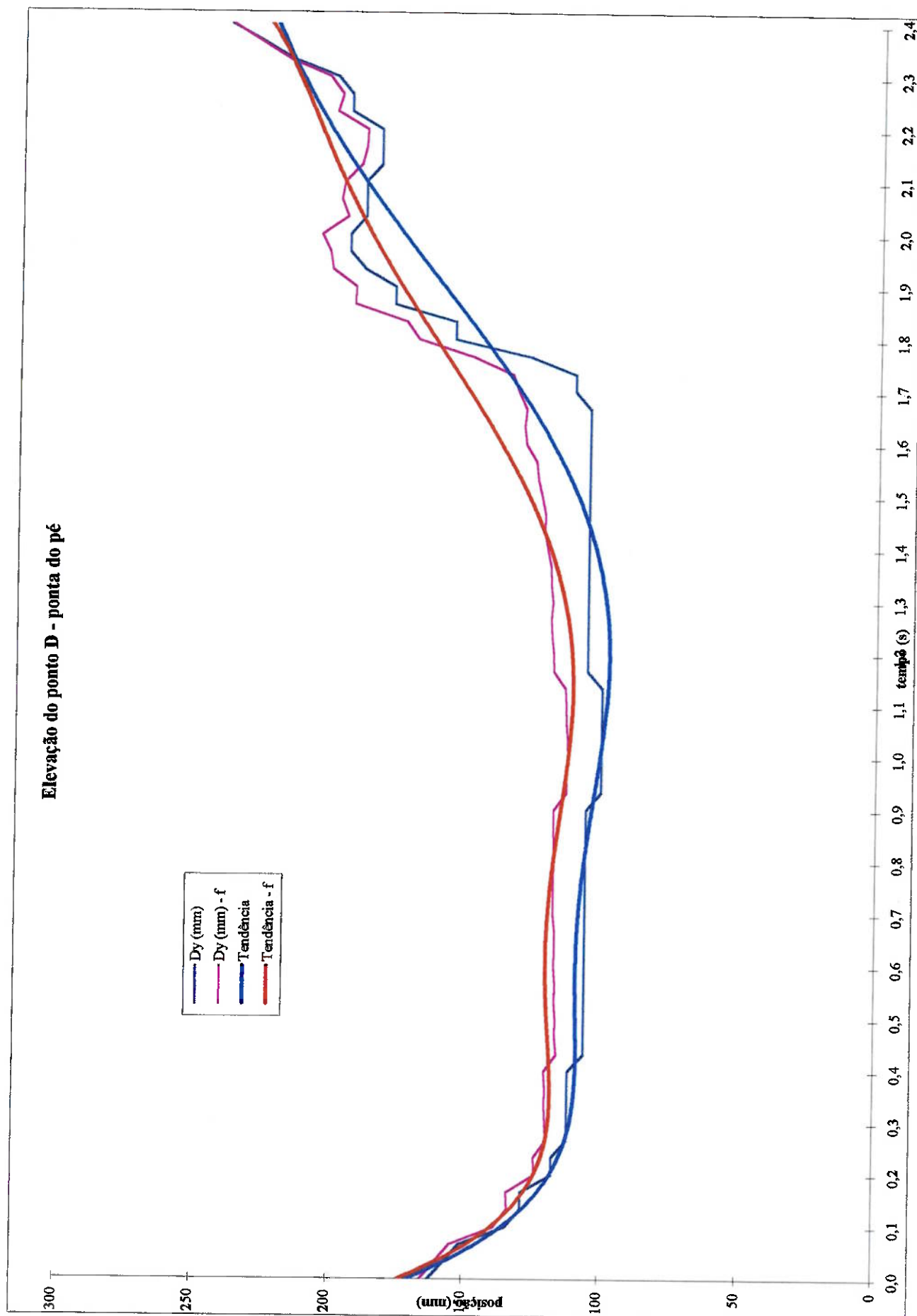




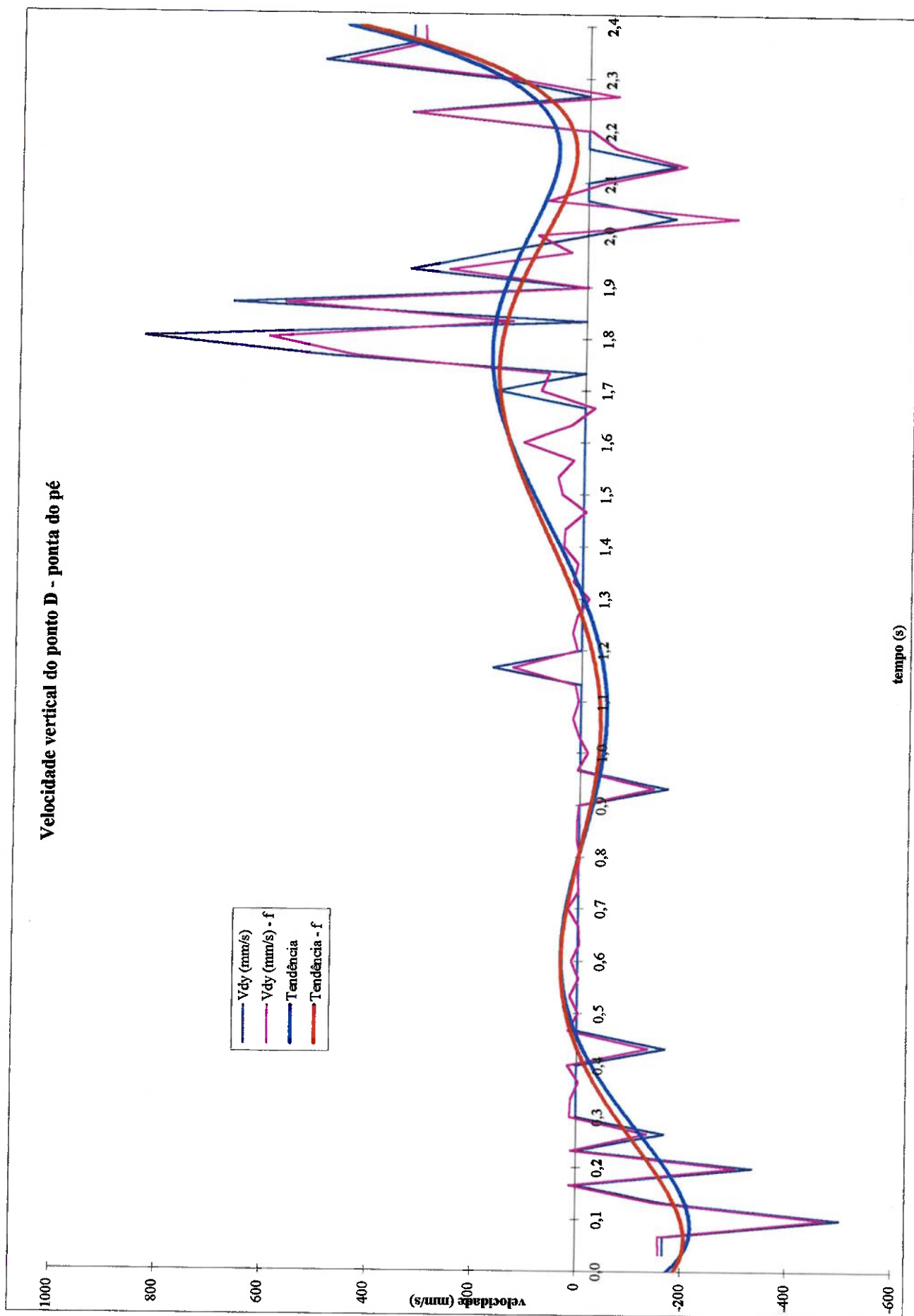




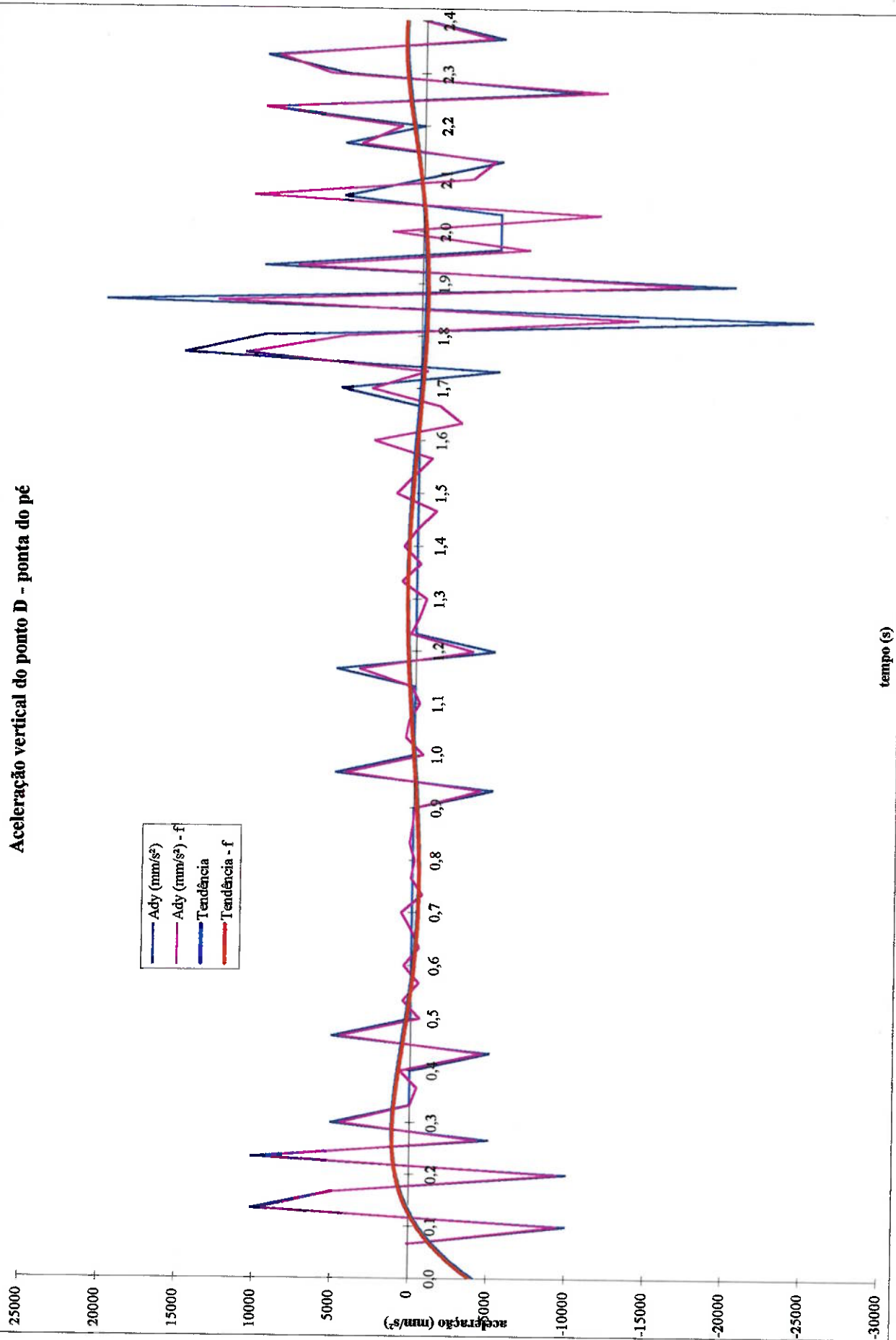
Elevação do ponto D - ponta do pé



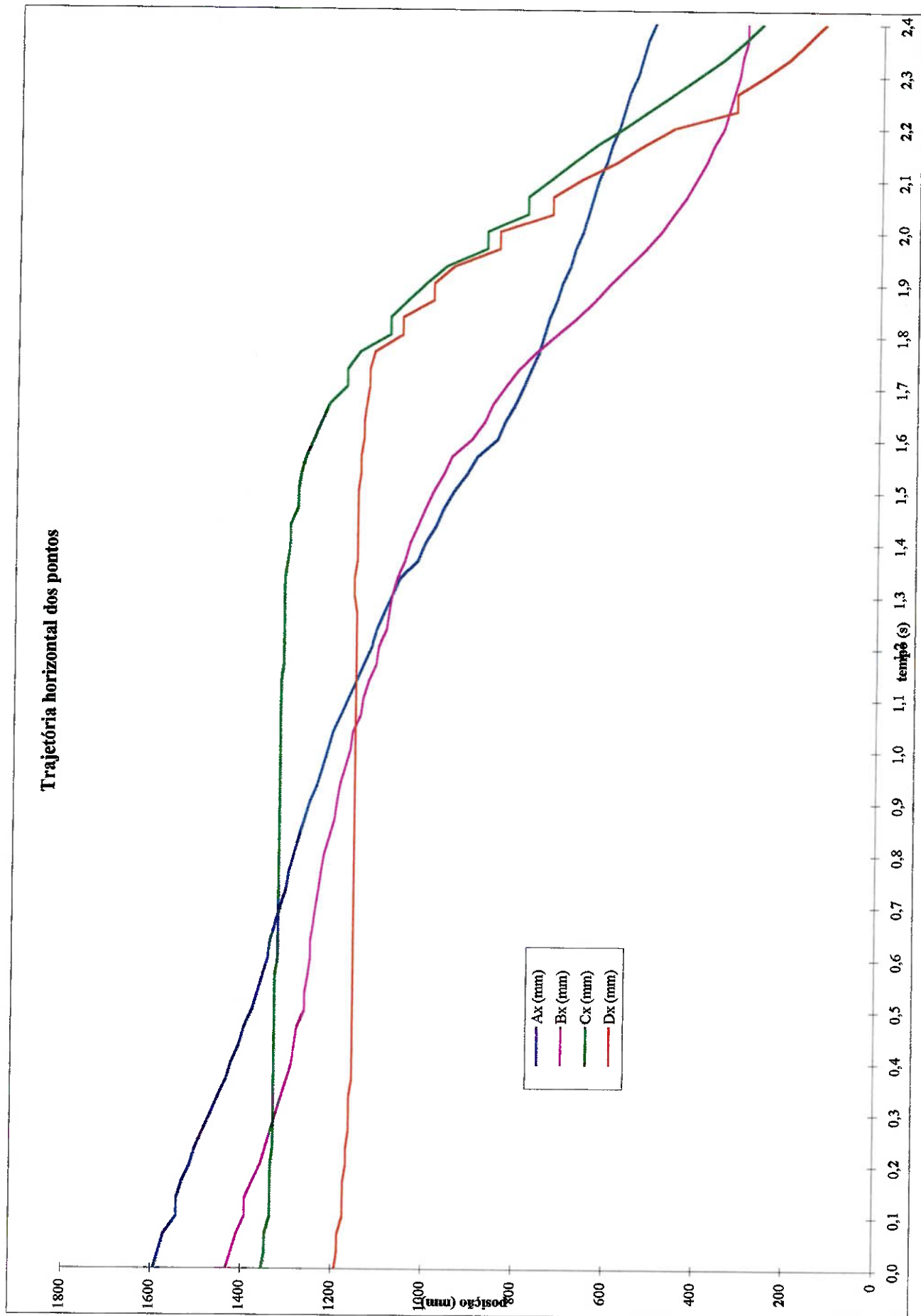
Velocidade vertical do ponto D - ponta do pé



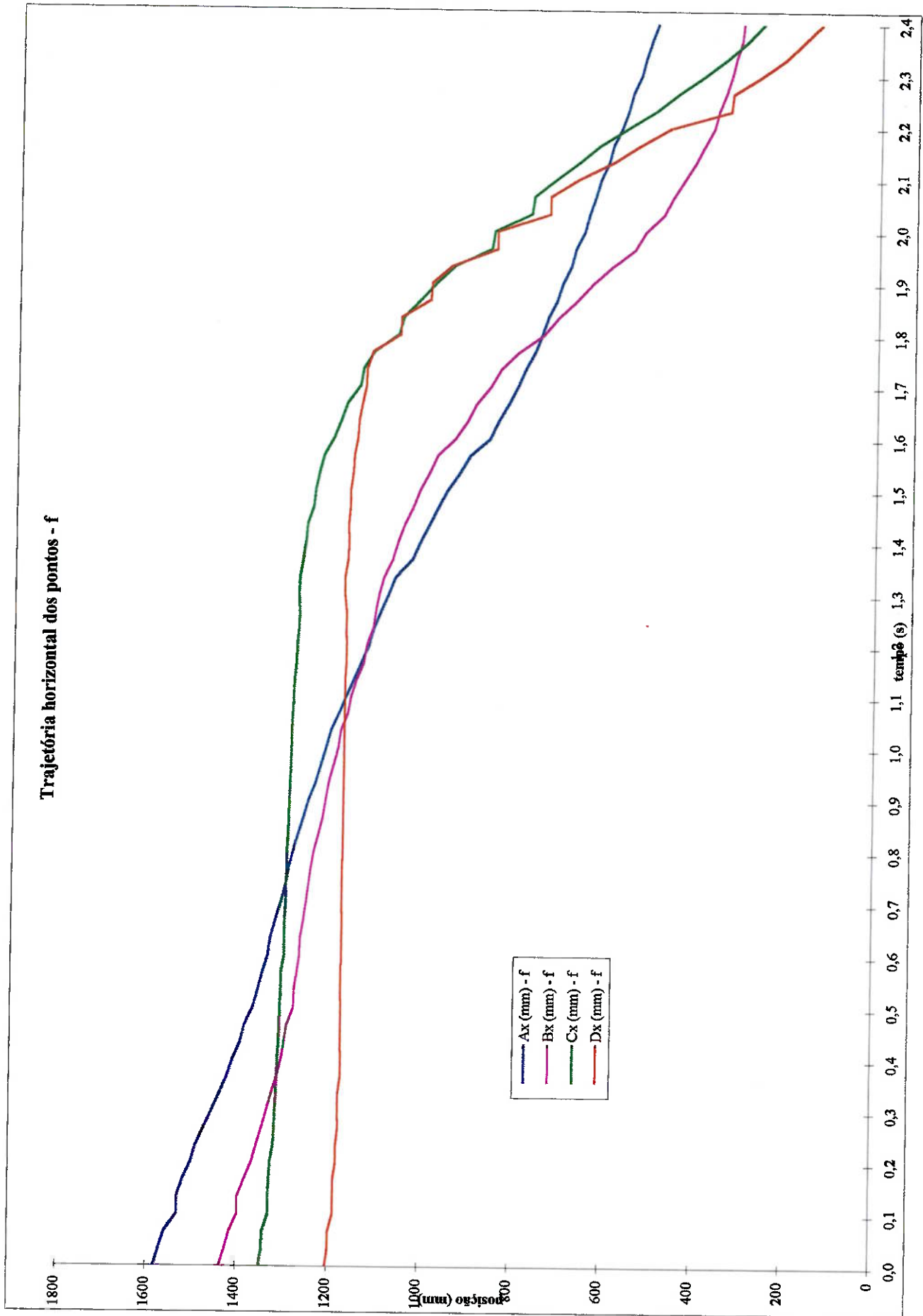
Aceleração vertical do ponto D - ponta do pé



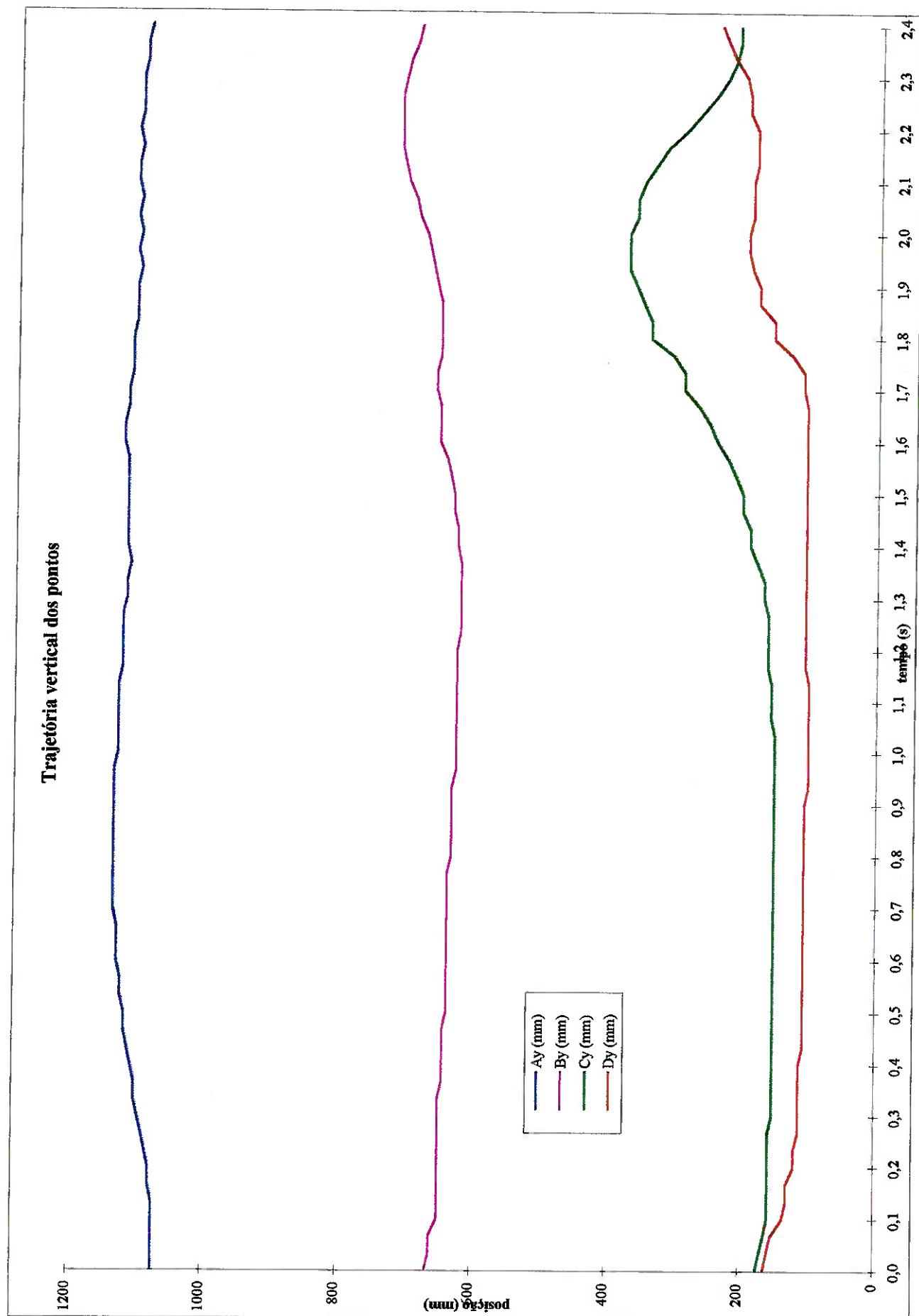
Trajetória horizontal dos pontos



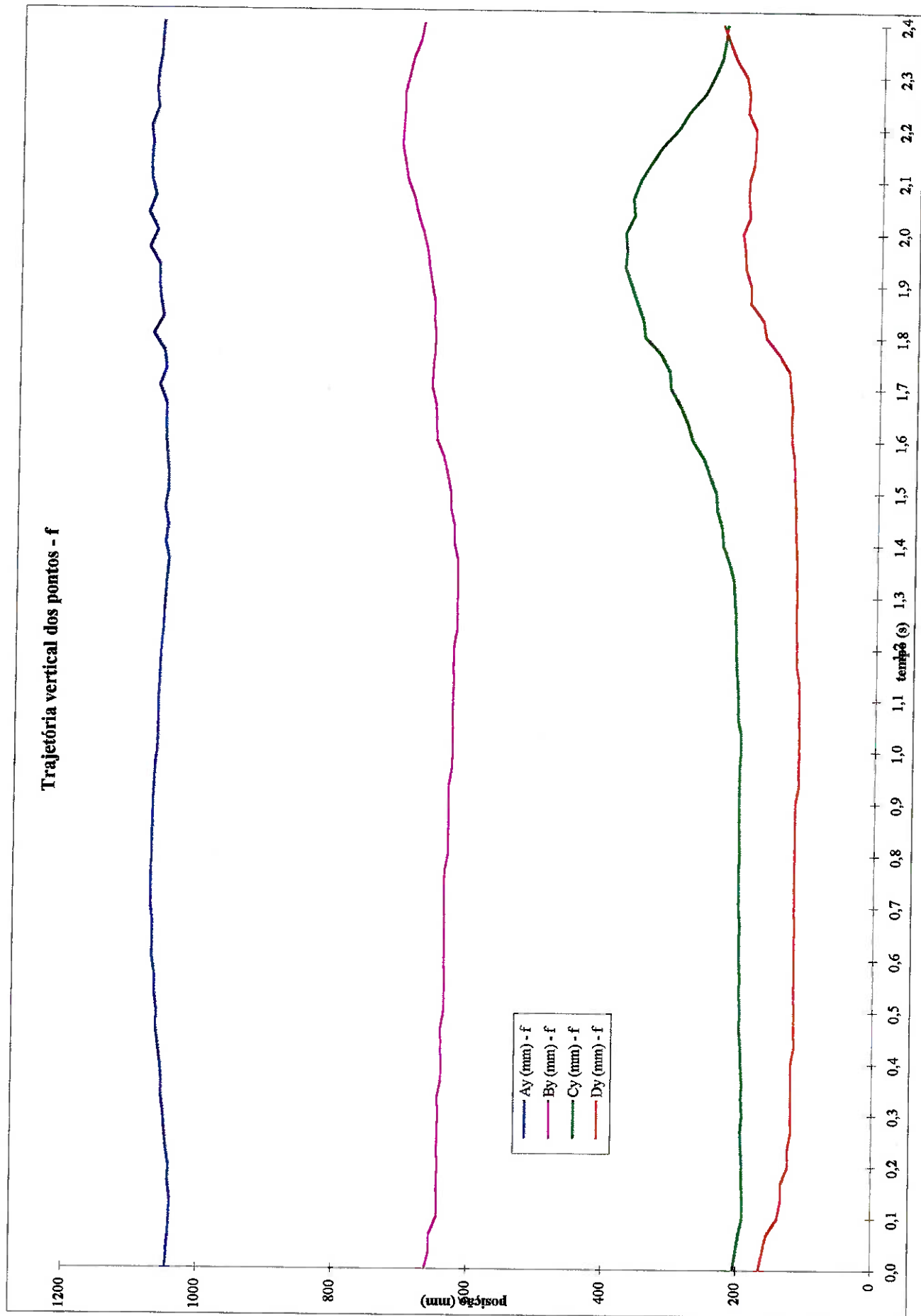
Trajetória horizontal dos pontos - f

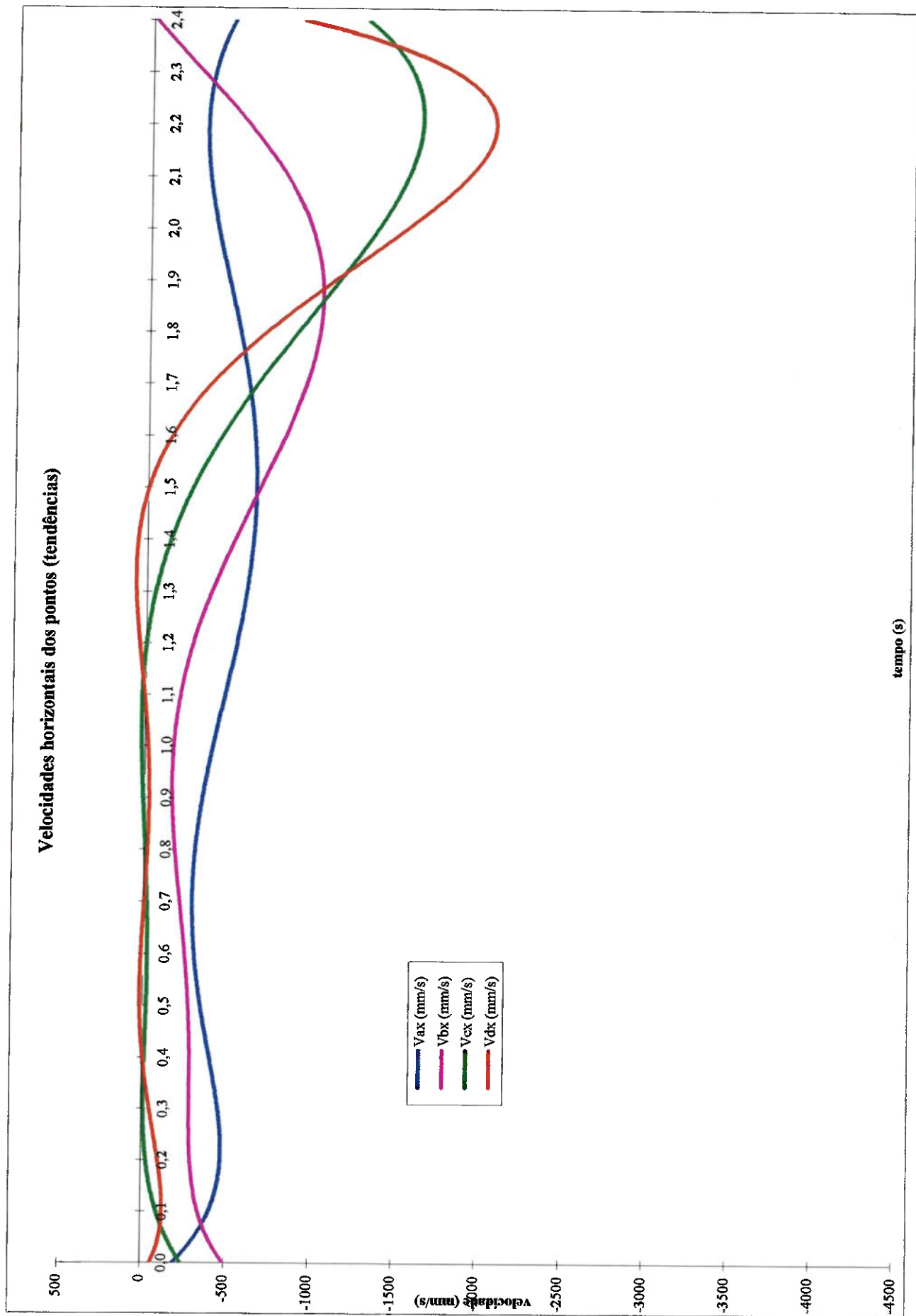


Trajetória vertical dos pontos

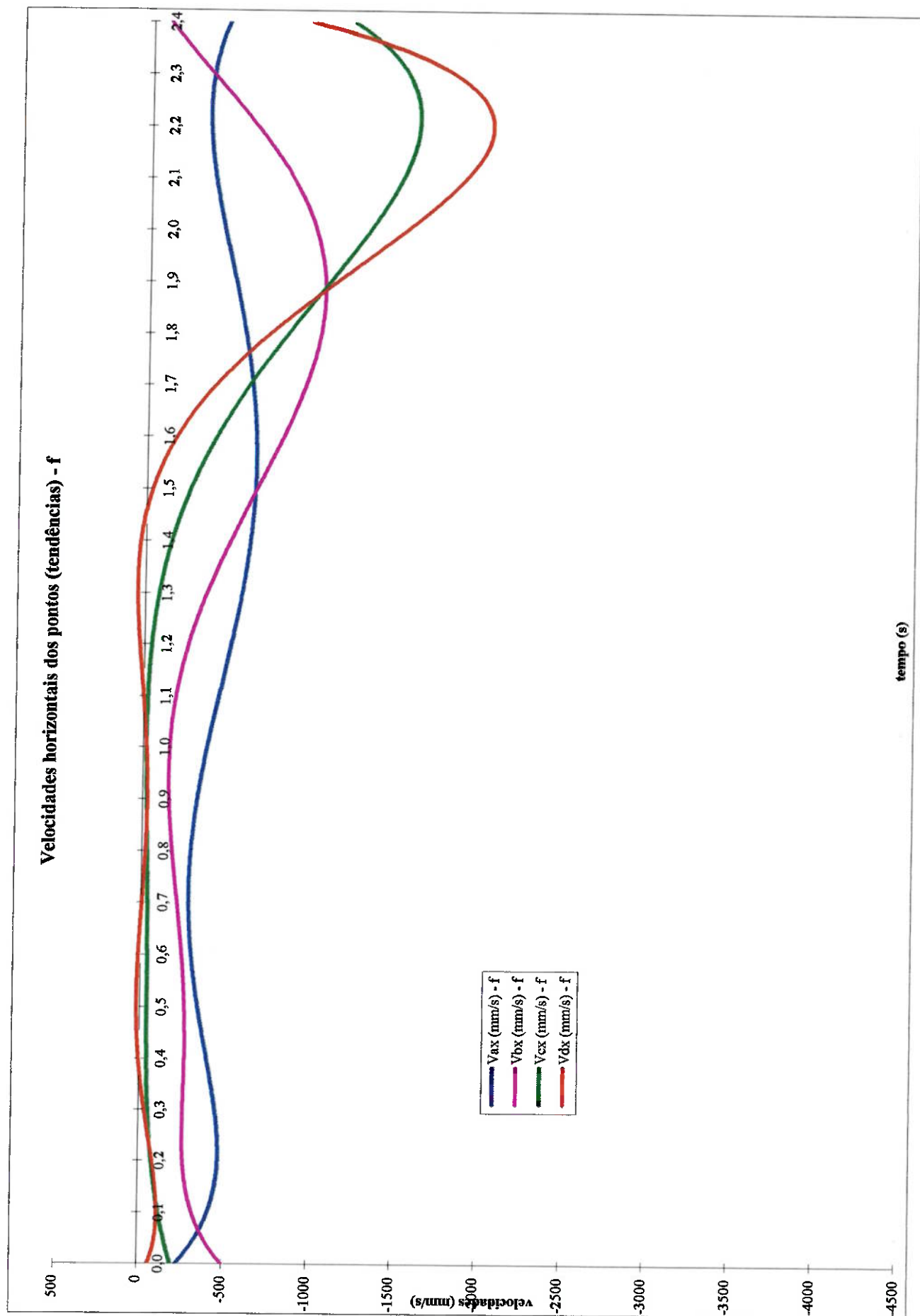


Trajetória vertical dos pontos - f

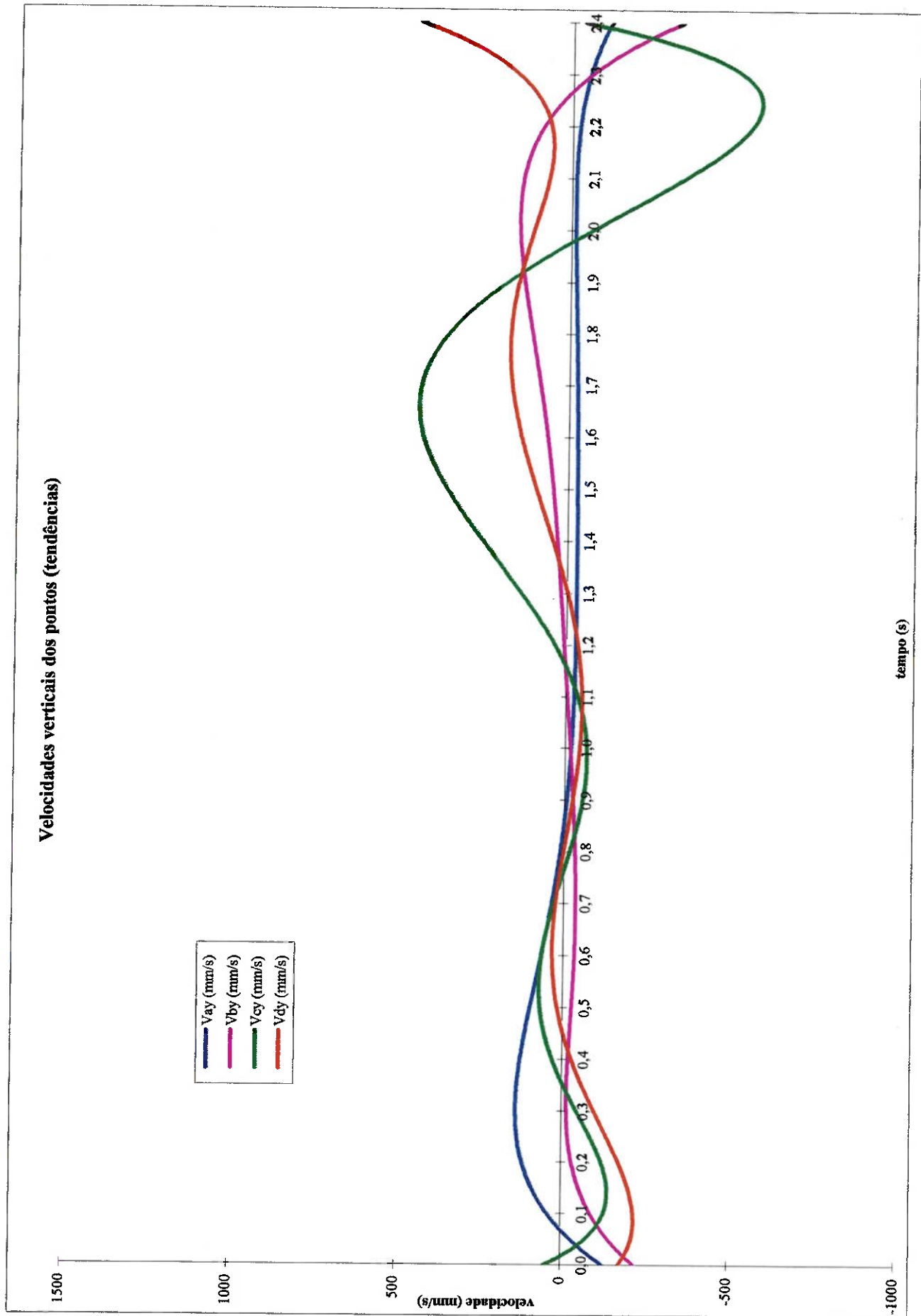




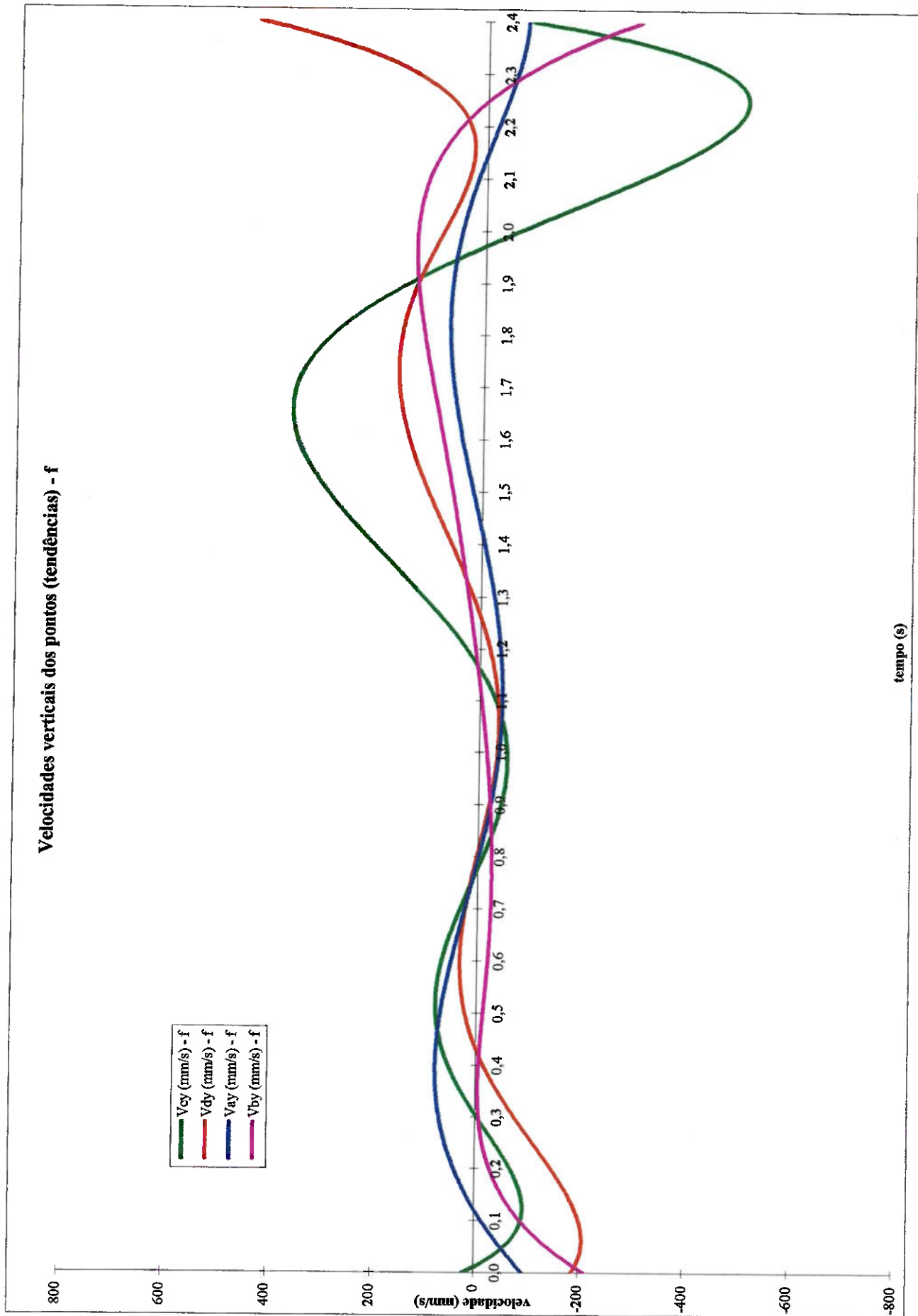
## Velocidades horizontais dos pontos (tendências) - f



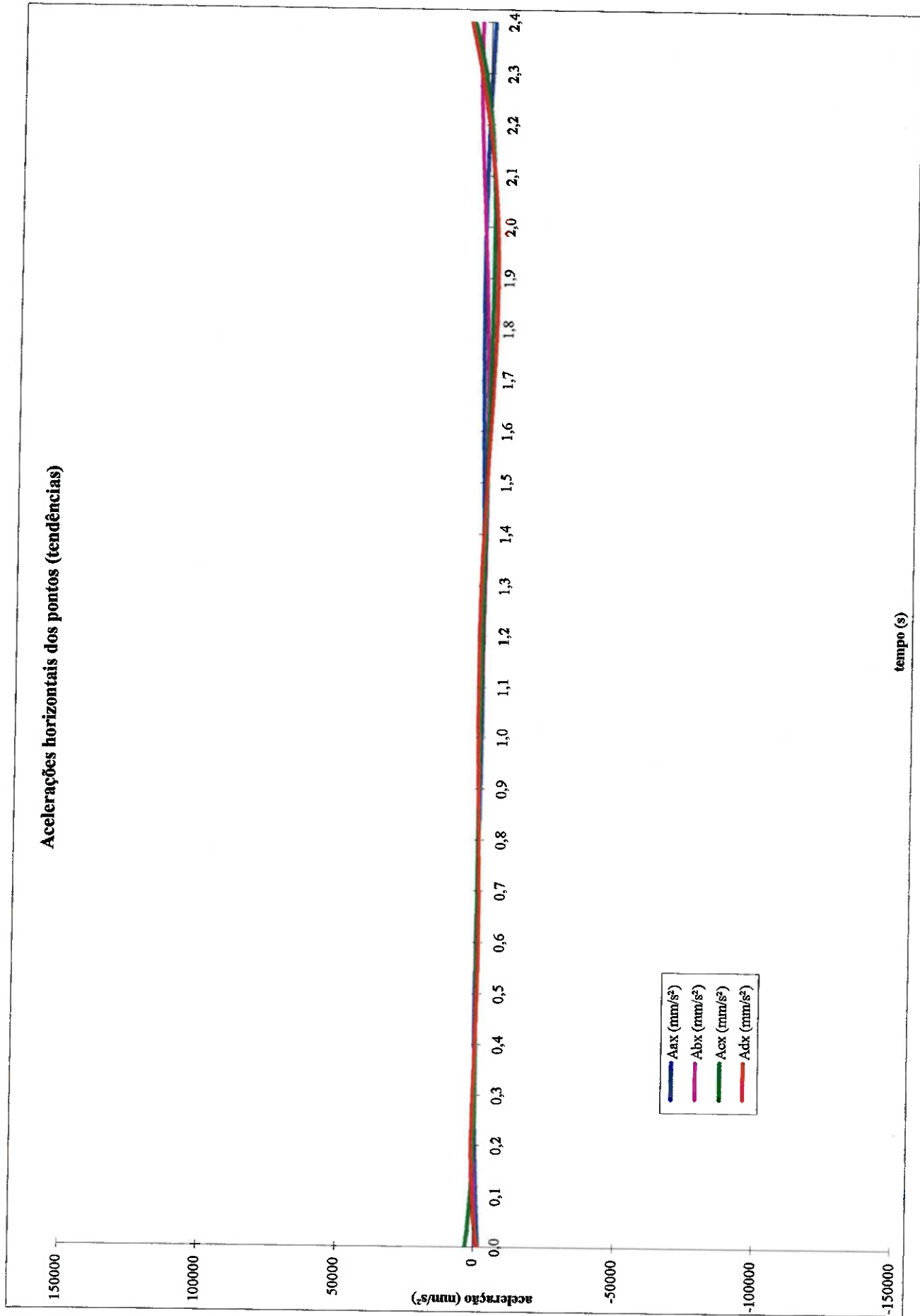
Velocidades verticais dos pontos (tendências)



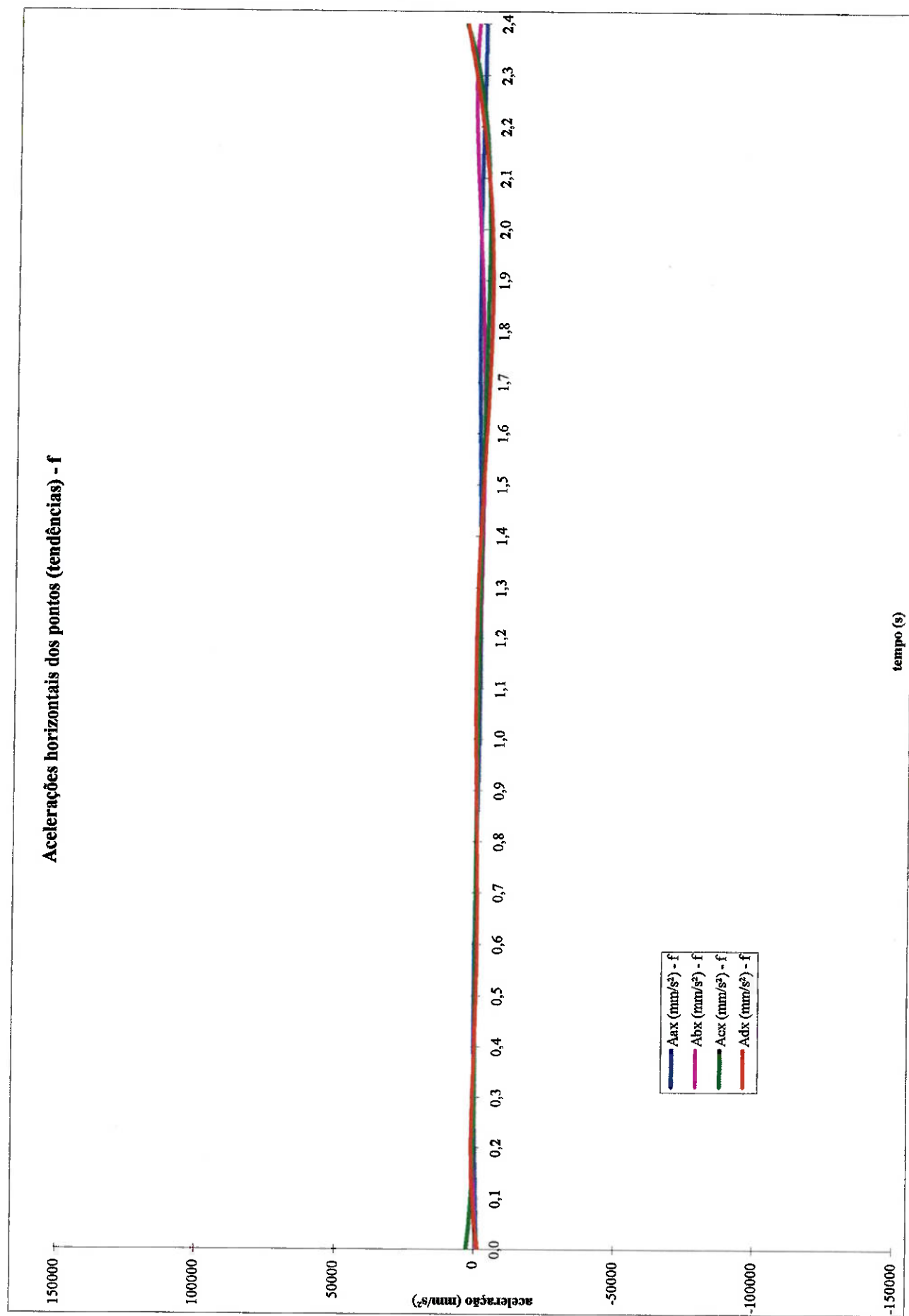
Velocidades verticais dos pontos (tendências) - f



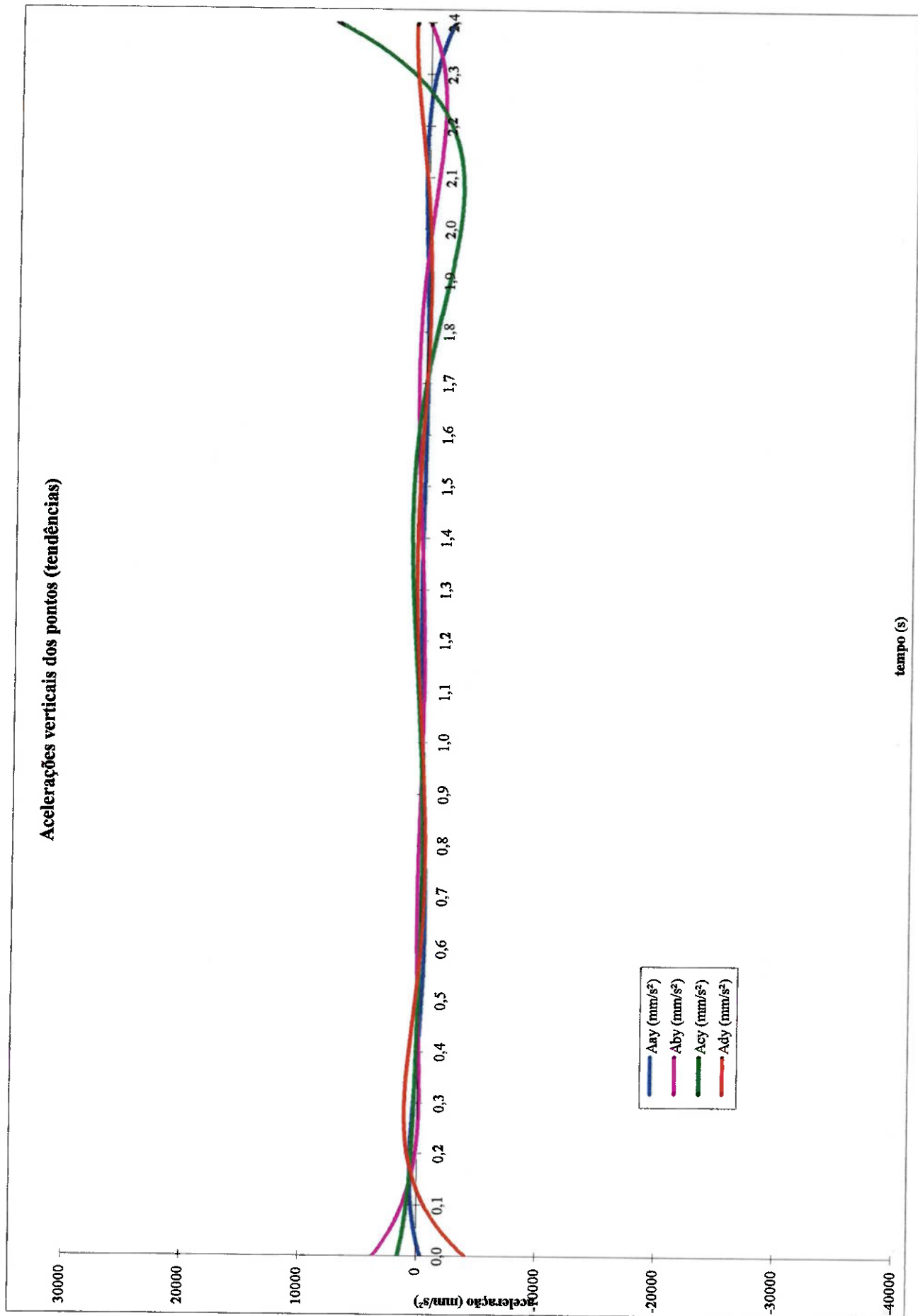
## Acelerações horizontais dos pontos (tendências)



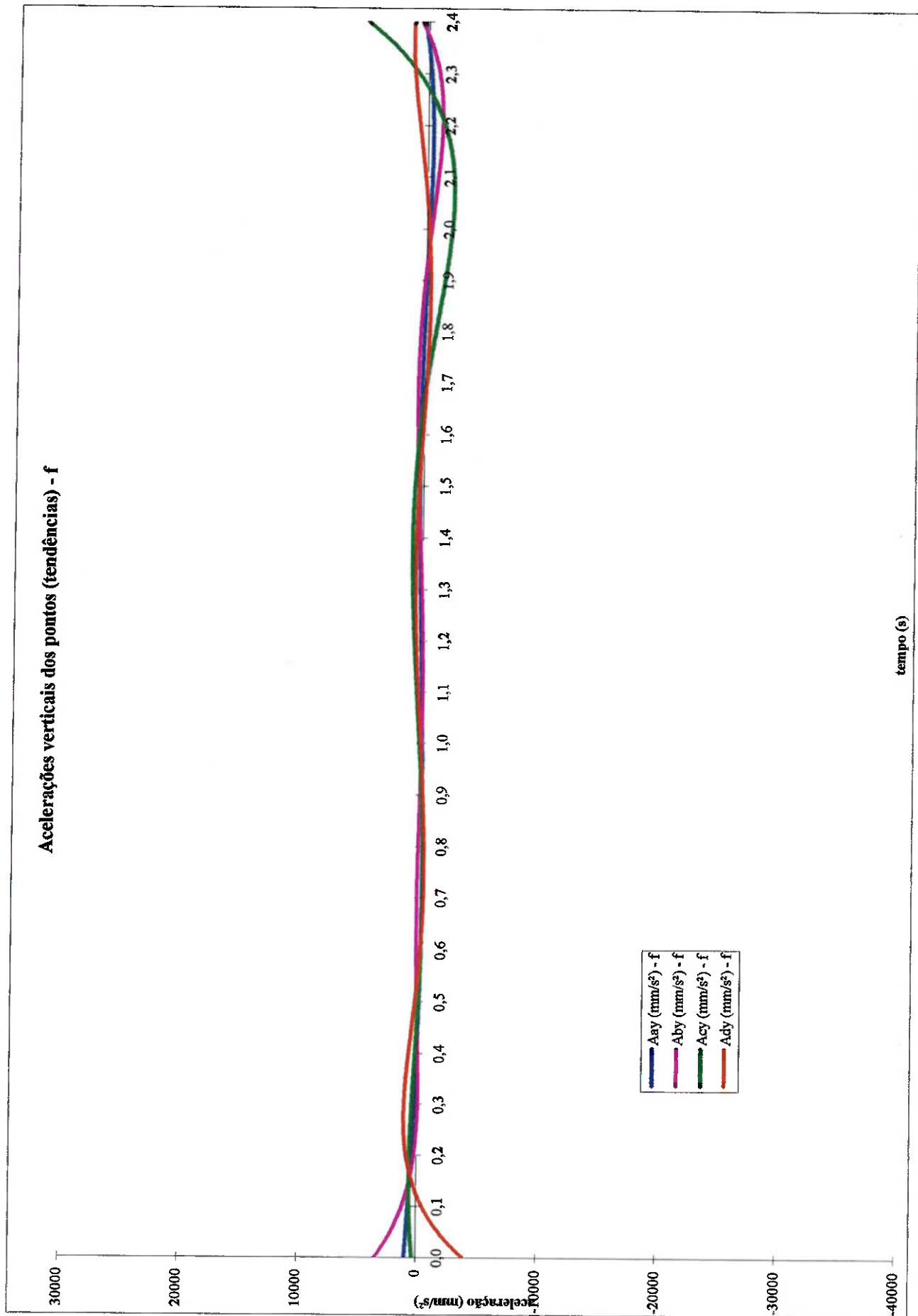
## Acelerações horizontais dos pontos (tendências) - f

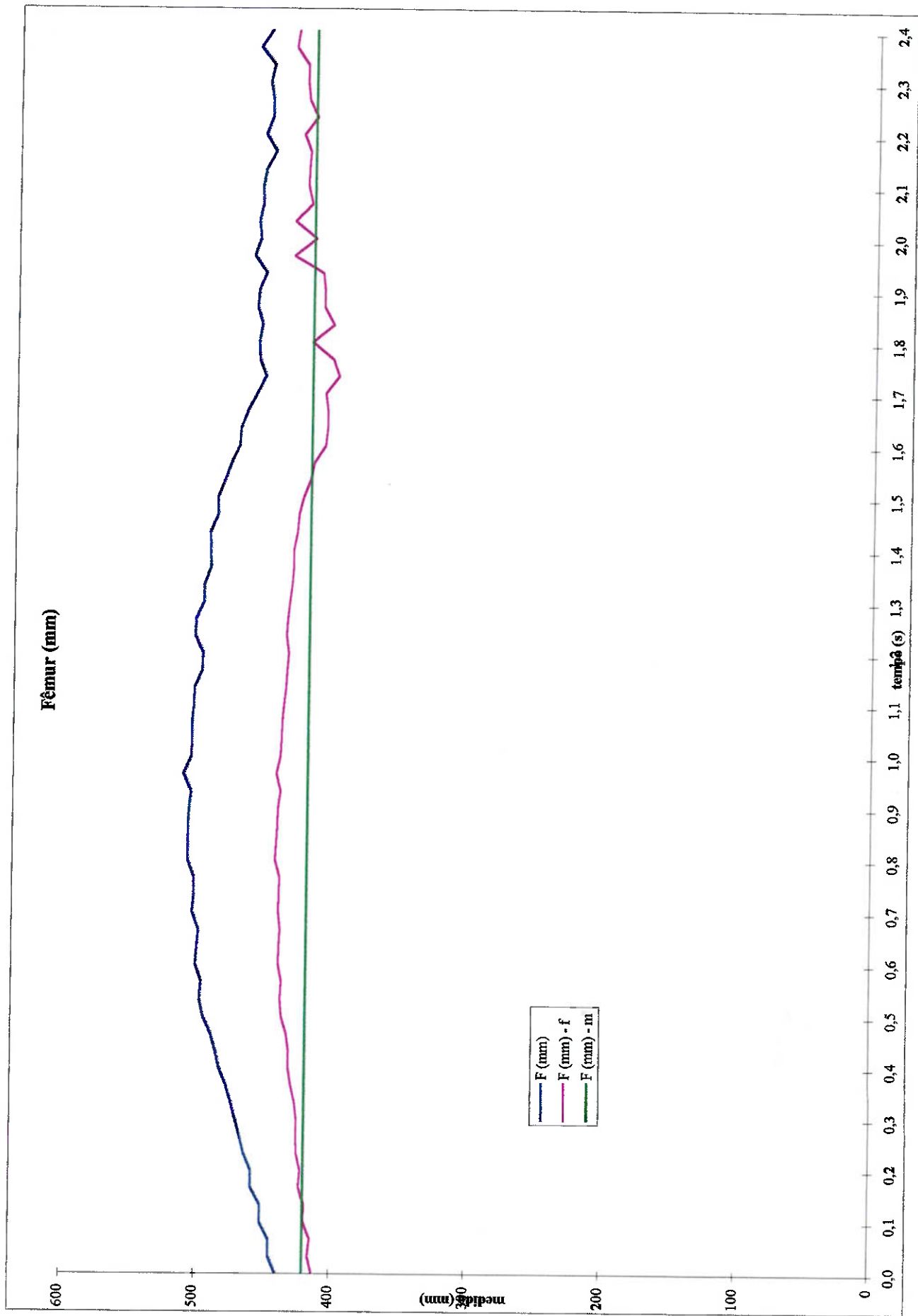


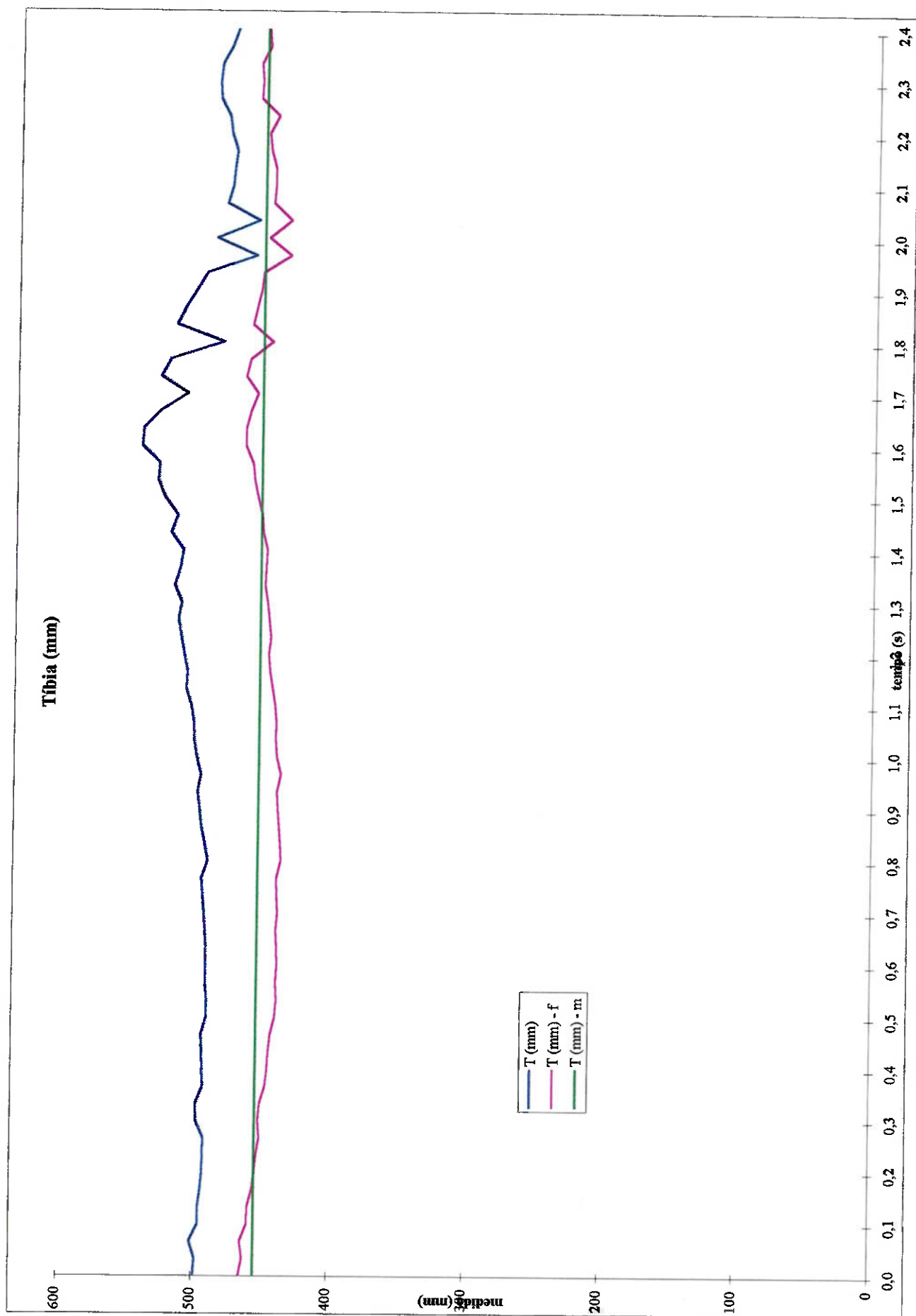
## Acelerações verticais dos pontos (tendências)

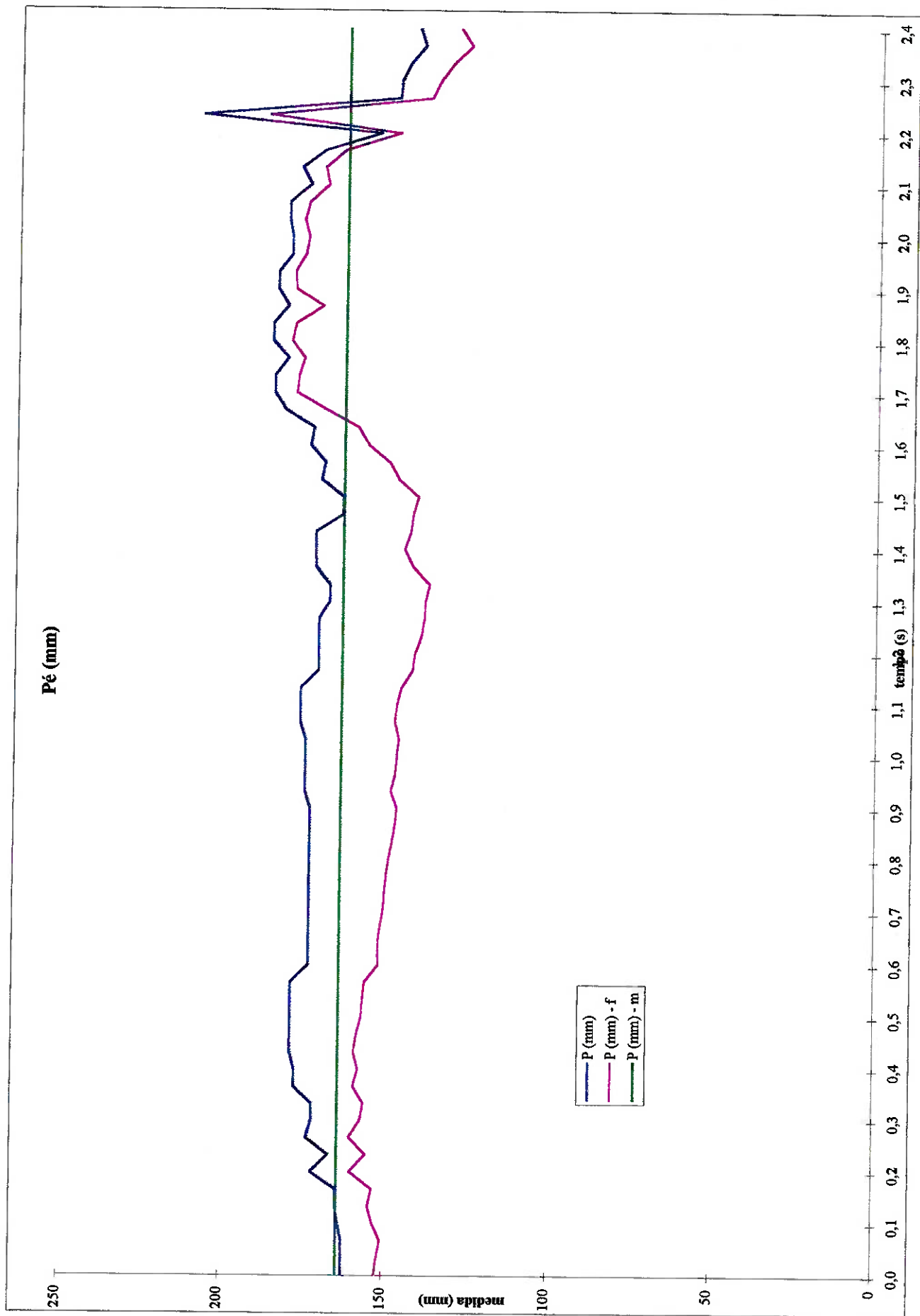


## Acelerações verticais dos pontos (tendências) - f









## ANEXO VI

### CÓDIGO FONTE DAS PRINCIPAIS PROCEDURES DOS APLICATIVOS DELPHI

#### a. Conversor AVI para BMPs (Delphi 3)

```
{
  This unit contains a form, components and procedures to select part
  of an AVI file and convert it to a sequence of BMP files.

  Written by:
    Luciano B. Chieco      - luchienco@hotmail.com
    Dirceu S.A. Azevedo    - dirceu.azevedo@hotmail.com
  As part of a graduation project for the
    Escola Politecnica da Universidade de Sao Paulo
  At
    Sao Paulo, Brazil, 1998
}

{ Carrega o avi no MediaPlayer, habilita os componentes e atualiza
  alguns dados do avi }
procedure TMainForm.LoadAvi (arqName : string);
begin
  if FileExists (arqname) then
  begin
    with MediaPlayer1 do
    begin
      Close;
      FileName := arqname;
      DeviceType := dtAviVideo;
      Display := AviPanel;
      Open;
      TimeFormat := tfFrames;
      Frames := 1;
      LengthLabel.Caption := inttostr(Length);
      TrackBar1.Enabled := true;
      Label1.Enabled := true;
      Label2.Enabled := true;
      LengthLabel.Enabled := true;
      CurrentPosLabel.Enabled := true;
      StartCheckBox.Enabled := true;
      EndCheckBox.Enabled := true;
      TrackBar1.Max := Length;
    end; { with MediaPlayer1 }
  end; { if FileExists }
end; { proc LoadAvi }

{ Atualiza os componentes que indicam a posicao atual do arquivo avi }
procedure TMainForm.ShowPositionStatus;
begin
  TrackBar1.Position := (MediaPlayer1.Position);
  CurrentPosLabel.Caption := inttostr(MediaPlayer1.Position);
  StartCheckBox.Checked := (MediaPlayer1.Position = TrackBar1.SelStart);
  EndCheckBox.Checked := (MediaPlayer1.Position = TrackBar1.SelEnd);
end; { proc ShowPositionStatus }

{ Dado um nome de arquivo (com o caminho completo e com apenas 3 letras)
  e um numero inteiro (ate 5 digitos), copia o frame (que deve estar parado
  no MediaPlayer) para o componente TImage e salva a imagem deste componente
  num arquivo formato BMP 24 bits com o nome e numeracao dados }
procedure TMainForm.SaveImage (BaseFileName : string; ImageNumber : integer);
var
  SourceRect, DestRect : TRect;
  s : string[5];
begin
  Image1.Height := AviPanel.Height;
  Image1.Width := AviPanel.Width;
  with Image1.Canvas do
  begin
    CopyMode := cmSrcCopy;
```

```

SourceRect := Rect (AviPanel.Left, AviPanel.Top,
                    AviPanel.Left+AviPanel.Width,
                    AviPanel.Top+AviPanel.Height);
DestRect := Rect(0, 0, Image1.Width, Image1.Height);
CopyRect(DestRect, MainForm.Canvas, SourceRect);
end;
s := inttostr(imageNumber);
while length(s) < 5 do
  s := '0' + s;
Image1.Picture.SaveToFile (BaseFileName + s + '.bmp');
end;

{ A cada intervalo de tempo atualiza os componentes que indicam a
  posicao atual do avi (quanto mais breve o intervalo, mais realistas
  as indicacoes de posicao) }
procedure TMainForm.Timer1Timer(Sender: TObject);
begin
  if MediaPlayer1.Mode = mpPlaying then
    ShowPositionStatus;
end;

{ Quando o usuario muda a posicao da TrackBar, o arquivo avi vai para a
  posicao indicada no TrackBar }
procedure TMainForm.TrackBar1Change(Sender: TObject);
begin
  MediaPlayer1.Position := TrackBar1.Position;
  ShowPositionStatus;
end;

{ Le o arquivo ini cujo nome e' dado como constante no cabeçalho desta unit
  e que deve conter o nome base dos arquivos BMP que serao salvos em sequencia.
  Este nome base deve estar na primeira linha do arquivo ini e deve conter
  o caminho completo mais um nome com 3 letras, sem extensao (Ex:
  "c:\imagens\img", sem aspas). Dado este nome, salva numa serie de
  BMPs a parte do arquivo avi definida pelas checkboxes de inicio e fim.
  Para tanto, percorre um loop de inicio ate fim, colocando o arquivo avi
  na posicao do loop, e chamando a funcao que salva o frame num BMP }
procedure TMainForm.SaveBtnClick(Sender: TObject);
var
  i : integer;
  Arq : textfile;
  ArqIniName, BaseFileName : string;
begin
  ArqIniName := ExtractFilePath (Application.ExeName);
  while ArqIniName[length(ArqIniName)] = '\' do
    ArqIniName := Copy (ArqIniName, 1, length(ArqIniName)-1);
  ArqIniName := ArqIniName + '\' + cgArqIniName;

  if FileExists (ArqIniName) then
    begin
      AssignFile (Arq, ArqIniName);
      Reset (Arq);
      try
        readln (Arq, BaseFileName);
      finally
        CloseFile (Arq);
      end; { try .. finally }

      if BaseFileName = '' then
        Application.MessageBox ('Nome base dos arquivos de imagens está nulo',
                                'Erro', MB_ICONSTOP)
      else
        begin
          if Application.MessageBox ('Confirma?', 'Confirmação', MB_YESNO) = mrYes then
            begin
              gfCancel := false;
              IniLabel.Caption := inttostr(TrackBar1.SelStart);
              EndLabel.Caption := inttostr(TrackBar1.SelEnd);
              i := TrackBar1.SelStart;
              MediaPlayer1.Position := i;
              while i <= TrackBar1.SelEnd do
                begin
                  if gfCancel then Break;
                  Gauge1.Progress := (i - TrackBar1.SelStart) * 100 div
                                (TrackBar1.SelEnd - TrackBar1.SelStart);
                  CurrentLabel.Caption := inttostr(MediaPlayer1.Position);

                  MediaPlayer1.Step;
                  MediaPlayer1.Display := AviPanel;
                end;
              i := i + 1;
            end;
          else
            gfCancel := true;
          end;
        end;
    end;
end;

```

```

WaitDelay (1000);
SaveImage (BaseFileName, i - TrackBar1.SelStart);
i := i + 1;
end;
end; { if Application.MessageBox (Confirmação )
Application.MessageBox ('Arquivos salvos', 'Informação', MB_ICONINFORMATION);
end; { else (if BaseFileName <> '') }
end
else
Application.MessageBox('Arquivo INI inexistente', 'Erro', MB_OK);
end;

```

## b. Identificador de Padrões em BMPs

```

{
This unit contains a form, components and procedures to apply a
threshold filtering on a BMP file and search for 4 markers in this image
or in a sequence os BMP files, producing a TXT file as results output.

```

Written by:

Luciano B. Chieco - luchieco@hotmail.com

Dirceu S.A. Azevedo - dirceu\_azevedo@hotmail.com

As part of a graduation project for the

Escola Politecnica da Universidade de Sao Paulo

At

Sao Paulo, Brazil, 1998

```

}

const

```

```

MAXMASK = 9;

```

```

cgResolucao = 0.179; { px / mm, da camera }

```

```

cgF = 75.001; { em pixels, para aplicar como restricao no filtro }

```

```

cgT = 81.266; { em pixels, para aplicar como restricao no filtro }

```

```

cgP = 29.356; { em pixels, para aplicar como restricao no filtro }

```

```

type

```

```

TMask = array[0..MAXMASK, 0..MAXMASK] of byte;

```

```

implementation

```

```

{ Checa se a imagem dada no OriginalImgEdit e' valida (BMP 256 cores)
e, caso afirmativo, le sua largura e altura (pixels) }

```

```

function TMainForm.LoadImage : boolean;

```

```

var

```

```

Str : TMemoryStream;

```

```

ImgType, BitsPerPixel : Word;

```

```

begin

```

```

Result := false;

```

```

Str := TMemoryStream.Create;

```

```

try

```

```

Str.LoadFromFile (OriginalImgEdit.Text);

```

```

Str.Read (ImgType, 2);

```

```

Str.Position := 28;

```

```

Str.Read (BitsPerPixel, 2);

```

```

if ((ImgType = (Ord('B') + 256*Ord('M'))) and (BitsPerPixel = 8)) then

```

```

begin

```

```

Str.Position := 10;

```

```

Str.Read (OffsetToData, 4);

```

```

Str.Position := 18;

```

```

Str.Read (ImgWidth, 4);

```

```

Str.Read (ImgHeight, 4);

```

```

SizeLabel.Caption := inttostr(ImgHeight) + ' x ' + inttostr(ImgWidth);

```

```

Str.Position := 0;

```

```

Result := true;

```

```

end

```

```

else

```

```

Application.MessageBox ('Imagem formato inválido.', 'Erro', MB_ICONSTOP);

```

```

finally

```

```

Str.Free;

```

```

end;

```

```

end; { proc LoadImage }

```

```

{ Dada uma posicao da Stream que contem o arquivo BMP, calcula as
coordenadas do ponto correspondente na imagem }

```

```

function TMainForm.GetXY (StrPosit : longint) : TPoint;

```

```

begin

```

```

Result.X := (StrPosit - OffsetToData) mod ((ImgWidth div 4) * 4 + (ImgWidth mod 4));

```

```

    Result.Y := (StrPosit - OffsetToData) div ((ImgWidth div 4) * 4 + (ImgWidth mod 4));
end; { func GetXY }

{ Dadas as coordenadas de um ponto na imagem, calcula a posicao correspondente
na Stream que contem o arquivo BMP (inverso da funcao GetXY) }
function TMainForm.GetStrPosit (Point : TPoint) : longint;
begin
    Result := OffsetToData + Point.Y * ((ImgWidth div 4) * 4 + (ImgWidth mod 4)) + Point.X;
end; { func GetStrPosit }

{ Dada uma imagem (Stream) e dois pontos que nela definem um retangulo,
calcula o centroide deste retangulo da imagem }
function GetCG (Stream : TStream; LowerLeft, UpperRight : TPoint) : TPoint;
var
    i, j : integer;
    numx, numy, den : longint;
    Point : TPoint;
    b : byte;
begin
    numx := 0;
    numy := 0;
    den := 0;
    for j := 0 to (UpperRight.y - LowerLeft.y) do
        for i := 0 to (UpperRight.x - LowerLeft.x) do
            begin
                Point.X := LowerLeft.x + i; Point.Y := LowerLeft.y + j;
                Stream.Position := MainForm.GetStrPosit(Point);
                Stream.read (b, 1);
                numx := numx + (b * i);
                numy := numy + (b * j);
                den := den + b;
            end;
        end;
    Result.x := LowerLeft.x + round(numx / den);
    Result.y := LowerLeft.y + round(numy / den);
end;

{ Dado um ponto da imagem, captura da Stream que contem o arquivo BMP
uma matriz nas dimensoes da mascara tendo x,y como canto inferior esquerdo
desta matriz }
function TMainForm.GetMaskFromStr (Stream : TMemoryStream; x, y : integer) : TMask;
var
    i, j : integer;
    Point : TPoint;
    b : byte;
begin
    for j := 0 to MAXMASK do
        for i := 0 to MAXMASK do
            begin
                Point.X := x + i; Point.Y := y + j;
                Stream.Position := GetStrPosit(Point);
                Stream.read (b, 1);
                Result[i][j] := b;
            end;
        end;
    end;

{ Compara duas matrizes (no caso, a mascara e uma porcao da imagem) e
retorna a porcentagem de pixels iguais nas duas. }
function TMainForm.CompareMasks (m1, m2 : TMask) : real;
var
    i, j, Equals : integer;
    f1, f2, f3 : longint;
    f : double;
    a, b : integer;
begin
    Equals := 0;
    for i := 0 to MAXMASK do
        for j := 0 to MAXMASK do
            begin
                if m2[i][j] = m1[i][j] then
                    Equals := Equals + 1;
                end;
            end;
        end;
    Result := Equals / ((MAXMASK+1) * (MAXMASK+1));
end;

{ Aplica thresholding na imagem original e salva-a na imagem tratada.
Para tanto, carrega o arquivo original numa Stream, verifica na Stream
se o arquivo e' valido e percorre a imagem dentro da Stream, comparando

```

```

cada pixel com o valor no ThresholdValueSpin, o que estiver abaixo
muda na propria Stream para preto (0) e o que estiver igual ou acima se
torna branco (255). Salva entao a Stream no arquivo tratado }
procedure TMainForm.ApplyThresButtonClick(Sender: TObject);
var
  Str : TMemoryStream;
  i : longint;
  b, v : byte;
  ImgType, BitsPerPixel : Word;
begin
  v := ThresholdValueSpin.Value;
  Str := TMemoryStream.Create;
  try
    Str.LoadFromFile (OriginalImgEdit.Text);
    Str.Read (ImgType, 2);
    Str.Position := 28;
    Str.Read (BitsPerPixel, 2);
    if ((ImgType = (Ord('B') + 256*Ord('M')) and (BitsPerPixel = 8)) then
    begin
      Str.Position := 10;
      Str.Read (OffsetToData, 4);
      Str.Position := 18;
      Str.Read (ImgWidth, 4);
      Str.Read (ImgHeight, 4);
      SizeLabel.Caption := inttostr(ImgHeight) + ' x ' + inttostr(ImgWidth);
      Str.Position := OffsetToData;
      for i := 1 to ImgWidth*ImgHeight do
      begin
        Str.Read (b, 1);
        if b < v then
          b := 0
        else
          b := 255;
        Str.Position := Str.Position - 1;
        Str.Write (b, 1);
        Gauge1.Progress := i * 100 div (ImgWidth*ImgHeight);
      end;
      Str.SaveToFile (ThresholdedImgEdit.Text);
      ViewThresImgButtonClick (Self);
    end
  else
    Application.MessageBox ('Imagem formato inválido.', 'Erro', MB_ICONSTOP);
  finally
    Str.Free;
  end;
end;

{ Percorre o arquivo tratado para identificar padroes semelhantes com a
mascara. Carrega o arquivo tratado numa stream e rastreia sua imagem num
loop comparando a mascara com a parte da imagem correspondente ao passo do
loop. Acha primeiro o marcador D, depois o C, B e A. Quando acha cada
marcador, verifica se nao e', na verdade, o anterior (pela distancia entre
os centros, que deve, no minimo, ser maior que o diametro do marcador) }
procedure TMainForm.MaskButtonClick(Sender: TObject);
var
  Mask, ImgMask : TMask;
  Stream : TMemoryStream;
  x : longint;
  i, j : integer;
  igualdade : real;
begin
  if ShowMask then
    Form3.Show;
  LoadImage;
  ViewThresImgButtonClick(Self);

  AxEdit.Text := '0'; AyEdit.Text := '0'; AprecLabel.Caption := '0';
  BxEdit.Text := '0'; ByEdit.Text := '0'; BprecLabel.Caption := '0';
  CxEdit.Text := '0'; CyEdit.Text := '0'; CprecLabel.Caption := '0';
  DxEdit.Text := '0'; DyEdit.Text := '0'; DprecLabel.Caption := '0';
  AlphaEdit.Text := '0'; BetaEdit.Text := '0';
  FpxEdit.Text := '0'; TpxEdit.Text := '0'; PpxEdit.Text := '0';
  FmmLabel.Caption := '(0 mm)'; TmmLabel.Caption := '(0 mm)'; PmmLabel.Caption := '(0 mm)';

  FiltroAxEdit.Text := '0'; FiltroAyEdit.Text := '0'; FiltroAprecLabel.Caption := '0';
  FiltroBxEdit.Text := '0'; FiltroByEdit.Text := '0'; FiltroBprecLabel.Caption := '0';
  FiltroCxEdit.Text := '0'; FiltroCyEdit.Text := '0'; FiltroCprecLabel.Caption := '0';
  FiltroDxEdit.Text := '0'; FiltroDyEdit.Text := '0'; FiltroDprecLabel.Caption := '0';
  FiltroAlphaEdit.Text := '0'; FiltroBetaEdit.Text := '0';
  FiltroFpxEdit.Text := '0'; FiltroTpxEdit.Text := '0'; FiltroPpxEdit.Text := '0';

```

```

FiltroFmmLabel.Caption := '(0 mm)'; FiltroTmmLabel.Caption := '(0 mm)'; FiltroPmmLabel.Caption :=
'(0 mm)';

Stream := TMemoryStream.Create;
try
  if LoadMaskFromFile(mask) then
  begin
    Stream.LoadFromFile (ThresholdedImgEdit.Text);
    fCancel := false;
    x := 0;
    for j := 0 to ImgHeight - MAXMASK - 1 do
    begin
      for i := 0 to ImgWidth - MAXMASK - 1 do
      begin
        ImgMask := GetMaskFromStr (Stream,i,j);
        Igualdade := CompareMasks (Mask, ImgMask);

        if Igualdade >= vgTolerancia / 100 then
        begin
          if strtoint(DxEdit.Text) = 0 then
          begin
            SetPoint ('D', i+5, j+5, igualdade);
          end
          else if strtoint(CxEdit.Text) = 0 then
          begin
            if GetDistance(i+5,j+5,strtoint(DxEdit.Text),strtoint(DyEdit.Text)) <= 10 then
              SetPoint ('D', i+5, j+5, igualdade)
            else
              SetPoint ('C', i+5, j+5, igualdade)
            end
          else if strtoint(BxEdit.Text) = 0 then
          begin
            if GetDistance(i+5,j+5,strtoint(DxEdit.Text),strtoint(DyEdit.Text)) <= 10 then
              SetPoint ('D', i+5, j+5, igualdade)
            else if GetDistance(i+5,j+5,strtoint(CxEdit.Text),strtoint(CyEdit.Text)) <= 10 then
              SetPoint ('C', i+5, j+5, igualdade)
            else
              SetPoint ('B', i+5, j+5, igualdade)
            end
          else if strtoint(AxEdit.Text) = 0 then
          begin
            if GetDistance(i+5,j+5,strtoint(BxEdit.Text),strtoint(ByEdit.Text)) <= 10 then
              SetPoint ('B', i+5, j+5, igualdade)
            else
              SetPoint ('A', i+5, j+5, igualdade)
            end
          else
          begin
            if GetDistance(i+5,j+5,strtoint(AxEdit.Text),strtoint(AyEdit.Text)) <= 10 then
              SetPoint ('A', i+5, j+5, igualdade)
            else
              Application.MessageBox ('Achou mais de quatro marcadores', 'Erro',
                MB_ICONSTOP);
            end;
          end;
        end;
        if fCancel then break;
        Gauge1.Progress := x * 100 div ((ImgWidth - MAXMASK - 1)*(ImgHeight - MAXMASK - 1));
        x := x + 1;
        Application.ProcessMessages;
      end;
      if fCancel then break;
    end; { for j }
  end { if LoadFromFile }
  else
    Application.MessageBox ('Erro ao carregar máscara.', 'Erro', MB_ICONSTOP);
  finally
    Stream.Free;
    if CalculaAngulos then
    begin
      AplicaFiltro;
      CalculaAngulosFiltro;
    end; { if CalculaAngulos }
    MessageBeep (MB_ICONQUESTION);
  end;
end;

{ Salva num arquivo texto a frequencia absoluta com que cada nivel de
cinza aparece na imagem. Cada linha do arquivo contera o nivel de cinza,
uma virgula e o numero de pixels da imagem original que tem este nivel }
procedure TMainForm.HistBtnClick(Sender: TObject);
var

```

```

Str : TMemoryStream;
i : longint;
b : byte;
ImgType, BitsPerPixel : Word;
OffsetToData, ImgHeight, ImgWidth : longint;
HistData : array[0..255] of longint;
arq : textfile;
arqname : string;
begin
  arqName := ExtractFilePath (Application.ExeName);
  while arqName[length(arqName)] = '\' do
    arqName := Copy (arqName, 1, length(arqName)-1);
  arqName := arqName + '\histog.txt';

  AssignFile (arq, arqname);
  rewrite (arq);
  for i := 0 to 255 do
    HistData[i] := 0;

  Str := TMemoryStream.Create;
  try
    Str.LoadFromFile (OriginalImgEdit.Text);
    Str.Read (ImgType, 2);
    Str.Position := 28;
    Str.Read (BitsPerPixel, 2);
    if ((ImgType = (Ord('B') + 256*Ord('M')) and (BitsPerPixel = 8)) then
    begin
      Str.Position := 10;
      Str.Read (OffsetToData, 4);
      Str.Position := 18;
      Str.Read (ImgWidth, 4);
      Str.Read (ImgHeight, 4);
      SizeLabel.Caption := inttostr(ImgHeight) + ' x ' + inttostr(ImgWidth);
      Str.Position := OffsetToData;
      for i := 1 to ImgWidth*ImgHeight do
        begin
          Str.Read (b, 1);
          HistData[b] := HistData[b] + 1;
          Gauge1.Progress := i * 100 div (ImgWidth*ImgHeight);
        end;
      for i := 0 to 255 do
        writeln (arq, inttostr(i) + ',' + inttostr(HistData[i]));
      end
    else
      Application.MessageBox ('Imagem formato inválido.', 'Erro', MB_ICONSTOP);
    finally
      CloseFile (arq);
      Str.Free;
      MessageBeep (MB_ICONQUESTION);
    end;
  end;
end;

```

#### c. Filtragem dos Resultados

/\* Programa que determina uma melhor estimativa para os valores das coordenadas cartesianas dos marcadores através do método de minimização de erros pela técnica do multiplicador de Lagrange. O sistema de equações resultante do método é solucionado pelo m\*/

```

#include <math.h>
#define NRANSI
#include "nrutil.h"

```

/\* Função definida pelo usuário para a entrada do sistema de equações correspondente ao método do multiplicador de Lagrange e todo Jacobiano da função \*/

```

void usrfun(float *x, float *xc, float *FTP, int n, float *fvec, float **fjac)
#define FREEReturn (free_matrix(fjac,1,n,1,n); free_vector(fvec,1,n); \
  free_vector(p,1,n); free_ivector(indx,1,n); return;)
{
  fvec[1]=2*(x[1]-xc[1])-2*x[9]*(x[3]-x[1]);
  fvec[2]=2*(x[2]-xc[2])-2*x[9]*(x[4]-x[2]);
  fvec[3]=2*(x[3]-xc[3])+2*x[9]*(x[3]-x[1])-(x[5]-x[3]);
  fvec[4]=2*(x[4]-xc[4])+2*x[9]*(x[4]-x[2])-(x[6]-x[4]);
  fvec[5]=2*(x[5]-xc[5])+2*x[9]*(x[5]-x[3])-(x[7]-x[5]);
  fvec[6]=2*(x[6]-xc[6])+2*x[9]*(x[6]-x[4])-(x[8]-x[6]);
  fvec[7]=2*(x[7]-xc[7])+2*x[9]*(x[7]-x[5]);
  fvec[8]=2*(x[8]-xc[8])+2*x[9]*(x[8]-x[6]);
  fvec[9]=pow((x[3]-x[1]),2)+pow((x[4]-x[2]),2)-pow(FTP[1],2)+
    pow((x[5]-x[3]),2)+pow((x[6]-x[4]),2)-pow(FTP[2],2)+
    pow((x[7]-x[5]),2)+pow((x[8]-x[6]),2)-pow(FTP[3],2);
  fjac[1][1]=2+2*x[9]; fjac[1][2]=0; fjac[1][3]=-
  2*x[9]; fjac[1][4]=0; fjac[1][5]=0; fjac[1][6]=0; fjac[1][7]=0; fjac[1][8]=0; fjac[1][9]=-2*(x[3]-x[1]);

```

```

    fjac[2][1]=0; fjac[2][2]=2+2*x[9]; fjac[2][3]=0; fjac[2][4]=-
2*x[9]; fjac[2][5]=0; fjac[2][6]=0; fjac[2][7]=0; fjac[2][8]=0; fjac[2][9]=-2*(x[4]-x[2]);
    fjac[3][1]=-2*x[9]; fjac[3][2]=0; fjac[3][3]=2+4*x[9]; fjac[3][4]=0; fjac[3][5]=-
2*x[9]; fjac[3][6]=0; fjac[3][7]=0; fjac[3][8]=0; fjac[3][9]=2*((x[3]-x[1])-(x[5]-x[3]));
    fjac[4][1]=0; fjac[4][2]=-2*x[9]; fjac[4][3]=0; fjac[4][4]=2+4*x[9]; fjac[4][5]=0; fjac[4][6]=-
2*x[9]; fjac[4][7]=0; fjac[4][8]=0; fjac[4][9]=2*((x[4]-x[2])-(x[6]-x[4]));
    fjac[5][1]=0; fjac[5][2]=0; fjac[5][3]=-
2*x[9]; fjac[5][4]=0; fjac[5][5]=2+4*x[9]; fjac[5][6]=0; fjac[5][7]=-
2*x[9]; fjac[5][8]=0; fjac[5][9]=2*((x[5]-x[3])-(x[7]-x[5]));
    fjac[6][1]=0; fjac[6][2]=0; fjac[6][3]=0; fjac[6][4]=-
2*x[9]; fjac[6][5]=0; fjac[6][6]=2+4*x[9]; fjac[6][7]=0; fjac[6][8]=-2*x[9]; fjac[6][9]=2*((x[6]-x[4])-(
x[8]-x[6]));
    fjac[7][1]=0; fjac[7][2]=0; fjac[7][3]=0; fjac[7][4]=0; fjac[7][5]=-
2*x[9]; fjac[7][6]=0; fjac[7][7]=2+2*x[9]; fjac[7][8]=0; fjac[7][9]=2*(x[7]-x[5]);
    fjac[8][1]=0; fjac[8][2]=0; fjac[8][3]=0; fjac[8][4]=0; fjac[8][5]=0; fjac[8][6]=-
2*x[9]; fjac[8][7]=0; fjac[8][8]=2+2*x[9]; fjac[8][9]=2*(x[8]-x[6]);
    fjac[9][1]=-2*(x[3]-x[1]); fjac[9][2]=-2*(x[4]-x[2]); fjac[9][3]=2*(x[3]-x[1])-2*(x[5]-
x[3]); fjac[9][4]=2*(x[4]-x[2])-2*(x[6]-x[4]); fjac[9][5]=2*(x[5]-x[3])-2*(x[7]-
x[5]); fjac[9][6]=2*(x[6]-x[4])-2*(x[8]-x[6]); fjac[9][7]=2*(x[7]-x[5]); fjac[9][8]=2*(x[8]-
x[7]); fjac[9][9]=0;
}

```

/\* Rotina responsável pela aplicação do método de Newton-Raphson para a solução de sistemas de equações não-lineares \*/

void mnnewt(int ntrial, float x[], float xc[], float FTP[], int n, float tolX, float tolF)

```

{
    void lubksb(float **a, int n, int *indx, float b[]);
    void ludcmp(float **a, int n, int *indx, float *d);
    int k,i,*indx;
    float errx,errf,d,*fvec,**fjac,*p;

    indx=ivector(1,n);
    p=vector(1,n);
    fvec=vector(1,n);
    fjac=matrix(1,n,1,n);
    for (k=1;k<=ntrial;k++) {
        usrfun(x,xc,FTP,n,fvec,fjac);
        errf=0.0;
        for (i=1;i<=n;i++) errf += fabs(fvec[i]);
        if (errf <= tolF) FREERETURN
        for (i=1;i<=n;i++) p[i] = -fvec[i];
        ludcmp(fjac,n,indx,&d);
        lubksb(fjac,n,indx,p);
        errx=0.0;
        for (i=1;i<=n;i++) {
            errx += fabs(p[i]);
            x[i] += p[i];
        }
        if (errx <= tolX) FREERETURN
    }
    FREERETURN
}
#undef FREERETURN
#undef NRANSI
#define n 9

```

/\* Rotina principal responsável pela interface entre os aplicativos Delphi e Excel, realizada através de transferência de arquivos texto \*/

```

void main(void)
{
    int i,ntrial;
    float tolX, tolF;
    float *x, *xc, *FTP;
    FILE *arq1;
    FILE *arq2;
    x=vector(1,n);
    xc=vector(1,n-1);
    arq1=fopen("coord.ent","r");
    arq2=fopen("coord.out","w");
    for(i=1; i<4; i++)
        fscanf(arq1,"%f", &FTP[i]);
    for(i=1; i<9; i++)
        fscanf(arq1,"%f", &xc[i]);

    x[1]=xc[1];
    x[2]=xc[2];
    x[3]=xc[3];
    x[4]=xc[4];
    x[5]=xc[5];
    x[6]=xc[6];
    x[7]=xc[7];
    x[8]=xc[8];
}

```

```
x[9]=0;
tolx=0.0001;
tolf=0.0001;

mnewt(ntrial, x, xc, FTP, n, tolx, tolf);
for (i=1;i<=n;i++) fprintf(arq2, "%5d %12.6f\n",i,x[i]);
free_vector(xc,1,n);
free_vector(x,1,n);
}
/* (C) Copr. 1986-92 Numerical Recipes Software . */
```

## ANEXO VIII

## TELAS DOS PROGRAMAS

## a. Conversor AVI para BMPs

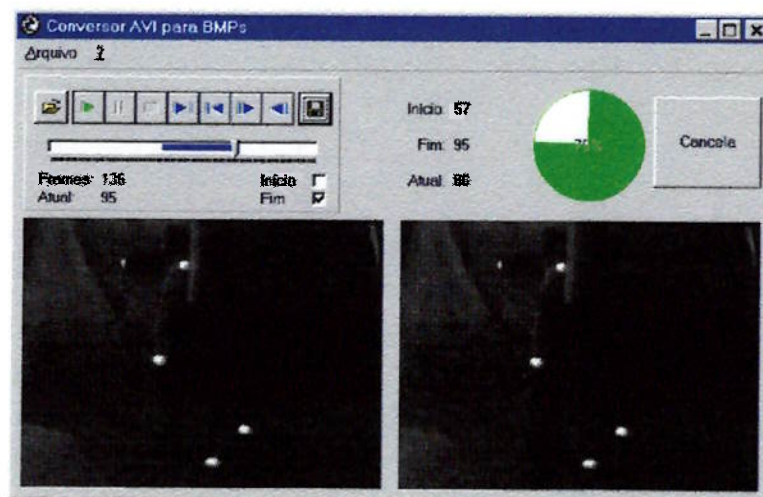


FIGURA 14 - TELA DO PROGRAMA CONVERSOR AVI PARA BMPs

## b. Identificador de Padrões em BMPs

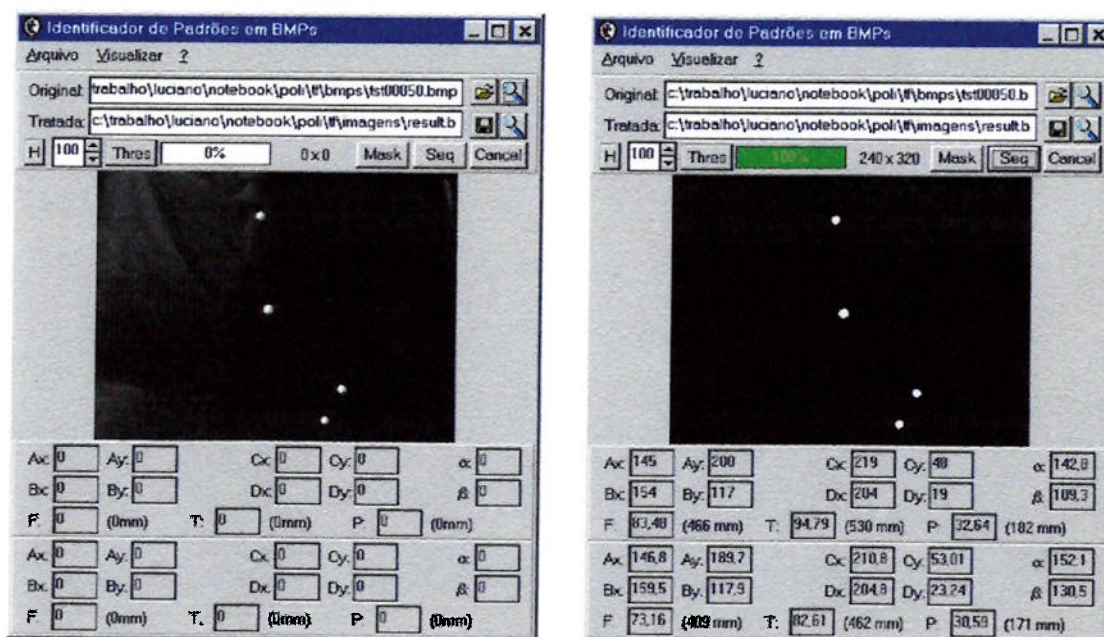


FIGURA 15 - TELA DO IDENTIFICADOR DE PADRÕES EM BMPs (ANTES/DEPOIS DE PROCESSAR A IMAGEM)

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Gonzalez, R.C.; Wintz, P. Digital Image Processing. 2.ed. Addison-Wesley Publishing Company.
- [2] Hull, D.G. Optimal Control Theory. 1.ed. The University of Texas at Austin, Department of Aerospace Engineering and Engineering Mechanics, 1990.
- [3] Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. Numerical Recipes in C, The Art of Scientific Computing. 2.ed. Cambridge University Press, 1992.
- [4] Saad, M. A Importância da Análise de Marcha. Sociedade Paulista para Desenvolvimento da Medicina.
- [5] Saad, M. Técnicas de Análise de Marcha. Sociedade Paulista para Desenvolvimento da Medicina.
- [6] Allard, P.; Stokes, I.A.F.; Blanche, J.P. Three-Dimensional Analysis of Human Movement. Optoelectronic-Based Systems. Human Kinetics.
- [7] Ferrigno, G.; Pedotti, A. ELITE: A Digital Dedicated Hardware System for Movement Analysis Via Real-Time TV Signal Processing. IEEE Transactions on Biomedical Engineering, vol. BME-32 no. 11, November 1985.