



*Universidade Federal de Santa Catarina  
Centro Tecnológico  
Curso de Engenharia de Controle e  
Automação Industrial*



# ***Implementação da Estrutura de Controle de um Punho de Manipulador***

*Monografia submetida à Universidade Federal de Santa Catarina  
como requisito para a aprovação da disciplina:  
**EEL 5901: Projeto de Fim de Curso***

***Jean do Amaral Lima***

*Florianópolis, Dezembro de 1995*

***Implementação da Estrutura de Controle de um Punho de Manipulador***

***Jean do Amaral Lima***

Esta monografia foi julgada no contexto da disciplina  
**EEL 5901: Projeto de Fim de Curso**  
e aprovada na sua forma final pelo  
**Curso de Engenharia de Controle e Automação Industrial**

Banca Examinadora:

Prof. Claudio Von Dokonal  
Orientador, Escola Técnica Tupy

Prof. Raul Guenther  
Orientador, UFSC

Prof. Augusto Humberto Bruciapaglia  
Responsável pela disciplina e Coordenador do Curso

Prof. Edson de Pieri, Avaliador

aluno Elmo D. Martins, Debatedor

aluno Richardi Fonseca, Debatedor

## **AGRADECIMENTOS**

Agradeço:

- Aos orientadores deste trabalho, professores Raul e Claudio, pela atenção e dedicação prestados ao longo de todo desenvolvimento deste trabalho.
- Aos professores do setor de automação da Escola Técnica Tupy, pela grande ajuda prestada na adaptação ao ambiente de trabalho.
- Aos meus amigos Elmo, Henrique, Richard e Rodolfo, pelo companheirismo e união estabelecidos ao longo de todo estágio.
- Aos professores Augusto e Vitório, pela importante ajuda prestada no início do desenvolvimento deste trabalho.
- Aos meus pais, Arnaldo e Maria Helena, por todo apoio e carinho, os quais viabilizaram o desenvolvimento deste trabalho fora da minha cidade.

## SUMÁRIO

AGRADECIMENTOS .....	2
RESUMO .....	5
ABSTRACT .....	6
CAPÍTULO 1 - INTRODUÇÃO .....	7
CAPÍTULO 2 - CONCEITOS PRELIMINARES .....	9
CAPÍTULO 3 - EQUIPAMENTOS .....	11
3.1 - Introdução .....	11
3.2 - Servoacionamentos brushless .....	11
3.2.1 - Servomotores CA Brushless .....	12
3.2.2 - Servoconversores .....	12
3.2.3 - Encoders .....	12
3.3 - Placas de Aquisição de Sinais .....	13
3.4 - Microcomputador .....	14
3.5 - Conclusão .....	15
CAPÍTULO 4 - SOFTWARE GERENCIADOR .....	16
4.1 - Introdução .....	16
4.2 - Leitura dos Sinais dos Encoders .....	16
4.2.1 - Algoritmo de Leitura .....	16
4.2.2 - Problema com a Taxa de Leitura .....	17
4.2.3 - Solução .....	18
4.2.4 - Resultados .....	18
4.3 - Marcação de Pontos no Espaço de Juntas .....	19
4.3.1 - Marcação de Pontos Via Teclado .....	20

4.4 - Gerador de Trajetória .....	20
4.4.1 - Considerações sobre a Descrição e Geração da Trajetória .....	20
4.4.2 - Geração de Trajetória Utilizando Polinômios Cúbicos .....	21
4.4.3 - Pontos Intermediários com Especificação de Velocidade .....	23
4.4.4 - Escolha da Velocidade nos Pontos Marcados .....	25
4.4.4.1 - Colocando Velocidade Fixa nos Pontos Intermediários .....	25
4.4.4.2 - Eliminação do Overshoot dos Perfis de Trajetória .....	28
4.4.5 - Cálculo da Velocidade Máxima e Aceleração Máxima .....	31
4.4.6 - Obtenção do Menor Tempo de Segmento .....	32
4.5 - Outras Opções do Software Gerenciador .....	34
4.5.1 - Gravação de Pontos em Arquivo .....	35
4.5.2 - Plotagem de Sinais .....	35
4.6 - Visão Geral do Software Gerenciador .....	35
4.7 - Conclusão .....	36
<b>CAPÍTULO 5 - CONTROLE DO SEGUIMENTO DE TRAJETÓRIA .....</b>	<b>38</b>
5.1 - Introdução .....	38
5.2 - Considerações sobre o Controle do Seguimento de Trajetória .....	38
5.3 - Estrutura de Controle .....	39
5.3.1 - Cálculo dos Ganhos $K_p$ e $K_{pp}$ .....	40
5.4 - Resultados do Seguimento de Trajetórias para Diferentes Valores de $K_C$ .....	42
5.5 - Aumento da Suavidade do Sinal de Controle .....	46
5.6 - Conclusão .....	48
<b>CAPÍTULO 6 - CONCLUSÃO .....</b>	<b>49</b>
<b>CAPÍTULO 7 - PERSPECTIVAS .....</b>	<b>50</b>
<b>BIBLIOGRAFIA .....</b>	<b>51</b>

## **RESUMO**

Este projeto procura resolver o problema de controle de seguimento de trajetória para um punho de manipulador com três graus de liberdade.

Inicialmente são apresentadas todas as características dos equipamentos que foram utilizados na estrutura de controle do punho.

Posteriormente, encontra-se a descrição do desenvolvimento, problemas e soluções de todas as tarefas executadas pelo software responsável pelo gerenciamento do sistema. Dentre as tarefas executadas pelo software, destacamos a geração dos perfis de trajetória a serem seguidos pelo punho do manipulador. É apresentado o método utilizado para geração desses perfis, bem como as otimizações realizadas com o objetivo de melhorar a performance do sistema.

Encontra-se também, a descrição da estrutura de controle montada e a definição da lei de controle utilizada para garantir que o punho do manipulador siga uma determinada trajetória.

Por fim, são apresentados os resultados obtidos de seguimento de trajetória com a lei de controle estabelecida.

## **ABSTRACT**

This project tries to solve the problem of trajectory following control for a robot wrist with three degrees of freedom.

At first, all the characteristics of the equipments used in the wrist control structure are presented.

Afterwards, we find the description of the development, problems and solutions of all the tasks executed by the software responsible for the management of the system. Among the tasks executed by the software, we detach the creation of the profiles trajectory that has to be followed by the robot wrist. The method used for this profiles creation is presented, such as the improvements realized with the purpose of improving the performance of the system.

we also find the control mounted structure description and the definition of the controller used to guarantee that the robot wrist follow a determined trajectory.

At last, we present the results of trajectory following gained with the controller established.

# CAPÍTULO 1

## INTRODUÇÃO

O desenvolvimento da robótica é crescente, e crescente também é a corrida que se trava atrás dessa tecnologia por parte das instituições de desenvolvimento e pesquisa.

A robótica é um importante ramo da automação industrial e a sua integração em processos modernos de manufatura está se tornando cada vez mais necessária.

A idéia de projetar e implementar um manipulador industrial vem a ser, ao mesmo tempo, arrojada e estratégica, pois temos a consciência da quantidade e complexidade das barreiras que devem ser superadas até chegarmos a um nível de conhecimento relevante, porém, tais conhecimentos estão sendo cada vez mais valorizados devido a crescente procura por parte daqueles que utilizam os serviços oferecidos pela robótica.

Com esse pensamento, a Escola Técnica Tupy resolveu, pela primeira vez, investir em um projeto desse tipo, sendo que os seus setores de mecânica e automação serviram como alicerces para a tomada desta decisão.

Para o setor de mecânica foi designada a tarefa de projeto e construção de toda a parte mecânica do manipulador. Para o setor de automação coube projetar e implementar a estrutura de controle do manipulador.

Este trabalho teve como principais objetivos:

- Aquisição de importantes conhecimentos em robótica pela equipe envolvida no projeto.
- Incrementar a infra-estrutura de ensino da Escola Técnica Tupy, já que futuros trabalhos poderão ser realizados utilizando o manipulador.
- Promover uma maior integração dos setores de mecânica e automação através do envolvimento e esforço de técnicos e engenheiros de ambos os setores a fim de atingir um objetivo comum. Como consequência, esta interação estimulará a realização de futuros trabalhos em parceria, aumentando, deste modo, a qualidade dos mesmos.
- Projetar e implementar um sistema de controle para um punho de manipulador com três graus de liberdade.

Neste relatório constam as atividades desenvolvidas no setor de automação no período de 08/08/95 a 31/11/95 dentro do projeto de implementação da estrutura de controle de um punho de manipulador.

O fato de tentarmos concluir primeiro a estrutura de controle e a parte mecânica do punho se fundamenta na existência de uma maior complexidade no projeto mecânico do mesmo, sendo que, com a conclusão desta etapa o término do corpo do manipulador se tornará mais fácil.

Este relatório está organizado da seguinte maneira:

**Capítulo 2 (Conceitos Preliminares):** Onde encontram-se os conceitos de alguns termos que serão utilizados ao longo de todo relatório.

**Capítulo 3 (Equipamentos):** É feita a descrição funcional de todos os equipamentos que foram utilizados no projeto.

**Capítulo 4 (Software Gerenciador):** Contém a descrição de todos os módulos que compõem o software que gerencia o sistema de controle do punho do manipulador. Algoritmos implementados, problemas e soluções também fazem parte deste capítulo.

**Capítulo 5 (Controle do Seguimento de Trajetória):** Onde é definida a estrutura e a lei de controle que garantirão que o punho do manipulador realizará a trajetória definida pelo usuário do sistema.

**Capítulo 6 (Conclusão):** São apresentadas as conclusões deste trabalho.

**Capítulo 7 (Perspectivas):** São apresentadas as perspectivas de continuidade deste projeto.

## CAPÍTULO 2

### CONCEITOS PRELIMINARES

Antes de relatarmos as atividades desenvolvidas neste trabalho, é importante definir alguns termos que irão ser utilizados ao longo deste relatório.

⇒ **Juntas Prismáticas:** Tipo de junta que quando acionada produz movimento linear. Veja figura 2.1.

⇒ **Juntas Rotacionais:** Tipo de junta que quando acionada produz movimento de rotação. Veja figura 2.1.

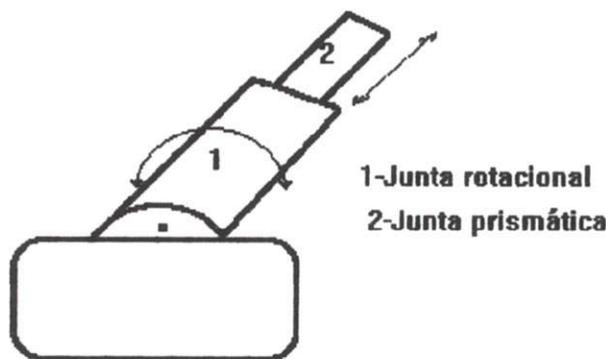


fig.2.1- tipos de junta

⇒ **Graus de Liberdade:** Graus de liberdade de um manipulador é o número de movimentos independentes que o mesmo pode realizar e é igual ao número de juntas presentes no manipulador.

⇒ **Espaço de Juntas:** O mesmo que sistema de coordenadas de juntas. É representado através de valores de ângulos (para juntas rotacionais) ou dimensões (para juntas prismáticas).

⇒ **Espaço de Trabalho:** É definido por um sistema de coordenadas cartesiano. As coordenadas são representadas por três valores (X,Y,Z).

⇒ **Volume de Trabalho:** Volume de trabalho de um manipulador representa o volume pelo qual todos os pontos do seu interior podem ser atingidos pelo efetuador do manipulador. Veja figura 2.2.

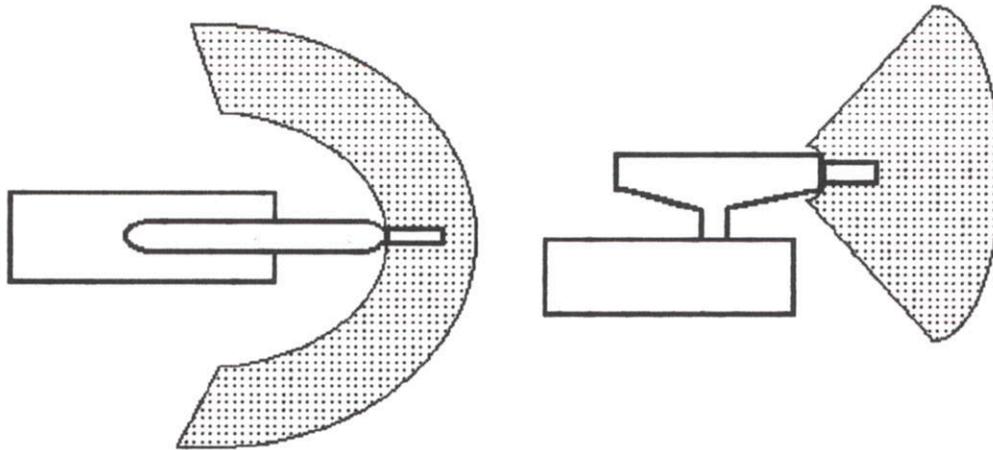


fig.2- Volume de trabalho de um manipulador esférico

⇒ **Harmonic Drive:** Elemento de transmissão mecânica utilizado em juntas rotacionais. Suas principais vantagens são:

- \_ Folga de engrenagem (backlash) próxima de zero.
- \_ Altamente compacto.
- \_ Grande relação de transmissão.

⇒ **Encoder:** Sensor de posição utilizado no eixo do servomotor. Possui um disco ranhurado sobre o qual incide um feixe de luz. A passagem de luz pelas ranhuras do disco resulta em um trem de pulsos os quais informam a posição do eixo do servomotor.

⇒ **Resolução dos Encoders:** Podemos definir resolução, com relação a encoders, como sendo o ângulo pelo qual resulta na transmissão de um pulso, isto é, o menor ângulo que o encoder pode medir. A resolução depende da densidade de ranhuras no disco do encoder, onde, quanto maior for o número de ranhuras, e mais próximas elas estiverem, melhor será a resolução.

⇒ **Faixa dos Encoders:** A faixa de um encoder é definida como o máximo valor de ângulo que o encoder pode medir.

# CAPÍTULO 3

## EQUIPAMENTOS

### 3.1. INTRODUÇÃO

Este capítulo descreve todos os equipamentos que foram utilizados no projeto. O objetivo é nos situarmos perante suas características de funcionamento bem como a função que cada um vai exercer dentro da estrutura de controle do punho do manipulador.

### 3.2. SERVOACIONAMENTOS BRUSHLESS

Estes sistemas são fabricados pela empresa *WEG AUTOMAÇÃO* e são compostos pôr servomotores CA Brushless, servoconvertores e encoders. A figura 3.1 mostra a configuração desses servoacionamentos e a seguir encontra-se a descrição dos três componentes que o formam separadamente.

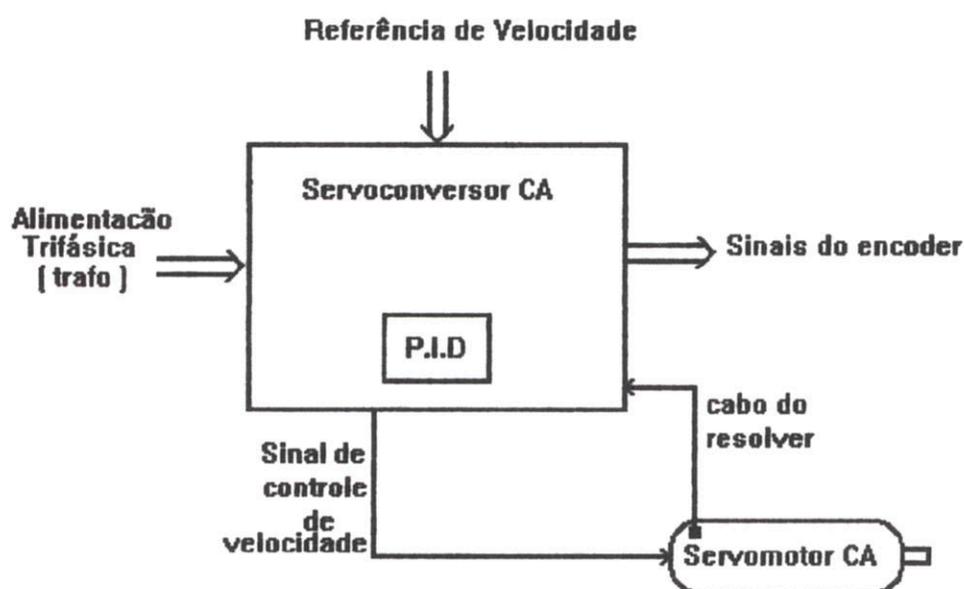


fig.3.1 - Visão Geral do Servoacionamento

### 3.2.1. Servomotores CA Brushless

São utilizados como atuadores no sistema de controle, ou seja, os servomotores recebem o sinal de controle vindo do controlador P.I.D situado no seu respectivo servoconversor e produzem o torque que movimentará o punho do manipulador.

Esses servomotores apresentam importantes vantagens em relação a outros tradicionais atuadores utilizados em manipuladores (servomotores CC, motores de passo, motores hidráulicos, motores pneumáticos, etc.). As principais vantagens são:

- Resposta dinâmica rápida;
- Baixo momento de inércia;
- Manutenção reduzida;
- Torque nominal em toda faixa de velocidade;
- Baixo nível de ruído;
- Rotação suave e uniforme mesmo em baixas velocidades (podemos considerar a ausência de zona-morta).

Abaixo encontram-se os dados técnicos dos servomotores.

- Massa : 4,5 Kg;
- Rotação nominal : 2000 rpm;
- Torque nominal : 2 Nm;
- Momento de Inércia :  $0,22 \times 10^{-3} \text{ Kg m}^2$ ;

### 3.2.2. Servoconversores

Possuem um controlador tipo P.I.D onde pode-se ajustar seus parâmetros proporcional, integral e derivativo através de potenciômetros.

Tal controlador regula a velocidade do eixo do servomotor através de realimentação via resolver (sensor de velocidade), onde um sinal analógico de referência de velocidade constitui uma das entradas.

Esses servoconversores possuem como saídas sinais de encoder para regulação de posição (veja a seguir a descrição deste componente).

Abaixo encontram-se os dados técnicos dos servomotores.

- Sinal de referência de velocidade :  $\pm 10 \text{ Volts}$ ;
- Alimentação trifásica : 100 a 200 Volts.

### 3.2.3. Encoders

Componente que fornece a informação de posição do eixo do servomotor através de trens de pulsos (sinais digitais), oriundos da passagem ou não de luz por um disco ranhurado.

Em geral existem dois tipos de encoders : absolutos e incrementais.

Encoders absolutos tem como saídas sinais digitais que representam a posição absoluta do eixo do servomotor, ou seja, fornecem um código distinto para cada posição e podem ser lidos diretamente, porém, possuem uma *faixa* limitada de indicação de posição e são mais caros que os encoders incrementais.

Encoders incrementais fornecem trens de pulsos, sinal A e B da figura 3.2, para uma dada quantidade de variação de posição.

Os encoders incrementais são mais baratos que os absolutos e possuem uma *faixa* ilimitada de indicação de posição.

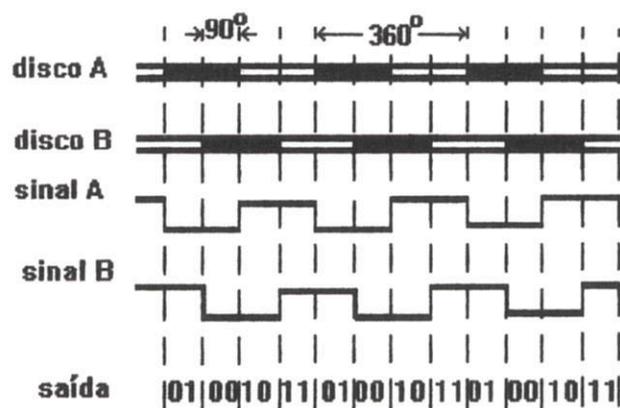


fig.3.2 - Sinais de encoder incremental

O encoder presente no servoacionamento é do tipo incremental e como sua informação de posição é não absoluta, tivemos que desenvolver um algoritmo para leitura de posição do eixo do servomotor. Este algoritmo está descrito no capítulo 4.

Abaixo encontram-se os dados técnicos dos encoders.

- Número de pulsos pôr giro do eixo do motor : 500;
- $Resolução = \frac{circunferencia\ do\ disco\ (graus)}{número\ de\ pulsos} = \frac{360°}{500} = 0,72\ graus/pulso;$
- *Faixa* = ilimitada.

### 3.3. PLACAS DE AQUISIÇÃO DE SINAIS

Devido a necessidade de leitura de sinais digitais, oriundos dos encoders, e envio de sinais analógicos, referentes ao controle de posição, tivemos que adquirir respectivamente, uma placa de entrada/saída digital (placa E/S) e uma placa conversora de sinais digitais para analógicos (placa D/A).

Tais placas permitem a integração de um microcomputador com o servoconversor descrito anteriormente, fazendo com que seja possível realizar todas as operações necessárias para o desejado funcionamento do sistema de controle do punho do manipulador. Essas operações são realizadas através de um software gerenciador que será descrito no próximo capítulo.

Dados técnicos da placa D/A :

- 8 bits;
- amplitude das saídas : [-5 .. +5] Volts, ou [0 .. +5] Volts;
- 8 saídas multiplexadas;
- tempo máximo de retenção das saídas : 50 ms;
- tempo de ajuste : 100 ns
- tempo de estabilização : 30  $\mu$ s.

Dados técnicos da placa E/S :

- 24 portas digitais paralelas programáveis;
- sinais nível TTL.

### 3.4. MICROCOMPUTADOR

Como comentado anteriormente, o software gerenciador vai estar localizado em um microcomputador. A escolha do uso de um microcomputador vem do fato da grande flexibilidade e facilidade que este equipamento oferece para programar, comparado com outros equipamentos programáveis, tais como, microcontroladores e CLP's, além disso, outra importante vantagem associada ao uso de um microcomputador é a possibilidade de plotagem de sinais em seu monitor, possibilitando ao usuário monitorar o funcionamento do sistema bem como permitir o seu melhor entendimento.

Dados técnicos do microcomputador utilizado no projeto :

- IBM PC 486;
- Clock de 33 M Hz.

A figura 3.3 apresenta uma visão geral do arranjo dos equipamentos. Podemos notar que necessitaremos de tantos servoacionamentos (servoconversores, servomotores e encoders) quantos forem o número de *graus de liberdade* do punho do manipulador.

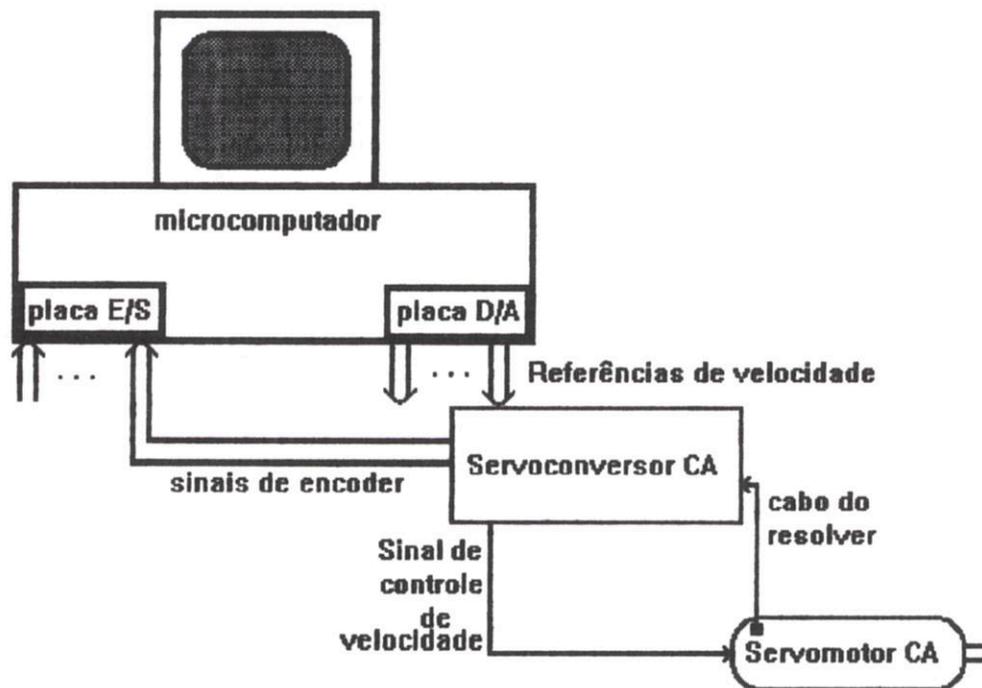


fig.3.3 - Integração dos equipamentos

### 3.5. CONCLUSÃO

Este capítulo apresentou as principais características técnicas e de funcionamento dos equipamentos que foram utilizados neste trabalho.

É importante ressaltar que infelizmente não tivemos a oportunidade de especificar esses equipamentos através de métodos existentes sobre o assunto, e isso pode acarretar na utilização de equipamentos superdimensionados ou, em uma situação mais grave, inviabilizar a implementação prática da estrutura de controle do punho do manipulador.

## CAPÍTULO 4

### SOFTWARE GERENCIADOR

#### 4.1. INTRODUÇÃO

Como mencionado anteriormente, o microcomputador conterá o software que gerenciará todas as atividades responsáveis pelo funcionamento desejado da estrutura de controle. Essas atividades constituem-se, basicamente de, leitura, processamento e envio de sinais, além da plotagem dos mesmos no monitor do microcomputador.

Este capítulo descreve as tarefas desempenhadas pelo software gerenciador, apresentando, em alguns casos, o algoritmo que foi utilizado para implementá-las.

Utilizamos como linguagem de programação o *TURBO C 2.0*. Esta escolha deve-se a experiência adquirida, com essa linguagem, em outros trabalhos realizados anteriormente e também pôr se tratar de uma linguagem modular e de fácil entendimento [Holzner93]. Porém, como veremos a seguir, o módulo de leitura de sinais dos encoders teve que ser implementado em linguagem *ASSEMBLY* devido a necessidade de grande velocidade de processamento [Santos89].

#### 4.2. LEITURA DOS SINAIS DOS ENCODERS

Como visto na figura 3.2, os encoders incrementais tem como saídas dois sinais ( A e B) defasados  $90^\circ$ . Com essa defasagem pode-se verificar o sentido de giro do eixo do servomotor, ou seja, o sinal que for ativado primeiro indica o sentido de giro, horário ou antihorário.

Dependendo do sentido de giro, um contador será incrementado ou decrementado, e o valor desse contador indicará a posição (em número de pulsos) do eixo do servomotor [Rhino90].

##### 4.2.1. Algoritmo de leitura

Para verificar se devemos incrementar ou decrementar o contador de pulsos, implementamos um algoritmo relativamente simples, o qual juntamente com a figura 4.1, está descrito a seguir.

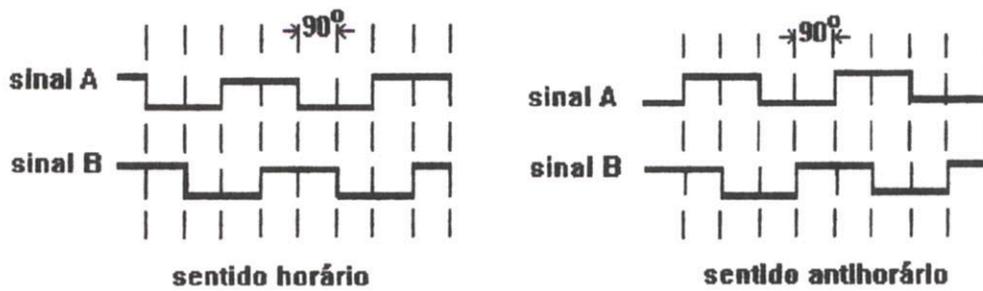


fig.4.1 - Verificação do sentido de giro

Algoritmo:

```

Ler sinal_A e sinal_B

Se sinal_A e sinal_B igual 0
Faz1
    verifica_sentido igual sim
Fim Faz1

Se verifica_sentido igual sim
Faz2
    Se sinal_A igual 1 e sinal_B igual 0
    Faz3
        incrementa contador (sentido horário)
        verifica_sentido igual não
    Fim Faz3

    Se sinal_A igual 0 e sinal_B igual 1
    Faz4
        decrementa contador (sentido antihorário)
        verifica_sentido igual não
    Fim Faz4

Fim Faz2
Fim algoritmo

```

**4.2.2. Problema com a taxa de leitura**

O algoritmo de leitura foi testado com a simulação de sinais de encoder através de um gerador de ondas, e verificamos que a taxa de leitura dos sinais estava extremamente baixa. Com isso teríamos que limitar enormemente a velocidade de giro do eixo do servomotor para que o algoritmo pudesse ler todos os pulsos fornecidos pelo encoder.

Dois fatores poderiam estar diminuindo demasiadamente a taxa de leitura:

#### \* Método de aquisição de dados via polling:

O algoritmo e o driver de leitura de sinais da placa E/S foram primeiramente implementados pelo método polling. Este método de aquisição permite obtermos a menor taxa de leitura dentre os três existentes (polling, interrupção e DMA). Isto ocorre porque este método utiliza a CPU do microcomputador para fazer a aquisição de dados, ocorrendo, desse modo, uma disputa pela CPU para que a mesma realize as tarefas de aquisição de dados e dos outros dispositivos do microcomputador. Essa disputa acarreta em um elevado tempo de espera pelos serviços requisitados à CPU e isso, conseqüentemente, diminui a taxa de leitura da placa E/S [Sargent84].

#### \* Linguagem de programação utilizada:

O algoritmo e o driver de leitura de sinais da placa E/S foram primeiramente implementados em linguagem C. Linguagens de alto nível, como C, gastam mais tempo de processamento se comparado com linguagens de baixo nível, já que linguagens de baixo nível apresentam suas instruções mais próximas do código de máquina [Santos89].

Com esses dois fatores em mente tentamos resolver o problema de baixa taxa de leitura como segue.

#### 4.2.3. Solução

Estudando melhor as características da placa E/S verificamos que esta permite que se faça aquisição de dados apenas pelo método polling.

Portanto, a única alternativa que restou para resolvermos este problema (em termos de programação) foi a implementação do algoritmo e do driver de leitura de sinais em linguagem ASSEMBLY.

Devido a previsão de uso do *harmonic drive* nas transmissões mecânicas do manipulador, decidimos reduzir o número de pulsos por giro tomados dos encoders. Com isso perdemos um pouco de precisão de posicionamento nos eixos dos servomotores, porém, essa perda será inteiramente compensada pelo uso do *harmonic drive* (o *harmonic drive* que será utilizado proporcionará uma redução de 100:1).

Com essa medida, reduziremos a taxa máxima de leitura dos sinais dos encoders que a placa E/S deve realizar para garantir que não haja perda dos sinais. Portanto decidimos reduzir em 10 vezes o número de pulsos por volta tomados dos encoders. Assim, temos:

$$\text{numero\_pulsos\_giro} = \frac{500}{10} = 50$$

#### 4.2.4. Resultados

Com o novo algoritmo e driver implementados em ASSEMBLY verificamos, através do gerador de ondas, que a taxa de leitura aumentou cerca de quatro vezes em relação a taxa anterior, onde:

Taxa anterior  $\cong$  1,4 K Hz (linguagem C);

Taxa desejada = velocidade nominal eixo motor x número pulsos por giro

para

velocidade nominal eixo motor =  $2000 / 60 \cong 33,33$  rps

número pulsos por giro = 50

temos

Taxa desejada  $\cong 1,66$  K Hz

Nova taxa  $\cong 5,5$  K Hz (linguagem ASSEMBLY);

### Observações:

O algoritmo de leitura de sinais foi implementado para ler sinais de até três encoders, pois o punho do manipulador terá três *graus de liberdade*. Portanto, podemos garantir que a nova taxa conseguida com a linguagem ASSEMBLY é suficiente para ler todos os pulsos fornecidos por três encoders com seus respectivos servomotores à velocidade máxima (2000 rpm).

Esses valores de taxa representam, na verdade, o número de pulsos que se consegue ler (taxa anterior e nova taxa) ou se deve ler (taxa desejada) por segundo e não a frequência de leitura atingida pelo microcomputador, já que esta é relativamente maior.

O redutor do número de pulsos por giro dos sinais dos encoders pode ser obtido através da implementação de um hardware divisor de frequência que, basicamente constitui-se de circuitos integrados contadores de pulsos.

Todo o projeto foi feito sem a parte mecânica do manipulador, ou seja, sem o punho e sem o *harmonic drive*. Portanto decidimos continuar adquirindo os 500 pulsos por giro fornecidos pelos encoders para realização dos testes finais de seguimento de trajetória. Porém, esses testes foram realizados com a velocidade máxima do servomotor limitada a um valor que não permitisse ao encoder fornecer uma taxa de pulsos maior que a taxa de leitura conseguida pelo algoritmo (5,5 KHz).

### 4.3. MARCAÇÃO DE PONTOS NO ESPAÇO DE JUNTAS

O objetivo final deste projeto é fazer com que o manipulador atinja pontos (posições) no espaço de trabalho para que o mesmo realize uma determinada tarefa. Porém, o modo pelo qual esses pontos são especificados deve ser o mais fácil possível para facilitar o trabalho de marcação de pontos por parte do usuário.

A técnica empregada visa justamente isso e utilizamos o teclado do microcomputador para implementá-la. Sua descrição está a seguir.

### 4.3.1. Marcação de pontos via teclado

Através do teclado do microcomputador, o usuário escolhe a junta a ser movimentada bem como o sentido do movimento. Posteriormente, utilizando teclas pré determinadas, o usuário movimenta a junta escolhida até a posição desejada.

Com esse procedimento, em seqüência, o usuário pode colocar o efetuator em qualquer posição dentro do volume de trabalho do manipulador, e assim escolher que pontos devem ser gravados para posterior geração da trajetória a ser seguida pelo manipulador.

Assim teremos uma tabela, como mostra a figura 4.2, onde cada junta possui o seu respectivo deslocamento. Podemos ver ainda que foi definido o ponto 0 como "home-position", ou seja, é o ponto de referência no espaço onde os contadores dos encoders são zerados.

JUNTAS	DESLOCAMENTO (graus)					
	homeposition	ponto 1	ponto 2	ponto 3	ponto 4	ponto 5
A	0	360	180	-50	0	-180
B	0	-220	0	470	500	360
C	0	20	100	300	300	-45
⋮	⋮	⋮	⋮	⋮	⋮	⋮
N						

N = número de graus de liberdade

fig.4.2 - exemplo de pontos marcados

Embora fique transparente para o usuário, pois este visa a posição do efetuator do manipulador, esses pontos gravados representam os deslocamentos dos eixos dos servomotores, ou seja, estamos gravando pontos no espaço de juntas.

Este modo de gravação de pontos utiliza o algoritmo de leitura de sinais dos encoders para quantificar os deslocamentos dos eixos dos servomotores.

## 4.4. GERADOR DE TRAJETÓRIA

Nesta sessão descreveremos o método computacional utilizado para gerar uma trajetória no espaço multidimensional, através da qual, definiremos o movimento desejado para o manipulador.

O objetivo é especificar uma determinada trajetória com o mínimo de informações possíveis, permitindo, desse modo, facilitar a tarefa do usuário de levar o manipulador de um ponto a outro dentro do seu respectivo volume de trabalho.

### 4.4.1. Considerações sobre a descrição e geração da trajetória

I. O método de geração da trajetória é computacional, e o mesmo ocorre em tempo real e a uma certa freqüência de amostragem. Em manipuladores típicos essa freqüência de amostragem situa-se na faixa entre 20 e 200 Hz [Craig86].

- II. O movimento do manipulador deve ser “suave”, pois, movimentos bruscos aumentam o desgaste do mecanismo de transmissão e causam vibrações que podem excitar os modos elásticos do sistema. Portanto, para obtermos trajetórias “suaves”, deveremos garantir que as funções que descrevem o movimento sejam contínuas e possuem sua derivada primeira também contínua [Craig86].
- III. Existem vários métodos que podem ser implementados para geração de trajetória [Craig86]. Neste trabalho utilizamos polinômios cúbicos para descrição desses perfis, pois, além de fácil compreensão do seu funcionamento, ele possui a característica de “suavidade” abordada acima. Sua descrição e desenvolvimento estão a seguir.

#### 4.4.2. Geração de Trajetória utilizando Polinômios Cúbicos

Dado que temos pontos marcados no espaço de juntas, desejamos fazer com que o manipulador desenvolva uma trajetória que passe por esses pontos em uma determinada quantidade de tempo.

Portanto, devemos encontrar uma função que descreva esse movimento “suavemente”. Essa função,  $\theta(t)$ , deve possuir pelo menos, quatro características. Duas delas vem da seleção do valor inicial  $\theta(0)$  e final  $\theta(t_s)$ :

$$\theta(0) = \theta_0 \quad \text{eq.1} \quad ; \text{ tempo inicial} = 0$$

$$\theta(t_s) = \theta_f \quad \text{eq.2} \quad ; t_s = \text{tempo de segmento ou tempo de movimentação}$$

As outras duas características são para garantir a continuidade da derivada primeira (velocidade) da função:

$$\dot{\theta}(0) = 0 \quad \text{eq.3}$$

$$\dot{\theta}(t_s) = 0 \quad \text{eq.4}$$

Essas quatro características definem uma função de, no mínimo, terceiro grau. Isso explica o uso de polinômios cúbicos para geração de trajetória, onde:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

assim, para

$$t = 0 \text{ e eq.1,} \quad \theta_0 = a_0 \quad \text{eq.5}$$

para

$$t = t_s \text{ e eq.2,} \quad \theta_f = a_0 + a_1 t_s + a_2 t_s^2 + a_3 t_s^3 \quad \text{eq.6}$$

tendo,  $\dot{\theta}(t) = a_1 + 2a_2t + 3a_3t^2$

para  $t = 0$  e eq.3,  $a_1 = 0$  eq.7

para  $t = t_s$  e eq.4,  $a_1 + 2a_2t_s + 3a_3t_s^2 = 0$  eq.8

Isolando os coeficientes  $a_i$  das eq.5,6,7 e 8 para  $i \in [0..3]$ , obtemos:

$$a_0 = \theta_0$$

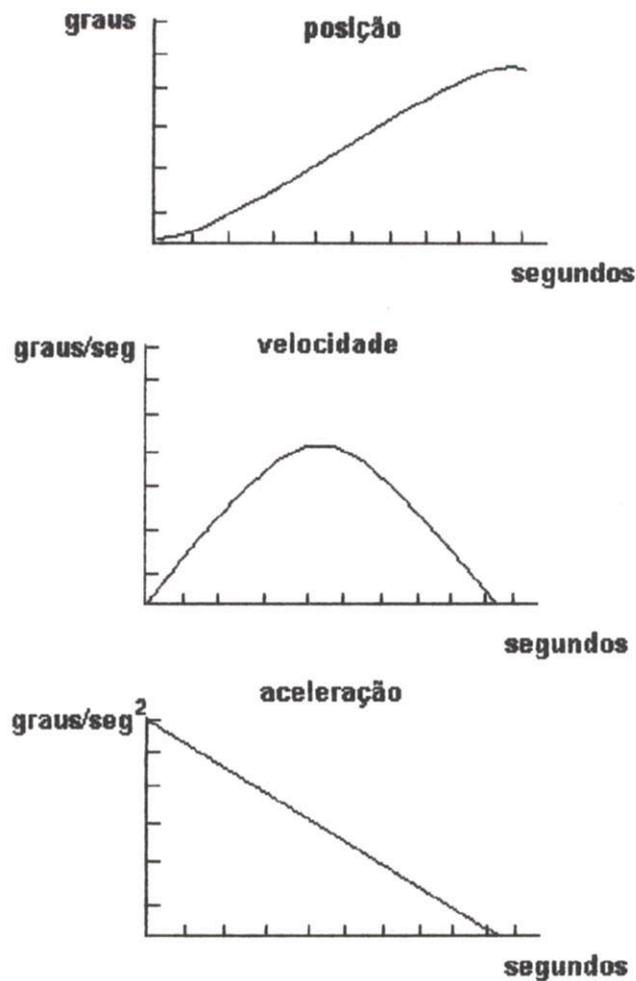
$$a_1 = 0$$

$$a_2 = \frac{3(\theta_f - \theta_0)}{t_s^2}$$

$$a_3 = \frac{-2(\theta_f - \theta_0)}{t_s^3}$$

Através do ponto inicial e final e do tempo de segmento, conseguimos obter os coeficientes do polinômio cúbico que descreve a trajetória de um ponto a outro.

A figura 4.3 mostra as formas das curvas de posição, velocidade e aceleração para um polinômio cúbico deste tipo.



**fig.4.3 - Perfis de posição, velocidade e aceleração para um único segmento**

#### 4.4.3 Pontos Intermediários com especificação de velocidade

Os coeficientes  $a_i$  determinados anteriormente definem um polinômio cúbico com características desejáveis de suavidade, porém esse polinômio impõe velocidade inicial ( $\dot{\theta}(0)$ ) e final ( $\dot{\theta}(t_s)$ ) igual a zero. Assim, dado uma seqüência de pontos marcados pelo usuário, o manipulador terá sempre o seu movimento interrompido em cada ponto e essa característica nem sempre é desejável.

Para resolver este problema, podemos determinar polinômios cúbicos como antes, porém agora a velocidade inicial e final em cada ponto pode ser especificada.

Assim, temos:

$$\dot{\theta}(0) = \dot{\theta}_0$$

$$\dot{\theta}(t_s) = \dot{\theta}_r$$

As quatro equações que descrevem o polinômio são:

$$\theta_0 = a_0$$

$$\theta_f = a_0 + a_1 t_s + a_2 t_s^2 + a_3 t_s^3$$

$$\dot{\theta}_0 = a_1$$

$$\dot{\theta}_f = a_1 + 2a_2 t_s + 3a_3 t_s^2$$

E isolando os  $a_i$  para  $i \in [0..3]$ , obtemos:

$$a_0 = \theta_0$$

$$a_1 = \dot{\theta}_0$$

$$a_2 = \frac{3(\theta_f - \theta_0)}{t_s^2} - \frac{2\dot{\theta}_0}{t_s} - \frac{\dot{\theta}_f}{t_s}$$

$$a_3 = \frac{-2(\theta_f - \theta_0)}{t_s^3} + \frac{(\dot{\theta}_f + \dot{\theta}_0)}{t_s^2}$$

Além da posição inicial e final e do tempo de segmento, deveremos especificar a velocidade inicial e final para definir o polinômio cúbico.

É importante salientar que deveremos calcular um polinômio para cada segmento (onde: número de segmentos = número de pontos - 1). Se pegarmos os pontos exemplo da junta A (fig. 4.2), teremos:

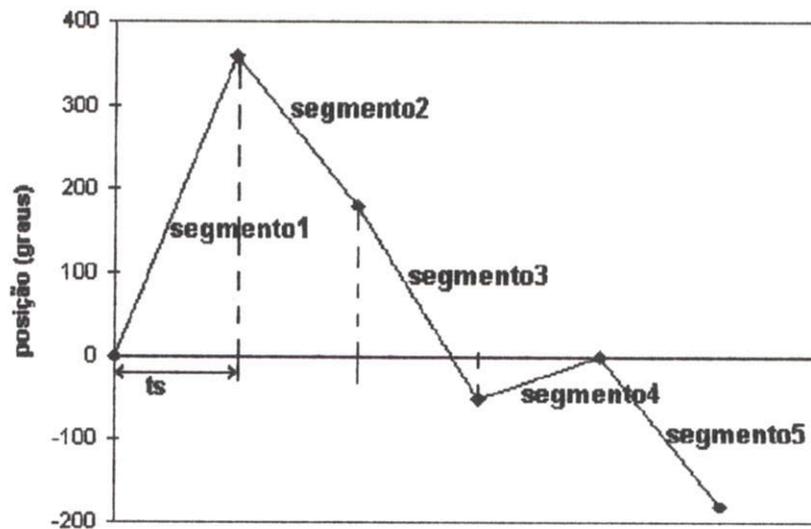


fig.4.4 - Pontos exemplo da junta A

#### 4.4.4. Escolha da velocidade nos pontos marcados

O problema agora é definirmos qual será a velocidade que colocaremos em cada ponto marcado pelo usuário. Sabemos que a velocidade deve ser nula para o ponto inicial e final da trajetória, porém deveremos encontrar a velocidade para os pontos intermediários.

##### 4.4.4.1. Colocando velocidade fixa nos pontos intermediários

Na primeira "versão" do algoritmo de cálculo dos polinômios cúbicos um valor de velocidade constante foi colocado nos pontos intermediários, dependendo apenas, do sentido de seus dois segmentos adjacentes para atribuição do sinal. Assim, se dois segmentos adjacentes possuem o mesmo sentido, a velocidade terá valor positivo ou negativo (sinal será igual aos sentidos) e quando os sentidos mudam, a velocidade será nula. A figura 4.5 apresenta seis pontos exemplo, onde se verifica o sentido do movimento para escolha do sinal da velocidade que será colocada nos pontos intermediários.

Ponto	deslocamento(graus)	sentido	velocidade(graus/seg)
home position	0		= 0 (ponto inicial)
1	45	positivo	> 0
2	70	positivo	= 0
3	10	negativo	< 0
4	-20	negativo	= 0
5	50	positivo	= 0 (ponto final)

fig. 4.5 - Escolha do sinal da velocidade nos pontos intermediários

O algoritmo para o cálculo dos coeficientes  $a_i$  se encontra abaixo.

Inicio algoritmo

Se sentido\_segmento\_anterior difere sentido\_segmento\_posterior

Faz1

velocidade\_ponto\_intermediário igual 0

Fim Faz1

Se não

Faz2

Se sentido\_segmento\_anterior e sentido\_segmento\_posterior maior 0

Faz3

velocidade\_ponto\_intermediário igual velocidade\_positiva

Fim Faz3

Se não

Faz4

velocidade\_ponto\_intermediário igual velocidade\_negativa

Fim Faz4

Fim Faz2

$\hat{\theta}_r = \text{velocidade\_ponto\_intermediário}$

$\hat{\theta}_0 = \text{velocidade\_ponto\_intermediário\_anterior}$

$t_s = \text{valor\_constante}$

$\theta_0 = \theta_{0\_segmento\_anterior}$

$\theta_r = \theta_{r\_segmento\_anterior}$

Calcular coeficientes  $a_i$  para  $i$  [0..3]

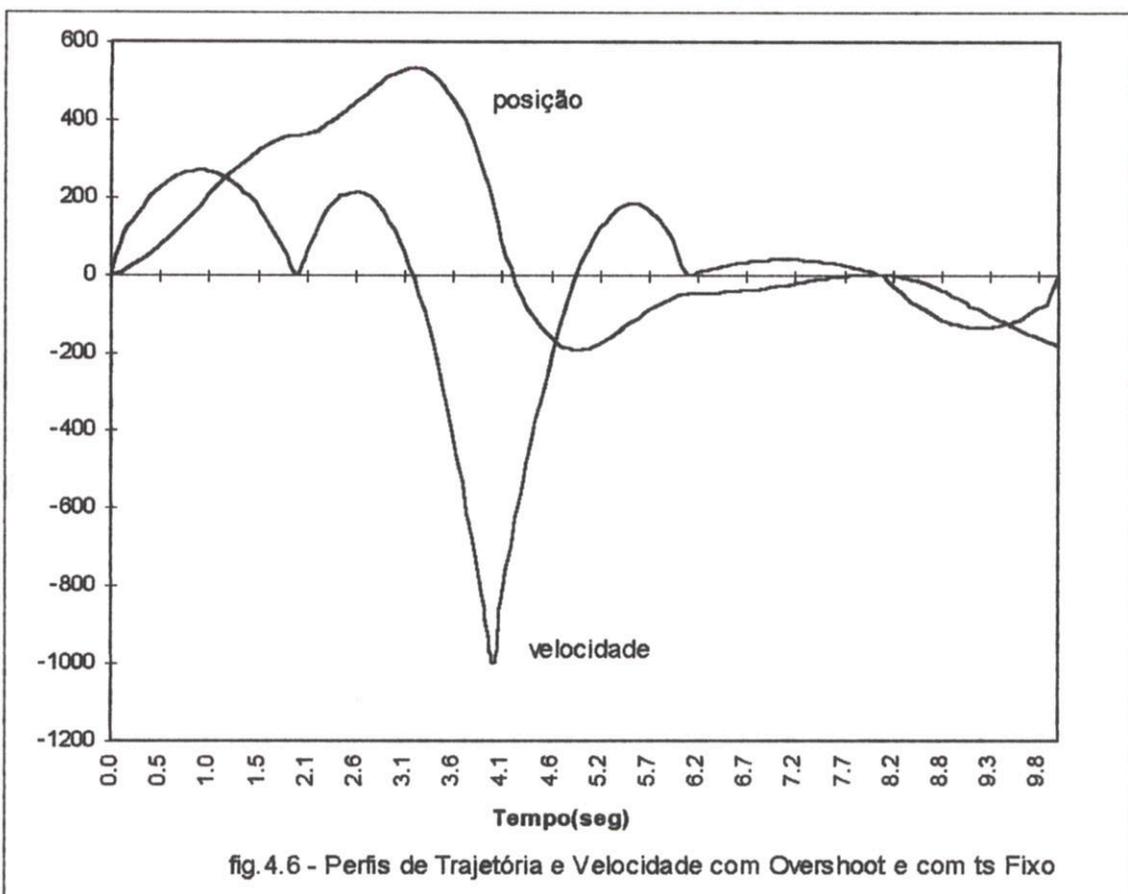
Fim algoritmo

Os pontos exemplo da junta A (figura 4.2) foram utilizados por esse algoritmo para geração dos perfis de trajetória e velocidade. A título de exemplo, atribuímos os seguintes valores:

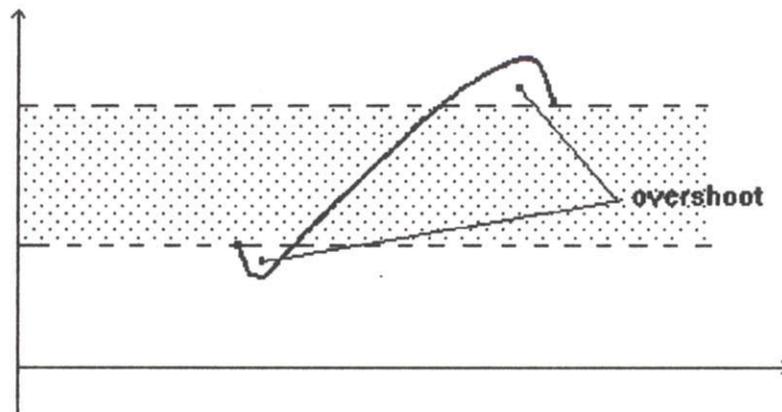
velocidade\_positiva = 1000 graus/seg  
velocidade\_negativa = -1000 graus/seg  
 $t_s = 2$  segundos.

A figura 4.6 apresenta esses perfis, onde:

tempo\_total\_movimento = número\_segmentos  $\times t_s = 10$  segundos.



Verificamos que o perfil de trajetória obtido com o algoritmo anterior, apesar de passar pelos pontos exemplo, apresenta overshoot, ou seja, o perfil contém pontos que estão fora da região limitada pelos seus respectivos pontos intermediários. Veja figura 4.7.



**fig.4.7- Perfil de trajetória com overshoot**

Trajetórias com overshoot são indesejáveis pois estas ficam pouco uniformes, acarretando, com isso, no aumento do desgaste das transmissões mecânicas.

Para resolver este problema modificamos o algoritmo anterior de geração de trajetória. O novo algoritmo atribui valores de velocidade para os pontos intermediários de acordo com o módulo dos segmentos adjacentes. Veja a seguir como isso foi feito.

#### **4.4.4.2. Eliminação do Overshoot dos Perfis de Trajetória**

O perfil de trajetória mostrado na figura 4.6 apresentou overshoot devido ao elevado módulo de velocidade atribuído aos pontos intermediários. Porém, se colocarmos um valor de velocidade muito menor, novamente estaremos prejudicando a uniformidade dos perfis de trajetória e velocidade.

Portanto, devemos encontrar um valor de velocidade "ideal" para cada ponto intermediário. No novo algoritmo, esse valor de velocidade "ideal" depende do módulo dos dois segmentos adjacentes ao ponto intermediário, onde o menor módulo definirá o valor inicial da velocidade. Com esse valor, calcula-se os coeficientes dos polinômios cúbicos. Caso exista algum ponto da trajetória que esteja fora da região limitada pelos dois pontos em questão, o valor inicial da velocidade é decrementado. Este procedimento ocorre até que se encontre os coeficientes que definem uma trajetória sem overshoot.

Para implementação deste algoritmo, utilizamos o período de amostragem da trajetória ( $T$ ) definido pelo usuário, a máxima velocidade permitida e um tempo de movimentação ( $t_s$ ) fixo. O algoritmo encontra-se a seguir.

Início algoritmo

$t_s = \text{valor\_constante}$

$\text{velocidade\_máxima} = \text{velocidade\_máxima\_servomotor}$

Se  $\text{sentido\_segmento\_anterior}$  difere  $\text{sentido\_segmento\_posterior}$

Faz1

$\text{velocidade\_ponto\_intermediário} = 0$

Fim Faz1

Se não

Faz2

Se  $\text{módulo\_segmento\_anterior} < \text{módulo\_segmento\_posterior}$

Faz3

Se  $\text{velocidade\_máxima} > \text{módulo\_segmento\_anterior} / t_s$

Faz4

$\text{velocidade\_máxima} = \text{módulo\_segmento\_anterior} / t_s$

Fim Faz4

$\text{velocidade\_ponto\_intermediário} = \text{velocidade\_máxima}$ .

volta1:

$\dot{\theta}_r = \text{velocidade\_ponto\_intermediário}$

$\dot{\theta}_0 = 0$

$\theta_0 = \theta_{0\_segmento\_anterior}$

$\theta_r = \theta_{r\_segmento\_anterior}$

Calcular coeficientes  $a_i$  para  $i \in [0..3]$

Se

para  $t = t_s - T$  (fim segmento)

$\text{modulo}[a_0 + a_1t + a_2t^2 + a_3t^3] \geq \text{modulo}[\theta_r]$

ou

para  $t = T$  (início segmento)

$\text{modulo}[a_0 + a_1t + a_2t^2 + a_3t^3] \leq \text{modulo}[\theta_0]$

Faz5

$\text{velocidade\_ponto\_intermediário} = \text{velocidade\_ponto\_intermediário} - 1$

retorna volta1

Fim Faz5

Fim Faz3

Se  $\text{módulo\_segmento\_anterior} > \text{módulo\_segmento\_posterior}$

Faz6

Se  $\text{velocidade\_máxima} > \text{módulo\_segmento\_posterior} / t_s$

Faz7

velocidade\_máxima igual módulo\_segmento\_posterior / t<sub>s</sub>  
Fim Faz7

velocidade\_ponto\_intermediário igual velocidade\_máxima.

volta2:

$$\dot{\theta}_f = 0$$

$$\dot{\theta}_0 = \text{velocidade\_ponto\_intermediário}$$

$$\theta_0 = \theta_{0\_segmento\_posterior}$$

$$\theta_f = \theta_{f\_segmento\_posterior}$$

Calcular coeficientes a<sub>i</sub> para i [0..3]

Se

para t = t<sub>s</sub> - T (fim segmento)

$$\text{módulo}[a_0 + a_1t + a_2t^2 + a_3t^3] \geq \text{módulo}[\dot{\theta}_f]$$

ou

para t = T (inicio segmento)

$$\text{módulo}[a_0 + a_1t + a_2t^2 + a_3t^3] \leq \text{módulo}[\dot{\theta}_0]$$

Faz8

velocidade\_ponto\_intermediário = velocidade\_ponto\_intermediário-1

retorna volta2

Fim Faz8

Fim Faz6

Fim Faz2

$$\dot{\theta}_f = \text{velocidade\_ponto\_intermediário}$$

$$\dot{\theta}_0 = \text{velocidade\_ponto\_intermediário\_anterior}$$

$$\theta_0 = \theta_{0\_segmento\_anterior}$$

$$\theta_f = \theta_{f\_segmento\_anterior}$$

Calcular coeficientes a<sub>i</sub> para i [0..3]

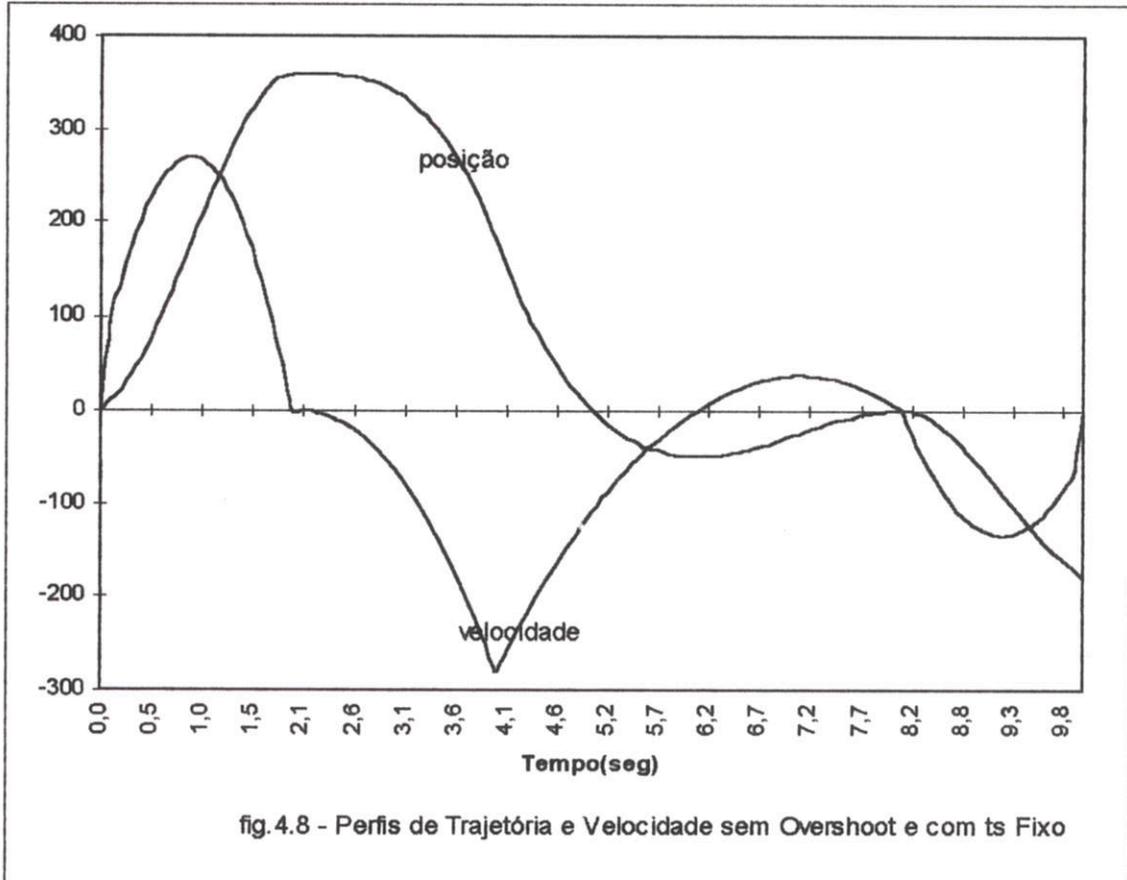
Fim algoritmo

O algoritmo descrito acima coloca a maior velocidade possível no ponto intermediário sendo que esta não ultrapassa a velocidade máxima do servomotor e não ocorre overshoot no perfil de trajetória.

Novamente os pontos exemplo da junta A (fig. 4.2) foram utilizados para geração dos perfis de trajetória e velocidade.

A figura 4.8 apresenta esses perfis, onde:

velocidade\_máxima = 1000 graus/seg  
 $t_s = 2$  segundos  
tempo\_total\_movimento = 10 segundos



Percebemos que não ocorreu overshoot no perfil da trajetória e aumentou a suavidade do perfil de velocidade.

Solucionado o problema de overshoot, procuramos otimizar a geração dos perfis de trajetória em relação ao tempo total de movimentação.

O algoritmo anterior coloca velocidades ideais em cada ponto intermediário para que não ocorra overshoot, porém o tempo de movimentação entre pontos ( $t_s$ ) é fixo. Isso torna o movimento demasiadamente lento quando o módulo do segmento é pequeno e, caso contrário, quando o módulo é grande, a velocidade entre pontos e a aceleração podem ultrapassar seus respectivos valores limite.

Por isso, otimizamos o algoritmo de cálculo dos polinômios cúbicos em relação ao tempo de segmento ( $t_s$ ) levando em conta os valores limites de velocidade e aceleração.

#### 4.4.5. Cálculo da Velocidade Máxima e Aceleração Máxima

O valor da velocidade máxima é dado diretamente pelo parâmetro do servomotor, apenas fizemos uma transformação de unidades (rpm -> graus/seg).

$$\text{velocidade\_maxima} = \frac{\text{velocidade\_nominal}}{2} = \frac{2000\text{rpm}}{2} = 1000\text{rpm}$$

$$\text{velocidade\_maxima} = \frac{1000 \times 360^{\circ}}{60} = 6000 \text{ graus/seg}$$

OBS.: Dividimos a velocidade nominal por 2 devido a saída da placa D/A ser de  $\pm 5\text{Volts}$ .

O valor da aceleração máxima obtém-se através da lei de Newton:

$$T = (J_m + J_s)\ddot{\theta}$$

$T = \text{torque(Nm)}$   
 $J_m = \text{Inercia\_manipulador(Kgm}^2\text{)}$   
 $J_s = \text{Inercia\_servomotor(Kgm}^2\text{)}$   
 $\ddot{\theta} = \text{aceleracao(rad/s}^2\text{)}$

Consideraremos  $J_m = 0$ , pois a estrutura do punho do manipulador ainda não está definida.

Para,

$$T = 2\text{Nm}$$

$$J_s = 0,22 \times 10^{-3}\text{Kgm}^2$$

Temos:

$$\ddot{\theta} = \frac{2}{0,22 \times 10^{-3}} = 9,09 \times 10^3 \text{ rad/s}^2$$

Fazendo a transformação de unidades:

$$\text{aceleracao\_maxima} = \frac{\ddot{\theta} \times 180^{\circ}}{\pi} \cong 0,521 \times 10^6 \text{ graus/s}^2$$

Tendo os valores máximos de velocidade e aceleração, descreveremos a otimização do algoritmo de cálculo dos polinômios cúbicos em relação ao tempo de segmento.

#### 4.4.6. Obtenção do Menor Tempo de Segmento ( $t_s$ )

O objetivo é obtermos os menores tempos de segmento para fazer com que o manipulador realize uma determinada tarefa o mais rápido possível [LIN93].

Porém, teremos que garantir que, em nenhum momento, sejam ultrapassados os valores limites de velocidade e aceleração dos servomotores utilizados.

Antes de descrever a otimização feita no algoritmo de cálculo dos polinômios cúbicos, deve-se verificar duas características que foram levadas em conta para encontrar o tempo de segmento mínimo.

A primeira está relacionada com o perfil de velocidade de um polinômio cúbico. Veja na figura 4.3 que a velocidade máxima (em módulo) se dá na metade de cada segmento. A segunda característica está relacionada com o perfil de aceleração. Também na figura 4.3 percebe-se que a aceleração máxima (em módulo) se encontra nas extremidades dos segmentos.

Essas duas características nos permitem verificar se os perfis de velocidade e aceleração calculados ultrapassam seus respectivos valores limites.

O novo algoritmo difere do anterior nos seguintes aspectos:

(no início):  $t_s = \text{tempo\_mínimo\_segmento}$

velocidade\_maxima = velocidade máxima calculada

aceleracao\_maxima = aceleração máxima calculada

(no fim) : calcular:

para  $t = \frac{t_s}{2}$  (metade do segmento)

velocidade\_maxima\_segmento = modulo[ $a_1 + 2a_2t + 3a_3t^2$ ]

calcular:

para  $t = T$  (início do segmento)

aceleracao\_inicial = modulo[ $2a_2 + 6a_3t$ ]

calcular:

para  $t = t_s - T$  (fim do segmento)

aceleracao\_final = modulo[ $2a_2 + 6a_3t$ ]

Se aceleracao\_inicial maior aceleracao\_final

Faz9

aceleracao\_segmento = aceleracao\_inicial

Fim Faz9

Se não

Faz10

aceleracao\_segmento = aceleracao\_final

Fim Faz10

Se velocidade\_maxima\_segmento maior velocidade\_maxima

ou aceleracao\_segmento maior aceleracao\_maxima

Faz11

$t_s = t_s + \text{incremento}$

voltar para recalcular os  $a_i$  para novo  $t_s$

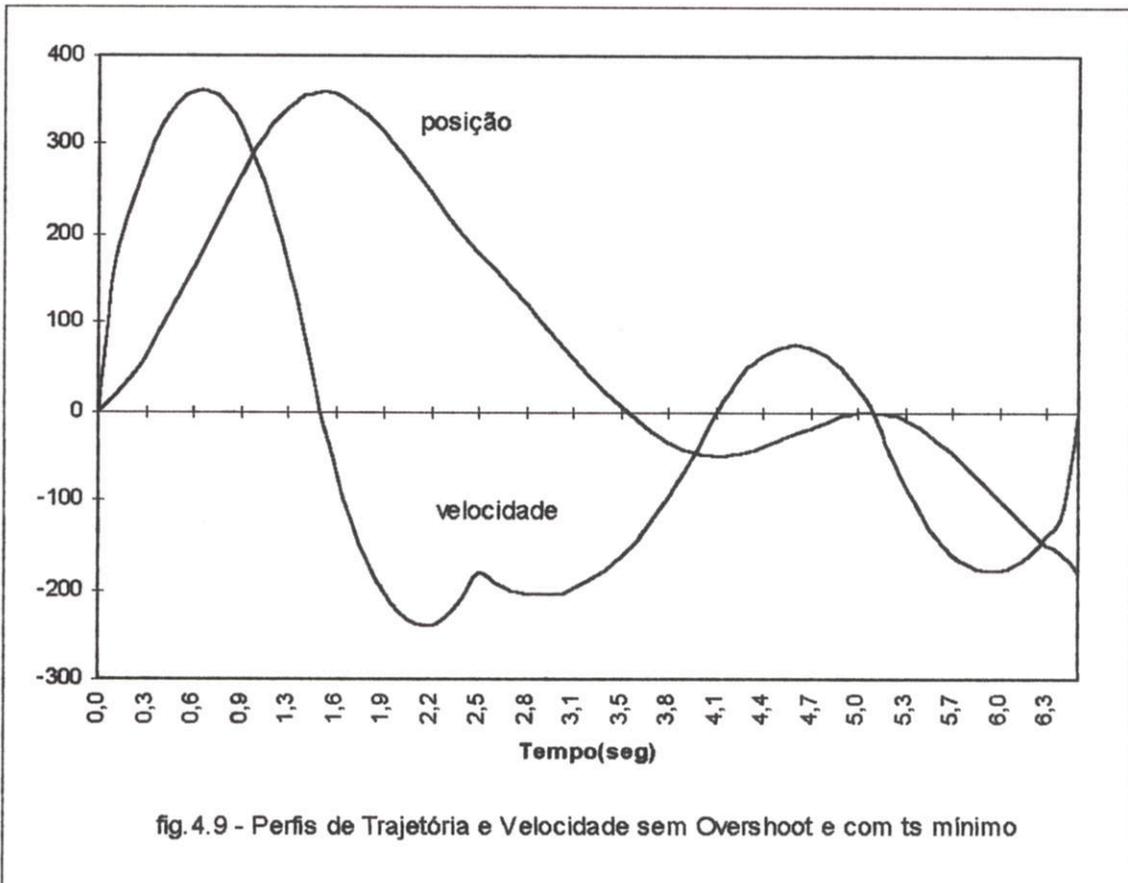
Fim Faz11

A figura 4.9 mostra os perfis de trajetória e velocidade para o algoritmo otimizado em relação ao tempo de segmento.

Para geração desses perfis colocamos:

tempo\_mínimo\_segmento = 1 segundo  
velocidade\_maxima = 1000 graus/seg  
aceleração\_maxima = 1000 graus/seg<sup>2</sup>

tempo\_total\_movimento = 6,5 segundos (encontrado pelo algoritmo)



Veja que o tempo total de movimentação diminuiu bastante, mantendo ainda, a velocidade e aceleração dentro dos valores limite.

#### 4.5. Outras Opções do Software Gerenciador

Além das três tarefas principais que o software gerenciador executa (leitura dos sinais dos encoders, marcação de pontos no espaço de juntas e geração de trajetória), outras duas foram implementadas para incrementar e facilitar o uso do software por parte do usuário. Essas duas opções estão descritas a seguir.

#### **4.5.1. Gravação de Pontos em Arquivo**

Esta função, apesar de simples de ser implementada, é importante, já que permite ao usuário gravar os pontos que geraram a trajetória que o manipulador está seguindo atualmente. Com isso, o usuário pode terminar a execução do software e posteriormente, quando desejar, recuperar tais pontos sem precisar marcá-los novamente.

#### **4.5.2. Plotagem de Sinais**

Esta função plota na tela do microcomputador sinais que são importantes de serem monitorados pelo usuário, como, por exemplo, trajetória a ser seguida pelas juntas do manipulador, real trajetória desenvolvida pelas juntas do manipulador, sinal de controle, etc.

#### **4.6. Visão Geral do Software Gerenciador**

Para finalizar este capítulo, colocamos a figura 4.10 que apresenta todos os módulos do software bem como as interações entre eles.

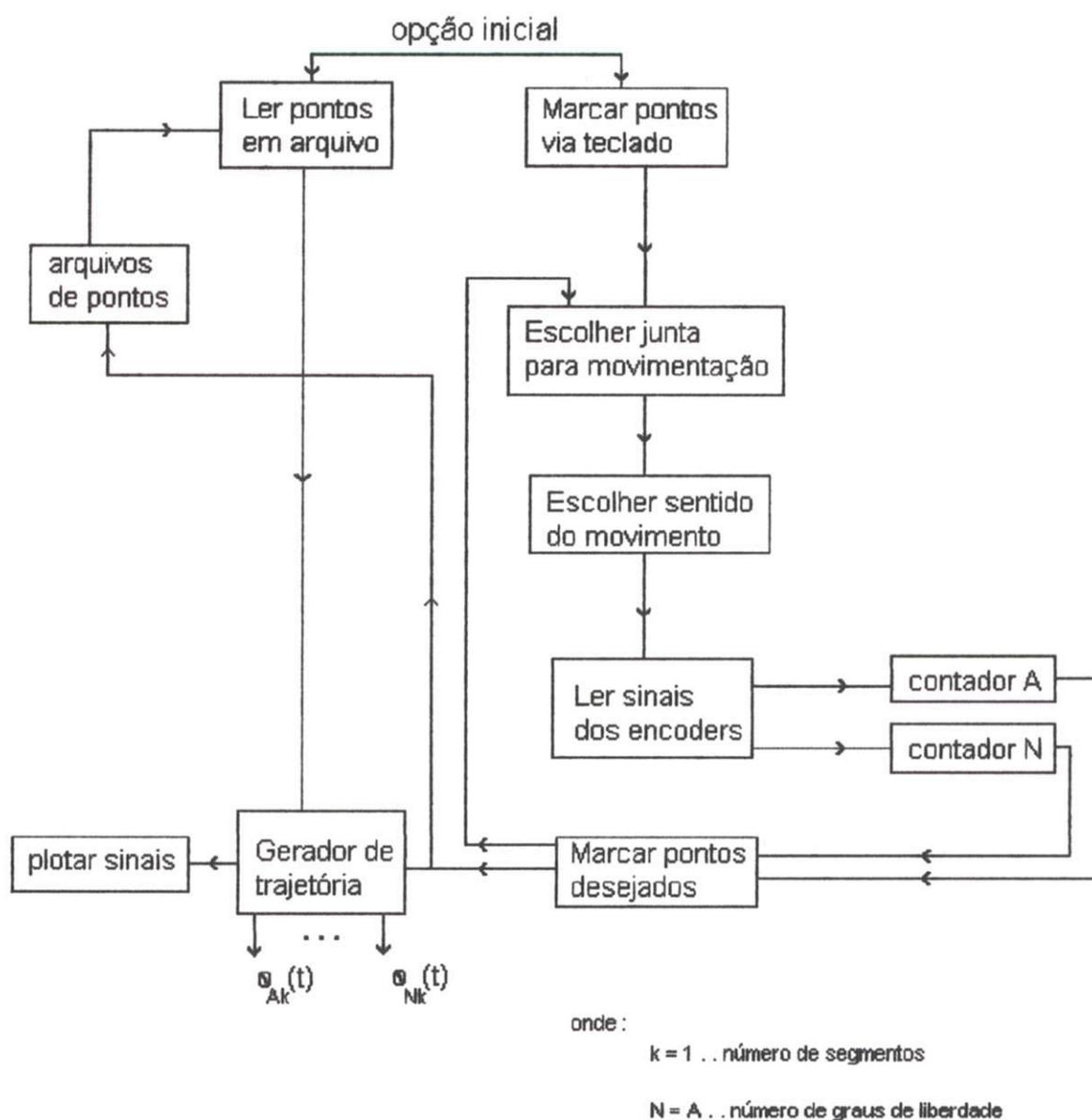


fig.4.10 - Visão geral do software gerenciador

#### 4.7. Conclusão

As atividades descritas neste capítulo consumiram grande parte do tempo de desenvolvimento deste projeto. A tarefa de programação levou tempo devido a necessidade de profundo conhecimento da linguagem que se está utilizando, principalmente na implementação do módulo de leitura de sinais dos encoders em linguagem ASSEMBLY e posterior união deste com os módulos implementados em linguagem C.

Caso a implementação do módulo de leitura de sinais dos encoders em linguagem ASSEMBLY não aumentasse a taxa de leitura a níveis desejáveis, outro caminho poderia ser tomado para solucionar o problema. Esse caminho seria o desenvolvimento de um

hardware que ficaria dedicado á sua tarefa de leitura de pulsos. Com isso a placa E/S faria diretamente a leitura do número de pulsos presente em um dos contadores do hardware, eliminando, deste modo, o tempo gasto pela CPU para processar o algoritmo que conta esses pulsos. Esta alternativa encontra-se em vários catálogos de fabricantes de manipuladores, o que validaria a sua implementação.

Quanto ao módulo gerador de trajetórias, percebemos que, através de um método pouco dispendioso computacionalmente e de fácil compreensão do seu funcionamento, conseguimos obter perfis de trajetória dentro das especificações de suavidade e continuidade. As otimizações feitas para obtermos perfis sem overshoot e com tempo de movimentação mínimo foram apenas para viabilizar seu uso na prática.

## CAPÍTULO 5

### CONTROLE DO SEGUIMENTO DE TRAJETÓRIA

#### 5.1. INTRODUÇÃO

Este capítulo apresenta o método que utilizamos para fazer com que o efetuador do manipulador atinja os pontos marcados pelo usuário através do seguimento dos perfis de trajetória calculados pelo módulo gerador descrito anteriormente.

#### 5.2. CONSIDERAÇÕES SOBRE O CONTROLE DO SEGUIMENTO DE TRAJETÓRIA

⇒ O método de controle do seguimento de trajetória utilizado garante apenas que o efetuador do manipulador irá atingir os pontos marcados pelo usuário, ficando fora do nosso conhecimento o perfil da trajetória que o efetuador vai realizar entre os pontos. Isso ocorre porque a geração da trajetória e o controle do seguimento da mesma foram feitos inteiramente no espaço de juntas, isto é, os perfis de trajetória gerados serão seguidos pelos eixos dos servomotores e o movimento sincronizado destes vai resultar em uma trajetória desconhecida para o efetuador entre os pontos marcados. Veja a figura 5.1.

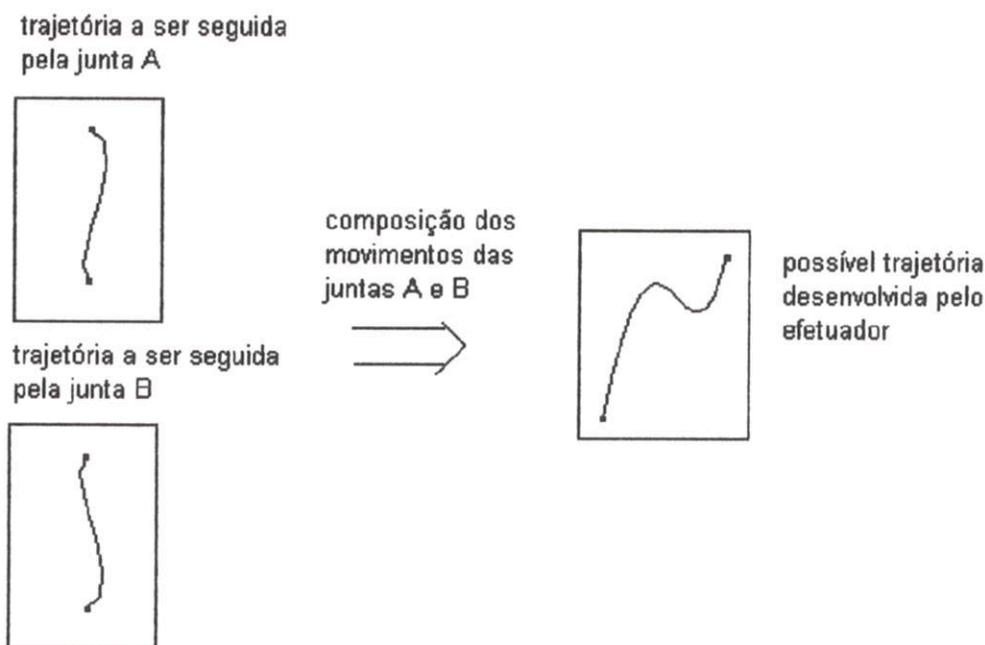


fig.5.1 - Relações entre trajetórias no espaço de juntas e de trabalho

Se desejarmos que o efetuador siga um determinado perfil (reta, parábola, etc.) entre os pontos marcados, teremos que resolver as equações cinemáticas do punho do manipulador. Essas equações transformam pontos do espaço de juntas em pontos no

espaço de trabalho. É importante ressaltar que as equações cinemáticas dependem da configuração mecânica do manipulador e a indefinição da mesma foi um dos motivos da sua não resolução.

⇒ Como mencionado no capítulo 4, valores típicos de freqüência de amostragem da trajetória para manipuladores se encontram entre 20 Hz e 200 Hz. Através dos testes realizados, verificamos que com 50 Hz o sistema funciona perfeitamente. Temos que ter em mente que o valor da freqüência de amostragem utilizado (50 Hz) funcionou bem para o servomotor sem carga, porém o valor ideal da freqüência de amostragem depende também das características estruturais do manipulador.

⇒ Tivemos que programar o timer do microcomputador para estabelecer o valor de freqüência de amostragem utilizado no loop de controle do seguimento de trajetória [Sargent84]. Com isso, podemos deixar a leitura dos sinais dos encoders com freqüência de amostragem máxima, limitada apenas pelo clock do microcomputador.

### 5.3. ESTRUTURA DE CONTROLE

A estrutura de controle montada pode ser visualizada na figura 5.2 [Klafter89].

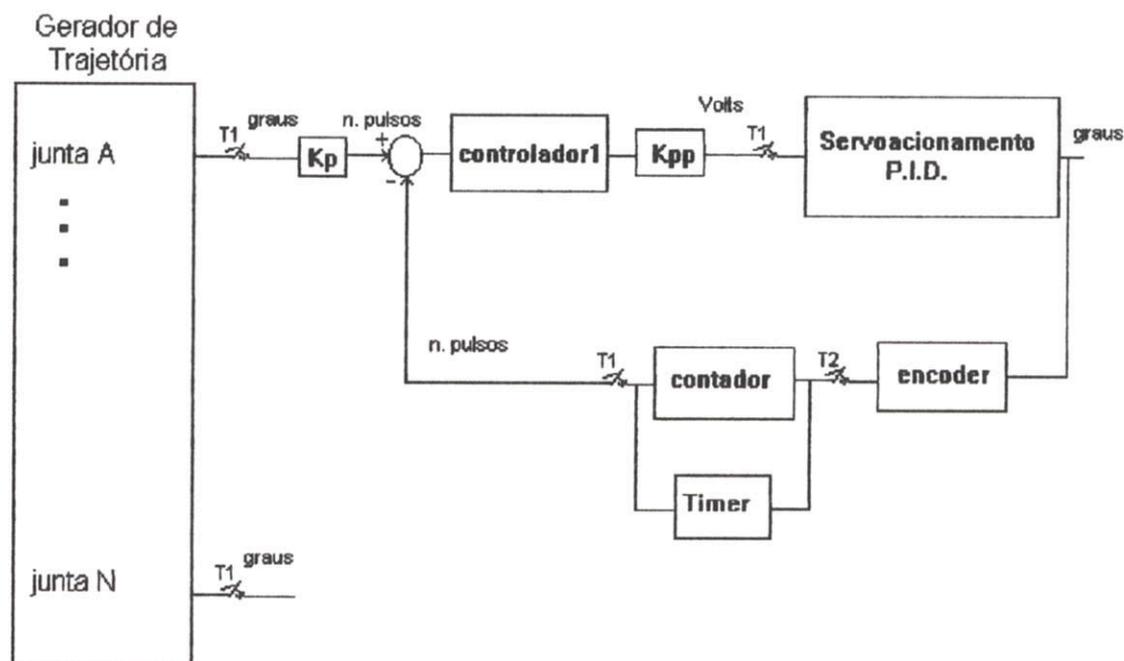


fig.5.2 - Estrutura de controle

A figura 5.2 mostra dois períodos de amostragem ( $T_1$  e  $T_2$ ).  $T_1$  define o período de amostragem dos perfis de trajetória e  $T_2$  define o período de leitura dos sinais dos encoders, onde:

$$T_1 = \frac{1}{50} = 0.02 \text{segundos}$$

$$T_2 = \frac{T_1 \times T_{pc}}{T_1 - T_{pc}}$$

$T_{pc}$  = tempo que o PC gasta para fazer uma leitura dos sinais dos encoders.

$$T_{pc} < T_2 \ll T_1$$

Também podemos observar na figura 5.2 duas constantes,  $K_p$  e  $K_{pp}$ . Essas duas constantes representam ganhos de compatibilização de sinais.

### 5.3.1. Cálculo dos Ganhos $K_p$ e $K_{pp}$

Para calcular os ganhos  $K_p$  e  $K_{pp}$  tomamos os dados técnicos dos equipamentos fornecidos no capítulo 3, onde:

- rotação máxima = 2000 rpm
- sinal de referência de velocidade máxima = 10 Volts
- resolução dos encoders = 0,72 graus/pulso
- número de pulsos por volta = 500

Assim temos:

$$K_p = \frac{1}{\text{resolucao\_encoders}} \cong 1,38889 \text{ pulsos/grau}$$

para, 10 Volts tem-se:

$$2000 \text{ rpm} = 33,33 \text{ rps}$$

$$33,33 \text{ rps} \times 500 \cong 16666,66 \text{ pulsos/seg}$$

para, frequência de amostragem da trajetória = 50 Hz, temos:

$$K_{pp} = \frac{10 \times \text{frequencia\_amostragem\_trajetoria}}{16666,66} \cong 0,03 \text{ Volts/pulso}$$

Com a estrutura de controle definida, temos que escolher a lei de controle que deve ser implementada. A lei de controle mencionada diz respeito somente ao controlador em cascata (controlador1) que aparece na figura 5.2, pois o controlador P.I.D que se encontra dentro do servoacionamento foi ajustado segundo um procedimento indicado no manual do fabricante. Com isso, assumimos inicialmente que temos o seguinte sistema para fazer o controle de posição:

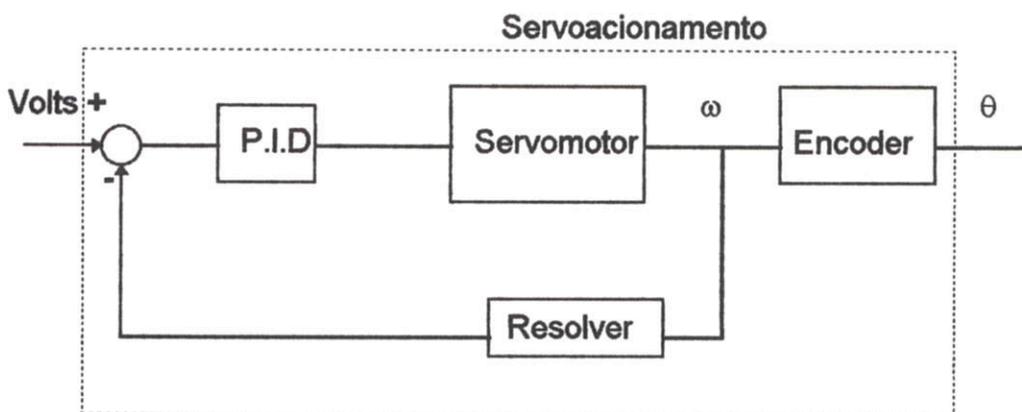


fig.5.3 - Estrutura de Controle de Velocidade

Por facilidade de ajuste, apenas foi colocado um ganho no controlador1, ou seja, o sinal de controle entregue ao servoacionamento é proporcional a diferença entre a posição desejada e a posição do eixo do servomotor em número de pulsos, assim temos:

$$\text{senal\_controle}(KT) = [n\_pulsos\_ref(KT) - n\_pulsos(KT)] \times K_{pp} \times K_C$$

Com a definição de colocar o controlador1 tipo proporcional temos a estrutura de controle mostrada na figura 5.4.

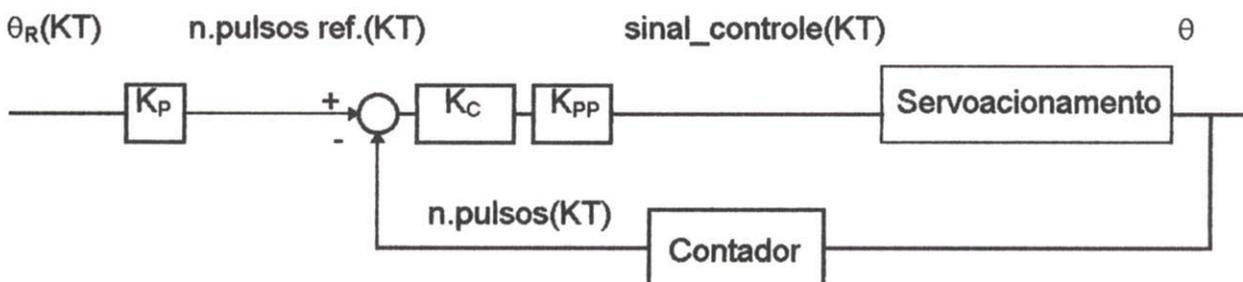


fig. 5.4 - Estrutura de Controle de Posição

Os resultados para diferentes valores de  $K_C$  se encontram a seguir.

#### 5.4. RESULTADOS DO SEGUIMENTO DE TRAJETÓRIAS PARA DIFERENTES VALORES DE $K_C$

Todos os testes de seguimento de trajetória foram realizados sem a estrutura mecânica do punho do manipulador e apenas com um único servoacionamento.

Os gráficos apresentam os perfis da trajetória a ser seguida (trajetória de referencia), a verdadeira trajetória realizada pelo eixo do servomotor e o sinal de controle entregue ao servoacionamento.

Para comparar o seguimento de trajetória para os diferentes valores de  $K_C$ , utilizamos os seis pontos exemplo da junta A para serem atingidos pelo eixo do servomotor, onde:

Ponto	home-position	=	$0^0$
"	1	=	$360^0$
"	2	=	$180^0$
"	3	=	$-50^0$
"	4	=	$0^0$
"	5	=	$-180^0$

utilizamos ainda:

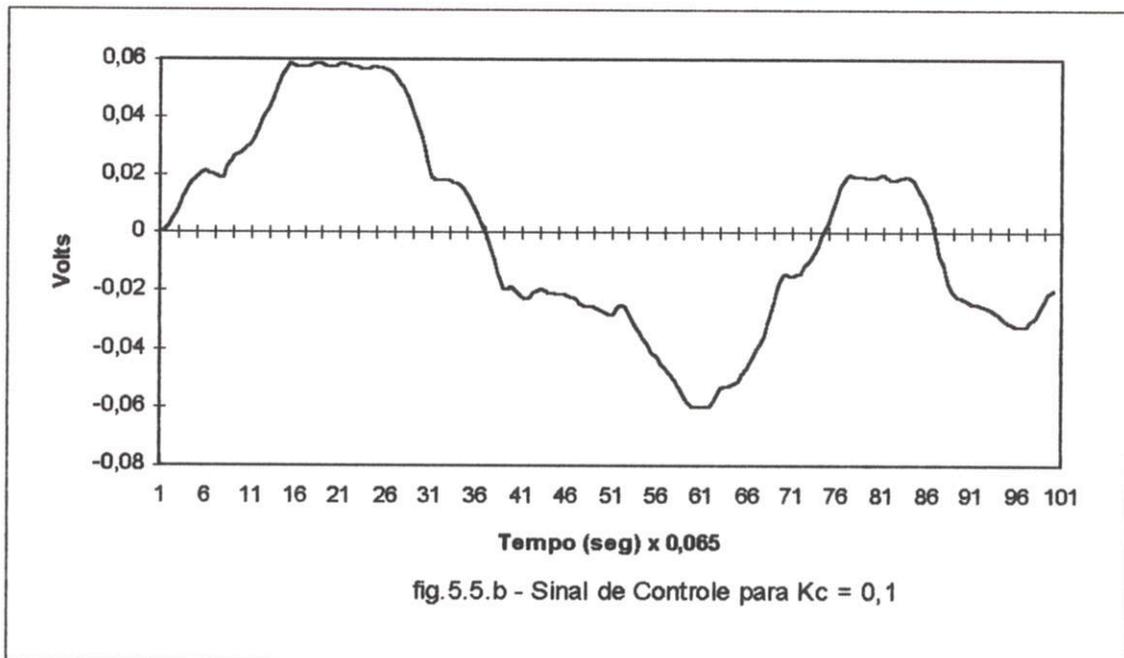
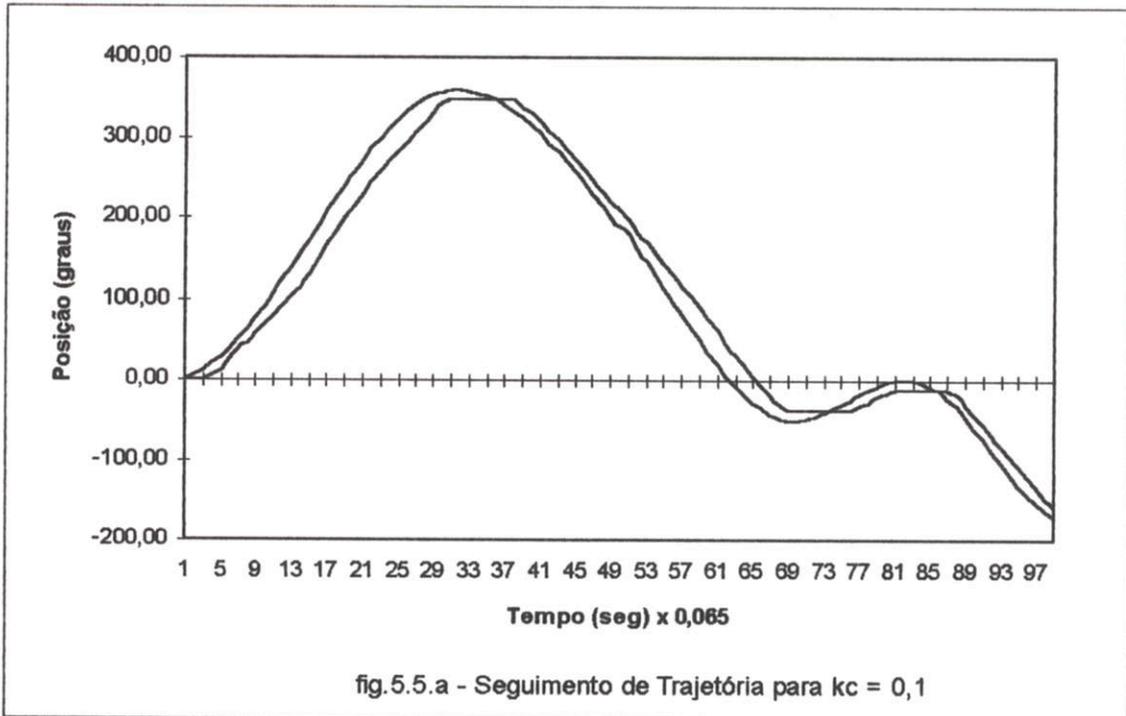
freqüência de amostragem da trajetória = 50 Hz

$$K_p = 1,38889 \text{ pulsos/gra}u$$

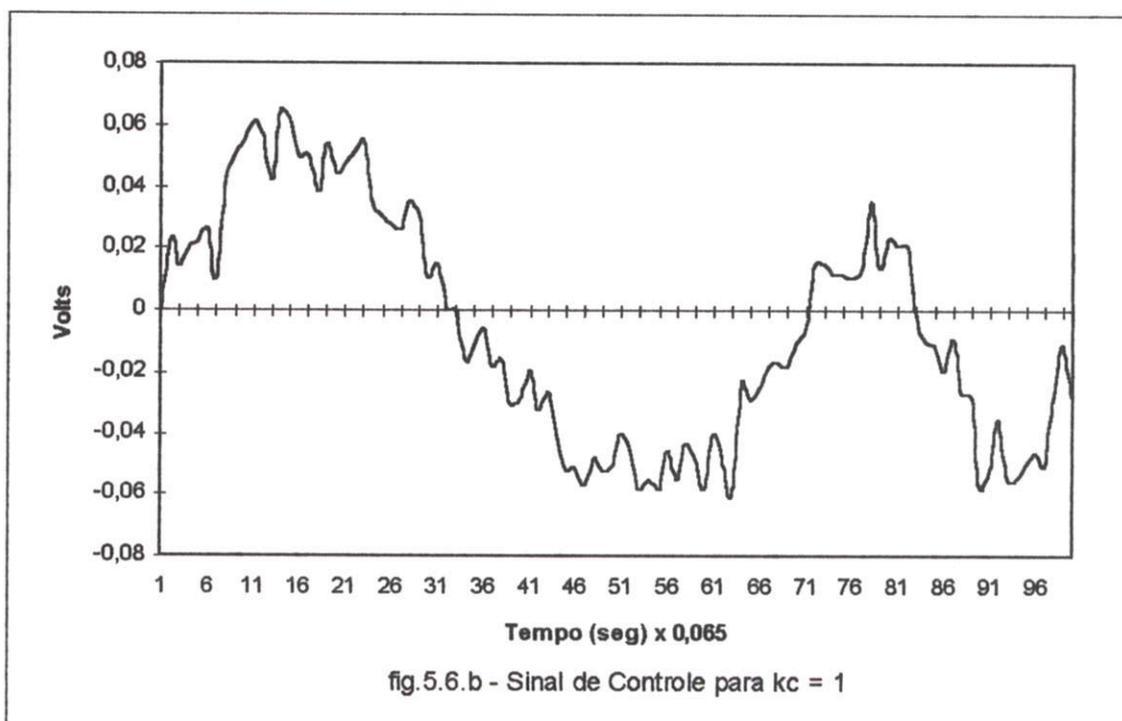
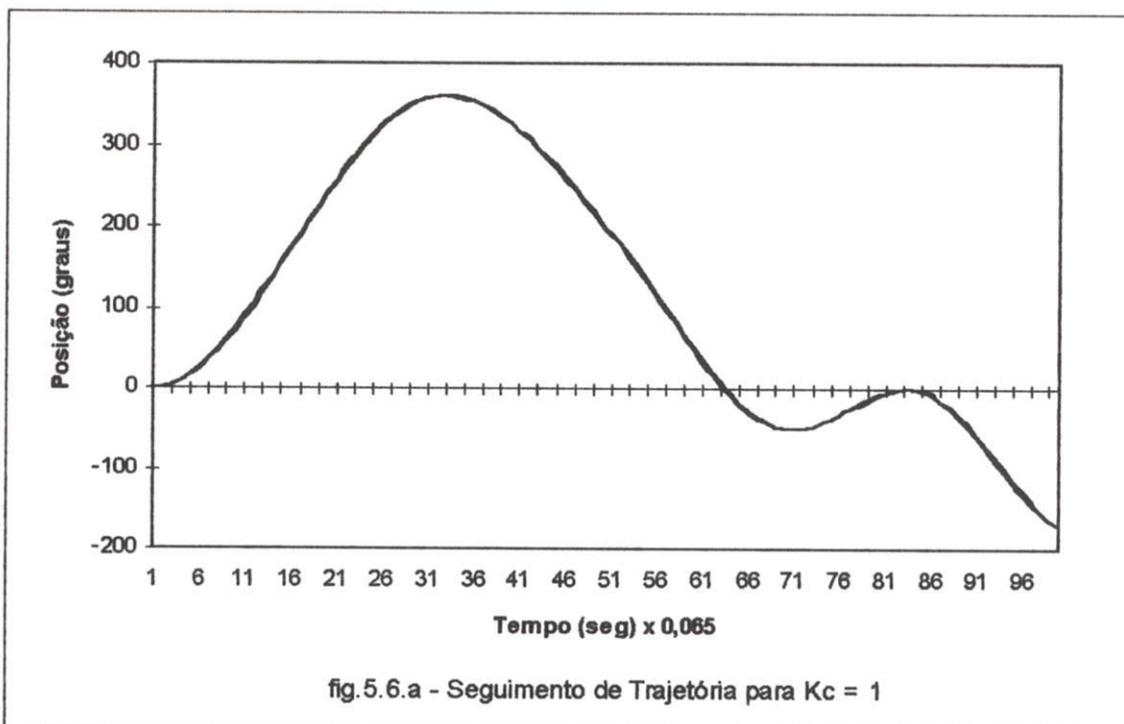
$$K_{pp} = 0,03 \text{ Volts/pulso}$$

Inicialmente definimos  $K_C = 0,1$ .

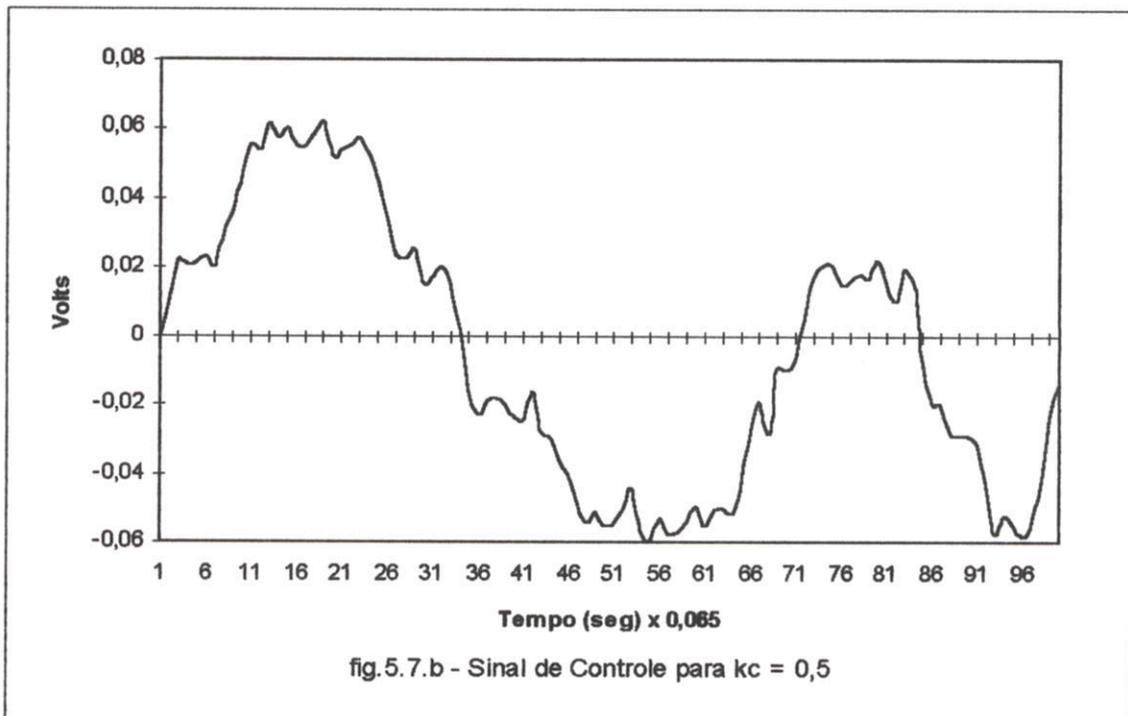
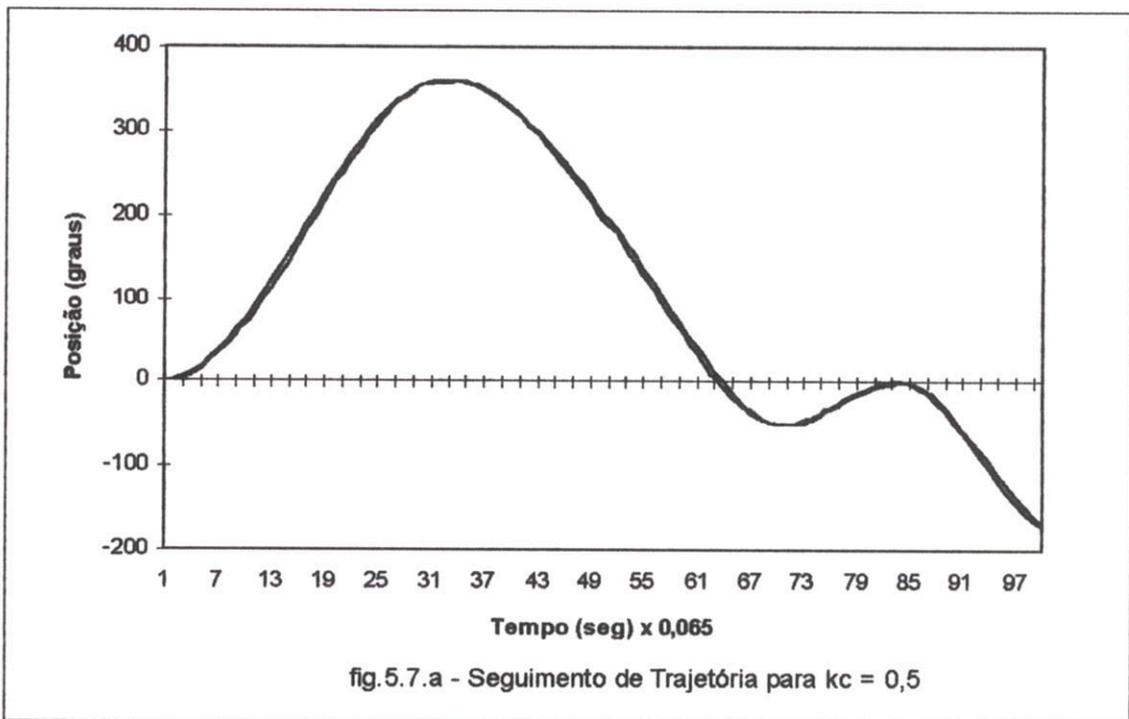
Veja nas figuras 5.5.a e 5.5.b os gráficos de seguimento de trajetória e sinal de controle para este valor de  $K_C$ .



Podemos observar na figura 5.5.a que a trajetória realizada pelo eixo do servomotor em muito difere da trajetória de referência. Para tentar resolver este problema, aumentamos o valor de  $K_C$  com a intenção de diminuir o erro existente entre as trajetórias. Fizemos  $K_C = 1$  e os resultados se encontram nas figuras 5.6.a e 5.6.b.



Agora o perfil da trajetória realizada pelo eixo do servomotor está praticamente sobreposto ao perfil da trajetória de referência, porém, a "suavidade" do sinal de controle ficou bastante comprometida. Por isso, reduzimos o valor de  $K_C$  pela metade, ou seja,  $K_C=0,5$ . As figuras 5.7.a e 5.7.b mostram os gráficos de seguimento de trajetória e sinal de controle para este novo valor de  $K_C$ .



Para  $K_C = 0,5$  podemos observar que a “suavidade” do sinal de controle melhorou bastante (figura 5.7.b) sem comprometer, com isso, a trajetória realizada pelo eixo do servomotor (figura 5.7.a).

Mesmo tendo melhorado a “suavidade” do sinal de controle, esta, ainda, pode vir a prejudicar as partes mecânicas do manipulador, pois é coerente de se pensar que quanto

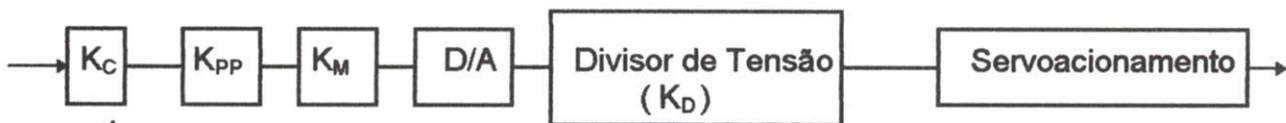
menos “suave” for o sinal de controle, maior será o desgaste das transmissões mecânicas devido a grande quantidade de vibração e atrito.

A seguir está descrito como tentamos resolver este problema.

### 5.5. AUMENTO DA SUAVIDADE DO SINAL DE CONTROLE

A pouca suavidade do sinal de controle, neste caso, está ocorrendo devido a baixa resolução da placa D/A (0,04 Volts) e também porque a faixa do sinal de controle se situa próximo a resolução da mesma.

Para resolver este problema aumentamos a faixa do sinal de controle para que este fique relativamente maior que a resolução da placa D/A. Isto foi feito multiplicando por uma constante ( $> 1$ ) o sinal de controle digital que é entregue a placa e posteriormente a ela, no domínio analógico, dividindo pela mesma constante o sinal de saída da placa. Este sinal de saída é entregue ao servoacionamento. A figura 5.8 mostra esta configuração.



onde:

$K_M$  = constante de multiplicação

$K_D$  = constante de divisão

$$K_D = \frac{1}{K_M}$$

fig.5.8 - Sistema para Aumentar a Suavidade do Sinal de Controle

A figura 5.9 apresenta a configuração do divisor de tensão utilizado

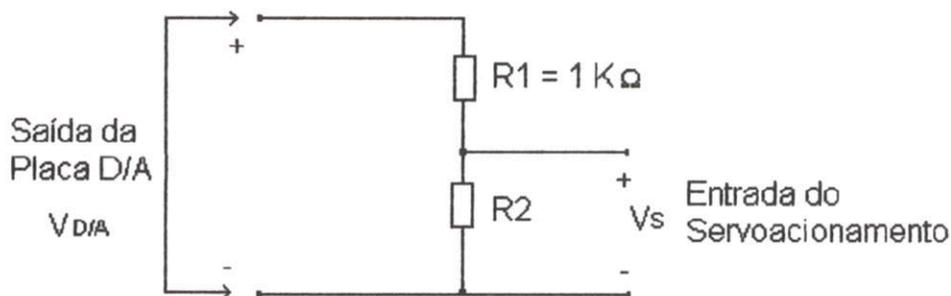


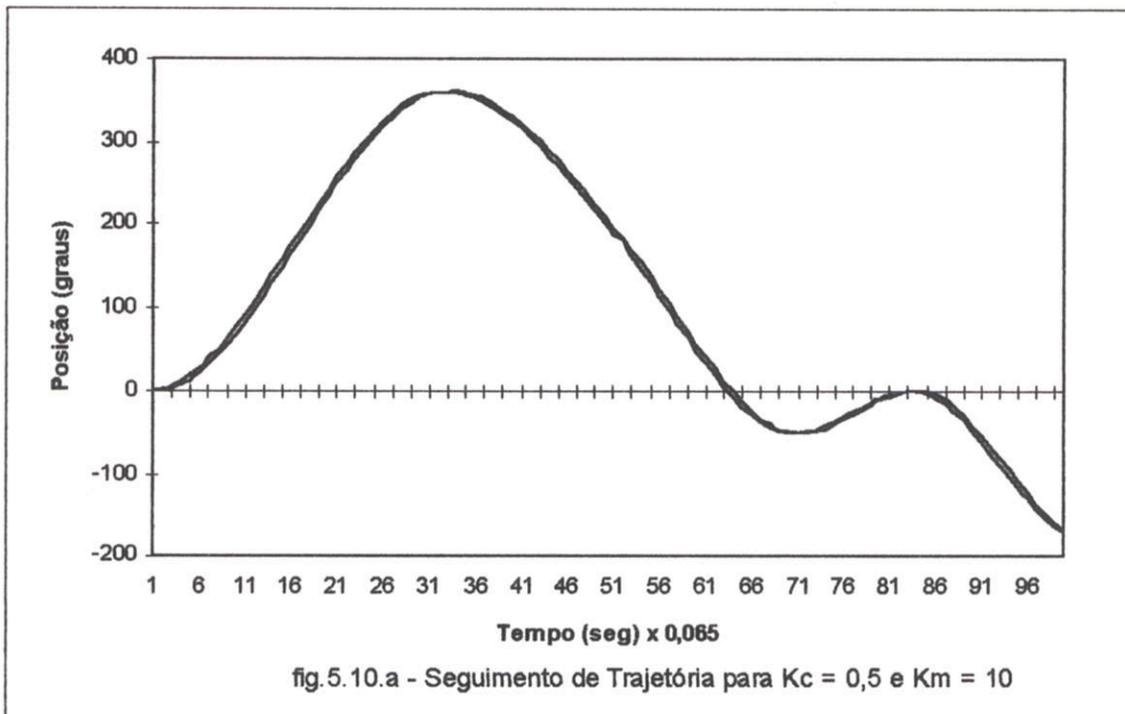
fig.5.9 - Divisor de Tensão Utilizado

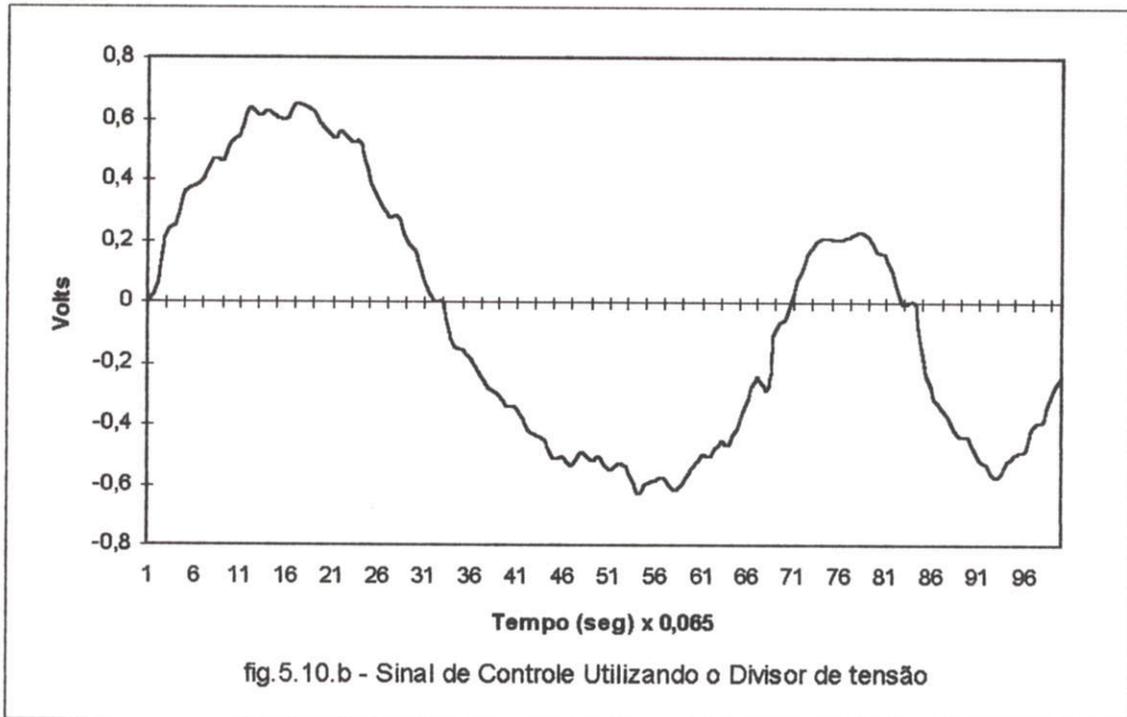
Onde,

$$\frac{V_s}{V_{D/A}} = \frac{R_2}{R_1 + R_2}$$

Colocamos  $K_M = 10$  (divisor de tensão por 10), logo  $R_2 = 111,11\Omega$ .

As figuras 5.10.a e 5.10.b contem os gráficos de seguimento de trajetória e sinal de controle para  $K_C = 0,5$  e com o sistema multiplicador e divisor de tensão.





Podemos observar que a trajetória realizada pelo eixo do servomotor (figura 5.10.a) permaneceu bastante próxima da trajetória de referência e o sinal de controle (figura 5.10.b) apresenta-se bem mais suave se comparado com a figura 5.7.b.

## 5.6. CONCLUSÃO

Verificamos que com apenas um controlador do tipo proporcional (fora o P.I.D do servoacionamento) conseguimos obter uma resposta de seguimento de trajetória bastante desejável, porém, caso o ganho  $K_C$  não fosse suficiente para obtermos tal resposta, teríamos que obter a função transferência do sistema e através de métodos clássicos implementar um controlador mais "poderoso" com características integrativas e/ou derivativas. Esta hipótese pode ainda vir a ser estudada caso ocorra problemas de ação de controle ou seguimento de trajetória quando a estrutura mecânica do punho do manipulador for acoplada aos servoacionamentos.

Com relação ao sistema multiplicador e divisor de tensão apresentado para solucionar o problema de suavidade do sinal de controle, na prática não funcionará perfeitamente, pois o sinal de controle facilmente atingirá o nível de saturação da placa D/A ( $\pm 5$ Volts) devido a constante de multiplicação  $K_M$ . Porém, podemos concluir que a solução para o problema de suavidade do sinal de controle esta no uso de uma placa D/A de melhor resolução.

## CAPÍTULO 6

### CONCLUSÃO

Como previsto, a interdisciplinaridade do projeto foi evidenciada pela diferenciação das atividades desenvolvidas. Conhecimentos em programação de alto e baixo nível, técnicas de aquisição de sinais, técnicas de geração de trajetórias para manipuladores e teoria de controle de sistemas realimentados foram as principais exigências para conseguirmos alcançar bons resultados neste projeto.

Por problemas diversos, não podemos contar com a parte mecânica do punho do manipulador, o que nos deixa na expectativa de como reagirá a estrutura de controle quando desta for exigida que funcione com cargas, atritos e torques variáveis.

Sem a parte mecânica também não podemos implementar a geração de trajetória para o espaço de trabalho, pois faltou dados para definição e resolução das equações cinemáticas do punho do manipulador, e mesmo que pudéssemos obter tais equações sem a parte mecânica pronta, não poderíamos testar o seguimento de trajetória sobre o sistema real, e isso torna sem sentido a realização desta tarefa.

O projeto total de implementação de um manipulador vai exigir que muito mais tarefas sejam realizadas, porém, temos a consciência de que atingimos um patamar importante em termos de conhecimentos, e isso, com certeza, ajudará a enfrentar os problemas que surgirão na continuidade dos trabalhos.

## CAPÍTULO 7

### PERSPECTIVAS

Como mencionado anteriormente, as atividades desenvolvidas neste projeto representam apenas o início de todo o conjunto de tarefas que devem ser implementadas para se conseguir obter o manipulador com seis *graus de liberdade* funcionando.

Portanto as perspectivas para este trabalho são:

- Teste da estrutura de controle sobre o sistema físico real, ou seja, verificar o seguimento de trajetória dos servomotores com estes acoplados á estrutura mecânica do punho do manipulador.
- Implementar outra técnica de geração de trajetória para realizar uma análise comparativa com a técnica de polinômios cúbicos utilizada neste trabalho.
- Implementação da geração e controle de seguimento de trajetória no *espaço de trabalho* para que se possa definir o tipo de trajetória que o punho do manipulador realizará entre pontos consecutivos.
- Implementação do algoritmo que coloca o manipulador na posição de home-position, atribuindo assim, uma posição de referência para o manipulador no *espaço de juntas*.

## BIBLIOGRAFIA

[Klafter89]

R. D. Klafter e M. Negin, "Robotic Engineering on Integrated Approach", Prentice Hall Internacional Editions, 1989.

[Craig86]

J. J. Craig, "Introduction to Robotics Mechanics & Control", Addison Wesley, 1986.

[Holzner93]

S. Holzner e The Peter Norton Computing Group, "C Programação: Guia Prático para Programação Eficiente", Editora Campos, Rio de Janeiro, 1993.

[Santos89]

J. P. Santos e E. R. Júnior, "Programando em Assembler 8086/8088", McGraw-Hill, São Paulo, 1989.

[Sargent84]

M. Sargent e R. L. Shoemaker, "The IBM Personal Computer From the Inside Out", Addison Wesley Publishing Company, Massachusetts, 1984.

[LIN93]

C. S. Lin, P. R. Chang e J. Y. S. Luh, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots", IEEE Trans. Automat. Contr. , vol. AC-28, No 12, pp. 1066-1073, December 1993.

[Rhino90]

Mark IV Owner's Manual, version 4.01, Rhino Robots, Inc, 1990.